

Oracle® Database

SQL 言語クイック・リファレンス

19c

F16135-12(原本部品番号:E96311-16)

2023年11月

タイトルおよび著作権情報

Oracle Database SQL言語クイック・リファレンス, 19c

F16135-12

[Copyright ©](#) 2003, 2023, Oracle and/or its affiliates.

原著者: Usha Krishnamurthy

目次

- [タイトルおよび著作権情報](#)
- [はじめに](#)
 - [対象読者](#)
 - [ドキュメントのアクセシビリティ](#)
 - [関連ドキュメント](#)
 - [表記規則](#)
- [1 SQL文](#)
 - [SQL文の構文](#)
- [2 SQLファンクション](#)
 - [SQLファンクションの構文](#)
- [3 SQL式](#)
 - [SQL式のタイプの構文](#)
- [4 SQL条件](#)
 - [SQL条件のタイプの構文](#)
- [5 副次句](#)
 - [副次句の構文](#)
- [6 データ型](#)
 - [データ型の概要](#)
 - [Oracle組込みデータ型](#)
 - [Oracleが提供するデータ型](#)
 - [Oracleデータ型への変換](#)
- [7 書式モデル](#)
 - [書式モデルの概要](#)
 - [数値書式モデル](#)
 - [数値書式の要素](#)
 - [日時書式モデル](#)
 - [日時書式要素](#)
- [A SQL*Plusコマンド](#)
 - [SQL*Plusコマンド](#)
- [索引](#)

はじめに

このマニュアルでは、Oracle Databaseの情報を管理するために使用されるStructured Query Language(SQL)について説明します。Oracle SQLは、ANSI(米国規格協会)およびISO(国際標準化機構)のSQL:2011規格にさらに多くの内容を盛り込んでいます。

ここでは、次の項目について説明します。

- [対象読者](#)
- [ドキュメントのアクセシビリティ](#)
- [関連ドキュメント](#)
- [表記規則](#)

対象読者

『Oracle Database SQL言語クイック・リファレンス』は、すべてのOracle SQLユーザーを対象としています。

ドキュメントのアクセシビリティ

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWebサイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>)を参照してください。

Oracleサポートへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>)か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>)を参照してください。

関連ドキュメント

詳細は、次のOracleドキュメントを参照してください。

- PL/SQL、Oracle SQLに対する手続き型言語の拡張の詳細は、[『Oracle Database PL/SQL言語リファレンス』](#)を参照してください。
- Oracle埋込みSQLの詳細は、[『Pro*C/C++プログラマーズ・ガイド』](#)および[『Pro*COBOLプログラマーズ・ガイド』](#)を参照してください。

このマニュアルの例の多くは、Oracle Databaseのインストール時に基本インストール・オプションを選択した場合にデフォルトでインストールされるサンプル・スキーマを使用しています。これらのスキーマがどのように作成されているか、およびその使用方法については、[『Oracle Databaseサンプル・スキーマ』](#)を参照してください。

表記規則

このドキュメントでは、次のテキスト表記規則が使用されます：

規則	意味
----	----

規則	意味
boldface	太字体は、アクションに関連付けられたグラフィカル・ユーザー・インタフェース要素や、本文または用語集で定義されている用語を示します。
italic	イタリック体は、ブック・タイトル、強調、またはユーザーが特定の値を指定するプレースホルダー変数を示します。
monospace	等幅体は、段落内のコマンド、URL、サンプル内のコード、画面に表示されるテキスト、またはユーザーが入力するテキストを示します。

1 SQL文

この章では、Oracle SQL文の構文について説明します。

この章の内容は次のとおりです。

- [SQL文の構文](#)

SQL文の構文

SQL文は、プログラムおよびユーザーがOracle Databaseのデータにアクセスするための手段です。

この項では、各SQL文および関連する構文を示します。副次句の構文については、[「副次句」](#)を参照してください。

関連項目:

SQL文の詳細は、[Oracle Database SQL言語リファレンス](#)を参照してください。

[ADMINISTER KEY MANAGEMENT](#)

```
ADMINISTER KEY MANAGEMENT
{ keystore_management_clauses
| key_management_clauses
| secret_management_clauses
} ;
```

[ALTER ANALYTIC VIEW](#)

```
ALTER ANALYTIC VIEW [ schema. ] analytic_view_name
{ RENAME TO new_av_name | COMPILE };
```

[ALTER ATTRIBUTE DIMENSION](#)

```
ALTER ATTRIBUTE DIMENSION [ schema. ]
attr_dim_name { RENAME TO new_attr_dim_name | COMPILE };
```

[ALTER AUDIT POLICY](#)

```
ALTER AUDIT POLICY policy
[ ADD [ privilege_audit_clause ] [ action_audit_clause ] [ role_audit_clause ] ]
[ DROP [ privilege_audit_clause ] [ action_audit_clause ] [ role_audit_clause ] ]
[ CONDITION { DROP | 'audit_condition'
EVALUATE PER { STATEMENT | SESSION | INSTANCE } } ]
[ ONLY TOPLEVEL ]
;
```

[ALTER CLUSTER](#)

```
ALTER CLUSTER [ schema. ] cluster
{ physical_attributes_clause
| SIZE size_clause
| [ MODIFY PARTITION partition ] allocate_extent_clause
| deallocate_unused_clause
| { CACHE | NOCACHE }
} ...
[ parallel_clause ] ;
```

[ALTER DATABASE](#)

```
ALTER DATABASE [ database ]
{ startup_clauses
| recovery_clauses
| database_file_clauses
| logfile_clauses
| controlfile_clauses
| standby_database_clauses
| default_settings_clauses
| instance_clauses
| security_clause
| prepare_clause
| drop_mirror_copy
| lost_write_protection
| cdb_fleet_clauses
| property_clause
} ;
```

[ALTER DATABASE DICTIONARY](#)

```
ALTER DATABASE DICTIONARY
{ ENCRYPT CREDENTIALS
| REKEY CREDENTIALS
| DELETE CREDENTIALS KEY
};
```

[ALTER DATABASE LINK](#)

```
ALTER [ SHARED ] [ PUBLIC ] DATABASE LINK dblink
{ CONNECT TO user IDENTIFIED BY password [ dblink_authentication ]
| dblink_authentication
};
```

[ALTER DIMENSION](#)

```
ALTER DIMENSION [ schema. ] dimension
{ ADD { level_clause
| hierarchy_clause
| attribute_clause
| extended_attribute_clause
}
} ...
|
{ DROP { LEVEL level [ RESTRICT | CASCADE ]
| HIERARCHY hierarchy
| ATTRIBUTE attribute [ LEVEL level [ COLUMN column ] ]...
}
} ...
|
COMPILE
;
```

[ALTER DISKGROUP](#)

```
ALTER DISKGROUP
{ diskgroup_name
{ { { add_disk_clause | drop_disk_clause }
| [, { add_disk_clause | drop_disk_clause } ]...
| resize_disk_clause
} [ rebalance_diskgroup_clause ]
| replace_disk_clause
| rename_disk_clause
| disk_online_clause
| disk_offline_clause
| rebalance_diskgroup_clause
| check_diskgroup_clause
```

```

| diskgroup_template_clauses
| diskgroup_directory_clauses
| diskgroup_alias_clauses
| diskgroup_volume_clauses
| diskgroup_attributes
| modify_diskgroup_file
| drop_diskgroup_file_clause
| convert_redundancy_clause
| usergroup_clauses
| user_clauses
| file_permissions_clause
| file_owner_clause
| scrub_clause
| quotagroup_clauses
| filegroup_clauses
}
| { diskgroup_name [, diskgroup_name ]...
| ALL
} { undrop_disk_clause
| diskgroup_availability
| enable_disable_volume
}
} ;

```

ALTER FLASHBACK ARCHIVE

```

ALTER FLASHBACK ARCHIVE flashback_archive
{ SET DEFAULT
| { ADD | MODIFY } TABLESPACE tablespace [flashback_archive_quota]
| REMOVE TABLESPACE tablespace_name
| MODIFY RETENTION flashback_archive_retention
| PURGE { ALL | BEFORE { SCN expr | TIMESTAMP expr } }
| [NO] OPTIMIZE DATA
};

```

ALTER FUNCTION

```

ALTER FUNCTION [ schema. ] function_name
{ function_compile_clause | { EDITIONABLE | NONEDITIONABLE } }

```

ALTER HIERARCHY

```

ALTER HIERARCHY [ schema. ] hierarchy_name
{ RENAME TO new_hier_name | COMPILE };

```

ALTER INDEX

```

ALTER INDEX [ schema. ] index
{ { deallocate_unused_clause
| allocate_extent_clause
| shrink_clause
| parallel_clause
| physical_attributes_clause
| logging_clause
| partial_index_clause
} ...
| rebuild_clause [ { DEFERRED | IMMEDIATE } INVALIDATION ]
| PARAMETERS ( 'ODCI_parameters' )
| COMPILE
| { ENABLE | DISABLE }
| UNUSABLE [ ONLINE ] [ { DEFERRED | IMMEDIATE } INVALIDATION ]
| VISIBLE | INVISIBLE
| RENAME TO new_name
| COALESCE [ CLEANUP ] [ ONLY ] [ parallel_clause ]
| { MONITORING | NOMONITORING } USAGE

```



```
| UPDATE BLOCK REFERENCES
| alter_index_partitioning
}
;
```

[ALTER INDEXTYPE](#)

```
ALTER INDEXTYPE [ schema. ] indextype
{ { ADD | DROP } [ schema. ] operator ( parameter_types )
  [ , { ADD | DROP } [ schema. ] operator ( parameter_types ) ]...
[ using_type_clause ]
| COMPILE
}
[ WITH LOCAL [ RANGE ] PARTITION ] [ storage_table_clause ]
;
```

[ALTER INMEMORY JOIN GROUP](#)

```
ALTER INMEMORY JOIN GROUP [ schema. ] join_group
{ ADD | REMOVE } ( [ schema. ] table ( column ) );
```

[ALTER JAVA](#)

```
ALTER JAVA
{ SOURCE | CLASS } [ schema. ] object_name
[ RESOLVER
  ( ( match_string [, ] { schema_name | - } )... )
]
{ { COMPILE | RESOLVE }
| invoker_rights_clause
} ;
```

[ALTER LIBRARY](#)

```
ALTER LIBRARY [ schema. ] library_name
{ library_compile_clause | { EDITIONABLE | NONEDITIONABLE } }
```

[ALTER LOCKDOWN PROFILE](#)

```
ALTER LOCKDOWN PROFILE
{ lockdown_features
| lockdown_options
| lockdown_statements
} ;
```

[ALTER MATERIALIZED VIEW](#)

```
ALTER MATERIALIZED VIEW
[ schema. ] materialized_view
[ physical_attributes_clause
| modify_mv_column_clause
| table_compression
| inmemory_table_clause
| LOB_storage_clause [, LOB_storage_clause ]...
| modify_LOB_storage_clause [, modify_LOB_storage_clause ]...
| alter_table_partitioning
| parallel_clause
| logging_clause
| allocate_extent_clause
| deallocate_unused_clause
| shrink_clause
| { CACHE | NOCACHE }
]
[ alter_iot_clauses ]
[ USING INDEX physical_attributes_clause ]
```

```

[ MODIFY scoped_table_ref_constraint
| alter_mv_refresh
]
[ evaluation_edition_clause ]
[ { ENABLE | DISABLE } ON QUERY COMPUTATION ]
[ alter_query_rewrite_clause
| COMPILE
| CONSIDER FRESH
] ;

```

[ALTER MATERIALIZED VIEW LOG](#)

```

ALTER MATERIALIZED VIEW LOG [ FORCE ]
ON [ schema. ]table
[ physical_attributes_clause
| add_mv_log_column_clause
| alter_table_partitioning
| parallel_clause
| logging_clause
| allocate_extent_clause
| shrink_clause
| move_mv_log_clause
| { CACHE | NOCACHE }
] [ mv_log_augmentation ] [ mv_log_purge_clause ] [ for_refresh_clause ]
;

```

[ALTER MATERIALIZED ZONEMAP](#)

```

ALTER MATERIALIZED ZONEMAP [ schema. ] zonemap_name
{ alter_zonemap_attributes
| zonemap_refresh_clause
| { ENABLE | DISABLE } PRUNING
| COMPILE
| REBUILD
| UNUSABLE
} ;

```

[ALTER OPERATOR](#)

```

ALTER OPERATOR [ schema. ] operator
{ add_binding_clause
| drop_binding_clause
| COMPILE
} ;

```

[ALTER OUTLINE](#)

```

ALTER OUTLINE [ PUBLIC | PRIVATE ] outline
{ REBUILD
| RENAME TO new_outline_name
| CHANGE CATEGORY TO new_category_name
| { ENABLE | DISABLE }
} ...
;

```

[ALTER PACKAGE](#)

```

ALTER PACKAGE [ schema. ] package_name
{ package_compile_clause | { EDITIONABLE | NONEDITIONABLE } }

```

[ALTER PLUGGABLE DATABASE](#)

```

ALTER PLUGGABLE DATABASE
{ pdb_unplug_clause
| pdb_settings_clauses

```

```

| pdb_datafile_clause
| pdb_recovery_clauses
| pdb_change_state
| pdb_change_state_from_root
| application_clauses
| snapshot_clauses
| prepare_clause
| drop_mirror_copy
| lost_write_protection
} ;

```

ALTER PROCEDURE

```

ALTER PROCEDURE [ schema. ] procedure_name
{ procedure_compile_clause | { EDITIONABLE | NONEDITIONABLE } }

```

ALTER PROFILE

```

ALTER PROFILE profile LIMIT
{ resource_parameters | password_parameters } ...
[ CONTAINER = { CURRENT | ALL } ] ;

```

ALTER RESOURCE COST

```

ALTER RESOURCE COST
{ { CPU_PER_SESSION
  | CONNECT_TIME
  | LOGICAL_READS_PER_SESSION
  | PRIVATE_SGA
  } integer
} ...
;

```

ALTER ROLE

```

ALTER ROLE role
{ NOT IDENTIFIED
| IDENTIFIED
  { BY password
  | USING [ schema. ] package
  | EXTERNALLY
  | GLOBALLY AS ' { domain_name_of_directory_group
                  | AZURE_ROLE=value
                  | IAM_GROUP_NAME=value } '
  }
}
[ CONTAINER = { CURRENT | ALL } ] ;

```

ALTER ROLLBACK SEGMENT

```

ALTER ROLLBACK SEGMENT rollback_segment
{ ONLINE
| OFFLINE
| storage_clause
| SHRINK [ TO size_clause ]
};

```

ALTER SEQUENCE

```

ALTER SEQUENCE [ schema. ] sequence
{
  { INCREMENT BY | START WITH } integer
| { MAXVALUE integer | NOMAXVALUE }
| { MINVALUE integer | NOMINVALUE }
| RESTART
}

```

```

| { CYCLE | NOCYCLE }
| { CACHE integer | NOCACHE }
| { ORDER | NOORDER }
| { KEEP | NOKEEP }
| { SCALE {EXTEND | NOEXTEND} | NOSCALE }
| { SHARD {EXTEND | NOEXTEND} | NOSHARD }
| { SESSION | GLOBAL }
} ...
;

```

ALTER SESSION

```

ALTER SESSION
{ ADVISE { COMMIT | ROLLBACK | NOTHING }
| CLOSE DATABASE LINK dblink
| { ENABLE | DISABLE } COMMIT IN PROCEDURE
| { ENABLE | DISABLE } GUARD
| { ENABLE | DISABLE | FORCE } PARALLEL
| { DML | DDL | QUERY } [ PARALLEL integer ]
| { ENABLE RESUMABLE [ TIMEOUT integer ] [ NAME string ]
| DISABLE RESUMABLE
}
| { ENABLE | DISABLE } SHARD DDL
| SYNC WITH PRIMARY
| alter_session_set_clause
} ;

```

ALTER SYNONYM

```

ALTER [ PUBLIC ] SYNONYM [ schema. ] synonym
{ EDITIONABLE | NONEDITIONABLE | COMPILE } ;

```

ALTER SYSTEM

```

ALTER SYSTEM
{ archive_log_clause
| checkpoint_clause
| check_datafiles_clause
| distributed_recov_clauses
| FLUSH { SHARED_POOL | GLOBAL CONTEXT | BUFFER_CACHE | FLASH_CACHE
| REDO TO target_db_name [ [ NO ] CONFIRM APPLY ] }
| end_session_clauses
| SWITCH LOGFILE
| { SUSPEND | RESUME }
| quiesce_clauses
| rolling_migration_clauses
| rolling_patch_clauses
| security_clauses
| affinity_clauses
| shutdown_dispatcher_clause
| REGISTER
| SET alter_system_set_clause
| [ alter_system_set_clause ]...
| RESET alter_system_reset_clause
| [ alter_system_reset_clause ]...
| RELOCATE CLIENT client_id
| cancel_sql_clause
| FLUSH PASSWORDFILE_METADATA_CACHE
} ;

```

ALTER TABLE

```

ALTER TABLE [ schema. ] table
[ memoptimize_read_clause ] [ memoptimize_write_clause ]
[ alter_table_properties

```

```

| column_clauses
| constraint_clauses
| alter_table_partitioning [ { DEFERRED | IMMEDIATE } INVALIDATION ]
| alter_external_table
| move_table_clause
| modify_to_partitioned
| modify_opaque_type
| immutable_table_clauses
| blockchain_table_clauses
]
[ enable_disable_clause
| { ENABLE | DISABLE }
| { TABLE LOCK | ALL TRIGGERS | CONTAINER_MAP | CONTAINERS_DEFAULT }
] ...
;

```

[ALTER TABLESPACE](#)

```
ALTER TABLESPACE tablespace alter_tablespace_attrs ;
```

[ALTER TABLESPACE SET](#)

```
ALTER TABLESPACE SET tablespace_set alter_tablespace_attrs ;
```

[ALTER TRIGGER](#)

```

ALTER TRIGGER [ schema. ] trigger_name
{ trigger_compile_clause
| { ENABLE | DISABLE }
| RENAME TO new_name
| { EDITIONABLE | NONEDITIONABLE }
} ;

```

[ALTER TYPE](#)

```

ALTER TYPE [ schema. ] type_name
{ alter_type_clause | { EDITIONABLE | NONEDITIONABLE } }

```

[ALTER USER](#)

```

ALTER USER
{ user
  { IDENTIFIED
    { BY password [ REPLACE old_password ]
    | EXTERNALLY [ AS 'certificate_DN' | AS 'kerberos_principal_name' ]
    | GLOBALLY [ AS ' { directory_DN | { {AZURE_USER | AZURE_ROLE }=value}
                | { { IAM_GROUP_NAME | IAM_PRINCIPAL_NAME }=value}
              }'
    ]
  }
  | NO AUTHENTICATION
  | DEFAULT COLLATION collation_name
  | DEFAULT TABLESPACE tablespace
  | [ LOCAL ] TEMPORARY TABLESPACE { tablespace | tablespace_group_name }
  | { QUOTA { size_clause
            | UNLIMITED
          } ON tablespace
  } ...
  | PROFILE profile
  | DEFAULT ROLE { role [, role ]...
                | ALL [ EXCEPT role [, role ]... ]
                | NONE
  }
  | PASSWORD EXPIRE
  | EXPIRE PASSWORD ROLLOVER PERIOD

```

```

| ACCOUNT { LOCK | UNLOCK }
| ENABLE EDITIONS [ FOR object_type [, object_type ]... ] [ FORCE ]
| [HTTP] DIGEST { ENABLE | DISABLE }
| CONTAINER = { CURRENT | ALL }
| container_data_clause
} ...
| user [, user ]... proxy_clause
};

```

[ALTER VIEW](#)

```

ALTER VIEW [ schema. ] view
{ ADD out_of_line_constraint
| MODIFY CONSTRAINT constraint
  { RELY | NORELY }
| DROP { CONSTRAINT constraint
        | PRIMARY KEY
        | UNIQUE (column [, column ]...)
      }
| COMPILE
| { READ ONLY | READ WRITE }
| { EDITIONABLE | NONEDITIONABLE }
};

```

[ANALYZE](#)

```

ANALYZE
{ { TABLE [ schema. ] table
  | INDEX [ schema. ] index
  } [ partition_extension_clause ]
| CLUSTER [ schema. ] cluster
}
{ validation_clauses
| LIST CHAINED ROWS [ into_clause ]
| DELETE [ SYSTEM ] STATISTICS
};

```

[ASSOCIATE STATISTICS](#)

```

ASSOCIATE STATISTICS WITH
{ column_association | function_association }
[ storage_table_clause ];

```

[AUDIT \(従来型監査\)](#)

```

AUDIT
{ audit_operation_clause [ auditing_by_clause | IN SESSION CURRENT ]
| audit_schema_object_clause
| NETWORK
| DIRECT_PATH LOAD [ auditing_by_clause ]
} [ BY { SESSION | ACCESS } ]
[ WHENEVER [ NOT ] SUCCESSFUL ]
[ CONTAINER = { CURRENT | ALL } ]
;

```

[AUDIT \(統合監査\)](#)

```

AUDIT
{ POLICY policy
  [ { BY user [, user]... }
  | { EXCEPT user [, user]... }
  | by_users_with_roles ]
[ WHENEVER [ NOT ] SUCCESSFUL ]
}
|

```

```
{ CONTEXT NAMESPACE namespace ATTRIBUTES attribute [, attribute ]...
  [, CONTEXT NAMESPACE namespace ATTRIBUTES attribute [, attribute ]... ]...
  [ BY user [, user]... ]
};
```

[CALL](#)

```
CALL
{ routine_clause
| object_access_expression
}
[ INTO :host_variable
  [ [ INDICATOR ] :indicator_variable ] ] ;
```

[COMMENT](#)

```
COMMENT ON
{ AUDIT POLICY policy
| COLUMN [ schema. ]
  { table. | view. | materialized_view. } column
| EDITION edition_name
| INDEXTYPE [ schema. ] indextype
| MATERIALIZED VIEW materialized_view
| MINING MODEL [ schema. ] model
| OPERATOR [ schema. ] operator
| TABLE [ schema. ] { table | view }
}
IS string ;
```

[COMMIT](#)

```
COMMIT [ WORK ]
[ [ COMMENT string ]
  | [ WRITE [ WAIT | NOWAIT ] [ IMMEDIATE | BATCH ]
  ]
| FORCE string [, integer ]
] ;
```

[CREATE ANALYTIC VIEW](#)

```
CREATE [ OR REPLACE ] [ { FORCE | NOFORCE } ]
ANALYTIC VIEW [ schema. ] analytic_view
[ sharing clause ]
[ classification_clause ]...
using_clause
dim_by_clause
measures_clause
[ default_measure_clause ]
[ default_aggregate_clause ]
[ cache_clause ]
;
```

[CREATE ATTRIBUTE DIMENSION](#)

```
CREATE [ OR REPLACE ] [ FORCE | NOFORCE ] ATTRIBUTE DIMENSION
[ schema. ] attr_dimension [ sharing_clause ] [ classification_clause ]... ]
[ DIMENSION TYPE { STANDARD | TIME } ]
attr_dim_using_clause
attributes_clause
[ attr_dim_level_clause ]...
[ all_clause ]
;
```

[CREATE AUDIT POLICY](#)

```

CREATE AUDIT POLICY policy
  [ privilege_audit_clause ] [ action_audit_clause ] [ role_audit_clause ]
  [ WHEN 'audit_condition' EVALUATE PER { STATEMENT | SESSION | INSTANCE } ]
  [ ONLY TOPLEVEL ]
  [ CONTAINER = { ALL | CURRENT } ] ;

```

[CREATE CLUSTER](#)

```

CREATE CLUSTER [ schema. ] cluster
  (column datatype [ COLLATE column_collation_name ] [ SORT ]
    [, column datatype [ COLLATE column_collation_name ] [ SORT ] ]...
  )
  [ { physical_attributes_clause
    | SIZE size_clause
    | TABLESPACE tablespace
    | { INDEX
      | [ SINGLE TABLE ]
      | HASHKEYS integer [ HASH IS expr ]
    }
    }...
  ]
  [ parallel_clause ]
  [ NOROWDEPENDENCIES | ROWDEPENDENCIES ]
  [ CACHE | NOCACHE ] [ cluster_range_partitions ] ;

```

[CREATE CONTEXT](#)

```

CREATE [ OR REPLACE ] CONTEXT namespace
  USING [ schema. ] package
  [ INITIALIZED { EXTERNALLY | GLOBALLY }
  | ACCESSED GLOBALLY
  ] ;

```

[CREATE CONTROLFILE](#)

```

CREATE CONTROLFILE
  [ REUSE ] [ SET ] DATABASE database
  [ logfile_clause ]
  { RESETLOGS | NORESETLOGS }
  [ DATAFILE file_specification
    [, file_specification ]... ]
  [ MAXLOGFILES integer
  | MAXLOGMEMBERS integer
  | MAXLOGHISTORY integer
  | MAXDATAFILES integer
  | MAXINSTANCES integer
  | { ARCHIVELOG | NOARCHIVELOG }
  | FORCE LOGGING
  | SET STANDBY LOGGING FOR {DATA AVAILABILITY | LOAD PERFORMANCE}
  ]...
  [ character_set_clause ] ;

```

[CREATE DATABASE](#)

```

CREATE DATABASE [ database ]
  { USER SYS IDENTIFIED BY password
  | USER SYSTEM IDENTIFIED BY password
  | CONTROLFILE REUSE
  | MAXDATAFILES integer
  | MAXINSTANCES integer
  | CHARACTER SET charset
  | NATIONAL CHARACTER SET charset
  | SET DEFAULT
    { BIGFILE | SMALLFILE } TABLESPACE
  | database_logging_clauses

```



```

| tablespace_clauses
| set_time_zone_clause
| [ BIGFILE | SMALLFILE ] USER_DATA TABLESPACE tablespace_name
  DATAFILE datafile_tempfile_spec [, datafile_tempfile_spec ]...
| enable_pluggable_database
}... ;

```

[CREATE DATABASE LINK](#)

```

CREATE [ SHARED ] [ PUBLIC ] DATABASE LINK dblink
  [ CONNECT TO
    { CURRENT_USER
      | user IDENTIFIED BY password [ dblink_authentication ]
    }
  | dblink_authentication
  ]...
[ USING connect_string ] ;

```

[CREATE DIMENSION](#)

```

CREATE DIMENSION [ schema. ] dimension
  level_clause ...
  { hierarchy_clause
  | attribute_clause
  | extended_attribute_clause
  }...
;

```

[CREATE DIRECTORY](#)

```

CREATE [ OR REPLACE ] DIRECTORY directory
  [ SHARING = { METADATA | NONE } ]
  AS 'path_name' ;

```

[CREATE DISKGROUP](#)

```

CREATE DISKGROUP diskgroup_name
  [ { HIGH | NORMAL | FLEX | EXTENDED [ SITE site_name ] | EXTERNAL } REDUNDANCY ]
  { [ QUORUM | REGULAR ] [ FAILGROUP failgroup_name ]
    DISK qualified_disk_clause [, qualified_disk_clause ]...
  }...
  [ ATTRIBUTE { 'attribute_name' = 'attribute_value' }
    [, 'attribute_name' = 'attribute_value' ]... ]
;

```

[CREATE EDITION](#)

```

CREATE EDITION edition
  [ AS CHILD OF parent_edition ]
;

```

[CREATE FLASHBACK ARCHIVE](#)

```

CREATE FLASHBACK ARCHIVE [DEFAULT] flashback_archive
  TABLESPACE tablespace
  [flashback_archive_quota]
  [ [NO] OPTIMIZE DATA ]
  flashback_archive_retention
;

```

[CREATE FUNCTION](#)

```

CREATE [ OR REPLACE ]
[ EDITIONABLE | NONEDITIONABLE ]
FUNCTION plsql_function_source

```

[CREATE HIERARCHY](#)

```
CREATE [ OR REPLACE ] [ FORCE | NOFORCE ]
  HIERARCHY [ schema. ] hierarchy
  [ sharing_clause ]
  [ classification_clause ]... ]
  hier_using_clause
  level_hier_clause
  [ hier_attrs_clause ]
;
```

[CREATE INDEX](#)

```
CREATE [ UNIQUE | BITMAP ] INDEX [ schema. ] index
  ON { cluster_index_clause
      | table_index_clause
      | bitmap_join_index_clause
      }
  [ USABLE | UNUSABLE ]
  [ { DEFERRED | IMMEDIATE } INVALIDATION ] ;
```

[CREATE INDEXTYPE](#)

```
CREATE [ OR REPLACE ] INDEXTYPE [ schema. ] indextype
  FOR [ schema. ] operator (parameter_type [, parameter_type ]...)
      [, [ schema. ] operator (parameter_type [, parameter_type ]...)
      ]...
  using_type_clause
  [WITH LOCAL [RANGE] PARTITION ]
  [ storage_table_clause ]
;
```

[CREATE INMEMORY JOIN GROUP](#)

```
CREATE INMEMORY JOIN GROUP [ schema. ] join_group
  ( [ schema. ] table ( column ) , [ schema. ] table ( column )
    [, [ schema. ] table ( column ) ]... ) ;
```

[CREATE JAVA](#)

```
CREATE [ OR REPLACE ] [ AND { RESOLVE | COMPILE } ] [ NOFORCE ]
  JAVA { { SOURCE | RESOURCE } NAMED [ schema. ] primary_name
        | CLASS [ SCHEMA schema ]
        }
  [ SHARING = { METADATA | NONE } ]
  [ invoker_rights_clause ]
  [ RESOLVER ( (match_string [,] { schema_name | - } )... ) ]
  { USING { BFILE (directory_object_name, server_file_name)
          | { CLOB | BLOB | BFILE } subquery
          | 'key_for_BLOB'
          }
  | AS source_char
  } ;
```

[CREATE LIBRARY](#)

```
CREATE [ OR REPLACE ]
  [ EDITIONABLE | NONEDITIONABLE ]
  LIBRARY plsql_library_source
```

[CREATE LOCKDOWN PROFILE](#)

```
CREATE LOCKDOWN PROFILE profile_name ;
```

[CREATE MATERIALIZED VIEW](#)

```
CREATE MATERIALIZED VIEW [ schema. ] materialized_view
  [ OF [ schema. ] object_type ]
  [ ( { scoped_table_ref_constraint
      | column_alias [ENCRYPT [encryption_spec]]
      }
    [, { scoped_table_ref_constraint
      | column_alias [ENCRYPT [encryption_spec]]
      }
    ]...
  )
]
[ DEFAULT COLLATION collation_name ]
{ ON PREBUILT TABLE
  [ { WITH | WITHOUT } REDUCED PRECISION ]
  | physical_properties materialized_view_props
}
[ USING INDEX
  [ physical_attributes_clause
  | TABLESPACE tablespace
  ]...
| USING NO INDEX
]
[ create_mv_refresh ]
[ evaluation_edition_clause ]
[ { ENABLE | DISABLE } ON QUERY COMPUTATION ]
[ query_rewrite_clause ]
AS subquery ;
```

[CREATE MATERIALIZED VIEW LOG](#)

```
CREATE MATERIALIZED VIEW LOG ON [ schema. ] table
  [ physical_attributes_clause
  | TABLESPACE tablespace
  | logging_clause
  | { CACHE | NOCACHE }
  ]...
[ parallel_clause ]
[ table_partitioning_clauses ]
[ WITH [ { OBJECT ID
      | PRIMARY KEY
      | ROWID
      | SEQUENCE
      | COMMIT SCN
      }
      [ { , OBJECT ID
      | , PRIMARY KEY
      | , ROWID
      | , SEQUENCE
      | , COMMIT SCN
      }
      ]... ]
  (column [, column ]...)
  [ new_values_clause ]
] [ mv_log_purge_clause ] [ for_refresh_clause ]
;
```

[CREATE MATERIALIZED ZONEMAP](#)

```
{ create_zonemap_on_table | create_zonemap_as_subquery } ;
```

[CREATE OPERATOR](#)

```
CREATE [ OR REPLACE ] OPERATOR
  [ schema. ] operator binding_clause ;
```

CREATE OUTLINE

```
CREATE [ OR REPLACE ]
  [ PUBLIC | PRIVATE ] OUTLINE [ outline ]
  [ FROM [ PUBLIC | PRIVATE ] source_outline ]
  [ FOR CATEGORY category ]
  [ ON statement ] ;
```

CREATE PACKAGE

```
CREATE [ OR REPLACE ]
  [ EDITIONABLE | NONEDITIONABLE ]
PACKAGE plsql_package_source
```

CREATE PACKAGE BODY

```
CREATE [ OR REPLACE ]
  [ EDITIONABLE | NONEDITIONABLE ]
PACKAGE BODY plsql_package_body_source
```

CREATE PFILE

```
CREATE PFILE [= 'pfile_name' ]
  FROM { SPFILE [= 'spfile_name' ]
        | MEMORY
        } ;
```

CREATE PLUGGABLE DATABASE

```
CREATE PLUGGABLE DATABASE
  { { pdb_name [ AS APPLICATION CONTAINER ] } | { AS SEED } }
  { create_pdb_from_seed | create_pdb_clone | create_pdb_from_xml |
  create_pdb_from_mirror_copy
  | using_snapshot_clause | container_map_clause } pdb_snapshot_clause;
```

CREATE PROCEDURE

```
CREATE [ OR REPLACE ]
  [ EDITIONABLE | NONEDITIONABLE ]
PROCEDURE plsql_procedure_source
```

CREATE PROFILE

```
CREATE PROFILE profile
  LIMIT { resource_parameters
        | password_parameters
        }...
  [ CONTAINER = { CURRENT | ALL } ] ;
```

CREATE RESTORE POINT

```
CREATE [ CLEAN ] RESTORE POINT restore_point
  [ FOR PLUGGABLE DATABASE pdb_name ]
  [ AS OF {TIMESTAMP | SCN} expr ]
  [ PRESERVE
  | GUARANTEE FLASHBACK DATABASE
  ];
```

CREATE ROLE

```
CREATE ROLE role
  [ NOT IDENTIFIED
  | IDENTIFIED { BY password
                | USING [ schema. ] package
```

```

        | EXTERNALLY
        | GLOBALLY AS ' { domain_name_of_directory_group
                        | AZURE_ROLE=value
                        | IAM_GROUP_NAME=value } '
    }
] [ CONTAINER = { CURRENT | ALL } ];

```

[CREATE ROLLBACK SEGMENT](#)

```

CREATE [ PUBLIC ] ROLLBACK SEGMENT rollback_segment
[ TABLESPACE tablespace | storage_clause ]...;

```

[CREATE SCHEMA](#)

```

CREATE SCHEMA AUTHORIZATION schema
{ create_table_statement
| create_view_statement
| grant_statement
}...
;

```

[CREATE SEQUENCE](#)

```

CREATE SEQUENCE [ schema. ] sequence
[ SHARING = { METADATA | DATA | NONE } ]
[ { INCREMENT BY | START WITH } integer
| { MAXVALUE integer | NOMAXVALUE }
| { MINVALUE integer | NOMINVALUE }
| { CYCLE | NOCYCLE }
| { CACHE integer | NOCACHE }
| { ORDER | NOORDER }
| { KEEP | NOKEEP }
| { SCALE {EXTEND | NOEXTEND} | NOSCALE }
| { SHARD {EXTEND | NOEXTEND} | NOSHARD }
| { SESSION | GLOBAL }
]...
;

```

[CREATE SPFILE](#)

```

CREATE SPFILE [= 'spfile_name' ]
FROM { PFILE [= 'pfile_name' ] [ AS COPY ]
| MEMORY
} ;

```

[CREATE SYNONYM](#)

```

CREATE [ OR REPLACE ] [ EDITIONABLE | NONEDITIONABLE ]
[ PUBLIC ] SYNONYM
[ schema. ] synonym
[ SHARING = { METADATA | NONE } ]
FOR [ schema. ] object [ @ dblink ] ;

```

[CREATE TABLE](#)

```

CREATE [ { GLOBAL | PRIVATE } TEMPORARY | SHARDED | DUPLICATED |
[ IMMUTABLE ] BLOCKCHAIN | IMMUTABLE ] TABLE
[ schema. ] table
[ SHARING = { METADATA | DATA | EXTENDED DATA | NONE } ]
{ relational_table | object_table | XMLType_table }
[ MEMOPTIMIZE FOR READ ]
[ MEMOPTIMIZE FOR WRITE ]
[ PARENT [ schema. ] table ] [ MEMOPTIMIZE FOR READ ];

```

[CREATE TABLESPACE](#)

```
CREATE
  [ BIGFILE | SMALLFILE ]
  { permanent_tablespace_clause
  | temporary_tablespace_clause
  | undo_tablespace_clause
  } ;
```

[CREATE TABLESPACE SET](#)

```
CREATE TABLESPACE SET tablespace_set
  [ IN SHARDSPACE shardspace ]
  [ USING TEMPLATE
    ( { DATAFILE [, file_specification ]... } permanent_tablespace_attrs )
  ] ;
```

[CREATE TRIGGER](#)

```
CREATE [ OR REPLACE ]
[ EDITIONABLE | NONEDITIONABLE ]
TRIGGER plsql_trigger_source
```

[CREATE TYPE](#)

```
CREATE [OR REPLACE]
[ EDITIONABLE | NONEDITIONABLE ]
TYPE plsql_type_source
```

[CREATE TYPE BODY](#)

```
CREATE [ OR REPLACE ]
[ EDITIONABLE | NONEDITIONABLE ]
TYPE BODY plsql_type_body_source
```

[CREATE USER](#)

```
CREATE USER user
  IDENTIFIED
    { BY password [ [HTTP] DIGEST { ENABLE | DISABLE } ]
    | EXTERNALLY [ AS 'certificate_DN' | AS 'kerberos_principal_name' ]
    | GLOBALLY [ AS ' { directory_DN | { {AZURE_USER | AZURE_ROLE }=value}
      | { { IAM_GROUP_NAME | IAM_PRINCIPAL_NAME }=value}
      }'
    ]
  }
  [ NO AUTHENTICATION
  [ DEFAULT COLLATION collation_name
  | DEFAULT TABLESPACE tablespace
  | [ LOCAL ] TEMPORARY TABLESPACE { tablespace | tablespace_group_name }
  | { QUOTA { size_clause | UNLIMITED } ON tablespace }...
  | PROFILE profile
  | PASSWORD EXPIRE
  | ACCOUNT { LOCK | UNLOCK }
  [ DEFAULT TABLESPACE tablespace
  | TEMPORARY TABLESPACE
    { tablespace | tablespace_group_name }
  | { QUOTA { size_clause | UNLIMITED } ON tablespace }...
  | PROFILE profile
  | PASSWORD EXPIRE
  | ACCOUNT { LOCK | UNLOCK }
  | ENABLE EDITIONS
  | CONTAINER = { CURRENT | ALL }
  ]...
  ] ;
```

[CREATE VIEW](#)

```
CREATE [OR REPLACE]
  [[NO] FORCE]
  [ EDITIONING | EDITIONABLE [ EDITIONING ] | NONEDITIONABLE ]
VIEW [schema.] view
  [ SHARING = { METADATA | DATA | EXTENDED DATA | NONE } ]
  [ ( { alias [ VISIBLE | INVISIBLE ] [ inline_constraint... ]
      | out_of_line_constraint
    }
    [, { alias [ VISIBLE | INVISIBLE ] [ inline_constraint... ]
      | out_of_line_constraint
    }
  ]
  )
  | object_view_clause
  | XMLType_view_clause
]
[ DEFAULT COLLATION collation_name ]
[ BEQUEATH { CURRENT_USER | DEFINER } ]
AS subquery [ subquery_restriction_clause ]
[ CONTAINER_MAP | CONTAINERS_DEFAULT ] ;
```

[DELETE](#)

```
DELETE [ hint ]
  [ FROM ]
  { dml_table_expression_clause
  | ONLY (dml_table_expression_clause)
  } [ t_alias ]
  [ where_clause ]
  [ returning_clause ]
  [error_logging_clause];
```

[DISASSOCIATE STATISTICS](#)

```
DISASSOCIATE STATISTICS FROM
  { COLUMNS [ schema. ]table.column
    [, [ schema. ]table.column ]...
  | FUNCTIONS [ schema. ]function
    [, [ schema. ]function ]...
  | PACKAGES [ schema. ]package
    [, [ schema. ]package ]...
  | TYPES [ schema. ]type
    [, [ schema. ]type ]...
  | INDEXES [ schema. ]index
    [, [ schema. ]index ]...
  | INDEXTYPES [ schema. ]indextype
    [, [ schema. ]indextype ]...
  }
  [ FORCE ] ;
```

[DROP ANALYTIC VIEW](#)

```
DROP ANALYTIC VIEW [ schema. ] analytic_view_name;
```

[DROP ATTRIBUTE DIMENSION](#)

```
DROP ATTRIBUTE DIMENSION [ schema. ] attr_dimension_name;
```

[DROP AUDIT POLICY](#)

```
DROP AUDIT POLICY policy ;
```

[DROP CLUSTER](#)

```
DROP CLUSTER [ schema. ] cluster
  [ INCLUDING TABLES [ CASCADE CONSTRAINTS ] ] ;
```

[DROP CONTEXT](#)

```
DROP CONTEXT namespace ;
```

[DROP DATABASE](#)

```
DROP DATABASE ;
```

[DROP DATABASE LINK](#)

```
DROP [ PUBLIC ] DATABASE LINK dblink ;
```

[DROP DIMENSION](#)

```
DROP DIMENSION [ schema. ] dimension ;
```

[DROP DIRECTORY](#)

```
DROP DIRECTORY directory_name ;
```

[DROP DISKGROUP](#)

```
DROP DISKGROUP diskgroup_name
  [ FORCE INCLUDING CONTENTS
  | { INCLUDING | EXCLUDING } CONTENTS
  ];
```

[DROP EDITION](#)

```
DROP EDITION edition [CASCADE];
```

[DROP FLASHBACK ARCHIVE](#)

```
DROP FLASHBACK ARCHIVE flashback_archive;
```

[DROP FUNCTION](#)

```
DROP FUNCTION [ schema. ] function_name ;
```

[DROP HIERARCHY](#)

```
DROP HIERARCHY [ schema. ] hierarchy_name;
```

[DROP INDEX](#)

```
DROP INDEX [ schema. ] index [ ONLINE ] [ FORCE ] [ { DEFERRED | IMMEDIATE }
INVALIDATION ] ;
```

[DROP INDEXTYPE](#)

```
DROP INDEXTYPE [ schema. ] indextype [ FORCE ] ;
```

[DROP INMEMORY JOIN GROUP](#)

```
DROP INMEMORY JOIN GROUP [ schema. ] join_group ;
```

[DROP JAVA](#)


```
DROP JAVA { SOURCE | CLASS | RESOURCE }  
  [ schema. ] object_name ;
```

[DROP LIBRARY](#)

```
DROP LIBRARY library_name ;
```

[DROP LOCKDOWN PROFILE](#)

```
DROP LOCKDOWN PROFILE profile_name ;
```

[DROP MATERIALIZED VIEW](#)

```
DROP MATERIALIZED VIEW [ schema. ] materialized_view  
  [ PRESERVE TABLE ] ;
```

[DROP MATERIALIZED VIEW LOG](#)

```
DROP MATERIALIZED VIEW LOG ON [ schema. ] table ;
```

[DROP MATERIALIZED ZONEMAP](#)

```
DROP MATERIALIZED ZONEMAP [ schema. ] zonemap_name ;
```

[DROP OPERATOR](#)

```
DROP OPERATOR [ schema. ] operator [ FORCE ] ;
```

[DROP OUTLINE](#)

```
DROP OUTLINE outline ;
```

[DROP PACKAGE](#)

```
DROP PACKAGE [ BODY ] [ schema. ] package ;
```

[DROP PLUGGABLE DATABASE](#)

```
DROP PLUGGABLE DATABASE pdb_name  
  [ { KEEP | INCLUDING } DATAFILES ] ;
```

[DROP PROCEDURE](#)

```
DROP PROCEDURE [ schema. ] procedure ;
```

[DROP PROFILE](#)

```
DROP PROFILE profile [ CASCADE ] ;
```

[DROP RESTORE POINT](#)

```
DROP RESTORE POINT restore_point [ FOR PLUGGABLE DATABASE pdb_name ] ;
```

[DROP ROLE](#)

```
DROP ROLE role ;
```

[DROP ROLLBACK SEGMENT](#)

```
DROP ROLLBACK SEGMENT rollback_segment ;
```

[DROP SEQUENCE](#)

```
DROP SEQUENCE [ schema. ] sequence_name ;
```

[DROP SYNONYM](#)

```
DROP [PUBLIC] SYNONYM [ schema. ] synonym [FORCE] ;
```

[DROP TABLE](#)

```
DROP TABLE [ schema. ] table  
[ CASCADE CONSTRAINTS ] [ PURGE ] ;
```

[DROP TABLESPACE](#)

```
DROP TABLESPACE tablespace  
[ { DROP | KEEP } QUOTA ]  
[ INCLUDING CONTENTS [ { AND | KEEP } DATAFILES ] [ CASCADE CONSTRAINTS ] ]  
;
```

[DROP TABLESPACE SET](#)

```
DROP TABLESPACE SET tablespace_set  
[ { DROP | KEEP } QUOTA ]  
[ INCLUDING CONTENTS [ { AND | KEEP } DATAFILES ] [ CASCADE CONSTRAINTS ] ]  
;
```

[DROP TRIGGER](#)

```
DROP TRIGGER [ schema. ] trigger ;
```

[DROP TYPE](#)

```
DROP TYPE [ schema. ] type_name [ FORCE | VALIDATE ] ;
```

[DROP TYPE BODY](#)

```
DROP TYPE BODY [ schema. ] type_name ;
```

[DROP USER](#)

```
DROP USER user [ CASCADE ] ;
```

[DROP VIEW](#)

```
DROP VIEW [ schema. ] view [ CASCADE CONSTRAINTS ] ;
```

[EXPLAIN PLAN](#)

```
EXPLAIN PLAN  
[ SET STATEMENT_ID = string ]  
[ INTO [ schema. ] table [ @ dblink ] ]  
FOR statement ;
```

[FLASHBACK DATABASE](#)

```
FLASHBACK [ STANDBY ] [ PLUGGABLE ] DATABASE [ database ]  
{ TO { { SCN | TIMESTAMP } expr  
  | RESTORE POINT restore_point  
  }  
}  
| { TO BEFORE { { SCN | TIMESTAMP } expr  
  | RESETLOGS  
  }  
}  
};
```

