

Oracle® NoSQL Database

Release Notes



Release 26.1

E91818-46

April 2026

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle NoSQL Database Release Notes, Release 26.1

E91818-46

Copyright © 2011, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Overview

Oracle NoSQL Database is a distributed key-value store capable of scaling horizontally to handle very large amounts of data.

This Release

This is release 26.1.8 of Oracle NoSQL Database.

We strongly recommend that users of earlier versions upgrade to this release. For a complete listing of all the changes made to Oracle NoSQL Database for this release, including all bug fixes, see the change log.

Supported Java Versions

Since the 26.1 release, the Oracle NoSQL Database client and server have been compatible with Java SE 17. Both the client and the server require at least Java SE 17, and should work with more recent Java SE versions. For the 26.1 release, the client and server have been tested and certified against Oracle Java SE 21.0.9 and OpenJDK 21.0.9. Java SE 21 is a long-term support release. We encourage you to upgrade to Java SE 21 to take advantage of the latest bug fixes and performance improvements.

We plan to deprecate use of versions earlier than Java SE 21 with the NoSQL server later in calendar year 2026, and eventually we will do the same for the NoSQL client. If you are using a Java earlier than version 21, please prepare for this change.

Although this release is compatible with the latest LTS release, Java 21, there is a known limitation when using Java 21 virtual threads with the Direct Java Driver API. See Known Issues.

Attempting to use this release with a version of Java earlier than the required version will produce an error message similar to:

```
Error: LinkageError occurred while loading main class oracle.kv.KVVersion
  java.lang.UnsupportedClassVersionError: oracle/kv/KVVersion has been
  compiled
    by a more recent version of the Java Runtime (class file version 61.0),
    this version of the Java Runtime only recognizes class file versions up to
  55.0
```

Requiring Java 21 in future releases

We expect to require Java SE 21 for the server starting later in 2026, and eventually for the client as well. Please plan to switch to a Java 21 or later runtime in preparation for use of future releases. We have been certifying with Java SE 21 since the beginning of 2024, starting with version 24.1.

Java Flow interface to replace Reactive Streams in future releases

We expect to replace the third-party Reactive Streams interface with the native-Java Flow interface in the near future.

Product Upgrade Policy

Release 26.1 supports upgrades starting with the 24.1 release. To upgrade a store directly to the current release, the store must be running release 24.1 or later.

If you have a store running a 20.x, 21.x or 22.x or 23.x release, we recommend that you upgrade it to the current release by first upgrading to a 24.x or 25.x release. Then you can upgrade to the current release.

If you have a store running a 19.x or earlier release and you are an Enterprise Edition user, please contact support. If you are a Community Edition user with this issue, please post a question to the Oracle NoSQL Database GitHub repository.

 **Note**

Release 26.1 introduced a new store-wide system-level compatibility state which can be upgraded with `plan upgrade-scv` command.

New Features in Release 26.1

System Compatibility Version (SCV) Upgrade and Row Metadata Enhancements

This release enhances upgrade safety and feature enablement through the **System Compatibility Version (SCV)** mechanism. New releases can introduce incompatible changes; SCV provides an explicit, post-upgrade step to ensure the store reaches a consistent compatibility state and to prevent partially-upgraded environments from inadvertently using features that are not yet enabled.

After upgrading all storage nodes and related components (SDK/direct drivers, multi-region agents, etc.), administrators must run `plan upgrade-scv -version` from the Admin CLI to enable features up to the specified version. Until the SCV upgrade completes, attempts to use newly introduced capabilities may fail with “incompatible client/service” or “feature requires version enabled” errors. The `verify configuration`, `verify upgrade`, and `verify prerequisite` commands will emit advisory notes when SCV has not been upgraded.

See the documentation for more details

Predicate Filtering on Streams

Streams API now extends server-side filtering beyond table subscriptions by supporting **application-supplied predicates**. The kvstore Feeder continues to block writes to non-subscribed tables, and for subscribed tables it now streams only the writes that match the provided predicates. This significantly reduces network traffic and downstream processing, and simplifies applications by removing the need for complex client-side filtering logic.

Last Write Metadata (GA in 26.1)

Last Write Metadata is now **officially released (GA) in 26.1**.

This feature allows applications to attach **user-supplied metadata** to write operations, provided as a **JSON** object, and **persist** that information alongside the record. Records can later be filtered using the system-defined function `last_write_metadata()`. If you previously used the experimental “User-supplied record metadata” capability, note that it has been **renamed to “Last Write Metadata”**.

Creation Time Metadata

This feature allows application to obtain the original insertion timestamp (UTC, ms precision) of a row, complementing the existing modification time, which returns the latest update time (or insertion time if never modified). Added `creation_time()` query row function returning the creation time. Note that rows inserted prior to enabling the system compatibility version to 26.1 will have ill-defined creation times. To ensure the values of creation times are accurate, please only obtain creation times from rows that are inserted after the system compatibility version is enabled to 26.1.

