

# Oracle Lite

ユーザーズ・ガイド

リリース 4.0.1

2000 年 10 月

部品番号 : J02406-01

ORACLE®

---

Oracle Lite ユーザーズ・ガイド, リリース 4.0.1

部品番号 : J02406-01

原本名 : Oracle Lite User's Guide, Release 4.0.1

原本部品番号 : A86160-01

Copyright © 2000, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

\* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

---

# 目次

はじめに .....	vii
<b>1 作業の準備</b>	
Oracle Lite の概要 .....	1-2
Oracle Lite のインストール .....	1-2
データベースの移行 .....	1-2
初期データベースの使用 .....	1-3
<b>2 アプリケーションの開発</b>	
開発インタフェースの概要 .....	2-2
ODBC .....	2-3
OCI/OCA .....	2-3
JDBC .....	2-3
オブジェクト・カーネル API (OKAPI) .....	2-4
Java アクセス・クラス (JAC) .....	2-4
Oracle Lite のトランザクション・サポート .....	2-5
原子性 .....	2-5
一貫性 .....	2-5
分離 .....	2-5
永続性 .....	2-7
ロック .....	2-7
デフォルト分離レベルの変更 .....	2-7
サポートされる分離レベルとカーソル・タイプの組合せ .....	2-7
アプリケーションのチューニング .....	2-8

### 3 データベースの操作

新規データベースの作成 .....	3-2
ODBC Administrator によるデータ・ソース名の作成 .....	3-2
Oracle8 Navigator による新規データベースの作成 .....	3-2
コマンドライン・ユーティリティによる新規データベースの作成 .....	3-3
新規データベースへの接続 .....	3-3
複数ユーザーの作成 .....	3-4
事前定義のロール .....	3-4
ユーザーの作成 .....	3-5
ユーザーの削除 .....	3-5
パスワードの変更 .....	3-5
ロールの付与 .....	3-6
権限の付与 .....	3-6
ロールの取消し .....	3-6
権限の取消し .....	3-6
デモ表の作成 .....	3-7
データベースへの値の移入 .....	3-7
SQL*Plus によるデータベースへの値の移入 .....	3-7
Oracle8 Navigator による新規データベースへの値の移入 .....	3-7
OLLOAD ユーティリティによるデータベースへの値の移入 .....	3-10
SQL*Loader によるデータベースへの値の移入 .....	3-10
データベースのバックアップ .....	3-10
データベースの暗号化と復号化 .....	3-11

### 4 アプリケーションの配布

Deployment Wizard .....	4-2
ヘルプ・ファイル .....	4-3
Deployment Utility .....	4-3
手動による配布 .....	4-4
手動による配布をする際に使用するパラメータ .....	4-4
コア DLL と EXE の配布 .....	4-5
ODBC ドライバのセットアップ .....	4-5
JDBC ドライバの配布 .....	4-6
Java クラスの配布 .....	4-6
レプリケーション API の配布 .....	4-7

追加ファイルの配布（オプション） .....	4-8
------------------------	-----

## 5 サンプル・アプリケーションの使用

概要 .....	5-2
Access サンプル・アプリケーションに関する注意 .....	5-3
BLOB Manager サンプル・アプリケーションに関する注意 .....	5-4
Delphi サンプル・アプリケーションの実行 .....	5-4
PowerBuilder サンプル・アプリケーションの実行 .....	5-6
Visual Basic サンプル・アプリケーションの実行 .....	5-8

## 6 Oracle Lite レプリケーションの使用

レプリケーションの概要 .....	6-2
作業の準備 .....	6-2
ステップ 1: レプリケーションのサーバー・サポートの確認 .....	6-3
ステップ 2: レプリケーション・エンジン・サポートの設定 .....	6-3
ステップ 3: 高速リフレッシュ用スナップショット・ログの作成 .....	6-3
ステップ 4: マスター表のレプリケーション準備 .....	6-4
ステップ 5: レプリケーション接続の設定 .....	6-4
マスター・サイトへの接続の設定 .....	6-4
Oracle8 Navigator でのマスター・サイト接続の追加 .....	6-5
ステップ 6: スナップショットの作成 .....	6-5
スナップショットの検証 .....	6-6
レプリケーションでサポートされているデータ型 .....	6-7

## A POLITE.INI のデータベース・パラメータ

POLITE.INI ファイルの概要 .....	A-2
POLITE.INI のパラメータ .....	A-2
CacheSize .....	A-2
DatabaseID .....	A-2
DbCharEncoding .....	A-2
MAXINDEXCOLUMNS .....	A-2
NLS_Date_Format .....	A-2
日付書式 .....	A-3
日付書式の例 .....	A-5
NLS_Locale .....	A-5
ReverseJoinOrder .....	A-6

SharedAddress .....	A-6
SQLCompatibility .....	A-7
SuggestedSharedAddress .....	A-7
SVTRACE .....	A-7
TempDB .....	A-7
TempDir .....	A-8
POLITE.INI ファイルのサンプル .....	A-8

## B システム・カタログ

ALL_COL_COMMENTS .....	B-2
ALL_CONSTRAINTS .....	B-2
ALL_CONS_COLUMNS .....	B-3
ALL_INDEXES .....	B-3
ALL_IND_COLUMNS .....	B-4
ALL_OBJECTS .....	B-4
ALL_SEQUENCES .....	B-5
ALL_SYNONYMS .....	B-5
ALL_TABLES .....	B-5
ALL_TAB_COLUMNS .....	B-7
ALL_TAB_COMMENTS .....	B-8
ALL_USERS .....	B-8
ALL_VIEWS .....	B-8
CAT .....	B-9
COLUMN_PRIVILEGES .....	B-9
DUAL .....	B-9
SNAPSHOTS .....	B-10
TABLE_PRIVILEGES .....	B-12
USER_OBJECTS .....	B-12

## C データベースのツールとユーティリティ

CREATEDB .....	C-2
DECRYPDB .....	C-3
ENCRYPDB .....	C-4
MIGRATE .....	C-7
ODBC Administrator と Oracle Lite ODBC ドライバ .....	C-8
ODBC Administrator を使用した DSN の追加 .....	C-9
読取り専用メディア (CD-ROM) を指す DSN の追加 .....	C-10

ODBC 拡張機能 .....	C-10
ODBINFO .....	C-11
OLitRM40 .....	C-13
OLLOAD .....	C-14
Oracle8 Navigator .....	C-17
Oracle8 Navigator の使用方法 .....	C-17
Oracle8 Navigator による EMP 表の検査 .....	C-18
Oracle Lite Designer .....	C-18
REMOVEDB .....	C-18

## D パフォーマンス・チューニング

単一表問合せの最適化 .....	D-2
結合問合せの最適化 .....	D-2
ORDER BY および GROUP BY 句による最適化 .....	D-4

## 索引





---

# はじめに

『Oracle Lite ユーザーズ・ガイド』では、Oracle Lite およびそのコンポーネントを紹介し  
ます。また、Oracle Lite を使用してデータベースを開発、配布およびメンテナンスする方法を  
示します。

このガイドの内容は次のとおりです。

## 第 1 章「作業の準備」

Oracle Lite の概要を紹介し、データベースの移行方法と初期データベースの使用方を説明します。

## 第 2 章「アプリケーションの開発」

Oracle Lite データベース・アプリケーションの開発方法に関するガイドラインを提供します。

## 第 3 章「データベースの操作」

Oracle Lite データベースの操作の概要を示します。データベースの作成、データベースへの接続、ユーザーの作成、さらにデータベースの管理について説明します。

## 第 4 章「アプリケーションの配布」

Oracle Lite データベースをアプリケーションとともに配布する開発者のために、Oracle Lite データベースで提供されているツールを説明します。

## 第 5 章「サンプル・アプリケーションの使用」

Oracle Lite データベースで提供されているサンプル・アプリケーションについて説明します。

## 第 6 章「Oracle Lite レプリケーションの使用」

Oracle Lite のデータ・レプリケーション機能およびレプリケーションの設定方法について説明します。

## 付録 A 「POLITE.INI のデータベース・パラメータ」

POLITE.INI ファイルで設定できるデータベース・パラメータをリストします。

## 付録 B 「システム・カタログ」

システム・カタログ内のオブジェクト型について説明します。

付録 C 「データベースのツールとユーティリティ」	Oracle Lite で使用できるデータベース・ツールとユーティリティを説明します。
付録 D 「パフォーマンス・チューニング」	SQL 問合せのパフォーマンスを向上させるためのヒントを提供します。

---

---

**注意：** このマニュアルでは、「Oracle Server」および「Oracle」は、Oracle7 リリース 7.3 「以上」を意味します。「Oracle8」および「Oracle8 Server」は、Oracle8 リリース 8.0.5 以上を意味します。

---

---

---

## 作業の準備

この章では、Oracle Lite およびそのコンポーネントの概要を説明します。説明する内容は次のとおりです。

- [Oracle Lite の概要](#)
- [Oracle Lite のインストール](#)
- [データベースの移行](#)
- [初期データベースの使用](#)

## Oracle Lite の概要

Oracle Lite は、ラップトップ、携帯端末、PDA およびその他の情報機器用として設計開発されたコンパクトな Java 対応のデータベースです。

---

---

**重要：** Oracle Lite の概要は、『Oracle8i Lite 開発者ロードマップ』を参照してください。ロードマップと Oracle Lite の各コンポーネントのマニュアルは、Oracle8i Lite のマニュアル・ページにあります。

---

---

## Oracle Lite のインストール

Oracle Lite とそのコンポーネントは、Oracle8i Lite の CD-ROM からインストールできます。インストール手順は、『Oracle8i Lite インストレーション・ガイド』を参照してください。

## データベースの移行

既存の Oracle Lite データベースは、簡単にこのリリースに移行できます。

### デフォルト DSN の POLITE の変更

Oracle Lite をインストールし移行する前に、ODBC デフォルト・データ・ソース名 (DSN) の POLITE を変更する必要があります。DSN は、次のようにして変更します。

- 32 ビットの ODBC Administrator を起動し、「ユーザー DSN」タブを選択し、DSN 名の POLITE をダブルクリックします。
- 「Oracle Lite ODBC ドライバセットアップ」ダイアログ・ボックスで、DSN を POLITEOLD に変更します。

---

---

**重要：** 移行できるのはユーザーの DSN のみです（システムの DSN は移行できません）。

---

---

### Oracle Lite 2.5 および 3.0 からの移行

Oracle Lite 2.5 または 3.0 からこのリリースの Oracle Lite に直接移行できません。

リリース 2.5 から移行する場合は、Oracle Lite 3.0 で提供されている MIGRATDB ユーティリティを使用して、まずリリース 3.0 に移行する必要があります。詳細は、『Oracle Lite ユーザーズ・ガイド リリース 3.0』を参照してください。

リリース 3.0 から移行する場合は、MIGRATE ユーティリティ（リリース 3.5 で提供されているもので、今回のリリースの MIGRATE ユーティリティではありません）または

MIGRATE36 ユーティリティ（リリース 3.6 で提供）を使用して、まずリリース 3.5 または 3.6 に移行する必要があります。詳細は、リリース 3.5 または 3.6 用の『Oracle Lite ユーザーズ・ガイド』を参照してください。

### Oracle Lite 3.5 および 3.6 からの移行

リリース 3.5 または 3.6 のデータベースは、MIGRATE ユーティリティを使用して Oracle Lite に移行できます。[MIGRATE](#) ユーティリティは `Oracle_Home¥lite¥migrate` ディレクトリにあります。デフォルトの `Oracle_Home` ディレクトリは、次のとおりです。

- ORAWIN95（Windows 95/98 の場合）
- ORANT（Windows NT の場合）

データベースを移行するには、次の手順に従います。

1. まず、Oracle Lite をインストールします。
2. 次に、DOS プロンプトで、`Oracle_Home¥lite¥migrate` ディレクトリに移動します。
3. 次のように入力します。

```
MIGRATE DSN DBName
```

`DSN` は Oracle Lite 3.5 または 3.6 のデータ・ソース名で、`DBName` はデータベースの名前またはデータベースのパスと名前です。例を次に示します。

```
MIGRATE polite db1
```

移行が正常に終了すると、Oracle Lite 3.5 または 3.6 の `DSN` は、今回のリリースの DLL にリンクするように変更されます。

移行済のデータベースは元のデータベースと同じディレクトリに入れられ、元と同じ名前になります。元のデータベースのバックアップ・コピーも作成されます。ファイル名には **.35** または **.36** という拡張子が付けられ、同じ場所に格納されます。詳細は、[付録 C「データベースのツールとユーティリティ」](#)、「[MIGRATE](#)」を参照してください。

`DSN` として `POLITE` を使用する場合は、移行後に、**POLITE.ODB** ファイルを `Oracle_Home¥OLDB40` ディレクトリに移す必要があります。

## 初期データベースの使用

Oracle Lite をインストールすると、ODBC データ・ソース名 `POLITE` が作成され、初期データベース **POLITE.ODB** が対応付けられます。`DSN POLITE` 用の新規データベースの場所は、`Oracle_Home¥OLDB40` に設定されます。

インストール中に、2 つのデフォルト・ユーザー `SYSTEM` と `SCOTT` が設定されます。`SYSTEM` にはデータベース権限がすべて含まれ、`NULL` パスワードが指定されます。`SCOTT` には `DDL` 権限が含まれ、パスワード「tiger」が割り当てられます。`ALTER USER` コマンドを使用すると、`SYSTEM` のパスワードを作成したり `SCOTT` のパスワードを変更できます。

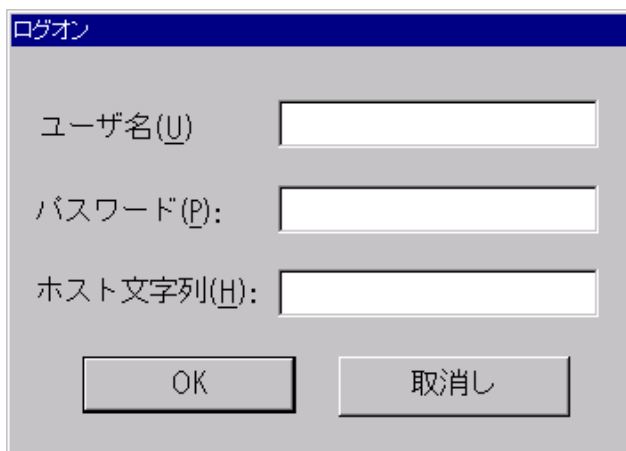
(サンプルの構文を次の項で説明します)。デフォルト・ユーザー名を使用するか、または独自のユーザー名を設定できます。

---

**注意：** 初期データベースを使用する前に、『Oracle Lite SQL リファレンス』をお読みください。このマニュアルには、Oracle Lite データベース情報の管理に使用される構造化問合せ言語 (SQL) が説明されています。

---

Oracle Lite の初期データベースには、SQL\*Plus などのアプリケーションを使用して接続できます。SQL\*Plus は、次のログオン用ダイアログを表示します。

A screenshot of the Oracle Lite SQL\*Plus logon dialog box. The dialog has a title bar labeled 'ログオン' (Logon). It contains three input fields: 'ユーザ名(U)' (Username), 'パスワード(P):' (Password), and 'ホスト文字列(H):' (Host string). Below the input fields are two buttons: 'OK' and '取消し' (Cancel).