FLASHBACK TABLE

```
FLASHBACK TABLE
  [ schema. ] table
  [, [ schema. ] table ]...
TO { { { SCN | TIMESTAMP } expr
    | RESTORE POINT restore_point
    } [ { ENABLE | DISABLE } TRIGGERS ]
    | BEFORE DROP [ RENAME TO table ]
  } ;
```

GRANT

```
GRANT
  { { { grant_system_privileges | grant_object_privileges }
    [ CONTAINER = { CURRENT | ALL } ] }
  | grant_roles_to_programs
  } ;
```

INSERT

```
INSERT [ hint ]
  { single_table_insert | multi_table_insert } ;
```

LOCK TABLE

```
LOCK TABLE [ schema. ] { table | view }
  [ partition_extension_clause
  | @ dblink
  ] [, [ schema. ] { table | view }
  [ partition_extension_clause
  | @ dblink
  ]
  ]...
IN lockmode MODE
[ NOWAIT
| WAIT integer
] ;
```

MERGE

```
MERGE [ hint ]
  INTO [ schema. ] { table | view } [ t_alias ]
  USING { [ schema. ] { table | view }
        | ( subquery )
        } [ t_alias ]
  ON ( condition )
  [ merge_update_clause ]
  [ merge_insert_clause ]
  [ error_logging_clause ] ;
```

NOAUDIT (従来型監査)

```
NOAUDIT
  { audit_operation_clause [ auditing_by_clause ]
  | audit_schema_object_clause
  | NETWORK
  | DIRECT_PATH LOAD [ auditing_by_clause ]
  }
  [ WHENEVER [ NOT ] SUCCESSFUL ]
  [ CONTAINER = { CURRENT | ALL } ] ;
```

NOAUDIT (統合監査)

```
NOAUDIT
```

```

{ POLICY policy [ { BY user [, user]... } | by_users_with_roles ]
  [ WHENEVER [ NOT ] SUCCESSFUL ] }
|
{ CONTEXT NAMESPACE namespace ATTRIBUTES attribute [, attribute ]...
  [, CONTEXT NAMESPACE namespace ATTRIBUTES attribute [, attribute ]... ]...
  [ BY user [, user]... ]
} ;

```

PURGE

```

PURGE
{ TABLE table
| INDEX index
| TABLESPACE tablespace [ USER username ]
| TABLESPACE SET tablespace_set [ USER username ]
| RECYCLEBIN
| DBA_RECYCLEBIN
} ;

```

RENAME

```

RENAME old_name TO new_name ;

```

REVOKE

```

REVOKE
{ { revoke_system_privileges | revoke_object_privileges }
  [ CONTAINER = { CURRENT | ALL } ] }
| revoke_roles_from_programs ;

```

ROLLBACK

```

ROLLBACK [ WORK ]
  [ TO [ SAVEPOINT ] savepoint
  | FORCE string
  ] ;

```

SAVEPOINT

```

SAVEPOINT savepoint ;

```

SELECT

```

subquery [ for_update_clause ] ;

```

SET CONSTRAINT[S]

```

SET { CONSTRAINT | CONSTRAINTS }
  { constraint [, constraint ]...
  | ALL
  }
  { IMMEDIATE | DEFERRED } ;

```

SET ROLE

```

SET ROLE
  { role [ IDENTIFIED BY password ]
  [, role [ IDENTIFIED BY password ] ]...
  | ALL [ EXCEPT role [, role ]... ]
  | NONE
  } ;

```

SET TRANSACTION

```
SET TRANSACTION
  { { READ { ONLY | WRITE }
    | ISOLATION LEVEL
    { SERIALIZABLE | READ COMMITTED }
    | USE ROLLBACK SEGMENT rollback_segment
    } [ NAME string ]
  | NAME string
  } ;
```

[TRUNCATE CLUSTER](#)

```
TRUNCATE CLUSTER [schema.] cluster
  [ {DROP | REUSE} STORAGE ] ;
```

[TRUNCATE TABLE](#)

```
TRUNCATE TABLE [schema.] table
  [ {PRESERVE | PURGE} MATERIALIZED VIEW LOG ]
  [ {DROP [ ALL ] | REUSE} STORAGE ] [ CASCADE ] ;
```

[UPDATE](#)

```
UPDATE [ hint ]
  { dml_table_expression_clause
  | ONLY (dml_table_expression_clause)
  } [ t_alias ]
update_set_clause
[ where_clause ]
[ returning_clause ]
[error_logging_clause] ;
```

2 SQLファンクション

この章では、SQLファンクションの構文について説明します。

この章の内容は次のとおりです。

- [SQLファンクションの構文](#)

SQLファンクションの構文

ファンクションはデータ項目を操作し、単一値を戻すコマンドです。

この項では、各SQLファンクションおよび関連する構文を示します。副次句の構文については、[「副次句」](#)を参照してください。

関連項目:

SQLファンクションの詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

[ABS](#)

```
ABS(n)
```

[ACOS](#)

```
ACOS(n)
```

[ADD_MONTHS](#)

```
ADD_MONTHS(date, integer)
```

[aggregate_function](#)

集計ファンクションは、単一行に基づく結果行を戻すのではなく、行のグループに基づく単一結果行を戻します。

[analytic_function](#)

```
analytic_function([ arguments ]) OVER (analytic_clause)
```

[ANY_VALUE](#)

```
ANY_VALUE ( [ DISTINCT | ALL ] expr )
```

[APPROX_COUNT](#)

```
APPROX_COUNT ( expr [ , expr 'MAX_ERROR' ]... )
```

[APPROX_COUNT_DISTINCT](#)

```
APPROX_COUNT_DISTINCT(expr)
```

[APPROX_COUNT_DISTINCT_AGG](#)

```
APPROX_COUNT_DISTINCT_AGG(detail)
```

[APPROX_COUNT_DISTINCT_DETAIL](#)

```
APPROX_COUNT_DISTINCT_DETAIL(expr)
```

[APPROX_MEDIAN](#)

```
APPROX_MEDIAN( expr [ DETERMINISTIC ] [, { 'ERROR_RATE' | 'CONFIDENCE' } ] )
```

[APPROX_PERCENTILE](#)

```
APPROX_PERCENTILE( expr [ DETERMINISTIC ] [, { 'ERROR_RATE' | 'CONFIDENCE' } ] )  
  WITHIN GROUP ( ORDER BY expr [ DESC | ASC ] )
```

[APPROX_PERCENTILE_AGG](#)

```
APPROX_PERCENTILE_AGG(expr)
```

[APPROX_PERCENTILE_DETAIL](#)

```
APPROX_PERCENTILE_DETAIL( expr [ DETERMINISTIC ] )
```

[APPROX_RANK](#)

```
APPROX_RANK ( expr [ PARTITION BY partition_by_clause ] [ ORDER BY order_by_clause  
DESC ] )
```

[APPROX_SUM](#)

```
APPROX_SUM ( expr [ , expr 'MAX_ERROR' ] ... )
```

[ASCII](#)

```
ASCII(char)
```

[ASCIISTR](#)

```
ASCIISTR(char)
```

[ASIN](#)

```
ASIN(n)
```

[ATAN](#)

```
ATAN(n)
```

[ATAN2](#)

```
ATAN2(n1 , n2)
```

[AVG](#)

```
AVG([ DISTINCT | ALL ] expr) [ OVER(analytic_clause) ]
```

[BFILENAME](#)

```
BFILENAME('directory', 'filename')
```

[BIN_TO_NUM](#)

```
BIN_TO_NUM(expr [, expr ]... )
```

[BITAND](#)

```
BITAND(expr1, expr2)
```

[BITMAP_BIT_POSITION](#)

```
BITMAP_BIT_POSITION ( expr )
```

[BITMAP_BUCKET_NUMBER](#)

```
BITMAP_BUCKET_NUMBER ( expr )
```

[BITMAP_CONSTRUCT_AGG](#)

```
BITMAP_CONSTRUCT_AGG ( expr )
```

[BITMAP_COUNT](#)

```
BITMAP_COUNT ( expr )
```

[BITMAP_OR_AGG](#)

```
BITMAP_OR_AGG ( expr )
```

[CARDINALITY](#)

```
CARDINALITY(nested_table)
```

[CAST](#)

```
CAST({ expr | MULTISSET (subquery) } AS type_name  
  [ DEFAULT return_value ON CONVERSION ERROR ]  
  [, fmt [, 'nlsparm' ] ] )
```

[CEIL](#)

```
CEIL(n)
```

[CHARTOROWID](#)

```
CHARTOROWID(char)
```

[CHR](#)

```
CHR(n [ USING NCHAR_CS ])
```

[CLUSTER_DETAILS\(集計\)](#)

```
CLUSTER_DETAILS ( [ schema . ] model  
  [ , cluster_id [ , topN ] ] [ DESC | ASC | ABS ]  
  mining_attribute_clause )
```

[CLUSTER_DETAILS\(分析\)](#)

```
CLUSTER_DETAILS ( INTO n  
  [ , cluster_id [ , topN ] ] [ DESC | ASC | ABS ]  
  mining_attribute_clause )  
  OVER ( mining_analytic_clause )
```

[CLUSTER_DISTANCE\(集計\)](#)

```
CLUSTER_DISTANCE ( [ schema . ] model [ , cluster_id ] mining_attribute_clause )
```

[CLUSTER_DISTANCE\(分析\)](#)

```
CLUSTER_DISTANCE ( INTO n [, cluster_id] mining_attribute_clause )  
  OVER ( mining_analytic_clause )
```


[CLUSTER_ID\(集計\)](#)

```
CLUSTER_ID ( [ schema . ] model mining_attribute_clause )
```

[CLUSTER_ID\(分析\)](#)

```
CLUSTER_ID ( INTO n mining_attribute_clause )  
OVER ( mining_analytic_clause )
```

[CLUSTER_PROBABILITY\(集計\)](#)

```
CLUSTER_PROBABILITY ( [ schema . ] model [, cluster_id ] mining_attribute_clause )
```

[CLUSTER_PROBABILITY\(分析\)](#)

```
CLUSTER_PROBABILITY ( INTO n [, cluster_id] mining_attribute_clause )  
OVER ( mining_analytic_clause )
```

[CLUSTER_SET\(集計\)](#)

```
CLUSTER_SET ( [ schema . ] model [ , topN [ , cutoff ] ] mining_attribute_clause )
```

[CLUSTER_SET\(分析\)](#)

```
CLUSTER_SET ( INTO n [, topN [, cutoff]] mining_attribute_clause )  
OVER ( mining_analytic_clause )
```

[COALESCE](#)

```
COALESCE(expr [, expr ]...)
```

[COLLATION](#)

```
COLLATION(expr)
```

[COLLECT](#)

```
COLLECT( [ DISTINCT | UNIQUE ] column [ ORDER BY expr ] )
```

[COMPOSE](#)

```
COMPOSE(char)
```

[CON_DBID_TO_ID](#)

```
CON_DBID_TO_ID(container_dbid)
```

[CON_GUID_TO_ID](#)

```
CON_GUID_TO_ID(container_guid)
```

[CON_ID_TO_CON_NAME](#)

```
CON_ID_TO_CON_NAME(container_guid)
```

[CON_ID_TO_DBID](#)

```
CON_ID_TO_DBID(container_guid)
```

[CON_NAME_TO_ID](#)

```
CON_NAME_TO_ID(container_name)
```

[CON_UID_TO_ID](#)

```
CON_UID_TO_ID(container_uid)
```

[CONCAT](#)

```
CONCAT(char1, char2)
```

[CONVERT](#)

```
CONVERT(char, dest_char_set[, source_char_set ])
```

[CORR](#)

```
CORR(expr1, expr2) [ OVER (analytic_clause) ]
```

[CORR_K](#), [CORR_S](#)

```
{ CORR_K | CORR_S }  
  (expr1, expr2  
    [, { COEFFICIENT  
          | ONE_SIDED_SIG  
          | ONE_SIDED_SIG_POS  
          | ONE_SIDED_SIG_NEG  
          | TWO_SIDED_SIG  
        }  
    ]  
  )
```

[COS](#)

```
COS(n)
```

[COSH](#)

```
COSH(n)
```

[COUNT](#)

```
COUNT({ * | [ DISTINCT | ALL ] expr }) [ OVER (analytic_clause) ]
```

[COVAR_POP](#)

```
COVAR_POP(expr1, expr2)  
  [ OVER (analytic_clause) ]
```

[COVAR_SAMP](#)

```
COVAR_SAMP(expr1, expr2) [ OVER (analytic_clause) ]
```

[CUBE_TABLE](#)

```
CUBE_TABLE  
( ' { schema.cube [ {HIERARCHY | HRR} dimension hierarchy ]...  
    | schema.dimension [ {HIERARCHY | HRR} [dimension] hierarchy ]  
    }  
  ,  
  )
```

[CUME_DIST](#)(集計)

```
CUME_DIST(expr[,expr ]...) WITHIN GROUP  
  (ORDER BY expr [ DESC | ASC ]
```

```
        [ NULLS { FIRST | LAST } ]
    [, expr [ DESC | ASC ]
        [ NULLS { FIRST | LAST } ]
    ]...
)
```

[CUME_DIST\(分析\)](#)

```
CUME_DIST() OVER ([ query_partition_clause ] order_by_clause)
```

[CURRENT_DATE](#)

```
CURRENT_DATE
```

[CURRENT_TIMESTAMP](#)

```
CURRENT_TIMESTAMP [ (precision) ]
```

[CV](#)

```
CV([ dimension_column ])
```

[DATAOBJ_TO_MAT_PARTITION](#)

```
DATAOBJ_TO_MAT_PARTITION( table, partition_id )
```

[DATAOBJ_TO_PARTITION](#)

```
DATAOBJ_TO_PARTITION( table, partition_id )
```

[DBTIMEZONE](#)

```
DBTIMEZONE
```

[DECODE](#)

```
DECODE(expr, search, result [, search, result ]... [, default ])
```

[DECOMPOSE](#)

```
DECOMPOSE( string [, { 'CANONICAL' | 'COMPATIBILITY' } ] )
```

[DENSE_RANK\(集計\)](#)

```
DENSE_RANK(expr [, expr ]...) WITHIN GROUP
  (ORDER BY expr [ DESC | ASC ]
    [ NULLS { FIRST | LAST } ]
    [,expr [ DESC | ASC ]
    [ NULLS { FIRST | LAST } ]
  )...
)
```

[DENSE_RANK\(分析\)](#)

```
DENSE_RANK( ) OVER([ query_partition_clause ] order_by_clause)
```

[DEPTH](#)

```
DEPTH(correlation_integer)
```

[DEREF](#)

```
DEREF(expr)
```

[DUMP](#)

```
DUMP(expr[, return_fmt [, start_position [, length ] ]])
```

[EMPTY_BLOB、EMPTY_CLOB](#)

```
{ EMPTY_BLOB | EMPTY_CLOB }( )
```

[EXISTSNODE](#)

```
EXISTSNODE(XMLType_instance, XPath_string [, namespace_string ])
```

[EXP](#)

```
EXP(n)
```

[EXTRACT\(日時\)](#)

```
EXTRACT( { YEAR  
        | MONTH  
        | DAY  
        | HOUR  
        | MINUTE  
        | SECOND  
        | TIMEZONE_HOUR  
        | TIMEZONE_MINUTE  
        | TIMEZONE_REGION  
        | TIMEZONE_ABBR  
        }  
        FROM { expr }  
        )
```

[EXTRACT \(XML\)](#)

```
EXTRACT(XMLType_instance, XPath_string [, namespace_string ])
```

[EXTRACTVALUE](#)

```
EXTRACTVALUE(XMLType_instance, XPath_string [, namespace_string ])
```

[FEATURE_COMPARE](#)

```
FEATURE_COMPARE ( [ schema . ] model  
                  mining_attribute_clause AND mining_attribute_clause )
```

[FEATURE_DETAILS\(集計\)](#)

```
FEATURE_DETAILS ( [ schema . ] model  
                  [, feature_id [ , topN ] ] [ DESC | ASC | ABS ]  
                  mining_attribute_clause )
```

[FEATURE_DETAILS\(分析\)](#)

```
FEATURE_DETAILS ( INTO n  
                  [, feature_id [ , topN ] ] [ DESC | ASC | ABS ]  
                  mining_attribute_clause )  
                  OVER ( mining_analytic_clause )
```

[FEATURE_ID\(集計\)](#)

```
FEATURE_ID( [ schema . ] model mining_attribute_clause )
```

[FEATURE_ID\(分析\)](#)

```
FEATURE_ID ( INTO n mining_attribute_clause )  
            OVER ( mining_analytic_clause )
```

[FEATURE_SET\(集計\)](#)

```
FEATURE_SET ( [ schema . ] model [, topN [, cutoff ]] mining_attribute_clause )
```

[FEATURE_SET\(分析\)](#)

```
FEATURE_SET ( INTO n [, topN [, cutoff ]] mining_attribute_clause )  
            OVER ( mining_analytic_clause )
```

[FEATURE_VALUE \(集計\)](#)

```
FEATURE_VALUE ( [ schema . ] model [, feature_id ] mining_attribute_clause )
```

[FEATURE_VALUE\(分析\)](#)

```
FEATURE_VALUE ( INTO n [, feature_id ] mining_attribute_clause )  
              OVER ( mining_analytic_clause )
```

[FIRST](#)

```
aggregate_function  
  KEEP  
  (DENSE_RANK FIRST ORDER BY  
   expr [ DESC | ASC ]  
     [ NULLS { FIRST | LAST } ]  
   [, expr [ DESC | ASC ]  
     [ NULLS { FIRST | LAST } ]  
   ]...  
  )  
  [ OVER ( [query_partition_clause] ) ]
```

[FIRST_VALUE](#)

```
FIRST_VALUE  
  { (expr) [ {RESPECT | IGNORE} NULLS ]  
    | (expr [ {RESPECT | IGNORE} NULLS ])  
  }  
  OVER (analytic_clause)
```

[FLOOR](#)

```
FLOOR(n)
```

[FROM_TZ](#)

```
FROM_TZ (timestamp_value, time_zone_value)
```

[GREATEST](#)

```
GREATEST(expr [, expr ]...)
```

[GROUP_ID](#)

```
GROUP_ID( )
```

[GROUPING](#)

```
GROUPING(expr)
```

[GROUPING_ID](#)

GROUPING_ID(expr [, expr]...)

[HEXTORAW](#)

HEXTORAW(char)

[INITCAP](#)

INITCAP(char)

[INSTR](#)

```
{ INSTR
| INSTRB
| INSTRC
| INSTR2
| INSTR4
}
(string , substring [, position [, occurrence ] ])
```

[ITERATION_NUMBER](#)

ITERATION_NUMBER

[JSON_ARRAY](#)

JSON_ARRAY (JSON_ARRAY_content) | JSON [JSON_ARRAY_content]

[JSON_ARRAYAGG](#)

```
JSON_ARRAYAGG
( expr [ FORMAT JSON ] [ order_by_clause ]
  [ JSON_on_null_clause ] [ JSON_agg_returning_clause ]
  [ STRICT ] )
```

[JSON_DATAGUIDE](#)

JSON_DATAGUIDE (expr [, format [, flag]])

[JSON_MERGEPATCH](#)

```
JSON_MERGEPATCH
( target_expr , patch_expr [ returning_clause ] [ PRETTY ] [ ASCII ]
  [ TRUNCATE ] [ on_error_clause ] )
```

[JSON_OBJECT](#)

JSON_OBJECT (JSON_OBJECT_content) JSON | { JSON_OBJECT_content }

[JSON_OBJECTAGG](#)

```
JSON_OBJECTAGG
( [ KEY ] key_expr VALUE val_expr [ FORMAT JSON ]
  [ JSON_on_null_clause ] [ JSON_agg_returning_clause ]
  [ STRICT ] [ WITH UNIQUE KEYS ] )
```

[JSON_QUERY](#)

```
JSON_QUERY
( expr [ FORMAT JSON ], JSON_basic_path_expression
  [ JSON_query_returning_clause ] [ JSON_query_wrapper_clause ]
  [ JSON_query_on_error_clause ] [ JSON_query_on_empty_clause ]
)
```

[JSON_SERIALIZE](#)

```
JSON_SERIALIZE
( expr [ JSON_returning_clause ] [ PRETTY ] [ ASCII ] [ TRUNCATE ]
  [ { NULL | ERROR | ( EMPTY {ARRAY | OBJECT} ) } ON ERROR ]
)
```

[JSON_TABLE](#)

```
JSON_TABLE
( expr [ FORMAT JSON ] [ , JSON_basic_path_expression ]
  [ JSON_table_on_error_clause ] [ JSON_table_on_empty_clause ]
  JSON_columns_clause )
```

[JSON_VALUE](#)

```
JSON_VALUE
( expr [ FORMAT JSON ] , [ JSON_basic_path_expression ]
  [ JSON_value_returning_clause ] [ JSON_value_on_error_clause ]
  [ JSON_value_on_empty_clause ] [ JSON_value_on_mismatch_clause ]
)
```

[JSON_TRANSFORM](#)

```
JSON_TRANSFORM
( input_expr , operation [ , operation ]... [ JSON_TRANSFORM_returning_clause ]
  [ JSON_passing_clause ] )
```

[LAG](#)

```
LAG
{ ( value_expr [, offset [, default]]) [ { RESPECT | IGNORE } NULLS ]
  | ( value_expr [ { RESPECT | IGNORE } NULLS ] [, offset [, default]])
}
OVER ([ query_partition_clause ] order_by_clause)
```

[LAST](#)

```
aggregate_function KEEP
(DENSE_RANK LAST ORDER BY
  expr [ DESC | ASC ]
    [ NULLS { FIRST | LAST } ]
  [, expr [ DESC | ASC ]
    [ NULLS { FIRST | LAST } ]
  ]...
)
[ OVER ( [query_partition_clause] ) ]
```

[LAST_DAY](#)

```
LAST_DAY(date)
```

[LAST_VALUE](#)

```
LAST_VALUE
{ (expr) [ { RESPECT | IGNORE } NULLS ]
  | (expr [ { RESPECT | IGNORE } NULLS ])
  OVER (analytic_clause)
```

[LEAD](#)

```
LEAD
{ ( value_expr [, offset [, default]]) [ { RESPECT | IGNORE } NULLS ]
```

```
| ( value_expr [ { RESPECT | IGNORE } NULLS ] [, offset [, default]] )  
}  
OVER ([ query_partition_clause ] order_by_clause)
```

[LEAST](#)

```
LEAST(expr [, expr ]...)
```

[LENGTH](#)

```
{ LENGTH  
| LENGTHB  
| LENGTHC  
| LENGTH2  
| LENGTH4  
}  
(char)
```

[LISTAGG](#)

```
LISTAGG( [ ALL | DISTINCT ] measure_expr  
        [, 'delimiter' [listagg_overflow_clause] ]  
        [ WITHIN GROUP order_by_clause ]  
        [OVER query_partition_clause]
```

[LN](#)

```
LN(n)
```

[LNNVL](#)

```
LNNVL(condition)
```

[LOCALTIMESTAMP](#)

```
LOCALTIMESTAMP [ (timestamp_precision) ]
```

[LOG](#)

```
LOG(n2, n1)
```

[LOWER](#)

```
LOWER(char)
```

[LPAD](#)

```
LPAD(expr1, n [, expr2 ])
```

[LTRIM](#)

```
LTRIM(char [, set ])
```

[MAKE_REF](#)

```
MAKE_REF({ table | view } , key [, key ]...)
```

[MAX](#)

```
MAX([ DISTINCT | ALL ] expr) [ OVER (analytic_clause) ]
```

[MEDIAN](#)

`MEDIAN(expr) [OVER (query_partition_clause)]`

[MIN](#)

`MIN([DISTINCT | ALL] expr) [OVER (analytic_clause)]`

[MOD](#)

`MOD(n2, n1)`

[MONTHS_BETWEEN](#)

`MONTHS_BETWEEN(date1, date2)`

[NANVL](#)

`NANVL(n2, n1)`

[NCHR](#)

`NCHR(number)`

[NEW_TIME](#)

`NEW_TIME(date, timezone1, timezone2)`

[NEXT_DAY](#)

`NEXT_DAY(date, char)`

[NLS_CHARSET_DECL_LEN](#)

`NLS_CHARSET_DECL_LEN(byte_count, char_set_id)`

[NLS_CHARSET_ID](#)

`NLS_CHARSET_ID(string)`

[NLS_CHARSET_NAME](#)

`NLS_CHARSET_NAME(number)`

[NLS_COLLATION_ID](#)

`NLS_COLLATION_ID(expr)`

[NLS_COLLATION_NAME](#)

`NLS_COLLATION_NAME(expr [, flag])`

[NLS_INITCAP](#)

`NLS_INITCAP(char [, 'nlsparam'])`

[NLS_LOWER](#)

`NLS_LOWER(char [, 'nlsparam'])`

[NLS_UPPER](#)

`NLS_UPPER(char [, 'nlsparam'])`

[NLSSORT](#)

```
NLSSORT(char [, 'nlsparm' ])
```

[NTH_VALUE](#)

```
NTH_VALUE(measure_expr, n)  
[ FROM { FIRST | LAST } ][ { RESPECT | IGNORE } NULLS ]  
OVER (analytic_clause)
```

[NTILE](#)

```
NTILE(expr) OVER ([ query_partition_clause ] order_by_clause)
```

[NULLIF](#)

```
NULLIF(expr1, expr2)
```

[NUMTODSINTERVAL](#)

```
NUMTODSINTERVAL(n, 'interval_unit')
```

[NUMTOYMINTERVAL](#)

```
NUMTOYMINTERVAL(n, 'interval_unit')
```

[NVL](#)

```
NVL(expr1, expr2)
```

[NVL2](#)

```
NVL2(expr1, expr2, expr3)
```

[ORA_DM_PARTITION_NAME](#)

```
ORA_DM_PARTITION_NAME ( [ schema . ] model mining_attribute_clause )
```

[ORA_DST_AFFECTED](#)

```
ORA_DST_AFFECTED(datetime_expr)
```

[ORA_DST_CONVERT](#)

```
ORA_DST_CONVERT(datetime_expr [, integer [, integer ]])
```

[ORA_DST_ERROR](#)

```
ORA_DST_ERROR(datetime_expr)
```

[ORA_HASH](#)

```
ORA_HASH(expr [, max_bucket [, seed_value ] ])
```

[ORA_INVOKING_USER](#)

```
ORA_INVOKING_USER
```

[ORA_INVOKING_USERID](#)

```
ORA_INVOKING_USERID
```

[PATH](#)

```
PATH(correlation_integer)
```

[PERCENT_RANK\(集計\)](#)

```
PERCENT_RANK(expr [, expr ]...) WITHIN GROUP  
(ORDER BY  
  expr [ DESC | ASC ]  
    [NULLS { FIRST | LAST } ]  
  [, expr [ DESC | ASC ]  
    [NULLS { FIRST | LAST } ]  
  ]...  
)
```

[PERCENT_RANK\(分析\)](#)

```
PERCENT_RANK( )  
  OVER ([ query_partition_clause ] order_by_clause)
```

[PERCENTILE_CONT](#)

```
PERCENTILE_CONT(expr) WITHIN GROUP  
(ORDER BY expr [ DESC | ASC ])  
  [ OVER (query_partition_clause) ]
```

[PERCENTILE_DISC](#)

```
PERCENTILE_DISC(expr) WITHIN GROUP  
(ORDER BY expr [ DESC | ASC ])  
  [ OVER (query_partition_clause) ]
```

[POWER](#)

```
POWER(n2, n1)
```

[POWERMULTISET](#)

```
POWERMULTISET(expr)
```

[POWERMULTISET_BY_CARDINALITY](#)

```
POWERMULTISET_BY_CARDINALITY(expr, cardinality)
```

[PREDICTION \(集計\)](#)

```
PREDICTION ( [ grouping_hint ] [ schema . ] model  
  [ cost_matrix_clause ] mining_attribute_clause )
```

[PREDICTION\(分析\)](#)

```
PREDICTION ( ( OF ANOMALY | FOR expr ) [ cost_matrix_clause ]  
  mining_attribute_clause )  
  OVER ( mining_analytic_clause )
```

[PREDICTION_BOUNDS](#)

```
PREDICTION_BOUNDS ( [schema.] model [, confidence_level [, class_value]]  
  mining_attribute_clause )
```

[PREDICTION_COST \(集計\)](#)

```
PREDICTION_COST ( [ schema . ] model [ , class ] cost_matrix_clause
mining_attribute_clause )
```

[PREDICTION_COST\(分析\)](#)

```
PREDICTION_COST ( ( OF ANOMALY | FOR expr ) [ , class ]
cost_matrix_clause mining_attribute_clause )
OVER (mining_analytic_clause)
```

[PREDICTION_DETAILS \(集計\)](#)

```
PREDICTION_DETAILS ( [ schema . ] model
[ , class_value [ , topN ] ] [ DESC | ASC | ABS ]
mining_attribute_clause )
```

[PREDICTION_DETAILS \(分析\)](#)

```
PREDICTION_DETAILS ( ( OF ANOMALY | FOR expr ) [ , class_value [ , topN ] ]
[ DESC | ASC | ABS ] mining_attribute_clause )
OVER ( mining_analytic_clause )
```

[PREDICTION_PROBABILITY \(集計\)](#)

```
PREDICTION_PROBABILITY ( [ schema . ] model [ , class ] mining_attribute_clause )
```

[PREDICTION_PROBABILITY \(分析\)](#)

```
PREDICTION_PROBABILITY ( ( OF ANOMALY | FOR expr ) [ , class ]
mining_attribute_clause )
OVER (mining_analytic_clause )
```

[PREDICTION_SET \(集計\)](#)

```
PREDICTION_SET ( [ schema . ] model [ , bestN [ , cutoff ] ]
[ cost_matrix_clause ] mining_attribute_clause )
```

[PREDICTION_SET \(分析\)](#)

```
PREDICTION_SET ( ( OF ANOMALY | FOR "expr" ) [ , bestN [ , cutoff ] ]
[ cost_matrix_clause ] mining_attribute_clause )
OVER ( mining_analytic_clause )
```

[PRESENTNNV](#)

```
PRESENTNNV(cell_reference, expr1, expr2)
```

[PRESENTV](#)

```
PRESENTV(cell_reference, expr1, expr2)
```

[PREVIOUS](#)

```
PREVIOUS(cell_reference)
```

[RANK\(集計\)](#)

```
RANK(expr [, expr ]...) WITHIN GROUP
(ORDER BY
  expr [ DESC | ASC ]
  [ NULLS { FIRST | LAST } ]
  [, expr [ DESC | ASC ]
  [ NULLS { FIRST | LAST } ]
]...
```

)

[RANK\(分析\)](#)

```
RANK( )  
  OVER ([ query_partition_clause ] order_by_clause)
```

[RATIO_TO_REPORT](#)

```
RATIO_TO_REPORT(expr)  
  OVER ([ query_partition_clause ])
```

[RAWTOHEX](#)

```
RAWTOHEX(raw)
```

[RAWTONHEX](#)

```
RAWTONHEX(raw)
```

[REF](#)

```
REF (correlation_variable)
```

[REFTOHEX](#)

```
REFTOHEX (expr)
```

[REGEXP_COUNT](#)

```
REGEXP_COUNT (source_char, pattern [, position [, match_param]])
```

[REGEXP_INSTR](#)

```
REGEXP_INSTR ( source_char, pattern  
                [, position  
                [, occurrence  
                [, return_opt  
                [, match_param  
                [, subexpr ]  
                ]  
              ]  
            ]  
          )
```

[REGEXP_REPLACE](#)

```
REGEXP_REPLACE ( source_char, pattern  
                 [, replace_string  
                 [, position  
                 [, occurrence  
                 [, match_param ]  
                 ]  
               ]  
             )
```

[REGEXP_SUBSTR](#)

```
REGEXP_SUBSTR ( source_char, pattern  
                [, position  
                [, occurrence  
                [, match_param
```

```
        [, subexpr ]
      ]
    ]
  )
```

[REGR_AVGX](#)、[REGR_AVGY](#)、[REGR_COUNT](#)、[REGR_INTERCEPT](#)、[REGR_R2](#)、[REGR_SLOPE](#)、[REGR_SXX](#)、[REGR_SXY](#)、[REGR_SYY](#)

```
{ REGR_SLOPE
| REGR_INTERCEPT
| REGR_COUNT
| REGR_R2
| REGR_AVGX
| REGR_AVGY
| REGR_SXX
| REGR_SYY
| REGR_SXY
}
(expr1 , expr2)
[ OVER (analytic_clause) ]
```

[REMAINDER](#)

```
REMAINDER(n2, n1)
```

[REPLACE](#)

```
REPLACE(char, search_string
        [, replacement_string ]
      )
```

[ROUND \(日付\)](#)

```
ROUND(date [, fmt ])
```

[ROUND \(数値\)](#)

```
ROUND(n [, integer ])
```

[ROUND_TIES_TO_EVEN \(number\)](#)

```
ROUND_TIES_TO_EVEN ( n [, integer ] )
```

[ROW_NUMBER](#)

```
ROW_NUMBER( )
  OVER ([ query_partition_clause ] order_by_clause)
```

[ROWIDTOCHAR](#)

```
ROWIDTOCHAR(rowid)
```

[ROWIDTONCHAR](#)

```
ROWIDTONCHAR(rowid)
```

[RPAD](#)

```
RPAD(expr1 , n [, expr2 ])
```

[RTRIM](#)

RTRIM(char [, set])

[SCN_TO_TIMESTAMP](#)

SCN_TO_TIMESTAMP(number)

[SESSIONTIMEZONE](#)

SESSIONTIMEZONE

[SET](#)

SET (nested_table)

[SIGN](#)

SIGN(n)

[SIN](#)

SIN(n)

[SINH](#)

SINH(n)

[SOUNDEX](#)

SOUNDEX(char)

[SQRT](#)

SQRT(n)

[STANDARD_HASH](#)

STANDARD_HASH(expr [, 'method'])

[STATS_BINOMIAL_TEST](#)

```
STATS_BINOMIAL_TEST(expr1, expr2, p
                    [, { TWO_SIDED_PROB
                        | EXACT_PROB
                        | ONE_SIDED_PROB_OR_MORE
                        | ONE_SIDED_PROB_OR_LESS
                      }
                    ]
                   )
```

[STATS_CROSSTAB](#)

```
STATS_CROSSTAB(expr1, expr2
               [, { CHISQ_OBS
                   | CHISQ_SIG
                   | CHISQ_DF
                   | PHI_COEFFICIENT
                   | CRAMERS_V
                   | CONT_COEFFICIENT
                   | COHENS_K
                 }
               ]
              )
```

[STATS_F_TEST](#)

```
STATS_F_TEST(expr1, expr2
             [, { { STATISTIC
                   | DF_NUM
                   | DF_DEN
                   | ONE_SIDED_SIG
             } , expr3
             | TWO_SIDED_SIG
             }
           ]
        )
```

[STATS_KS_TEST](#)

```
STATS_KS_TEST(expr1, expr2
              [, { STATISTIC | SIG } ]
            )
```

[STATS_MODE](#)

```
STATS_MODE(expr)
```

[STATS_MW_TEST](#)

```
STATS_MW_TEST(expr1, expr2
              [, { STATISTIC
                   | U_STATISTIC
                   | ONE_SIDED_SIG , expr3
                   | TWO_SIDED_SIG
             }
            ]
           )
```

[STATS_ONE_WAY_ANOVA](#)

```
STATS_ONE_WAY_ANOVA(expr1, expr2
                    [, { SUM_SQUARES_BETWEEN
                          | SUM_SQUARES_WITHIN
                          | DF_BETWEEN
                          | DF_WITHIN
                          | MEAN_SQUARES_BETWEEN
                          | MEAN_SQUARES_WITHIN
                          | F_RATIO
                          | SIG
                    }
                   ]
                  )
```

[STATS_T_TEST_INDEP](#)、[STATS_T_TEST_INDEPU](#)、[STATS_T_TEST_ONE](#)、[STATS_T_TEST_PAIR](#)

```
{
  STATS_T_TEST_ONE ( expr1 [, expr2 ]
  |
  { { STATS_T_TEST_PAIR
      | STATS_T_TEST_INDEP
      | STATS_T_TEST_INDEPU
    } ( expr1, expr2
  }
}
[, { { STATISTIC | ONE_SIDED_SIG } , expr3 | TWO_SIDED_SIG | DF } ] )
```

[STATS_WSR_TEST](#)

```
STATS_WSR_TEST(expr1, expr2
```



```
[, { STATISTIC
    | ONE_SIDED_SIG
    | TWO_SIDED_SIG
  }
]
```

[STDDEV](#)

```
STDDEV([ DISTINCT | ALL ] expr)
  [ OVER (analytic_clause) ]
```

[STDDEV_POP](#)

```
STDDEV_POP(expr)
  [ OVER (analytic_clause) ]
```

[STDDEV_SAMP](#)

```
STDDEV_SAMP(expr)
  [ OVER (analytic_clause) ]
```

[SUBSTR](#)

```
{ SUBSTR
| SUBSTRB
| SUBSTRC
| SUBSTR2
| SUBSTR4
}
(char, position [, substring_length ])
```

[SUM](#)

```
SUM([ DISTINCT | ALL ] expr)
  [ OVER (analytic_clause) ]
```

[SYS_CONNECT_BY_PATH](#)

```
SYS_CONNECT_BY_PATH(column, char)
```

[SYS_CONTEXT](#)

```
SYS_CONTEXT('namespace', 'parameter' [, length ])
```

[SYS_DBURIGEN](#)

```
SYS_DBURIGEN({ column | attribute }
  [ rowid ]
  [, { column | attribute }
  [ rowid ]
  ]...
  [, 'text ( )' ]
)
```

[SYS_EXTRACT_UTC](#)

```
SYS_EXTRACT_UTC(datetime_with_timezone)
```

[SYS_GUID](#)

```
SYS_GUID( )
```

[SYS_OP_ZONE_ID](#)

```
SYS_OP_ZONE_ID( [ [ schema. ] table. | t_alias. ] rowid [, scale ] )
```

[SYS_TYPEID](#)

```
SYS_TYPEID(object_type_value)
```

[SYS_XMLAGG](#)

```
SYS_XMLAGG(expr [, fmt ])
```

[SYS_XMLGEN](#)

```
SYS_XMLGEN(expr [, fmt ])
```

[SYSDATE](#)

```
SYSDATE
```

[SYSTIMESTAMP](#)

```
SYSTIMESTAMP
```

[TAN](#)

```
TAN(n)
```

[TANH](#)

```
TANH(n)
```

[TIMESTAMP_TO_SCN](#)

```
TIMESTAMP_TO_SCN(timestamp)
```

[TO_APPROX_COUNT_DISTINCT](#)

```
TO_APPROX_COUNT_DISTINCT(detail)
```

[TO_APPROX_PERCENTILE](#)

```
TO_APPROX_PERCENTILE(detail, expr, 'datatype'  
[, { 'DESC' | 'ASC' | 'ERROR_RATE' | 'CONFIDENCE' } ])
```

[TO_BINARY_DOUBLE](#)

```
TO_BINARY_DOUBLE(expr [ DEFAULT return_value ON CONVERSION ERROR ]  
[, fmt [, 'nlsparam' ] ])
```

[TO_BINARY_FLOAT](#)

```
TO_BINARY_FLOAT(expr [ DEFAULT return_value ON CONVERSION ERROR ]  
[, fmt [, 'nlsparam' ] ])
```

[TO_BLOB \(bfile\)](#)

```
TO_BLOB( bfile [, mime_type] )
```

[TO_BLOB \(raw\)](#)

```
TO_BLOB( raw_value )
```

[TO_CHAR \(bfile|blob\)](#)

```
TO_CHAR( { bfile | blob } [, csid] )
```

[TO_CHAR\(文字\)](#)

```
TO_CHAR(nchar | clob | nclob)
```

[TO_CHAR\(日時\)](#)

```
TO_CHAR({ datetime | interval } [, fmt [, 'nlsparam' ] ])
```

[TO_CHAR\(数值\)](#)

```
TO_CHAR(n [, fmt [, 'nlsparam' ] ])
```

[TO_CLOB \(bfile|blob\)](#)

```
TO_CLOB( { bfile | blob } [, csid] [, mime_type] )
```

[TO_CLOB\(文字\)](#)

```
TO_CLOB(lob_column | char)
```

[TO_DATE](#)

```
TO_DATE(char [ DEFAULT return_value ON CONVERSION ERROR ]  
[, fmt [, 'nlsparam' ] ])
```

[TO_DSINTERVAL](#)

```
TO_DSINTERVAL ( ' { sql_format | ds_iso_format } '  
[ DEFAULT return_value ON CONVERSION ERROR ] )
```

[TO_LOB](#)

```
TO_LOB(long_column)
```

[TO_MULTI_BYTE](#)

```
TO_MULTI_BYTE(char)
```

[TO_NCHAR\(文字\)](#)

```
TO_NCHAR({char | clob | nclob})
```

[TO_NCHAR\(日時\)](#)

```
TO_NCHAR({ datetime | interval }  
[, fmt [, 'nlsparam' ] ]  
)
```

[TO_NCHAR\(数值\)](#)

```
TO_NCHAR(n [, fmt [, 'nlsparam' ] ])
```

[TO_NCLOB](#)

```
TO_NCLOB(lob_column | char)
```

[TO_NUMBER](#)

```
TO_NUMBER(expr [ DEFAULT return_value ON CONVERSION ERROR ]
[, fmt [, 'nlsparam' ] ])
```

[TO_SINGLE_BYTE](#)

```
TO_SINGLE_BYTE(char)
```

[TO_TIMESTAMP](#)

```
TO_TIMESTAMP(char [ DEFAULT return_value ON CONVERSION ERROR ]
[, fmt [, 'nlsparam' ] ])
```

[TO_TIMESTAMP_TZ](#)

```
TO_TIMESTAMP_TZ(char [ DEFAULT return_value ON CONVERSION ERROR ]
[, fmt [, 'nlsparam' ] ])
```

[TO_UTC_TIMESTAMP_TZ](#)

```
TO_UTC_TIMESTAMP_TZ ( varchar )
)
```

[TO_YMINTERVAL](#)

```
TO_YMINTERVAL
( ' { [+|-] years - months
    | ym_iso_format
  } '
[ DEFAULT return_value ON CONVERSION ERROR ]
)
```

[TRANSLATE](#)

```
TRANSLATE(expr, from_string, to_string)
```

[TRANSLATE ... USING](#)

```
TRANSLATE ( char USING
            { CHAR_CS | NCHAR_CS }
            )
```

[TREAT](#)

```
TREAT(expr AS [ REF ] [ schema. ]type)
```

[TRIM](#)

```
TRIM([ { { LEADING | TRAILING | BOTH }
        [ trim_character ]
        | trim_character
      }
FROM
]
trim_source
)
```

[TRUNC\(日付\)](#)

```
TRUNC(date [, fmt ])
```

[TRUNC\(数值\)](#)

```
TRUNC(n1 [, n2 ])
```

[TZ_OFFSET](#)

```
TZ_OFFSET({ 'time_zone_name'  
           | '{ + | - } hh : mi'  
           | SESSIONTIMEZONE  
           | DBTIMEZONE  
           }  
)
```

[UID](#)

UID

[UNISTR](#)

UNISTR(string)

[UPPER](#)

UPPER(char)

[USER](#)

USER

[ユーザー定義ファンクション](#)

```
[ schema. ]  
{ [ package. ]function | user_defined_operator }  
[ @ dblink. ]  
[ ( [ [ DISTINCT | ALL ] expr [, expr ]... ] ) ]
```

[USERENV](#)

USERENV('parameter')

[VALIDATE_CONVERSION](#)

```
VALIDATE_CONVERSION(expr AS type_name  
[, fmt [, 'nlsparam' ] ])
```

[VALUE](#)

VALUE(correlation_variable)

[VAR_POP](#)

VAR_POP(expr) [OVER (analytic_clause)]

[VAR_SAMP](#)

VAR_SAMP(expr) [OVER (analytic_clause)]

[VARIANCE](#)

```
VARIANCE([ DISTINCT | ALL ] expr)  
[ OVER (analytic_clause) ]
```

[VSIZE](#)

VSIZE(expr)

[WIDTH_BUCKET](#)

```
WIDTH_BUCKET  
(expr, min_value, max_value, num_buckets)
```

[XMLAGG](#)

```
XMLAGG(XMLType_instance [ order_by_clause ])
```

[XMLCAST](#)

```
XMLCAST ( value_expression AS datatype )
```

[XMLCDATA](#)

```
XMLCDATA ( value_expr )
```

[XMLCOLATTVAL](#)

```
XMLCOLATTVAL  
(value_expr [ AS { c_alias | EVALNAME value_expr } ]  
 [, value_expr [ AS { c_alias | EVALNAME value_expr } ]  
 ]...  
)
```

[XMLCOMMENT](#)

```
XMLCOMMENT ( value_expr )
```

[XMLCONCAT](#)

```
XMLCONCAT(XMLType_instance [, XMLType_instance ]...)
```

[XMLDIFF](#)

```
XMLDIFF ( XMLType_document, XMLType_document [ , integer, string ] )
```

[XMLELEMENT](#)

```
XMLELEMENT ( [ ENTITYESCAPING | NOENTITYESCAPING ]  
 { ( [ NAME ] identifier ) | ( EVALNAME value_expr ) }  
 [ , XML_attributes_clause ]  
 [ , value_expr [ [ AS ] c_alias ] ]... )
```

[XMLEXISTS](#)

```
XMLEXISTS ( XQuery_string [ XML_passing_clause ] )
```

[XMLFOREST](#)

```
XMLFOREST  
( value_expr [ AS { c_alias | EVALNAME value_expr } ]  
 [, value_expr [ AS { c_alias | EVALNAME value_expr } ]  
 ]...  
)
```

[XMLISVALID](#)

```
XMLISVALID ( XMLType_instance [, XMLSchema_URL [, element ]] )
```

[XMLPARSE](#)

```
XMLPARSE
```

```
( { DOCUMENT | CONTENT } value_expr [ WELLFORMED ]  
)
```

[XMLPATCH](#)

```
XMLPATCH ( XMLType_document, XMLType_document )
```

[XMLPI](#)

```
XMLPI  
( { ( [ NAME ] identifier ) | ( EVALNAME value_expr ) }  
  [ , value_expr ]  
)
```

[XMLQUERY](#)

```
XMLQUERY  
( XQuery_string  
  [ XML_passing_clause ]  
  RETURNING CONTENT [NULL ON EMPTY]  
)
```

[XMLROOT](#)

```
XMLROOT  
( value_expr, VERSION  
  { value_expr | NO VALUE }  
  [, STANDALONE { YES | NO | NO VALUE } ]  
)
```

[XMLSEQUENCE](#)

```
XMLSEQUENCE( XMLType_instance  
             | sys_refcursor_instance [, fmt ]  
)
```

[XMLSERIALIZE](#)

```
XMLSERIALIZE  
( { DOCUMENT | CONTENT } value_expr [ AS datatype ]  
  [ ENCODING xml_encoding_spec ]  
  [ VERSION string_literal ]  
  [ NO INDENT | { INDENT [SIZE = number] } ]  
  [ { HIDE | SHOW } DEFAULTS ]  
)
```

[XMLTABLE](#)

```
XMLTABLE  
(  
  [ XMLnamespaces_clause , ] XQuery_string XMLTABLE_options  
)
```

[XMLTRANSFORM](#)

```
XMLTRANSFORM(XMLType_instance, { XMLType_instance  
                                  | string  
                                  }  
)
```

3 SQL式

この章では、値、演算子およびファンクションを式の中で組み合わせて使用する場合の構文について説明します。

この章の内容は次のとおりです。

- [SQL式のタイプの構文](#)

SQL式のタイプの構文

式は、1つ以上の値、演算子、およびある値に評価されるSQLファンクションの組合せです。一般に、式のデータ型は、そのコンポーネントのデータ型になります。

式にはいくつかの書式があります。この項では、各書式の式に対する構文を示します。副次句の構文については、[「副次句」](#)を参照してください。

関連項目:

SQL式の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

[計算済メジャー式](#)

```
{
  av_meas_expression
| av_simple_expression
| single_row_function_expression
| case_expression
| compound_expression
| datetime_expression
| interval_expression
}
```

[CASE式](#)

```
CASE { simple_case_expression
      | searched_case_expression
      }
      [ else_clause ]
END
```

[列の式](#)

列の式は、単純式、複合式またはファンクション式であったり、あるいは対象となる表の列、制約および決定的ファンクションのみを含む式のリストの場合もあります。

[複合式](#)

```
{ (expr)
| { + | - | PRIOR } expr
| expr { * | / | + | - | || } expr
| expr COLLATE collation_name
}
```

Note: The double vertical bars are part of the syntax (indicating concatenation) rather than BNF notation.