KVInsight Monitoring Framework (Experimental)

Added an experimental version of **KVInsight**, a **tool-agnostic monitoring framework** for Oracle NoSQL Database. KVInsight provides a unified monitoring infrastructure to simplify how performance metrics are collected, aggregated, and exported, enabling integration with a variety of monitoring solutions.

Today, Oracle NoSQL Database exposes performance monitoring through multiple mechanisms (for example, **JMX MBeans**, **MonitorAgent StatsPacket**, and **the Collector Service**) that differ in datasets, formats, and storage locations (CSV, JSON, JMX messages).

This fragmentation makes it difficult for administrators to set up monitoring dashboards and pipelines, and it can limit flexibility in choosing monitoring backends. It can also make it harder for operations teams to interpret metrics consistently and take timely, actionable steps to keep clusters healthy.

KVInsight addresses these challenges by offering a consistent configuration and export model for monitoring data. In this release, KVInsight supports exporting metrics via **CSV and Prometheus**.

Before using this experimental feature, review the documentation for current limitations and deployment considerations. We appreciate your feedback on the usefulness of KVInsight and encourage you to post feedback in the Oracle NoSQL Database GitHub repository.

Support for UNION ALL (Experimental)

Added support for the UNION ALL operator to combine results from multiple SELECT statements into a single result set without de-duplication, preserving all rows returned by each query.

Support for SUBQUERY (Experimental)

Added support for SUBQUERY expressions, enabling nested SELECT statements within a query (e.g., in WHERE, FROM, or SELECT clauses) to support more complex filtering and data shaping patterns.

Deprecation Notice: Coherence Package

Release 26.1 is the last release to ship with the Coherence package. The Coherence package is deprecated and is no longer certified with Oracle NoSQL Database.

Deprecation Notice: External Table Package

Deprecation Notice: External Table Package Release 26.1 is the last release to ship with the External Table package. The External Table (exttab) package is deprecated and is no longer certified with Oracle NoSQL Database.

Deprecation Notice: Hadoop

Release 26.1 is the last release to ship with the Hadoop package. The Hadoop package is deprecated and is no longer certified with Oracle NoSQL Database.

New Features in Release 25.3

Before Image in Streams API

Added support for before image in the Streams API. Users can configure subscription streams to include the before images of streamed operations.

Streaming Transactions

Added support for streaming by transaction in the Streams API. A user can now configure subscription streams to stream transactions instead of individual put and delete operations.

User-supplied record metadata

Added an experimental version of user supplied metadata. Users can add their own metadata, formatted as a JSON construct, to write operations and persistently store that information alongside the record. Records can subsequently be filtered using a new system defined function, `row_metadata()`. Before using this experimental feature, see the Known Issues section below to understand its limitations. We would appreciate your feedback on the usefulness of this feature and APIs and encourage you to post your feedback in the NoSQL GitHub repository.

New Features in Release 25.1

Inner Joins

Extended the SQL dialect of Oracle NoSQL Database to support inner joins among tables in the same table hierarchy. Two or more tables in the same hierarchy may be joined in a query.

Support of JSON Merge Patch

Extended the SQL UPDATE statement to support updates of JSON fields via the JSON merge patch specification (described here).

Support for Updating Multiple Records in a Query

An update query can now update multiple records in a single transaction if the query predicate contains a shard key.

New Features in Release 24.4

No new features were added in release 24.4.

New Features in Release 24.3

New Timestamp SQL functions

Implemented new SQL functions to round a timestamp, cast a timestamp to/from a string, or extract information from a timestamp (calendar quarter, day of week/month/year, last day of month).

Support of elasticity operations for multiple sharded subscribers

The Streams API now supports elasticity operations with multiple subscribers. When an application creates multiple sharded subscribers as a group, the Streams API supports elasticity operations in the kvstore. That is, the Streams API will deliver data in the proper order while the topology of the store is being changed by an elasticity operation.

Please note: before this change, users could configure multiple sharded subscribers without a checkpoint table mapper. This API has been deprecated. With the new feature, the checkpoint table mapper is required for multiple sharded subscribers, and an exception will be thrown if the stream is configured without the mapper.

Multi-region tables with multiple cross-region service agents now support elasticity operations

When a user configures a group of cross-region service agents connecting different regions, multi-region tables now support elasticity operations in any region. That is, data in a multi-region table will be eventually consistent in the presence of elasticity operations in any region.

New Features in Release 24.1

SQL Mathematical Functions

Implemented the following mathematical SQL functions: ABS, ACOS, ASIN, ATAN, ATAN2, CEIL, COS, COT, DEGREES, EXP, FLOOR, LOG, LOG10, POWER, RADIANS, RAND, ROUND, SIGN, SIN, SQRT, TAN, TRUNC.

New Features in Release 23.3

JSON Collections

Added JSON Collection as a new table type. The new type makes it more convenient to use JSON to represent unstructured data.

New SQL Aggregate Functions

Implemented three new SQL aggregate functions: `count(distinct)`, `array_collect()`, and `array_collect(distinct)`.

Multi-region Table Agent Groups

Stores with multi-region tables can now achieve greater throughput for data transfer from remote regions by deploying multiple multi-region agents to replicate data from a remote region.