ユーザー名として「SYSTEM」と入力します。次に、パスワードには任意の文字列を入力します。(SQL\*Plus ではパスワード・フィールドに文字列を入力する必要があります。SYSTEM ユーザーには最初は NULL パスワードが指定されています。) 次に、ホスト文字列として「ODBC:POLITE」と入力します。

次のコマンドを入力すると、システムにパスワードを割り当てることができます。

```
ALTER USER SYSTEM IDENTIFIED BY <password>
```

---

**注意：** 詳細は、第3章「データベースの操作」の「パスワードの変更」を参照してください。

---

ODBC アプリケーションから初期データベースに接続するときは、デフォルトの ODBC DSN の POLITE を使用します。

---

## アプリケーションの開発

この章では、Oracle Lite データベース・アプリケーションの開発方法に関するガイドラインを提供します。説明する内容は次のとおりです。

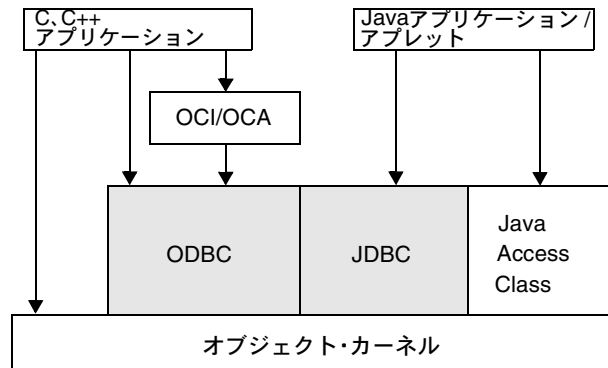
- [開発インタフェースの概要](#)
- [ODBC](#)
- [OCI/OCA](#)
- [JDBC](#)
- [オブジェクト・カーネル API \(OKAPI\)](#)
- [Java アクセス・クラス \(JAC\)](#)
- [Oracle Lite のトランザクション・サポート](#)
- [アプリケーションのチューニング](#)

## 開発インタフェースの概要

Oracle Lite では、データベース・アプリケーションの開発用に次のインタフェースが用意されています。

- リレーショナル・データベース開発用：
  - ODBC
  - OCI/OCA
  - JDBC
- オブジェクト・データベース開発用：
  - オブジェクト・カーネル API (OKAPI)
  - Java アクセス・クラス (JAC) API

これらのモデルは、単独で、または組み合わせて使用できます。たとえば、JAC コールと JDBC コールを同一プログラムで使用できます。次の図は、Oracle Lite アプリケーション・インタフェースを示します。背景が灰色のインタフェースは、リレーショナル・モデルをサポートします。





## ODBC

Microsoft の Open Database Connectivity (ODBC) インタフェースは、SQL データベース・アクセス用の手続き型コール・レベル・インタフェースで、ほとんどのデータベース・ベンダーによりサポートされています。ODBC インタフェースは、アプリケーションがデータベースに接続し、実行時に SQL 文を準備して実行し、問合せ結果を取り出すための一連の機能を指定したものです。Oracle Lite は、レベル 3 準拠の ODBC 2.0 をサポートし、このサポートは、ODBC コールを解釈して Oracle Lite に渡す Oracle Lite ODBC ドライバを介して行われます。

ODBC の詳細は、次を参照してください。

- Microsoft の ODBC 関連マニュアル。
- Oracle Lite ODBC サンプル・アプリケーション。サンプル・アプリケーションの場所については、[第 5 章「サンプル・アプリケーションの使用」](#)を参照してください。
- [付録 C「データベースのツールとユーティリティ」](#)の「[ODBC Administrator と Oracle Lite ODBC ドライバ](#)」。

## OCI/OCA

Oracle Call Interface (OCI) は、C で作成されたアプリケーションと Oracle サーバーとの対話を可能にするインタフェースです。OCI により、SQL 文の処理やオブジェクト操作など、Oracle サーバーとの間で可能な広範なデータベース操作をプログラムで実行できます。OCBC 用の Oracle Open Client Adapter (OCA) により、アプリケーションから様々なデータベースに一貫性のある方法でアクセスできるようになります。OCA では、ODBC ドライバを提供する Oracle データベースおよび非 Oracle データベースへの読取り専用アクセスと、読取り / 書き込みアクセスがサポートされます。

## JDBC

Java Database Connectivity (JDBC) インタフェースは、Java アプリケーションが SQL データベースに対して ODBC のようなインタフェースを利用できるようになる一連の Java クラスのことを指します。JDBC は JDK (Java Developer's Kit) コアの一部で、リレーショナル・データベースに対してオブジェクト・インタフェースを提供します。Oracle Lite は、Oracle Lite のタイプ 2 JDBC ドライバを介して JDBC をサポートします。このドライバは JDBC コールを解釈して Oracle Lite に渡します。JDBC と Oracle Lite の詳細は、『Oracle Lite Java 開発者ガイド』を参照してください。

## オブジェクト・カーネル API (OKAPI)

OKAPI は、Oracle Lite のオブジェクト・カーネルに対するアプリケーション・プログラミング・インタフェース (API) です。OKAPI には、次のデータベース機能のサポートが含まれています。

- 実行時のクラス作成とクラス情報へのアクセス
- オブジェクトの個別性とナビゲーションを基にした直接的オブジェクト・アクセス
- オブジェクトのクラスタ化とグループ化
- クラスとそのサブクラスの間合せ
- オブジェクトのネーミングとオブジェクト間の関連
- バイナリ・ラージ・オブジェクト (BLOB) データ
- トランザクションおよびクラッシュ・リカバリ

詳細は、『Oracle Lite オブジェクト・カーネル API リファレンス』を参照してください。

## Java アクセス・クラス (JAC)

Java アクセス・クラス (JAC) は Java クラスのコレクションで、これを使用すると、Oracle Lite データベースに格納されている永続情報に Java アプリケーションからアクセスできます。JDBC を使用したデータベースへのアクセス方法と違い、JAC ではデータをリレーショナル・モデルにマップする必要はありません。JAC パッケージを使用すると、Oracle Lite の Java アプリケーションで次の機能が使用できます。

- Oracle Lite オブジェクト・スキーマの作成
- Oracle Lite のスキーマをモデル化するための Java プロキシ・クラスの生成
- Java 永続オブジェクトの作成 / 削除、アクティブ化 / 非アクティブ化、属性値取出しと変更および間合せ
- Java ストアド・プロシージャ

詳細は、『Oracle Lite Java 開発者ガイド』を参照してください。

## Oracle Lite のトランザクション・サポート

アプリケーションが Oracle Lite データベースに接続すると、そのデータベースとの間でトランザクションが開始されます。Oracle Lite データベースに対する最大接続数は 32 です。アプリケーションによっては、Oracle8 Navigator などのように、データベースに対して複数の接続を持つものがあります。Oracle Lite データベースへの接続ごとに、それぞれのトランザクションがメンテナンスされます。

### 原子性

トランザクションとは、SELECT、UPDATE、DELETE および INSERT などの一連のデータベース操作のことです。操作はすべて、成功してコミットされるか、ロールバックされます。これを、トランザクションの原子性プロパティといいます。

Oracle Lite では、データベースのコミットが実行されるまで実際のデータベース・ファイルを更新しないことにより原子性を実装しています。コミット時に一時ロールバック・ログ・ファイルが作成され、その後にデータベース・ファイルが更新されます。停電などによりコミットが中断された場合、次の接続時にログからデータベースが復旧されます。

Oracle Lite には、オプションの Recovery Manager (OlitRM40) が含まれています。このマネージャは、接続をオープンしたアプリケーションが適切な切断操作なしで終了した場合に、自動的にトランザクションをロールバックします。詳細は、[付録 C「データベースのツールとユーティリティ」](#)の「[OLitRM40](#)」を参照してください。

### 一貫性

トランザクションは、データベースの一貫性を保ちます。トランザクションは、データベースをある一貫した状態から別の一貫した状態に変換しますが、この際、すべての中間点で一貫性を保つ必要はありません。Oracle Lite では、制約に違反、つまり一貫性に違反するトランザクションはコミットできません。

### 分離

トランザクションは、相互に分離されています。多くのトランザクションが同時に実行されても、あるトランザクションによる更新はそのトランザクションがコミットするまで他のトランザクションから隠されています。Oracle Lite がサポートするトランザクション分離レベルは、次のとおりです。

分離レベル	説明
Read Committed	<p>この分離レベルでは、最新のコミット済データの一時スナップショットを作成する SELECT 文をロックする必要はありません。READ COMMITTED トランザクションは、SELECT 文により要求された表を更新したり、その表に新規データを挿入する他のトランザクションによってブロックされることはありません。結果的に、同じ SELECT 文を使用して異なるデータのセットを返すことができます。</p> <p>FOR UPDATE 句を含む SELECT 文は、REPEATABLE READ 分離レベルで実行されているかのように実行されます。</p> <p>Oracle Lite では、Java ストアド・プロシージャは SELECT 文によって実行されます。Java ストアド・プロシージャを実行するトランザクションが READ COMMITTED の分離レベルにあり、Java ストアド・プロシージャによりデータベースが更新される場合は、Java ストアド・プロシージャを実行する SELECT 文には、FOR UPDATE 句が必要です。FOR UPDATE 句を指定しなかった場合、Oracle Lite からエラーが発行されます。</p>
Repeatable Read	<p>この分離レベルでは、問合せはその問合せにより返されたすべての行に対して読取りロックを取得します。問合せ自身やその表に定義された索引が複雑なため、または問合せオプティマイザが選択した実行計画のために、より多くの行が読取りロックされる場合があります。さらに短時間の間に読取りロックされる行もあるため、REPEATABLE READ 分離レベルのほうが READ COMMITTED 分離レベルのトランザクションよりも有効性は低くなります。</p> <p>あるトランザクションが、問合せに指定された検索条件を満たす行を追加挿入すると、その同じ問合せのサブシーケンスが実行され、検索条件を満たす行がさらに返されます。</p> <p>問合せ内に FOR UPDATE 句を使用すると、選択したカレント行に短時間の更新ロックが取得されます。行が更新されると、ロックが排他的ロックに変換されます。排他的ロックは、READ COMMITTED 以外の分離レベルで実行している他のトランザクションがこの行にアクセスしないようにブロックします。行は更新されないが次の行がフェッチされた場合、更新ロックは読取りロックにダウングレードされ、他のトランザクションがこの行を読み取れるようになります。</p>
Serializable	<p>この分離レベルでは、問合せにかかわるすべての表に対して共有ロックを取得します。同じトランザクション内の問合せを反復実行すると、同じ行セットが返されます。問合せ内の表にある行を更新しようとするトランザクションはすべてブロックされます。</p>
Single User	<p>この分離レベルでは、データベースに対して 1 つの接続しか許可されません。トランザクションにはロックがなく、消費するメモリーも少なくてすみます。</p>

分離レベルの詳細、具体的にはトランザクションの分離レベルを定義する「内容を保証しない読取り」、「非リピータブル・リード」および「仮読取り」という用語については、ODBCのマニュアルを参照してください。

## 永続性

トランザクションでは、永続性が保証されています。つまり、いったんトランザクションがコミットすると、その後どの時点でシステムに障害が発生してもその変更はすべてデータベース・ファイル内で永続的になります。何らかのシステム障害によりトランザクションがコミットまたはロールバック時に失敗した場合、データベースを一貫した状態にリストアするには、ロールバック・ログ・ファイルが必要です。

## ロック

Oracle Lite では、行レベル・ロックをサポートしています。行が読み取られるときはいつも、読取りロックされます。行が変更されるときはいつも、書き込みロックされます。異なるトランザクションから、読取りロックされた同一の行を読み取ることができます。ただし、別のトランザクションから書き込みロックされた行にアクセスできません。

## デフォルト分離レベルの変更

Oracle Lite では、READ COMMITTED 分離レベルがデフォルトです。

ODBC Administrator を使用するか、または **ODBC.INI** ファイルに手動で次の行を含めると、データ・ソース名 (DSN) のデフォルト分離レベルを変更できます。

```
IsolationLevel = XX
```

XX は、RC、RR、SR または SU です。

また、次の SQL 文を使用しても、トランザクションの分離レベルを設定できます。

```
SET TRANSACTION ISOLATION LEVEL <ISOLATION_LEVEL>;
```

*ISOLATION\_LEVEL* は、READ COMMITTED、REPEATABLE READ、SERIALIZABLE または SINGLE USER のいずれかです。

詳細は、「[サポートされる分離レベルとカーソル・タイプの組合せ](#)」を参照してください。

## サポートされる分離レベルとカーソル・タイプの組合せ

次の表に、サポートされる分離レベルとカーソル・タイプの組合せを示します。分離レベルが左側の列、カーソル・タイプが一番上の行に示してあります。「S」はサポート (supported)、「U」はサポートなし (unsupported) を示します。

	Forward Only	Static	Keyset Driven	Dynamic
Read Committed	S	S	U	U
Repeatable Read	S	U	S	S
Serializable	S	U	S	S
SINGLE USER	S	S	S	S

サポートされていない組合せは、エラー・メッセージが生成されます。

## アプリケーションのチューニング

アプリケーション設計のチューニングは、アプリケーションを実装する前に行うのが理想的です。設計を開始する前に、利用できる Oracle Lite の機能について学習し、どの機能が要件に最適であるかを検討する必要があります。また、Oracle データベース管理者と協力して、開発するアプリケーションに Oracle のマスター・サイトが対応するようにマスター・サイトをチューニングする方法を検討する必要があります。[付録 D「パフォーマンス・チューニング」](#)に設計する際に考慮する必要のあるいくつかのヒントが記載されています。

---

## データベースの操作

この章では、Oracle Lite データベースの作成、データベースへの接続、ユーザーの作成、さらにデータベースの管理など、データベースの操作の概要を説明します。説明する内容は次のとおりです。

- 新規データベースの作成
- 新規データベースへの接続
- 複数ユーザーの作成
- デモ表の作成
- データベースへの値の移入
- データベースのバックアップ
- データベースの暗号化と復号化

## 新規データベースの作成

データ・ソース名 POLITE を使用して新規データベースを作成すると、この新規データベース・ファイルは `Oracle_Home\OLDB40` ディレクトリに保存されます。メンテナンスに便利のように、すべてのデータベースに同じデータベース・ディレクトリを使用することをお勧めします。

---

---

**注意：** 新しく作成されたデータベースにはすべて、ユーザー SYSTEM が含まれますが、このユーザーのパスワードは NULL です。

---

---

新規データ・ソース名は、ODBC Administrator を使用して作成できます。詳細は、「[ODBC Administrator によるデータ・ソース名の作成](#)」を参照してください。

新規データベースは、Oracle8 Navigator またはコマンドライン・ユーティリティを使用して作成できます。

---

---

**注意：** インストールが簡単に行えるように、Oracle8 Navigator を使用して POLITE のデータ・ソース名ですべてのデータベース・ファイルを作成することをお勧めします。Oracle8 Navigator によって作成されるファイルはすべて、Windows レジストリに登録されますが、コマンドラインから作成されるファイルや、その他の ODBC データ・ソース名で作成されるファイルは登録されません。

---

---

## ODBC Administrator によるデータ・ソース名の作成

ODBC Administrator は Microsoft 社が提供するツールで、Windows 95/98 および Windows NT の `ODBC.INI` ファイルおよびそれに関連付けられたレジストリ・エントリを管理します。このツールにより、データ・ソース名を追加して、そのデータ・ソース名のデフォルト・データベースとして使用するデータベース・ファイルを指定できます。ODBC Administrator の詳細と、このツールを使用してデータ・ソース名を作成する手順は、[付録 C 「データベースのツールとユーティリティ」](#) の「[ODBC Administrator と Oracle Lite ODBC ドライバ](#)」を参照してください。

## Oracle8 Navigator による新規データベースの作成

新規データベースは、Oracle8 Navigator を使用して作成できます。Oracle8 Navigator の機能、およびこの製品を使用してデータベースに接続し、データベース・オブジェクトにアクセスする方法は、[付録 C 「データベースのツールとユーティリティ」](#) の「[Oracle8 Navigator](#)」を参照してください。



Oracle8 Navigator を使用して新規データベースを作成するには、次の手順を実行します。

1. Oracle8 Navigator が起動していない場合は、起動します。
2. 「Oracle Lite データベース」フォルダを右ボタンでクリックします。
3. 「新規作成」をクリックします。  
新規データベースのプロパティ・シートが表示されます。
4. 新規データベースの名前を入力します。Oracle Lite がデータベースを作成し、そのデータベースに自動的に接続します。

Oracle8 Navigator を使用して作成されるデータベース・ファイルは、すべて *Oracle\_Home*\OLDDB40 ディレクトリ内に作成され、POLITE データ・ソース名に関連付けられます。

## コマンドライン・ユーティリティによる新規データベースの作成

新規データベースをコマンドラインから作成するには、CREATEDB ユーティリティを使用します。

例を次に示します。

```
CREATEDB polite dbname
```

*polite* は DSN 名で、*dbname* は新規データベースの名前です。

詳細は、付録 C 「データベースのツールとユーティリティ」の「[CREATEDB](#)」を参照してください。

## 新規データベースへの接続

SQL\*Plus を使用して新規データベースに接続するには、ダミーのパスワードと接続文字列を使用してユーザー SYSTEM として接続します。

```
ODBC:POLITE:
```

POLITE を、定義済の ODBC データ・ソース名で置き換えることもできます。

データ・ソース名に複数のデータベースが関連付けられている場合は、次の書式を使用します。

```
ODBC:dsn:dbname
```

*dsn* はデータ・ソース名で、*dbname* はデータベースの名前です。

## 複数ユーザーの作成

CREATE USER コマンドを使用すると、Oracle Lite に複数のユーザーを作成できます。ユーザーはスキーマではありません。ユーザーを作成すると、Oracle Lite はユーザーと同じ名前のスキーマを作成し、デフォルト・スキーマとしてそのユーザーに自動的に割り当てます。デフォルト・スキーマ内のデータベース・オブジェクトは、接頭辞としてスキーマ名を付けなくてもアクセスできます。

適切な権限を持つユーザーは、CREATE SCHEMA コマンドを使用して追加スキーマを作成できますが、データベースに接続できるのはデフォルト・スキーマのみです。これらのスキーマは、スキーマを作成したユーザーにより所有され、そのオブジェクトにアクセスするにはスキーマ名に接頭辞が必要になります。

CREATEDB ユーティリティまたは CREATE DATABASE コマンドを使用してデータベースを作成すると、Oracle Lite が SYSTEM という特別のユーザーを作成します。このユーザーはすべてのデータベース権限を持ち、パスワードは割り当てられません。必要であれば SYSTEM にパスワードを割り当てることもできます。個人のユーザー名を設定するまでは、SYSTEM をデフォルト・ユーザー名として使用できます。

Oracle Lite では、SYSTEM 以外のユーザーは、ユーザー自身のスキーマ以外のデータにアクセスしたり操作を実行することはできません。別のユーザーのスキーマ内のデータにアクセスしたり操作を実行するには、特定の権限が必要です。

## 事前定義のロール

Oracle Lite では、便宜上、いくつかの権限が事前定義のロールにまとめられています。多くの場合、ユーザーに対して事前定義のロールを付与する方が、別スキーマの特定権限を付与するより簡単です。Oracle Lite では、ロールの作成や削除はサポートされていません。Oracle Lite の事前定義ロールは次のとおりです。

ロール名	ロールに付与される権限
ADMIN	ユーザーは、他のユーザーを作成し、次に示す DDL および ADMIN 以外の権限をスキーマ内の任意のオブジェクトに対して付与できます。  CREATE SCHEMA、CREATE USER、ALTER USER、DROP USER、DROP SCHEMA、GRANT、REVOKE
DDL	すべての ADMIN 権限と次の権限を含みます。  CREATE TABLE、CREATE VIEW、CREATE INDEX、CREATE CONSTRAINTS、ALTER TABLE、ALTER VIEW、ALTER INDEX、ALTER CONSTRAINT、DROP TABLE、DROP VIEW、DROP INDEX、DROP CONSTRAINT