[CURSOR式](#)

```
CURSOR (subquery)
```


[日時式](#)

```
expr AT
{ LOCAL
| TIME_ZONE { ' [ + | - ] hh:mi'
              | DBTIMEZONE
              | 'time_zone_name'
              | expr
              }
}
```

[ファンクション式](#)

組み込みSQLファンクションまたはユーザー定義ファンクションを式として使用できます。

[期間式](#)

```
( expr1 - expr2 )
{ DAY [ (leading_field_precision) ] TO
  SECOND [ (fractional_second_precision) ]
| YEAR [ (leading_field_precision) ] TO
  MONTH
}
```

[JSONオブジェクト・アクセス式](#)

```
table_alias.JSON_column [ .JSON_object_key [ array_step ]... ]...
```

[モデル式](#)

```
{ measure_column [ { condition | expr } [, { condition | expr } ]... ]
| aggregate_function
  { [ { condition | expr } [, { condition | expr } ]... ]
  | [ single_column_for_loop [, single_column_for_loop ]... ]
  | [ multi_column_for_loop ]
  }
| analytic_function
}
```

Note: The outside square brackets shown in boldface type are part of the syntax. In this case, they do not represent optionality.

[オブジェクト・アクセス式](#)

```
{ table_alias.column.
| object_table_alias.
| (expr).
}
{ attribute [.attribute ]...
  [.method ([ argument [, argument ]... ]) ]
| method ([ argument [, argument ]... ])
}
```

[プレースホルダ式](#)

```
:host_variable
[ [ INDICATOR ]
  :indicator_variable
]
```

[スカラー副問合せ式](#)

スカラー副問合せ式は、1つの行から1つの列値のみを戻す副問合せです。

[単純式](#)

```
{ [ query_name.  
  | [schema.] { table. | view. | materialized view. }  
  | t_alias.  
  ] { column | ROWID }  
| ROWNUM  
| string  
| number  
| sequence. { CURRVAL | NEXTVAL }  
| NULL  
}
```

型コンストラクタ式

```
[ NEW ] [ schema. ]type_name  
  ([ expr [, expr ]... ])
```

4 SQL条件

この章では、1つ以上の式と論理(ブール)演算子を組み合わせて条件を指定する構文について説明します。

この章の内容は次のとおりです。

- [SQL条件のタイプの構文](#)

SQL条件のタイプの構文

条件は、1つ以上の式と論理(ブール)演算子の組合せで指定し、TRUE、FALSEまたはUNKNOWNのいずれかの値を戻します。

条件にはいくつかの書式があります。この項では、各書式の条件に対する構文を示します。副次句の構文については、[「副次句」](#)を参照してください。

関連項目:

SQL条件の詳細は、[『Oracle Database SQL言語リファレンス』](#)を参照してください。

[BETWEEN条件](#)

```
expr1 [ NOT ] BETWEEN expr2 AND expr3
```

[複合条件](#)

```
{ (condition)
| NOT condition
| condition { AND | OR } condition
}
```

[EQUALS_PATH条件](#)

```
EQUALS_PATH
(column, path_string [, correlation_integer ])
```

[EXISTS条件](#)

```
EXISTS (subquery)
```

[浮動小数点条件](#)

```
expr IS [ NOT ] { NAN | INFINITE }
```

[グループ比較条件](#)

```
{ expr
  { = | != | ^= | <> | > | < | >= | <= }
  { ANY | SOME | ALL }
  ({ expression_list | subquery })
| ( expr [, expr ]... )
  { = | != | ^= | <> }
  { ANY | SOME | ALL }
  ({ expression_list
    [, expression_list ]...
  | subquery
  }
)
```

```
}
```

!=、^=および<>は不等性テストを表します。

[IN条件](#)

```
{ expr [ NOT ] IN ( { expression_list | subquery } )  
| ( expr [, expr ]... )  
  [ NOT ] IN ( { expression_list [, expression_list ]...  
               | subquery  
               }  
             )  
}
```

[IS A SET条件](#)

```
nested_table IS [ NOT ] A SET
```

[IS ANY条件](#)

```
[ dimension_column IS ] ANY
```

[IS EMPTY条件](#)

```
nested_table IS [ NOT ] EMPTY
```

[IS JSON条件](#)

```
expr IS [ NOT ] JSON [ FORMAT JSON ] [ STRICT | LAX ]  
[ { WITH | WITHOUT } UNIQUE KEYS ]
```

[IS OF type条件](#)

```
expr IS [ NOT ] OF [ TYPE ]  
  ( [ ONLY ] [ schema. ] type  
    [, [ ONLY ] [ schema. ] type ]...  
  )
```

[IS PRESENT条件](#)

```
cell_reference IS PRESENT
```

[JSON_EQUAL条件](#)

```
JSON_EQUAL ( (expr), (expr) )
```

[JSON_EXISTS条件](#)

```
JSON_EXISTS( expr [ FORMAT JSON ], JSON_basic_path_expression  
  [ JSON_passing_clause ] [ JSON_exists_on_error_clause ]  
  [ JSON_exists_on_empty_clause ] )
```

[JSON_TEXTCONTAINS条件](#)

```
JSON_TEXTCONTAINS( column, JSON_basic_path_expression, string )
```

[LIKE条件](#)

```
char1 [ NOT ] { LIKE | LIKEC | LIKE2 | LIKE4 }  
  char2 [ ESCAPE esc_char ]
```

[論理条件](#)

```
{ NOT | AND | OR }
```

[MEMBER条件](#)

```
expr [ NOT ] MEMBER [ OF ] nested_table
```

[NULL条件](#)

```
expr IS [ NOT ] NULL
```

[REGEXP_LIKE条件](#)

```
REGEXP_LIKE(source_char, pattern  
            [, match_param ]  
            )
```

[単純比較条件](#)

```
{ expr  
  { = | != | ^= | <> | > | < | >= | <= }  
  expr  
  | (expr [, expr ]...)  
  { = | != | ^= | <> }  
  ( expression_list | subquery )  
}
```

!=、 ^=および<>は不等性テストを表します。

[SUBMULTISET条件](#)

```
nested_table1  
[ NOT ] SUBMULTISET [ OF ]  
nested_table2
```

[UNDER_PATH条件](#)

```
UNDER_PATH (column [, levels ], path_string  
            [, correlation_integer ]  
            )
```

5 副次句

この章では、SQL文、ファンクション、式および条件の構文で使用する副次句の構文について説明します。

この章の内容は次のとおりです。

- [副次句の構文](#)

副次句の構文

この項では、次の章にある各副次句の構文を示します。

- [SQL文](#)
- [SQLファンクション](#)
- [SQL式](#)
- [SQL条件](#)

関連項目:

SQL副次句の詳細は、[Oracle Database SQL言語リファレンス](#)を参照してください。

action_audit_clause

```
{ standard_actions | component_actions }...
```

activate_standby_db_clause

```
ACTIVATE  
  [ PHYSICAL | LOGICAL ]  
  STANDBY DATABASE  
  [ FINISH APPLY ]
```

add_binding_clause

```
ADD BINDING  
  (parameter_type [, parameter_type ]...)  
  RETURN (return_type)  
  [ implementation_clause ]  
  using_function_clause
```

add_column_clause

```
ADD  
  { column_definition | virtual_column_definition }  
  | ( { column_definition | virtual_column_definition }  
    [, { column_definition | virtual_column_definition } ]...  
    )  
  [ column_properties ]  
  [ ( out_of_line_part_storage [, out_of_line_part_storage]... ) ]
```

add_disk_clause

```
ADD  
  { SITE sitename [ QUORUM | REGULAR ] [ FAILGROUP failgroup_name ]  
    DISK qualified_disk_clause [, qualified_disk_clause ]...  
  }...
```

add_external_partition_attrs

```
ADD EXTERNAL PARTITION ATTRIBUTES external_table_clause  
[ REJECT LIMIT ]
```

add_filegroup_clause

```
ADD FILEGROUP filegroup_name  
  { DATABASE database_name  
  | CLUSTER cluster_name  
  | VOLUME asm_volume  
  }  
[ SET '[ file_type. ] property_name' = 'property_value' ]
```

add_hash_index_partition

```
ADD PARTITION  
  [ partition_name ]  
  [ TABLESPACE tablespace_name ]  
  [ index_compression ]  
  [ parallel_clause ]
```

add_hash_partition_clause

```
partitioning_storage_clause  
[ update_index_clauses ]  
[ parallel_clause ]  
[ read_only_clause ]  
[ indexing_clause ]
```

add_hash_subpartition

```
ADD individual_hash_subparts  
  [ dependent_tables_clause ]  
  [ update_index_clauses ]  
  [ parallel_clause ]
```

add_list_partition_clause

```
list_values_clause  
[ table_partition_description ]  
[ external_part_subpart_data_props ]  
[ ( { range_subpartition_desc [, range_subpartition_desc] ...  
  | list_subpartition_desc [, list_subpartition_desc] ...  
  | individual_hash_subparts [, individual_hash_subparts] ...  
  }  
  ) | hash_subparts_by_quantity ]  
[ update_index_clauses ]
```

add_list_subpartition

```
ADD list_subpartition_desc [, list_subpartition_desc ]...  
[ dependent_tables_clause ] [ update_index_clauses ]
```

add_logfile_clauses

```
ADD [ STANDBY ] LOGFILE  
  {  
    { [ INSTANCE 'instance_name' ] | [ THREAD 'integer' ] }  
    [ GROUP integer ] redo_log_file_spec  
    [, [ GROUP integer ] redo_log_file_spec ]...  
  | MEMBER 'filename' [ REUSE ] [, 'filename' [ REUSE ] ]...  
    TO logfile_descriptor [, logfile_descriptor ]...  
  }
```

add_mv_log_column_clause

```
ADD (column)
```

add_overflow_clause

```
ADD OVERFLOW [ segment_attributes_clause ]  
  [ ( PARTITION [ segment_attributes_clause ]  
    [, PARTITION [ segment_attributes_clause ] ]...  
    )  
  ]
```

add_period_clause

```
ADD ( period_definition )
```

add_range_partition_clause

```
range_values_clause  
[ table_partition_description ]  
[ external_part_subpart_data_props ]  
[ ( { range_subpartition_desc [, range_subpartition_desc] ...  
  | list_subpartition_desc [, list_subpartition_desc] ...  
  | individual_hash_subparts [, individual_hash_subparts] ...  
  }  
  ) | hash_subparts_by_quantity ]  
[ update_index_clauses ]
```

add_range_subpartition

```
ADD range_subpartition_desc [, range_subpartition_desc ]...  
[ dependent_tables_clause ] [ update_index_clauses ]
```

add_system_partition_clause

```
[table_partition_description]  
[update_index_clauses]
```

add_table_partition

```
ADD {  
PARTITION [ partition ] add_range_partition_clause  
  [, PARTITION [ partition ] add_range_partition_clause ]...  
| PARTITION [ partition ] add_list_partition_clause  
  [, PARTITION [ partition ] add_list_partition_clause ]...  
| PARTITION [ partition ] add_system_partition_clause  
  [, PARTITION [ partition ] add_system_partition_clause ]...  
  [ BEFORE { partition_name | partition_number } ]  
| PARTITION [ partition ] add_hash_partition_clause  
} [ dependent_tables_clause ]
```

add_update_secret

```
{ ADD | UPDATE } SECRET 'secret' FOR CLIENT 'client_identifier'  
  [ USING TAG 'tag' ]  
  [ FORCE KEYSTORE ]  
  IDENTIFIED BY { EXTERNAL STORE | keystore_password }  
  [ WITH BACKUP [ USING 'backup_identifier' ] ]
```

add_update_secret_seps

```
{ ADD | UPDATE } SECRET 'secret' FOR CLIENT 'client_identifier'  
  [ USING TAG 'tag' ]  
  TO [ LOCAL ] AUTO_LOGIN KEYSTORE directory
```


add_volume_clause

```
ADD VOLUME asm_volume SIZE size_clause [redundancy_clause]
  [ STRIPE_WIDTH integer {K | M} ]
  [ STRIPE_COLUMNS integer ]
  [ ATTRIBUTE (disk_region_clause) ]
```

advanced_index_compression

```
{ COMPRESS ADVANCED [ LOW | HIGH ] } | NOCOMPRESS
```

affinity_clauses

```
{ ENABLE AFFINITY [ schema.]table [SERVICE service_name ]
|
DISABLE AFFINITY [ schema.]table
}
```

alias_file_name

```
+diskgroup_name [ (template_name) ] /alias_name
```

all_clause

```
ALL MEMBER { NAME expression [ MEMBER CAPTION expression ]
|
CAPTION expression [ MEMBER DESCRIPTION expression ]
|
DESCRIPTION expression
}
```

allocate_extent_clause

```
ALLOCATE EXTENT
  [ ( { SIZE size_clause
|
DATAFILE 'filename'
|
INSTANCE integer
} ...
)
]
```

allow_disallow_clustering

```
{ ALLOW | DISALLOW } CLUSTERING
```

alter_automatic_partitioning

```
{ SET PARTITIONING { AUTOMATIC | MANUAL }
| SET STORE IN ( tablespace [, tablespace ]... )
}
```

alter_datafile_clause

```
DATAFILE
  { 'filename' | filenumber }
  [, 'filename' | filenumber ]...
}
{ ONLINE
| OFFLINE [ FOR DROP ]
| RESIZE size_clause
| autoextend_clause
| END BACKUP
| ENCRYPT
| DECRYPT
}
```

alter_external_table

```
{ add_column_clause
| modify_column_clauses
| drop_column_clause
| parallel_clause
| external_table_data_props
| REJECT LIMIT { integer | UNLIMITED }
| PROJECT COLUMN { ALL | REFERENCED }
}
[ add_column_clause
| modify_column_clauses
| drop_column_clause
| parallel_clause
| external_table_data_props
| REJECT LIMIT { integer | UNLIMITED }
| PROJECT COLUMN { ALL | REFERENCED }
]...
```

alter_index_partitioning

```
{ modify_index_default_attrs
| add_hash_index_partition
| modify_index_partition
| rename_index_partition
| drop_index_partition
| split_index_partition
| coalesce_index_partition
| modify_index_subpartition
}
```

alter_interval_partitioning

```
{ SET INTERVAL ( [ expr ] )
| SET STORE IN ( tablespace [, tablespace]... )
}
```

alter_iot_clauses

```
{ index_org_table_clause
| alter_overflow_clause
| alter_mapping_table_clauses
| COALESCE
}
```

alter_keystore_password

```
ALTER KEYSTORE PASSWORD
[ FORCE KEYSTORE ]
IDENTIFIED BY old_keystore_password
SET new_keystore_password
[ WITH BACKUP [ USING 'backup_identifier' ] ]
```

alter_mapping_table_clauses

```
MAPPING TABLE
{ allocate_extent_clause
| deallocate_unused_clause
}
```

alter_mv_refresh

```
REFRESH
{ { FAST | COMPLETE | FORCE }
| ON { DEMAND | COMMIT }
```

```

| { START WITH | NEXT } date
| WITH PRIMARY KEY
| USING
    { DEFAULT MASTER ROLLBACK SEGMENT
    | MASTER ROLLBACK SEGMENT rollback_segment
    }
| USING { ENFORCED | TRUSTED } CONSTRAINTS
}

```

alter_overflow_clause

```

{ add_overflow_clause
| OVERFLOW
    { segment_attributes_clause
    | allocate_extent_clause
    | shrink_clause
    | deallocate_unused_clause
    }...
}

```

alter_query_rewrite_clause

```
[ ENABLE | DISABLE ] QUERY REWRITE [ unusable_editions_clause ]
```

alter_session_set_clause

```

SET { { parameter_name = parameter_value }...
    | EDITION = edition_name
    | CONTAINER = container_name [ SERVICE = service_name ]
    | ROW ARCHIVAL VISIBILITY = { ACTIVE | ALL }
    | DEFAULT_COLLATION = { collation_name | NONE }
}

```

alter_system_reset_clause

```

parameter_name
[ { SCOPE = { MEMORY | SPFILE | BOTH }
  | SID = { 'sid' | '*' }
  }...
]

```

alter_system_set_clause

```

{ set_parameter_clause
| USE_STORED_OUTLINES = (TRUE | FALSE | category_name)
| GLOBAL_TOPIC_ENABLED = (TRUE | FALSE)
}

```

alter_table_partitioning

```

{ modify_table_default_attrs
| alter_automatic_partitioning
| alter_interval_partitioning
| set_subpartition_template
| modify_table_partition
| modify_table_subpartition
| move_table_partition
| move_table_subpartition
| add_external_partition_attrs
| add_table_partition
| coalesce_table_partition
| drop_external_partition_attrs
| drop_table_partition
| drop_table_subpartition
| rename_partition_subpart
}

```

```

| truncate_partition_subpart
| split_table_partition
| split_table_subpartition
| merge_table_partitions
| merge_table_subpartitions
| exchange_partition_subpart
}

```

alter_table_properties

```

{ { { physical_attributes_clause
| logging_clause
| table_compression
| inmemory_table_clause
| ilm_clause
| supplemental_table_logging
| allocate_extent_clause
| deallocate_unused_clause
| { CACHE | NOCACHE }
| RESULT_CACHE ( MODE {DEFAULT | FORCE} )
| upgrade_table_clause
| records_per_block_clause
| parallel_clause
| row_movement_clause
| flashback_archive_clause
}...
| RENAME TO new_table_name
} [ alter_iot_clauses ] [ alter_XMLSchema_clause ]
| { shrink_clause
| READ ONLY
| READ WRITE
| REKEY encryption_spec
| DEFAULT COLLATION collation_name
| [NO] ROW ARCHIVAL
| ADD attribute_clustering_clause
| MODIFY CLUSTERING [ clustering_when ] [ zonemap_clause ]
| DROP CLUSTERING
}
}

```

alter_tablespace_attrs

```

{ default_tablespace_params
| MINIMUM EXTENT size_clause
| RESIZE size_clause
| COALESCE
| SHRINK SPACE [ KEEP size_clause ]
| RENAME TO new_tablespace_name
| { BEGIN | END } BACKUP
| datafile_tempfile_clauses
| tablespace_logging_clauses
| tablespace_group_clause
| tablespace_state_clauses
| autoextend_clause
| flashback_mode_clause
| tablespace_retention_clause
| alter_tablespace_encryption
}

```

alter_tablespace_encryption

```

ENCRYPTION
{ { OFFLINE [ tablespace_encryption_spec ] { ENCRYPT | DECRYPT } }
| { ONLINE { [ tablespace_encryption_spec ] { ENCRYPT | REKEY } }
| DECRYPT }
[ ts_file_name_convert ] }

```

```
| { FINISH { ENCRYPT | REKEY | DECRYPT } [ ts_file_name_convert ] }  
}
```

alter_tempfile_clause

```
TEMPFILE  
  { 'filename' [, 'filename' ]...  
  | filenumber [, filenumber ]...  
  }  
  { RESIZE size_clause  
  | autoextend_clause  
  | DROP [ INCLUDING DATAFILES ]  
  | ONLINE  
  | OFFLINE  
  }
```

alter_varray_col_properties

```
MODIFY VARRAY varray_item  
  ( modify_LOB_parameters )
```

alter_XMLSchema_clause

```
{ ALLOW ANYSCHEMA  
| ALLOW NONSCHEMA  
| DISALLOW NONSCHEMA  
}
```

alter_zonemap_attributes

```
{ PCTFREE integer  
| PCTUSED integer  
| { CACHE | NOCACHE }  
}...
```

alternate_key_clause

```
ALTERNATE KEY { [ ( ] attribute [ ) ]  
                |  
                ( attribute [, attribute ]... )  
                }
```

analytic_clause

```
[ query_partition_clause ] [ order_by_clause [ windowing_clause ] ]
```

application_clauses

```
APPLICATION  
{ { app_name  
  { BEGIN INSTALL 'app_version' [ COMMENT 'comment' ]  
  | END INSTALL [ 'app_version' ]  
  | BEGIN PATCH number [ MINIMUM VERSION 'app_version' ] [ COMMENT 'comment' ]  
  | END PATCH [ number ]  
  | BEGIN UPGRADE [ 'start_app_version' ] TO 'end_app_version' [ COMMENT  
'comment' ]  
  | END UPGRADE [ TO 'end_app_version' ]  
  | BEGIN UNINSTALL  
  | END UNINSTALL  
  | SET PATCH number  
  | SET VERSION 'app_version'  
  | SET COMPATIBILITY VERSION { 'app_version' | CURRENT }  
  | SYNC TO { 'app_version' | PATCH 'patch_number' }  
  | SYNC  
  }  
}
```

```

}
|
{ ALL SYNC }
}

```

archive_log_clause

```

ARCHIVE LOG
  [ INSTANCE 'instance_name' ]
  { { SEQUENCE integer
    | CHANGE integer
    | CURRENT [ NOSWITCH ]
    | GROUP integer
    | LOGFILE 'filename'
      [ USING BACKUP CONTROLFILE ]
    | NEXT
    | ALL
    } }
  [ TO 'location' ]
}

```

array_DML_clause

```

[ WITH | WITHOUT ]
ARRAY DML
[ ([ schema. ]type
  [, [ schema. ]varray_type ]
  [, ([ schema. ]type
    [, [ schema. ]varray_type ])...
  ]
]

```

array_step

```

[ { integer | integer TO integer [, integer | integer TO integer ]... } | * ]

```

Note: The outside square brackets shown in boldface type are part of the syntax. In this case, they do not represent optionality.

ASM_filename

```

{ fully_qualified_file_name
| numeric_file_name
| incomplete_file_name
| alias_file_name
}

```

attr_dim_attributes_clause

```

[ alias. ] column [ [ AS ] attribute_name ] [ classification_clause ]...

```

attr_dim_level_clause

```

LEVEL level [ { NOT NULL | SKIP WHEN NULL } ]
[ classification_clause [ classification_clause ]...
[ LEVEL TYPE
  {
    STANDARD
    | YEARS
    | HALF_YEARS
    | QUARTERS
    | MONTHS
    | WEEKS
    | DAYS
    | HOURS
    | MINUTES
    | SECONDS
  }
]
}

```

```

]
key_clause [ alternate_key_clause ]
[ MEMBER NAME expression ]
[ MEMBER CAPTION expression ]
[ MEMBER DESCRIPTION expression ]
[ ORDER BY [ MIN | MAX ] dim_order_clause
           [, [ MIN | MAX ] dim_order_clause ]... ]
]
[ DETERMINES ( attribute [, attribute]... ) ]

```

attr_dim_using_clause

```

USING [ schema. ] dim_source [ [ AS ] alias]

```

attribute_clause

```

ATTRIBUTE level DETERMINES
  { dependent_column
  | ( dependent_column
    [, dependent_column ]... )
  }

```

attribute_clustering_clause

```

CLUSTERING [ clustering_join ] cluster_clause
           [ clustering_when ] [ zonemap_clause ]

```

attributes_clause

```

ATTRIBUTES ( attr_dim_attribute_clause [, attr_dim_attribute_clause ]... )

```

audit_operation_clause

```

{ { sql_statement_shortcut
  | ALL
  | ALL STATEMENTS
  } [, { sql_statement_shortcut
      | ALL
      }
  ]
| { system_privilege
  | ALL PRIVILEGES
  } [, { system_privilege
      | ALL PRIVILEGES
      }
  ]
}

```

audit_schema_object_clause

```

{ sql_operation [, sql_operation ]
| ALL
} auditing_on_clause

```

auditing_by_clause

```

BY user [, user ]...

```

auditing_on_clause

```

ON { [ schema. ] object
  | DIRECTORY directory_name
  | MINING MODEL [ schema. ] model
  | SQL TRANSLATION PROFILE [ schema. ] profile
}

```

```
| DEFAULT  
}
```

autoextend_clause

```
AUTOEXTEND  
{ OFF  
| ON [ NEXT size_clause ]  
  [ maxsize_clause ]  
}
```

av_meas_expression

```
{ lead_lag_expression  
| window_expression  
| share_of_expression  
| qdr_expression  
}
```

av_measure

```
meas_name [{ base_measure_clause | calc_measure_clause }]  
[ classification_clause ]...
```

av_simple_expression

```
{ string | number | NULL | measure_ref }
```

backup_keystore

```
BACKUP KEYSTORE [ USING 'backup_identifier' ]  
[ FORCE KEYSTORE ]  
IDENTIFIED BY { EXTERNAL STORE | keystore_password }  
[ TO 'keystore_location' ]
```

base_measure_clause

```
[ FACT [alias.] ] column [ meas_aggregate_clause ]
```

binding_clause

```
BINDING  
(parameter_type [, parameter_type ]...)  
RETURN return_type  
[ implementation_clause ]  
using_function_clause  
[, (parameter_type [, parameter_type ]...)  
  RETURN return_type  
  [ implementation_clause ]  
  using_function_clause  
]...
```

bitmap_join_index_clause

```
[ schema.]table  
( [ [ schema.]table. | t_alias. ]column  
  [ ASC | DESC ]  
  [, [ [ schema.]table. | t_alias. ]column  
    [ ASC | DESC ]  
  ]...  
)  
FROM [ schema.]table [ t_alias ]  
  [, [ schema.]table [ t_alias ]  
  ]...
```



```
WHERE condition
  [ local_partitioned_index ] index_attributes
```

blockchain_drop_table_clause

```
NO DROP [ UNTIL integer DAYS IDLE ]
```

blockchain_hash_and_data_format_clause

```
HASHING USING sha2_512 VERSION v1
```

blockchain_row_retention_clause

```
NO DELETE ( ([ LOCKED ]) | (UNTIL integer DAYS AFTER INSERT [ LOCKED ]) )
```

blockchain_table_clauses

```
blockchain_drop_table_clause
    blockchain_row_retention_clause
    blockchain_hash_and_data_format_clause
```

build_clause

```
BUILD { IMMEDIATE | DEFERRED }
```

by_users_with_roles

```
BY USERS WITH GRANTED ROLES role [, role]...
```

cache_clause

```
CACHE cache_specification [, cache_specification]...
```

cache_specification

```
MEASURE GROUP
{
  ALL
  | ( measure_name [, measure_name ]... ) [ levels_clause MATERIALIZED ]...
}
```

calc_meas_order_by_clause

```
calc_meas_expression [ { ASC | DESC } ] [ NULLS { FIRST | LAST } ]
```

calc_measure_clause

```
AS ( calc_meas_expression )
```

cancel_sql_clause

```
CANCEL SQL ' session_id , serial_number [ , @ instance_id ] [ , sql_id ] '
```

cell_assignment

```
measure_column [ { { condition
                    | expr
                    | single_column_for_loop
                  }
                [, { condition
                    | expr
                    | single_column_for_loop
                  }
                ]...
```

```
        | multi_column_for_loop
      }
    ]
```

Note: The outer square brackets are part of the syntax.
In this case, they do not indicate optionality.

cell_reference_options

```
[ { IGNORE | KEEP } NAV ]
[ UNIQUE { DIMENSION | SINGLE REFERENCE } ]
```

character_set_clause

```
CHARACTER SET character_set
```

check_datafiles_clause

```
CHECK DATAFILES [ GLOBAL | LOCAL ]
```

check_diskgroup_clause

```
CHECK [ REPAIR | NOREPAIR ]
```

checkpoint_clause

```
CHECKPOINT [ GLOBAL | LOCAL ]
```

classification_clause

```
[ CAPTION caption ]
[ DESCRIPTION description ]
[ CLASSIFICATION classification_name
  [ VALUE classification_value ]
  [ LANGUAGE language ]
]...
```

clause_options

```
OPTION
{ { = ( 'clause_option' | 'clause_option_pattern'
      [, 'clause_option' | 'clause_option_pattern' ]... ) }
  | { = ( 'clause_option' ) option_values }
  | { ALL [ EXCEPT = ( 'clause_option' | 'clause_option_pattern'
                      [, 'clause_option' | 'clause_option_pattern' ]... ) ] }
}
```

clear_free_space_clause

```
CLEAR FREE SPACE
```

close_keystore

```
SET KEYSTORE CLOSE
  [ IDENTIFIED BY { EXTERNAL STORE | keystore_password } ]
  [ CONTAINER = { ALL | CURRENT } ]
```

cluster_clause

```
BY [ LINEAR | INTERLEAVED ] ORDER clustering_columns
```

cluster_index_clause

```
CLUSTER [ schema. ] cluster_index_attributes
```

cluster_range_partitions

```
PARTITION BY RANGE (column[, column ]...)  
( PARTITION [ partition ]  
  range_values_clause table_partition_description  
  [, PARTITION [ partition ]  
    range_values_clause table_partition_description  
  ]...  
)
```

clustering_column_group

```
( column [, column ]... )
```

clustering_columns

```
clustering_column_group  
| ( clustering_column_group [, clustering_column_group ]... )
```

clustering_join

```
[ schema. ] table JOIN [ schema. ] table ON ( equijoin_condition )  
  [, JOIN [ schema. ] table ON ( equijoin_condition ) ]...
```

clustering_when

```
[ { YES | NO } ON LOAD ] [ { YES | NO } ON DATA MOVEMENT ]
```

coalesce_index_partition

```
COALESCE PARTITION [ parallel_clause ]
```

coalesce_table_partition

```
COALESCE PARTITION  
  [ update_index_clauses ]  
  [ parallel_clause ]  
  [ allow_disallow_clustering ]
```

coalesce_table_subpartition

```
COALESCE SUBPARTITION subpartition  
  [update_index_clauses]  
  [parallel_clause]  
  [allow_disallow_clustering]
```

column_association

```
COLUMNS [ schema. ]table.column  
  [, [ schema. ]table.column ]...  
  using_statistics_type
```

column_clauses

```
{ { add_column_clause  
  | modify_column_clauses  
  | drop_column_clause  
  | add_period_clause  
  | drop_period_clause  
  }...  
| rename_column_clause  
| { modify_collection_retrieval }...  
| { modify_LOB_storage_clause }...  
| { alter_varray_col_properties }...
```

```
}
```

column_definition

```
column [ datatype [ COLLATE column_collation_name ] ]  
  [ SORT ] [ VISIBLE | INVISIBLE ]  
  [ DEFAULT [ ON NULL ] expr | identity_clause ]  
  [ ENCRYPT encryption_spec ]  
  [ { inline_constraint }...  
    | inline_ref_constraint  
  ]
```

column_properties

```
{ object_type_col_properties  
| nested_table_col_properties  
| { varray_col_properties | LOB_storage_clause }  
  [ (LOB_partition_storage [, LOB_partition_storage ]...) ]  
| XMLType_column_properties  
}...
```

commit_switchover_clause

```
{ PREPARE | COMMIT } TO SWITCHOVER  
[ TO { { [ PHYSICAL | LOGICAL ] PRIMARY  
        | [ PHYSICAL ] STANDBY  
        } [ { WITH | WITHOUT } SESSION SHUTDOWN  
          { WAIT | NOWAIT }  
        ]  
        | LOGICAL STANDBY  
        }  
| CANCEL  
]
```

component_actions

```
ACTIONS COMPONENT =  
  { DATAPUMP | DIRECT_LOAD | OLS | XS } component_action [, component_action ]...  
  |  
  DV component_action ON object_name [, component_action ON object_name ]...
```

composite_hash_partitions

```
PARTITION BY HASH (column [, column ] ...)  
  { subpartition_by_range  
    | subpartition_by_list  
    | subpartition_by_hash  
  }  
  { individual_hash_partitions  
    | hash_partitions_by_quantity  
  }
```

composite_list_partitions

```
PARTITION BY LIST ( column [, column]... )  
[ AUTOMATIC [ STORE IN ( tablespace [, tablespace ]... ) ] ]  
  { subpartition_by_range  
    | subpartition_by_list  
    | subpartition_by_hash  
  }  
( list_partition_desc [, list_partition_desc]... )
```

composite_range_partitions

```
PARTITION BY RANGE ( column [, column]... )
```

```
[ INTERVAL ( expr ) [ STORE IN ( tablespace [, tablespace]... ) ] ]
{ subpartition_by_range
| subpartition_by_list
| subpartition_by_hash
}
( range_partition_desc [, range_partition_desc]... )
```

conditional_insert_clause

```
[ ALL | FIRST ]
WHEN condition
THEN insert_into_clause
  [ values_clause ]
  [ error_logging_clause ]
  [ insert_into_clause [ values_clause ] [ error_logging_clause ] ]...
[ WHEN condition
  THEN insert_into_clause
    [ values_clause ]
    [ error_logging_clause ]
    [ insert_into_clause [ values_clause ] [ error_logging_clause ] ]...
]...
[ ELSE insert_into_clause
  [ values_clause ]
  [ error_logging_clause ]
  [ insert_into_clause [ values_clause ] [ error_logging_clause ] ]...
]
```

consistent_hash_partitions

```
PARTITION BY CONSISTENT HASH (column [, column ]...)
  [ PARTITIONS AUTO ] TABLESPACE SET tablespace_set
```

consistent_hash_with_subpartitions

```
PARTITION BY CONSISTENT HASH (column [, column ]...)
  { subpartition_by_range
  | subpartition_by_list
  | subpartition_by_hash
  }
  [ PARTITIONS AUTO ]
```

constraint

```
{ inline_constraint
| out_of_line_constraint
| inline_ref_constraint
| out_of_line_ref_constraint
}
```

constraint_clauses

```
{ ADD { { out_of_line_constraint }...
  | out_of_line_REF_constraint
}
| MODIFY { CONSTRAINT constraint_name
  | PRIMARY KEY
  | UNIQUE (column [, column ]...)
} constraint_state [ CASCADE ]
| RENAME CONSTRAINT old_name TO new_name
| { drop_constraint_clause }...
}
```

constraint_state

```
[ [NOT] DEFERRABLE [INITIALLY {IMMEDIATE | DEFERRED}] ]
```

```
| INITIALLY { IMMEDIATE | DEFERRED } [ NOT ] [ DEFERRABLE ]  
]  
[ RELY | NORELY ]  
[ using_index_clause ]  
[ ENABLE | DISABLE ]  
[ VALIDATE | NOVALIDATE ]  
[ exceptions_clause
```

container_data_clause

```
{  
SET CONTAINER_DATA = { ALL | DEFAULT | ( container_name [, container_name ]... ) }  
|  
ADD CONTAINER_DATA = ( container_name [, container_name ]... )  
|  
REMOVE CONTAINER_DATA = ( container_name [, container_name ]... )  
}  
[ FOR [ schema. ] container_data_object ]
```

container_map_clause

```
CONTAINER_MAP UPDATE { add_table_partition | split_table_partition }
```

containers_clause

```
CONTAINERS( [schema.] { table | view } )
```

context_clause

```
[ WITH INDEX CONTEXT,  
SCAN CONTEXT implementation_type  
[ COMPUTE ANCILLARY DATA ]  
]  
[ WITH COLUMN CONTEXT ]
```

controlfile_clauses

```
CREATE { [ LOGICAL | PHYSICAL ] STANDBY | FAR SYNC INSTANCE }  
CONTROLFILE AS  
'filename' [ REUSE ]  
| BACKUP CONTROLFILE TO  
{ 'filename' [ REUSE ]  
| trace_file_clause  
}
```

convert_database_clause

```
CONVERT TO ( PHYSICAL | SNAPSHOT ) STANDBY
```

convert_redundancy_clause

```
CONVERT TO FLEX REDUNDANCY
```

cost_matrix_clause

```
COST  
{ MODEL [AUTO]  
| ( class_value [, class_value]... )  
VALUES ( ( cost_value [, cost_value]... )  
[ , (cost_value [, cost_value]... ) ]...  
)  
}
```

create_datafile_clause

```
CREATE DATAFILE
  { 'filename' | filenumber }
  [, 'filename' | filenumber ]...
}
[ AS { file_specification
      [, file_specification ]...
      | NEW
    }
]
```

create_file_dest_clause

```
CREATE_FILE_DEST = { NONE | 'directory_path_name' | diskgroup_name }
```

create_key

```
CREATE [ ENCRYPTION ] KEY { mkid:mk | mk }
  [ USING TAG 'tag' ]
  [ USING ALGORITHM 'encrypt_algorithm' ]
  [ FORCE KEYSTORE ]
  IDENTIFIED BY { EXTERNAL STORE | keystore_password }
  [ WITH BACKUP [ USING 'backup_identifier' ] ]
  [ CONTAINER = { ALL | CURRENT } ]
```

create_keystore

```
CREATE
  { KEYSTORE 'keystore_location'
  | [ LOCAL ] AUTO_LOGIN KEYSTORE FROM KEYSTORE 'keystore_location'
  }
  IDENTIFIED BY keystore_password
```

create_mv_refresh

```
{ REFRESH
  { { FAST | COMPLETE | FORCE }
  | { ON DEMAND
    | ON COMMIT
    | ON STATEMENT
  }
  | { START WITH date |
    NEXT date
  }...
  | WITH { PRIMARY KEY | ROWID }
  | USING
    { DEFAULT [ MASTER | LOCAL ] ROLLBACK SEGMENT
    | [ MASTER | LOCAL ] ROLLBACK SEGMENT rollback_segment
    }...
  | USING
    { ENFORCED | TRUSTED } CONSTRAINTS
  }...
| NEVER REFRESH
}
```

create_pdb_clone

```
{ { FROM { src_pdb_name [ @ dblink ] } | { NON$CDB @ dblink } }
|
  { AS PROXY FROM src_pdb_name @ dblink }
}
[ parallel_pdb_creation_clause ]
[ default_tablespaces ]
[ pdb_storage_clause ]
```

```

[ file_name_convert ]
[ service_name_convert ]
[ path_prefix_clause ]
[ tempfile_reuse_clause ]
[ SNAPSHOT COPY ]
[ user_tablespaces_clause ]
[ standbys_clause ]
[ logging_clause ]
[ create_file_dest_clause ]
[ keystore_clause ]
[ pdb_refresh_mode_clause ]
[ RELOCATE ]
[ NO DATA ]
[ HOST = 'hostname' ]
[ PORT = number ]

```

create_pdb_from_mirror_copy

```
new_pdb_name FROM base_pdb_name USING MIRROR COPY mirror_name
```

create_pdb_from_seed

```
ADMIN USER admin_user_name IDENTIFIED BY password
```

```

[ pdb_dba_roles ]
[ parallel_pdb_creation_clause ]
[ default_tablespace ]
[ pdb_storage_clause ]
[ file_name_convert ]
[ service_name_convert ]
[ path_prefix_clause ]
[ tempfile_reuse_clause ]
[ user_tablespaces_clause ]
[ standbys_clause ]
[ logging_clause ]
[ create_file_dest_clause ]
[ HOST = 'hostname' ]
[ PORT = number ]

```

create_pdb_from_xml

```

[ AS CLONE ] USING filename
[ source_file_name_convert | source_file_directory ]
[ { [ COPY | MOVE ] file_name_convert } | NOCOPY ]
[ service_name_convert ]
[ default_tablespace ]
[ pdb_storage_clause ]
[ path_prefix_clause ]
[ tempfile_reuse_clause ]
[ user_tablespaces_clause ]
[ standbys_clause ]
[ logging_clause ]
[ create_file_dest_clause ]
[ HOST = 'hostname' ]
[ PORT = number ]

```

create_zonemap_as_subquery

```

CREATE MATERIALIZED ZONEMAP
[ schema. ] zonemap_name
[ zonemap_attributes ]
[ zonemap_refresh_clause ]
[ { ENABLE | DISABLE } PRUNING ]
AS query_block

```

create_zonemap_on_table


```
CREATE MATERIALIZED ZONEMAP
  [ schema. ] zonemap_name
  [ zonemap_attributes ]
  [ zonemap_refresh_clause ]
  [ { ENABLE | DISABLE } PRUNING ]
  ON [ schema. ] { table | materialized_view } ( column [, column]... )
```

cross_outer_apply_clause

```
{ CROSS | OUTER } APPLY { table_reference | collection_expression }
```

cycle_clause

```
{CYCLE c_alias [, c_alias]...
  SET cycle_mark_c_alias TO cycle_value
  DEFAULT no_cycle_value
}
```

database_file_clauses

```
{ RENAME FILE 'filename' [, 'filename' ]...
  TO 'filename'
| create_datafile_clause
| alter_datafile_clause
| alter_tempfile_clause
| move_datafile_clause
}
```

database_logging_clauses

```
{ LOGFILE
  [ GROUP integer ] file_specification
  [, [ GROUP integer ] file_specification ]...
| MAXLOGFILES integer
| MAXLOGMEMBERS integer
| MAXLOGHISTORY integer
| { ARCHIVELOG | NOARCHIVELOG }
| FORCE LOGGING
| SET STANDBY NOLOGGING FOR {DATA AVAILABILITY | LOAD PERFORMANCE}
}
```

datafile_tempfile_clauses

```
{ ADD { DATAFILE | TEMPFILE }
  [ file_specification [, file_specification ]... ]
| DROP {DATAFILE | TEMPFILE} { 'filename' | file_number }
| SHRINK TEMPFILE { 'filename' | file_number } [KEEP size_clause]
| RENAME DATAFILE 'filename' [, 'filename' ]...
  TO 'filename' [, 'filename' ]...
| { DATAFILE | TEMPFILE } { ONLINE | OFFLINE }
}
```

datafile_tempfile_spec

```
[ 'filename' | 'ASM_filename' ]
[ SIZE size_clause ]
[ REUSE ]
[ autoextend_clause ]
```

db_user_proxy_clauses

```
[ WITH
  { ROLE { role_name [, role_name]...
    | ALL EXCEPT role_name [, role_name]...
  }
}
```

```
| NO ROLES
}
]
[ AUTHENTICATION REQUIRED ]
```

dblink

```
database[.domain [.domain ]... ] [ @ connection_qualifier ]
```

dblink_authentication

```
AUTHENTICATED BY user IDENTIFIED BY password
```

deallocate_unused_clause

```
DEALLOCATE UNUSED [ KEEP size_clause ]
```

default_aggregate_clause

```
DEFAULT AGGREGATE BY aggr_function
```

default_cost_clause

```
DEFAULT COST (cpu_cost, io_cost, network_cost)
```

default_index_compression

```
INDEX { COMPRESS ADVANCED { LOW | HIGH }
      | NOCOMPRESS
      }
```

default_measure_clause

```
DEFAULT MEASURE measure
```

default_selectivity_clause

```
DEFAULT SELECTIVITY default_selectivity
```

default_settings_clauses

```
{ DEFAULT EDITION = edition_name
| SET DEFAULT { BIGFILE | SMALLFILE } TABLESPACE
| DEFAULT TABLESPACE tablespace
| DEFAULT [ LOCAL ] TEMPORARY TABLESPACE { tablespace | tablespace_group_name }
| RENAME GLOBAL_NAME TO database.domain [.domain ]...
| ENABLE BLOCK CHANGE TRACKING [ USING FILE 'filename' [ REUSE ] ]
| DISABLE BLOCK CHANGE TRACKING
| [NO] FORCE FULL DATABASE CACHING
| CONTAINERS DEFAULT TARGET = { (container_name) | NONE }
| flashback_mode_clause
| undo_mode_clause
| set_time_zone_clause
}
```

default_table_compression

```
TABLE { COMPRESS FOR OLTP
      | COMPRESS FOR QUERY { LOW | HIGH }
      | COMPRESS FOR ARCHIVE { LOW | HIGH }
      | NOCOMPRESS
      }
```

default_tablespace

```
DEFAULT TABLESPACE tablespace  
[ DATAFILE datafile_tempfile_spec ]  
[ extent_management_clause ]
```

default_tablespace_params

```
DEFAULT [ default_table_compression ] [ default_index_compression ]  
[ inmemory_clause ] [ ilm_clause ] [ storage_clause ]
```

default_temp_tablespace

```
[ BIGFILE | SMALLFILE ] DEFAULT  
{ { TEMPORARY TABLESPACE }  
| { LOCAL TEMPORARY TABLESPACE FOR { ALL | LEAF } }  
} tablespace  
[ TEMPFILE file_specification [, file_specification ]...]  
[ extent_management_clause ]
```

deferred_segment_creation

```
SEGMENT CREATION { IMMEDIATE | DEFERRED }
```

delete_secret

```
DELETE SECRET FOR CLIENT 'client_identifier'  
[ FORCE KEYSTORE ]  
IDENTIFIED BY { EXTERNAL STORE | keystore_password }  
[ WITH BACKUP [ USING 'backup_identifier' ] ]
```

delete_secret_seps

```
DELETE SECRET secret FOR CLIENT 'client_identifier'  
FROM [ LOCAL ] AUTO_LOGIN KEYSTORE directory
```

dependent_tables_clause

```
DEPENDENT TABLES  
( table ( partition_spec [, partition_spec]...  
[ , table ( partition_spec [, partition_spec]... ]  
)  
)
```

dim_by_clause

```
DIMENSION BY ( dim_key [, dim_key ]... )
```

dim_key

```
dim_ref  
[classification_clause]...  
KEY  
{[(] [alias.] fact_column [)]  
|  
( [alias.] fact_column [, [alias.] fact_column]... )  
}  
REFERENCES  
{[(] attribute [)]  
|  
( attribute [, attribute]... )  
}  
HIERARCHIES ( hier_ref [, hier_ref]... )
```

dim_order_clause

```
attribute [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
```

dim_ref

```
[ schema. ] attr_dim_name [ [AS] dim_alias ]
```

dimension_join_clause

```
{ JOIN KEY  
  { child_key_column  
    | (child_key_column [, child_key_column ]...)  
  }  
  REFERENCES parent_level  
}...
```

disk_offline_clause

```
OFFLINE  
  { [ QUORUM | REGULAR ] DISK disk_name [, disk_name ]...  
    | DISKS IN [ QUORUM | REGULAR ] FAILGROUP failgroup_name [, failgroup_name ]...  
  }... [ timeout_clause ]
```

disk_online_clause

```
ONLINE  
  { { [ QUORUM | REGULAR ] DISK disk_name [, disk_name ]...  
      | DISKS IN [ QUORUM | REGULAR ] FAILGROUP failgroup_name [, failgroup_name ]...  
    }...  
    | ALL  
  } [ POWER integer ] [ WAIT | NOWAIT ]
```

disk_region_clause

```
[ HOT | COLD ] [ MIRRORHOT | MIRRORCOLD ]
```

diskgroup_alias_clauses

```
{ ADD ALIAS  
  'alias_name' FOR 'filename'  
  [, 'alias_name' FOR 'filename' ]...  
| DROP ALIAS 'alias_name' [, 'alias_name' ]...  
| RENAME ALIAS  
  'old_alias_name' TO 'new_alias_name'  
  [, 'old_alias_name' TO 'new_alias_name' ]...  
}
```

diskgroup_attributes

```
SET ATTRIBUTE 'attribute_name' = 'attribute_value'
```

diskgroup_availability

```
{ MOUNT [ RESTRICTED | NORMAL ]  
      [ FORCE | NOFORCE ]  
| DISMOUNT [ FORCE | NOFORCE ]  
}
```

diskgroup_directory_clauses

```
{ ADD DIRECTORY 'filename' [, 'filename' ]...  
| DROP DIRECTORY  
  'filename' [ FORCE | NOFORCE ]
```

```

    [, 'filename' [ FORCE | NOFORCE ] ]...
| RENAME DIRECTORY
    'old_dir_name' TO 'new_dir_name'
    [, 'old_dir_name' TO 'new_dir_name' ]...
}