Deprecating Eclipse Plugin

The final release of the Eclipse plugin is 1.0.2. The plugin will be deprecated in Spring 2024. Development and enhancements will continue on the Visual Studio and IntelliJ plugins. We recommend that developers move over to one of those two plugins.

New Features in Release 23.1

Added Support for SQL BETWEEN Operator

Implemented the SQL BETWEEN operator. This is a convenience operator that makes it easier to write queries that include range predicates with both a lower and an upper bound. For example, the query

```
select * from users where age between 20 and 30
```

is equivalent to the query

```
select * from users where 20 <= age and age <= 30
```

Java 11 is Required

This release now requires using Java 11 or later for both client applications and for running the store.

Removed Support for Hive

Removed support for Hive, including all of the classes in the `oracle.kv.hadoop.hive.table` package. Hive support was removed because this release requires Java 11 and the Hive package has not been updated to support that Java release. Hive support was needed for BigDataSQL, which is now also no longer supported.

New Features in Release 22.3

Indexing Functions in SQL

Indexes can now be created on the values of one or more functions over index paths. The set of functions that support indexes is the following subset of the builtin functions:

`modification_time`, `expiration_time`, `expiration_time_millis`, `row_storage_size`,
`year`, `month`, `day`, `hour`, `minute`, `second`, `millisecond`, `microsecond`, `nanosecond`,
`week`, `length`, `replace`, `reverse`, `substring`, `trim`, `ltrim`, `rtrim`, `lower`, `upper`.

New Features in Release 22.2

Removed Import/Export Utility

As of release 22.2 the import/export utility is no longer part of the release. It was previously deprecated and its functionality has been moved to the data migrator utility, which is an independent download. Chapter 7 of the Oracle NoSQL Database Administrator's Guide contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator.

Erasing Obsolete User Data Has Been Enabled By Default

Data erasure, added in 21.2, will now be enabled by default. This feature erases obsolete user data within a specified time period after becoming obsolete, without adding significant performance impact to the application.

Two Rep Node parameters control erasure: "enableErasure" is used to enable erasure, which is now enabled by default, and "erasurePeriod" specifies the time required to complete an erasure cycle, which defaults to 6 days.

Note

Snapshot-based backups create hard links to original files. Until these backups are copied to their target location and the corresponding hard links are removed, erasure will not process obsolete data in those files. This is to preserve the integrity of the backups. Erasure ignores files with hard links to them.

New Features in Release 22.1

JSON MRCounter Types

The MR_COUNTER SQL type can now be used in JSON fields for multi-region tables. The change adds support for merging independent modifications made in multiple regions into a single combined value to schemaless JSON fields.

Elasticity for MR Tables

Stores that contain multi-region tables now support elasticity changes that change the store topology.

Child tables for MR Tables

Multi-region tables now support child tables. Child tables always inherit the regions of the associated top level tables.

KVLocal

Applications can now create and manage a small, standalone store instance using the new KVLocal class.

Deprecating Import/Export Utility

Release 22.1 is the final release to include the import/export utility. It will be removed as of the next quarterly release. We have moved the functionality of the import/export utility over to the data migrator. Chapter 7 of the Oracle NoSQL Database Administrator's Guide contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator. The import/export utility has been deprecated since the 20.3 release and we encourage customers to move to the new data migrator utility now.

New Features in Release 21.2

Unnesting of Arrays and Maps in Queries

The UNNEST clause has been introduced, which converts an array or map into a series of rows. The syntax of the FROM clause has also been extended to allow specifying a list of expressions after the table or tables used by the query, which can influence which indexes the query runtime selects for queries that have UNNEST clauses.

Erasing User Data

Added support for erasing obsolete user data within a specified time period after becoming obsolete, without adding significant performance impact to the application. This new feature, data erasure, runs in the background to zero out obsolete data. All data that is obsolete from the database point of view is removed: i.e. deleted data, older versions of updated records, and expired data. Data erasure also removes corresponding obsolete index records.

Two new Rep Node parameters control erasure: "enableErasure" is used to enable erasure, which is disabled by default, and "erasurePeriod" specifies the time required to complete an erasure cycle.

Note

Snapshot-based backups create hard links to original files. Until these backups are copied to their target location and the corresponding hard links are removed, erasure will not process obsolete data in those files. Erasure ignores files with hard links to them.

Deprecating Property Graph Package

Starting with release 21.2, the RDF Property Graph package is deprecated and is no longer certified with Oracle NoSQL Database.

Disabling DSA (ssh-dss) SSH Keys in the Diagnostics Utility

The diagnostics utility no longer supports DSA (ssh-dss) SSH keys due to security weaknesses with that algorithm. Attempting to use the diagnostics utility with a DSA (ssh-dss) SSH key will produce an error message similar to:

```
DefaultAuthFuture[ssh-connection]: Failed (IllegalArgumentException) to execute: No signer could be located for key type=ssh-dss
```

In a future version, RSA (ssh-rsa) SSH key support may also be removed for similar reasons. It's recommended to migrate to other SSH key algorithms, for example ECDSA or ED25519.