---

**一般的な注意：** Oracle Server とは異なり、Oracle Lite では COMMIT コマンドを明示的に発行しない限り、データ定義言語（DDL）コマンドはコミットされません。

---

## ユーザーの作成

「SYSTEM」としてデータベースに接続している場合、または ADMIN あるいは DDL ロールを付与されている場合は、ユーザーを作成できます。ユーザーを作成するには、次の文を発行します。

```
CREATE USER user IDENTIFIED BY password
```

ここで、*user* は、文字で始まる最大 128 文字の一意のユーザー名で、*password* は最大 128 文字の名前です。この文により、指定したユーザー名を持つスキーマが作成され、このスキーマがこのユーザーのデフォルト・スキーマとして割り当てられます。

## ユーザーの削除

「SYSTEM」としてデータベースに接続している場合、または ADMIN あるいは DDL ロールを付与されている場合は、ユーザーを削除できます。

ユーザーとそのユーザーが所有するすべてのスキーマを削除するには、次の構文を使用します。

```
DROP USER user
```

ユーザーを削除する前にそのユーザーのスキーマにあるすべてのオブジェクトを削除するには、次の構文を使用します。

```
DROP USER user CASCADE
```

DROP USER コマンドの詳細は、『Oracle Lite SQL リファレンス』を参照してください。

## パスワードの変更

次の条件のいずれかを満たす場合、ユーザーのパスワードを変更できます。

- そのユーザーとしてデータベースに接続している。
- SYSTEM としてデータベースに接続している。
- ADMIN または DDL ロールが付与されている。

ユーザーのパスワードを変更するには、次の文を発行します。

```
ALTER USER user IDENTIFIED BY password
```

## ロールの付与

次の文を発行すると、ユーザーに ADMIN または DDL ロールを付与できます。

```
GRANT role TO user_list
```

ここで、*user\_list* は 1 人のユーザーか、カンマで区切った複数のユーザーのリストです。

## 権限の付与

次の文を発行すると、データベース・オブジェクトに対する権限をユーザーに付与できます。

```
GRANT privilege_list ON object_name TO user_list
```

ここで、*privilege\_list* は、次に示す権限をカンマで区切ったリストか、ALL というそれらをすべて組み合わせたものです。

- INSERT
- DELETE
- UPDATE (col\_list)
- SELECT

*Object\_name* は、スキーマ名が接頭辞として付けられた表名です。

*privilege\_list* が ALL の場合、ユーザーは表またはビューに対して INSERT、DELETE、UPDATE または SELECT を実行できます。*privilege\_list* が INSERT、DELETE、UPDATE または SELECT のいずれかの場合は、ユーザーは表に対してその権限があります。

## ロールの取消し

次の文を発行すると、ユーザー・ロールを取り消すことができます。

```
REVOKE role FROM user_list
```

## 権限の取消し

次の文を発行すると、データベース・オブジェクトに対する権限をユーザーから取り消すことができます。

```
REVOKE privilege_list ON table_name FROM user_list
```

---

---

**注意：** どのアプリケーションでも、ODBC インタフェースを使用すると、CREATE USER、DROP USER または ALTER USER 文を発行できます。

---

---

## デモ表の作成

Oracle Lite には **POLDEMO.SQL** というスクリプトが付属しており、これを使用すると、Oracle Lite の初期データベース (**POLITE.ODB**) 内の表と同じ表を作成できます。

## データベースへの値の移入

この項では、SQL\*Plus、Oracle8 Navigator および SQL\*Loader を使用してデータベースに値を移入する方法について説明します。

### SQL\*Plus によるデータベースへの値の移入

SQL スクリプトを使用すると、表およびスキーマを作成し、表にデータを挿入できます。SQL スクリプトは通常は、**.SQL** 拡張子の付いたテキスト・ファイルで、SQL コマンドが含まれています。SQL スクリプトは、SQL\*Plus のプロンプトで次を入力すると実行できます。

```
SQL> @filename
```

*filename* は、パスおよびファイル名です。

また、次のように入力することもできます。

```
SQL> START filename
```

---

---

**注意：** スクリプトを実行するときに、**.SQL** ファイル拡張子を指定する必要はありません。

---

---

### Oracle8 Navigator による新規データベースへの値の移入

Oracle サーバーから Oracle8 Navigator を使用してデータベースに値を移入するには、まず、サーバーに接続する必要があります。

1. Oracle8 Navigator の左のペインで「データベース接続」を選択し右ボタンでクリックします。
2. 「新規作成」を選択します。  
「新規データベース接続プロパティ」ダイアログが表示されます。
3. 「一般」タブに、新規接続の名前を入力します。次に、ユーザー名、パスワードおよび接続情報を入力します。

---

---

**注意：**「接続」フィールドで、接続先のデータベースが識別されます。別名、接続文字列、サービス名または接続記述子のいずれかの形態で表記されます。このフィールドへの入力項目がわからない場合は、システム管理者に問い合わせてください。

---

---

4. 「構成」ボタンをクリックします。  
「Net8 Assistant - 警告」ダイアログが表示されます。
5. 「はい」ボタンをクリックします。  
SQL\*Net Easy Config ウィザードが表示されます。

---

---

**注意：** リモート・データベース用にネットワーク別名がすでに定義されている PC の場合、ステップ 6 からステップ 13 までを省略し、ステップ 14 に進みます。

---

---

6. 「新規サービスの追加」を選択し、サービス名を入力します。（「一般」タブに入力した名前と同じ名前を使用します。）「次へ」ボタンをクリックします。
7. TCP/IP など、使用中のプロトコルを選択し、「次へ」ボタンをクリックします。
8. TCP/IP ホスト名を入力し、「次へ」ボタンをクリックします。このフィールドへの入力項目がわからない場合は、システム管理者にお問い合わせください。
9. データベースのシステム識別子（SID）を入力し、「次へ」ボタンをクリックします。
10. 「サービスのテスト」をクリックします。
11. 手順 3 のユーザー名とパスワードを入力し、「テスト」ボタンをクリックします。
12. 「接続テストに成功しました」というメッセージが表示されたら、「済み」ボタンをクリックします。
13. 「次へ」ボタン、「完了」ボタンの順にクリックして、Oracle Net8 Easy Config を終了します。
14. 「OK」をクリックし、「新規データベース接続プロパティ」ダイアログを終了します。  
「データベース接続」フォルダの下に新規データベースが表示されます。
15. 新規データベースをダブルクリックしてオープンします。

### コピー・アンド・ペースト（ドラッグ・アンド・ドロップ）

データベースに移入する1つの方法は、データベースにオブジェクトをコピー・アンド・ペーストする方法です。（この方法は、Windows NT エクスプローラを使用したオブジェクトのコピーとよく似ています。）次の手順に従います。

1. 「データベース接続」フォルダを拡張します。
2. コピー元のデータベース接続を拡張します。
3. コピーするオブジェクトを選択（ダブルクリック）します。
4. 右のペインで、オブジェクト（または [Ctrl] キーを使用して複数のオブジェクト）を選択します。
5. 右ボタンでクリックし、「コピー」を選択します。
6. データベースに移動し、オブジェクト型を選択します。
7. 右ボタンでクリックし、「ペースト」を選択します。

オブジェクトを1つのみ選択した場合は、「名前を付けて表をコピー」ウィンドウが表示されます。

---

---

**注意：** 複数のオブジェクトをコピー・アンド・ペーストする場合は、「名前を付けて表をコピー」ウィンドウのかわりに「表をコピー」ウィンドウが表示されます。

---

---

8. 名前を入力し、次のいずれかを選択します。
  - 「構造のみ」  
表定義だけをコピーします。
  - 「構造とデータ」  
表定義と表データをコピーします。

### エクスポートとインポート

Oracle8 Navigator の「エクスポート」および「インポート」の各メニュー・オプションを使用し、次の手順に従ってデータベースに値を移入できます。

ファイルをエクスポートするには、次の手順を実行します。

1. 作業対象のオブジェクトをマウスの右ボタンでクリックします。
2. 「ファイルにエクスポート」を選択します。このファイルは、タブ区切りまたはカンマ区切りのファイルである必要があります。  
「ファイルにエクスポート」ダイアログが表示されます。

3. ファイル名を入力し、区切り文字の種類を選択します。

4. 「保存」ボタンをクリックします。

これでオブジェクトがエクスポートされます。オブジェクトは、編集したりディスクに保存したりできます。

ファイルをインポートするには、次の手順を実行します。

1. ファイルからデータを移入する表を右ボタンをクリックします。

2. 「ファイルからインポート」メニュー・オプションを選択します。

「ファイルからインポート」ダイアログが表示されます。

3. ファイル名を入力し、区切り文字の種類を選択します。

4. 「開く」ボタンをクリックします。

データがインポートされます。

## OLLOAD ユーティリティによるデータベースへの値の移入

データベースには、OLLOAD ユーティリティを使用して値を移入することもできます。このコマンドライン・ツールを使用すると、Oracle Lite データベースにある表に外部ファイルからデータをロードしたり、Oracle Lite データベースの表から外部ファイルにデータをアンロード（ダンプ）できます。SQL\*Loader と異なり、OLLOAD では制御ファイルを使用しません。データ・パラメータと書式情報は、すべてコマンドラインに指定します。詳細は、[付録 C「データベースのツールとユーティリティ」](#)、[「OLLOAD」](#)を参照してください。

## SQL\*Loader によるデータベースへの値の移入

SQL\*Loader ユーティリティ（SQLLDR80.EXE）は、一般的に ASCII テキスト・ファイルなどの外部ファイルから、Oracle データベースの表にデータを移動します。SQL\*Loader は、タイプの異なる複数のデータ・ファイルからデータをロードできます。SQL\*Loader は DOS コマンドラインから起動できます。詳細は、SQL\*Loader のマニュアルを参照してください。

## データベースのバックアップ

Oracle Lite データベースはファイルを 1 つしか占有しないため、ファイルを別の場所にコピーするとバックアップできます。データベースのバックアップには、Oracle8 Navigator も使用できます。詳細は、Oracle8 Navigator のオンライン・ヘルプを参照してください。

---

---

**注意：** ユーザーを作成してある場合、または SYSTEM ユーザーにパスワードを割り当ててある場合は、.ODB ファイルだけでなく .OPW ファイルもコピーする必要があります。

---

---



## データベースの暗号化と復号化

[ENCRYPDB](#) と [DECRYPDB](#) という 2 つのツールにより、Oracle Lite データベースを暗号化および復号化できます。ツールの機能により、Oracle Lite データベースをパスワードで暗号化できます。パスワードを使用すると、データベースへの不正アクセスを防止したり、データベース・ファイルを調べることによって、その中のデータが解読されないようにデータベースを暗号化できます。パスワードを使用することにより、40 ビットの暗号キーを導出できます。Oracle Lite では、CAST5 と呼ばれるデータ暗号化規格 (DES) アルゴリズムを使用します。

この 2 つのユーティリティの詳細は、[付録 C 「データベースのツールとユーティリティ」](#) の「[ENCRYPDB](#)」および「[DECRYPDB](#)」を参照してください。



---

## アプリケーションの配布

Oracle Lite データベース用に作成されたアプリケーションには、特定のプログラム・ファイルが必要です。これらのファイルは、アプリケーションの機能に応じて異なります。この章では、データベース、およびデータベース・アプリケーションの関連プログラム・ファイルを配布する開発者用に提供されているツールを説明します。説明する内容は次のとおりです。

- [Deployment Wizard](#)
- [Deployment Utility](#)
- 手動による配布

# Deployment Wizard

Deployment Wizard を使用すると、必要な Oracle Lite データベース・ファイルを選択してアプリケーションとともに配布できます。Deployment Wizard は、指定された情報を基にファイルを選択し、コンピュータ上の宛先ディレクトリにそのファイルをコピーします。アプリケーションを別のコンピュータにインストールする場合は、コピーされたファイルをアプリケーションとともに配布する必要があります。ファイルは手動で配布するか（「[手動による配布](#)」を参照）、または Deployment Utility を使用して配布します（「[Deployment Utility](#)」を参照）。

Deployment Wizard は、Oracle8i Lite の CD-ROM にありますが、インストール・プロセスではインストールされません。Deployment Wizard を起動するには、次の手順に従います。

- 1. Oracle8i Lite の CD-ROM 上の Redist¥Deploy ディレクトリにナビゲートします。
- 2. **DEPLOY.EXE** をダブルクリックします。

次に示すリストは、Deployment Wizard が、コンピュータ上の宛先ディレクトリにどのファイルをコピーするかを判断するためのオプションです。

オプション	機能
Destination Directory	宛先ディレクトリを指定できます。再配布に必要なファイルが、すべてこのディレクトリに入れられます。
Database Access	アプリケーションが Oracle Lite データベースにアクセスするために必要な接続モード（ODBC、JDBC または JAC）を選択できます。
JDBC Options	アプリケーションが Oracle Lite データベースのアクセスに JDBC を使用する場合に、JDBC 接続オプションを指定できます。
Java Stored Procedures	アプリケーションが Java ストアド・プロシージャを使用するかどうかを決定します。
Replication	アプリケーションが Oracle Lite データベースとリモート Oracle Server 間でデータをレプリケートするかどうかを決定します。
Replication Options	アプリケーションがレプリケーションの実行に使用できる転送方法を選択できます。このオプションは、アプリケーションがレプリケーションを実行する場合にのみ表示されます。
Replication Access	アプリケーションが OLE コントロールと Java Replication Class (JRC) のいずれを使用するかを決定します。このオプションは、アプリケーションがレプリケーションを実行する場合にのみ表示されます。
Verify Selections	選択したオプションが表示されます。選択をレビューし、必要な場合に変更できます。

オプション	機能
POLITE.INI 設定	POLITE.INI ファイルの変更オプションを提供します。
Database File	デフォルトのデータベース・ファイル <b>POLITE.ODB</b> を提供します。別のデータベース・ファイルを選択することもできます。
Deployment Utility	Deployment Utility を使用するというオプションを提供します。このユーティリティは、クライアント・コンピュータ上にファイルを自動的にインストールします。詳細は、「 <a href="#">Deployment Utility</a> 」を参照してください。このオプションを選択しない場合、ファイルは手動で配布する必要があります。詳細は、「 <a href="#">手動による配布</a> 」を参照してください。
ODBC Data Source	<b>ODBC.INI</b> ファイルの変更オプションを提供します。

## ヘルプ・ファイル

Deployment Wizard は、以上のオプションのそれぞれに対してヘルプ・ファイルを提供します。ヘルプ・ファイルにアクセスするには、任意のウィザード・パネルで「Help」ボタンをクリックします。

Deployment Wizard は、コンピュータの宛先ディレクトリに **DEPLOYMENT.TXT** というファイルもインストールします。このファイルには次の情報が含まれています。

- Deployment Wizard で選択したオプションに従って提供される Oracle Lite データベース・ファイル。
- **POLITE.INI** ファイルまたは **ODBC.INI** ファイルで変更が必要なパラメータ。

## Deployment Utility

Deployment Utility を使用すると、アプリケーションに必要なファイルを自動的に配布できます。Deployment Utility は、必要な Oracle Lite データベース・ファイルを、コンピュータの宛先ディレクトリからアプリケーションを含むコンピュータの適切なディレクトリにコピーします。

Deployment Utility は、次に示す機能を実行します。

- **POLITE.INI** ファイルを変更する。
- CLASSPATH と PATH を設定する。
- 必要な ODBC エントリを **ODBC.INI** ファイルとレジストリに設定する。

Deployment Utility を起動するには、次のように入力します。

```
polutil.exe <destination_directory>
```

**注意：** Deployment Utility は、Deployment Wizard でこのユーティリティを選択した場合にのみ使用できます。

パッチを受け取った場合は、宛先ディレクトリにコピーしたファイルを、受け取ったパッチ・ファイルで置き換える必要があります。

## 手動による配布

アプリケーションに必要な Oracle Lite データベース・ファイルは、手動で配布することもできます。これには、宛先コンピュータの適切なディレクトリにこれらのファイルをコピーし、POLITE.INI ファイルまたは ODBC.INI ファイルの必要なパラメータを変更します。

## 手動による配布をする際に使用するパラメータ

手動による配布をする際に使用するパラメータを次にリストします。

パラメータ	説明
<i>executable_path</i>	アプリケーションの実行可能ファイルと DLL をインストールするディレクトリのドライブ指定子を含むフル・パスです。
PATH	現在の環境での検索パスです。パスには <i>executable_path</i> を含める必要があります。
CLASSPATH	Java アプリケーションは、CLASSPATH にリストされているファイル内のクラスを検索します。これは環境変数です。
<i>class_directory</i>	Java クラスと JAR ファイル用のディレクトリです。このディレクトリは、CLASSPATH 検索リスト内に含まれている必要があります。

Deployment Wizard は、Oracle8i Lite の CD-ROM 上の DLL を使用します。アプリケーションがその後のパッチ・リリースの DLL に依存している場合は、必ず次を実行する必要があります。

- CD-ROM から ¥REDIST ディレクトリをハード・ディスクにコピーする。
- ¥REDIST の該当するサブディレクトリにある DLL を、受け取ったパッチで更新する。
- ハード・ディスクから Deployment Wizard を実行する。

## コア DLL と EXE の配布

Oracle Lite データベースのランタイム・エンジンはいくつかの DLL で構成されていますが、この DLL をターゲット・システムのバスにある実行可能ディレクトリにコピーする必要があります。これらの DLL は、Oracle Lite データベースを配布するときはいつも配布する必要があります。

- **OLOD2040.DLL** (ODBC ドライバ)
- **OLSQL40.DLL** (SQL ランタイム)
- **OLOBJ40.DLL** (オブジェクト・カーネルと API)

さらに、**POLITE.INI** を Windows ディレクトリにインストールする必要があります。メッセージ・ファイル **OLITE40.MSB** (または使用する言語の該当ファイル) もバス内にインストールする必要があります。