```

diskgroup_template_clauses

```

{ { ADD | MODIFY } TEMPLATE template_name qualified_template_clause
    [, template_name qualified_template_clause ]...
| DROP TEMPLATE template_name [, template_name ]...
}

```

diskgroup_volume_clauses

```

{ add_volume_clause
| modify_volume_clause
| RESIZE VOLUME asm_volume SIZE size_clause
| DROP VOLUME asm_volume
}

```

distributed_recov_clauses

```

{ ENABLE | DISABLE } DISTRIBUTED RECOVERY

```

dml_table_expression_clause

```

{ [ schema. ]
  { table
    [ partition_extension_clause
      | @ dblink
    ]
  | { view | materialized view } [ @ dblink ]
  }
| ( subquery [ subquery_restriction_clause ] )
| table_collection_expression
}

```

domain_index_clause

```

indextype
  [ local_domain_index_clause ]
  [ parallel_clause ]
  [ PARAMETERS ('ODCI_parameters') ]

```

drop_binding_clause

```

DROP BINDING (parameter_type [, parameter_type ]...)
  [ FORCE ]

```

drop_column_clause

```

{ SET UNUSED { COLUMN column
                | (column [, column ]...)
              }
  [ { CASCADE CONSTRAINTS | INVALIDATE }... ]
  [ ONLINE ]
| DROP { COLUMN column
        | (column [, column ]...)
      }
  [ { CASCADE CONSTRAINTS | INVALIDATE }... ]
  [ CHECKPOINT [ integer ] ]
| DROP { UNUSED COLUMNS
        | COLUMNS CONTINUE
      }
}

```

```
[ CHECKPOINT [ integer ] ]  
}
```

drop_constraint_clause

```
DROP (  
    ( PRIMARY KEY  
      | UNIQUE "(" (column)... ")"  
      | CONSTRAINT constraint_name )  
      [ CASCADE ] [( KEEP | DROP ) INDEX ]  
    )  
    [ ONLINE ]
```

drop_disk_clause

```
DROP  
{ [ QUORUM | REGULAR ] DISK  
  disk_name [ FORCE | NOFORCE ]  
  [, disk_name [ FORCE | NOFORCE ] ]...  
| DISKS IN [ QUORUM | REGULAR ] FAILGROUP  
  failgroup_name [ FORCE | NOFORCE ]  
  [, failgroup_name [ FORCE | NOFORCE ] ]...  
}
```

drop_diskgroup_file_clause

```
DROP FILE 'filename' [, 'filename' ]...
```

drop_external_partition_attrs

```
DROP EXTERNAL PARTITION ATTRIBUTES
```

drop_filegroup_clause

```
DROP FILEGROUP filegroup_name [ CASCADE ]
```

drop_index_partition

```
DROP PARTITION partition_name
```

drop_logfile_clauses

```
DROP [ STANDBY ] LOGFILE  
  { logfile_descriptor  
    [, logfile_descriptor ]...  
  | MEMBER 'filename'  
    [, 'filename' ]...  
  }
```

drop_mirror_copy

```
DROP MIRROR COPY mirror_name
```

drop_period_clause

```
DROP ( PERIOD FOR valid_time_column )
```

drop_table_partition

```
DROP partition_extended_names  
  [ update_index_clauses [ parallel_clause ] ]
```

drop_table_subpartition

```
DROP subpartition_extended_names  
[ update_index_clauses [ parallel_clause ] ]
```

ds_iso_format

```
[ - ] P [ days D ]  
[ T [ hours H ] [ minutes M ] [ seconds [ . frac_secs ] S ] ]
```

dynamic_base_profile

```
INCLUDING base_profile
```

else_clause

```
ELSE else_expr
```

enable_disable_clause

```
{ ENABLE | DISABLE }  
[ VALIDATE | NOVALIDATE ]  
{ UNIQUE (column [, column ]...)  
| PRIMARY KEY  
| CONSTRAINT constraint_name  
}  
[ using_index_clause ]  
[ exceptions_clause ]  
[ CASCADE ]  
[ { KEEP | DROP } INDEX ]
```

enable_disable_volume

```
{ ENABLE | DISABLE } VOLUME  
{ asm_volume [, asm_volume]...  
| ALL  
}
```

enable_pluggable_database

```
ENABLE PLUGGABLE DATABASE  
[ SEED  
[ file_name_convert ]  
[ SYSTEM tablespace_datafile_clauses ]  
[ SYSAUX tablespace_datafile_clauses ]  
]  
[ undo_mode_clause ]
```

encryption_spec

```
[ USING 'encrypt_algorithm' ]  
[ IDENTIFIED BY password ]  
[ 'integrity_algorithm' ]  
[ [ NO ] SALT ]
```

end_session_clauses

```
{ DISCONNECT SESSION 'session_id, serial_number'  
[ POST_TRANSACTION ]  
| KILL SESSION 'session_id, serial_number [, @ instance_id ]'  
}  
[ IMMEDIATE | NOREPLAY ]
```

entry

```
( regular_entry [ format_clause ] ) | wildcard
```

error_logging_clause

```
LOG ERRORS  
[ INTO [schema.] table ]  
[ (simple_expression) ]  
[ REJECT LIMIT { integer | UNLIMITED } ]
```

evaluation_edition_clause

```
EVALUATE USING { CURRENT EDITION | EDITION edition | NULL EDITION }
```

exceptions_clause

```
EXCEPTIONS INTO [ schema. ] table
```

exchange_partition_subpart

```
EXCHANGE { partition_extended_name  
          | subpartition_extended_name  
          }  
WITH TABLE [ schema. ] table  
[ { INCLUDING | EXCLUDING } INDEXES ]  
[ { WITH | WITHOUT } VALIDATION ]  
[ exceptions_clause ]  
[ update_index_clauses [ parallel_clause ] ]  
[ CASCADE ]
```

export_keys

```
EXPORT [ ENCRYPTION ] KEYS WITH SECRET secret  
TO 'filename'  
[ FORCE KEYSTORE ]  
IDENTIFIED BY keystore_password  
[ WITH IDENTIFIER IN { 'key_id' [, 'key_id' ]... | ( subquery ) } ]
```

expr

```
{ simple_expression  
| compound_expression  
| calc_meas_expression  
| case_expression  
| cursor_expression  
| datetime_expression  
| function_expression  
| interval_expression  
| JSON_object_access_expr  
| model_expression  
| object_access_expression  
| scalar_subquery_expression  
| type_constructor_expression  
| variable_expression  
}
```

expression_list

```
{ expr [, expr ]...  
| ( [expr [, expr ]] ... )  
}
```

extended_attribute_clause


```

ATTRIBUTE attribute
{ LEVEL level
  DETERMINES { dependent_column
              | (dependent_column [, dependent_column ]... )
              }
}...

```

extent_management_clause

```

EXTENT MANAGEMENT LOCAL
[ AUTOALLOCATE
  | UNIFORM [ SIZE size_clause ]
]

```

external_part_subpart_data_props

```

[ DEFAULT DIRECTORY directory ]
[ LOCATION
  ([ directory: ] 'location_specifier'
    [, [ directory: ] 'location_specifier' ]...
  )
]

```

external_table_clause

```

([ TYPE access_driver_type ]
 [ external_table_data_props ]
)
[ REJECT LIMIT { integer | UNLIMITED } ]
[ inmemory_table_clause ]

```

external_table_data_props

```

[ DEFAULT DIRECTORY directory ]
[ ACCESS PARAMETERS
  { ('opaque_format_spec')
    | ( opaque_format_spec )
    | USING CLOB subquery
  }
]
[ LOCATION
  ([ directory: ] 'location_specifier'
    [, [ directory: ] 'location_specifier' ]...
  )
]

```

failover_clause

```

FAILOVER TO target_db_name [ FORCE ]

```

file_name_convert

```

FILE_NAME_CONVERT =
{ ( 'filename_pattern', 'replacement_filename_pattern'
  [, 'filename_pattern', 'replacement_filename_pattern' ]... )
  | NONE
}

```

file_owner_clause

```

SET OWNERSHIP { OWNER = 'user' | GROUP = 'usergroup'
               [, OWNER = 'user' | GROUP = 'usergroup' ]...
               } FOR FILE 'filename' [, 'filename']...

```

file_permissions_clause

```
SET PERMISSION { OWNER | GROUP | OTHER }  
= { NONE | READ ONLY | READ WRITE }  
[, { OWNER | GROUP | OTHER | ALL }  
= { NONE | READ ONLY | READ WRITE } ]...  
FOR FILE 'filename' [, 'filename']...
```

file_specification

```
{ datafile_tempfile_spec  
| redo_log_file_spec  
}
```

filegroup_clauses

```
{ add_filegroup_clause  
| modify_filegroup_clause  
| move_to_filegroup_clause  
| drop_filegroup_clause  
}
```

filter_condition

```
INCLUDING ROWS where_clause
```

flashback_archive_clause

```
FLASHBACK ARCHIVE [flashback_archive] | NO FLASHBACK ARCHIVE
```

flashback_archive_quota

```
QUOTA integer { M | G | T | P | E }
```

flashback_archive_retention

```
RETENTION integer {YEAR | MONTH | DAY}
```

flashback_mode_clause

```
FLASHBACK { ON | OFF }
```

flashback_query_clause

```
{ VERSIONS BETWEEN { SCN | TIMESTAMP }  
  { expr | MINVALUE } AND { expr | MAXVALUE }  
| VERSIONS PERIOD FOR valid_time_column BETWEEN  
  { expr | MINVALUE } AND { expr | MAXVALUE }  
| AS OF { SCN | TIMESTAMP } expr  
| AS OF PERIOD FOR valid_time_column expr  
}
```

following_boundary

```
{ CURRENT MEMBER | offset_expr FOLLOWING }  
AND  
{ offset_expr FOLLOWING | UNBOUNDED FOLLOWING }
```

for_refresh_clause

```
{ FOR SYNCHRONOUS REFRESH USING staging_log_name  
| FOR FAST REFRESH  
}
```

for_update_clause

```
FOR UPDATE
  [ OF [ [ schema. ] { table | view } . ] column
        [, [ [ schema. ] { table | view } . ] column
          ]...
    ]
  [ { NOWAIT | WAIT integer
    | SKIP LOCKED
    }
  ]
```

format_clause

```
FORMAT JSON
```

full_database_recovery

```
[ STANDBY ] DATABASE
[ { UNTIL { CANCEL
          | TIME date
          | CHANGE integer
          | CONSISTENT
          }
  | USING BACKUP CONTROLFILE
  | SNAPSHOT TIME date
  }...
]
```

fully_qualified_file_name

```
+diskgroup_name/db_name/file_type/
  file_type_tag.filenumber.incarnation_number
```

function_association

```
{ FUNCTIONS
  [ schema. ]function [, [ schema. ]function ]...
| PACKAGES
  [ schema. ]package [, [ schema. ]package ]...
| TYPES
  [ schema. ]type [, [ schema. ]type ]...
| INDEXES
  [ schema. ]index [, [ schema. ]index ]...
| INDEXTYPES
  [ schema. ]indextype [, [ schema. ]indextype ]...
}
{ using_statistics_type
| { default_cost_clause [, default_selectivity_clause ]
  | default_selectivity_clause [, default_cost_clause ]
  }
}
```

general_recovery

```
RECOVER
[ AUTOMATIC ]
[ FROM 'location' ]
{ { full_database_recovery
  | partial_database_recovery
  | LOGFILE 'filename'
  }
[ { TEST
  | ALLOW integer CORRUPTION
  | parallel_clause
```

```

    }...
  ]
| CONTINUE [ DEFAULT ]
| CANCEL
}

```

global_partitioned_index

```

GLOBAL PARTITION BY
  { RANGE (column_list)
    (index_partitioning_clause)
  | HASH (column_list)
    { individual_hash_partitions
      | hash_partitions_by_quantity
    }
  }

```

grant_object_privileges

```

{ object_privilege | ALL [ PRIVILEGES ] }
[ (column [, column ]...) ]
[, { object_privilege | ALL [ PRIVILEGES ] }
  [ (column [, column ]...) ]
]...
on_object_clause
TO grantee_clause
  [ WITH HIERARCHY OPTION ]
  [ WITH GRANT OPTION ]

```

grant_roles_to_programs

```

role [, role ]... TO program_unit [, program_unit ]...

```

grant_system_privileges

```

{ system_privilege | role | ALL PRIVILEGES }
[, { system_privilege | role | ALL PRIVILEGES } ]...
TO { grantee_clause | grantee_identified_by } [ WITH { ADMIN | DELEGATE } OPTION ]

```

grantee_clause

```

{ user | role | PUBLIC }
[, { user | role | PUBLIC } ]...

```

grantee_identified_by

```

user [, user ]... IDENTIFIED BY password [, password ]...

```

group_by_clause

```

GROUP BY
  { expr
  | rollup_cube_clause
  | grouping_sets_clause
  }
  [, { expr
      | rollup_cube_clause
      | grouping_sets_clause
    }
  ]...
  [ HAVING condition ]

```

grouping_expression_list

```
expression_list [, expression_list ]...
```

grouping_sets_clause

```
GROUPING SETS  
( { rollup_cube_clause | grouping_expression_list } )
```

hash_partitions

```
PARTITION BY HASH (column [, column ] ...)  
{ individual_hash_partitions  
| hash_partitions_by_quantity  
}
```

hash_partitions_by_quantity

```
PARTITIONS hash_partition_quantity  
[ STORE IN (tablespace [, tablespace ]...) ]  
[ table_compression | index_compression ]  
[ OVERFLOW STORE IN (tablespace [, tablespace ]...) ]
```

hash_subparts_by_quantity

```
SUBPARTITIONS integer [STORE IN ( tablespace [, tablespace]... )]
```

heap_org_table_clause

```
[ table_compression ] [ inmemory_table_clause ] [ ilm_clause ]
```

hier_ancestor_expression

```
HIER_ANCESTOR ( member_expression AT  
                { LEVEL level_ref  
                  | DEPTH depth_expression  
                }  
              )
```

hier_attr_clause

```
hier_attr_name [ classification_clause ]...
```

hier_attr_name

```
{ MEMBER_NAME  
| MEMBER_UNIQUE_NAME  
| MEMBER_CAPTION  
| MEMBER_DESCRIPTION  
| LEVEL_NAME  
| HIER_ORDER  
| DEPTH  
| IS_LEAF  
| PARENT_LEVEL_NAME  
| PARENT_UNIQUE_NAME  
}
```

hier_attrs_clause

```
HIERARCHICAL ATTRIBUTES ( hier_attr_clause [, hier_attr_clause ]... )
```

hier_lead_lag_clause

```
member_expression OFFSET offset_expr  
[ WITHIN  
  { { LEVEL | PARENT }
```

```
    | ACROSS ANCESTOR AT LEVEL level_ref [ POSITION FROM { BEGINNING | END } ]  
  }  
]
```

hier_lead_lag_expression

```
{ HIER_LEAD | HIER_LAG } ( hier_lead_lag_clause )
```

hier_navigation_expression

```
{  
  hier_ancestor_expression  
  | hier_parent_expression  
  | hier_lead_lag_expression  
}
```

hier_parent_expression

```
HIER_PARENT ( member_expression )
```

hier_ref

```
[ schema. ] hier_name [ [ AS ] hier_alias ] [ DEFAULT ]
```

hier_using_clause

```
USING [ schema. ] attribute_dimension level_hier_clause
```

hierarchical_query_clause

```
{ CONNECT BY [ NOCYCLE ] condition [ START WITH condition ]  
  | START WITH condition CONNECT BY [ NOCYCLE ] condition  
}
```

hierarchy_clause

```
HIERARCHY hierarchy  
(child_level { CHILD OF parent_level }...  
  [ dimension_join_clause ]  
)
```

hierarchy_ref

```
[ attr_dim_alias. ] hier_alias
```

identity_clause

```
GENERATED  
[ ALWAYS | BY DEFAULT [ ON NULL ] ]  
AS IDENTITY [ ( identity_options ) ]
```

identity_options

```
{ START WITH ( integer | LIMIT VALUE )  
  | INCREMENT BY integer  
  | ( MAXVALUE integer | NOMAXVALUE )  
  | ( MINVALUE integer | NOMINVALUE )  
  | ( CYCLE | NOCYCLE )  
  | ( CACHE integer | NOCACHE )  
  | ( ORDER | NOORDER ) }...
```

ilm_clause

```
ILM
{ ADD POLICY ilm_policy_clause
| { DELETE | ENABLE | DISABLE } POLICY ilm_policy_name
| DELETE_ALL | ENABLE_ALL | DISABLE_ALL
}
```

ilm_compression_policy

```
{ table_compression { SEGMENT | GROUP }
  { { AFTER ilm_time_period OF { { NO ACCESS } | { NO MODIFICATION } | CREATION } }
  | { ON function_name } }
}
|
{ { ROW STORE COMPRESS ADVANCED
  | COLUMN STORE COMPRESS FOR QUERY
  }
  ROW AFTER ilm_time_period OF NO MODIFICATION
}
```

ilm_inmemory_policy

```
{ SET INMEMORY [ inmemory_attributes ]
| MODIFY INMEMORY inmemory_memcompress
| NO INMEMORY
}
[ SEGMENT ]
{ AFTER ilm_time_period OF { NO ACCESS | NO MODIFICATION | CREATION }
  | ON function_name
}
```

ilm_policy_clause

```
{ ilm_compression_policy | ilm_tiering_policy | ilm_inmemory_policy }
```

ilm_tiering_policy

```
{ TIER TO tablespace [ SEGMENT | GROUP ] [ ON function_name ] }
|
{ TIER TO tablespace READ ONLY [ SEGMENT | GROUP ]
  { { AFTER ilm_time_period OF { { NO ACCESS } | { NO MODIFICATION } | CREATION } }
  | { ON function_name } } }
```

ilm_time_period

```
integer { { DAY | DAYS } | { MONTH | MONTHS } | { YEAR | YEARS } }
```

immutable_table_clauses

```
immutable_table_no_drop_clause immutable_table_no_delete_clause
```

immutable_table_no_delete_clause

```
NO DELETE ( [ LOCKED ] | ( UNTIL integer DAYS AFTER INSERT [ LOCKED ] ) )
```

immutable_table_no_drop_clause

```
NO DROP [ UNTIL integer DAYS IDLE ]
```

implementation_clause

```
{ ANCILLARY TO primary_operator
  ( parameter_type [, parameter_type ]...)
  [, primary_operator
  ( parameter_type [, parameter_type ]...)
```

```
]...  
| context_clause  
}
```

import_keys

```
IMPORT [ ENCRYPTION ] KEYS WITH SECRET secret  
FROM 'filename'  
[ FORCE KEYSTORE ]  
IDENTIFIED BY keystore_password  
[ WITH BACKUP [ USING 'backup_identifler' ] ]
```

incomplete_file_name

```
+diskgroup_name [ (template_name) ]
```

index_attributes

```
[ { physical_attributes_clause  
| logging_clause  
| ONLINE  
| TABLESPACE { tablespace | DEFAULT }  
| index_compression  
| { SORT | NOSORT }  
| REVERSE  
| VISIBLE | INVISIBLE  
| partial_index_clause  
| parallel_clause  
}...  
]
```

index_compression

```
{ prefix_compression  
| advanced_index_compression  
}
```

index_expr

```
{ column | column_expression }
```

index_org_overflow_clause

```
[ INCLUDING column_name ]  
OVERFLOW [ segment_attributes_clause ]
```

index_org_table_clause

```
[ { mapping_table_clause  
| PCTTHRESHOLD integer  
| prefix_compression  
}...  
]  
[ index_org_overflow_clause ]
```

index_partition_description

```
PARTITION  
[ partition  
| { segment_attributes_clause  
| index_compression  
}...  
| PARAMETERS ( 'ODCI_parameters' )  
]
```



```
[ USABLE | UNUSABLE ]  
]
```

index_partitioning_clause

```
PARTITION [ partition ]  
VALUES LESS THAN (literal[, literal]... )  
[ segment_attributes_clause ]
```

index_properties

```
[ { { global_partitioned_index  
    | local_partitioned_index  
    }  
  | index_attributes  
  }...  
| INDEXTYPE IS { domain_index_clause  
                | XMLIndex_clause  
                }  
]
```

index_subpartition_clause

```
{ STORE IN (tablespace[, tablespace ]...)  
| (SUBPARTITION  
    [ subpartition ][ TABLESPACE tablespace ] [ index_compression ] [ USABLE |  
UNUSABLE ]  
    [, SUBPARTITION  
    [ subpartition ][ TABLESPACE tablespace ] [ index_compression ] [ USABLE |  
UNUSABLE ]  
    ]...  
    )  
}
```

indexing_clause

```
INDEXING { ON | OFF }
```

individual_hash_partitions

```
( PARTITION [partition] [read_only_clause] [indexing_clause]  
[partitioning_storage_clause]  
[, PARTITION [partition] [read_only_clause] [indexing_clause]  
[partitioning_storage_clause]]... )
```

individual_hash_subparts

```
SUBPARTITION [subpartition] [read_only_clause] [indexing_clause]  
[partitioning_storage_clause]
```

inline_constraint

```
[ CONSTRAINT constraint_name ]  
{ [ NOT ] NULL  
| UNIQUE  
| PRIMARY KEY  
| references_clause  
| CHECK (condition)  
}  
[ constraint_state ]
```

inline_external_table

```
EXTERNAL '(' '(' column_definition ',' ')' inline_external_table_properties ')'
```

inline_external_table_properties

```
TYPE [ access_driver_type ] external_table_data_props  
[ REJECT LIMIT { integer | UNLIMITED }
```

inline_ref_constraint

```
{ SCOPE IS [ schema. ] scope_table  
| WITH ROWID  
| [ CONSTRAINT constraint_name ]  
  references_clause  
| [ constraint_state ]  
}
```

inmemory_attributes

```
[ inmemory_memcompress ] [ inmemory_priority ] [ inmemory_distribute ]  
[ inmemory_duplicate ]
```

inmemory_clause

```
INMEMORY [ inmemory_attributes ]  
| NO INMEMORY
```

inmemory_column_clause

```
{ INMEMORY [ inmemory_memcompress ] | NO INMEMORY } ( column [, column ]... )  
[ { INMEMORY [ inmemory_memcompress ] | NO INMEMORY } ( column [, column ]... ) ]...
```

inmemory_distribute

```
DISTRIBUTE [ AUTO | BY { ROWID RANGE | PARTITION | SUBPARTITION } ]  
[ FOR SERVICE { DEFAULT | ALL | service_name | NONE } ]
```

inmemory_duplicate

```
DUPLICATE | DUPLICATE ALL | NO DUPLICATE
```

inmemory_memcompress

```
MEMCOMPRESS FOR { DML | QUERY [ LOW | HIGH ] | CAPACITY [ LOW | HIGH ] }  
| NO MEMCOMPRESS
```

inmemory_priority

```
PRIORITY { NONE | LOW | MEDIUM | HIGH | CRITICAL }
```

inmemory_table_clause

```
[ { INMEMORY [ inmemory_attributes ] } | { NO INMEMORY } ]  
[ inmemory_column_clause ]
```

inner_cross_join_clause

```
{ [ INNER ] JOIN table_reference  
  { ON condition  
  | USING (column [, column ]...)  
  }  
| { CROSS  
  | NATURAL [ INNER ]  
  }  
  JOIN table_reference  
}
```

insert_into_clause

```
INTO dml_table_expression_clause [ t_alias ]  
[ (column [, column ]...) ]
```

instance_clauses

```
{ ENABLE | DISABLE } INSTANCE 'instance_name'
```

instances_clause

```
INSTANCES = { ( 'instance_name' [, 'instance_name' ]... )  
              | ALL [ EXCEPT ( 'instance_name' [, 'instance_name' ]... ) ] }
```

integer

```
[ + | - ] digit [ digit ]...
```

interval_day_to_second

```
INTERVAL '{ integer | integer time_expr | time_expr }'  
{ { DAY | HOUR | MINUTE } [ (leading_precision) ]  
  | SECOND [ (leading_precision [, fractional_seconds_precision) ] ]  
  }  
[ TO { DAY | HOUR | MINUTE | SECOND [ (fractional_seconds_precision) ] } ]
```

interval_year_to_month

```
INTERVAL 'integer [- integer ]'  
{ YEAR | MONTH } [ (precision) ] [ TO { YEAR | MONTH } ]
```

into_clause

```
INTO [ schema. ] table
```

invoker_rights_clause

```
AUTHID { CURRENT_USER | DEFINER }
```

isolate_keystore

```
ISOLATE KEYSTORE IDENTIFIED BY isolated_keystore_password  
FROM ROOT KEYSTORE [ FORCE KEYSTORE ]  
IDENTIFIED BY { EXTERNAL STORE | united_keystore_password }  
[ WITH BACKUP [ USING 'backup_identifier' ] ]
```

join_clause

```
table_reference  
{ inner_cross_join_clause | outer_join_clause | cross_outer_apply_clause }...
```

JSON_agg_returning_clause

```
RETURNING { VARCHAR2 [ ( size [BYTE | CHAR] ) ]  
          | CLOB  
          | BLOB }
```

JSON_ARRAY_content

```
( , [ JSON_ARRAY_element ] ... )  
[ JSON_on_null_clause ] [ JSON_returning_clause ]  
[ STRICT ]
```

JSON_ARRAY_element

expr [format_clause]

JSON_column_definition

JSON_exists_column
| JSON_query_column
| JSON_value_column
| JSON_nested_path
| ordinality_column

JSON_columns_clause

COLUMNS (JSON_column_definition [, JSON_column_definition]...)

JSON_exists_column

column_name [JSON_value_return_type]
EXISTS [PATH] [JSON_path] [JSON_exists_on_error_clause]
[JSON_exists_on_empty_clause]

JSON_exists_on_empty_clause

{ ERROR | TRUE | FALSE } ON EMPTY

JSON_exists_on_error_clause

{ ERROR | TRUE | FALSE } ON ERROR

JSON_nested_path

NESTED [PATH] JSON_path JSON_columns_clause

JSON_object_content

("*" | [entry] ...)
[JSON_on_null_clause] [JSON_returning_clause]
[STRICT]
[WITH UNIQUE KEYS]

JSON_on_null_clause

{ NULL | ABSENT } ON NULL

JSON_on_null_clause

{ NULL | ABSENT } ON NULL

JSON_passing_clause

PASSING expr AS identifier [, expr AS identifier]...

JSON_path

JSON_basic_path_expression | JSON_relative_object_access

JSON_query_column

column_name JSON_query_return_type { TRUNCATE }
FORMAT JSON [JSON_query_wrapper_clause]

```
PATH JSON_basic_path_expression [ JSON_query_on_error_clause ]  
[ JSON_query_on_empty_clause ]
```

JSON_query_on_empty_clause

```
{ ERROR  
| NULL  
| EMPTY  
| EMPTY ARRAY  
| EMPTY OBJECT  
} ON EMPTY
```

JSON_query_on_error_clause

```
{ ERROR  
| NULL  
| EMPTY  
| EMPTY ARRAY  
| EMPTY OBJECT  
} ON ERROR
```

JSON_query_return_type

```
VARCHAR2 [ ( size [BYTE | CHAR] [ TRUNCATE ] ) ]  
| CLOB  
| BLOB
```

JSON_query_returning_clause

```
[ RETURNING JSON_query_return_type ] [ PRETTY ] [ ASCII ]
```

JSON_query_wrapper_clause

```
WITHOUT [ ARRAY ] WRAPPER  
| WITH [ UNCONDITIONAL | CONDITIONAL ] [ ARRAY ] WRAPPER
```

JSON_relative_object_access

```
JSON_object_key [ array_step ]  
( "." JSON_object_key [ array_step ] )...
```

JSON_returning_clause

```
RETURNING VARCHAR2 [ ( size [BYTE | CHAR] ) ] | CLOB | BLOB
```

JSON_table_on_empty_clause

```
{ ERROR | NULL | DEFAULT literal } ON EMPTY
```

JSON_table_on_error_clause

```
{ ERROR | NULL | DEFAULT literal } ON ERROR
```

JSON_value_column

```
column_name [ JSON_value_return_type ] [ TRUNCATE ]  
[ PATH ] [ JSON_path ] [ JSON_value_on_error_clause ]  
[ JSON_value_on_empty_clause ]
```

JSON_value_mapper_clause

```
USING CASE_SENSITIVE MAPPING
```

JSON_value_on_empty_clause

```
{ ERROR | NULL | DEFAULT literal } ON EMPTY
```

JSON_value_on_error_clause

```
{ ERROR | NULL | DEFAULT literal } ON ERROR
```

JSON_value_on_mismatch_clause

```
JSON_value_on_mismatch (
  ( IGNORE | ERROR | NULL )
  ON MISMATCH
  [ ( (MISSING DATA) | (EXTRA DATA) | (TYPE ERROR) ) ]
) ...
```

JSON_value_return_object_instance

```
object_type_name [ JSON_value_mapper_clause ]
```

JSON_value_return_type

```
{ VARCHAR2 [ ( size [BYTE | CHAR] ) TRUNCATE ]
| CLOB
| NUMBER [ ( precision [, scale] ) ]
| DATE
| TIMESTAMP
| TIMESTAMP WITH TIME ZONE
| SDO_GEOMETRY
| JSON_value_return_object_instance
}
```

JSON_value_returning_clause

```
RETURNING JSON_value_return_type [ ASCII ]
```

key_clause

```
KEY { [(] attribute [)] | ( attribute [, attribute]... ) }
```

key_management_clauses

```
{ set_key
| create_key
| use_key
| set_key_tag
| export_keys
| import_keys
| migrate_key
| reverse_migrate_key
| move_keys
}
```

keystore_clause

```
KEystore IDENTIFIED BY [ ( EXTERNAL STORE ) | keystore_password ] [ (NO REKEY) ]
);
```

keystore_management_clauses

```
{ create_keystore
| open_keystore
| close_keystore
| backup_keystore
```

```
| alter_keystore_password
| merge_into_new_keystore
| merge_into_existing_keystore
| isolate_keystore
| unite_keystore
}
```

lead_lag_clause

```
HIERARCHY hierarchy_ref OFFSET offset_expr
[ {
  WITHIN { LEVEL | PARENT }
  | ACROSS ANCESTOR AT LEVEL level_ref [ POSITION FROM { BEGINNING | END }
}
]
```

lead_lag_expression

```
lead_lag_function_name ( calc_meas_expression ) OVER ( lead_lag_clause )
```

lead_lag_function_name

```
{ LAG | LAG_DIFF | LAG_DIFF_PERCENT | LEAD | LEAD_DIFF | LEAD_DIFF_PERCENT }
```

level_clause

```
LEVEL level IS
{ level_table.level_column
| ( level_table.level_column
  [, level_table.level_column ]...
)
} [ SKIP WHEN NULL ]
```

level_hier_clause

```
( level [ CHILD OF level ]... )
```

level_member_literal

```
level_ref { pos_member_keys | named_member_keys }
```

level_specification

```
( [ [ dim_name. ] hier_name. ] level_name )
```

levels_clause

```
LEVELS ( level_specification [, level_specification ]... )
```

list_partition_desc

```
PARTITION [partition]
list_values_clause
table_partition_description
[ ( range_subpartition_desc [, range_subpartition_desc]...
  | list_subpartition_desc [, list_subpartition_desc]...
  | individual_hash_subparts [, individual_hash_subparts]...
)
| hash_subparts_by_quantity
]
```

list_partitions

```
PARTITION BY LIST ( column [, column]... )
```

```
[ AUTOMATIC [ STORE IN ( tablespace [, tablespace ]... ) ] ]
(PARTITION [ partition ]
  list_values_clause table_partition_description
  [, PARTITION [ partition ]
    list_values_clause table_partition_description
    [ external_part_subpart_data_props ]
  ]...
)
```

list_partitionset_clause

```
PARTITIONSET BY LIST (column)
  PARTITION BY CONSISTENT HASH (column [, column]...)
  [ SUBPARTITION BY { { RANGE | HASH } (column [, column]...)
    | LIST (column)
  }
  [ subpartition_template ]
  ]
PARTITIONS AUTO ( list_partitionset_desc [, list_partitionset_desc]... )
```

list_partitionset_desc

```
PARTITIONSET partition_set list_values_clause
  [ TABLESPACE SET tablespace_set ]
  [ LOB_storage_clause ]
  [ SUBPARTITIONS STORE IN ( tablespace_set )... ]
```

list_subpartition_desc

```
SUBPARTITION [subpartition] list_values_clause
  [read_only_clause] [indexing_clause] [partitioning_storage_clause]
  [external_part_subpart_data_props]
```

list_values

```
list_values
{ { literal | NULL } [, { literal | NULL } ]... }
| { ( { literal | NULL } [, { literal | NULL } ]... )
  [, ( { literal | NULL } [, { literal | NULL } ]... ) ] }
```

list_values_clause

```
VALUES ( list_values | DEFAULT )
```

listagg_overflow_clause

```
{ ON OVERFLOW ERROR }
|
{ ON OVERFLOW TRUNCATE 'truncation-indicator' [ { WITH | WITHOUT } COUNT ] }
```

LOB_compression_clause

```
{ COMPRESS [HIGH | MEDIUM | LOW ]
| NOCOMPRESS
}
```

LOB_deduplicate_clause

```
{ DEDUPLICATE
| KEEP_DUPLICATES
}
```

LOB_parameters


```
{ { ENABLE | DISABLE } STORAGE IN ROW
  | CHUNK integer
  | PCTVERSION integer
  | FREEPools integer
  | LOB_retention_clause
  | LOB_deduplicate_clause
  | LOB_compression_clause
  | { ENCRYPT encryption_spec | DECRYPT }
  | { CACHE | NOCACHE | CACHE READS } [ logging_clause ]
}...
```

LOB_partition_storage

```
PARTITION partition
{ LOB_storage_clause | varray_col_properties }...
  [ (SUBPARTITION subpartition
    { LOB_partitioning_storage | varray_col_properties }...
  )
]
```

LOB_partitioning_storage

```
LOB (LOB_item) STORE AS [BASICFILE | SECUREFILE]
  [ LOB_segname [ ( TABLESPACE tablespace | TABLESPACE SET tablespace_set ) ]
  | ( TABLESPACE tablespace | TABLESPACE SET tablespace_set )
]
```

LOB_retention_storage

```
RETENTION [ MAX | MIN integer | AUTO | NONE ]
```

LOB_storage_clause

```
LOB
{ (LOB_item [, LOB_item ]...)
  STORE AS { {SECUREFILE | BASICFILE}
            | (LOB_storage_parameters)
          }...
| (LOB_item)
  STORE AS { {SECUREFILE | BASICFILE}
            | LOB_segname
            | (LOB_storage_parameters)
          }...
}
```

LOB_storage_parameters

```
{ { { TABLESPACE tablespace | TABLESPACE SET tablespace_set }
  | LOB_parameters [storage_clause]
}...
| storage_clause
```

local_domain_index_clause

```
LOCAL
  [ ( PARTITION partition [ PARAMETERS ( 'ODCI_parameters' ) ]
    [, PARTITION partition [ PARAMETERS ( 'ODCI_parameters' ) ] ]...
  )
]
```

local_partitioned_index

```
LOCAL
  [ on_range_partitioned_table
  | on_list_partitioned_table
```

```
| on_hash_partitioned_table
| on_comp_partitioned_table
]
```

local_XMLIndex_clause

```
LOCAL
[ ( PARTITION partition [ XMLIndex_parameters_clause ]
  [, PARTITION partition [ XMLIndex_parameters_clause ] ]...
  )
]
```

lockdown_features

```
{ DISABLE | ENABLE } FEATURE
{ { = ( 'feature' [, 'feature' ]... ) }
| { ALL [ EXCEPT = ( 'feature' [, 'feature' ]... ) ] }
}
```

lockdown_options

```
{ DISABLE | ENABLE } OPTION
{ { = ( 'option' [, 'option' ]... ) }
| { ALL [ EXCEPT = ( 'option' [, 'option' ]... ) ] }
}
```

lockdown_statements

```
{ DISABLE | ENABLE } STATEMENT
{ { = ( 'SQL_statement' [, 'SQL_statement' ]... ) }
| { = ( 'SQL_statement' ) statement_clauses }
| { ALL [ EXCEPT = ( 'SQL_statement' [, 'SQL_statement' ]... ) ] }
}
```

logfile_clause

```
LOGFILE
[ GROUP integer ] file_specification
[, [ GROUP integer ] file_specification ]...
```

logfile_clauses

```
{ { ARCHIVELOG [ MANUAL ]
| NOARCHIVELOG
}
| [ NO ] FORCE LOGGING
| SET STANDBY NOLOGGING FOR {DATA AVAILABILITY | LOAD PERFORMANCE}
| RENAME FILE 'filename' [, 'filename' ]...
  TO 'filename'
| CLEAR [ UNARCHIVED ]
  LOGFILE logfile_descriptor [, logfile_descriptor ]...
  [ UNRECOVERABLE DATAFILE ]
| add_logfile_clauses
| drop_logfile_clauses
| switch_logfile_clause
| supplemental_db_logging
}
```

logfile_descriptor

```
{ GROUP integer
| ('filename' [, 'filename' ]...)
| 'filename'
}
```

logging_clause

```
{ LOGGING | NOLOGGING | FILESYSTEM_LIKE_LOGGING }
```

main_model

```
[ MAIN main_model_name ]  
model_column_clauses  
[ cell_reference_options ]  
model_rules_clause
```

managed_standby_recovery

```
RECOVER  
{ MANAGED STANDBY DATABASE  
  [ { USING ARCHIVED LOGFILE  
    | DISCONNECT [FROM SESSION]  
    | NODELAY  
    | UNTIL CHANGE integer  
    | UNTIL CONSISTENT  
    | USING INSTANCES { ALL | integer }  
    | parallel_clause  
    }...  
  | FINISH  
  | CANCEL  
  ]  
| TO LOGICAL STANDBY { db_name | KEEP IDENTITY }  
}
```

mapping_table_clauses

```
{ MAPPING TABLE | NOMAPPING }
```

materialized_view_props

```
[ column_properties ]  
[ table_partitioning_clauses ]  
[ CACHE | NOCACHE ]  
[ parallel_clause ]  
[ build_clause ]
```

maximize_standby_db_clause

```
SET STANDBY DATABASE TO MAXIMIZE  
{ PROTECTION | AVAILABILITY | PERFORMANCE }
```

maxsize_clause

```
MAXSIZE { UNLIMITED | size_clause }
```

meas_aggregate_clause

```
AGGREGATE BY aggr_function
```

measure_ref

```
[ MEASURES. ] meas_name
```

measures_clause

```
MEASURES ( av_measure [, av_measure]... )
```

member_expression

```

{ level_member_literal
  | hier_navigation_expression
  | CURRENT MEMBER
  | NULL
  | ALL
}

```

memoptimize_read_clause

```
[ { (MEMOPTIMIZE FOR READ) | (NO MEMOPTIMIZE FOR READ) } ]
```

memoptimize_write_clause

```
[ { (MEMOPTIMIZE FOR WRITE) | (NO MEMOPTIMIZE FOR WRITE) } ]
```

merge_insert_clause

```

WHEN NOT MATCHED THEN
INSERT [ (column [, column ]...) ]
VALUES ( { expr | DEFAULT }
        [, { expr | DEFAULT } ]...
        )
[ where_clause ]

```

merge_into_existing_keystore

```

MERGE KEYSTORE 'keystore1_location' [ IDENTIFIED BY keystore1_password ]
  INTO EXISTING KEYSTORE 'keystore2_location' IDENTIFIED BY keystore2_password
  [ WITH BACKUP [ USING 'backup_identifier' ] ]

```

merge_into_new_keystore

```

MERGE KEYSTORE 'keystore1_location' [ IDENTIFIED BY keystore1_password ]
  AND KEYSTORE 'keystore2_location' [ IDENTIFIED BY keystore2_password ]
  INTO NEW KEYSTORE 'keystore3_location' IDENTIFIED BY keystore3_password

```

merge_table_partitions

```

MERGE PARTITIONS partition_or_key_value
  { , partition_or_key_value [, partition_or_key_value ]...
  | TO partition_or_key_value }
  [ INTO partition_spec ]
  [ filter_condition ]
  [ dependent_tables_clause ]
  [ update_index_clauses ]
  [ parallel_clause ]
  [ ONLINE ]
  [ allow_disallow_clustering ]

```

merge_table_subpartitions

```

MERGE SUBPARTITIONS subpartition_or_key_value
  { , subpartition_or_key_value [, subpartition_or_key_value ]...
  | TO subpartition_or_key_value }
  [ INTO { range_subpartition_desc
          | list_subpartition_desc
        }
  ]
  [ filter_condition ]
  [ dependent_tables_clause ]
  [ update_index_clauses ]
  [ parallel_clause ]
  [ ONLINE ]
  [ allow_disallow_clustering ]

```

merge_update_clause

```
WHEN MATCHED THEN
UPDATE SET column = { expr | DEFAULT }
        [, column = { expr | DEFAULT } ]...
[ where_clause ]
[ DELETE where_clause ]
```

migrate_key

```
SET [ ENCRYPTION ] KEY
    IDENTIFIED BY HSM_auth_string
    [ FORCE KEYSTORE ]
    MIGRATE USING software_keystore_password
    [ WITH BACKUP [ USING 'backup_identifier' ] ]
```

mining_analytic_clause

```
[ query_partition_clause ] [ order_by_clause ]
```

mining_attribute_clause

```
USING
{ *
| { [ schema . ] table . *
  | expr [ AS alias ]
}
  [, { [ schema . ] table . *
    | expr [ AS alias ]
  }
]...
}
```

model_clause

```
MODEL
  [ cell_reference_options ]
  [ return_rows_clause ]
  [ reference_model ]...
main_model
```

model_column_clauses

```
[ PARTITION BY (expr [ c_alias ] [, expr [c_alias] ]...) ]
DIMENSION BY (expr [c_alias] [, expr [c_alias] ]...)
MEASURES (expr [c_alias] [, expr [c_alias] ]...)
```

model_iterate_clause

```
ITERATE ( number ) [ UNTIL ( condition ) ]
```

model_rules_clause

```
[ RULES
  [ { UPDATE | UPSERT [ ALL ] } ]
  [ { AUTOMATIC | SEQUENTIAL } ORDER ]
  [ model_iterate_clause ]
]
( [ { UPDATE | UPSERT [ ALL ] } ]
cell_assignment [ order_by_clause ] = expr
  [, [ { UPDATE | UPSERT [ ALL ] } ]
    cell_assignment [ order_by_clause ] = expr
  ]...
)
```

modified_external_table

```
EXTERNAL MODIFY modify_external_table_properties
```

modify_col_properties

```
column [ datatype ]  
      [ COLLATE column_collation_name ]  
      [ DEFAULT [ ON NULL ] expr | identity_clause | DROP IDENTITY ]  
      [ { ENCRYPT encryption_spec } | DECRYPT ]  
      [ inline_constraint ... ]  
      [ LOB_storage_clause ]  
      [ alter_XMLSchema_clause ]
```

modify_col_substitutable

```
COLUMN column  
[ NOT ] SUBSTITUTABLE AT ALL LEVELS  
[ FORCE ]
```

modify_col_visibility

```
column { VISIBLE | INVISIBLE }
```

modify_collection_retrieval

```
MODIFY NESTED TABLE collection_item  
RETURN AS { LOCATOR | VALUE }
```

modify_column_clauses

```
MODIFY  
{ ( modify_col_properties | modify_virtcol_properties  
  [, modify_col_properties | modify_virtcol_properties ]... )  
| ( modify_col_visibility [, modify_col_visibility ]... )  
| modify_col_substitutable  
}
```

modify_diskgroup_file

```
MODIFY FILE 'filename' ATTRIBUTE ( disk_region_clause )  
[, 'filename' ATTRIBUTE ( disk_region_clause ) ]...
```

modify_external_table_properties

```
DEFAULT DIRECTORY directory  
[ LOCATION '(' directory ':' ' ' location_specifier ' ' ')' ]  
[ ACCESS PARAMETERS  
  [ BADFILE filename ]  
  [ LOGFILE filename ]  
  [ DISCARDFILE filename ] ]  
[ REJECT LIMIT { integer | UNLIMITED } ]
```

modify_filegroup_clause

```
MODIFY FILEGROUP filegroup_name  
SET '[ file_type. ] property_name' = 'property_value'
```

modify_hash_partition

```
MODIFY partition_extended_name  
{ partition_attributes  
| coalesce_table_subpartition  
| alter_mapping_table_clause
```

```

| [ REBUILD ] UNUSABLE LOCAL INDEXES
| read_only_clause
| indexing_clause
}

```

modify_index_default_attrs

```

MODIFY DEFAULT ATTRIBUTES
  [ FOR PARTITION partition ]
  { physical_attributes_clause
  | TABLESPACE { tablespace | DEFAULT }
  | logging_clause
  }...