New Features in Release 21.1

MRCounter Type

Added support for the new `MR_COUNTER` SQL type, which can be used with `INTEGER`, `LONG` and `NUMBER` value types in multi-region tables. The new type supports merging independent modifications made in multiple regions into a single combined value.

Left Outer Join

Added support for the `LEFT OUTER JOIN` syntax to be compatible with standard ANSI SQL.

Indexes for Nested Arrays and Maps

This release adds support for indexing arrays and maps that are nested within other arrays or maps.

New SQL Functions

Added new SQL functions to access information about row storage size, index storage size, modification time, containing partition, and containing shard. Also added a function to convert a string to JSON.

New Features in Release 20.3

Universally Unique Identifier

Added SQL support for randomly generated UUIDs, including the `random_uuid()` function, the new "AS UUID" option for STRING data types, and the new "GENERATED BY DEFAULT" option for "STRING AS UUID" data types. These new facilities provide an easy way to create a field that contains a unique ID for each row. UUID fields can be used to create unique primary keys in multi-region tables, since multi-region tables do not support the IDENTITY fields that serve this purpose for local tables.

Deprecating Import/Export Utility

In the Oracle NoSQL Database 20.2 release, we released a new Data Migrator utility. We have moved the functionality of the import/export utility over to the data migrator. The Using Oracle NoSQL Data Migrator section in the *Administrator's Guide* contains details on using the data migrator as well as a comparison between the import/export utility and the data migrator. We are deprecating the import/export utility as of the 20.3 release and encourage customers to use the new data migrator utility.

New Features in Release 20.2

Generic ORDER BY and GROUP BY

In previous releases, ORDER BY and GROUP BY were possible only if there was an index that sorted the rows by the ORDER BY/GROUP BY expressions. Furthermore, it was not possible for queries to include both ORDER BY and GROUP BY. These restrictions are lifted in this release. Note that generic ORDER BY and GROUP BY may consume a lot of driver memory, because of the need to materialize the full query result in driver memory.

SELECT DISTINCT

The DISTINCT keyword is now supported in the SELECT clause. If present, duplicate results will be removed from query result set. As with generic ORDER BY and GROUP BY, SELECT DISTINCT needs to cache the full result set in driver memory.

TTL in Streams API and Multi-Region Tables

Added support for Time to Live (TTL) in the Streams API and in Multi-Region tables. The Rows supplied by the Streams API to subscribers will include the TTL expiration time for the operation from source store. For Multi-Region tables, the TTL expiration time of each row is replicated to other regions, and thus a row in Multi-Region table in any region will expire at the expiration time replicated from the region where and when operation is made.

New Features in Release 20.1

IN Operator

Added support for the IN operator in SQL for Oracle NoSQL Database. The IN operator provides a convenient way to match one of several equality conditions. For example, the query

```
SELECT * FROM Foo WHERE a IN (1, 5, 4)
```

is equivalent to

```
SELECT * FROM Foo WHERE a = 1 OR a = 5 OR a = 4
```

Untyped JSON indexes

Before this release, indexing a field inside a JSON document required that the field have a concrete JSON atomic type - one of integer, long, double, number, string, or boolean. If, for some document, the index field value did not conform to the declared type, index creation or the insertion of that document in a table would fail. This release adds support for the new anyAtomic type as a valid type for indexed JSON fields. A field indexed with the anyAtomic type can be of any valid JSON atomic type and the index will store all the heterogeneous values of that field.

General Availability of Multi-Region Tables

Multi-Region Tables have received several improvements in this release and is now considered a general availability feature.

New additions and improvements:

- Regions can now be added or removed from an existing multi-region table.
- A multi-region table can be dropped in one region and the associated tables can remain in other regions.
- Regions can now be added or removed from an existing multi-region table.
- When creating a multi-region table, the associated tables in remote regions no longer need to be empty.

Note

Multi-Region tables are now only available in the Enterprise Edition.

New Features in Release 19.5

Multi-Region Tables

Multi-Region Tables are a new feature that lets users create "read-anywhere" and "write-anywhere" tables that live in multiple regions, where each region is a separate Oracle NoSQL Database store. This is a preview release to give users early access to the new feature. It is not recommended to deploy this version of multi-region tables in a production environment. A general availability version will be available in a future release. Data stored in multi-region tables created with this release may need to be discarded when upgrading to the general availability version. Please refer to the Admin Guide and related documents for more information on the new feature and see the Known Issues section below for a list of limitations in this release.

Asynchronous Table API Methods

The table API now has asynchronous methods that applications can use to make calls without using a thread to wait for results, which can improve the efficiency of clients that make many concurrent calls.

The client uses a new network protocol that multiplexes multiple calls on the same socket which supports asynchronous operations and reduces the number of socket connections needed. The new client is only compatible with this version of the server, although the new server continues to support older clients.

String manipulation SQL functions

A set of new functions have been added to SQL queries to help with string manipulation: concatenation, substring, trim, ltrim, rtrim, length, contains, starts_with, ends_with, index_of, replace, reverse, upper, lower.