失敗したプロセスが保持していたリソースを解放しロールバックする場合は、最初に接続する前に、**OLITRM40.EXE** をインストールして起動する必要があります。この機能が不要な場合は、このファイルをインストールする必要がありません。

## ODBC ドライバのセットアップ

次に示す説明では、Microsoft ODBC に関する知識が前提です。ODBC ファイルの再配布方法、ODBC ドライバのインストール方法、さらにデータ・ソース名 (DSN) のセットアップ方法を理解する必要があります。詳細は、Microsoft ODBC のマニュアルを参照してください。

ODBC ドライバをインストールするには、次の DLL ファイルを *executable\_path* にコピーします。

- **OLAD2040.DLL** (新規 ODBC DSN のセットアップの場合)

ODBC ドライバをセットアップするには、次の項目をレジストリに登録します。

- 32bit=1
- APILevel=0
- ConnectFunctions=YYN
- Driver=<executable\_path>%OLOD2040.DLL
- DriverODBCVer=02.00
- Setup=<executable\_path>%OLAD2040.DLL
- SQLLevel=0

データ・ソース名 (DSN) のセットアップ時に、**ODBC.INI** ファイル内の該当する DSN に次の情報を追加します。

```
[Your DSN]
DataDirectory=<C:¥MYAPP¥MYDIR>    (ユーザーのデータベース・ディレクトリ)
Database=<MYDB>    (データベース・ファイル名, 拡張子なし)
```

## JDBC ドライバの配布

JDBC ドライバは、**OLJDBC40.DLL** ファイルに含まれています。このファイルを実行可能ディレクトリにインストールし、以下の JAR ファイルを *class\_directory* にインストールします。

- **OLJDBC40.DLL** (JDBC ランタイム)
- **OLITE40.JAR** (JDBC、JAC および JSP に対するサポートを含む)
- **OLJDK11.JAR** (JDK 1.1 を使用している場合は CLASSPATH に追加、JDK 1.2 では不要)

**ODBC.INI** ファイルを使用せずに Oracle Lite データベースに接続するには、次の接続文字列を使用します。

```
"jdbc:Polite:olite;DataDirectory=<DatabaseDirectory>;Database=<databasename>"
```

## Java クラスの配布

Java サポートは、次の DLL ファイルと JAR ファイルにより提供されています。これらの DLL ファイルは、実行可能ディレクトリにコピーする必要がある、JAR ファイルは *class\_directory* に必要です。

- **OLJAC40.DLL** (JAC ランタイム)
- **OLJDBC40.DLL** (JDBC ランタイム)
- **OLITE40.JAR** (JDBC、JAC および Java ストアド・プロシージャをサポート)
- **OLJDK11.JAR** (CLASSPATH に追加、JDK 1.2 では不要)



## レプリケーション API の配布

Oracle Lite データベース・レプリケーションには、いくつかの DLL を *executable\_path* にコピーする必要があります。どのファイルが必要になるかは、使用する転送方法により異なります。

Oracle Mobile Agents レプリケーションの場合：

- **POLOMA8.DLL** (Oracle Mobile Agents 転送)
- **POLREP8.DLL** (Oracle Net8 転送)
- **OLREP840.DLL** (レプリケーション・コールバック)

ファイル・ベースのレプリケーション (ファイル転送) の場合：

- **POLFILE8.DLL** (ファイル・ベースのレプリケーション)
- **POLREP8.DLL** (Oracle Net8 転送)
- **OLREP840.DLL** (レプリケーション・コールバック)

HTTP レプリケーションの場合：

- **POLHTTP8.DLL** (HTTP トランスポート・レイヤー)
- **POLREP8.DLL** (Oracle Net8 転送)
- **OLREP840.DLL** (レプリケーション・コールバック)

転送方法に関係なく、レプリケーション用 OLE コントロール **REPSVR8.OCX** を実行可能ディレクトリにインストールする必要があります。

- **REPSVR8.OCX** (レプリケーション用 OLE コントロール)

このコントロールは、インストール後に登録する必要もあります。登録するには、次のように指定します。

```
REGSVR32.EXE /s REPSVR8.OCX
```

レプリケーションを Java アプリケーションから起動する場合は、実行可能ディレクトリに **JRC840.DLL** ファイルをインストールします。

- **JRC840.DLL** (JRC)

また、次のファイルが CLASSPATH になければなりません。

- **OLITE40.JAR**
- **OLJDK11.JAR** (CLASSPATH に追加、JDK 1.2 では不要)

Oracle Net8 転送を使用するレプリケーションでは、ネットワーク・ソフトウェア、トランスポート・アダプタおよび必須サポートファイルをインストールする必要があります。このマニュアルでは、これらの必須ファイルについては説明しません。

## 追加ファイルの配布（オプション）

SQL\*Plus などの Oracle 製品経由で Oracle Lite データベースに接続する必要がある場合は、Open Client Adapter（OCA）の DLL を実行可能ディレクトリにコピーします。

- **UB80W32.DLL**（OCA）
- **UBUS.MSB**（OCA のエラー・メッセージ・テキスト）

---

---

**注意：** OCA には、次の場所に OCA60 というレジストリ・キーが含まれた Oracle インストール環境が必要になります。

HKEY\_LOCAL\_MACHINE/Software/ORACLE

このキーには、**UBUS.MSB** メッセージ・ファイルの場所を指す文字列値が含まれている必要があります。このキーは、このマニュアルでは説明されていない他の Oracle 製品に依存します。

---

---

---

## サンプル・アプリケーションの使用

この章では、Oracle Lite データベースに提供されているサンプル・アプリケーションについて説明します。説明する内容は次のとおりです。

- 概要
- Access サンプル・アプリケーションに関する注意
- BLOB Manager サンプル・アプリケーションに関する注意
- Delphi サンプル・アプリケーションの実行
- PowerBuilder サンプル・アプリケーションの実行
- Visual Basic サンプル・アプリケーションの実行

概要

Oracle Lite のインストールを完了した後、`Oracle_Home¥Lite¥Examples` ディレクトリにあるサンプルを利用できます。

**注意：** サンプルのほとんどは、データ・ソース名（DSN）として POLITE を使用しています。このデータ・ソース名を削除および再作成する必要がある場合は、[REMOVEDB](#) および [CREATEDB](#) ユーティリティを使用します。

ツール	サンプル・アプリケーションの格納場所	説明
Access	<code>Oracle_Home¥Lite¥Examples¥Access</code>	Oracle Lite データベースのフロントエンドとしての Microsoft Access の使用方法を示します。詳細は、「 <a href="#">Access サンプル・アプリケーションに関する注意</a> 」を参照してください。
AQ Lite	<code>Oracle_Home¥Lite¥AQLITE¥AQLSAMP</code>	モバイル・クライアントとサーバー・アプリケーション間のメッセージ通信に AQ Lite を使用する方法を示します。詳細は、『Oracle Lite AQ Lite 開発者ガイド』を参照してください。
Blob Manager	<code>Oracle_Home¥Lite¥Examples¥Blob Manager</code>	Oracle データ型 BLOB の使用方法および Visual Basic の ODBC プログラミング手法とオブジェクト操作を紹介します。詳細は、「 <a href="#">BLOB Manager サンプル・アプリケーションに関する注意</a> 」を参照してください。
Delphi	<code>Oracle_Home¥Lite¥Examples¥Delphi¥Delphi1</code> <code>Oracle_Home¥Lite¥Examples¥Delphi¥Delphi2</code>	Oracle Lite データベースのフロントエンドとしての Delphi の使用方法を示します。詳細は、「 <a href="#">Delphi サンプル・アプリケーションの実行</a> 」を参照してください。

ツール	サンプル・アプリケーションの格納場所	説明
Java	<code>Oracle_Home\Lite\Examples\Java</code>	Java ストアド・プロシージャ、トリガー、Java アクセス・クラス (JAC)、および JDBC を使用したプログラミングを示します。詳細は、『Oracle Lite Java 開発者ガイド』を参照してください。
ODBC	<code>Oracle_Home\Lite\Examples\Odbc</code>	C で作成された ODBC プログラムを提供します。
PowerBuilder	<code>Oracle_Home\Lite\Examples\Pb</code>	Oracle Lite データベースの表に対して問合せを実行する PowerBuilder アプリケーションの作成方法を紹介します。詳細は、『PowerBuilder サンプル・アプリケーションの実行』を参照してください。
REPSVR	<code>Oracle_Home\Lite\Examples\Repsvr</code>	Oracle Lite データベースで OLE コントロールを使用する Microsoft Visual Basic プログラムを提供し、レプリケーションの例を紹介します。詳細は、『Oracle Lite レプリケーション・ガイド』を参照してください。
Visual Basic	<code>Oracle_Home\Lite\Examples\Vb</code>	Visual Basic ツールを使用して Oracle Lite データベースの表に対する問合せを簡単に実行できる例を紹介します。詳細は、『Visual Basic サンプル・アプリケーションの実行』を参照してください。

## Access サンプル・アプリケーションに関する注意

Microsoft Access のサンプルは、Access で Oracle Lite データベースを操作する方法を示します。このサンプルでは、ODBC DSN の POLITE とデフォルトの Oracle Lite データベースを使用しています。

連結表、Access ベースで SQL 固有の問合せ、さらにサンプルのフォームとレポートを表示できます。

このサンプルでは Microsoft Access 95 を使用しています。

## BLOB Manager サンプル・アプリケーションに関する注意

BLOB Manager のサンプルをインストールするには、`Oracle_Home¥Lite¥Examples¥Blob Manager` にある `¥setup` フォルダをオープンし、**setup.exe** を実行します。インストール完了後、「スタート」ボタンをクリックし、「プログラム」メニューから「BLOB Manager」を選択します。

このサンプルをアンインストールするには、「スタート」ボタンをクリックし、「設定」、次いで「コントロール パネル」を選択します。「アプリケーションの追加と削除」を選択します。「BLOB Manager」を選択し、「追加と削除」ボタンをクリックします。

---

---

**注意：** BLOB Manager はデモ用です。デフォルトの ODBC DSN の POLITE を使用してデフォルト・データベースがインストールされていることが前提です。そうでない場合は、ODBC Administrator を使用して POLITE DSN を作成できます。また、SYSTEM がデータベースにとって有効なユーザーであることを確認する必要があります。

---

---

## Delphi サンプル・アプリケーションの実行

Delphi のサンプル・アプリケーションは2つあります。

- 最初のサンプルは、ODBC 経由で Oracle Lite データベースにアクセスする Borland Delphi アプリケーションの開発方法を示します。このサンプルを実行するには、Oracle Lite のインストールにより作成されているデフォルトの ODBC DSN およびデータベースの POLITE が必要です。また、表 SYSTEM.EMP も必要です。
- 2つ目のサンプルは、ODBC 経由で Oracle Lite データベースにアクセスする Borland Delphi アプリケーションの開発方法と、JOIN SQL 文を使用してデータにアクセスする方法を示します。このサンプルを実行するには、Oracle Lite のインストールにより作成されているデフォルトの ODBC DSN およびデータベースの POLITE が必要です。また、デフォルトの表 SYSTEM.EMP と表 SYSTEM.DEPT も必要です。

Delphi サンプル・アプリケーションのインストールと実行に関する次の説明では、Oracle Lite データベースと Delphi（バージョン 2.0 以降）がインストール済であることが前提です。

### ODBC ドライバの別名の作成

1. Borland Delphi プログラム・グループで、「Borland Delphi」フォルダを開きます。「BDE Configuration」アイコンをダブルクリックします。
2. BDE Configuration ユーティリティの「ドライバ」タブで、「新規 ODBC ドライバ」をクリックし、「ODBC ドライバの追加」ダイアログを表示します。このダイアログに次を入力します。
  - SQL Link Driver: xxx（「xxx」を入力します）

- Default ODBC Driver: (リストから ODBC ドライバを選択します)
  - Default Data Source Name: POLITE
3. 「OK」をクリックします。次に「別名」タブをクリックし、「新規別名」をクリックして「新規別名の追加」ダイアログを表示します。このダイアログに次を入力します。
    - Add New Alias Name: poldb
    - Alias Type: ODBC\_xxx
  4. 「OK」をクリックします。次に「ファイル」メニューから「終了」を選択します。変更の保存を求めるプロンプトが表示されたら、「はい」をクリックします。

### Delphi の開始

Borland Delphi プログラム・グループの「Delphi」アイコンをダブルクリックし、Delphi を開始します。

### サンプル・アプリケーションの開始

1. 「ファイル」メニューの「オープン」を選択します。表示されるダイアログで、`Oracle_Home¥Lite¥Examples¥Delphi¥Delphi1` (または `Delphi2`) ディレクトリに移動し、`Empform.dpr` を選択して、「OK」をクリックします。
2. 「Database Logon」ダイアログで、フィールドに次のように入力します。
  - User Name: SYSTEM
  - Password: (少なくとも 1 文字を入力します。)
3. 「OK」をクリックします。EMP 表を使用するサンプル・アプリケーションが開発モードで表示されます。

### 開発モードでの表とデータの表示

「データベース」メニューの「検索」を選択して、Explorer ユーティリティを開始します。Database Explorer を使用して、Oracle Lite の表とデータを表示します。

### ランタイム・モードでのデータの表示と操作

1. ツールバーの「実行」をクリックして、「Database Logon」ダイアログを表示します。ログオンするには、フィールドに次のように入力します。
  - User Name: SYSTEM
  - Password: (パスワードを入力します)
2. サンプル・アプリケーションの EMP 表がランタイム・モードで表示されるので、アプリケーション・ボタンを使用してデータを表示し、操作できます。データを表示するには、ナビゲーション・ボタンを使用して、最初、次、前および最後のレコードへ移動し

ます。データを操作するには、「編集」、「転記」および「リフレッシュ」の各ボタンを使用します。編集内容を取り消すには、「キャンセル」ボタンを使用します。

## PowerBuilder サンプル・アプリケーションの実行

このサンプルは、PowerBuilder の DataWindow と Oracle Lite データベースがどのように動作するかを示します。DataWindow は d\_polite という名前です。DataWindow はデフォルト・データベースの EMP 表と ODBC DSN の POLITE を使用します。このサンプルでは PowerBuilder 5.0 を使用します。

PowerBuilder サンプル・アプリケーションのインストールと実行に関する以下の説明では、Oracle Lite と PowerBuilder（バージョン 5.0 以降）がインストール済であることが前提です。PowerBuilder ODBC ドライバをインストールしていない場合は、以下の手順を開始する前にドライバをインストールしてください。

### PowerBuilder の開始

PowerBuilder プログラム・グループ内の「PowerBuilder」アイコンをダブルクリックして、PowerBuilder を開始します。

### プロファイルの作成

1. PowerBar の「DBProf」アイコンをクリックして「Database Profiles」ダイアログを表示し、「新規作成」をクリックします。フィールドに次のとおりに入力します。
  - Profile Name: polite
  - DBMS: ODBC
  - UserID: SYSTEM
  - Password: (少なくとも 1 文字を入力します。)
  - Database Name: POLITE
2. 新規プロファイルを受け入れるには、「More」をクリックします。「DBPARM」ドロップダウン・リスト・ボックスで、デフォルトの「ConnectionString」を選択します。次のように編集して、データ・ソース名 (DSN) のパラメータを指定します。
  - Set DSN = POLITE
  - UID = SYSTEM
  - PWD = (変更するかデフォルトをそのまま残します)
3. パラメータを受け入れるには、「OK」をクリックします。次に「OK」をもう一度クリックしてプロファイルを完了し、ODBC 初期化ファイルにデータ・ソースを入力します。



### 接続の確認

1. 作成したばかりの DSN に接続できることを確認するには、PowerBar の「Database」アイコンをクリックします。

---

**注意：** PowerBuilder が Oracle Lite データベースに接続すると、「カタログ表は作成できなかったため、使用できません」というメッセージが表示される場合があります。PowerBuilder がデータベースに初めて接続するときにはカタログ表が作成されるため、このメッセージは無視します。ただし、このプロセスは数秒かかることがあります。

---

Database Painter で、次のことを確認してください。

- 新規プロファイルが Database Painter のタイトル・バーに表示される。
  - デフォルトのデータベース表が「Select Tables」ダイアログに表示される。
2. 「Select Tables」ダイアログの「Show System Tables」チェックボックスをクリックし、次の 5 つの PowerBuilder システム表が Oracle Lite データベースにインストールされていることを確認します。
    - SYSTEM.PBCAT COL
    - SYSTEM.PBCAT EDT
    - SYSTEM.PBCAT FMT
    - SYSTEM.PBCAT TBL
    - SYSTEM.PBCAT VLD

### サンプル・アプリケーションの開始

1. PowerBar の「Application」アイコンをクリックして、Application Painter を表示します。最後に作成したアプリケーションが Application Painter によって自動的に開始されます。アプリケーションを作成していない場合は、新規アプリケーションの作成または既存アプリケーションの開始を求めるプロンプトが表示されます。
2. サンプル・アプリケーションを開始するには、「オープン」アイコンをクリックします。「Select Application Library」ダイアログで、`Oracle_Home¥Lite¥Examples¥PB` ディレクトリに移動し、**Polite.pbl** を選択して、「OK」をクリックします。

### EMP 表データの表示

PowerBar の「DataWindow」アイコンをクリックして、「Select DataWindow」ダイアログを表示します。「d\_polite」を選択し、「OK」をクリックします。DataWindow Painter で「Preview」アイコンをクリックして、EMP 表のデータを表示します。

### データベース内の別の表に基づく DataWindow の作成

1. PowerBar の「DataWindow」アイコンをクリックして「Select DataWindow」ダイアログを表示し、「新規作成」をクリックします。
2. 「New DataWindow」ダイアログの「データ・ソース」ペインで、「SQLSelect」をクリックします。「Presentation Style」ペインで、「表」をクリックします。次に「OK」をクリックします。
3. 「Select Tables」ダイアログで表を 1 つ以上選択して新しい DataWindow 内に表示し、「オープン」をクリックします。