```

modify_index_partition

```

MODIFY PARTITION partition
{ { deallocate_unused_clause
  | allocate_extent_clause
  | physical_attributes_clause
  | logging_clause
  | index_compression
  }...
| PARAMETERS ('ODCI_parameters')
| COALESCE [ CLEANUP ] [ parallel_clause ]
| UPDATE BLOCK REFERENCES
| UNUSABLE
}

```

modify_index_subpartition

```

MODIFY SUBPARTITION subpartition
{ UNUSABLE
| allocate_extent_clause
| deallocate_unused_clause
}

```

modify_list_partition

```

MODIFY partition_extended_name
  { partition_attributes
  | { ADD | DROP } VALUES ( list_values )
  | { add_range_subpartition
  | add_list_subpartition
  | add_hash_subpartition
  }
  | coalesce_table_subpartition
  | [ REBUILD ] UNUSABLE LOCAL INDEXES
  | read_only_clause
  | indexing_clause
  }

```

modify_LOB_parameters

```

{ storage_clause
| PCTVERSION integer
| FREEPools integer
| REBUILD FREEPools
| LOB_retention_clause
| LOB_deduplicate_clause
| LOB_compression_clause
| { ENCRYPT encryption_spec | DECRYPT }
| { CACHE
  | { NOCACHE | CACHE READS } [ logging_clause ]
}

```

```
| allocate_extent_clause  
| shrink_clause  
| deallocate_unused_clause  
} ...
```

modify_LOB_storage_clause

```
MODIFY LOB (LOB_item)  
  (modify_LOB_parameters)
```

modify_mv_column_clause

```
MODIFY ( column [ ENCRYPT encryption_spec  
  | DECRYPT ]  
)
```

modify_opaque_type

```
MODIFY OPAQUE TYPE anydata_column  
STORE ( type_name [, type_name ]... ) UNPACKED
```

modify_range_partition

```
MODIFY partition_extended_name  
  { partition_attributes  
  | { add_range_subpartition  
  | add_hash_subpartition  
  | add_list_subpartition  
  }  
  | coalesce_table_subpartition  
  | alter_mapping_table_clause  
  | [ REBUILD ] UNUSABLE LOCAL INDEXES  
  | read_only_clause  
  | indexing_clause  
  }
```

modify_table_default_attrs

```
MODIFY DEFAULT ATTRIBUTES  
  [ FOR partition_extended_name ]  
  [ deferred_segment_creation ]  
  [ read_only_clause ]  
  [ indexing_clause ]  
  [ segment_attributes_clause ]  
  [ table_compression ]  
  [ inmemory_clause ]  
  [ PCTTHRESHOLD integer ]  
  [ prefix_compression ]  
  [ alter_overflow_clause ]  
  [ { LOB (LOB_item) | VARRAY varray } (LOB_parameters) ]...
```

modify_table_partition

```
{ modify_range_partition  
| modify_hash_partition  
| modify_list_partition  
}
```

modify_table_subpartition

```
MODIFY subpartition_extended_name  
{ allocate_extent_clause  
| deallocate_unused_clause  
| shrink_clause  
| { { LOB LOB_item | VARRAY varray } (modify_LOB_parameters) }...
```



```

| [ REBUILD ] UNUSABLE LOCAL INDEXES
| { ADD | DROP } VALUES ( list_values )
| read_only_clause
| indexing_clause
}

```

modify_to_partitioned

```

MODIFY [ table_partitioning_clauses | NONPARTITIONED ]
  [ filter_condition ]
  [ ONLINE ]
  [ UPDATE INDEXES
    [ ( index { local_partitioned_index | global_partitioned_index | GLOBAL }
      [, index { local_partitioned_index | global_partitioned_index |
GLOBAL } ]... )
    ]
  ]
]

```

modify_virtcol_properties

```

column [ datatype ]
[ COLLATE column_collation_name ]
[ GENERATED ALWAYS ] AS (column_expression) [ VIRTUAL ]
evaluation_edition_clause [ unusable_editions_clause ]

```

modify_volume_clause

```

MODIFY VOLUME asm_volume
  [ ATTRIBUTE (disk_region_clause) ]
  [ MOUNTPATH 'mountpath_name' ]
  [ USAGE 'usage_name' ]

```

modify_table_default_attrs

```

MODIFY DEFAULT ATTRIBUTES
  [ FOR partition_extended_name ]
  [ DEFAULT DIRECTORY directory ]
  [ deferred_segment_creation ]
  [ read_only_clause ]
  [ indexing_clause ]
  [ segment_attributes_clause ]
  [ table_compression ]
  [ inmemory_clause ]
  [ PCTTHRESHOLD integer ]
  [ prefix_compression ]
  [ alter_overflow_clause ]
  [ { LOB (LOB_item) | VARRAY varray } (LOB_parameters) ]...

```

move_datafile_clause

```

MOVE DATAFILE ( 'filename' | 'ASM_filename' | file_number )
  [ TO ( 'filename' | 'ASM_filename' ) ]
  [ REUSE ] [ KEEP ]

```

move_mv_log_clause

```

MOVE segment_attributes_clause [parallel_clause]

```

move_table_clause

```

MOVE
  [ filter_condition ]
  [ ONLINE ]
  [ segment_attributes_clause ]
  [ table_compression ]

```

```

[ index_org_table_clause ]
[ { LOB_storage_clause | varray_col_properties }... ]
[ parallel_clause ]
[ allow_disallow_clustering ]
[ UPDATE INDEXES
  [ ( index { segment_attributes_clause
        | update_index_partition }
    [, index { segment_attributes_clause
              | update_index_partition } ]...
    )
  ]
]

```

move_table_partition

```

MOVE partition_extended_name
  [ MAPPING TABLE ]
  [ table_partition_description ]
  [ filter_condition ]
  [ update_index_clauses ]
  [ parallel_clause ]
  [ allow_disallow_clustering ]
  [ ONLINE ]

```

move_table_subpartition

```

MOVE subpartition_extended_name [ indexing_clause ]
  [ partitioning_storage_clause ]
  [ update_index_clauses ]
  [ filter_condition ]
  [ parallel_clause ]
  [ allow_disallow_clustering ]
  [ ONLINE ]

```

move_to_filegroup_clause

```

MOVE FILE 'ASM_filename' TO FILEGROUP filegroup_name

```

move_keys

```

MOVE [ENCRYPTION] KEYS
  TO NEW KEYSTORE keystore_location1
  IDENTIFIED BY keystore1_password
  FROM [FORCE] KEYSTORE
  IDENTIFIED BY keystore_password
  [WITH IDENTIFIER IN
    { 'key_identifier' [, 'key_identifier']... | ( subquery ) } ]
  [WITH BACKUP [USING 'backup_identifier'] ];

```

multi_column_for_loop

```

FOR (dimension_column
     [, dimension_column ]...)
IN ( { (literal [, literal ]...)
     [ (literal [, literal ]...) ]...
     | subquery
     }
    )

```

multi_table_insert

```

{ ALL
  { insert_into_clause [ values_clause ] [error_logging_clause] }...
  | conditional_insert_clause
} subquery

```

multiset_except

```
nested_table1  
MULTISET EXCEPT [ ALL | DISTINCT ]  
nested_table2
```

multiset_intersect

```
nested_table1  
MULTISET INTERSECT [ ALL | DISTINCT ]  
nested_table2
```

multiset_union

```
nested_table1  
MULTISET UNION [ ALL | DISTINCT ]  
nested_table2
```

mv_log_augmentation

```
ADD { { OBJECT ID  
      | PRIMARY KEY  
      | ROWID  
      | SEQUENCE  
      } [ (column [, column ]...) ]  
      | (column [, column ]... )  
      } [, { { OBJECT ID  
            | PRIMARY KEY  
            | ROWID  
            | SEQUENCE  
            }  
            [ (column [, column ]...) ]  
            | (column [, column ]... )  
            }  
      ]...  
      [ new_values_clause ]
```

mv_log_purge_clause

```
PURGE { IMMEDIATE [ SYNCHRONOUS | ASYNCHRONOUS ] )  
      | START WITH datetime_expr  
        [ NEXT datetime_expr  
          | REPEAT INTERVAL interval_expr  
          ]  
      | [ START WITH datetime_expr ] { NEXT datetime_expr  
                                       | REPEAT INTERVAL interval_expr  
                                       }  
      }
```

named_member_keys

```
'[ ' attr_name = [, attr_name = member_key_expr ]... ' ]'
```

nested_clause

```
table_reference (NESTED [PATH]) identifier  
[  
  ( "." [ JSON_object_key array_step ] ) |  
  ( "," JSON_basic_path_expression )  
]  
[ JSON_table_on_error_clause ]  
[ JSON_table_on_empty_clause ]  
JSON_columns_clause
```

nested_table_col_properties

```
NESTED TABLE
{ nested_item | COLUMN_VALUE }
[ substitutable_column_clause ]
[ LOCAL | GLOBAL ]
STORE AS storage_table
[ ( { (object_properties)
    | [ physical_properties ]
    | [ column_properties ]
    }...
)
]
[ RETURN [ AS ] { LOCATOR | VALUE } ]
```

nested_table_partition_spec

```
PARTITION partition [segment_attributes_clause]
```

new_values_clause

```
{ INCLUDING | EXCLUDING } NEW VALUES
```

number

```
[ + | - ]
{ digit [ digit ]... [ . ] [ digit [ digit ]... ]
| . digit [ digit ]...
}
[ [ e | E ] [ + | - ] digit [ digit ]... ] [ f | F | d | D ]
```

numeric_file_name

```
+diskgroup_name.filenumber.incarnation_number
```

object_properties

```
{ { column | attribute }
  [ DEFAULT expr ]
  [ { inline_constraint }... | inline_ref_constraint ]
| { out_of_line_constraint
  | out_of_line_ref_constraint
  | supplemental_logging_props
  }
}
```

object_step

```
.{ simple_name | "complex_name" | * }
```

object_table

```
OF
[ schema. ] object_type
[ object_table_substitution ]
[ (object_properties) ]
[ ON COMMIT { DELETE | PRESERVE } ROWS ]
[ OID_clause ]
[ OID_index_clause ]
[ physical_properties ]
[ table_properties ]
```

object_table_substitution

```
[ NOT ] SUBSTITUTABLE AT ALL LEVELS
```

object_type_col_properties

```
COLUMN column substitutable_column_clause
```

object_view_clause

```
OF [ schema. ] type_name  
{ WITH OBJECT { IDENTIFIER | ID }  
  { DEFAULT | ( attribute [, attribute ]... ) }  
| UNDER [ schema. ] superview  
}  
[ ( { out_of_line_constraint  
  | attribute { inline_constraint }...  
  } [, { out_of_line_constraint  
    | attribute { inline_constraint }...  
    }  
  ]...  
)  
]
```

OID_clause

```
OBJECT IDENTIFIER IS  
{ SYSTEM GENERATED | PRIMARY KEY }
```

OID_index_clause

```
OIDINDEX [ index ]  
( { physical_attributes_clause  
  | TABLESPACE tablespace  
  }...  
)
```

on_comp_partitioned_table

```
[ STORE IN ( tablespace [, tablespace ]... ) ]  
( PARTITION  
  [ partition ]  
  [ { segment_attributes_clause  
    | index_compression  
    }...  
  ] [ USABLE | UNUSABLE ] [ index_subpartition_clause ]  
  [, PARTITION  
    [ partition ]  
    [ { segment_attributes_clause  
      | index_compression  
      }...  
    ] [ USABLE | UNUSABLE ] [ index_subpartition_clause ]  
  ]...  
)
```

on_error_clause

```
( ERROR | NULL ) ON ERROR
```

on_hash_partitioned_table

```
{ STORE IN ( tablespace [, tablespace ]... )  
| ( PARTITION [ partition ] [ TABLESPACE tablespace ]  
  [ index_compression ] [ USABLE | UNUSABLE ]  
  [, PARTITION [ partition ] [ TABLESPACE tablespace ]  
    [ index_compression ] [ USABLE | UNUSABLE ] ] ...  
)
```

```
}
```

on_list_partitioned_table

```
( PARTITION
  [ partition ]
  [ { segment_attributes_clause
    | index_compression
  }...
  ] [ USABLE | UNUSABLE ]
  [, PARTITION
    [ partition ]
    [ { segment_attributes_clause
      | index_compression
    }...
    ] [ USABLE | UNUSABLE ]
  ]...
)
```

on_object_clause

```
ON { [ schema. ] object
  | USER user [, user]...
  | DIRECTORY directory_name
  | EDITION edition_name
  | MINING MODEL [ schema. ] mining_model_name
  | JAVA { SOURCE | RESOURCE } [ schema. ] object
  | SQL TRANSLATION PROFILE [ schema. ] profile
}
```

on_range_partitioned_table

```
( PARTITION
  [ partition ]
  [ { segment_attributes_clause
    | index_compression
  }...
  ] [ USABLE | UNUSABLE ]
  [, PARTITION
    [ partition ]
    [ { segment_attributes_clause
      | index_compression
    }...
    ] [ USABLE | UNUSABLE ]
  ]...
)
```

open_keystore

```
SET KEYSTORE OPEN
  [ FORCE KEYSTORE ]
  IDENTIFIED BY { EXTERNAL STORE | keystore_password }
  [ CONTAINER = { ALL | CURRENT } ]
```

option_values

```
{ { VALUE = ( 'option_value' [, 'option_value' ]... ) }
  |
  { MINVALUE = 'option_value' }
  |
  { MAXVALUE = 'option_value' }
}...
```

order_by_clause

```
ORDER [ SIBLINGS ] BY
{ expr | position | c_alias }
[ ASC | DESC ]
[ NULLS FIRST | NULLS LAST ]
[, { expr | position | c_alias }
  [ ASC | DESC ]
  [ NULLS FIRST | NULLS LAST ]
]...
```

ordinality_column

```
column_name FOR ORDINALITY
```

out_of_line_constraint

```
[ CONSTRAINT constraint_name ]
{ UNIQUE (column [, column ]...)
| PRIMARY KEY (column [, column ]...)
| FOREIGN KEY (column [, column ]...) references_clause
| CHECK (condition)
} [ constraint_state ]
```

out_of_line_part_storage

```
PARTITION partition
{ nested_table_col_properties | LOB_storage_clause | varray_col_properties }
[ nested_table_col_properties | LOB_storage_clause | varray_col_properties ]...
[ ( SUBPARTITION subpartition
  { nested_table_col_properties | LOB_storage_clause | varray_col_properties }
  [ nested_table_col_properties | LOB_storage_clause | varray_col_properties ]...
  [, SUBPARTITION subpartition
  { nested_table_col_properties | LOB_storage_clause | varray_col_properties }
  [ nested_table_col_properties | LOB_storage_clause | varray_col_properties ]...
  ]...
)
]
```

out_of_line_ref_constraint

```
{ SCOPE FOR ({ ref_col | ref_attr })
  IS [ schema. ] scope_table
| REF ({ ref_col | ref_attr }) WITH ROWID
| [ CONSTRAINT constraint_name ] FOREIGN KEY
  ( { ref_col [, ref_col ] | ref_attr [, ref_attr ] } ) references_clause
  [ constraint_state ]
}
```

outer_join_clause

```
[ query_partition_clause ] [ NATURAL ]
outer_join_type JOIN table_reference
[ query_partition_clause ]
[ ON condition
| USING ( column [, column ]...)
]
```

outer_join_type

```
{ FULL | LEFT | RIGHT } [ OUTER ]
```

parallel_clause

```
{ NOPARALLEL | PARALLEL [ integer ] }
```

parallel_pdb_creation_clause

```
PARALLEL [ integer ]
```

partial_database_recovery

```
{ TABLESPACE tablespace [, tablespace ]...  
| DATAFILE { 'filename' | filenumber }  
            [, 'filename' | filenumber ]...  
}
```

partial_index_clause

```
INDEXING { PARTIAL | FULL }
```

partition_attributes

```
[ { physical_attributes_clause  
  | logging_clause  
  | allocate_extent_clause  
  | deallocate_unused_clause  
  | shrink_clause  
  }...  
]  
[ OVERFLOW  
  { physical_attributes_clause  
  | logging_clause  
  | allocate_extent_clause  
  | deallocate_unused_clause  
  }...  
]  
[ table_compression ]  
[ inmemory_clause ]  
[ { { LOB LOB_item | VARRAY varray } (modify_LOB_parameters) }...]
```

partition_extended_name

```
PARTITION partition  
|  
PARTITION FOR ( partition_key_value [, partition_key_value]... )
```

partition_extended_names

```
{ PARTITION | PARTITIONS }  
partition | { FOR ( partition_key_value [, partition_key_value ]... ) }  
            [, partition | { FOR ( partition_key_value [, partition_key_value ]... ) } ]...
```

partition_extension_clause

```
{ PARTITION (partition)  
| PARTITION FOR (partition_key_value [, partition_key_value]...)  
| SUBPARTITION (subpartition)  
| SUBPARTITION FOR (subpartition_key_value [, subpartition_key_value]...)  
}
```

partition_or_key_value

```
partition  
|  
FOR ( partition_key_value [, partition_key_value ]... )
```

partition_spec

```
PARTITION [ partition ] [ table_partition_description ]
```


partitioning_storage_clause

```
[ { { TABLESPACE tablespace | TABLESPACE SET tablespace_set }
  | OVERFLOW [ TABLESPACE tablespace ] | TABLESPACE SET tablespace_set ]
  | table_compression
  | index_compression
  | inmemory_clause
  | ilm_clause
  | LOB_partitioning_storage
  | VARRAY varray_item STORE AS [ SECUREFILE | BASICFILE ] LOB LOB_segname
} ...
]
```

partitionset_clauses

```
{ range_partitionset_clause | list_partitionset_clause }
```

password_parameters

```
{ { FAILED_LOGIN_ATTEMPTS
  | PASSWORD_LIFE_TIME
  | PASSWORD_REUSE_TIME
  | PASSWORD_REUSE_MAX
  | PASSWORD_LOCK_TIME
  | PASSWORD_GRACE_TIME
  | INACTIVE_ACCOUNT_TIME
}
{ expr | UNLIMITED | DEFAULT }
| PASSWORD_VERIFY_FUNCTION
{ function | NULL | DEFAULT }
| PASSWORD_ROLLOVER_TIME
}
```

patch_common

```
target_expr [ json_query_returning_clause ] [ pretty ]
[ ASCII ] [ TRUNCATE ] [ json_query_on_error_clause ]
```

path_prefix_clause

```
PATH_PREFIX = { 'path_name' | directory_object_name | NONE }
```

pdb_change_state

```
[ pdb_name ] { pdb_open | pdb_close | pdb_save_or_discard_state }
```

pdb_change_state_from_root

```
{ pdb_name [, pdb_name ]... | ALL [ EXCEPT pdb_name [, pdb_name ]... ] }
{ pdb_open | pdb_close | pdb_save_or_discard_state }
```

pdb_close

```
CLOSE [ IMMEDIATE ] [ instances_clause | relocate_clause ]
```

pdb_datafile_clause

```
[ pdb_name ] DATAFILE
{ { { 'filename' | filenumber } [, 'filename' | filenumber ]... } | ALL }
{ ONLINE | OFFLINE }
```

pdb_dba_roles

```
ROLES = ( role [, role ]... )
```

pdb_force_logging_clause

```
{ ENABLE | DISABLE } FORCE { LOGGING | NOLOGGING }  
| SET STANDBY NOLOGGING FOR {DATA AVAILABILITY | LOAD PERFORMANCE}
```

pdb_general_recovery

```
RECOVER [ AUTOMATIC ] [ FROM 'location' ]  
  [ DATABASE  
  |  
  TABLESPACE tablespace [, tablespace ]...  
  |  
  DATAFILE { 'filename' | filenumber }  
            [, 'filename' | filenumber ]...  
  |  
  LOGFILE 'filename'  
  |  
  CONTINUE [ DEFAULT ]  
  ]
```

pdb_logging_clauses

```
{ logging_clause  
| pdb_force_logging_clause  
}
```

pdb_open

```
OPEN  
  { [ READ WRITE | READ ONLY ] [ RESTRICTED ] [ FORCE ]  
  | [ READ WRITE ] UPGRADE [ RESTRICTED ]  
  | RESETLOGS  
  }  
  [ instances_clause ]  
  [ services_clause ]
```

pdb_recovery_clauses

```
[ pdb_name ] { pdb_general_recovery  
              | { BEGIN | END } BACKUP  
              | { ENABLE | DISABLE } RECOVERY  
              }
```

pdb_refresh_mode_clause

```
REFRESH MODE { MANUAL | EVERY refresh_interval { MINUTES | HOURS } | NONE }
```

pdb_save_or_discard_state

```
{ SAVE | DISCARD } STATE [ instances_clause ]
```

pdb_settings_clauses

```
{ [ pdb_name ]  
  { DEFAULT EDITION = edition_name  
  | SET DEFAULT ( BIGFILE | SMALLFILE ) TABLESPACE  
  | DEFAULT TABLESPACE tablespace_name  
  | DEFAULT TEMPORARY TABLESPACE { tablespace | tablespace_group_name }  
  | RENAME GLOBAL_NAME TO database.domain [. domain ]...  
  | set_time_zone_clause  
  | database_file_clauses  
  | supplemental_db_logging  
  | pdb_storage_clause  
  | pdb_logging_clauses  
  }
```

```

| pdb_refresh_mode_clause
| REFRESH
| SET CONTAINER_MAP = 'map_object'
}
}
| CONTAINERS DEFAULT TARGET = { (container_name) | NONE
| HOST "=" "" hostname ""
| PORT "=" number
}

```

pdb_storage_clause

```

STORAGE
{ ( { MAXSIZE { UNLIMITED | size_clause }
|
| MAX_AUDIT_SIZE { UNLIMITED | size_clause }
|
| MAX_DIAG_SIZE { UNLIMITED | size_clause }
}...
)
|
UNLIMITED
}

```

pdb_snapshot_clause

```
ENABLE SNAPSHOT { MANUAL | EVERY snapshot_interval { HOURS | MINUTES } | NONE}
```

pdb_unplug_clause

```
pdb_name UNPLUG INTO 'filename'
```

period_definition

```
PERIOD FOR valid_time_column [ ( start_time_column, end_time_column ) ]
```

permanent_tablespace_attrs

```

{ MINIMUM EXTENT size_clause
| BLOCKSIZE integer [ K ]
| logging_clause
| FORCE LOGGING
| tablespace_encryption_clause
| default_tablespace_params
| { ONLINE | OFFLINE }
| extent_management_clause
| segment_management_clause
| flashback_mode_clause
| lost_write_protection
}...

```

permanent_tablespace_clause

```

TABLESPACE tablespace
[ DATAFILE file_specification [, file_specification ]... ]
[ permanent_tablespace_attrs ]

```

physical_attributes_clause

```

[ { PCTFREE integer
| PCTUSED integer
| INITRANS integer
| storage_clause
}...
]

```

physical_properties

```
{ [ deferred_segment_creation ] segment_attributes_clause [ table_compression ]
  [ inmemory_table_clause ] [ ilm_clause ]
| [ deferred_segment_creation ] ORGANIZATION
  { HEAP [ segment_attributes_clause ] heap_org_table_clause
  | INDEX [ segment_attributes_clause ] index_org_table_clause
  | EXTERNAL PARTITION ATTRIBUTES external_table_clause [ REJECT LIMIT ]
  }
| CLUSTER cluster (column [, column ]...)
}
```

pivot_clause

```
PIVOT [ XML ]
  ( aggregate_function ( expr ) [[AS] alias ]
    [, aggregate_function ( expr ) [[AS] alias ] ]...
    pivot_for_clause
    pivot_in_clause
  )
```

pivot_for_clause

```
FOR { column
    | ( column [, column]... )
  }
```

pivot_in_clause

```
IN ( { { { expr
        | ( expr [, expr]... )
        } [ [ AS] alias]
      }...
    | subquery
    | ANY [, ANY]...
  }
)
```

plsql_declarations

```
{ function_declaration | procedure_declaration }...
```

pos_member_keys

```
'[ ' member_key_expr [, member_key_expr]... ]'
```

preceding_boundary

```
{ UNBOUNDED PRECEDING | offset_expr PRECEDING }
AND
{ CURRENT MEMBER
  | offset_expr { PRECEDING | FOLLOWING }
  | UNBOUNDED FOLLOWING
}
```

prefix_compression

```
COMPRESS [ integer ] | NOCOMPRESS
```

prepare_clause

```
PREPARE MIRROR COPY copy_name
WITH { EXTERNAL | NORMAL | HIGH } REDUNDANCY
```

privilege_audit_clause

```
PRIVILEGES system_privilege [, system_privilege ]...
```

program_unit

```
{ FUNCTION [ schema. ] function_name  
|  
PROCEDURE [ schema. ] procedure_name  
|  
PACKAGE [ schema. ] package_name }
```

property_clause

```
PROPERTY { SET | REMOVE } DEFAULT_CREDENTIAL = SYSTEM.OPCTEST
```

proxy_clause

```
{ GRANT CONNECT THROUGH { ENTERPRISE USERS | db_user_proxy db_user_proxy_clauses }  
| REVOKE CONNECT THROUGH { ENTERPRISE USERS | db_user_proxy }}
```

qdr_expression

```
QUALIFY ( calc_meas_expression, qualifier )
```

qualified_disk_clause

```
search_string  
[ NAME disk_name ]  
[ SIZE size_clause ]  
[ FORCE | NOFORCE ]
```

qualified_template_clause

```
ATTRIBUTE  
( redundancy_clause  
  striping_clause  
  disk_region_clause  
)
```

qualifier

```
hierarchy_ref = member_expression
```

query_block

```
[ with_clause ]  
SELECT [ hint ] [ { { DISTINCT | UNIQUE } | ALL } ] select_list  
FROM { table_reference | join_clause | ( join_clause ) }  
     [ , { table_reference | join_clause | (join_clause) } ] ...  
[ where_clause ]  
[ hierarchical_query_clause ]  
[ group_by_clause ]  
[ model_clause ]
```

query_partition_clause

```
PARTITION BY  
  { expr[, expr ]...  
  | ( expr[, expr ]... )  
  }
```

query_rewrite_clause

```
{ ENABLE | DISABLE } QUERY REWRITE [ unusable_editions_clause ]
```

query_table_expression

```
{ query_name  
| [ schema. ]  
  { table [ modified_external_table  
          | partition_extension_clause  
          | @ dblink  
          ]  
  | { view | materialized view } [ @ dblink ]  
  | hierarchy  
  | analytic_view [ HIERARCHIES  
    ( [ [ attr_dim. ] hierarchy [, [ attr_dim. ] hierarchy ]... ] ) ]  
  | inline_external_table  
  } [sample_clause]  
| [ LATERAL ] (subquery [ subquery_restriction_clause ] )  
| table_collection_expression  
}
```

quiesce_clauses

```
QUIESCE RESTRICTED | UNQUIESCE
```

quotagroup_clauses

```
{ ADD QUOTAGROUP quotagroup_name [ SET property_name = property_value ]  
| MODIFY QUOTAGROUP quotagroup_name SET property_name = property_value  
| MOVE FILEGROUP filegroup_name TO quotagroup_name  
| DROP QUOTAGROUP quotagroup_name  
}
```

range_partition_desc

```
PARTITION [partition]  
range_values_clause  
table_partition_description  
( ( { range_subpartition_desc [, range_subpartition_desc] ...  
    | list_subpartition_desc [, list_subpartition_desc] ...  
    | individual_hash_subparts [, individual_hash_subparts] ...  
    }  
  ) | hash_subparts_by_quantity ]
```

range_partitions

```
PARTITION BY RANGE (column[, column]...)  
[ INTERVAL (expr) [ STORE IN ( tablespace [, tablespace]... ) ] ]  
( PARTITION [ partition ]  
  range_values_clause table_partition_description  
  [, PARTITION [ partition ]  
    range_values_clause table_partition_description  
    [ external_part_subpart_data_props ]  
  ]...  
)
```

range_partitionset_clause

```
PARTITIONSET BY RANGE (column [, column]...)  
PARTITION BY CONSISTENT HASH (column [, column]...)  
[ SUBPARTITION BY { { RANGE | HASH } (column [, column]...)  
  | LIST (column)  
  }  
[ subpartition_template ]  
]  
PARTITIONS AUTO ( range_partitionset_desc [, range_partitionset_desc]... )
```

range_partitionset_desc

```
PARTITIONSET partition_set range_values_clause  
[ TABLESPACE SET tablespace_set ]  
[ LOB_storage_clause ]  
[ SUBPARTITIONS STORE IN ( tablespace_set )... ]
```

range_subpartition_desc

```
SUBPARTITION [subpartition] range_values_clause  
[read_only_clause] [indexing_clause] [partitioning_storage_clause]  
[external_part_subpart_data_props]
```

range_values_clause

```
VALUES LESS THAN  
( { literal | MAXVALUE }  
  [, { literal | MAXVALUE } ]...  
)
```

read_only_clause

```
{ READ ONLY } | { READ WRITE }
```

rebalance_diskgroup_clause

```
REBALANCE  
[ [ { WITH | WITHOUT } phase [, phase]... ] [ POWER integer ] [ WAIT | NOWAIT ] }  
|  
{ MODIFY POWER [ integer ] }  
]
```

rebuild_clause

```
REBUILD  
[ { PARTITION partition  
  | SUBPARTITION subpartition  
  }  
| { REVERSE | NOREVERSE }  
]  
[ parallel_clause  
| TABLESPACE tablespace  
| PARAMETERS ( 'ODCI_parameters' )  
| XMLIndex_parameters_clause  
| ONLINE  
| physical_attributes_clause  
| index_compression  
| logging_clause  
| partial_index_clause  
]...
```

records_per_block_clause

```
{ MINIMIZE | NOMINIMIZE } RECORDS_PER_BLOCK
```

recovery_clauses

```
{ general_recovery  
| managed_standby_recovery  
| BEGIN BACKUP  
| END BACKUP  
}
```

redo_log_file_spec

```
[ 'filename | ASM_filename'
| ('filename | ASM_filename'
  [, 'filename | ASM_filename' ]...)
]
[ SIZE size_clause ]
[ BLOCKSIZE size_clause
[ REUSE ]
```

redundancy_clause

```
[ MIRROR | HIGH | UNPROTECTED | PARITY ]
```

reference_model

```
REFERENCE reference_model_name ON (subquery)
  model_column_clauses [ cell_reference_options ]
```

reference_partition_desc

```
PARTITION [partition] [table_partition_description] )
```

reference_partitioning

```
PARTITION BY REFERENCE ( constraint )
  [ (reference_partition_desc...) ]
```

references_clause

```
REFERENCES [ schema. ] object [ (column [, column ]...) ]
  [ON DELETE { CASCADE | SET NULL } ]
```

register_logfile_clause

```
REGISTER [ OR REPLACE ]
  [ PHYSICAL | LOGICAL ]
LOGFILE [ file_specification [, file_specification ]...
  [ FOR logminer_session_name ]
```

regular_entry

```
[ KEY ] expr VALUE expr
  | expr [ ":" expr ]
```

relational_properties

```
{ column_definition
| virtual_column_definition
| period_definition
| { out_of_line_constraint | out_of_line_ref_constraint }
| supplemental_logging_props
}
[, { column_definition
  | virtual_column_definition
  | period_definition
  | { out_of_line_constraint | out_of_line_ref_constraint }
  | supplemental_logging_props
}
]...
```

relational_table

```
[ (relational_properties) ]
[ immutable_table_clauses ]
[ blockchain_table_clauses ]
```



```
[ DEFAULT COLLATION collation_name ]
[ ON COMMIT { DROP | PRESERVE } DEFINITION ]
[ ON COMMIT { DELETE | PRESERVE } ROWS ]
[ physical_properties ]
[ table_properties ]
```

relocate_clause

```
RELOCATE [ TO 'instance_name' ]
| NORELOCATE
```

rename_column_clause

```
RENAME COLUMN old_name TO new_name
```

rename_disk_clause

```
RENAME
{ DISK old_disk_name TO new_disk_name [, old_disk_name TO new_disk_name ]...
| DISKS ALL }
```

rename_index_partition

```
RENAME
{ PARTITION partition | SUBPARTITION subpartition }
TO new_name
```

rename_partition_subpart

```
RENAME { partition_extended_name
| subpartition_extended_name
} TO new_name
```

replace_disk_clause

```
REPLACE DISK disk_name WITH 'path_name' [ FORCE | NOFORCE ]
[, disk_name WITH 'path_name' [ FORCE | NOFORCE ] ]...
[ POWER integer ] [ WAIT | NOWAIT ]
```

resize_disk_clause

```
RESIZE ALL [ SIZE size_clause ]
```

resource_parameters

```
{ { SESSIONS_PER_USER
| CPU_PER_SESSION
| CPU_PER_CALL
| CONNECT_TIME
| IDLE_TIME
| LOGICAL_READS_PER_SESSION
| LOGICAL_READS_PER_CALL
| COMPOSITE_LIMIT
}
{ integer | UNLIMITED | DEFAULT }
| PRIVATE_SGA
{ size_clause | UNLIMITED | DEFAULT }
}
```

return_rows_clause

```
RETURN { UPDATED | ALL } ROWS
```

returning_clause

```
{ RETURN | RETURNING } expr [, expr ]...  
INTO data_item [, data_item ]...
```

reverse_migrate_key

```
SET [ ENCRYPTION ] KEY  
  IDENTIFIED BY software_keystore_password  
  [ FORCE KEYSTORE ]  
  REVERSE MIGRATE USING HSM_auth_string
```

revoke_object_privileges

```
{ object_privilege | ALL [ PRIVILEGES ] }  
  [, { object_privilege | ALL [ PRIVILEGES ] } ]...  
on_object_clause  
FROM revokee_clause  
[ CASCADE CONSTRAINTS | FORCE ]
```

revoke_roles_from_programs

```
{ role [, role ]... | ALL } FROM program_unit [, program_unit ]...
```

revoke_system_privileges

```
{ system_privilege | role | ALL PRIVILEGES }  
  [, { system_privilege | role | ALL PRIVILEGES } ]...  
FROM revokee_clause
```

revokee_clause

```
{ user | role | PUBLIC }  
  [, { user | role | PUBLIC } ]...
```

role_audit_clause

```
ROLES role [, role ]...
```

rolling_migration_clauses

```
{ START ROLLING MIGRATION TO 'ASM_version'  
  | STOP ROLLING MIGRATION  
}
```

rolling_patch_clauses

```
{ START ROLLING PATCH  
  | STOP ROLLING PATCH  
}
```

rollup_cube_clause

```
{ ROLLUP | CUBE } (grouping_expression_list)
```

routine_clause

```
[ schema. ] [ type. | package. ]  
{ function | procedure | method }  
[ @dblink_name ]  
( [ argument [, argument ]... ] )
```

row_limiting_clause

```
[ OFFSET offset { ROW | ROWS } ]  
[ FETCH { FIRST | NEXT } [ { rowcount | percent PERCENT } ]  
  { ROW | ROWS } { ONLY | WITH TIES } ]
```

row_movement_clause

```
{ ENABLE | DISABLE } ROW MOVEMENT
```

row_pattern

```
[ row_pattern | ] row_pattern_term
```

Note: The vertical bar is part of the syntax rather than BNF notation.

row_pattern_aggregate_func

```
[ RUNNING | FINAL ] aggregate_function
```

row_pattern_classifier_func

```
CLASSIFIER()
```

row_pattern_clause

```
MATCH_RECOGNIZE (  
  [ row_pattern_partition_by ]  
  [ row_pattern_order_by ]  
  [ row_pattern_measures ]  
  [ row_pattern_rows_per_match ]  
  [ row_pattern_skip_to ]  
  PATTERN (row_pattern)  
  [ row_pattern_subset_clause ]  
  DEFINE row_pattern_definition_list  
)
```

row_pattern_definition

```
variable_name AS condition
```

row_pattern_definition_list

```
row_pattern_definition [, row_pattern_definition ]...
```

row_pattern_factor

```
row_pattern_primary [ row_pattern_quantifier ]
```

row_pattern_match_num_func

```
MATCH_NUMBER()
```

row_pattern_measure_column

```
expr AS c_alias
```

row_pattern_measures

```
MEASURES row_pattern_measure_column [, row_pattern_measure_column ]...
```

row_pattern_nav_compound

```
{ PREV | NEXT }  
( [ RUNNING | FINAL ] { FIRST | LAST } ( expr [, offset ] ) [, offset] )
```

row_pattern_nav_logical

```
[ RUNNING | FINAL ] { FIRST | LAST } ( expr [, offset ] )
```

row_pattern_nav_physical

```
{ PREV | NEXT } ( expr [, offset ] )
```

row_pattern_navigation_func

```
row_pattern_nav_logical  
| row_pattern_nav_physical  
| row_pattern_nav_compound
```

row_pattern_order_by

```
ORDER BY column [, column ]...
```

row_pattern_partition_by

```
PARTITION BY column [, column ]...
```

row_pattern_permute

```
PERMUTE ( row_pattern [, row_pattern ]...)
```

row_pattern_primary

variable_name

```
| $  
| ^  
| ( [ row_pattern ] )  
| { - row_pattern - }  
| row_pattern_permute
```

Note: The curly brackets are part of the syntax rather than BNF notation.

row_pattern_quantifier

```
* [ ? ]  
| + [ ? ]  
| ? [ ? ]  
| { [ unsigned_integer ] , [ unsigned_integer ] } [ ? ]  
| { unsigned_integer }
```

Note: The curly brackets are part of the syntax rather than BNF notation.

row_pattern_rec_func

```
row_pattern_classifier_func  
| row_pattern_match_num_func  
| row_pattern_navigation_func  
| row_pattern_aggregate_func
```

row_pattern_rows_per_match

```
ONE ROW PER MATCH  
| ALL ROWS PER MATCH
```

row_pattern_skip_to

```
AFTER MATCH {  
  SKIP TO NEXT ROW  
  | SKIP PAST LAST ROW  
  | SKIP TO FIRST variable_name  
  | SKIP TO LAST variable_name
```

```
| SKIP TO variable_name  
}
```

row_pattern_subset_clause

```
SUBSET row_pattern_subset_item [, row_pattern_subset_item ]...
```

row_pattern_subset_item

```
variable_name = ( variable_name [, variable_name ] )
```

row_pattern_term

```
[ row_pattern_term ] row_pattern_factor
```

sample_clause

```
SAMPLE [ BLOCK ]  
      (sample_percent)  
      [ SEED (seed_value) ]
```

scoped_table_ref_constraint

```
{ SCOPE FOR ({ ref_column | ref_attribute })  
  IS [ schema. ] { scope_table_name | c_alias }  
}
```

scrub_clause

```
SCRUB [ FILE 'ASM_filename' | DISK disk_name ]  
      [ REPAIR | NOREPAIR ]  
      [ POWER { AUTO | LOW | HIGH | MAX } ]  
      [ WAIT | NOWAIT ]  
      [ FORCE | NOFORCE ]  
      [ STOP ]
```

search_clause

```
{ SEARCH  
  { DEPTH FIRST BY c_alias [, c_alias]...  
    [ ASC | DESC ]  
    [ NULLS FIRST | NULLS LAST ]  
  | BREADTH FIRST BY c_alias [, c_alias]...  
    [ ASC | DESC ]  
    [ NULLS FIRST | NULLS LAST ]  
  }  
  SET ordering_column  
}
```

searched_case_expression

```
{ WHEN condition THEN return_expr }...
```

secret_management_clauses

```
{ add_update_secret  
| delete_secret  
| add_update_secret_seps  
| delete_secret_seps  
}
```

security_clause

```
GUARD { ALL | STANDBY | NONE }
```

security_clauses

```
{ { ENABLE | DISABLE } RESTRICTED SESSION
  | SET ENCRYPTION WALLET OPEN
  IDENTIFIED BY { "wallet_password" | "HSM_auth_string" }
  | SET ENCRYPTION WALLET CLOSE
  [ IDENTIFIED BY { "wallet_password" | "HSM_auth_string" } ]
  | set_encryption_key
}
```

segment_attributes_clause

```
{ physical_attributes_clause
  | { TABLESPACE tablespace | TABLESPACE SET tablespace_set }
  | logging_clause
}...
```

segment_management_clause

```
SEGMENT SPACE MANAGEMENT { AUTO | MANUAL }
```

select_list

```
{ *
  | { query_name.*
    | [ schema. ] { table | view | materialized view } .*
    | t_alias.*
    | expr [ [ AS ] c_alias ]
  }
  [, { query_name.*
    | [ schema. ] { table | view | materialized view } .*
    | t_alias.*
    | expr [ [ AS ] c_alias ]
  }
  ]...
}
```

service_name_convert

```
SERVICE_NAME_CONVERT =
  { ( 'service_name', 'replacement_service_name'
    [, 'service_name', 'replacement_service_name' ]... )
  |
  NONE
}
```

set_encryption_key

```
{ SET ENCRYPTION KEY
  {
    [ "certificate_id" ] IDENTIFIED BY "wallet_password"
    |
    IDENTIFIED BY "HSM_auth_string" [ MIGRATE USING "wallet_password" ]
  }
}
```

set_key

```
SET [ ENCRYPTION ] KEY { mkid:mk | mk }
  [ USING TAG 'tag' ]
  [ USING ALGORITHM 'encrypt_algorithm' ]
  [ FORCE KEYSTORE ]
  IDENTIFIED BY { EXTERNAL STORE | keystore_password }
  [ WITH BACKUP [ USING 'backup_identifier' ] ]
  [ CONTAINER = { ALL | CURRENT } ]
```

set_key_tag

```
SET TAG 'tag' FOR 'key_id'  
  [ FORCE KEYSTORE ]  
  IDENTIFIED BY { EXTERNAL STORE | keystore_password }  
  [ WITH BACKUP [ USING 'backup_identifier' ] ]
```

set_parameter_clause

```
parameter_name =  
  parameter_value [, parameter_value ]...  
  [ COMMENT = string ]  
  [ DEFERRED ]  
  [ CONTAINER = { CURRENT | ALL } ]  
  [ { SCOPE = { MEMORY | SPFILE | BOTH }  
    | SID = { 'sid' | '*' }  
  }...  
  ]
```

set_subpartition_template

```
SET SUBPARTITION TEMPLATE  
  { ( range_subpartition_desc [, range_subpartition_desc]... )  
  | ( list_subpartition_desc [, list_subpartition_desc]... )  
  | ( individual_hash_subparts [, individual_hash_subparts]... )  
  | ( )  
  | hash_subpartition_quantity  
  }
```

set_time_zone_clause

```
SET TIME_ZONE =  
  '{ { + | - } hh : mi | time_zone_region }'
```

shards_clause

```
SHARDS ( [schema.] { table | view } )
```

share_clause

```
HIERARCHY hierarchy_ref  
  { PARENT  
  | LEVEL level_ref  
  | MEMBER member_expression  
  }
```

share_of_expression

```
SHARE_OF ( calc_meas_expression share_clause )
```

sharing_clause

```
SHARING = { METADATA | DATA | NONE }
```

shrink_clause

```
SHRINK SPACE [ COMPACT ] [ CASCADE ]
```

shutdown_dispatcher_clause

```
SHUTDOWN [ IMMEDIATE ] dispatcher_name
```

simple_case_expression

```
expr
{ WHEN comparison_expr THEN return_expr }...
```

single_column_for_loop

```
FOR dimension_column
{ IN ( { literal [, literal ]...
      | subquery
      }
    )
  | [ LIKE pattern ] FROM literal TO literal
  { INCREMENT | DECREMENT } literal
}
```

single_table_insert

```
insert_into_clause
{ values_clause [ returning_clause ]
  | subquery
} [ error_logging_clause ]
```

size_clause

```
integer [ K | M | G | T | P | E ]
```

source_file_directory

```
SOURCE_FILE_DIRECTORY = { 'directory_path_name' | NONE }
```

source_file_name_convert

```
SOURCE_FILE_NAME_CONVERT =
{ ( 'filename_pattern', 'replacement_filename_pattern'
  [, 'filename_pattern', 'replacement_filename_pattern' ]... )
  |
  NONE
}
```

split_index_partition

```
SPLIT PARTITION partition_name_old
  AT (literal [, literal ]...)
  [ INTO (index_partition_description,
          index_partition_description
        )
  ]
  [ parallel_clause ]
```

split_nested_table_part

```
NESTED TABLE column INTO
( nested_table_partition_spec, nested_table_partition_spec
  [split_nested_table_part]
) [split_nested_table_part]
```

split_table_partition

```
SPLIT partition_extended_name
{ AT (literal [, literal ]... )
  [ INTO ( range_partition_desc, range_partition_desc ) ]
  | VALUES ( list_values )
  [ INTO ( list_partition_desc, list_partition_desc ) ]
  | INTO ( { range_partition_desc [, range_partition_desc ]...
          | list_partition_desc [, list_partition_desc ]... }
  )
```



```

    , partition_spec )
} [ split_nested_table_part ]
  [ filter_condition ]
  [ dependent_tables_clause ]
  [ update_index_clauses ]
  [ parallel_clause ]
  [ allow_disallow_clustering ]
  [ ONLINE ]

```

split_table_subpartition

```

SPLIT subpartition_extended_name
{ AT ( literal [, literal]... )
  [ INTO ( range_subpartition_desc, range_subpartition_desc ) ]
  | VALUES ( list_values )
  [ INTO ( list_subpartition_desc, list_subpartition_desc ) ]
  | INTO ( { range_subpartition_desc [, range_subpartition_desc ]...
           | list_subpartition_desc [, list_subpartition_desc ]... }
          , subpartition_spec )
} [ filter_condition ]
  [ dependent_tables_clause ]
  [ update_index_clauses ]
  [ parallel_clause ]
  [ allow_disallow_clustering ]
  [ ONLINE ]

```

sql_format

```
[+ | -] days hours : minutes : seconds [. frac_secs ]
```

standard_actions

```

ACTIONS
{ { object_action | ALL }
  ON { DIRECTORY directory_name
      | MINING MODEL [ schema. ] object_name
      | [ schema. ] object_name }
  | { system_action | ALL }
}
[ { object_action | ALL }
  ON { DIRECTORY directory_name
      | MINING MODEL [ schema. ] object_name
      | [ schema. ] object_name }
  | { system_action | ALL } ]...