Deprecating "kvproxy drivers"

In the 19.3 release, Oracle NoSQL Database released a database proxy component that lets Oracle NoSQL Database drivers communicate with the Oracle NoSQL Database cluster, called the "HTTP proxy drivers." Prior to the 19.3 release, there was another set of Oracle NoSQL Database drivers that used a thrift based proxy for communication, called the "kvproxy drivers." These drivers are available in Java, C#, Javascript, and Python. The "HTTP proxy drivers" provide richer capabilities and will be usable for both on-prem and cloud applications. The "kvproxy drivers" have been deprecated in this release and are subject to removal from future versions. They should be used with that possibility in mind. The APIs are different between the "kvproxy drivers" and the "HTTP proxy drivers", so existing applications will need to be modified to use the "HTTP proxy drivers." Deprecated drivers will be clearly marked on the Oracle Technology Network.

New Features in Release 19.3

String pattern matching in queries

Queries now support a new built-in function, `regex_like`, which can be used to specify string pattern matching in queries.

Hybrid Cloud Deployment

Oracle NoSQL Database can now be used with the drivers for the Oracle NoSQL Database Cloud Service by deploying the new Oracle NoSQL Database Proxy. The JAR file for the Oracle NoSQL Database Proxy is included in the Enterprise Edition distribution of Oracle NoSQL Database. All users can download the latest version of the JAR from the Oracle Technology Network.

New Features in Release 19.1

JSON datatype support for Full Text Search

Users can create full text search indexes on a JSON field and on the attributes within that JSON field.

Export/Import Enhancement

The Export/Import tool has been enhanced to allow users to migrate their data into Oracle NoSQL Database using either JSON or MongoDB JSON formats.

Deprecating Export/Import support for Oracle Storage Cloud Services

Export/Import support for the Oracle Storage Cloud Service that is part of Oracle Cloud Infrastructure Classic has been deprecated for use as a source or sink. Support for the cloud storage that is part of the new Oracle Cloud Infrastructure will be provided in the future.

New Features in Release 18.3

Querying GeoJson Data

Introduced support for a number of built-in functions in SQL queries to interpret JSON objects that represent geographical locations as specified in the GeoJSON specification (<https://tools.ietf.org/html/rfc7946>).

Support for Namespaces

Starting this release users can create namespaces. Tables can be created for a particular namespace. If users do not specify a namespace at the time of table creation the tables are placed under the default namespace of `sysDefault`. Prior to this release there was no concept of namespace in Oracle NoSQL Database. All tables created using prior releases will be placed in `sysDefault` namespace on upgrade.

IDENTITY Column Support

Users can create a table with an IDENTITY column for which the system will automatically generate numeric values using a sequence generator.

Support for INSERT and UPSERT in SQL Query

Support for INSERT and UPSERT statements has been added in SQL queries.

Support for Sequence Aggregation Functions in SQL Query

The following new functions have been added to support sequence aggregation operations in SQL queries: `seq_count`, `seq_sum`, `seq_avg`, `seq_min`, and `seq_max`.

Admin Web Service

Oracle NoSQL Database Admin can now be started as a web service that processes Admin CLI commands through a REST API over the HTTP and HTTPS protocol. The input and output to the Admin Web Service is in JSON format. This feature is intended to help DevOps and other NoSQL admins automate administrative operations. See [REST API for Administering Oracle NoSQL Database](#).

Data Migrator - Preview

This release includes a preview version of the Data Migrator utility, which allows users to migrate their data into Oracle NoSQL Database that is either in JSON or MongoDB JSON formats. This utility will be integrated into the Oracle NoSQL Database IMPORT/EXPORT utility in future releases. See the `MIGRATOR-README.txt` found in the `doc` subdirectory of the Oracle NoSQL Database package for its usage.

Monitoring Using ELK Framework

Oracle NoSQL Database can be setup to generate logs that can be used by the popular ELK (Elasticsearch - Logstash - Kibana) framework to visually monitor NoSQL clusters.

Support for Encryption at Rest

Oracle NoSQL Database can be setup to encrypt its data using dm-crypt, a kernel-level disk encryption mechanism available for Linux.

New Features in Release 18.1

Support for Group-by and Aggregate Function

This release introduces support for the group-by clause and the following aggregate functions: count(*), count(expr), sum(expr), avg(expr), min(expr) and max (expr).

Joining Tables in the Same Table Hierarchy

Support for joining tables that belong to the same table hierarchy — parent-child table joins — is now available.

Zone Affinity

Users can specify master affinity for the primary zones that are in close proximity to the client application. This feature allows the system to favor placing master nodes in primary zones that have master affinity set so that write operations are directed to nearby zones for more predictable latency.

Distributions

Oracle NoSQL Database comes in three distributions: Community Edition (CE), Enterprise Edition (EE), and Client.

- The CE version is open source. It ships with source code and is released under the Apache License, Version 2.0 (Apache 2.0). It includes the client and server distribution, but does not include some server features.
- The EE version does not include source code and it ships with an Oracle license. The EE version includes the client and server distribution, and includes several features not found in the CE version: Oracle External Tables support, Oracle Wallet support for external password storage, and support for Kerberos authentication.
- The Client version is open source. It ships with source code and is released under the Apache 2.0 License (Apache 2.0). The Client version only contains the implementation of the client API, which may be used to access servers running under CE or EE.