### 表のオープンとプレビューする列の選択

表のヘッダーを右ボタンでクリックし、「すべて選択」をクリックします。「選択」アイコンをクリックして、DataWindow の定義を表示します。「Preview」をクリックして、DataWindow を表示します。

## Visual Basic サンプル・アプリケーションの実行

(Visual Basic 5.0 を使用する) このサンプルは、Oracle Lite データベースを使用する Visual Basic アプリケーションの開発方法を示したものです。ODBC DSN の POLITE が使用されます。AddNew、Update および Delete マクロを使用するには、EMP 表に一意の EMPNO 列が必要です。これが、デフォルト・データベースに接続するときのデフォルトの状態です。

Visual Basic サンプル・アプリケーションのインストールと実行に関する以下の説明では、Oracle Lite と Visual Basic (バージョン 4.0 以降) がインストール済であることが前提です。

---

---

**注意：** Visual Basic ODBC ドライバをインストールしていない場合は、以下の手順を開始する前にドライバをインストールしてください。

---

---

### Visual Basic の開始

Visual Basic プログラム・グループにある「Visual Basic」アイコンをダブルクリックして、Visual Basic を開始します。

### サンプル・アプリケーションの表とデータの表示

このステップでは、Visual Data Manager を使用します。Visual Data Manager は、Visual Basic 5.0 でのみ使用できます。バージョン 5.0 より前の Visual Basic を使用している場合は、ステップ 3 に進みます。

1. 「アドイン」メニューの「Visual Data Manager」を選択します。「VisData」ウィンドウで、「ファイル」メニューの「データベースを開く」を選択し、「ODBC」を選択します。

2. 「Database Logon」ダイアログで、フィールドに次のように入力します。
  - DSN: POLITE
  - UID: SYSTEM
  - PW: (少なくとも 1 文字を入力します。)
  - Database: POLITE
3. 「OK」をクリックします。「データベース」ウィンドウに Oracle Lite データベースの表が表示されます。表をハイライトして、右ボタンでクリックすると、表が開いて、レコードが表示されます。

### サンプル・アプリケーションの開始

1. サンプル・アプリケーションを開始するには、「ファイル」メニューの「Open Project」を選択します。ダイアログ・ボックスで、*Oracle\_Home¥Lite¥Examples¥VB* ディレクトリに移動します。**MAKEFILE** を選択し、「オープン」をクリックします。

---

---

**注意：** ファイル **MAKEFILE** が表示されない場合は、ファイル・タイプに \*.\* を選択して、すべてのファイル・タイプを表示します。リストにファイルが表示されます。

---

---

2. 「実行」メニューの「開始」を選択してサンプル・アプリケーションを開始し、EMP 表を表示します。

### EMP 表内のデータの表示と操作

1. EMP 表データの表示
  - 「表示」をクリックすると、EMP 表のデータが表示されます。
  - 「次」をクリックすると、次のレコードが表示されます。
  - 「前」をクリックすると、前のレコードが表示されます。
2. EMP 表のデータを操作するには、「追加」、「更新」および「削除」の各機能を使用します。



---

## Oracle Lite レプリケーションの使用

この章では、Oracle Lite のデータ・レプリケーション機能の使用方法について説明します。説明する内容は次のとおりです。

- [レプリケーションの概要](#)
- [作業の準備](#)
- [ステップ 1: レプリケーションのサーバー・サポートの確認](#)
- [ステップ 2: レプリケーション・エンジン・サポートの設定](#)
- [ステップ 3: 高速リフレッシュ用スナップショット・ログの作成](#)
- [ステップ 4: マスター表のレプリケーション準備](#)
- [ステップ 5: レプリケーション接続の設定](#)
- [ステップ 6: スナップショットの作成](#)
- [レプリケーションでサポートされているデータ型](#)

## レプリケーションの概要

Oracle Lite では、プライマリ・リモート・データベースのデータのローカル・コピー、つまり「複製」を作成することにより、レプリケーションの実行が可能になります。情報を瞬間的に捕えるため、データのローカル・コピーは「スナップショット」と呼ばれます。一度、スナップショットが作成されると、データの最新状況を反映するために、定期的にリフレッシュできます。

とても簡単に聞こえますが、レプリケーションは、マスター・サイトとスナップショット・サイト間での正確な調整に依存します。マスター・サイトとして機能する Oracle サーバーにはバージョン 7.3 またはバージョン 8.0.5 を使用する必要があり、マスター・サイトはレプリケーション・サービス処理するデータベース管理者が構成する必要があります。マスター・サイトが正しく構成されていないと、レプリケーションはできません。

Oracle サーバーを「マスター・サイト」と呼び、マスター・サイトにある表を「マスター表」と呼びます。ローカル・データベースまたは Oracle Lite クライアントは、「スナップショット・サイト」と呼びます。

Oracle Lite データベースは、スナップショット・サイトとしてのみ機能します。スナップショットの 2 つの種類、「読取り専用」と「更新可能」がサポートされます。読取り専用スナップショットは、問合せにのみ使用できます。スナップショットには、マスター表への変更のみがレプリケート（プル）されます。スナップショットに変更を加えることはできません。更新可能スナップショットはマスター表の更新可能なコピーで、マスター表全体のコピーがその内容となるように定義することも、マスター表の行のうち、値による選択条件を満たす行のサブセットがその内容となるように定義することもできます。スナップショットは変更可能で、変更は元のマスター表へ伝播（プッシュ）できます。

---

---

**注意：** レプリケーションの詳細は、『Oracle Lite レプリケーション・ガイド』に記載されています。

---

---

## 作業の準備

レプリケーションの例を紹介するにあたって、レプリケーション機能付きの Oracle Lite をローカル・マシンにインストールする必要があります。インストール手順は、『Oracle8i Lite インストレーション・ガイド』を参照してください。

マスター・サイトで作業する場合は、管理者権限またはデータベース管理者の補助が必要です。

## ステップ 1: レプリケーションのサーバー・サポートの確認

Oracle Lite を使用するレプリケーション・アプリケーションでは、アドバンスト・レプリケーションをサポートするマスター・サイトが必要です。マスター・サイトについて、データベース管理者と次の点を確認します。

- Oracle8 Enterprise Edition のバージョン 8.0.5（以降）または Oracle7 Server のバージョン 7.3 である。
- アドバンスト・レプリケーション・サポートがアクティブで、正しく構成されている。

## ステップ 2: レプリケーション・エンジン・サポートの設定

マスター・サイトが Oracle7 データベースの場合、Oracle Lite レプリケーション・エンジン REPAPI のサポートに必要な所定の機能をマスター・サイトにインストールする SQL スクリプトを実行する必要があります。このスクリプトは **REP73.SQL** と呼ばれ、*Oracle\_Home¥dbs* サブディレクトリにあります。このスクリプトをマスター・サイトで実行するには、データベース管理者の補助または管理者権限が必要です。

## ステップ 3: 高速リフレッシュ用スナップショット・ログの作成

高速リフレッシュ（推奨）を使用する場合、レプリケートする各マスター表のマスター・サイトで、スナップショット・ログを作成する必要があります。ログは、SQL\*Plus のコマンドラインで SQL を実行するか、または Oracle Replication Manager を使用して作成できます。

---

**注意：** Oracle Replication Manager は、Oracle Enterprise Manager のコンポーネントであり、Oracle サーバーに付属しています。Oracle Lite には含まれていません。

---

たとえば、マスター表 EMP のスナップショット・ログを作成するには、マスター・サイトで次の SQL コマンドを発行します。

```
CREATE SNAPSHOT LOG ON EMP;
```

これにより、MLOG\$\_EMP という名前のスナップショット・ログ表がマスター・サイトに作成されます。マスター・データベースが Oracle8 の場合は、更新されたすべての行の主キーがスナップショット・ログにデフォルトで記録されるようになります。

更新された行の ROWID を記録するマスター表 EMP のスナップショット・ログも作成できます。

```
CREATE SNAPSHOT LOG ON EMP WITH ROWID;
```

また、ROWID と主キーを記録するマスター表のスナップショット・ログも作成できます。

```
CREATE SNAPSHOT LOG ON EMP WITH ROWID, PRIMARY KEY;
```

## ステップ 4: マスター表のレプリケーション準備

レプリケーション用にマスター表を準備するには、レプリケートの対象となるすべてのマスター表を含むマスター・グループを作成する必要があります。Oracle8 の主キー・ベースのスナップショットの場合、マスター表には主キーが含まれ、副問合せのサブセットに参加する表は、副問合せのサブセット規則に従う必要があります。詳細は、Oracle8 サーバー・レプリケーションのマニュアルを参照してください。

たとえば、Replication Manager を使用して次の手順を実行します。

1. 「データベース接続」フォルダ内の接続をダブルクリックして、マスター・サイトのデータベースに接続します。
2. 「構成」フォルダを拡張し、「マスター・グループ」を右ボタンでクリックします。次に、「新規マスター・グループ作成」を選択します。
3. 「新規マスター・グループ作成」プロパティ・シートに、マスター・グループ名として RepTest を入力します。「オブジェクト」タブを選択し、レプリケートするマスター表を追加します。「追加」をクリックし、ドロップダウン・リストから、たとえば「Scott」を選択します。EMP 表（スナップショット・サイトでのスナップショットの作成対象）を選択します。

## ステップ 5: レプリケーション接続の設定

Oracle8 Navigator と Replication OLE コントロールは、どちらも Oracle Net8 を使用してマスター・サイトに接続します。Oracle Net8 Assistant と Net8 Easy Config は、Net8 の構成ツールです。

### マスター・サイトへの接続の設定

Oracle Net8 Easy Config を使用してマスター・サイトにサービス名を設定するには、次の手順を実行します。

1. Oracle for Windows 95、Oracle for Windows 98 または Oracle for Windows NT の各プログラム・グループの「Net8 Easy Config」アイコンを選択します。
2. 「Net8 Assistant - 警告」ダイアログで、「はい」をクリックします。  
Net8 構成ウィザードが表示されます。
3. 「Oracle Net8 Easy Config」画面で、デフォルト・アクション（「新規サービスの追加」）を選択し、「新規サービス名」フィールドに新規サービス名を入力して、「次へ」をクリックします。



4. サービス名のあるマスター・サイトに接続するためのネットワーク・プロトコルを選択し、「次へ」をクリックします。
5. マスター・サイトと通信できるコンピュータの名前を入力し、「次へ」をクリックします。
6. デフォルトのデータベース・インスタンス名「ORCL」を受け入れるか、接続先のデータベース・インスタンス名を入力します（データベース・インスタンス名がわからない場合は、データベース管理者に問い合わせてください）。「次へ」をクリックします。
7. 新規サービス名をテストするには、「サービスのテスト」ボタンをクリックして「接続テスト」ダイアログを表示します。次に、データベースにログオンするためのユーザー名とパスワードを入力します。「テスト」をクリックして、新規サービス名を検証します。確認メッセージにより、正常に接続されたかが示されます。テストが完了したら、「済み」をクリックします。
8. 最後に、「次へ」をクリックして、ウィザードの最終ページに進みます。「完了」をクリックして、サービス名を保存します。

## Oracle8 Navigator でのマスター・サイト接続の追加

Oracle8 Navigator でマスター・サイト接続を追加するには、次の手順を実行します。

1. Oracle Lite プログラム・グループの「Oracle8 Navigator」アイコンを選択します。
2. 「ファイル」メニューで、「新規作成」、「新規データベース接続」の順に選択します。
3. 「新規データベース接続プロパティ」ダイアログ・ボックスにデータベース別名を入力します。この別名は、「名前」フィールドでマスター・サイト用に作成したものです。マスター・サイトのユーザー名とパスワードを入力します（この情報がわからない場合は、データベース管理者に問い合わせてください）。

「接続」フィールドで、データベース別名の Oracle Net8 接続文字列をドロップダウン・リスト・ボックスから選択し、「OK」をクリックします。マスター・サイト接続が、「データベース接続」フォルダに追加されます。次回、Oracle8 Navigator を開始するときは、その接続をダブルクリックしてマスター・サイトに接続できます。

## ステップ 6: スナップショットの作成

スナップショットは、Oracle8 Navigator を使用する場合は手動で作成でき、Replication OLE コントロール・インタフェースまたは Java Replication Class インタフェースを使用する場合はプログラムによって作成できます。Replication OLE コントロールの詳細は、『Oracle Lite レプリケーション・ガイド』を参照してください。Java レプリケーション・クラスの詳細は、Oracle8i Lite マニュアル・ページからアクセスできる『Java レプリケーション・クラス API リファレンス』を参照してください。

Oracle8 Navigator でスナップショットを作成するには、次の手順を実行します。

1. 「Oracle Lite データベース」フォルダおよび「デフォルト Oracle Lite データベース」フォルダを拡張します。また、作成したばかりのマスター・サイト接続の下にある「表」を拡張します。
2. マスター・サイトの EMP 表をドラッグし、「デフォルト Oracle Lite データベース」フォルダ内の「スナップショット」フォルダにドロップします。
3. 「スナップショットの作成」ダイアログで、スナップショットを「更新可能」または「読取り専用」のどちらにするかを指定できます。「更新可能」を選択し、「OK」をクリックします。
4. 「新規スナップショット」プロパティ・シートで、「一般」タブを選択します。「名前」フィールドに「EMP\_S」と入力します。「所有者」フィールドで、ドロップダウン・リスト・ボックスから所有者を選択します。オプション操作として、「条件」タブを選択し、スナップショットのベースとなる問合せをさらに細かく定義できます。
5. 作業が終わったら、「OK」をクリックします。
6. 作成したスナップショットは、「デフォルト Oracle Lite データベース」フォルダ内の「スナップショット」フォルダをハイライトすることによって表示できます。「スナップショット」フォルダ内のスナップショットはすべて、Oracle8 Navigator の右のペインに表示されます。検査するスナップショットを右ボタンでクリックし、「オープン」を選択します。

---

---

**注意：** Oracle Lite で初めてスナップショットを作成するとき、Oracle Lite がスナップショット・データを格納するベース表が作成されます。ベース表には、マスター表と同じ列定義とデータ定義があります。マスター・サイトが Oracle8 データベースの場合、複製される制約は主キー制約と NOT NULL 制約です。マスター・サイトが Oracle7 データベースの場合は、複製される制約は NOT NULL 制約のみです。スナップショットにさらに多くのデータ整合性制約が必要な場合は、スナップショットを作成した後、ベース表に制約を追加できます。制約を追加するには、SQL コマンド ALTER TABLE を使用します。詳細は、『Oracle Lite SQL リファレンス』を参照してください。

---

---

## スナップショットの検証

スナップショットが正しく定義されていることを検証するには、次の手順を実行します。

1. 「デフォルト Oracle Lite データベース」フォルダ内の「スナップショット」フォルダを右ボタンでクリックします。

2. 「検索」メニュー・オプションを選択します。  
「スナップショット」フォルダ内のスナップショットはすべて、Oracle8 Navigator の右のペインに表示されます。
3. 検査するスナップショットを右ボタンでクリックします。
4. 「プロパティ」メニュー・オプションを選択します。  
「プロパティ」ダイアログで、「一般」タブを選択して名前、タイプ、所有者、さらにスナップショットに関するコメントを表示します。「プロパティ」ダイアログの「定義」タブは、スナップショットを定義する SQL 問合せを表示します。

## レプリケーションでサポートされているデータ型

レプリケーションでは次のデータ型がサポートされます。

- CHAR
- NUMBER
- DATE
- VARCHAR2
- RAW
- ROWID



---

# POLITE.INI のデータベース・パラメータ

POLITE.INI ファイルで定義されているデータベース・パラメータの値を変更して、Oracle Lite データベースをカスタマイズできます。この章では、POLITE.INI ファイルと、関連するパラメータについて説明します。説明する内容は次のとおりです。

- [POLITE.INI ファイルの概要](#)
- [POLITE.INI のパラメータ](#)
- [POLITE.INI ファイルのサンプル](#)

## POLITE.INI ファイルの概要

**POLITE.INI** ファイルにより、データベース・ボリューム ID の割当てが一元化され、システム上の全データベースのパラメータが定義されます。**Oracle Lite** をインストールすると、**POLITE.INI** ファイルが Windows NT または Windows 95/98 のホーム・ディレクトリに作成されます。

**POLITE.INI** ファイルのパラメータは、インストール時に自動設定されますが、それらを変更して、製品動作をカスタマイズできます。**POLITE.INI** ファイルの変更には、ASCII テキスト・エディタを使用します。

## POLITE.INI のパラメータ

次の項以降で、**POLITE.INI** ファイルの *All Databases* セクションに指定するパラメータを示します。

### CacheSize

オブジェクト・キャッシュのサイズを KB 単位で指定します。最小値は 128 です。指定がない場合は、デフォルトの 4096 (4MB) が使用されます。

### DatabaseID

**Oracle8 Navigator** プログラムまたは CREATE DATABASE SQL コマンドにより次に割り当てられるデータベース・ボリューム ID 番号を定義します。システム上の各データベース・ファイルに対するデータベース・ボリューム ID 番号は、一意のものである必要があります。

### DbCharEncoding

**Oracle Lite** により実行される UTF 変換を指定します。**NATIVE** に設定すると、UTF 変換は実行されません。**UTF8** に設定すると、UTF 変換が実行されます。このパラメータを指定しない場合、デフォルトは **UTF8** です。