```

standby_database_clauses

```

{ { activate_standby_db_clause
  | maximize_standby_db_clause
  | register_logfile_clause
  | commit_switchover_clause
  | start_standby_clause
  | stop_standby_clause
  | convert_database_clause
} [ parallel_clause ] }
|
{ switchover_clause | failover_clause }

```

standbys_clause

```

STANDBYS = { ( 'cdb_name' [, 'cdb_name' ]... )
             | { ALL [ EXCEPT ( 'cdb_name' [, 'cdb_name' ]... ) ] }
             | NONE
           }

```

start_standby_clause

```
START LOGICAL STANDBY APPLY
[ IMMEDIATE ]
[ NODELAY ]
[ NEW PRIMARY dblink
| INITIAL [ scn_value ]
| { SKIP FAILED TRANSACTION | FINISH }
]
```

startup_clauses

```
{ MOUNT [ { STANDBY | CLONE } DATABASE ]
| OPEN
  { [ READ WRITE ]
    [ RESETLOGS | NORESETLOGS ]
    [ UPGRADE | DOWNGRADE ]
  | READ ONLY
  }
}
```

statement_clauses

```
CLAUSE
{ { = ( 'clause' [, 'clause' ]... ) }
| { = ( 'clause' ) clause_options }
| { ALL [ EXCEPT = ( 'clause' [, 'clause' ]... ) ] }
}
```

static_base_profile

```
FROM base_profile
```

still_image_object_types

```
{ SI_StillImage
| SI_AverageColor
| SI_PositionalColor
| SI_ColorHistogram
| SI_Texture
| SI_FeatureList
| SI_Color
}
```

stop_standby_clause

```
{ STOP | ABORT } LOGICAL STANDBY APPLY
```

storage_clause

```
STORAGE
({ INITIAL size_clause
| NEXT size_clause
| MINEXTENTS integer
| MAXEXTENTS { integer | UNLIMITED }
| maxsize_clause
| PCTINCREASE integer
| FREELISTS integer
| FREELIST GROUPS integer
| OPTIMAL [ size_clause | NULL ]
| BUFFER_POOL { KEEP | RECYCLE | DEFAULT }
| FLASH_CACHE { KEEP | NONE | DEFAULT }
| CELL_FLASH_CACHE { KEEP | NONE | DEFAULT }
| ENCRYPT
} ...
```

)

storage_table_clause

```
WITH {SYSTEM | USER} MANAGED STORAGE TABLES
```

string

```
[ {N | n} ]  
{ '[ c ]...'  
| { Q | q } 'quote_delimiter c [ c ]... quote_delimiter'  
}
```

striping_clause

```
[ FINE | COARSE ]
```

subpartition_by_hash

```
SUBPARTITION BY HASH (column [, column ]...)  
  [ SUBPARTITIONS integer  
    [ STORE IN (tablespace [, tablespace ]...) ]  
  | subpartition_template  
  ]
```

subpartition_by_list

```
SUBPARTITION BY LIST ( column [, column]... ) [ subpartition_template ]
```

subpartition_by_range

```
SUBPARTITION BY RANGE ( column [, column]... ) [subpartition_template]
```

subpartition_extended_name

```
SUBPARTITION subpartition  
|  
SUBPARTITION FOR ( subpartition_key_value [, subpartition_key_value]... )
```

subpartition_extended_names

```
{ SUBPARTITION | SUBPARTITIONS }  
subpartition | { FOR ( subpartition_key_value [, subpartition_key_value ]... ) }  
  [, subpartition | { FOR ( subpartition_key_value [,  
subpartition_key_value ]... ) } ]...
```

subpartition_or_key_value

```
subpartition  
|  
FOR ( subpartition_key_value [, subpartition_key_value ]... )
```

subpartition_spec

```
SUBPARTITION [ subpartition ] [ partitioning_storage_clause ]
```

subpartition_template

```
SUBPARTITION TEMPLATE  
( { range_subpartition_desc [, range_subpartition_desc] ...  
  | list_subpartition_desc [, list_subpartition_desc] ...  
  | individual_hash_subparts [, individual_hash_subparts] ...  
  }
```

```
) | hash_subpartition_quantity
```

subquery

```
{ query_block  
| subquery { UNION [ALL] | INTERSECT | MINUS } subquery  
  [ { UNION [ALL] | INTERSECT | MINUS } subquery ]...  
| ( subquery )  
} [ order_by_clause ] [ row_limiting_clause ]
```

subquery_factoring_clause

```
query_name ([c_alias [, c_alias]...]) AS (subquery) [search_clause] [cycle_clause]  
[, query_name ([c_alias [, c_alias]...]) AS (subquery) [search_clause]  
[cycle_clause]]...
```

subquery_restriction_clause

```
WITH { READ ONLY  
      | CHECK OPTION  
      } [ CONSTRAINT constraint ]
```

substitutable_column_clause

```
{ [ ELEMENT ] IS OF [ TYPE ] ( ONLY type )  
| [ NOT ] SUBSTITUTABLE AT ALL LEVELS  
}
```

supplemental_db_logging

```
{ ADD | DROP } SUPPLEMENTAL LOG  
{ DATA  
| supplemental_id_key_clause  
| supplemental_plsql_clause  
| supplemental_subset_replication_clause  
}
```

supplemental_id_key_clause

```
DATA  
( { ALL | PRIMARY KEY | UNIQUE | FOREIGN KEY }  
  [, { ALL | PRIMARY KEY | UNIQUE | FOREIGN KEY } ]...  
)  
COLUMNS
```

supplemental_log_grp_clause

```
GROUP log_group  
(column [ NO LOG ]  
  [, column [ NO LOG ] ]...)  
[ ALWAYS ]
```

supplemental_logging_props

```
SUPPLEMENTAL LOG { supplemental_log_grp_clause  
                  | supplemental_id_key_clause  
                  }
```

supplemental_plsql_clause

```
DATA FOR PROCEDURAL REPLICATION
```

supplemental_subset_replication_clause

supplemental_table_logging

```
{ ADD SUPPLEMENTAL LOG
  { supplemental_log_grp_clause | supplemental_id_key_clause }
  [, SUPPLEMENTAL LOG
    { supplemental_log_grp_clause | supplemental_id_key_clause }
  ]...
| DROP SUPPLEMENTAL LOG
  { supplemental_id_key_clause | GROUP log_group }
  [, SUPPLEMENTAL LOG
    { supplemental_id_key_clause | GROUP log_group }
  ]...
}
```

switch_logfile_clause

```
SWITCH ALL LOGFILES TO BLOCKSIZE integer
```

switchover_clause

```
SWITCHOVER TO target_db_name [ VERIFY | FORCE ]
```

system_partitioning

```
PARTITION BY SYSTEM [ PARTITIONS integer
  | reference_partition_desc
  [, reference_partition_desc ...]
]
```

table_collection_expression

```
TABLE (collection_expression) [ (+) ]
```

table_compression

```
COMPRESS
| ROW STORE COMPRESS [ BASIC | ADVANCED ]
| COLUMN STORE COMPRESS [ FOR { QUERY | ARCHIVE } [ LOW | HIGH ] ]
  [ [NO] ROW LEVEL LOCKING ]
| NOCOMPRESS
```

table_index_clause

```
[ schema. ] table [ t_alias ]
(index_expr [ ASC | DESC ]
[, index_expr [ ASC | DESC ] ]...)
[ index_properties ]
```

table_partition_description

```
[ { INTERNAL | EXTERNAL } ]
[ deferred_segment_creation ]
[ read_only_clause ]
[ indexing_clause ]
[ segment_attributes_clause ]
[ table_compression | prefix_compression ]
[ inmemory_clause ]
[ ilm_clause ]
[ OVERFLOW [ segment_attributes_clause ] ]
[ { LOB_storage_clause
  | varray_col_properties
  | nested_table_col_properties
```

```
}...  
]
```

table_partitioning_clauses

```
{ range_partitions  
| list_partitions  
| hash_partitions  
| composite_range_partitions  
| composite_list_partitions  
| composite_hash_partitions  
| reference_partitioning  
| system_partitioning  
| consistent_hash_partitions  
| consistent_hash_with_subpartitions  
| partitionset_clauses  
}
```

table_properties

```
[ column_properties ]  
[ read_only_clause ]  
[ indexing_clause ]  
[ table_partitioning_clauses ]  
[ attribute_clustering_clause ]  
[ CACHE | NOCACHE ]  
[ RESULT_CACHE ( MODE {DEFAULT | FORCE } ) ]  
[ parallel_clause ]  
[ ROWDEPENDENCIES | NOROWDEPENDENCIES ]  
[ enable_disable_clause ]...  
[ row_movement_clause ]  
[ flashback_archive_clause ]  
[ ROW ARCHIVAL ]  
[ { AS subquery } | { FOR EXCHANGE WITH TABLE [ schema .] table } ]
```

table_reference

```
{ { { ONLY (query_table_expression) | query_table_expression }  
| flashback_query_clause ]  
| pivot_clause | unpivot_clause | row_pattern_clause ] }  
| containers_clause  
| shards_clause  
}  
[ t_alias ]
```

tablespace_clauses

```
{ EXTENT MANAGEMENT LOCAL  
| DATAFILE file_specification [, file_specification ]...  
| SYSAUX DATAFILE file_specification [, file_specification ]...  
| default_tablespace  
| default_temp_tablespace  
| undo_tablespace  
}
```

tablespace_datafile_clauses

```
DATAFILES { SIZE size_clause | autoextend_clause }...
```

tablespace_encryption_clause

```
ENCRYPTION [ { [ tablespace_encryption_spec ] ENCRYPT } | DECRYPT ]
```

tablespace_encryption_spec

```
USING 'encrypt_algorithm'
```

tablespace_group_clause

```
TABLESPACE GROUP { tablespace_group_name | '' }
```

tablespace_logging_clauses

```
{ logging_clause  
| [ NO ] FORCE LOGGING  
}
```

tablespace_retention_clause

```
RETENTION { GUARANTEE | NOGUARANTEE }
```

tablespace_state_clauses

```
{ { ONLINE  
| OFFLINE [ NORMAL | TEMPORARY | IMMEDIATE ]  
}  
| READ { ONLY | WRITE }  
| { PERMANENT | TEMPORARY }  
}
```

tempfile_reuse_clause

```
TEMPFILE REUSE
```

temporary_tablespace_clause

```
{ { TEMPORARY TABLESPACE }  
| { LOCAL TEMPORARY TABLESPACE FOR { ALL | LEAF } }  
} tablespace  
[ TEMPFILE file_specification [, file_specification ]... ]  
[ tablespace_group_clause ]  
[ extent_management_clause ]  
[ tablespace_encryption_clause ]
```

timeout_clause

```
DROP AFTER integer { M | H }
```

trace_file_clause

```
TRACE  
[ AS 'filename' [ REUSE ] ]  
[ RESETLOGS | NORESETLOGS ]
```

truncate_partition_subpart

```
TRUNCATE { partition_extended_names | subpartition_extended_names }  
[ { DROP [ ALL ] | REUSE } STORAGE ]  
[ update_index_clauses [ parallel_clause ] ] [ CASCADE ]
```

ts_file_name_convert

```
FILE_NAME_CONVERT =  
( 'filename_pattern', 'replacement_filename_pattern'  
[, 'filename_pattern', 'replacement_filename_pattern' ]... )  
[ KEEP ]
```

undo_mode_clause

```
LOCAL UNDO { ON | OFF }
```

undo_tablespace

```
[ BIGFILE | SMALLFILE ]  
UNDO TABLESPACE tablespace  
[ DATAFILE file_specification [, file_specification ]...]
```

undo_tablespace_clause

```
UNDO TABLESPACE tablespace  
[ DATAFILE file_specification [, file_specification ]... ]  
[ extent_management_clause ]  
[ tablespace_retention_clause ]  
[ tablespace_encryption_clause ]
```

undrop_disk_clause

```
UNDROP DISKS
```

unite_keystore

```
UNITE KEYSTORE IDENTIFIED BY isolated_keystore_password  
WITH ROOT KEYSTORE [ FORCE KEYSTORE ]  
IDENTIFIED BY { EXTERNAL STORE | united_keystore_password }  
[ WITH BACKUP [ USING 'backup_identifier' ] ]
```

unpivot_clause

```
UNPIVOT [ {INCLUDE | EXCLUDE} NULLS ]  
( { column | ( column [, column]... ) }  
  pivot_for_clause  
  unpivot_in_clause  
)
```

unpivot_in_clause

```
IN  
( { column | ( column [, column]... ) }  
  [ AS { literal | ( literal [, literal]... ) } ]  
  [, { column | ( column [, column]... ) }  
    [ AS {literal | ( literal [, literal]... ) } ]  
  ]...  
)
```

unusable_editions_clause

```
[ UNUSABLE BEFORE { CURRENT EDITION | EDITION edition } ]  
[ UNUSABLE BEGINNING WITH { CURRENT EDITION | EDITION edition | NULL EDITION } ]
```

update_all_indexes_clause

```
UPDATE INDEXES  
[ ( index ( update_index_partition  
           | update_index_subpartition  
         )  
  [, index ( update_index_partition  
            | update_index_subpartition  
          )  
  ]...  
  )  
]
```


update_global_index_clause

```
{ UPDATE | INVALIDATE } GLOBAL INDEXES
```

update_index_clauses

```
{ update_global_index_clause  
| update_all_indexes_clause  
}
```

update_index_partition

```
index_partition_description [ index_subpartition_clause ]  
[, index_partition_description [ index_subpartition_clause ] ]...
```

update_index_subpartition

```
SUBPARTITION [ subpartition ]  
[ TABLESPACE tablespace ]  
[, SUBPARTITION [ subpartition ]  
[ TABLESPACE tablespace ]  
]...
```

update_set_clause

```
SET  
{ { (column [, column ]...) = (subquery)  
| column = { expr | (subquery) | DEFAULT }  
}  
[, { (column [, column]...) = (subquery)  
| column = { expr | (subquery) | DEFAULT }  
}  
]...  
| VALUE (t_alias) = { expr | (subquery) }  
}
```

upgrade_table_clause

```
UPGRADE [ [NOT ] INCLUDING DATA ]  
[ column_properties ]
```

use_key

```
USE [ ENCRYPTION ] KEY 'key_id'  
[ USING TAG 'tag' ]  
[ FORCE KEYSTORE ]  
IDENTIFIED BY { EXTERNAL STORE | keystore_password }  
[ WITH BACKUP [ USING 'backup_identifier' ] ]
```

user_clauses

```
{ ADD USER user [, 'user']...  
| DROP USER user [, 'user']... [CASCADE]  
| REPLACE USER 'old_user' WITH 'new_user' [, 'old_user' WITH 'new_user']...  
}
```

user_tablespaces_clause

```
USER_TABLESPACES =  
{ ( 'tablespace' [, 'tablespace' ]... )  
| ALL [ EXCEPT ( 'tablespace' [, 'tablespace' ]... ) ]  
| NONE  
}  
[ SNAPSHOT COPY | NO DATA | COPY | MOVE | NOCOPY ]
```

usergroup_clauses

```
{ ADD USERGROUP 'usergroup' WITH MEMBER 'user' [, 'user']...  
| MODIFY USERGROUP 'usergroup' { ADD | DROP } MEMBER 'user' [, 'user']...  
| DROP USERGROUP 'usergroup'  
}
```

using_clause

```
USING [ schema. ] fact_table_or_view [ [ AS ] alias ]
```

using_function_clause

```
USING [ schema. ] [ package. | type. ] function_name
```

using_index_clause

```
USING INDEX  
{ [ schema. ] index  
| (create_index_statement)  
| index_properties  
}
```

using_snapshot_clause

```
USING SNAPSHOT { snapshot_name | AT SCN snapshot_SCN | AT snapshot_timestamp }
```

using_statistics_type

```
USING { [ schema. ] statistics_type | NULL }
```

using_type_clause

```
USING [ schema. ] implementation_type [ array_DML_clause ]
```

validation_clauses

```
{ VALIDATE REF UPDATE [ SET DANGLING TO NULL ]  
| VALIDATE STRUCTURE  
  [ CASCADE { FAST | COMPLETE { OFFLINE | ONLINE } [ into_clause ] } ]  
}
```

values_clause

```
VALUES ( { expr | DEFAULT }  
        [, { expr | DEFAULT } ]...  
        )
```

varray_col_properties

```
VARRAY varray_item  
{ [ substitutable_column_clause ] varray_storage_clause  
| substitutable_column_clause  
}
```

varray_storage_clause

```
STORE AS [SECUREFILE | BASICFILE] LOB  
{ [LOB_segname] ( LOB_storage_parameters )  
| LOB_segname  
}
```

virtual_column_definition

```
column [ datatype [ COLLATE column_collation_name ] ]  
  [ VISIBLE | INVISIBLE ]  
  [ GENERATED ALWAYS ] AS (column_expression) [ VIRTUAL ]  
  [ evaluation_edition_clause ] [ unusable_editions_clause ]  
  [ inline_constraint [ inline_constraint ]... ]
```

where_clause

```
WHERE condition
```

wildcard

```
[ id "." ] id "." "*"
```

window_clause

```
HIERARCHY hierarchy_ref  
  BETWEEN { preceding_boundary | following_boundary }  
[ WITHIN { LEVEL  
          | PARENT  
          | ANCESTOR AT LEVEL level_name  
          }]
```

window_expression

```
aggregate_function OVER ( window_clause )
```

windowing_clause

```
{ ROWS | RANGE }  
{ BETWEEN  
  { UNBOUNDED PRECEDING  
    | CURRENT ROW  
    | value_expr { PRECEDING | FOLLOWING }  
  }  
  AND  
  { UNBOUNDED FOLLOWING  
    | CURRENT ROW  
    | value_expr { PRECEDING | FOLLOWING }  
  }  
  | { UNBOUNDED PRECEDING  
    | CURRENT ROW  
    | value_expr PRECEDING  
  }  
}
```

with_clause

```
WITH [ plsql_declarations ] [ subquery_factoring_clause ]
```

XML_attributes_clause

```
XMLATTRIBUTES  
  ( [ ENTITYESCAPING | NOENTITYESCAPING ]  
    [ SCHEMACHECK | NOSCHEMACHECK ]  
    value_expr [ { [AS] c_alias } | { AS EVALNAME value_expr } ]  
    [, value_expr [ { [AS] c_alias } | { AS EVALNAME value_expr } ] ]...  
  )
```

XMLnamespaces_clause

```
XMLNAMESPACES  
  ( { string AS identifier } | { DEFAULT string }  
    [, { string AS identifier } | { DEFAULT string } ]...  
  )
```

)

XML_passing_clause

```
PASSING [ BY VALUE ]
    expr [ AS identifier ]
    [, expr [ AS identifier ]
    ]...
```

XML_table_column

```
column
    { FOR ORDINALITY
    | { datatype | XMLTYPE [ (SEQUENCE) BY REF ] }
    [ PATH string ] [ DEFAULT expr ]
    }
```

XMLIndex_clause

```
[XDB.] XMLINDEX [ local_XMLIndex_clause ]
    [ parallel_clause ]
    [ XMLIndex_parameters_clause ]
```

XMLSchema_spec

```
[ XMLSCHEMA XMLSchema_URL ]
ELEMENT { element | XMLSchema_URL # element }
    [ STORE ALL VARRAYS AS { LOBS | TABLES } ]
    [ { ALLOW | DISALLOW } NONSCHEMA ]
    [ { ALLOW | DISALLOW } ANYSCHEMA ]
```

XMLTABLE_options

```
[ XML_passing_clause ]
[ RETURNING SEQUENCE BY REF ]
[ COLUMNS XML_table_column [, XML_table_column]...]
```

XMLType_column_properties

```
XMLTYPE [ COLUMN ] column
    [ XMLType_storage ]
    [ XMLSchema_spec ]
```

XMLType_storage

```
STORE
{ AS
{ OBJECT RELATIONAL
| [SECUREFILE | BASICFILE]
{ CLOB | BINARY XML }
    [ { LOB_segname [ (LOB_parameters) ]
    | (LOB_parameters)
    }
    ]
}
| { ALL VARRAYS AS { LOBS | TABLES } }
}
```

XMLType_table

```
OF XMLTYPE
    [ (object_properties) ]
    [ XMLTYPE XMLType_storage ]
    [ XMLSchema_spec ]
```

```
[ XMLType_virtual_columns ]  
[ ON COMMIT { DELETE | PRESERVE } ROWS ]  
[ OID_clause ]  
[ OID_index_clause ]  
[ physical_properties ]  
[ table_properties ]
```

XMLType_view_clause

```
OF XMLTYPE [ XMLSchema_spec ]  
WITH OBJECT { IDENTIFIER | ID }  
{ DEFAULT | ( expr [, expr ]... ) }
```

XMLType_virtual_columns

```
VIRTUAL COLUMNS ( column AS (expr) [, column AS (expr) ]... )
```

ym_iso_format

```
[ - ] P [ years Y ] [ months M ] [ days D ]  
[ T [ hours H ] [ minutes M ] [ seconds [. frac_secs] S ] ]
```

zonemap_attributes

```
{ TABLESPACE tablespace  
| SCALE integer  
| { CACHE | NOCACHE }  
}...
```

zonemap_clause

```
{ WITH MATERIALIZED ZONEMAP [ ( zonemap_name ) ] }  
|  
{ WITHOUT MATERIALIZED ZONEMAP }
```

zonemap_refresh_clause

```
REFRESH  
[ FAST | COMPLETE | FORCE ]  
[ ON { DEMAND | COMMIT | LOAD | DATA MOVEMENT | LOAD DATA MOVEMENT } ]
```

6 データ型

この章では、Oracleで認識され、SQLで使用可能なデータ型について説明します。

この章の内容は次のとおりです。

- [データ型の概要](#)
- [Oracle組込みデータ型](#)
- [Oracleが提供するデータ型](#)
- [Oracleデータ型への変換](#)

データ型の概要

データ型は、特定のタイプの情報またはデータの分類です。Oracleが処理する値は、それぞれデータ型を持ちます。値のデータ型により、固定された一連のプロパティが値に関連付けられます。このプロパティに応じて、Oracleは、あるデータ型の値を別のデータ型の値と区別して扱います。

次に、Oracleが認識するデータ型を示します。

ANSIでサポートされているデータ型

```
{ CHARACTER [VARYING] (size)
| { CHAR | NCHAR } VARYING (size)
| VARCHAR (size)
| NATIONAL { CHARACTER | CHAR }
  [VARYING] (size)
| { NUMERIC | DECIMAL | DEC }
  [ (precision [, scale ]) ]
| { INTEGER | INT | SMALLINT }
| FLOAT [ (size) ]
| DOUBLE PRECISION
| REAL
}
```

Oracle組込みデータ型

```
{ character_datatypes
| number_datatypes
| long_and_raw_datatypes
| datetime_datatypes
| large_object_datatypes
| rowid_datatypes
}
```

Oracleが提供するデータ型

```
{ any_types
| XML_types
| spatial_types
| media_types
}
```

ユーザー定義データ型

ユーザー定義データ型は、Oracle組込みデータ型と、その他のユーザー定義データ型を使用して、アプリケーション内のデータの構造と動作をモデル化します。

関連項目:

データ・タイプの詳細は、『[Oracle Database SQL言語リファレンス](#)』を参照してください。

Oracle組込みデータ型

この項では、Oracle組込みデータ型の種類を示します。

character_datatypes

```
{ CHAR [ (size [ BYTE | CHAR ]) ]  
| VARCHAR2 (size [ BYTE | CHAR ])  
| NCHAR [ (size) ]  
| NVARCHAR2 (size)  
}
```

datetime_datatypes

```
{ DATE  
| TIMESTAMP [ (fractional_seconds_precision) ]  
  [ WITH [ LOCAL ] TIME ZONE ]  
| INTERVAL YEAR [ (year_precision) ] TO MONTH  
| INTERVAL DAY [ (day_precision) ] TO SECOND  
  [ (fractional_seconds_precision) ]  
}
```

large_object_datatypes

```
{ BLOB | CLOB | NCLOB | BFILE }
```

long_and_raw_datatypes

```
{ LONG | LONG RAW | RAW (size) }
```

number_datatypes

```
{ NUMBER [ (precision [, scale ]) ]  
| FLOAT [ (precision) ]  
| BINARY_FLOAT  
| BINARY_DOUBLE  
}
```

rowid_datatypes

```
{ ROWID | UROWID [ (size) ] }
```

データ型のコードは、Oracle Databaseが内部的に使用します。DUMP関クションによって、列またはオブジェクト属性のデータ型コードが戻されます。

表6-1 組込みデータ型の概要

コード	データ型	説明
1	VARCHAR2(size [BYTE CHAR])	最大長が size バイトまたは文字の可変長文字列。VARCHAR2 には、size を指定する必要があります。最小の size は 1 バイトまたは 1 文字です。最大の size は、次のとおりです。

コード	データ型	説明
		<ul style="list-style-type: none"> ● MAX_STRING_SIZE = EXTENDED の場合は、32,767 バイト(文字) ● MAX_STRING_SIZE = STANDARD の場合は、4,000 バイト(文字) <p>MAX_STRING_SIZE 初期化パラメータの詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。</p> <p>BYTE は、列がバイト長セマンティクスを持つことを示します。CHAR は、列がキャラクタ・セマンティクスを持つことを示します。</p>
1	NVARCHAR2(size)	<p>最大長が size 文字の可変長 Unicode 文字列。NVARCHAR2 の size を指定する必要があります。最大バイト数は、AL16UTF16 エンコーディングの場合は size の 2 倍、UTF8 エンコーディングの場合は size の 3 倍です。最大の size は、各国語文字セット定義によって決定されます。上限は次のとおりです。</p> <ul style="list-style-type: none"> ● 32767 バイト(MAX_STRING_SIZE = EXTENDED の場合) ● 4000 バイト(MAX_STRING_SIZE = STANDARD の場合) <p>MAX_STRING_SIZE 初期化パラメータの詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。</p>
2	NUMBER [(p [, s])]	<p>精度 p、位取り s を持つ数。精度 p には 1 から 38 の値を指定できます。位取り s には -84 から 127 の値を指定できます。精度とスケールは両方とも 10 進数です。NUMBER の値は 1 から 22 バイトである必要があります。</p>
2	FLOAT [(p)]	<p>精度 p を持つ NUMBER データ型のサブタイプ。FLOAT の値は、内部的に NUMBER と表されます。精度 p には 1 から 126 の 2 進数を指定できます。FLOAT の値は 1 から 22 バイトである必要があります。</p>
8	LONG	<p>最大 2GB(2^{31} から 1 を引いたバイト数)の可変長文字データ。下位互換性を維持するために提供されています。</p>
12	DATE	<p>紀元前 4712 年 1 月 1 日から紀元 9999 年 12 月 31 日までの日付を指定します。デフォルトの書式は、NLS_DATE_FORMAT パラ</p>

コード	データ型	説明
		メータを使用すると明示的に、NLS_TERRITORY パラメータを使用すると暗黙的に決定される。サイズは 7 バイトに固定されています。このデータ型には、YEAR、MONTH、DAY、HOUR、MINUTE および SECOND の日時フィールドが含まれます。小数部の秒数およびタイムゾーンは含まれません。
100	BINARY_FLOAT	32 ビットの浮動小数点数。このデータ型には 4 バイトが必要です。
101	BINARY_DOUBLE	64 ビットの浮動小数点数。このデータ型には 8 バイトが必要です。
180	TIMESTAMP [(fractional_seconds_precision)]	日付の年、月、日および時刻の時、分、秒の値。 fractional_seconds_precision は、SECOND 日時フィールドの小数部の桁数です。fractional_seconds_precision の有効範囲は 0 から 9 です。デフォルトは 6 です。デフォルトの書式は、NLS_TIMESTAMP_FORMAT パラメータによって明示的に決まるか、または NLS_TERRITORY パラメータによって暗黙的に決まります。サイズは、精度に応じて 7 または 11 バイトです。このデータ型には、YEAR、MONTH、DAY、HOUR、MINUTE および SECOND の日時フィールドが含まれます。小数部の秒数は含まれますが、タイムゾーンは含まれません。
181	TIMESTAMP [(fractional_seconds_precision)] WITH TIME ZONE	タイムゾーンによる時差などのすべての TIMESTAMP の値。 fractional_seconds_precision は、SECOND 日時フィールドの小数部の桁数です。0 から 9 までの値を使用できます。デフォルトは 6 です。デフォルトの書式は、NLS_TIMESTAMP_FORMAT パラメータによって明示的に決まるか、または NLS_TERRITORY パラメータによって暗黙的に決まります。サイズは 13 バイトに固定されています。このデータ型には、YEAR、MONTH、DAY、HOUR、MINUTE、SECOND、TIMEZONE_HOUR および TIMEZONE_MINUTE の日時フィールドが含まれます。小数部の秒数および明示的なタイムゾーンが含まれます。
231	TIMESTAMP [(fractional_seconds_precision)] WITH LOCAL TIME ZONE	TIMESTAMP WITH TIME ZONE のすべての値。ただし、次に示す例外があります。 <ul style="list-style-type: none"> ● データベースへの格納時、データがデータベースのタイムゾーンに正規化されている場合。 ● データの検索時、ユーザーがセッションのタイムゾーンでデータを検索する場合。

コード	データ型	説明
		デフォルトの書式は、NLS_TIMESTAMP_FORMAT パラメータによって明示的に決まるか、または NLS_TERRITORY パラメータによって暗黙的に決まります。サイズは、精度に応じて 7 または 11 バイトです。
182	INTERVAL YEAR [(year_precision)] TO MONTH	年および月で期間を格納。year_precision は、YEAR 日時フィールドの桁数です。0 から 9 までの値を使用できます。デフォルトは 2 です。サイズは 5 バイトに固定されています。
183	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]	<p>日、時、分および秒で期間を格納。</p> <ul style="list-style-type: none"> ● day_precision は、DAY 日時フィールドの最大桁数です。0 から 9 までの値を使用できます。デフォルトは 2 です。 ● fractional_seconds_precision は、SECOND フィールドの小数部の桁数です。0 から 9 までの値を使用できます。デフォルトは 6 です。 <p>サイズは 11 バイトに固定されています。</p>
23	RAW(size)	<p>長さ size バイトのバイナリ・データ。RAW 値には、size を指定する必要があります。最大の size は、次のとおりです。</p> <ul style="list-style-type: none"> ● 32767 バイト(MAX_STRING_SIZE = EXTENDED の場合) ● 2000 バイト(MAX_STRING_SIZE = STANDARD の場合) <p>MAX_STRING_SIZE 初期化パラメータの詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。</p>
24	LONG RAW	最大 2GB の可変長バイナリ・データ。
69	ROWID	表の行のアドレスを一意に表す BASE64 文字列。主に、ROWID 疑似列によって戻される値のためのデータ型です。
208	UROWID [(size)]	索引構成表の行の論理アドレスを表す BASE64 文字列。オプションの size は、UROWID 型の列のサイズです。最大サイズおよびデフォルトは 4000 バイトです。

コード	データ型	説明
96	CHAR [(size [BYTE CHAR])]	長さ size バイトまたは size 文字の固定長文字データ。最大の size は 2000 バイトまたは文字です。デフォルトは size の最小値の 1 バイトです。 BYTE および CHAR は、VARCHAR2 と同じセマンティクスを持ちます。
96	NCHAR[(size)]	長さ size 文字の固定長文字データ。バイト数は、AL16UTF16 エンコードの場合は size の 2 倍まで、UTF8 エンコードの場合は size の 3 倍までです。最大の size は、各国語文字セット定義(上限 2000 バイト)によって決定されます。デフォルトは size の最小値の 1 文字です。
112	CLOB	シングルバイト文字またはマルチバイト文字を含むキャラクタ・ラージ・オブジェクト。固定幅および可変幅の文字セットがサポートされ、両方ともデータベース文字セットで使用されます。最大サイズは、4GB から 1 を引いたバイト数にデータベース・ブロック・サイズを掛けた値です。
112	NCLOB	Unicode キャラクタを含むキャラクタ・ラージ・オブジェクト。固定幅および可変幅の文字セットがサポートされます。両方の文字セットでデータベースの各国語文字セットを使用します。最大サイズは、4GB から 1 を引いたバイト数にデータベース・ブロック・サイズを掛けた値です。各国語文字セット・データが格納されます。
113	BLOB	バイナリ・ラージ・オブジェクト。最大サイズは、4GB から 1 を引いたバイト数にデータベース・ブロック・サイズを掛けた値です。
114	BFILE	データベース外に保存された大きなバイナリ・ファイルへの参照を格納。データベース・サーバー上に存在する外部 LOB へのバイト・ストリーム I/O アクセスを可能にします。最大サイズは 4GB です。

関連項目:

組み込みデータ型の詳細は、『Oracle Database SQL言語リファレンス』を参照してください。

Oracleが提供するデータ型

この項では、Oracleが提供するデータ型の構文を示します。

any_types

```
{ SYS.AnyData | SYS.AnyType | SYS.AnyDataSet }
```

spatial_types

{ SDO_Geometry | SDO_Topo_Geometry | SDO_GeoRaster }

XML_types

{ XMLType | URIType }

Oracleデータ型への変換

表とクラスタを作成するSQL文では、ANSIデータ型、およびIBM社の製品SQL/DSとDB2のデータ型も使用できます。Oracleでは、Oracleデータ型の名前と異なるANSIまたはIBMのデータ型の名前を認識し、列のデータ型の名前として記録します。その後、次の表に示す変換に基づいて、Oracleのデータ型で列データを格納します。

表6-2 Oracleデータ型に変換されるANSIデータ型

ANSI SQLデータ型	Oracleデータ型
CHARACTER(n) CHAR(n)	CHAR(n)
CHARACTER VARYING(n) CHAR VARYING(n)	VARCHAR2(n)
NATIONAL CHARACTER(n) NATIONAL CHAR(n) NCHAR(n)	NCHAR(n)
NATIONAL CHARACTER VARYING(n)、 NATIONAL CHAR VARYING(n)、 NCHAR VARYING(n)	NVARCHAR2(n)
NUMERIC[(p, s)] DECIMAL[(p, s)](ノート 1)	NUMBER(p, s)
INTEGER INT SMALLINT	NUMBER(38)
FLOAT (ノート 2)	FLOAT(126) FLOAT(126)

ANSI SQLデータ型	Oracleデータ型
DOUBLE PRECISION (ノート 3)	FLOAT(63)
REAL (ノート 4)	

ノート:

1. NUMERICデータ型およびDECIMALデータ型では、固定小数点数のみを指定できます。これらのデータ型では、位取りsのデフォルトは0です。
2. FLOATデータ型は、2進精度bを持つ浮動小数点数です。このデータ型のデフォルト精度は、126桁の2進精度または38桁の10進精度です。
3. DOUBLE PRECISIONデータ型は126桁の2進精度を持つ浮動小数点数です。
4. REALデータ型は63桁の2進精度(18桁の10進精度)を持つ浮動小数点数です。

次のSQL/DSとDB2のデータ型には、対応するOracleデータ型がありません。次のデータ型を持つ列は定義しないでください。

- GRAPHIC
- LONG VARCHAR
- VARCHAR
- TIME

データ型がTIMEであるデータは、Oracleの日時データとしても表現できます。

関連項目:

データ型の詳細は、『Oracle Database SQL言語リファレンス』を参照してください。

7 書式モデル

この章では、文字列に格納される日時データおよび名前データの書式モデルについて説明します。

この章の内容は次のとおりです。

- [書式モデルの概要](#)
- [数値書式モデル](#)
- [日時書式モデル](#)

書式モデルの概要

書式モデルは、文字列に格納されるDATETIMEデータやNUMBERデータの書式を記述する文字リテラルです。文字列を日時または数値に変換する場合、書式モデルは文字列の変換方法を指定します。

関連項目:

書式モデルの詳細は、『Oracle Database SQL言語リファレンス』を参照してください。

数値書式モデル

次の場所で数値書式モデルを使用できます。

- NUMBERデータ型の値をVARCHAR2データ型の値に変換するTO_CHARファンクション
- CHARまたはVARCHAR2データ型の値をNUMBERデータ型に変換するTO_NUMBERファンクション

数値書式要素

数値書式モデルは、1つ以上の数値書式要素で構成されます。次の表に、数値書式モデルの要素を示します。

表7-1 数値書式の要素

要素	例	説明
, (カンマ)	9,999	指定した位置にカンマを戻します。1つの数値書式モデルに複数のカンマを指定できます。 制限: <ul style="list-style-type: none">● 数値書式モデルは、カンマ要素で始めることはできません。● 数値書式モデルでは、カンマを小数点文字やピリオドの右側に指定できません。
. (ピリオド)	99.99	指定した位置に小数点(ピリオド(.))を戻します。 制限事項: 1つの数値書式モデルには1つのピリオドのみ指定できます。

要素	例	説明
\$	\$9999	値の前にドル記号を付けて戻します。
0	0999 9990	先行 0(ゼロ)を戻します。 後続 0(ゼロ)を戻します。
9	9999	指定された桁数の値を戻します。正の場合は先行空白が含まれ、負の場合は先行マイナスが含まれます。先行 0(ゼロ)は空白ですが、0(ゼロ)値の場合は固定小数点数の整数部分に 0(ゼロ)を戻します。
B	B9999	整数部が 0(ゼロ)の場合、書式モデル内の 0(ゼロ)にかかわらず、固定小数点数の整数部に対して空白を戻します。
C	C999	指定した位置に ISO 通貨記号(NLS_ISO_CURRENCY パラメータの現在の値)を戻します。
D	99D99	指定した位置に小数点文字(NLS_NUMERIC_CHARACTER パラメータの現在の値)を戻します。デフォルトはピリオド(.)です。 制限事項: 1 つの数値書式モデルには 1 つの小数点文字のみ指定できます。
EEEE	9.9EEEE	科学表記法で値を戻します。
G	9G999	指定した位置に桁区切り(NLS_NUMERIC_CHARACTER パラメータの現在の値)を戻します。単一の数値書式モデルに複数の桁区切りを指定できます。 制限事項: 数値書式モデルにおいて、桁区切りは小数点文字やピリオドの右側に指定できません。
L	L999	指定した位置にローカル通貨記号(NLS_CURRENCY パラメータの現在の値)を戻します。
MI	9999MI	負の値の後に負の符号(-)を戻します。 正の値の後に空白を戻します。 制限事項: 書式要素 MI は、数値書式モデルの最後の位置にのみ指定できます。
PR	9999PR	山カッコ<>の中に負の値を戻します。 正の値の前後に空白を付けて戻します。

要素	例	説明
		制限事項: 書式要素 PR は、数値書式モデルの最後の位置にのみ指定できます。
RN	RN	大文字のローマ数字で値を戻します。
rn	rn	小文字のローマ数字で値を戻します。 値は 1 から 3999 の整数となります。
S	S9999	負の値の前に負の符号(-)を戻します。
	9999S	正の値の前に正の符号(+)を戻します。 負の値の後に負の符号(-)を戻します。 正の値の後に正の符号(+)を戻します。 制限事項: 書式要素 S は、数値書式モデルの最初または最後の位置にのみ指定できません。
TM	TM	テキストの最小数値書式モデルは、文字の最小数を(10 進出力で)戻します。この要素の大文字と小文字は区別されません。 デフォルトは TM9 で、出力が 64 文字を超えないかぎり、固定表記法で数値を戻します。戻り値が 64 文字を超える場合、Oracle Database は自動的に科学表記法で文字数を戻します。 制限: <ul style="list-style-type: none"> ● この要素の前に他の要素を指定することはできません。 ● この要素の後には、9、E(1 つ)または e(1 つ)のみを指定できます。次の文は、エラーを戻します。 SELECT TO_CHAR(1234, 'TM9e') FROM DUAL;
U	U9999	指定した位置にユーロ(または他の)第 2 通貨記号(NLS_DUAL_CURRENCY パラメータの現在の値によって決定される)を戻します。
V	999V99	値を 10 の ⁿ 乗にして戻します(必要に応じて数値を丸めます)。ここで、n は V の後の 9 の数です。
X	XXXX	指定の桁数の 16 進数値を戻します。指定した値が整数でない場合、Oracle Database

要素	例	説明
	XXXX	はその値を整数に丸めます。 制限: <ul style="list-style-type: none"> ● この要素は、正の値または 0(ゼロ)のみを受け入れます。負の値の場合、エラーを戻します。 ● この要素の前には、0(先行 0(ゼロ)を戻す)または FM のみを指定できます。その他の要素の場合、エラーを戻します。0(ゼロ)または FM を X と同時に指定しないと、戻り値の前に常に空白が 1 つ追加されます。FM 書式モデル修飾子の詳細は、『Oracle Database SQL 言語リファレンス』を参照してください。

関連項目:

数値書式モデルの詳細は、『Oracle Database SQL言語リファレンス』を参照してください。

日時書式モデル

次の場所で日時書式モデルを使用できます。

- デフォルト日時書式以外の文字列をDATETIME値に変換するTO_CHAR、TO_DATE、TO_TIMESTAMP、TO_TIMESTAMP_TZ、TO_YMINTERVALおよびTO_DSINTERVAL日時ファンクション
- デフォルト日時書式以外の書式のDATETIME値を文字列に変換するTO_CHARファンクション

日時書式要素

日時書式モデルは、1つ以上の日時書式要素で構成されます。次の表に、日時書式モデルの要素を示します。

表7-2 日時書式要素

要素	TO_*日時ファンクションで指定できるか	説明
- / , . ; : "text"	Yes	句読点と引用符付きテキストは、結果で複製されます。
AD A.D.	Yes	ピリオド付き/なしで西暦を示します。
AM A.M.	Yes	ピリオド付き/なしで子午線インジケータを示します。
BC B.C.	Yes	ピリオド付き/なしで紀元前を示します。

要素	TO_*日時ファンクシ ンで指定できるか	説明
CC SCC	No	世紀。 <ul style="list-style-type: none"> ● 4桁の年の下2桁が01から99(01および99を含む)の場合、世紀はその年の上2桁に1を加えた値になります。 ● 4桁の年の下2桁が00の場合、世紀はその年の上2桁と等しい値になります。 <p>たとえば、2002は21を戻し、2000は20を戻します。</p>
D	Yes	週における日(1から7)。この要素はセッションのNLS地域に依存します。
DAY	Yes	曜日。
DD	Yes	月における日(1から31)。
DDD	Yes	年における日(1から366)。
DL	Yes	長い日付書式で値を戻します。これは、NLS_DATE_FORMATパラメータの現在の値によって決定される、Oracle DatabaseのDATEの書式を拡張したものです。日付の構成要素(曜日や月など)の表示が、NLS_TERRITORYパラメータおよびNLS_LANGUAGEパラメータによって決まるようにします。たとえば、AMERICAN_AMERICAロケールでは、これは書式'fmDay, Month dd, yyyy'を指定することと同じです。GERMAN_GERMANYロケールでは、書式'fmDay, dd.Month yyyy'を指定することと同じです。 制限: この書式は、TS要素でのみ、空白で区切って指定可能です。
DS	Yes	短い日付書式の値を戻します。日付の構成要素(曜日や月など)の表示が、NLS_TERRITORYパラメータおよびNLS_LANGUAGEパラメータによって決まるようにします。たとえば、AMERICAN_AMERICAロケールでは、'MM/DD/RRRR'形式で指定したのと同じです。ENGLISH_UNITED_KINGDOMロケールでは、'DD/MM/RRRR'形式で指定したのと同じです。 制限: この書式は、TS要素でのみ、空白で区切って指定可能です。
DY	Yes	曜日の省略形。

要素	TO_*日時ファンクショ ンで指定できるか	説明
E	Yes	年号(日本の元号暦、台湾暦およびタイ仏教暦)の省略形。
EE	Yes	年号(日本の元号暦、台湾暦およびタイ仏教暦)の完全形。
FF [1..9]	Yes	<p>小数部。基数は出力されません。基数文字の追加には、X 書式要素を使用します。FF の後に 1 から 9 の数字を設定して、戻される日時値のミリ秒部分の桁数を指定します。数字を指定しない場合は、日時データ型に指定された精度またはデータ型のデフォルトの精度が使用されます。タイムスタンプ書式および期間書式では有効ですが、DATE 書式では有効ではありません。</p> <p>例: 'HH:MI:SS.FF'</p> <pre>SELECT TO_CHAR(SYSTIMESTAMP, 'SS.FF3') from dual;</pre>
FM	Yes	<p>先行および後続する空白を持たない値を返します。</p> <p>FM 書式モデル修飾子の詳細は、『Oracle Database SQL 言語リファレンス』も参照</p>
FX	Yes	<p>文字データと書式モデルの完全一致を必要とします。</p> <p>FX 書式モデル修飾子の詳細は、『Oracle Database SQL 言語リファレンス』も参照</p>
HH HH12	Yes	1 日の時間(1 から 12)。
HH24	Yes	1 日の時間(0 から 23)。
IW	No	ISO 規格に基づく、年における週(1 から 52 または 1 から 53)。
IYY IY I	No	ISO 年の最後の 3 桁、2 桁または 1 桁。
IYYY	No	ISO 規格に基づく 4 桁の年。
J	Yes	ユリウス日(紀元前 4712 年 1 月 1 日からの経過日数)。J で指定する数字は、整数である必要があります。
MI	Yes	分(0 から 59)。

要素	TO_*日時ファンクショ ンで指定できるか	説明
MM	Yes	月(01 から 12。1 月は 01)。
MON	Yes	月の名前の省略形。
MONTH	Yes	月の名前。
PM P.M.	Yes	ピリオド付き/なしで子午線インジケータを示します。
Q	No	年の四半期(1、2、3、4。1 月から 3 月は 1)。
RM	Yes	ローマ数字による月(I から XII。1 月は I)。
RR	Yes	2 桁のみを使用して、21 世紀に 20 世紀の日付を格納できます。 RR 日時書式要素の詳細は、 『Oracle Database SQL 言語リファレンス』 も参照
RRRR	Yes	年。4 桁または 2 桁で入力できます。2 桁の場合、RR の場合と同様の結果が戻ります。この機能を使用しない場合は、4 桁の年を入力します。
SS	Yes	秒(0 から 59)。
SSSSS	Yes	午前 0 時から経過した秒(0 から 86399)。
TS	Yes	短い時間書式で値を戻します。時間の構成要素(時間や分など)の表示が、NLS_TERRITORY 初期化パラメータおよび NLS_LANGUAGE 初期化パラメータによって決まるようにします。 制限: この書式は、DL または DS 要素でのみ、空白で区切って指定可能です。
TZD	Yes	夏時間の情報。TZD の値は、夏時間の情報を持つタイムゾーン文字列の省略形です。TZR で指定した地域と対応している必要があります。タイムスタンプ書式および期間書式では有効ですが、DATE 書式では有効ではありません。 例: PST(アメリカ太平洋標準時)、PDT(アメリカ太平洋夏時間)。
TZH	Yes	タイムゾーンの時間。(TZM 書式要素を参照。)タイムスタンプ書式および期間書