In addition, example code is provided in its own package, which can be used with any of the distributions.

Known Issues

Topics

- [Limitations on KVInsight Monitoring Framework \(Experimental\)](#)
- [Limitations on UNION ALL and SUBQUERY \(Experimental\)](#)
- [Limitations on the Last Write Metadata feature](#)
- [bulkPut with LastWriteMetadata \(LWM\) against older server may write rows that cannot be read by queries](#)
- [Need a minimum of 5GB of free disk space to deploy a storage node that hosts an admin](#)
- [JMX SSL Connection Failure with Recent Java](#)
- [Virtual threads on Java 21 \(performance limitation\)](#)
- [Limitation on Diagnostics Utility](#)
- [Data Verifier is disabled by default](#)
- [Users must manage Admin directory size, can put all admins into "RUNNING,UNKNOWN" state](#)
- [Topology Changes May Fail During Software Upgrades](#)
- [Store With Full Text Search May Become Unsynchronized](#)
- [Subscription Cannot Connect and Fails With InternalException](#)

Limitations on KVInsight Monitoring Framework (Experimental)

Please carefully review the limitations listed below for this feature.

- In this experimental release, there is no https support.
- In this experimental release, KVInsight supports exporting metrics via **CSV** and **Prometheus**.

Limitations on UNION ALL and SUBQUERY (Experimental)

Please carefully review the limitations listed below for this feature.

- This feature is only available in the Java Direct Driver or via SQL cli command. It is not yet available for usage with the SDK.

Limitations on the Last Write Metadata feature

Please carefully review the limitations listed below for this feature.

- This feature is only available in the Java SDK, Go SDK, Python SDK and Java Direct Driver. It is not yet available for Rust, Node.Js, C# SDK.

bulkPut with LastWriteMetadata (LWM) against older server may write rows that cannot be read by queries

When executing bulkPut operations that include LastWriteMetadata (LWM) using the kv-26.1.x driver against a kv-24.3.4 server or earlier (which does not support LWM), the operation may complete without returning an error.

In this mixed-version scenario, the row can be written to the store, but it may not be readable through queries because the older server cannot deserialize the newer format (deserialization

occurs on the Replication Node). Direct key-based retrieval using `get()` may still succeed. Older-version clients may also be unable to read the row via queries.

This behavior is unexpected because LWM is not supported by the target server version; the expected behavior is a clear error indicating an incompatible/unsupported feature.

Workaround Do not use LWM when targeting servers older than the version that supports it.

[KVSTORE-3257]

Need a minimum of 5GB of free disk space to deploy a storage node that hosts an admin

If a Storage Node that hosts an admin is deployed on a system with less than 5GB of free disk space, the following exception will occur:

```
Connected to Admin in read-only mode
oracle.kv.impl.admin.Admin$DBOperationException:Disk limit violated in
Admin environment directory (the-version)
```

Make sure you have at least 5GB of free disk space to successfully deploy a storage node. This same problem will occur when deploying KVLite.

```
$ java -jar $KVHOME/lib/kvstore.jar kvlite -root ./ici
Process exiting due to fault
oracle.kv.impl.fault.CommandFaultException: Disk limit violated in Admin
environment directory
```

We expect to remove this restriction in a future release.

[#26818]

JMX SSL Connection Failure with Recent Java

Newer JDKs enforce stricter TLS hostname verification for JMX/RMI. JConsole and the Collector Service may fail to connect over TLS if the server certificate lacks Subject Alternative Name (SAN) entries matching the hostname used. The default product-generated self-signed certificate has no SAN, so it may be rejected. Customers/admins must provide certificates with appropriate SANs (wildcard for a controlled DNS suffix or explicit host list). A new `makebootconfig/securityconfig` option allows generating certs with SAN hostnames. KVLite includes `-host` in SAN automatically.

[KVSTORE-3133]

Virtual threads on Java 21 (performance limitation)

When the Direct Java Driver API is called from Java 21 virtual threads, performance may be reduced due to virtual-thread pinning in certain blocking and synchronization scenarios in the overall application stack (including third-party dependencies). Customers using Java 21 virtual threads should validate and profile end-to-end performance.

This is primarily a Java platform limitation and is expected to improve in a future JDK (see JDK 25 / JEP 491).