### MAXINDEXCOLUMNS

索引作成文で使用される列数を定義します。詳細は、『**Oracle Lite SQL** リファレンス』の「索引作成オプション」を参照してください。

### NLS\_Date\_Format

**Oracle Lite** のデフォルト以外の日付書式を使用できるようになります。日付値が想定されている場所にリテラル文字列が指定されると、**Oracle Lite** データベースはその文字列をテストして、文字列が **Oracle**、**SQL92**、または **POLITE.INI** ファイルでこのパラメータ用に指定されている値のいずれかに一致するかどうかを調べます。このパラメータを設定すると、他の

書式文字列が指定されていない場合、TO\_CHAR 関数または TO\_DATE 関数で使用するデフォルト書式も定義されることになります。

Oracle ではデフォルトは dd-mon-yy または dd-mon-yyyy で、SQL-92 ではデフォルトは yy-mm-dd または yyyy-mm-dd です。

書式に「RR」を使用すると、49 以下の 2 桁の年表記は 21 世紀（2000 ～ 2049）と解釈され、50 以上の年表記は 20 世紀（1950 ～ 1999）と解釈されます。すべての 2 桁年エントリのデフォルトとして RR 書式を設定すると、西暦 2000 年対応になります。例を次に示します。

```
NLS_DATE_FORMAT='RR-MM-DD'
```

日付書式は、ALTER SESSION コマンドを使用して変更することもできます。詳細は、『Oracle Lite SQL リファレンス』を参照してください。

日付書式

日付書式には、次の表にリストされている要素が 1 つ以上含まれています。同じような情報を表す要素を組み合わせることはできません。たとえば、「SYYYYY」と「BC」を同一書式文字列で使用することはできません。

書式	説明
AM または A.M.	正午標識。ピリオドはオプションです。
PM または P.M.	正午標識。ピリオドはオプションです。
CC または SCC	世紀。「S」によって紀元前の日付の先頭に「-」が付けられます。
D	曜日（1 ～ 7）。
Day	曜日。9 文字になるまでブランクが埋め込まれます。
DD	月間通算日（1 ～ 31）。
DDD	年間通算日（1 ～ 366）。
DY	曜日の省略形。
IW	ISO 標準に基づく年間通算週（1 ～ 52 または 1 ～ 53）。
IYY、IY または I	ISO 年表記の最後の 3 桁、2 桁、または 1 桁。
IYYY	ISO 標準に基づく 4 桁の年表記。
HH または HH12	時刻（1 ～ 12）。
HH24	時刻（0 ～ 23）。
MI	分（0 ～ 59）。

書式	説明
MM	月（01 ～ 12。たとえば JAN=01）。
MONTH	月の名前。9 文字になるまでブランクが埋め込まれます。
MON	月の名前の省略形。
Q	四半期（1、2、3、4。たとえば、JAN ～ MAR=1）。
RR	年号の最後の 2 桁。外国での年号。
WW	年間通算週（1 ～ 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。
SS	秒（0 ～ 59）。
SSSSS	午前 0 時以降の秒数（0 ～ 86399）。
Y,YYY	指定した位置にカンマの付いた年表記。
YEAR または SYEAR	綴りで表した年表記。「S」によって紀元前の日付の先頭に「-」が付けられます。
YYYY または SYYYY	4 桁の年表記。「S」によって紀元前の日付の先頭に「-」が付けられます。
YYY、YY または Y	年表記の最後の 3 桁、2 桁、または 1 桁。



### 日付書式の例

次のリストは、NLS\_DATE\_FORMAT パラメータの例です。

NLS\_DATE\_FORMAT= *format*

書式	
YYYY-MONTH-DAY:HH24:MI:P.M.	MONTH/DD, YYYY, HH:MI A.M.
YYYY/MONTH/DD, HH24:MI A.M.	MONTH   DD, YYYY, HH:MI A.M.
YYYY-MONTH-DAY:HH24:MI:P.M.	MONTH DD, YYYY, HH:SSSS:MI A.M.
MM D, YYY, HH:MI A.M.	MONTH DD, HH:SS:MI CC
MM, WW, RR, HH:MI A.M.	MONTH DD, HH:SS:MI SCC
MM, IW, RR, HH:MI A.M.	MONTH W, YYYY, HH:MI A.M.
MM, DY, RR, HH:MI A.M.	MONTH WW, YYYY, HH:MI A.M.
MM; DY; IYY, HH:MI A.M.	MONTH WW, RR, HH:MI A.M.
MON WW, RR, HH:MI A.M.	MONTH WW, Q, HH:MI A.M.
MONTH.DD, SYYY, HH:MI A.M.	MONTH WW, RR, HH:MI A.M.

### NLS\_Locale

Oracle Lite の言語依存の動作を指定するには、**POLITE.INI** ファイルで NLS\_Locale パラメータを定義します。NLS\_Locale の値は、次の書式です。

*language\_territory.codepage*

*territory* と *codepage* はオプションです。指定しないと、デフォルト値が使用されます。

たとえば、次のように指定します。

NLS\_LOCALE=JAPANESE\_JAPAN

この例の場合、言語が日本語で、地域が日本となります。

月や日の名前、その省略形や数値による書式は、言語に基づいてカスタマイズされます。サポートされている言語は、次のとおりです。

サポートされている言語		
AMERICAN	FINNISH	NORWEGIAN
BRAZILIAN PORTUGUESE	FRENCH	POLISH
BULGARIAN	GERMAN	PORTUGUESE
CANADIAN FRENCH	GREEK	ROMANIAN
CATALAN	HEBREW	RUSSIAN
CROATIAN	HUNGARIAN	SLOVAK
CZECH	ICELANDIC	SLOVENIAN
DANISH	ITALIAN	SPANISH
DUTCH	LATIN AMERICAN SPANISH	SWEDISH
EGYPTIAN	LITHUANIAN	THAI
ENGLISH	MALAY	TURKISH
ESTONIAN	MEXICAN SPANISH	UKRAINIAN

ReverseJoinOrder

問合せで表を結合する順序を指定します。オプションは TRUE または FALSE です。TRUE の場合は、FROM 句における順序とは反対の順序で表が結合されます。FALSE の場合は、FROM 句における順序と同じ順序で表が結合されます。指定されていない場合は、Oracle Lite の問合せ最適化により、最適な結合順序が決定されます。

SharedAddress

Oracle Lite では、アプリケーションにまたがって必要なデータは、共有メモリーを使用して管理します。Oracle Lite は、プロセス・メモリーの特定の位置に共有メモリーを連結します。ごくまれに、この位置が他のツールによりすでに使用中のことがあり、その場合はエラーが発生します。この問題に対応するために、Oracle Lite では次の規則をサポートして、共有メモリー用に連結するメモリー・アドレスを判断します。

共有メモリーを連結する前に、Oracle Lite は SharedAddress 変数と 16 進の 32 ビット・アドレス（たとえば、18000000）を指定する [SuggestedSharedAddress](#) 変数を調べます。Oracle Lite は、見つかった最初の値を使用します。いずれの変数も設定されていない場合、Oracle Lite はアドレス 30000000 を最初に試します。この値は、ほとんどのアプリケーションで使用されている範囲より上位にあります。

Oracle Lite クライアントがすでに実行中の場合、2 番目のプロセスが同じ共有メモリー・アドレスを取得できないと、このクライアントはエラーで失敗します。ただし、2 番目のプロセスで使用可能なアドレスを [SuggestedSharedAddress](#) として **POLITE.INI** ファイルに書き

込みます。ここでユーザーが Oracle Lite クライアントをすべて終了し、同じアプリケーション・セットを実行した場合、問題は再発しません。

自動競合解決に失敗した場合は、問題が解決されるまでの間、SharedAddress 変数を変更する必要があります。たとえば、値の間隔を 256MB に設定し、20000000、24000000、28000000 などで試してみます。

## SQLCompatibility

Oracle Lite は、Oracle SQL と SQL-92 の両方の機能をサポートします。Oracle SQL と SQL-92 の詳細は、『Oracle Lite SQL リファレンス』を参照してください。

Oracle SQL と SQL-92 の間に矛盾がある場合、SQLCompatibility フラグが参照されます。このパラメータに Oracle を指定すると Oracle SQL が優先され、SQL92 を指定すると SQL-92 が優先されます。このパラメータを **POLITE.INI** に含めないと、デフォルトで Oracle SQL が優先されます。

## SuggestedSharedAddress

説明は、「[SharedAddress](#)」の項を参照してください。

## SVTRACE

トレースがブランチ・オフィス・データベース上の ODBC API 用ログ・ファイルにメンテナンスされるかどうかを指定します。

トレースは、system32 サブディレクトリにある **olsv.log** ファイル、または **Olite Service** 起動用の起動パラメータの作業ディレクトリ内に獲得されます。

トレースを行う場合は ON、行わない場合は OFF (デフォルト) に設定できます。

## TempDB

一時データベースは、デフォルトではメイン・メモリー内に作成されます。これにより、一時表の使用が必要な問合せのパフォーマンスが向上します。一時データベースを意図的にファイル・システム内に作成する場合を除いて、**poltempx.odt** ファイルは作成されません。セーブポイント情報の格納に使用されることのある **\*.slx** ファイルも作成されません。大規模な結果セットを作成する場合は、結果を保持する十分なスワップ領域を用意するか、または一時データベースにファイル・オプションを選択する必要があります。

このオプションを含めるには、**POLITE.INI** ファイルに次の構文を使用します。

```
TempDB=<path temporary_database_name>
```

例を次に示します。

```
TempDB=c:¥temp¥olite_
```

この例のように設定すると、Oracle Lite では次の一時データベースが作成されます。

```
c:¥temp¥olite_0.odb, c:¥temp¥olite_1.odb, ...
```

## TempDir

一時データベース **POLTEMP.ODB** が作成されるディレクトリを指定します。設定されていない場合は、各環境で定義されている TEMP、TMP または WINDIR の設定がデフォルトとして使用されます。

## POLITE.INI ファイルのサンプル

次に、**POLITE.INI** ファイルのサンプルを示します。

```
[All Databases]
DatabaseID=128
DBCharEncoding=NATIVE
SuggestedSharedAddress=10270000
CacheSize=4096
MAXINDEXCOLUMNS=2
SQLCompatibility=SQL92
NLS_Date_Format=RR/MM/DD H24,MI,SS
NLS_Locale=ENGLISH
TempDB=c:¥temp¥olite_
TempDir=D:¥TMP
ReverseJoinOrder=TRUE
```

---

## システム・カタログ

この付録では、システム・カタログの次のビューを説明します。

- |                    |                     |
|--------------------|---------------------|
| ■ ALL_COL_COMMENTS | ■ ALL_TAB_COMMENTS  |
| ■ ALL_CONSTRAINTS  | ■ ALL_USERS         |
| ■ ALL_CONS_COLUMNS | ■ ALL_VIEWS         |
| ■ ALL_INDEXES      | ■ CAT               |
| ■ ALL_IND_COLUMNS  | ■ COLUMN_PRIVILEGES |
| ■ ALL_OBJECTS      | ■ DUAL              |
| ■ ALL_SEQUENCES    | ■ SNAPSHOTS         |
| ■ ALL_SYNONYMS     | ■ TABLE_PRIVILEGES  |
| ■ ALL_TABLES       | ■ USER_OBJECTS      |
| ■ ALL_TAB_COLUMNS  |                     |

---

**注意：** 次の表で、アスタリスクの付いた列は Oracle Lite では使用されません。このような列は Oracle8 との互換性を保つために使用され、通常、NULL またはデフォルト値を返します。

---

# ALL\_COL\_COMMENTS

このビューは、表の列に関するユーザーのコメントをリストします。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	表の所有者。
TABLE_NAME	VARCHAR2(128)	NOT NULL	オブジェクト名。
COLUMN_NAME	VARCHAR2(128)	NOT NULL	列の名前。
COMMENTS	VARCHAR2(4096)		列のコメントのテキスト。

# ALL\_CONSTRAINTS

このビューは、アクセス可能な表の制約定義について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	制約定義の所有者。
CONSTRAINT_NAME	VARCHAR2(128)	NOT NULL	制約定義に関連付けられている名前。
CONSTRAINT_TYPE	VARCHAR2(128)	NOT NULL	制約定義のタイプ： C（表のチェック制約） P（主キー） U（一意キー） R（参照整合性） V（チェック・オプション付き、 ビューに対して）
TABLE_NAME	VARCHAR2(128)	NOT NULL	制約定義のある表の名前。
SEARCH_CONDITION	VARCHAR2(1000)		表チェック用の検索条件のテキスト。
R_OWNER	VARCHAR2(128)		参照制約で使用した表の所有者。
R_CONSTRAINT_NAME	VARCHAR2(128)		参照された表の一意性制約定義の名前。
DELETE_RULE	VARCHAR2(128)		参照制約の削除規則：「NO ACTION」。

列	データ型	NULL	説明
STATUS	VARCHAR2(20)	NOT NULL	制約の状態:「ENABLED」または「DISABLED」。
VALIDATED	VARCHAR2(13)		制約の状態:「VALIDATED」または「NOT VALIDATED」。

## ALL\_CONS\_COLUMNS

このビューは、アクセス可能な列の制約定義について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)		制約定義の所有者のユーザー名。
CONSTRAINT_NAME	VARCHAR2(128)		制約定義に関連付けられている名前。
TABLE_NAME	VARCHAR2(128)		制約定義のある表の名前。
COLUMN_NAME	VARCHAR2(128)		制約定義で指定された列に関連付けられている名前。
POSITION	NUMBER(10)		定義における列の元の位置。

## ALL\_INDEXES

このビューは、表に定義されているすべての索引の説明を含んでいます。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	INDEX 定義の所有者。
INDEX_NAME	VARCHAR2(128)	NOT NULL	INDEX 定義に関連付けられている名前。
TABLE_OWNER	VARCHAR2(128)	NOT NULL	INDEX が定義されている表の所有者。
TABLE_NAME	VARCHAR2(128)	NOT NULL	INDEX 定義のある表の名前。
TABLE_TYPE	VARCHAR2(10)		オブジェクト型。
UNIQUENESS	VARCHAR2(128)	NOT NULL	「UNIQUE」または「NONUNIQUE」を含む文字列。

# ALL\_IND\_COLUMNS

このビューは、データベース内の全索引の索引キー列をリストします。

列	データ型	NULL	説明
INDEX_OWNER	VARCHAR2(128)	NOT NULL	INDEX 定義の所有者。
INDEX_NAME	VARCHAR2(128)	NOT NULL	INDEX 定義に関連付けられている名前。
TABLE_OWNER	VARCHAR2(128)	NOT NULL	INDEX が定義されている表の所有者。
TABLE_NAME	VARCHAR2(128)	NOT NULL	INDEX 定義のある表の名前。
COLUMN_NAME	VARCHAR2(128)	NOT NULL	INDEX 定義で指定された列に関連付けられている名前。
COLUMN_POSITION	NUMBER(10)	NOT NULL	INDEX 定義内での列の位置。

# ALL\_OBJECTS

このビューには、オブジェクト（表、ビュー、シノニム、索引および順序）の説明が含まれています。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	OBJECTS 定義の所有者。
OBJECT_NAME	VARCHAR2(128)	NOT NULL	OBJECTS 定義に関連付けられている名前。
OBJECT_TYPE	VARCHAR2(128)		オブジェクト型: TABLE、VIEW、INDEX、SEQUENCE、SYNONYM。
CREATED	DATE		OBJECTS 作成のためのタイムスタンプ。
STATUS	VARCHAR2(128)		OBJECTS の状態: VALID、INVALID または N/A（常に有効）。



## ALL\_SEQUENCES

このビューは、データベース内のすべての順序の説明をリストします。

列	データ型	NULL	説明
SEQUENCE_OWNER	VARCHAR2(128)	NOT NULL	SEQUENCES 定義の所有者。
SEQUENCE_NAME	VARCHAR2(128)	NOT NULL	SEQUENCES 定義に関連付けられている名前。
MIN_VALUE	NUMBER(10)	NOT NULL	順序の最小値。
MAX_VALUE	NUMBER(10)	NOT NULL	順序の最大値。
INCREMENT_BY	NUMBER(10)	NOT NULL	順序の増分。

## ALL\_SYNONYMS

このビューは、データベース内のすべてのシノニムをリストします。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)		SYNONYMS 定義の所有者。
SYNONYM_NAME	VARCHAR2(128)		SYNONYMS 定義に関連付けられている名前。
TABLE_OWNER	VARCHAR2(128)		SYNONYM が定義される表の所有者。
TABLE_NAME	VARCHAR2(128)		SYNONYM 定義のある表の名前。
DB_LINK	VARCHAR2(128)		予約済。

## ALL\_TABLES

このビューは、ユーザーがアクセス可能な表について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	表の所有者のユーザー名。
TABLE_NAME	VARCHAR2(128)	NOT NULL	表の名前。

列	データ型	NULL	説明
TABLESPACE_NAME	VARCHAR2(128)		表のあるデータベース・ファイルカタログの名前。
CLUSTER_NAME*	VARCHAR2(128)		表が属するクラスタがある場合、そのクラスタの名前。
PCT_FREE*	NUMBER(10)		ブロックの空き領域の最小パーセンテージ。
PCT_USED*	NUMBER(10)		ブロックの使用領域の最小パーセンテージ。
INI_TRANS*	NUMBER(10)		最初のトランザクション数。
MAX_TRANS*	NUMBER(10)		トランザクションの最大数。
INITIAL_EXTENT*	NUMBER(10)		最初のエクステントのバイト・サイズ。
NEXT_EXTENT*	NUMBER(10)		次のエクステントのバイト・サイズ。
MIN_EXTENTS*	NUMBER(10)		セグメントで許されるエクステントの最小限の数。
MAX_EXTENTS*	NUMBER(10)		セグメントで許されるエクステントの最大数。
PCT_INCREASE*	NUMBER(10)		エクステント・サイズの増加率。
BACKED_UP*	VARCHAR2(1)		最後の変更以後にバックアップされた表。
NUM_ROWS*	NUMBER(10)		表の行数。
BLOCKS*	NUMBER(10)		表に割り当てられたデータ・ブロックの数。
EMPTY_BLOCKS*	NUMBER(10)		データを含まない表に割り当てられたデータ・ブロックの数。
AVG_SPACE*	NUMBER(10)		表に割り当てられたデータ・ブロックの空き領域の平均量（バイト単位）。
CHAIN_CNT*	NUMBER(10)		1つのデータ・ブロックから他のデータ・ブロックへと連鎖されているか、古い ROWID を保つためのリンクを要求して新規ブロックに移った表の行数。
AVG_ROW_LEN*	NUMBER(10)		表の行の平均的な長さ（バイト単位）。