要素	TO_*日時ファンクシ ンで指定できるか	説明
		式では有効ですが、DATE 書式では有効ではありません。 例: 'HH:MI:SS.FFTZH:TZM'
TZM	Yes	タイムゾーンの分。(TZH 書式要素を参照。)タイムスタンプ書式および期間書式では有効ですが、DATE 書式では有効ではありません。 例: 'HH:MI:SS.FFTZH:TZM'
TZR	Yes	タイムゾーン地域の情報。値は、データベースでサポートされるタイムゾーン地域である必要があります。タイムスタンプ書式および期間書式では有効ですが、DATE 書式では有効ではありません。 例: US/Pacific
WW	No	年における週(1 から 53)。第 1 週はその年の 1 月 1 日で始まり、1 月 7 日で終了します。
W	No	月における週(1 から 5)。第 1 週はその月の 1 日で始まり、7 日で終了します。
X	Yes	ローカル基数文字。 例: 'HH:MI:SSXFF'
Y,YYY	Yes	指定の位置にカンマを付けた年。
YEAR SYEAR	No	スペルアウトした年。S の場合、紀元前の日付の先頭にマイナス記号(-)が付加されます。
YYYY SYYYY	Yes	4 桁の年。S の場合、紀元前の日付の先頭にマイナス記号が付加されます。
YYY YY Y	Yes	年の最後の 3 桁、2 桁または 1 桁。

関連項目:

日時書式モデルの詳細は、『Oracle Database SQL言語リファレンス』を参照してください。

A SQL*Plusコマンド

この付録では、SQL*Plusコマンドについて説明します。

この付録の内容は次のとおりです。

- [SQL*Plusコマンド](#)

SQL*Plusコマンド

SQL*Plusは、Oracle RDBMSへのアクセスを提供するコマンドライン・ツールです。SQL*Plusを使用すると、次のことができます。

- SQL*Plusコマンドの入力によるSQL*Plus環境の構成
- Oracle Databaseの起動および停止
- Oracle Databaseへの接続
- SQLコマンドおよびPL/SQLブロックの入力および実行
- 問合せ結果の書式設定および出力

SQL*Plusは数種類のプラットフォームで使用可能です。

[表A-1](#)に示すコマンドは、コマンドライン・インタフェースで使用可能なSQL*Plusコマンドです。なお、ここでは、一部のコマンドまたはコマンド・パラメータのみを示しています。

関連項目:

- [SQL*Plusクイック・リファレンス](#)
- [SQL*Plusユーザーズ・ガイドおよびリファレンス](#)

表A-1 基本的なSQL*Plusコマンド

データベース操作	SQL*Plusコマンド
SQL*Plus へのログイン	SQLPLUS [[{username[/password]][@connect_identifier] / } [AS {SYSASM SYSBACKUP SYSDBA SYSDG SYSOPER SYSKM}] [edition=value]] /NOLOG]
SQL*Plus で使用可能なヘルプ・トピックの表示	HELP [INDEX topic]
ホスト・コマンドの実行	HOST [command]
SQL*Plus のシステム変数および環境設定の表示	SHOW { ALL ERRORS USER system_variable [, system_variable] ...}

データベース操作	SQL*Plusコマンド
SQL*Plus のシステム変数および環境設定の変更	<pre>SET system_variable value</pre>
データベースの起動	<pre>STARTUP { db_options cdb_options upgrade_options }</pre> <p>db_options の構文は、次のとおりです。</p> <pre>[FORCE] [RESTRICT] [PFILE=filename] [QUIET] [MOUNT [dbname] [OPEN [open_db_options] [dbname]] NOMOUNT]</pre> <p>open_db_options の構文は、次のとおりです。</p> <pre>READ {ONLY WRITE [RECOVER]} RECOVER</pre> <p>cdb_options の構文は、次のとおりです。</p> <pre>root_connection_options pdb_connection_options</pre> <p>root_connection_options の構文は、次のとおりです。</p> <pre>PLUGGABLE DATABASE pdbname [FORCE] [RESTRICT] [OPEN {open_pdb_options}]</pre> <p>pdb_connection_options の構文は、次のとおりです。</p> <pre>[FORCE] [RESTRICT] [OPEN {open_pdb_options}]</pre> <p>open_pdb_options の構文は、次のとおりです。</p> <pre>READ WRITE READ ONLY</pre> <p>upgrade_options の構文は、次のとおりです。</p> <pre>[PFILE=filename] {UPGRADE DOWNGRADE} [QUIET]</pre>
データベースへの接続	<pre>CONNECT [{username[/password] [@connect_identifier] / proxy_user [username] [/password] [@connect_identifier]}] [AS {SYSASM SYSBACKUP SYSDBA SYSDBG SYSOPER SYSKM}] [edition=value]]</pre> <p>ノート：太字で示されている大カッコは構文の一部です。オプションを表すものではありません。</p>
表、ビュー、シノニムに対する列定義、またはファンクションやプロシージャの仕様の表示	<pre>DESCRIBE [schema.] object</pre>

データベース操作	SQL*Plusコマンド
SQL バッファまたはファイルの内容の編集	EDIT [filename [.ext]]
ファイルの取得、および SQL バッファへのその内容のロード	GET filename [.ext] [LIST NOLLIST]
ファイルへの SQL バッファの内容の保存	SAVE filename [.ext] [CREATE REPLACE APPEND]
SQL バッファの内容の表示	LIST [n n m n LAST]
SQL バッファの内容の削除	DEL [n n m n LAST]
SQL バッファ内のカレント行の後に新しい行を追加	INPUT [text]
SQL バッファ内のカレント行の最後にテキストを追加	APPEND text
SQL バッファ内のカレント行の最初に一致したテキスト文字列を検索および置換	CHANGE sepchar old [sepchar [new [sepchar]]] sepchar には、英数字以外の ASCII 文字(「/」や「!」など)を指定できません。
問合せ結果のファイルへの記録。ファイルの内容をデフォルトのプリンタへ送信(オプション)	SPOOL [filename[.ext] [CREATE REPLACE APPEND] OFF OUT]
ファイルに格納されている SQL*Plus 文の実行	@ { url filename [.ext] } [arg ...] START { url filename [.ext] } [arg ...] ファイル名の拡張子が.sql の場合は、ext を省略できます。
SQL バッファに格納されているコマンドの実行	/
SQL バッファに格納されているコマンドの表示および実行	RUN
単一の PL/SQL 文の実行またはストアド・プロシージャの実行	EXECUTE statement

データベース操作	SQL*Plusコマンド
データベースからの切断	DISCONNECT
データベースの停止	SHUTDOWN [ABORT IMMEDIATE NORMAL TRANSACTIONAL [LOCAL]]
SQL*Plus からのログアウト	{ EXIT QUIT } [SUCCESS FAILURE WARNING n variable :BindVariable] [COMMIT ROLLBACK]

索引

記号 [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

記号

- / (スラッシュ) SQL*Plusコマンド [A.1](#)
 - @ (アットマーク) SQL*Plusコマンド [A.1](#)
-

A

- ABS関数 [2.1](#)
- ACOS関数 [2.1](#)
- action_audit_clause [5.1](#)
- activate_standby_db_clause [5.1](#)
- add_binding_clause [5.1](#)
- add_column_clause [5.1](#)
- add_disk_clause [5.1](#)
- add_filegroup_clause [5.1](#)
- add_hash_index_partition [5.1](#)
- add_hash_partition_clause [5.1](#)
- add_hash_subpartition [5.1](#)
- add_list_partition_clause [5.1](#)
- add_list_subpartition [5.1](#)
- add_logfile_clauses [5.1](#)
- ADD_MONTHS関数 [2.1](#)
- add_mv_log_column_clause [5.1](#)
- add_overflow_clause [5.1](#)
- add_period_clause [5.1](#)
- add_range_partition_clause [5.1](#)
- add_range_subpartition [5.1](#)
- add_system_partition_clause [5.1](#)
- add_table_partition [5.1](#)
- add_update_secret [5.1](#)
- add_volume_clause [5.1](#)
- ADMINISTER KEY MANAGEMENT文 [1.1](#)
- advanced_index_compression [5.1](#)
- 集計関数 [2.1](#)
- alias_file_name [5.1](#)
- all_clause [5.1](#)
- allocate_extent_clause [5.1](#)
- allow_disallow_clustering [5.1](#)

- [alter_automatic_partitioning 5.1](#)
- [alter_datafile_clause 5.1](#)
- [alter_external_table 5.1](#)
- [alter_index_partitioning 5.1](#)
- [alter_interval_partitioning 5.1](#)
- [alter_iot_clauses 5.1](#)
- [alter_keystore_password 5.1](#)
- [alter_mapping_table_clauses 5.1](#)
- [alter_mv_refresh 5.1](#)
- [alter_overflow_clause 5.1](#)
- [alter_query_rewrite_clause 5.1](#)
- [alter_session_set_clause 5.1](#)
- [alter_system_reset_clause 5.1](#)
- [alter_system_set_clause 5.1](#)
- [alter_table_partitioning 5.1](#)
- [alter_table_properties 5.1](#)
- [alter_tablespace_attrs 5.1](#)
- [alter_tablespace_encryption 5.1](#)
- [alter_tempfile_clause 5.1](#)
- [alter_varray_col_properties 5.1](#)
- [alter_XMLSchema_clause 5.1](#)
- [alter_zonemap_attributes 5.1](#)
- [ALTER ANALYTIC VIEW文 1.1](#)
- [ALTER ATTRIBUTE DIMENSION文 1.1](#)
- [ALTER AUDIT POLICY文 1.1](#)
- [ALTER CLUSTER文 1.1](#)
- [ALTER DATABASE LINK文 1.1](#)
- [ALTER DATABASE文 1.1](#)
- [ALTER DIMENSION文 1.1](#)
- [ALTER DISKGROUP文 1.1](#)
- [ALTER FLASHBACK ARCHIVE文 1.1](#)
- [ALTER FUNCTION文 1.1](#)
- [ALTER HIERARCHY文 1.1](#)
- [ALTER INDEX文 1.1](#)
- [ALTER INDEXTYPE文 1.1](#)
- [ALTER INMEMORY JOIN GROUP文 1.1](#)
- [ALTER JAVA文 1.1](#)
- [ALTER LIBRARY文 1.1](#)
- [ALTER LOCKDOWN PROFILE文 1.1](#)
- [ALTER MATERIALIZED VIEW LOG文 1.1](#)
- [ALTER MATERIALIZED VIEW文 1.1](#)
- [ALTER MATERIALIZED ZONEMAP文 1.1](#)
- [alternate_key_clause 5.1](#)

- ALTER OPERATOR文 [1.1](#)
- ALTER OUTLINE文 [1.1](#)
- ALTER PACKAGE文 [1.1](#)
- ALTER PLUGGABLE DATABASE文 [1.1](#)
- ALTER PROCEDURE文 [1.1](#)
- ALTER PROFILE文 [1.1](#)
- ALTER RESOURCE COST文 [1.1](#)
- ALTER ROLE文 [1.1](#)
- ALTER ROLLBACK SEGMENT文 [1.1](#)
- ALTER SEQUENCE文 [1.1](#)
- ALTER SESSION文 [1.1](#)
- ALTER SYNONYM文 [1.1](#)
- ALTER SYSTEM文 [1.1](#)
- ALTER TABLESPACE SET文 [1.1](#)
- ALTER TABLESPACE文 [1.1](#)
- ALTER TABLE文 [1.1](#)
- ALTER TRIGGER文 [1.1](#)
- ALTER TYPE文 [1.1](#)
- ALTER USER文 [1.1](#)
- ALTER VIEW文 [1.1](#)
- ANSI(米国規格協会)
 - Oracleデータ型への変換 [6.4](#)
- analytic_clause [5.1](#)
- 分析関数 [2.1](#)
- ANALYZE文 [1.1](#)
- ANSIでサポートされるデータ型 [6.1](#)
- any_types [6.3](#)
- APPENDCHILDXML関数 [2.1](#)
- APPEND SQL*Plusコマンド [A.1](#)
- application_clauses [5.1](#)
- APPROX_COUNT_DISTINCT_AGG関数 [2.1](#)
- APPROX_COUNT_DISTINCT_DETAIL関数 [2.1](#)
- APPROX_COUNT_DISTINCT関数 [2.1](#)
- APPROX_MEDIAN関数 [2.1](#)
- APPROX_PERCENTILE_AGG関数 [2.1](#)
- APPROX_PERCENTILE_DETAIL関数 [2.1](#)
- APPROX_PERCENTILE関数 [2.1](#)
- archive_log_clause [5.1](#)
- array_DML_clause [5.1](#)
- array_step [5.1](#)
- ASCII関数 [2.1](#)
- ASCIISTR関数 [2.1](#)
- ASIN関数 [2.1](#)

- [ASM_filename 5.1](#)
 - [ASSOCIATE STATISTICS文 1.1](#)
 - [ATAN2ファンクション 2.1](#)
 - [ATANファンクション 2.1](#)
 - [attr_dim_attributes_clause 5.1](#)
 - [attr_dim_level_clause 5.1](#)
 - [attr_dim_using_clause 5.1](#)
 - [attribute_clause 5.1](#)
 - [attribute_clustering_clause 5.1](#)
 - [attributes_clause 5.1](#)
 - [audit_operation_clause 5.1](#)
 - [audit_schema_object_clause 5.1](#)
 - [AUDIT\(従来型監査\)文 1.1](#)
 - [AUDIT\(統合監査\)文 1.1](#)
 - [auditing_by_clause 5.1](#)
 - [auditing_on_clause 5.1](#)
 - [autoextend_clause 5.1](#)
 - [av_meas_expression 5.1](#)
 - [av_measure 5.1](#)
 - [av_simple_expression 5.1](#)
 - [AVGファンクション 2.1](#)
-

B

- [backup_keystore 5.1](#)
 - [base_measure_clause 5.1](#)
 - [BETWEEN条件 4.1](#)
 - [BFILENAMEファンクション 2.1](#)
 - [BIN_TO_NUMファンクション 2.1](#)
 - [binding_clause 5.1](#)
 - [BITANDファンクション 2.1](#)
 - [bitmap_join_index_clause 5.1](#)
 - [build_clause 5.1](#)
 - [組込みデータ型 6.1, 6.2](#)
 - [by_users_with_roles 5.1](#)
-

C

- [cache_clause 5.1](#)
- [cache_specification 5.1](#)
- [calc_meas_order_by_clause 5.1](#)
- [calc_measure_clause 5.1](#)

- 計算済メジャー式 [3.1](#)
- CALL文 [1.1](#)
- CARDINALITYファンクション [2.1](#)
- CASE式 [3.1](#)
- CASTファンクション [2.1](#)
- CEILファンクション [2.1](#)
- cell_assignment [5.1](#)
- cell_reference_options [5.1](#)
- CHANGE SQL*Plusコマンド [A.1](#)
- character_datatypes [6.2](#)
- character_set_clause [5.1](#)
- CHARTOROWIDファンクション [2.1](#)
- check_datafiles_clause [5.1](#)
- check_diskgroup_clause [5.1](#)
- checkpoint_clause [5.1](#)
- CHRファンクション [2.1](#)
- classification_clause [5.1](#)
- clause_options [5.1](#)
- clear_free_space_clause [5.1](#)
- close_keystore [5.1](#)
- cluster_clause [5.1](#)
- CLUSTER_DETAILS(分析)ファンクション [2.1](#)
- CLUSTER_DETAILSファンクション [2.1](#)
- CLUSTER_DISTANCE(分析)ファンクション [2.1](#)
- CLUSTER_DISTANCEファンクション [2.1](#)
- CLUSTER_ID(分析)ファンクション [2.1](#)
- CLUSTER_IDファンクション [2.1](#)
- cluster_index_clause [5.1](#)
- CLUSTER_PROBABILITY(分析)ファンクション [2.1](#)
- CLUSTER_PROBABILITYファンクション [2.1](#)
- cluster_range_partitions [5.1](#)
- CLUSTER_SET(分析)ファンクション [2.1](#)
- CLUSTER_SETファンクション [2.1](#)
- clustering_column_group [5.1](#)
- clustering_columns [5.1](#)
- clustering_join [5.1](#)
- clustering_when [5.1](#)
- coalesce_index_partition [5.1](#)
- coalesce_table_partition [5.1](#)
- coalesce_table_subpartition [5.1](#)
- COALESCEファンクション [2.1](#)
- COLLATIONファンクション [2.1](#)
- COLLECTファンクション [2.1](#)

- [column_association 5.1](#)
- [column_clauses 5.1](#)
- [column_definition 5.1](#)
- [column_properties 5.1](#)
- [列式 3.1](#)
- [COMMENT文 1.1](#)
- [commit_switchover_clause 5.1](#)
- [COMMIT文 1.1](#)
- [component_actions 5.1](#)
- [COMPOSEファンクション 2.1](#)
- [composite_hash_partitions 5.1](#)
- [composite_list_partitions 5.1](#)
- [composite_range_partitions 5.1](#)
- [複合条件 4.1](#)
- [複合式 3.1](#)
- [CON_DBID_TO_IDファンクション 2.1](#)
- [CON_GUID_TO_IDファンクション 2.1](#)
- [CON_NAME_TO_IDファンクション 2.1](#)
- [CON_UID_TO_IDファンクション 2.1](#)
- [CONCATファンクション 2.1](#)
- [conditional_insert_clause 5.1](#)
- [条件 4](#)
 - 「SQL条件」も参照 [4](#)
- [CONNECT SQL*Plusコマンド A.1](#)
- [consistent_hash_partitions 5.1](#)
- [consistent_hash_with_subpartitions 5.1](#)
- [constraint 5.1](#)
- [constraint_clauses 5.1](#)
- [constraint_state 5.1](#)
- [container_data_clause 5.1](#)
- [containers_clause 5.1](#)
- [context_clause 5.1](#)
- [controlfile_clauses 5.1](#)
- [convert_database_clause 5.1](#)
- [convert_redundancy_clause 5.1](#)
- [CONVERTファンクション 2.1](#)
- [Oracleデータ型への変換 6.4](#)
- [CORR_Kファンクション 2.1](#)
- [CORR_Sファンクション 2.1](#)
- [CORRファンクション 2.1](#)
- [COSファンクション 2.1](#)
- [COSHファンクション 2.1](#)
- [cost_matrix_clause 5.1](#)

- COUNT関数 [2.1](#)
- COVAR_POP関数 [2.1](#)
- COVAR_SAMP関数 [2.1](#)
- create_datafile_clause [5.1](#)
- create_file_dest_clause [5.1](#)
- create_key [5.1](#)
- create_keystore [5.1](#)
- create_mv_refresh [5.1](#)
- create_pdb_clone [5.1](#)
- create_pdb_from_seed [5.1](#)
- create_pdb_from_xml [5.1](#)
- create_zonemap_as_subquery [5.1](#)
- create_zonemap_on_table [5.1](#)
- CREATE ANALYTIC VIEW文 [1.1](#)
- CREATE ATTRIBUTE DIMENSION文 [1.1](#)
- CREATE AUDIT POLICY文 [1.1](#)
- CREATE CLUSTER文 [1.1](#)
- CREATE CONTEXT文 [1.1](#)
- CREATE CONTROLFILE文 [1.1](#)
- CREATE DATABASE LINK文 [1.1](#)
- CREATE DATABASE文 [1.1](#)
- CREATE DIMENSION文 [1.1](#)
- CREATE DIRECTORY文 [1.1](#)
- CREATE DISKGROUP文 [1.1](#)
- CREATE EDITION文 [1.1](#)
- CREATE FLASHBACK ARCHIVE文 [1.1](#)
- CREATE FUNCTION文 [1.1](#)
- CREATE HIERARCHY文 [1.1](#)
- CREATE INDEX文 [1.1](#)
- CREATE INDEXTYPE文 [1.1](#)
- CREATE INMEMORY JOIN GROUP文 [1.1](#)
- CREATE JAVA文 [1.1](#)
- CREATE LIBRARY文 [1.1](#)
- CREATE LOCKDOWN PROFILE文 [1.1](#)
- CREATE MATERIALIZED VIEW LOG文 [1.1](#)
- CREATE MATERIALIZED VIEW文 [1.1](#)
- CREATE MATERIALIZED ZONEMAP文 [1.1](#)
- CREATE OPERATOR文 [1.1](#)
- CREATE OUTLINE文 [1.1](#)
- CREATE PACKAGE BODY文 [1.1](#)
- CREATE PACKAGE文 [1.1](#)
- CREATE PFILE文 [1.1](#)
- CREATE PLUGGABLE DATABASE文 [1.1](#)

- CREATE PROCEDURE文 [1.1](#)
- CREATE PROFILE文 [1.1](#)
- CREATE RESTORE POINT文 [1.1](#)
- CREATE ROLE文 [1.1](#)
- CREATE ROLLBACK SEGMENT文 [1.1](#)
- CREATE SCHEMA文 [1.1](#)
- CREATE SEQUENCE文 [1.1](#)
- CREATE SPFILE文 [1.1](#)
- CREATE SYNONYM文 [1.1](#)
- CREATE TABLESPACE SET文 [1.1](#)
- CREATE TABLESPACE文 [1.1](#)
- CREATE TABLE文 [1.1](#)
- CREATE TRIGGER文 [1.1](#)
- CREATE TYPE BODY文 [1.1](#)
- CREATE TYPE文 [1.1](#)
- CREATE USER文 [1.1](#)
- CREATE VIEW文 [1.1](#)
- cross_outer_apply_clause [5.1](#)
- CUBE_TABLEファンクション [2.1](#)
- CUME_DIST(集計)ファンクション [2.1](#)
- CUME_DIST(分析)ファンクション [2.1](#)
- 通貨
 - 桁区切り [7.1.1.1](#)
- 通貨記号
 - ISO [7.1.1.1](#)
 - ローカル [7.1.1.1](#)
 - union [7.1.1.1](#)
- CURRENT_DATEファンクション [2.1](#)
- CURRENT_TIMESTAMPファンクション [2.1](#)
- CURSOR式 [3.1](#)
- CVファンクション [2.1](#)
- cycle_clause [5.1](#)

D

- database_file_clauses [5.1](#)
- database_logging_clauses [5.1](#)
- datafile_tempfile_clauses [5.1](#)
- datafile_tempfile_spec [5.1](#)
- DATAOBJ_TO_MAT_PARTITIONファンクション [2.1](#)
- DATAOBJ_TO_PARTITIONファンクション [2.1](#)
- データ型
 - ANSIのサポート [6.1](#)

- Oracleへの変換 [6.4](#)
- Oracle組込み [6.1](#), [6.2](#)
- Oracle提供 [6.1](#), [6.3](#)
- 概要 [6.1](#)
- ユーザー定義 [6.1](#)
- 日付書式モデル [7.1.2](#), [7.1.2.1](#)
 - long [7.1.2.1](#)
 - short [7.1.2.1](#)
- datetime_datatypes [6.2](#)
- 日時式 [3.1](#)
- db_user_proxy_clauses [5.1](#)
- DB2のデータ型
 - 制限事項 [6.4](#)
- dblink [5.1](#)
- dblink_authentication [5.1](#)
- DBTIMEZONEファンクション [2.1](#)
- deallocate_unused_clause [5.1](#)
- 小数点文字
 - 指定 [7.1.1.1](#)
- DECODEファンクション [2.1](#)
- DECOMPOSEファンクション [2.1](#)
- default_aggregate_clause [5.1](#)
- default_cost_clause [5.1](#)
- default_index_compression [5.1](#)
- default_measure_clause [5.1](#)
- default_selectivity_clause [5.1](#)
- default_settings_clauses [5.1](#)
- default_table_compression [5.1](#)
- default_tablespace [5.1](#)
- default_tablespace_params [5.1](#)
- default_temp_tablespace [5.1](#)
- deferred_segment_creation [5.1](#)
- delete_secret [5.1](#)
- DELETE文 [1.1](#)
- DEL SQL*Plusコマンド [A.1](#)
- DENSE_RANK(集計)ファンクション [2.1](#)
- DENSE_RANK(分析)ファンクション [2.1](#)
- dependent_tables_clause [5.1](#)
- DEPTHファンクション [2.1](#)
- Derefファンクション [2.1](#)
- DESCRIBE SQL*Plusコマンド [A.1](#)
- dim_by_clause [5.1](#)
- dim_key [5.1](#)

- [dim_order_clause 5.1](#)
- [dim_ref 5.1](#)
- [dimension_join_clause 5.1](#)
- [DISASSOCIATE STATISTICS文 1.1](#)
- [DISCONNECT SQL*Plusコマンド A.1](#)
- [disk_offline_clause 5.1](#)
- [disk_online_clause 5.1](#)
- [disk_region_clause 5.1](#)
- [diskgroup_alias_clauses 5.1](#)
- [diskgroup_attributes 5.1](#)
- [diskgroup_availability 5.1](#)
- [diskgroup_directory_clauses 5.1](#)
- [diskgroup_template_clauses 5.1](#)
- [diskgroup_volume_clauses 5.1](#)
- [distributed_recov_clauses 5.1](#)
- [dml_table_expression_clause 5.1](#)
- [domain_index_clause 5.1](#)
- [drop_binding_clause 5.1](#)
- [drop_column_clause 5.1](#)
- [drop_constraint_clause 5.1](#)
- [drop_disk_clause 5.1](#)
- [drop_diskgroup_file_clause 5.1](#)
- [drop_filegroup_clause 5.1](#)
- [drop_index_partition 5.1](#)
- [drop_logfile_clauses 5.1](#)
- [drop_period_clause 5.1](#)
- [drop_table_partition 5.1](#)
- [drop_table_subpartition 5.1](#)
- [DROP ANALYTIC VIEW文 1.1](#)
- [DROP ATTRIBUTE DIMENSION文 1.1](#)
- [DROP AUDIT POLICY文 1.1](#)
- [DROP CLUSTER文 1.1](#)
- [DROP CONTEXT文 1.1](#)
- [DROP DATABASE LINK文 1.1](#)
- [DROP DATABASE文 1.1](#)
- [DROP DIMENSION文 1.1](#)
- [DROP DIRECTORY文 1.1](#)
- [DROP DISKGROUP文 1.1](#)
- [DROP EDITION文 1.1](#)
- [DROP FLASHBACK ARCHIVE文 1.1](#)
- [DROP FUNCTION文 1.1](#)
- [DROP HIERARCHY文 1.1](#)
- [DROP INDEX文 1.1](#)

- DROP INDEXTYPE文 [1.1](#)
 - DROP INMEMORY JOIN GROUP文 [1.1](#)
 - DROP JAVA文 [1.1](#)
 - DROP LIBRARY文 [1.1](#)
 - DROP LOCKDOWN PROFILE文 [1.1](#)
 - DROP MATERIALIZED VIEW LOG文 [1.1](#)
 - DROP MATERIALIZED VIEW文 [1.1](#)
 - DROP MATERIALIZED ZONEMAP文 [1.1](#)
 - DROP OPERATOR文 [1.1](#)
 - DROP OUTLINE文 [1.1](#)
 - DROP PACKAGE文 [1.1](#)
 - DROP PLUGGABLE DATABASE文 [1.1](#)
 - DROP PROCEDURE文 [1.1](#)
 - DROP PROFILE文 [1.1](#)
 - DROP RESTORE POINT文 [1.1](#)
 - DROP ROLE文 [1.1](#)
 - DROP ROLLBACK SEGMENT文 [1.1](#)
 - DROP SEQUENCE文 [1.1](#)
 - DROP SYNONYM文 [1.1](#)
 - DROP TABLESPACE SET文 [1.1](#)
 - DROP TABLESPACE文 [1.1](#)
 - DROP TABLE文 [1.1](#)
 - DROP TRIGGER文 [1.1](#)
 - DROP TYPE BODY文 [1.1](#)
 - DROP TYPE文 [1.1](#)
 - DROP USER文 [1.1](#)
 - DROP VIEW文 [1.1](#)
 - TO_DSINTERVAL関数 ds_iso_format [5.1](#)
 - DUMP関数 [2.1](#)
-

E

- EDIT SQL*Plusコマンド [A.1](#)
- else_clause [5.1](#)
- EMPTY_BLOB関数 [2.1](#)
- EMPTY_CLOB関数 [2.1](#)
- enable_disable_clause [5.1](#)
- enable_disable_volume [5.1](#)
- enable_pluggable_database [5.1](#)
- encryption_spec [5.1](#)
- end_session_clauses [5.1](#)
- EQUALS_PATH条件 [4.1](#)
- error_logging_clause [5.1](#)

- [evaluation_edition_clause 5.1](#)
- [exceptions_clause 5.1](#)
- [exchange_partition_subpart 5.1](#)
- [EXECUTE SQL*Plusコマンド A.1](#)
- [EXISTS条件 4.1](#)
- [EXISTSNODEファンクション 2.1](#)
- [EXIT SQL*Plusコマンド A.1](#)
- [EXPファンクション 2.1](#)
- [EXPLAIN PLAN文 1.1](#)
- [export_keys 5.1](#)
- [expr 5.1](#)
- [expression_list 5.1](#)
- [式 3](#)
 - 「SQL式」も参照 [3](#)
- [extended_attribute_clause 5.1](#)
- [extent_management_clause 5.1](#)
- [external_part_subpart_data_props 5.1](#)
- [external_table_clause 5.1](#)
- [external_table_data_props 5.1](#)
- [EXTRACT \(日時\)ファンクション 2.1](#)
- [EXTRACT \(XML\)ファンクション 2.1](#)
- [EXTRACTVALUEファンクション 2.1](#)

F

- [failover_clause 5.1](#)
- [FEATURE_COMPAREファンクション 2.1](#)
- [FEATURE_DETAILS\(分析\)ファンクション 2.1](#)
- [FEATURE_DETAILSファンクション 2.1](#)
- [FEATURE_ID\(分析\)ファンクション 2.1](#)
- [FEATURE_IDファンクション 2.1](#)
- [FEATURE_SET\(分析\)ファンクション 2.1](#)
- [FEATURE_SETファンクション 2.1](#)
- [FEATURE_VALUE\(分析\)ファンクション 2.1](#)
- [FEATURE_VALUEファンクション 2.1](#)
- [file_name_convert 5.1](#)
- [file_owner_clause 5.1](#)
- [file_permissions_clause 5.1](#)
- [file_specification 5.1](#)
- [filegroup_clauses 5.1](#)
- [filter_condition 5.1](#)
- [FIRST_VALUEファンクション 2.1](#)
- [FIRSTファンクション 2.1](#)

- flashback_archive_clause [5.1](#)
 - flashback_archive_quota [5.1](#)
 - flashback_archive_retention [5.1](#)
 - flashback_mode_clause [5.1](#)
 - flashback_query_clause [5.1](#)
 - FLASHBACK DATABASE文 [1.1](#)
 - FLASHBACK TABLE文 [1.1](#)
 - 浮動小数点条件 [4.1](#)
 - FLOORファンクション [2.1](#)
 - following_boundary [5.1](#)
 - for_refresh_clause [5.1](#)
 - for_update_clause [5.1](#)
 - 書式モデル [7.1](#)
 - 日付書式モデル [7.1.2](#)
 - 数値書式モデル [7.1.1.1](#)
 - FROM_TZファンクション [2.1](#)
 - full_database_recovery [5.1](#)
 - fully_qualified_file_name [5.1](#)
 - function_association [5.1](#)
 - ファンクション式 [3.1](#)
 - ファンクション [2](#)
 - 「SQLファンクション」も参照 [2](#)
-

G

- general_recovery [5.1](#)
- GET SQL*Plusコマンド [A.1](#)
- global_partitioned_index [5.1](#)
- grant_object_privileges [5.1](#)
- grant_roles_to_programs [5.1](#)
- grant_system_privileges [5.1](#)
- grantee_clause [5.1](#)
- grantee_identified_by [5.1](#)
- GRANT文 [1.1](#)
- GRAPHICデータ型
 - DB2 [6.4](#)
 - SQL/DS [6.4](#)
- GREATESTファンクション [2.1](#)
- group_by_clause [5.1](#)
- GROUP_IDファンクション [2.1](#)
- グループ比較条件 [4.1](#)
- grouping_expression_list [5.1](#)
- GROUPING_IDファンクション [2.1](#)

- [grouping_sets_clause 5.1](#)
 - [GROUPINGファンクション 2.1](#)
 - グループ・セパレータ
 - [指定 7.1.1.1](#)
-

H

- [hash_partitions 5.1](#)
 - [hash_partitions_by_quantity 5.1](#)
 - [hash_subparts_by_quantity 5.1](#)
 - [heap_org_table_clause 5.1](#)
 - [HELP SQL*Plusコマンド A.1](#)
 - 16進数値
 - [返す 7.1.1.1](#)
 - [HEXTORAWファンクション 2.1](#)
 - [hier_ancestor_expression 5.1](#)
 - [hier_attr_clause 5.1](#)
 - [hier_attr_name 5.1](#)
 - [hier_attrs_clause 5.1](#)
 - [hier_lead_lag_clause 5.1](#)
 - [hier_lead_lag_expression 5.1](#)
 - [hier_navigation_expression 5.1](#)
 - [hier_parent_expression 5.1](#)
 - [hier_ref 5.1](#)
 - [hier_using_clause 5.1](#)
 - [hierarchical_query_clause 5.1](#)
 - [hierarchy_clause 5.1](#)
 - [hierarchy_ref 5.1](#)
 - [HOST SQL*Plusコマンド A.1](#)
-

I

- [identity_clause 5.1](#)
- [identity_options 5.1](#)
- [ilm_clause 5.1](#)
- [ilm_compression_policy 5.1](#)
- [ilm_inmemory_policy 5.1](#)
- [ilm_policy_clause 5.1](#)
- [ilm_tiering_policy 5.1](#)
- [ilm_time_period 5.1](#)
- [implementation_clause 5.1](#)
- [import_keys 5.1](#)

- [incomplete_file_name 5.1](#)
- [IN条件 4.1](#)
- [index_attributes 5.1](#)
- [index_compression 5.1](#)
- [index_expr 5.1](#)
- [index_org_overflow_clause 5.1](#)
- [index_org_table_clause 5.1](#)
- [index_partition_description 5.1](#)
- [index_partitioning_clause 5.1](#)
- [index_properties 5.1](#)
- [index_subpartition_clause 5.1](#)
- [indexing_clause 5.1](#)
- [individual_hash_partitions 5.1](#)
- [individual_hash_subparts 5.1](#)
- [INITCAP関数 2.1](#)
- [inline_constraint 5.1](#)
- [inline_ref_constraint 5.1](#)
- [inmemory_attributes 5.1](#)
- [inmemory_clause 5.1](#)
- [inmemory_column_clause 5.1](#)
- [inmemory_distribute 5.1](#)
- [inmemory_duplicate 5.1](#)
- [inmemory_memcompress 5.1](#)
- [inmemory_priority 5.1](#)
- [inmemory_table_clause 5.1](#)
- [inner_cross_join_clause 5.1](#)
- [INPUT SQL*Plusコマンド A.1](#)
- [insert_into_clause 5.1](#)
- [INSERT文 1.1](#)
- [instance_clauses 5.1](#)
- [instances_clause 5.1](#)
- [INSTR関数 2.1](#)
- [integer 5.1](#)
- [interval_day_to_second 5.1](#)
- [interval_year_to_month 5.1](#)
- [INTERVAL式 3.1](#)
- [into_clause 5.1](#)
- [invoker_rights_clause 5.1](#)
- [IS ANY条件 4.1](#)
- [IS A SET条件 4.1](#)
- [IS EMPTY条件 4.1](#)
- [IS JSON条件 4.1](#)
- [IS OF type条件 4.1](#)

- IS PRESENT条件 [4.1](#)
 - ITERATION_NUMBERファンクション [2.1](#)
-

J

- join_clause [5.1](#)
 - JSON_agg_returning_clause [5.1](#)
 - JSON_ARRAYAGGファンクション [2.1](#)
 - JSON_ARRAYファンクション [2.1](#)
 - JSON_column_definition [5.1](#)
 - JSON_columns_clause [5.1](#)
 - JSON_DATAGUIDEファンクション [2.1](#)
 - JSON_exists_column [5.1](#)
 - JSON_exists_on_error_clause [5.1](#)
 - JSON_EXISTS条件 [4.1](#)
 - JSON_nested_path [5.1](#)
 - JSON_OBJECTAGGファンクション [2.1](#)
 - JSON_OBJECTファンクション [2.1](#)
 - JSON_on_null_clause [5.1](#)
 - JSON_passing_clause [5.1](#)
 - JSON_query_column [5.1](#)
 - JSON_query_on_empty_clause [5.1](#)
 - JSON_query_on_error_clause [5.1](#)
 - JSON_query_return_type [5.1](#)
 - JSON_query_returning_clause [5.1](#)
 - JSON_query_wrapper_clause [5.1](#)
 - JSON_QUERYファンクション [2.1](#)
 - JSON_returning_clause [5.1](#)
 - JSON_table_on_error_clause [5.1](#)
 - JSON_TABLEファンクション [2.1](#)
 - JSON_TEXTCONTAINS条件 [4.1](#)
 - JSON_TRANSFORMファンクション [2.1](#)
 - JSON_value_column [5.1](#)
 - JSON_value_on_empty_clause [5.1](#)
 - JSON_value_on_error_clause [5.1](#)
 - JSON_value_return_type [5.1](#)
 - JSON_value_returning_clause [5.1](#)
 - JSON_VALUEファンクション [2.1](#)
 - JSONオブジェクト・アクセス式 [3.1](#)
-

K

- [key_clause 5.1](#)
 - [key_management_clauses 5.1](#)
 - [keystore_clause 5.1](#)
 - [keystore_management_clauses 5.1](#)
-

L

- [LAG関数 2.1](#)
- [large_object_datatypes 6.2](#)
- [LAST_DAY関数 2.1](#)
- [LAST_VALUE関数 2.1](#)
- [LAST関数 2.1](#)
- [lead_lag_clause 5.1](#)
- [lead_lag_expression 5.1](#)
- [lead_lag_function_name 5.1](#)
- [LEAD関数 2.1](#)
- [LEAST関数 2.1](#)
- [LENGTH関数 2.1](#)
- [level_clause 5.1](#)
- [level_hier_clause 5.1](#)
- [level_member_literal 5.1](#)
- [level_specification 5.1](#)
- [levels_clause 5.1](#)
- [LIKE条件 4.1](#)
- [list_partition_desc 5.1](#)
- [list_partitions 5.1](#)
- [list_partitionset_clause 5.1](#)
- [list_partitionset_desc 5.1](#)
- [list_subpartition_desc 5.1](#)
- [list_values 5.1](#)
- [list_values_clause 5.1](#)
- [listagg_overflow_clause 5.1](#)
- [LISTAGG関数 2.1](#)
- [LIST SQL *Plusコマンド A.1](#)
- [LN関数 2.1](#)
- [LNNVL関数 2.1](#)
- [LOB_compression_clause 5.1](#)
- [LOB_deduplicate_clause 5.1](#)
- [LOB_parameters 5.1](#)
- [LOB_partition_storage 5.1](#)
- [LOB_partitioning_storage 5.1](#)
- [LOB_retention_storage 5.1](#)
- [LOB_storage_clause 5.1](#)

- LOB_storage_parameters [5.1](#)
- local_domain_index_clause [5.1](#)
- local_partitioned_index [5.1](#)
- local_XMLIndex_clause [5.1](#)
- ロケールの独立性 [7.1.2.1](#)
- LOCALTIMESTAMP関数 [2.1](#)
- lockdown_features [5.1](#)
- lockdown_options [5.1](#)
- lockdown_statements [5.1](#)
- LOCK TABLE文 [1.1](#)
- logfile_clause [5.1](#)
- logfile_clauses [5.1](#)
- logfile_descriptor [5.1](#)
- LOG関数 [2.1](#)
- logging_clause [5.1](#)
- 論理条件 [4.1](#)
- long_and_raw_datatypes [6.2](#)
- LONG VARCHARデータ型
 - DB2 [6.4](#)
 - SQL/DS [6.4](#)
- LOWER関数 [2.1](#)
- LPAD関数 [2.1](#)
- LTRIM関数 [2.1](#)

M

- main_model [5.1](#)
- MAKE_REF関数 [2.1](#)
- managed_standby_recovery [5.1](#)
- mapping_table_clauses [5.1](#)
- materialized_view_props [5.1](#)
- MAX関数 [2.1](#)
- maximize_standby_db_clause [5.1](#)
- maxsize_clause [5.1](#)
- meas_aggregate_clause [5.1](#)
- measure [5.1](#)
- measure_ref [5.1](#)
- measures_clause [5.1](#)
- media_types [6.3](#)
- MEDIAN関数 [2.1](#)
- member_expression [5.1](#)
- MEMBER条件 [4.1](#)
- merge_insert_clause [5.1](#)

- [merge_into_existing_keystore 5.1](#)
- [merge_into_new_keystore 5.1](#)
- [merge_table_partitions 5.1](#)
- [merge_table_subpartitions 5.1](#)
- [merge_update_clause 5.1](#)
- [MERGE文 1.1](#)
- [migrate_key 5.1](#)
- [MIN関数 2.1](#)
- [mining_analytic_clause 5.1](#)
- [mining_attribute_clause 5.1](#)
- [model_clause 5.1](#)
- [model_column_clauses 5.1](#)
- [model_iterate_clause 5.1](#)
- [model_rules_clause 5.1](#)
- [モデル式 3.1](#)
- [MOD関数 2.1](#)
- [modify_col_properties 5.1](#)
- [modify_col_substitutable 5.1](#)
- [modify_col_visibility 5.1](#)
- [modify_collection_retrieval 5.1](#)
- [modify_column_clauses 5.1](#)
- [modify_diskgroup_file 5.1](#)
- [modify_filegroup_clause 5.1](#)
- [modify_hash_partition 5.1](#)
- [modify_index_default_attrs 5.1](#)
- [modify_index_partition 5.1](#)
- [modify_index_subpartition 5.1](#)
- [modify_list_partition 5.1](#)
- [modify_LOB_parameters 5.1](#)
- [modify_LOB_storage_clause 5.1](#)
- [modify_mv_column_clause 5.1](#)
- [modify_opaque_type 5.1](#)
- [modify_range_partition 5.1](#)
- [modify_table_default_attrs 5.1](#)
- [modify_table_partition 5.1](#)
- [modify_table_subpartition 5.1](#)
- [modify_to_partitioned 5.1](#)
- [modify_virtcol_properties 5.1](#)
- [modify_volume_clause 5.1](#)
- [MONTHS_BETWEEN関数 2.1](#)
- [move_datafile_clause 5.1](#)
- [move_mv_log_clause 5.1](#)
- [move_table_clause 5.1](#)

- [move_table_partition 5.1](#)
 - [move_table_subpartition 5.1](#)
 - [move_to_filegroup_clause 5.1](#)
 - [multi_column_for_loop 5.1](#)
 - [multi_table_insert 5.1](#)
 - [multiset_except 5.1](#)
 - [multiset_intersect 5.1](#)
 - [multiset_union 5.1](#)
 - [mv_log_augmentation 5.1](#)
 - [mv_log_purge_clause 5.1](#)
-

N

- [named_member_keys 5.1](#)
- [NANVL関数 2.1](#)
- [NCHR関数 2.1](#)
- [nested_table_col_properties 5.1](#)
- [nested_table_partition_spec 5.1](#)
- [NEW_TIME関数 2.1](#)
- [new_values_clause 5.1](#)
- [NEXT_DAY関数 2.1](#)
- [NLS_CHARSET_DECL_LEN関数 2.1](#)
- [NLS_CHARSET_ID関数 2.1](#)
- [NLS_CHARSET_NAME関数 2.1](#)
- [NLS_COLLATION_ID関数 2.1](#)
- [NLS_COLLATION_NAME関数 2.1](#)
- [NLS_INITCAP関数 2.1](#)
- [NLS_LOWER関数 2.1](#)
- [NLS_UPPER関数 2.1](#)
- [NLSSORT関数 2.1](#)
- [NOAUDIT\(従来型監査\)文 1.1](#)
- [NOAUDIT\(統合監査\)文 1.1](#)
- [NTH_VALUE関数 2.1](#)
- [NTILE関数 2.1](#)
- [null条件 4.1](#)
- [NULLIF関数 2.1](#)
- [数値 5.1](#)
- [number_datatypes 6.2](#)
- [数値書式の要素 7.1.1.1](#)
- [数値書式モデル 7.1.1.1](#)
- [numeric_file_name 5.1](#)
- [NUMTODSINTERVAL関数 2.1](#)
- [NUMTOYMINTERVAL関数 2.1](#)

- NVL2関数 [2.1](#)
 - NVL関数 [2.1](#)
-

O

- object_properties [5.1](#)
 - object_step [5.1](#)
 - object_table [5.1](#)
 - object_table_substitution [5.1](#)
 - object_type_col_properties [5.1](#)
 - object_view_clause [5.1](#)
 - オブジェクト・アクセス式 [3.1](#)
 - OID_clause [5.1](#)
 - OID_index_clause [5.1](#)
 - on_comp_partitioned_table [5.1](#)
 - on_hash_partitioned_table [5.1](#)
 - on_list_partitioned_table [5.1](#)
 - on_object_clause [5.1](#)
 - on_range_partitioned_table [5.1](#)
 - open_keystore [5.1](#)
 - option_values [5.1](#)
 - ORA_DM_PARTITION_NAME関数 [2.1](#)
 - ORA_DST_AFFECTED関数 [2.1](#)
 - ORA_DST_CONVERT関数 [2.1](#)
 - ORA_DST_ERROR関数 [2.1](#)
 - ORA_HASH関数 [2.1](#)
 - ORA_INVOKING_USER関数 [2.1](#)
 - ORA_INVOKING_USERID関数 [2.1](#)
 - Oracle組込みデータ型 [6.1](#), [6.2](#)
 - Oracleが提供するデータ型 [6.1](#), [6.3](#)
 - order_by_clause [5.1](#)
 - ordinality_column [5.1](#)
 - out_of_line_constraint [5.1](#)
 - out_of_line_part_storage [5.1](#)
 - out_of_line_ref_constraint [5.1](#)
 - outer_join_clause [5.1](#)
 - outer_join_type [5.1](#)
-