[KVSTORE-2121]

Limitation on Diagnostics Utility

The diagnostics utility may fail to authenticate when an EdDSA (ed25519) SSH key is present in ~/.ssh. In this scenario, it can unexpectedly prompt for the user's password and then fail with an authentication error.

```
Error load ssh keys, Unsupported key type (ssh-ed25519)
```

To enable Ed25519 (EdDSA) support, add `net.i2p.crypto.eddsa` to the runtime classpath. It provides the required EdDSA implementation and prevents the utility from falling back to password-based authentication when an Ed25519 key is present in ~/.ssh.

```
java -cp kv/lib/kvstore.jar:eddsa.jar oracle.kv.impl.util.KVStoreMain
diagnostics
```

[KVSTORE-1799]

Data Verifier is disabled by default

The data verifier is turned off by default. In some cases, the data verifier was using a lot of I/O bandwidth and causing the system to slow down. Users can turn on the data verifier by issuing the following two commands from the Admin CLI:

```
plan change-parameters -wait -all-rns -params
"configProperties=je.env.runVerifier=true"
change-policy -params "configProperties=je.env.runVerifier=true"
```

Note that, if the store has services with preexisting settings for the `configProperties` parameter, then users will need to get the current values and merge them with the new setting to disable the verifier:

```
show param -service rg1-rn1
show param -policy
```

For example, suppose `rg1-rn1` has set the following cleaner parameter:

```
kv-> show param -service rg1-rn1
[...]
configProperties=je.cleaner.minUtilization=40
```

When updating the `configProperties` parameter, the new setting for the verifier should be added, separating the existing settings with semicolons:

```
plan change-parameters -wait -all-rns -params
"configProperties=je.cleaner.minUtilization=40;je.env.runVerifier=true"
```

[KVSTORE-639]

Users must manage Admin directory size, can put all admins into "RUNNING,UNKNOWN" state

Every Admin is allocated a maximum of 3 GB of disk space by default, which is sufficient space for the vast majority of installations. However, under some rare circumstances you might want to change this 3 GB limit, especially if the Admin is sharing a disk with a Storage Node. For more information, see [Managing Admin Directory Size](#).

If Admins run out of disk space, then there will be entries in the Admin logs saying "Disk usage is not within je.maxDisk or je.freeDisk limits and write operations are prohibited" and the output of the ping command will show all the Admins in the "RUNNING,UNKNOWN" state. Follow the procedure described in [Managing Admin Directory Size](#) to bring the Admins back to the "RUNNING,MASTER" or "RUNNING,REPLICA" state.

Below is sample output of the ping command, verify configuration notes and log entries that indicate that Admin ran out of disk space.

```
Pinging components of store OUG based upon topology sequence #308
300 partitions and 1 storage nodes
Time: 2024-03-27 17:13:33 UTC   Version: 24.4.9
Shard Status: healthy: 3 writable-degraded: 0 read-only: 0 offline: 0 total: 3
Admin Status: read-only
Zone [name=Cloud id=zn1 type=PRIMARY allowArbiters=false
masterAffinity=false]   RN Status: online: 3 read-only: 0 offline: 0
Storage Node [sn1] on node1-nosql: 5000   Zone: [name=Cloud id=zn1
type=PRIMARY allowArbiters=false masterAffinity=false]   Status: RUNNING
Ver: 24.4.9 2024-11-21 17:05:14 UTC   Build id: 95fa28ea4441 Edition:
Enterprise   isMasterBalanced: true   serviceStartTime: 2024-03-27
17:12:20 UTC
      Admin [admin1]           Status: RUNNING,UNKNOWN serviceStartTime:
2024-03-27 17:13:22 UTC   stateChangeTime: 2024-03-27 17:13:22 UTC
availableStorageSize: -4 GB
      Rep Node [rg1-rn1]       Status: RUNNING,MASTER sequenceNumber: 245
haPort: 5011 availableStorageSize: -4 GB storageType: HD
serviceStartTime: 2024-03-27 17:12:34 UTC   stateChangeTime: 2024-03-27
17:12:35 UTC
      Rep Node [rg2-rn1]       Status: RUNNING,MASTER sequenceNumber: 242
haPort: 5012 availableStorageSize: -4 GB storageType: HD
serviceStartTime: 2024-03-27 17:12:37 UTC   stateChangeTime: 2024-03-27
17:12:38 UTC
      Rep Node [rg3-rn1]       Status: RUNNING,MASTER sequenceNumber: 241
haPort: 5013 availableStorageSize: -4 GB storageType: HD
serviceStartTime: 2024-03-27 17:12:41 UTC   stateChangeTime: 2024-03-27
17:12:42 UTC

Verification complete, 3 violations, 0 notes found.
Verification violation: [rg1-rn1]   Storage directory on rg1-rn1 has
exceeded storage size
[/scratch/test/nosql/data/disk1 size: 10 GB available: -4 GB]
Verification violation: [rg2-rn1]   Storage directory on rg2-rn1 has
exceeded storage size
[/scratch/test/nosql/data/disk2 size: 10 GB available: -4 GB]
Verification violation: [rg3-rn1]   Storage directory on rg3-rn1 has
exceeded storage size
[/scratch/test/nosql/data/disk3 size: 10 GB available: -4 GB]
```

```

2024-03-27 17:13:12.643 UTC SEVERE - [admin1] JE: Disk usage is not within
the je.freeDisk limit and write operations are prohibited:
maxDiskLimit=3,221,225,472 freeDiskLimit=5,368,709,120
adjustedMaxDiskLimit=3,221,225,472
maxDiskOverage=-3,220,031,078 freeDiskShortage=4,543,057,920
diskFreeSpace=825,651,200
availableLogSize=-4,543,057,920 totalLogSize=1,194,394 activeLogSize=1,194,394
reservedLogSize=0 protectedLogSize=0 protectedLogSizeMap={}

```

[#26922]

Topology Changes May Fail During Software Upgrades

Making modifications to the store topology that include partition migration may fail if the modifications are performed while the store is being upgraded to a new software version. If you run a plan to deploy a new topology and the plan fails with problems during partition migration, check if the nodes of the store are running different software versions, and upgrade any nodes running old versions before retrying the plan.