## ALL\_TAB\_COLUMNS

このビューは、ユーザーがアクセス可能な表、ビューとクラスタの列について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	表、ビューまたはクラスタの所有者のユーザー名。
TABLE_NAME	VARCHAR2(128)	NOT NULL	表、ビューまたはクラスタの名前。
COLUMN_NAME	VARCHAR2(128)	NOT NULL	列の名前。
DATA_TYPE	VARCHAR2(30)		列のデータ型。
DATA_LENGTH	NUMBER(10)		バイト単位による列の長さ。
DATA_PRECISION	NUMBER(10)		NUMERIC と DECIMAL データ型の 小数点以下の精度、FLOAT、REAL および DOUBLE データ型のバイナリ 精度、その他のデータ型の場合は NULL。
DATA_SCALE	NUMBER(10)		NUMERIC または DECIMAL での小 数点以下の桁。
NULLABLE	VARCHAR2(1)		列に NULL を指定できるかどうか。 列に NOT NULL 制約がある場合か、 列が主キーの一部になる場合は、値 が N になります。
COLUMN_ID	NUMBER(10)	NOT NULL	作成された列の順序番号。
DEFAULT_LENGTH	NUMBER(10)		列のデフォルト値の長さ。
DATA_DEFAULT	VARCHAR2(4096)		列のデフォルト値。
NUM_DISTINCT*	NUMBER(10)		表の各列の固有値の数。
LOW_VALUE*	NUMBER(10)		HIGH_VALUE の説明を参照。
HIGH_VALUE*	NUMBER(10)		4 行以上ある表で、列の 2 番目の最低 値と最高値。3 行以下の表で、最低値 と最高値。これらの統計表示は、値の 最初の 32 バイトの内部表現で、16 進 の表記法で表されます。

## ALL\_TAB\_COMMENTS

このビューは、表およびビューに対してユーザーが入力したコメントをリストします。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	TAB_COMMENTS 定義の所有者。
TABLE_NAME	VARCHAR2(128)	NOT NULL	TAB_COMMENTS 定義のある表の名前。
TABLE_TYPE	VARCHAR2(128)	NOT NULL	オブジェクト型。
COMMENTS	VARCHAR2(4096)	NOT NULL	コメント・テキスト。

## ALL\_USERS

このビューは、接続されたデータベースで作成されたすべてのスキーマについて、次の情報を提供します。

列	データ型	NULL	説明
USERNAME	VARCHAR2(30)	NOT NULL	ユーザーの名前。
USER_ID	NUMBER	NOT NULL	ユーザーの ID 番号。
CREATED	DATE	NOT NULL	ユーザー作成日付。

## ALL\_VIEWS

このビューは、ユーザーがアクセス可能なビューについて、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	ビューの所有者のユーザー名。
VIEW_NAME	VARCHAR2(128)	NOT NULL	ビューの名前。
TEXT_LENGTH	NUMBER(10)	NOT NULL	ビュー・テキストの長さ。
TEXT	VARCHAR2(1000)	NOT NULL	ビュー・テキスト。

## CAT

このビューは、ユーザーがアクセス可能な表とビューについて、次の情報を提供します。

列	データ型	NULL	説明
TABLE_NAME	VARCHAR2(128)	NOT NULL	オブジェクト名。
TABLE_TYPE	VARCHAR2(128)	NOT NULL	オブジェクト型: TABLE または VIEW。

## COLUMN\_PRIVILEGES

このビューは、ユーザーが付与者、受領者または所有者である列の権限付与、あるいは PUBLIC が受領者である列の権限付与について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	オブジェクトの所有者のユーザー名。
TABLE_NAME	VARCHAR2(128)	NOT NULL	オブジェクト名。
COLUMN_NAME	VARCHAR2(128)		列の名前。
GRANTOR	VARCHAR2(128)		権限付与を実行したユーザー名。
GRANTEE	VARCHAR2(128)		アクセス権限が付与されたユーザー名。
GRANT_TYPE	VARCHAR2(128)	NOT NULL	オブジェクトの権限。値は、SELECT、INSERT または DELETE のいずれかです。
GRANTABLE	VARCHAR2(128)		権限が GRANT OPTION で付与された場合は YES、それ以外は NO。

## DUAL

このビューは、単一行を返すときに、問合せで使用できるダミー表です。たとえば、CURRENT\_TIMESTAMP を選択するために DUAL を使用できます。

列	データ型	NULL	説明
DUMMY	VARCHAR2(1)	NOT NULL	常に「X」。

# SNAPSHOTS

このビューはデフォルトでは存在しません。レプリケーション使用時にのみ作成されます。  
ユーザーがアクセス可能なスナップショットについて、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(30)		スナップショットの所有者。
NAME	VARCHAR2(30)		ユーザーとアプリケーションがスナップショットの表示に使用するビューの名前。
TABLE_NAME	VARCHAR2(30)		スナップショットが保存される表。
MASTER_VIEW	VARCHAR2(30)		スナップショット・マスター・ビューの名前。
MASTER_OWNER	VARCHAR2(30)		マスター表の所有者。
MASTER	VARCHAR2(30)		このスナップショットがコピーであるマスター表の名前。
MASTER_LINK	VARCHAR2(128)		マスター・サイトへのデータベース・リンク名。
MASTER_ROLLBACK	VARCHAR2(30)		マスター・サイトで使用するロールバック・セグメント。
CAN_USE_LOG	VARCHAR2(3)		スナップショットでスナップショット・ログを使用できるかどうかの指定。このスナップショットがスナップショット・ログを使用できる場合は YES、複雑すぎてログを使用できない場合は NO です。
UPDATABLE	VARCHAR2(3)		スナップショットが更新可能かどうかの指定。可能な場合は YES、不可能な場合は NO です。NO の場合、スナップショットは読取り専用になります。
SUBQUERY	VARCHAR2(3)		スナップショット問合せが副問合せを持つかどうかの指定。副問合せがある場合は YES、ない場合は NO です。
KEYTYPE	VARCHAR2(4)		Oracle7 の場合、スナップショットが ROWID スナップショットか主キー・スナップショットかの指定。指定できる値 : ROWID は R、主キーは P。

列	データ型	NULL	説明
SNAPSHOT_ID	DATE		Oracle7 の場合、スナップショットの固有の ID。
SNAPSHOT_ID8	NUMBER		Oracle8 の場合、スナップショットの固有の ID。
SNAPTYPE	NUMBER		Oracle8 の場合、スナップショットが ROWID スナップショットか主キー・スナップショットかの指定。指定できる値 : ROWID は R、主キーは P。
REFMETHOD	NUMBER		リフレッシュのタイプ。指定できる値 : COMPLETE、FAST、OPTIMUM (FORCE)。この列は Oracle 内部でのみ使用されます。
LAST_REFRESH	DATE		最後のリフレッシュのマスター・サイトでの日付と時間。
TYPE	VARCHAR2(8)		リフレッシュのタイプ。指定できる値は、複合は C、シンプルは S です。
NEXT	VARCHAR2(200)		次のリフレッシュ日を計算するために使用する日付機能。
START_WITH	DATE		最初のリフレッシュ日を計算するために使用する日付機能。
REFRESH_GROUP	NUMBER		リフレッシュ・グループの ID。
UPDATE_TRIG	VARCHAR2(30)		UPDATE_LOG を満たすトリガー名。
UPDATE_LOG	VARCHAR2(30)		更新可能なスナップショットの変更を記録する表。
STATUS	VARCHAR2(8)		ランタイム・リフレッシュ状態。
PKCOLS	VARCHAR2(1056)		主キー列を保存。
TABLE_COUNT	NUMBER		副問合せスナップショットの表カウント。
SCHEMA_CHANGED	CHAR(1)		マスター・スキーマが変更されたかどうか。変更した場合は YES、変更しない場合は NO です。
HIDDEN_COLUMNS	VARCHAR2(1056)		非表示列を格納します。
QUERY	LONG		スナップショットを定義する SQL 問合せ。

## TABLE\_PRIVILEGES

このビューは、ユーザーか PUBLIC が受領者であるオブジェクトの権限付与について、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	オブジェクトの所有者のユーザー名。
TABLE_NAME	VARCHAR2(128)	NOT NULL	オブジェクト名。
GRANTOR	VARCHAR2(128)		権限付与を実行したユーザー名。
GRANTEE	VARCHAR2(128)		アクセス権限が付与されるユーザー名。
GRANT_TYPE	VARCHAR2(128)	NOT NULL	オブジェクトの権限。指定する値は次のいずれかです。SELECT、INSERT または DELETE。
GRANTABLE	VARCHAR2(128)		権限が GRANT OPTION で付与された場合は YES、そうでない場合は NO です。

## USER\_OBJECTS

このビューは、ユーザーがアクセス可能なオブジェクトについて、次の情報を提供します。

列	データ型	NULL	説明
OWNER	VARCHAR2(128)	NOT NULL	オブジェクトの所有者のユーザー名。
OBJECT_NAME	VARCHAR2(128)	NOT NULL	オブジェクト名。
OBJECT_ID	NUMBER(10)	NOT NULL	オブジェクトのオブジェクト識別子。
OBJECT_TYPE	VARCHAR2(128)		オブジェクト型:TABLE、VIEW、INDEX、SEQUENCE、SYNONYM。
CREATED	DATE		オブジェクト作成のタイムスタンプ。
LAST_DDL_TIME	DATE		データ定義言語コマンド（付与と取消しを含む）によって生じたオブジェクトの最終変更のタイムスタンプ。



列	データ型	NULL	説明
CREATED_TIME	VARCHAR2(128)		オブジェクト（文字データ）作成のタイムスタンプ。
STATUS*	VARCHAR2(128)		OBJECTS の状態 : VALID、INVALID または N/A（常に有効）。



# データベースのツールとユーティリティ

この付録では、次のデータベース・ユーティリティの使用方法を説明します。ユーティリティの名前はアルファベット順に並んでいます。

ユーティリティ	説明
CREATEDB	Oracle Lite データベースの作成に使用します。
DECRYPTDB	Oracle Lite データベースの復号化に使用します。
dropjava	Oracle Lite データベースから Java クラスを削除するために使用できるコマンドライン・ユーティリティです。詳細は、『Oracle Lite Java 開発者ガイド』を参照してください。
ENCRYPTDB	Oracle Lite データベースの暗号化に使用します。
loadjava	Oracle Lite に Java クラスをロードするために使用できるコマンドライン・ユーティリティです。詳細は、『Oracle Lite Java 開発者ガイド』を参照してください。
MIGRATE	以前のリリースから Oracle Lite に移行するために使用します。
ODBC Administrator と Oracle Lite ODBC ドライバ	ODBC 接続の管理に使用します。Oracle Lite ODBC ドライバ経由でアクセスする Oracle Lite データベースに ODBC ドライバを関連付けるデータ・ソース名（DSN）を作成します。
ODBINFO	このユーティリティは、Oracle Lite データベースのバージョン・ナンバーとボリューム ID を見つけるために使用します。
OLitRM40	トランザクションを監視し、失敗したトランザクションのロールバックとクリーンアップを行います。
OLLOAD	このコマンドライン・ツールは、Oracle Lite データベースにある表に外部ファイルからデータをロードしたり、Oracle Lite データベースの表から外部ファイルにデータをアンロード（ダンプ）するために使用します。

ユーティリティ	説明
Oracle8 Navigator	Oracle Lite データベースを管理する GUI ツールです。このツールは、プロジェクト、データベース接続およびデータベース・オブジェクトのすべてにアクセスするために使用します。
Oracle Lite Designer	データベース・オブジェクトを表示し管理する Java GUI ツールです。
REMOVEDB	Oracle Lite データベースの削除に使用します。

# CREATEDB

## 説明

データベースを作成するためのユーティリティです。

## 構文

CREATEDB DataSourceName DatabaseName [VolID]

## キーワードとパラメータ

### DataSourceName

データ・ソース名で、デフォルトのデータベース・ディレクトリの **ODBC.INI** ファイルを検索するために使用します。

**注意：** 無効な DSN を指定すると、Oracle Lite はその DSN を無視し、カレント・ディレクトリにデータベースを作成します。このデータベースを ODBC 経由でアクセスするには、データベースが存在するディレクトリを指す DSN を作成する必要があります。DSN を追加する方法は、[「ODBC Administrator を使用した DSN の追加」](#)を参照してください。

### DatabaseName

作成するデータベース名です。フルパス名、またはデータベース名のみを指定できます。データベース名のみが指定された場合、**ODBC.INI** ファイルに指定されたデータ・ソース名のデータ・ディレクトリにデータベースが作成されます。データベース名の拡張子は、常に **.ODB** である必要があります。指定した名前に **.ODB** が付いていない場合は、**.ODB** が付加されます。

### VolID

VolID が指定された場合、VolID は **POLITE.INI** ファイルに指定されたデータベース ID のかわりに使用されます。この ID はデータベースごとに一意でなければなりません。

## コメント

CREATEDB で作成されたデータベースは、Oracle8 Navigator には表示されません。Oracle8 Navigator を使用してデータベースを作成することをお勧めします。

## 例

```
createdb polite db1
createdb polite c:\testdir\db2.odb 300
```

# DECRYPDB

## 説明

このツールを使用すると、暗号化された Oracle Lite データベースを復号化できます。詳細は、「[ENCRYPDB](#)」を参照してください。

## 構文

```
DECRYPDB DSN | NONE DBName [Passwd]
```

## キーワードとパラメータ

### DSN

復号化する Oracle Lite データベースのデータ・ソース名です。NONE を指定すると、DBName には（.ODB 拡張子を除く）フルパス名で入力する必要があります。

### DBName

復号化するデータベース名です。DSN に NONE が指定されていると、DBName はフルパス名で入力する必要があります。

### Passwd

オプションです。Oracle Lite データベースの暗号化に使用されたパスワードです。このパスワードは、大 / 小文字は区別されません。パスワードを指定しないと、DECRYPDB により入力を要求するプロンプトが表示されます。

## コメント

Oracle Lite データベースに対してオープンされている接続がある場合は、データベースを復号化できません。

このユーティリティを別のアプリケーションからコールすると、結果は次のようになります。

リターン・コード	説明
EXIT_SUCCESS	成功。
EXIT_USAGE	コマンドライン引数の使用方法が適切でないか、引数にエラーがあります。
EXIT_PATH_TOO_LONG	パスが長すぎます。
EXIT_SYSCALL	復号化された新規コピーをディスク上に作成中に I/O エラーが発生しました。
EXIT_BAD_PASSWD	不適切なパスワードが指定されました。

詳細は、「[ENCRYPDB](#)」のコメントを参照してください。

## ENCRYPDB

### 説明

このツールにより、パスワードを使用して Oracle Lite データベースを暗号化したり、データベースのパスワードを変更できます。パスワードを使用すると、データベースへの許可されていないアクセスを防止し、データベースを暗号化できるので、データベース・ファイル内に格納されているデータを解釈できないようにすることが可能です。詳細は、「[DECRYPDB](#)」を参照してください。

### 構文

```
ENCRYPDB DSN | NONE DBName [New_Passwd [Old_Passwd]]
```

### キーワードとパラメータ

- DSN**  
暗号化する Oracle Lite データベースのデータ・ソース名です。NONE を指定すると、DBName には（.ODB 拡張子を除く）完全修飾されたデータベース名を入力する必要があります。DSN に NONE 以外の値を指定する場合、ODBC.INI ファイル内のデータ・ソース名を指定する必要があります。
- DBName**  
暗号化するデータベース名です。DSN に NONE が指定されていると、DBName はフルパス名で入力する必要があります。

### New\_Passwd と Old\_Passwd

オプションです。データベースを暗号化する（または暗号化のために以前に使用された）パスワードです。このパスワードでは大 / 小文字は区別されず、128 文字まで指定できます。パスワードを指定しないと、ENCRYPDB により入力を要求するプロンプトが表示されます。どちらのパスワードもオプションであるため、ユーティリティを起動するときのコマンドラインは次の 3 つの書式になります。

- パスワードなし。データベースがすでに暗号化されている場合は、ENCRYPDB はユーザーがデータベースのパスワードを変更しようとしていると解釈します。古いパスワードの入力を 1 回、新しいパスワードの入力を 2 回要求され、新しいパスワードを使用してデータベースが暗号化されます。データベースが暗号化されていない場合、ENCRYPDB によって、新しいパスワードの入力が 2 回要求され、新しいパスワードを使用してデータベースが暗号化されます。
- 片方のパスワードを指定。このパスワードは新しいパスワードとして解釈されます。データベースがすでに暗号化されている場合、ENCRYPDB によって、古いパスワードの入力が要求され、新しいパスワードを使用してデータベースが暗号化されます。
- 両方のパスワードを指定。ENCRYPDB では、最初のパスワードが新しいパスワードで、2 番目のパスワードが古いパスワードであると解釈します。