P

- parallel_clause [5.1](#)
- parallel_pdb_creation_clause [5.1](#)

- [partial_database_recovery 5.1](#)
- [partial_index_clause 5.1](#)
- [partition_attributes 5.1](#)
- [partition_extended_name 5.1](#)
- [partition_extended_names 5.1](#)
- [partition_extension_clause 5.1](#)
- [partition_or_key_value 5.1](#)
- [partition_spec 5.1](#)
- [partitioning_storage_clause 5.1](#)
- [partitionset_clauses 5.1](#)
- [password_parameters 5.1](#)
- [path_prefix_clause 5.1](#)
- [PATH関数 2.1](#)
- [pdb_change_state 5.1](#)
- [pdb_change_state_from_root 5.1](#)
- [pdb_close 5.1](#)
- [pdb_datafile_clause 5.1](#)
- [pdb_dba_roles 5.1](#)
- [pdb_force_logging_clause 5.1](#)
- [pdb_general_recovery 5.1](#)
- [pdb_logging_clauses 5.1](#)
- [pdb_open 5.1](#)
- [pdb_recovery_clauses 5.1](#)
- [pdb_refresh_mode_clause 5.1](#)
- [pdb_save_or_discard_state 5.1](#)
- [pdb_settings_clauses 5.1](#)
- [pdb_storage_clause 5.1](#)
- [pdb_unplug_clause 5.1](#)
- [PERCENT_RANK\(集計\)関数 2.1](#)
- [PERCENT_RANK\(分析\)関数 2.1](#)
- [PERCENTILE_CONT関数 2.1](#)
- [PERCENTILE_DISC関数 2.1](#)
- [period_definition 5.1](#)
- [permanent_tablespace_attrs 5.1](#)
- [permanent_tablespace_clause 5.1](#)
- [physical_attributes_clause 5.1](#)
- [physical_properties 5.1](#)
- [pivot_clause 5.1](#)
- [pivot_for_clause 5.1](#)
- [pivot_in_clause 5.1](#)
- [プレースホルダ式 3.1](#)
- [plsqli_declarations 5.1](#)
- [pos_member_keys 5.1](#)

- POWER関数 [2.1](#)
 - POWERMULTISET_BY_CARDINALITY関数 [2.1](#)
 - POWERMULTISET関数 [2.1](#)
 - preceding_boundary [5.1](#)
 - PREDICTION_BOUNDS関数 [2.1](#)
 - PREDICTION_COST(分析)関数 [2.1](#)
 - PREDICTION_COST関数 [2.1](#)
 - PREDICTION_DETAILS(分析)関数 [2.1](#)
 - PREDICTION_DETAILS関数 [2.1](#)
 - PREDICTION_PROBABILITY(分析)関数 [2.1](#)
 - PREDICTION_PROBABILITY関数 [2.1](#)
 - PREDICTION_SET(分析)関数 [2.1](#)
 - PREDICTION_SET関数 [2.1](#)
 - PREDICTION(分析)関数 [2.1](#)
 - PREDICTION関数 [2.1](#)
 - prefix_compression [5.1](#)
 - PRESENTNNV関数 [2.1](#)
 - PRESENTV関数 [2.1](#)
 - PREVIOUS関数 [2.1](#)
 - privilege_audit_clause [5.1](#)
 - program_unit [5.1](#)
 - proxy_clause [5.1](#)
 - PURGE文 [1.1](#)
-

Q

- qdr_expression [5.1](#)
 - qualified_disk_clause [5.1](#)
 - qualified_template_clause [5.1](#)
 - qualifier [5.1](#)
 - query_block [5.1](#)
 - query_partition_clause [5.1](#)
 - query_rewrite_clause [5.1](#)
 - query_table_expression [5.1](#)
 - quiesce_clauses [5.1](#)
 - QUIT SQL*Plusコマンド [A.1](#)
 - quotagroup_clauses [5.1](#)
-

R

- range_partition_desc [5.1](#)
- range_partitions [5.1](#)

- [range_partitionset_clause 5.1](#)
- [range_partitionset_desc 5.1](#)
- [range_subpartition_desc 5.1](#)
- [range_values_clause 5.1](#)
- [RANK\(集計\)ファンクション 2.1](#)
- [RANK\(分析\)ファンクション 2.1](#)
- [RATIO_TO_REPORTファンクション 2.1](#)
- [RAWTOHEXファンクション 2.1](#)
- [RAWTONHEXファンクション 2.1](#)
- [read_only_clause 5.1](#)
- [rebalance_diskgroup_clause 5.1](#)
- [rebuild_clause 5.1](#)
- [records_per_block_clause 5.1](#)
- [recovery_clauses 5.1](#)
- [redo_log_file_spec 5.1](#)
- redo_thread_clauses
 - 「instance_clauses」を参照 [5.1](#)
- [redundancy_clause 5.1](#)
- [reference_model 5.1](#)
- [reference_partition_desc 5.1](#)
- [reference_partitioning 5.1](#)
- [references_clause 5.1](#)
- [REFファンクション 2.1](#)
- [REFTOHEXファンクション 2.1](#)
- [REGEXP_COUNTファンクション 2.1](#)
- [REGEXP_INSTRファンクション 2.1](#)
- [REGEXP_LIKE条件 4.1](#)
- [REGEXP_REPLACEファンクション 2.1](#)
- [REGEXP_SUBSTRファンクション 2.1](#)
- [register_logfile_clause 5.1](#)
- [REGR_AVGXファンクション 2.1](#)
- [REGR_AVGYファンクション 2.1](#)
- [REGR_COUNTファンクション 2.1](#)
- [REGR_INTERCEPTファンクション 2.1](#)
- [REGR_R2ファンクション 2.1](#)
- [REGR_SLOPEファンクション 2.1](#)
- [REGR_SXXファンクション 2.1](#)
- [REGR_SXYファンクション 2.1](#)
- [REGR_SYYファンクション 2.1](#)
- [relational_properties 5.1](#)
- [relational_table 5.1](#)
- [relocate_clause 5.1](#)
- [REMAINDERファンクション 2.1](#)

- [rename_column_clause 5.1](#)
- [rename_disk_clause 5.1](#)
- [rename_index_partition 5.1](#)
- [rename_partition_subpart 5.1](#)
- [RENAME文 1.1](#)
- [replace_disk_clause 5.1](#)
- [REPLACE関数 2.1](#)
- [resize_disk_clause 5.1](#)
- [resource_parameters 5.1](#)
- [return_rows_clause 5.1](#)
- [returning_clause 5.1](#)
- [reverse_migrate_key 5.1](#)
- [revoke_object_privileges 5.1](#)
- [revoke_roles_from_programs 5.1](#)
- [revoke_system_privileges 5.1](#)
- [revokee_clause 5.1](#)
- [REVOKE文 1.1](#)
- [role_audit_clause 5.1](#)
- [ROLLBACK文 1.1](#)
- [rolling_migration_clauses 5.1](#)
- [rolling_patch_clauses 5.1](#)
- [rollup_cube_clause 5.1](#)
- [ROUND\(日付\)関数 2.1](#)
- [ROUND\(数値\)関数 2.1](#)
- [routine_clause 5.1](#)
- [row_limiting_clause 5.1](#)
- [row_movement_clause 5.1](#)
- [ROW_NUMBER関数 2.1](#)
- [row_pattern 5.1](#)
- [row_pattern_aggregate_func 5.1](#)
- [row_pattern_classifier_func 5.1](#)
- [row_pattern_clause 5.1](#)
- [row_pattern_definition 5.1](#)
- [row_pattern_definition_list 5.1](#)
- [row_pattern_factor 5.1](#)
- [row_pattern_match_num_func 5.1](#)
- [row_pattern_measure_column 5.1](#)
- [row_pattern_measures 5.1](#)
- [row_pattern_nav_compound 5.1](#)
- [row_pattern_nav_logical 5.1](#)
- [row_pattern_nav_physical 5.1](#)
- [row_pattern_navigation_func 5.1](#)
- [row_pattern_order_by 5.1](#)

- [row_pattern_partition_by 5.1](#)
 - [row_pattern_permute 5.1](#)
 - [row_pattern_primary 5.1](#)
 - [row_pattern_quantifier 5.1](#)
 - [row_pattern_rec_func 5.1](#)
 - [row_pattern_rows_per_match 5.1](#)
 - [row_pattern_skip_to 5.1](#)
 - [row_pattern_subset_clause 5.1](#)
 - [row_pattern_subset_item 5.1](#)
 - [row_pattern_term 5.1](#)
 - [rowid_datatypes 6.2](#)
 - [ROWIDTOCHAR関数 2.1](#)
 - [ROWTONCHAR関数 2.1](#)
 - [RPAD関数 2.1](#)
 - [RTRIM関数 2.1](#)
 - [RUN SQL*Plusコマンド A.1](#)
-

S

- [sample_clause 5.1](#)
- [SAVEPOINT文 1.1](#)
- [SAVE SQL*Plus コマンド A.1](#)
- [スカラー副問合せ式 3.1](#)
- [科学表記法 7.1.1.1](#)
- [SCN_TO_TIMESTAMP関数 2.1](#)
- [scoped_table_ref_constraint 5.1](#)
- [scrub_clause 5.1](#)
- [search_clause 5.1](#)
- [searched_case_expression 5.1](#)
- [secret_management_clauses 5.1](#)
- [security_clause 5.1](#)
- [security_clauses 5.1](#)
- [segment_attributes_clause 5.1](#)
- [segment_management_clause 5.1](#)
- [select_list 5.1](#)
- [SELECT文 1.1](#)
- [service_name_convert 5.1](#)
- [SESSIONTIMEZONE関数 2.1](#)
- [set_encryption_key 5.1](#)
- [set_key 5.1](#)
- [set_key_tag 5.1](#)
- [set_parameter_clause 5.1](#)
- [set_subpartition_template 5.1](#)

- [set_time_zone_clause 5.1](#)
- [SET CONSTRAINT文 1.1](#)
- [SETファンクション 2.1](#)
- [SET ROLE文 1.1](#)
- [SET SQL*Plusコマンド A.1](#)
- [SET TRANSACTION文 1.1](#)
- [share_clause 5.1](#)
- [share_of_expression 5.1](#)
- [sharing_clause 5.1](#)
- [SHOW SQL*Plusコマンド A.1](#)
- [shrink_clause 5.1](#)
- [shutdown_dispatcher_clause 5.1](#)
- [SHUTDOWN SQL*Plusコマンド A.1](#)
- [SIGNファンクション 2.1](#)
- [simple_case_expression 5.1](#)
- [単純比較条件 4.1](#)
- [単純式 3.1](#)
- [SINファンクション 2.1](#)
- [single_column_for_loop 5.1](#)
- [single_table_insert 5.1](#)
- [SINHファンクション 2.1](#)
- [size_clause 5.1](#)
- [SOUNDEXファンクション 2.1](#)
- [source_file_directory 5.1](#)
- [source_file_name_convert 5.1](#)
- [spatial_types 6.3](#)
- [split_index_partition 5.1](#)
- [split_nested_table_part 5.1](#)
- [split_table_partition 5.1](#)
- [split_table_subpartition 5.1](#)
- [SPOOL SQL*Plusコマンド A.1](#)
- [TO_DSINTERVALファンクションのsql_format 5.1](#)
- [SQL/DSのデータ型](#)
 - [制限事項 6.4](#)
- [SQL*Plusコマンド A, A.1](#)
 - [/ \(スラッシュ\) A.1](#)
 - [@ \(アットマーク\) A.1](#)
 - [APPEND A.1](#)
 - [CHANGE A.1](#)
 - [CONNECT A.1](#)
 - [DEL A.1](#)
 - [DESCRIBE A.1](#)
 - [DISCONNECT A.1](#)

- EDIT [A.1](#)
- EXECUTE [A.1](#)
- EXIT [A.1](#)
- GET [A.1](#)
- HELP [A.1](#)
- HOST [A.1](#)
- INPUT [A.1](#)
- LIST [A.1](#)
- QUIT [A.1](#)
- RUN [A.1](#)
- SAVE [A.1](#)
- SET [A.1](#)
- SHOW [A.1](#)
- SHUTDOWN [A.1](#)
- SPOOL [A.1](#)
- SQLPLUS [A.1](#)
- START [A.1](#)
- STARTUP [A.1](#)
- SQL条件 [4](#)
 - BETWEEN条件 [4.1](#)
 - 複合条件 [4.1](#)
 - EQUALS_PATH条件 [4.1](#)
 - EXISTS条件 [4.1](#)
 - 浮動小数点条件 [4.1](#)
 - グループ比較条件 [4.1](#)
 - IN条件 [4.1](#)
 - IS ANY条件 [4.1](#)
 - IS A SET条件 [4.1](#)
 - IS EMPTY条件 [4.1](#)
 - IS JSON条件 [4.1](#)
 - IS OF type条件 [4.1](#)
 - IS PRESENT条件 [4.1](#)
 - JSON_EXISTS条件 [4.1](#)
 - JSON_TEXTCONTAINS条件 [4.1](#)
 - LIKE条件 [4.1](#)
 - 論理条件 [4.1](#)
 - MEMBER条件 [4.1](#)
 - null条件 [4.1](#)
 - REGEXP_LIKE条件 [4.1](#)
 - 単純比較条件 [4.1](#)
 - SUBMULTISET条件 [4.1](#)
 - UNDER_PATH条件 [4.1](#)
- SQL式 [3](#)

- 計算済メジャー式 [3.1](#)
- CASE式 [3.1](#)
- 列式 [3.1](#)
- 複合式 [3.1](#)
- CURSOR式 [3.1](#)
- 日時式 [3.1](#)
- ファンクション式 [3.1](#)
- INTERVAL式 [3.1](#)
- JSONオブジェクト・アクセス式 [3.1](#)
- モデル式 [3.1](#)
- オブジェクト・アクセス式 [3.1](#)
- プレースホルダ式 [3.1](#)
- スカラー副問合せ式 [3.1](#)
- 単純式 [3.1](#)
- 型コンストラクタ式 [3.1](#)
- SQLファンクション [2](#), [2.1](#)
 - ABS [2.1](#)
 - ACOS [2.1](#)
 - ADD_MONTHS [2.1](#)
 - 集計ファンクション [2.1](#)
 - 分析ファンクション [2.1](#)
 - APPROX_COUNT_DISTINCT [2.1](#)
 - APPROX_COUNT_DISTINCT_AGG [2.1](#)
 - APPROX_COUNT_DISTINCT_DETAIL [2.1](#)
 - APPROX_MEDIAN [2.1](#)
 - APPROX_PERCENTILE [2.1](#)
 - APPROX_PERCENTILE_AGG [2.1](#)
 - APPROX_PERCENTILE_DETAIL [2.1](#)
 - ASCII [2.1](#)
 - ASCIISTR [2.1](#)
 - ASIN [2.1](#)
 - ATAN [2.1](#)
 - ATAN2 [2.1](#)
 - AVG [2.1](#)
 - BFILENAME [2.1](#)
 - BIN_TO_NUM [2.1](#)
 - BITAND [2.1](#)
 - CARDINALITY [2.1](#)
 - CAST [2.1](#)
 - CEIL [2.1](#)
 - CHARTOROWID [2.1](#)
 - CHR [2.1](#)
 - CLUSTER_DETAILS [2.1](#)

- CLUSTER_DETAILS(分析) [2.1](#)
- CLUSTER_DISTANCE [2.1](#)
- CLUSTER_DISTANCE(分析) [2.1](#)
- CLUSTER_ID [2.1](#)
- CLUSTER_ID(分析) [2.1](#)
- CLUSTER_PROBABILITY [2.1](#)
- CLUSTER_PROBABILITY(分析) [2.1](#)
- CLUSTER_SET [2.1](#)
- CLUSTER_SET(分析) [2.1](#)
- COALESCE [2.1](#)
- COLLATION [2.1](#)
- COLLECT [2.1](#)
- COMPOSE [2.1](#)
- CON_DBID_TO_ID [2.1](#)
- CON_GUID_TO_ID [2.1](#)
- CON_NAME_TO_ID [2.1](#)
- CON_UID_TO_ID [2.1](#)
- CONCAT [2.1](#)
- CONVERT [2.1](#)
- CORR [2.1](#)
- CORR_K [2.1](#)
- CORR_S [2.1](#)
- COS [2.1](#)
- COSH [2.1](#)
- COUNT [2.1](#)
- COVAR_POP [2.1](#)
- COVAR_SAMP [2.1](#)
- CUBE_TABLE [2.1](#)
- CUME_DIST(集計) [2.1](#)
- CUME_DIST(分析) [2.1](#)
- CURRENT_DATE [2.1](#)
- CURRENT_TIMESTAMP [2.1](#)
- CV [2.1](#)
- DATAOBJ_TO_MAT_PARTITION [2.1](#)
- DATAOBJ_TO_PARTITION [2.1](#)
- DBTIMEZONE [2.1](#)
- DECODE [2.1](#)
- DECOMPOSE [2.1](#)
- DENSE_RANK(集計) [2.1](#)
- DENSE_RANK(分析) [2.1](#)
- DEPTH [2.1](#)
- Deref [2.1](#)
- DUMP [2.1](#)

- [EMPTY_BLOB 2.1](#)
- [EMPTY_CLOB 2.1](#)
- [EXISTSNODE 2.1](#)
- [EXP 2.1](#)
- [EXTRACT\(日時\) 2.1](#)
- [EXTRACT \(XML\) 2.1](#)
- [EXTRACTVALUE 2.1](#)
- [FEATURE_COMPARE 2.1](#)
- [FEATURE_DETAILS 2.1](#)
- [FEATURE_DETAILS\(分析\) 2.1](#)
- [FEATURE_ID 2.1](#)
- [FEATURE_ID\(分析\) 2.1](#)
- [FEATURE_SET 2.1](#)
- [FEATURE_SET\(分析\) 2.1](#)
- [FEATURE_VALUE 2.1](#)
- [FEATURE_VALUE\(分析\) 2.1](#)
- [FIRST 2.1](#)
- [FIRST_VALUE 2.1](#)
- [FLOOR 2.1](#)
- [FROM_TZ 2.1](#)
- [GREATEST 2.1](#)
- [GROUP_ID 2.1](#)
- [GROUPING 2.1](#)
- [GROUPING_ID 2.1](#)
- [HEXTORAW 2.1](#)
- [INITCAP 2.1](#)
- [INSTR 2.1](#)
- [ITERATION_NUMBER 2.1](#)
- [JSON_ARRAY 2.1](#)
- [JSON_ARRAYAGG 2.1](#)
- [JSON_DATAGUIDE 2.1](#)
- [JSON_OBJECT 2.1](#)
- [JSON_OBJECTAGG 2.1](#)
- [JSON_QUERY 2.1](#)
- [JSON_TABLE 2.1](#)
- [JSON_TRANSFORM 2.1](#)
- [JSON_VALUE 2.1](#)
- [LAG 2.1](#)
- [LAST 2.1](#)
- [LAST_DAY 2.1](#)
- [LAST_VALUE 2.1](#)
- [LEAD 2.1](#)
- [LEAST 2.1](#)

- [LENGTH 2.1](#)
- [LISTAGG 2.1](#)
- [LN 2.1](#)
- [LNNVL 2.1](#)
- [LOCALTIMESTAMP 2.1](#)
- [LOG 2.1](#)
- [LOWER 2.1](#)
- [LPAD 2.1](#)
- [LTRIM 2.1](#)
- [MAKE_REF 2.1](#)
- [MAX 2.1](#)
- [MEDIAN 2.1](#)
- [MIN 2.1](#)
- [MOD 2.1](#)
- [MONTHS_BETWEEN 2.1](#)
- [NANVL 2.1](#)
- [NCGR 2.1](#)
- [NEW_TIME 2.1](#)
- [NEXT_DAY 2.1](#)
- [NLS_CHARSET_DECL_LEN 2.1](#)
- [NLS_CHARSET_ID 2.1](#)
- [NLS_CHARSET_NAME 2.1](#)
- [NLS_COLLATION_ID 2.1](#)
- [NLS_COLLATION_NAME 2.1](#)
- [NLS_INITCAP 2.1](#)
- [NLS_LOWER 2.1](#)
- [NLS_UPPER 2.1](#)
- [NLSSORT 2.1](#)
- [NTH_VALUE 2.1](#)
- [NTILE 2.1](#)
- [NULLIF 2.1](#)
- [NUMTODSINTERVAL 2.1](#)
- [NUMTOYMINTERVAL 2.1](#)
- [NVL 2.1](#)
- [NVL2 2.1](#)
- [ORA_DM_PARTITION_NAME 2.1](#)
- [ORA_DST_AFFECTED 2.1](#)
- [ORA_DST_CONVERT 2.1](#)
- [ORA_DST_ERROR 2.1](#)
- [ORA_HASH 2.1](#)
- [ORA_INVOKING_USER 2.1](#)
- [ORA_INVOKING_USERID 2.1](#)
- [PATH 2.1](#)

- PERCENT_RANK(集計) [2.1](#)
- PERCENT_RANK(分析) [2.1](#)
- PERCENTILE_CONT [2.1](#)
- PERCENTILE_DISC [2.1](#)
- POWER [2.1](#)
- POWERMULTISET [2.1](#)
- POWERMULTISET_BY_CARDINALITY [2.1](#)
- PREDICTION [2.1](#)
- PREDICTION_BOUNDS [2.1](#)
- PREDICTION_COST [2.1](#)
- PREDICTION_COST(分析) [2.1](#)
- PREDICTION_DETAILS [2.1](#)
- PREDICTION_DETAILS(分析) [2.1](#)
- PREDICTION_PROBABILITY [2.1](#)
- PREDICTION_PROBABILITY(分析) [2.1](#)
- PREDICTION_SET [2.1](#)
- PREDICTION_SET(分析) [2.1](#)
- PREDICTION(分析) [2.1](#)
- PRESENTNNV [2.1](#)
- PRESENTV [2.1](#)
- PREVIOUS [2.1](#)
- RANK(集計) [2.1](#)
- RANK(分析) [2.1](#)
- RATIO_TO_REPORT [2.1](#)
- RAWTOHEX [2.1](#)
- RAWTONHEX [2.1](#)
- REF [2.1](#)
- REFTOHEX [2.1](#)
- REGEXP_COUNT [2.1](#)
- REGEXP_INSTR [2.1](#)
- REGEXP_REPLACE [2.1](#)
- REGEXP_SUBSTR [2.1](#)
- REGR_AVGX [2.1](#)
- REGR_AVGY [2.1](#)
- REGR_COUNT [2.1](#)
- REGR_INTERCEPT [2.1](#)
- REGR_R2 [2.1](#)
- REGR_SLOPE [2.1](#)
- REGR_SXX [2.1](#)
- REGR_SXY [2.1](#)
- REGR_SYY [2.1](#)
- REMAINDER [2.1](#)
- REPLACE [2.1](#)

- ROUND(日付) [2.1](#)
- ROUND(数值) [2.1](#)
- ROW_NUMBER [2.1](#)
- ROWIDTOCHAR [2.1](#)
- ROWTONCHAR [2.1](#)
- RPAD [2.1](#)
- RTRIM [2.1](#)
- SCN_TO_TIMESTAMP [2.1](#)
- SESSIONTIMEZONE [2.1](#)
- SET [2.1](#)
- SIGN [2.1](#)
- SIN [2.1](#)
- SINH [2.1](#)
- SOUNDEX [2.1](#)
- SQRT [2.1](#)
- STANDARD_HASH [2.1](#)
- STATS_BINOMIAL_TEST [2.1](#)
- STATS_CROSSTAB [2.1](#)
- STATS_F_TEST [2.1](#)
- STATS_KS_TEST [2.1](#)
- STATS_MODE [2.1](#)
- STATS_MW_TEST [2.1](#)
- STATS_ONE_WAY_ANOVA [2.1](#)
- STATS_T_TEST_INDEP [2.1](#)
- STATS_T_TEST_INDEPU [2.1](#)
- STATS_T_TEST_ONE [2.1](#)
- STATS_T_TEST_PAIRED [2.1](#)
- STATS_WSR_TEST [2.1](#)
- STDDEV [2.1](#)
- STDDEV_POP [2.1](#)
- STDDEV_SAMP [2.1](#)
- SUBSTR [2.1](#)
- SUM [2.1](#)
- SYS_CONNECT_BY_PATH [2.1](#)
- SYS_CONTEXT [2.1](#)
- SYS_DBURIGEN [2.1](#)
- SYS_EXTRACT_UTC [2.1](#)
- SYS_GUID [2.1](#)
- SYS_OP_ZONE_ID [2.1](#)
- SYS_TYPEID [2.1](#)
- SYS_XMLAGG [2.1](#)
- SYS_XMLGEN [2.1](#)
- SYSDATE [2.1](#)

- SYSTIMESTAMP [2.1](#)
- TAN [2.1](#)
- TANH [2.1](#)
- TIMESTAMP_TO_SCN [2.1](#)
- TO_APPROX_COUNT_DISTINCT [2.1](#)
- TO_APPROX_PERCENTILE [2.1](#)
- TO_BINARY_DOUBLE [2.1](#)
- TO_BINARY_FLOAT [2.1](#)
- TO_BLOB (bfile) [2.1](#)
- TO_BLOB (raw) [2.1](#)
- TO_CHAR (bfile|blob)[2.1](#)
- TO_CHAR(文字) [2.1](#)
- TO_CHAR(日時) [2.1](#)
- TO_CHAR(数値) [2.1](#)
- TO_CLOB (bfile|blob) [2.1](#)
- TO_CLOB(文字) [2.1](#)
- TO_DATE [2.1](#)
- TO_DSINTERVAL [2.1](#)
- TO_LOB [2.1](#)
- TO_MULTI_BYTE [2.1](#)
- TO_NCHAR(文字) [2.1](#)
- TO_NCHAR(日時) [2.1](#)
- TO_NCHAR(数値) [2.1](#)
- TO_NCLOB [2.1](#)
- TO_NUMBER [2.1](#)
- TO_SINGLE_BYTE [2.1](#)
- TO_TIMESTAMP [2.1](#)
- TO_TIMESTAMP_TZ [2.1](#)
- TO_YMINTERVAL [2.1](#)
- TRANSLATE [2.1](#)
- TRANSLATE...USING [2.1](#)
- TREAT [2.1](#)
- TRIM [2.1](#)
- TRUNC(日付)[2.1](#)
- TRUNC(数値)[2.1](#)
- TZ_OFFSET [2.1](#)
- UID [2.1](#)
- UNISTR [2.1](#)
- UPPER [2.1](#)
- USER [2.1](#)
- ユーザー定義関数 [2.1](#)
- USERENV [2.1](#)
- VALIDATE_CONVERSION [2.1](#)

- VALUE [2.1](#)
- VAR_POP [2.1](#)
- VAR_SAMP [2.1](#)
- VARIANCE [2.1](#)
- VSIZE [2.1](#)
- WIDTH_BUCKET [2.1](#)
- XMLAGG [2.1](#)
- XMLCAST [2.1](#)
- XMLCDATA [2.1](#)
- XMLCOLATTVAL [2.1](#)
- XMLCOMMENT [2.1](#)
- XMLCONCAT [2.1](#)
- XMLDIFF [2.1](#)
- XMLELEMENT [2.1](#)
- XMLEXISTS [2.1](#)
- XMLFOREST [2.1](#)
- XMLISVALID [2.1](#)
- XMLPARSE [2.1](#)
- XMLPATCH [2.1](#)
- XMLPI [2.1](#)
- XMLQUERY [2.1](#)
- XMLROOT [2.1](#)
- XMLSEQUENCE [2.1](#)
- XMLSERIALIZE [2.1](#)
- XMLTABLE [2.1](#)
- XMLTRANSFORM [2.1](#)
- SQLPLUS SQL*Plusコマンド [A.1](#)
- SQL文 [1](#)
 - ADMINISTER KEY MANAGEMENT [1.1](#)
 - ALTER ANALYTIC VIEW [1.1](#)
 - ALTER ATTRIBUTE DIMENSION [1.1](#)
 - ALTER AUDIT POLICY [1.1](#)
 - ALTER CLUSTER [1.1](#)
 - ALTER DATABASE [1.1](#)
 - ALTER DATABASE LINK [1.1](#)
 - ALTER DIMENSION [1.1](#)
 - ALTER DISKGROUP [1.1](#)
 - ALTER FLASHBACK ARCHIVE [1.1](#)
 - ALTER FUNCTION [1.1](#)
 - ALTER HIERARCHY [1.1](#)
 - ALTER INDEX [1.1](#)
 - ALTER INDEXTYPE [1.1](#)
 - ALTER INMEMORY JOIN GROUP [1.1](#)

- ALTER JAVA [1.1](#)
- ALTER LIBRARY [1.1](#)
- ALTER LOCKDOWN PROFILE [1.1](#)
- ALTER MATERIALIZED VIEW [1.1](#)
- ALTERMATERIALIZEDVIEWLOG [1.1](#)
- ALTER MATERIALIZED ZONEMAP [1.1](#)
- ALTER OPERATOR [1.1](#)
- ALTER OUTLINE [1.1](#)
- ALTER PACKAGE [1.1](#)
- ALTER PLUGGABLE DATABASE [1.1](#)
- ALTER PROCEDURE [1.1](#)
- ALTER PROFILE [1.1](#)
- ALTER RESOURCE COST [1.1](#)
- ALTER ROLE [1.1](#)
- ALTER ROLLBACK SEGMENT [1.1](#)
- ALTER SEQUENCE [1.1](#)
- ALTER SESSION [1.1](#)
- ALTER SYNONYM [1.1](#)
- ALTER SYSTEM [1.1](#)
- ALTER TABLE [1.1](#)
- ALTER TABLESPACE [1.1](#)
- ALTER TABLESPACE SET [1.1](#)
- ALTER TRIGGER [1.1](#)
- ALTER TYPE [1.1](#)
- ALTER USER [1.1](#)
- ALTER VIEW [1.1](#)
- ANALYZE [1.1](#)
- ASSOCIATE STATISTICS [1.1](#)
- AUDIT (従来型監査) [1.1](#)
- AUDIT (統合監査) [1.1](#)
- CALL [1.1](#)
- COMMENT [1.1](#)
- COMMIT [1.1](#)
- CREATE ANALYTIC VIEW [1.1](#)
- CREATE ATTRIBUTE DIMENSION [1.1](#)
- CREATE AUDIT POLICY [1.1](#)
- CREATE CLUSTER [1.1](#)
- CREATE CONTEXT [1.1](#)
- CREATE CONTROLFILE [1.1](#)
- CREATE DATABASE [1.1](#)
- CREATE DATABASE LINK [1.1](#)
- CREATE DIMENSION [1.1](#)
- CREATE DIRECTORY [1.1](#)

- CREATE DISKGROUP [1.1](#)
- CREATE EDITION [1.1](#)
- CREATE FLASHBACK ARCHIVE [1.1](#)
- CREATE FUNCTION [1.1](#)
- CREATE HIERARCHY [1.1](#)
- CREATE INDEX [1.1](#)
- CREATE INDEXTYPE [1.1](#)
- CREATE INMEMORY JOIN GROUP [1.1](#)
- CREATE JAVA [1.1](#)
- CREATE LIBRARY [1.1](#)
- CREATE LOCKDOWN PROFILE [1.1](#)
- CREATE MATERIALIZED VIEW [1.1](#)
- CREATEMATERIALIZEDVIEWLOG [1.1](#)
- CREATE MATERIALIZED ZONEMAP [1.1](#)
- CREATE OPERATOR [1.1](#)
- CREATE OUTLINE [1.1](#)
- CREATE PACKAGE [1.1](#)
- CREATE PACKAGE BODY [1.1](#)
- CREATE PFILE [1.1](#)
- CREATE PLUGGABLE DATABASE [1.1](#)
- CREATE PROCEDURE [1.1](#)
- CREATE PROFILE [1.1](#)
- CREATE RESTORE POINT [1.1](#)
- CREATE ROLE [1.1](#)
- CREATE ROLLBACK SEGMENT [1.1](#)
- CREATE SCHEMA [1.1](#)
- CREATE SEQUENCE [1.1](#)
- CREATE SPFILE [1.1](#)
- CREATE SYNONYM [1.1](#)
- CREATE TABLE [1.1](#)
- CREATE TABLESPACE [1.1](#)
- CREATE TABLESPACE SET [1.1](#)
- CREATE TRIGGER [1.1](#)
- CREATE TYPE [1.1](#)
- CREATE TYPE BODY [1.1](#)
- CREATE USER [1.1](#)
- CREATE VIEW [1.1](#)
- DELETE [1.1](#)
- DISASSOCIATE STATISTICS [1.1](#)
- DROP ANALYTIC VIEW [1.1](#)
- DROP ATTRIBUTE DIMENSION [1.1](#)
- DROP AUDIT POLICY [1.1](#)
- DROP CLUSTER [1.1](#)

- DROP CONTEXT [1.1](#)
- DROP DATABASE [1.1](#)
- DROP DATABASE LINK [1.1](#)
- DROP DIMENSION [1.1](#)
- DROP DIRECTORY [1.1](#)
- DROP DISKGROUP [1.1](#)
- DROP EDITION [1.1](#)
- DROP FLASHBACK ARCHIVE [1.1](#)
- DROP FUNCTION [1.1](#)
- DROP HIERARCHY [1.1](#)
- DROP INDEX [1.1](#)
- DROP INDEXTYPE [1.1](#)
- DROP INMEMORY JOIN GROUP [1.1](#)
- DROP JAVA [1.1](#)
- DROP LIBRARY [1.1](#)
- DROP LOCKDOWN PROFILE [1.1](#)
- DROP MATERIALIZED VIEW [1.1](#)
- DROPMATERIALIZEDVIEWLOG [1.1](#)
- DROP MATERIALIZED ZONEMAP [1.1](#)
- DROP OPERATOR [1.1](#)
- DROP OUTLINE [1.1](#)
- DROP PACKAGE [1.1](#)
- DROP PLUGGABLE DATABASE [1.1](#)
- DROP PROCEDURE [1.1](#)
- DROP PROFILE [1.1](#)
- DROP RESTORE POINT [1.1](#)
- DROP ROLE [1.1](#)
- DROP ROLLBACK SEGMENT [1.1](#)
- DROP SEQUENCE [1.1](#)
- DROP SYNONYM [1.1](#)
- DROP TABLE [1.1](#)
- DROP TABLESPACE [1.1](#)
- DROP TABLESPACE SET [1.1](#)
- DROP TRIGGER [1.1](#)
- DROP TYPE [1.1](#)
- DROP TYPE BODY [1.1](#)
- DROP USER [1.1](#)
- DROP VIEW [1.1](#)
- EXPLAIN PLAN [1.1](#)
- FLASHBACK DATABASE [1.1](#)
- FLASHBACK TABLE [1.1](#)
- GRANT [1.1](#)
- INSERT [1.1](#)

- LOCK TABLE [1.1](#)
- MERGE [1.1](#)
- NOAUDIT (従来型監査) [1.1](#)
- NOAUDIT (統合監査) [1.1](#)
- PURGE [1.1](#)
- RENAME [1.1](#)
- REVOKE [1.1](#)
- ROLLBACK [1.1](#)
- SAVEPOINT [1.1](#)
- SELECT [1.1](#)
- SET CONSTRAINT [1.1](#)
- SET ROLE [1.1](#)
- SET TRANSACTION [1.1](#)
- TRUNCATE CLUSTER [1.1](#)
- TRUNCATE TABLE [1.1](#)
- UPDATE [1.1](#)
- SQRT関数 [2.1](#)
- standard_actions [5.1](#)
- STANDARD_HASH関数 [2.1](#)
- standby_database_clauses [5.1](#)
- standbys_clause [5.1](#)
- start_standby_clause [5.1](#)
- START SQL*Plusコマンド [A.1](#)
- startup_clauses [5.1](#)
- STARTUP SQL*Plusコマンド [A.1](#)
- statement_clauses [5.1](#)
- 文 [1](#)
 - 「SQL文」も参照 [1](#)
- STATS_BINOMIAL_TEST関数 [2.1](#)
- STATS_CROSSTAB関数 [2.1](#)
- STATS_F_TEST関数 [2.1](#)
- STATS_KS_TEST関数 [2.1](#)
- STATS_MODE関数 [2.1](#)
- STATS_MW_TEST関数 [2.1](#)
- STATS_ONE_WAY_ANOVA関数 [2.1](#)
- STATS_T_TEST_INDEP関数 [2.1](#)
- STATS_T_TEST_INDEPU関数 [2.1](#)
- STATS_T_TEST_ONE関数 [2.1](#)
- STATS_T_TEST_PAISED関数 [2.1](#)
- STATS_WSR_TEST関数 [2.1](#)
- STDDEV_POP関数 [2.1](#)
- STDDEV_SAMP関数 [2.1](#)
- STDDEV関数 [2.1](#)

- [still_image_object_types 5.1](#)
- [stop_standby_clause 5.1](#)
- [storage_clause 5.1](#)
- [storage_table_clause 5.1](#)
- [文字列 5.1](#)
- [striping_clause 5.1](#)
- [SUBMULTISET条件 4.1](#)
- [subpartition_by_hash 5.1](#)
- [subpartition_by_list 5.1](#)
- [subpartition_by_range 5.1](#)
- [subpartition_extended_name 5.1](#)
- [subpartition_extended_names 5.1](#)
- [subpartition_or_key_value 5.1](#)
- [subpartition_spec 5.1](#)
- [subpartition_template 5.1](#)
- [subquery 5.1](#)
- [subquery_factoring_clause 5.1](#)
- [subquery_restriction_clause 5.1](#)
- [substitutable_column_clause 5.1](#)
- [SUBSTRファンクション 2.1](#)
- [SUMファンクション 2.1](#)
- [supplemental_db_logging 5.1](#)
- [supplemental_id_key_clause 5.1](#)
- [supplemental_log_grp_clause 5.1](#)
- [supplemental_logging_props 5.1](#)
- [supplemental_plsql_clause 5.1](#)
- [supplemental_table_logging 5.1](#)
- [提供されるデータ型 6.1, 6.3](#)
- [switch_logfile_clause 5.1](#)
- [switchover_clause 5.1](#)
- [副次句の構文 5.1](#)
- [SYS_CONNECT_BY_PATHファンクション 2.1](#)
- [SYS_CONTEXTファンクション 2.1](#)
- [SYS_DBURIGENファンクション 2.1](#)
- [SYS_EXTRACT_UTCファンクション 2.1](#)
- [SYS_GUIDファンクション 2.1](#)
- [SYS_OP_ZONE_IDファンクション 2.1](#)
- [SYS_TYPEIDファンクション 2.1](#)
- [SYS_XMLAGGファンクション 2.1](#)
- [SYS_XMLGENファンクション 2.1](#)
- [SYSDATEファンクション 2.1](#)
- [system_partitioning 5.1](#)
- [SYSTIMESTAMPファンクション 2.1](#)

T

- [table_collection_expression 5.1](#)
- [table_compression 5.1](#)
- [table_index_clause 5.1](#)
- [table_partition_description 5.1](#)
- [table_partitioning_clauses 5.1](#)
- [table_properties 5.1](#)
- [table_reference 5.1](#)
- [tablespace_clauses 5.1](#)
- [tablespace_datafile_clauses 5.1](#)
- [tablespace_encryption_clause 5.1](#)
- [tablespace_encryption_spec 5.1](#)
- [tablespace_group_clause 5.1](#)
- [tablespace_logging_clauses 5.1](#)
- [tablespace_retention_clause 5.1](#)
- [tablespace_state_clauses 5.1](#)
- [TAN関数 2.1](#)
- [TANH関数 2.1](#)
- [tempfile_reuse_clause 5.1](#)
- [temporary_tablespace_clause 5.1](#)
- TIMEデータ型
 - [DB2 6.4](#)
 - [SQL/DS 6.4](#)
- [時刻書式モデル 7.1.2.1](#)
- [timeout_clause 5.1](#)
- [TIMESTAMP_TO_SCN関数 2.1](#)
- TIMESTAMPデータ型
 - [DB2 6.4](#)
 - [SQL/DS 6.4](#)
- [タイムゾーンの書式化 7.1.2.1](#)
- [TO_APPROX_COUNT_DISTINCT関数 2.1](#)
- [TO_APPROX_PERCENTILE関数 2.1](#)
- [TO_BINARY_DOUBLE関数 2.1](#)
- [TO_BINARY_FLOAT関数 2.1](#)
- [TO_BLOB \(bfile\)関数 2.1](#)
- [TO_BLOB \(raw\)関数 2.1](#)
- [TO_CHAR \(bfile|blob\)関数 2.1](#)
- [TO_CHAR \(文字\)関数 2.1](#)
- [TO_CHAR \(日時\)関数 2.1](#)
- [TO_CHAR \(数値\)関数 2.1](#)
- [TO_CLOB \(bfile|blob\)関数 2.1](#)

- TO_CLOB (文字)ファンクション [2.1](#)
 - TO_DATEファンクション [2.1](#)
 - TO_DSINTERVALファンクション [2.1](#)
 - TO_LOBファンクション [2.1](#)
 - TO_MULTI_BYTEファンクション [2.1](#)
 - TO_NCHAR (文字)ファンクション [2.1](#)
 - TO_NCHAR (日時)ファンクション [2.1](#)
 - TO_NCHAR (数値)ファンクション [2.1](#)
 - TO_NCLOBファンクション [2.1](#)
 - TO_NUMBERファンクション [2.1](#)
 - TO_SINGLE_BYTEファンクション [2.1](#)
 - TO_TIMESTAMP_TZファンクション [2.1](#)
 - TO_TIMESTAMPファンクション [2.1](#)
 - TO_YMINTERVALファンクション [2.1](#)
 - trace_file_clause [5.1](#)
 - TRANSLATE...USINGファンクション [2.1](#)
 - TRANSLATEファンクション [2.1](#)
 - TREATファンクション [2.1](#)
 - TRIMファンクション [2.1](#)
 - TRUNC (日付)ファンクション [2.1](#)
 - TRUNC (数値)ファンクション [2.1](#)
 - truncate_partition_subpart [5.1](#)
 - TRUNCATE CLUSTER文 [1.1](#)
 - TRUNCATE TABLE文 [1.1](#)
 - ts_file_name_convert [5.1](#)
 - 型コンストラクタ式 [3.1](#)
 - TZ_OFFSETファンクション [2.1](#)
-

U

- UIDファンクション [2.1](#)
- UNDER_PATH条件 [4.1](#)
- undo_mode_clause [5.1](#)
- undo_tablespace [5.1](#)
- undo_tablespace_clause [5.1](#)
- undrop_disk_clause [5.1](#)
- UNISTRファンクション [2.1](#)
- unpivot_clause [5.1](#)
- unpivot_in_clause [5.1](#)
- unusable_editions_clause [5.1](#)
- update_all_indexes_clause [5.1](#)
- update_global_index_clause [5.1](#)
- update_index_clauses [5.1](#)

- [update_index_partition 5.1](#)
 - [update_index_subpartition 5.1](#)
 - [update_set_clause 5.1](#)
 - [UPDATE文 1.1](#)
 - [upgrade_table_clause 5.1](#)
 - [UPPERファンクション 2.1](#)
 - [use_key 5.1](#)
 - [user_clauses 5.1](#)
 - [user_tablespaces_clause 5.1](#)
 - [ユーザー定義データ型 6.1](#)
 - [ユーザー定義関数 2.1](#)
 - [USERENVファンクション 2.1](#)
 - [USERファンクション 2.1](#)
 - [usergroup_clauses 5.1](#)
 - [using_clause 5.1](#)
 - [using_function_clause 5.1](#)
 - [using_index_clause 5.1](#)
 - [using_statistics_type 5.1](#)
 - [using_type_clause 5.1](#)
-

V

- [VALIDATE_CONVERSIONファンクション 2.1](#)
 - [validation_clauses 5.1](#)
 - [VALUEファンクション 2.1](#)
 - [values_clause 5.1](#)
 - [VAR_POPファンクション 2.1](#)
 - [VAR_SAMPファンクション 2.1](#)
 - [VARGRAPHICデータ型](#)
 - [DB2 6.4](#)
 - [SQL/DS 6.4](#)
 - [VARIANCEファンクション 2.1](#)
 - [varray_col_properties 5.1](#)
 - [varray_storage_clause 5.1](#)
 - [virtual_column_definition 5.1](#)
 - [VSIZEファンクション 2.1](#)
-

W

- [where_clause 5.1](#)
- [WIDTH_BUCKETファンクション 2.1](#)
- [window_clause 5.1](#)

- [window_expression 5.1](#)
 - [windowing_clause 5.1](#)
 - [with_clause 5.1](#)
-

X

- [XML_attributes_clause 5.1](#)
 - [XML_passing_clause 5.1](#)
 - [XML_table_column 5.1](#)
 - [XML_types 6.3](#)
 - [XMLAGG関数 2.1](#)
 - [XMLCast関数 2.1](#)
 - [XMLCDATA関数 2.1](#)
 - [XMLCOLATTVAL関数 2.1](#)
 - [XMLCOMMENT関数 2.1](#)
 - [XMLCONCAT関数 2.1](#)
 - [XMLDIFF関数 2.1](#)
 - [XMLELEMENT関数 2.1](#)
 - [XMLEXISTS関数 2.1](#)
 - [XMLFOREST関数 2.1](#)
 - [XMLIndex_clause 5.1](#)
 - [XMLISVALID関数 2.1](#)
 - [XMLnamespaces_clause 5.1](#)
 - [XMLPARSE関数 2.1](#)
 - [XMLPATCH関数 2.1](#)
 - [XMLPI関数 2.1](#)
 - [XMLQUERY関数 2.1](#)
 - [XMLROOT関数 2.1](#)
 - [XMLSchema_spec 5.1](#)
 - [XMLSEQUENCE関数 2.1](#)
 - [XMLSERIALIZE関数 2.1](#)
 - [XMLTABLE_options 5.1](#)
 - [XMLTABLE関数 2.1](#)
 - [XMLTRANSFORM関数 2.1](#)
 - [XMLType_column_properties 5.1](#)
 - [XMLType_storage 5.1](#)
 - [XMLType_table 5.1](#)
 - [XMLType_view_clause 5.1](#)
 - [XMLType_virtual_columns 5.1](#)
-

Y

- TO_YMINTERVAL関数のym_iso_format [5.1](#)
-

Z

- zonemap_attributes [5.1](#)
- zonemap_clause [5.1](#)
- zonemap_refresh_clause [5.1](#)