Modifying a topology using one of the following topology commands can result in the need for partition migration. Deploying the resulting topology with the 'plan deploy-topology' command can then fail if the plan is performed during a store software version upgrade. The topology commands that can produce partition migrations are:

- topology change-repfactor
- topology contract
- topology rebalance
- topology redistribute

Other topology commands do not produce partition migration and do not cause this problem.

If a topology deployment fails, you can tell if it is related to partition migrations during a software version upgrade by looking for errors like the following:

```

Plan 24 ended with errors. Use "show plan -id 24" for more information
Plan Deploy Topo
Id:                24
State:             ERROR
Attempt number:    1
Started:          2025-04-10 15:19:59 UTC
Ended:            2025-04-10 15:24:48 UTC
Plan failures:
  Failure 1: 17/MigratePartition PARTITION-2 from rg1 to rg2
  failed. target=rg2-rn1 state=ERROR java.lang.Exception:
  Migration of PARTITION-2 failed. Giving up after 10 attempt(s)

```

If you see a plan failure involving partition migrations like this, particularly if there are similar failures for all partition migration tasks, use the 'ping' or 'verify topology' commands to display information about the store and check to see if different storage nodes are running different major or minor software versions. If so, upgrade the nodes running the older software to the latest version before retrying the 'plan deploy-topology' command.

Store With Full Text Search May Become Unsynchronized

A store that has enabled support for Full Text Search may, on rare occasions, encounter a bug in which internal components of a master Replication Node become unsynchronized, causing updates from that Replication Node to stop flowing to the Elasticsearch engine. This problem will cause data to be out of sync between the store and Elasticsearch.

When the problem occurs, the Elasticsearch indices stop being populated. The problem involves the shutdown of the feeder channel for a component called the TextIndexFeeder, and is logged in the debug logs for the Replication Node. For example:

```
2018-03-16 11:23:46.055 UTC INFO [rg1-rn1] JE: Inactive channel:
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78(2147483647)
forced close. Timeout: 10000ms.
2018-03-16 11:23:46.059 UTC INFO [rg1-rn1] JE: Shutting down feeder for
replica TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 Reason:
null write time: 32ms Avg write time: 100us
2018-03-16 11:23:46.060 UTC INFO [rg1-rn1] JE: Feeder Output for
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 soft shutdown
initiated.
2018-03-16 11:23:46.064 UTC WARNING [rg1-rn1] internal exception Expected
bytes: 6 read bytes: 0
com.sleepycat.je.utilint.InternalException: Expected bytes: 6 read bytes: 0
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.loopInternal(Subscription
Thread.java:719)
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.run(SubscriptionThread.ja
va:180)
Caused by: java.io.IOException: Expected bytes: 6 read bytes: 0
    at
com.sleepycat.je.rep.utilint.BinaryProtocol.fillBuffer(BinaryProtocol.java:446
)
    at
com.sleepycat.je.rep.utilint.BinaryProtocol.read(BinaryProtocol.java:466)
    at
com.sleepycat.je.rep.subscription.SubscriptionThread.loopInternal(Subscription
Thread.java:656)
    ... 1 more

2018-03-16 11:23:46.064 UTC INFO [rg1-rn1] SubscriptionProcessMessageThread
soft shutdown initiated.
2018-03-16 11:23:46.492 UTC INFO [rg1-rn1] JE: Feeder output for
TextIndexFeeder-rg1-rn1-b4e92291-3c73-4128-9557-62dbd4e9ac78 shutdown. feeder
VLSN: 4,066 currentTxnEndVLSN: 4,065
```

If the TextIndexFeeder channel is shutdown, then the user can restore it by creating a dummy full text search index. Here is an example of how you can do that.

Assuming that Elasticsearch is already registered, execute the following commands from the Admin CLI:

```
execute 'CREATE TABLE dummy (id INTEGER,title STRING,PRIMARY KEY (id))'
execute 'CREATE FULLTEXT INDEX dummytextindex ON dummy (title)'
execute 'DROP TABLE dummy'
```

Note that *dummy* is the name of a temporary table that should not exist previously.

Creating a full text search index reestablishes the channel from the store to Elasticsearch and ensures that data is synced up to date.

[#26859]

Subscription Cannot Connect and Fails With InternalException

If a master transfer occurs due to a failure after the publisher is started and before a subscriber connects, an `InternalException` can occur when the subscriber tries to connect. The exception message will read "Failed to connect, will retry after sleeping 3000 ms". Restart the publisher to work around this problem.

[#27723]