## コメント

このユーティリティを別のアプリケーションからコールすると、結果は次のようになります。

リターン・コード	説明
EXIT_SUCCESS	成功。
EXIT_USAGE	コマンドライン引数の使用方法が適切でないか、引数にエラーがあります。
EXIT_PATH_TOO_LONG	パスが長すぎます。
EXIT_SYSCALL	暗号化された新規コピーをディスク上に作成中に I/O エラーが発生しました。
EXIT_BAD_PASSWD	不適切なパスワードが指定されました。

デフォルトの Oracle Lite データベース (**POLITE.ODB**) は暗号化されていません。Oracle Lite データベースを暗号化すると、暗号化された Oracle Lite データベースに対して接続を確立しようとするスキーマはすべて、有効なパスワードを提供する必要があります。パスワードが提供されない場合、Oracle Lite はエラーを返します。Oracle Lite データベースに対してオープンされている接続がある場合は、データベースを暗号化できません。

Oracle Lite データベースを暗号化および復号化する場合、次の点を考慮する必要があります。

- 暗号化されたデータベースは、パスワードなしでは復号化できない。データベースは暗号化する前に、必ず安全な場所にバックアップを作成します。
- 他の Oracle Lite アプリケーションを実行中は、ENCRYPDB を実行しない。データベース・ファイルに他のアプリケーションが接続している場合、エラーが表示されます。
- データベースを暗号化した後、そのデータベースに接続するには、接続文字列にパスワードを含める必要がある。
- パスワードによってデータベース全体が暗号化される。ユーザー固有またはスキーマ固有のパスワードではありません。
- データベースを暗号化しても、第三者が Oracle Lite データベースを削除するのを防ぐことはできない。つまり、removedb と rmdb は、パスワードをチェックせずにデータベースを削除します。許可されていないユーザーがファイル・システムを操作できないように保護するツールを使用します。
- 暗号化された Oracle Lite データベースに接続する ODBC アプリケーションは、有効なパスワードを指定する必要がある。パスワードは通常、アプリケーション内でコーディングされずに、実行時に入力が必要されます。ほとんどの ODBC アプリケーションでは、Oracle Lite の ODBC ドライバが実行時にパスワードの入力を要求する場合、SQLConnect 関数ではなく、SQLDriverConnect 関数で DRIVER= オプションを指定できます。
- Oracle Lite のこのリリースに付属しているサンプル・アプリケーションはすべて、暗号化されていないデータベースに対して実行できる。
- DECRYPTDB と ENCRYPDB を使用して（この順番で）、データベースのパスワードを変更できる。ただし、DECRYPTDB によって Oracle Lite データベースが最初にプレーン・テキストで作成され、次に ENCRYPDB によって暗号化されます。プレーン・テキスト形式のデータベースが一時的に作成されるため、この方法はお勧めできません。
- 暗号化されたデータベースに関しては、ユーザー名とパスワードがすべて **DSN.OPW** というファイルに書き込まれる。これで、各ユーザーは **.ODB** ファイルがアクセスされる前に、パスワードをキーとして使用して **.OPW** ファイルをロック解除できます。データベースをコピーまたはバックアップするときは、**.OPW** ファイルを含める必要があります。



# MIGRATE

## 説明

Oracle Lite データベースを前のバージョンからこのリリースに移行するユーティリティです。このユーティリティは、Oracle Lite 3.5 または 3.6 のデータベースを移行し、.35 または .36 の拡張子を付けたバックアップ・コピーを作成します。以前のリリースの Oracle Lite がある場合、詳細は、第 1 章「作業の準備」の「データベースの移行」を参照してください。

このユーティリティを使用する前に、Oracle Lite の現リリースをインストールしておく必要があります。また、データベースが暗号化してある場合は、このユーティリティを使用する前に復号化しておく必要があります。

## 構文

```
MIGRATE DSN DBName
```

*DBName* は、データベース名またはデータベース名とパス名のいずれでもかまいません。

## キーワードとパラメータ

### DSN

移行するデータベースのデータ・ソース名です。**ODBC.INI** ファイルにあるデフォルト・データベース・ディレクトリを参照するために使用されます。

### DBName

移行するデータベース名、またはパスとデータベース名です。データベースのみを指定した場合は、**ODBC.INI** ファイルの（データ・ソース名の下の）DataDirectory パラメータに指定されているディレクトリにデータベース・ファイルが必要です。

## コメント

この項で説明したとおり、このユーティリティを使用する前に Oracle Lite をインストールする必要があります。

MIGRATE ユーティリティにより生成されるメッセージは、画面上のコマンド・ウィンドウに表示されます。

このユーティリティを使用すると、既存の Oracle Lite データベースを圧縮できます。

## 例

```
MIGRATE polite db1  
MIGRATE polite c:\testdir\db1.odp
```

## ODBC Administrator と Oracle Lite ODBC ドライバ

データ・ソース名（DSN）により、Oracle Lite ODBC ドライバと、そのドライバを介してアクセスする Oracle Lite データベースが関連付けられます。Oracle Lite のインストール・プロセスにより、デフォルト DSN の POLITE が Oracle Lite データベース用に作成されます。追加作成する Oracle Lite データベースには、追加 DSN を作成することもできます。

ODBC Administrator は Microsoft 社が提供するツールです。Windows 95/98 および Windows NT の **ODBC.INI** ファイルおよびそれに関連付けられたレジストリ・エントリを管理します。**ODBC.INI** ファイルと Windows レジストリには、ODBC Administrator によりキャプチャされる DSN エントリが格納されます。ODBC Administrator を使用すると、DSN を Oracle Lite ODBC ドライバに関連付けられます。

**注意：** このマニュアルでは、ODBC Administrator の使用方法については説明していません。この情報は、オンライン・ヘルプを参照してください。

ODBC Administrator では DSN 以外にも次のパラメータを指定します。

DSN パラメータ	説明
データの記述	データ・ソースのオプションの記述。
データベース・ディレクトリ	データベースが存在するデータ・ディレクトリのパス。これは既存のパスです。
データベース	作成する Oracle Lite データベースの名前。 <b>.ODB</b> 拡張子は付けなくてもください。
デフォルトの分離レベル	異なるトランザクション間で、どの程度操作をお互いに参照できるようにするかを決定します。サポートされている分離レベルの詳細は、 <a href="#">第 2 章「アプリケーションの開発」</a> の「 <a href="#">分離</a> 」を参照してください。
自動コミット	トランザクション内のデータベース更新操作が実行されるたびに、更新操作をコミットします。自動コミットの値は、「Off」または「On」です。

DSN パラメータ	説明
デフォルトのカーソル・タイプ	<ul style="list-style-type: none"><li>■ <i>Dynamic</i>: カーソルをオープンした後、結果セットのメンバー、順序または値に変更があった場合に検出できます。動的カーソルが行をフェッチし、その後、別アプリケーションによりその行が削除または更新された場合、動的カーソルは再フェッチしたときにこれらの変更を検出します。</li><li>■ <i>Forward Only</i>: 結果セット全体を前方にのみ進み、後方には進まないスクロールできないカーソル。この結果、このカーソルは前にフェッチされた行には戻れません。</li><li>■ <i>Keyset Driven</i>: 結果セットのメンバーまたは順序に対する変更は検出しませんが、結果セットの行の値に対する変更は検出します。</li><li>■ <i>Static</i>: カーソルをオープンした後、結果セットのメンバー、順序または値に対する変更を検出しません。静的カーソルが行をフェッチした後、別のアプリケーションによりその行が更新された場合、このカーソルは行を再フェッチした場合でもこの変更を検出しません。</li></ul>

たとえば、**ODBC.INI** ファイルの **POLITE** の DSN エントリには次のような行が含まれます。

```
[POLITE]
Description=Oracle Lite Data Source
DataDirectory=C:\ORANT\OLDB40
Database=POLITE
IsolationLevel=Repeatable Read
CursorType=Dynamic
```

ODBC Administrator を使用した DSN の追加

ODBC Administrator を使用して DSN を追加するには、次を実行します。

- 1. ODBC Administrator を起動します。これには、「Oracle for Windows NT or 95」プログラム・グループにある ODBC Administrator のアイコンを選択するか、または DOS プロンプトで次をタイプします。  
  
C:\>ODBCAD32
- 2. 「追加」をクリックします。
- 3. インストール済の ODBC ドライバのリストから、「Oracle Lite nm ODBC ドライバ」（nm はリリース番号）をダブルクリックします。
- 4. 次に、DSN 名を追加し、ODBC ドライバのセットアップ用ダイアログ・ボックスにパラメータを定義します。パラメータの定義については、前述の表を参照してください。

## 読取り専用メディア（CD-ROM）を指す DSN の追加

1. 「[ODBC Administrator を使用した DSN の追加](#)」で説明されている方法で DSN を作成します。
2. **ODBC.INI** ファイルの新規 DSN に次の行を追加します。

```
ReadOnly = TRUE
```

---

---

**注意：** CD-ROM 上のファイルを指す DSN を定義できます。DSN で CD-ROM ドライブおよびディレクトリを指すようにし、データベース・ファイルのファイル名を指定します。次に、**ODBC.INI** ファイルのデータ・ソース定義に「ReadOnly = TRUE」という行を追加します。ODBC プログラムは、(**ODBC.INI** ファイルに行を追加するかわりに) データベースをオープンする前に次をコールして、この機能を使用可能にできます。

```
SQLSetConnectOption( hdbc, SQL_ACCESS_MODE, SQL_MODE_READ_ONLY )
```

データベース・ファイルを読取り専用に設定すると、ログ・ファイルは作成されなくなります。更新、挿入、削除またはコミットは、表のメモリー内イメージに対して操作されているように見えますが、コミットしても、これらの変更はデータベース・ファイルに書き込まれません。アプリケーションを終了し、再接続し、問合せを発行すると、元のデータが表示されます。

---

---

## ODBC 拡張機能

このリリースでは、次の ODBC 拡張機能がサポートされています。

- `SQLDriverConnect()` は、`DRIVER` キーワードを受け入れ、接続情報を要求するプロンプトを表示します。
- ODBC 2.0 の `SQLSetPos()` 機能は、`REFRESH`、`DELETE` および `UPDATE` オプションをサポートします。
- このリリースでは、ODBC の外部結合構文 "{oj ...}" をサポートします。
- `SQLProcedureColumns()` は、直接コールできるストアド・プロシージャに対して引数を正しい順序で返します。
- `SQLGetStmtOption()` は、`SQL_ROW_NUMBER` オプションをサポートします。

# ODBINFO

## 説明

ODBINFO は、Oracle Lite データベースのバージョン・ナンバーとボリューム ID を見つけるために使用できます。また、ODBINFO はパラメータをいくつか表示し設定することもできます。

## 構文

変更を加えずに現在の情報を表示するには、次の構文を使用します。

```
odbinfo [-p passwd] DSN DBName
```

また、次の構文も使用できます。

```
odbinfo [-p passwd] NONE dbpath¥dbname.odb
```

例を次に示します。

```
odbinfo -p tiger polite polite
odbinfo NONE c:¥orant¥oldb40¥polite.odb
```

データベースを暗号化してある場合は、パスワードを含める必要があります。

オプションを設定または消去するには、DSN または NONE の前に「+」または「-」引数を 1 つ以上使用します。例を次に示します。

```
odbinfo +reuseoid -pagelog -fsync polite polite
```

## パラメータ

ODBINFO では次のパラメータを使用できます。

パラメータ	説明
pagelog	デフォルトでは、コミットにより、 <b>filename.ODB</b> に変更が実際に書き込まれる前に、変更されたデータベース・ページが <b>filename.plg</b> にバックアップされます。コミット中にアプリケーションまたはオペレーティング・システムで障害が発生した場合、トランザクションは次の接続時に完全にロールバックされます。 <b>-pagelog</b> を指定した場合、バックアップは作成されないため、障害が発生するとデータベースが破損する可能性があります。

パラメータ	説明
fsync	<p>Oracle Lite は、通常、データベースに関連付けられている変更済バッファをコミット時にすべてディスクに書き込むよう、オペレーティング・システムに対して強制します。このオプションが使用禁止になっていると（-fsync）、オペレーティング・システムは変更を後までメモリー内に保持しておくことができます。バッファがフラッシュされる前に（アプリケーションではなく）システムがクラッシュした場合、データベースも破損する可能性があります。</p> <p>odbinfo -fsync -pagelog を使用すると、多数の（自動コミットがオンに指定された）小規模なトランザクションや大規模な更新を伴うトランザクションを使用するアプリケーションのパフォーマンスが向上します。ただしデータベースが破損した場合、簡単にデータベースを修復したりデータをリカバリする方法はありません。このため、この2つのオプションは、(1) <b>.ODB</b> が定期的にバックアップされる場合、または (2) データベース内のデータが他のソースからリカバリできる場合にのみ、データベースの初期ロード中に消去するようにします。</p> <p>このオプションを使用しても、データベースをあまり更新しないアプリケーションには何の影響もありません。この場合は、トランザクション分離レベルを SINGLE USER に設定する方が効果があります。</p>
reuseoid	<p>Oracle Lite は、デフォルトでは、表にあるどの行の ROWID も表が削除されるまで再利用しません。削除されたオブジェクトにアクセスしようとすると、「スロットが削除されました」エラーが返されます。この場合、削除されたオブジェクトごとに2バイトのストレージが使用されるため、行が頻繁に挿入および削除される場合は、時間が経つとパフォーマンスが低下し使用ディスク領域が増えます。</p> <p>odbinfo +reuseoid を使用すると、削除された行の ROWID を新しい行で再利用できます。ただし、これでは削除されたオブジェクトが多数ある表の領域をすべて解放できないことがあります。最善の方法は、データベースの作成直後にこのオプションを設定することです。</p> <p>このオプションは、純粋にリレーショナルなアプリケーションの場合は安全です。ただし、ROWID を使用するアプリケーションやオブジェクト間で直接ポインタを使用する JAC/OKAPI アプリケーションの場合は、オブジェクトが削除される前に、そのオブジェクトに対する参照がすべて NULL に設定されるようにする必要があります。そうでないと、参照先のない参照が別の無関係のオブジェクトを指してしまう場合があります。</p>

パラメータ	説明
compress	<p>このオプション（デフォルトは「on」）は、オブジェクトの実行長の圧縮を可能にします。実行長の圧縮には CPU 時間がほとんどかからないため、このオプションの選択を解除する（-compress）のは、次の場合のみです。</p> <ul style="list-style-type: none"><li>■ オペレーティング・システム・レベルのファイル圧縮（ドライブスペースや NTFS 圧縮属性など）が使用される場合。この場合、同一データを 2 回圧縮しないので圧縮率が上がります。</li><li>■ データベース内のほとんどのオブジェクトが高度に圧縮可能な状態（たとえば列がすべて NULL に設定されるなど）に頻繁に更新され、（ランダム・データを指定したバイナリ列のように）データがうまく圧縮できない場合。このような場合は、このオプションを指定する（+compress）と、表が非常に断片化される可能性があります。</li></ul> <p>このオプションを変更しても、データベース内の既存のオブジェクトは圧縮も圧縮解除もされません。</p>

# OLitRM40

## 説明

失敗して変更をロールバックできないトランザクション用に提供されている Oracle Lite の Recovery Manager です。OLitRM40 は常にトランザクションを監視し、失敗したトランザクションのロールバックとクリーンアップを実行します。

Oracle Lite をインストールすると、OLitRM40 は、「スタートアップ」プログラム・グループに入ります。OLitRM40 は自動的に起動されます。必要であれば、Oracle\_Home¥bin ディレクトリから手動で起動することもできます。

## 構文

OLitRM40 を起動するには、コマンド・プロンプトで次を入力します。

c:¥OLitRM40

OLitRM40 を停止するには、コマンド・プロンプトで次を入力します。

c:¥OLitRM40 -k

# OLLOAD

## 説明

このコマンドライン・ツールを使用すると、Oracle Lite データベースにある表に外部ファイルからデータをロードしたり、Oracle Lite データベースの表から外部ファイルにデータをアンロード（ダンプ）できます。SQL\*Loader と異なり、OLLOAD では制御ファイルを使用しません。データ・パラメータと書式情報は、すべてコマンドラインに指定します。

データのロード時、OLLOAD は、フィールドがセパレータ文字で区切られたレコードが、1 行に 1 レコードずつ含まれている入力ファイルを取ります。デフォルトのフィールド・セパレータはカンマ (,) です。レコードには、引用符のついた文字列を値として持つフィールドも含めることができます。デフォルトは一重引用符 (') です。データ解析の詳細は、「[コメント](#)」を参照してください。

## 構文

**データ・ファイルをロードするには、次のように指定します。**

```
olload [options] -load dbpath tbl [col1 col2 ...] [<datafile]
```

**出力ファイルにアンロード（ダンプ）するには、次のように指定します。**

```
olload [options] -dump dbpath tbl [col1 col2 ...] [>outfile]
```

## キーワードとパラメータ

### [options]

オプションのリストは、「[オプション](#)」を参照してください。

### -load

ロード・ユーティリティを使用します。

### -dump

アンロード（ダンプ）ユーティリティを使用します。

### dbpath

Oracle Lite データベース・ファイル（.ODB ファイル）へのパスです。

### tbl

表の名前です。OLLOAD は、まずユーザーが指定した大文字 / 小文字の形式で表の名前を見つけようとします。これに失敗した場合、ユーザーが指定した名前を大文字にして検索します。

---

---

**注意：** デフォルト・ユーザーは「SYSTEM」です。別のユーザー名の表に対して OLLOAD 操作を指定する場合は、tbl パラメータの前にユーザー名とピリオド (.) を指定します。

---

---



**col1 col2**

列名です。OLLOAD は、まずユーザーが指定した大文字 / 小文字の形式で列の名前を見つけようとします。これに失敗した場合、ユーザーが指定した名前を大文字にして検索します。

**[datafile] [outfile]**

ロードまたはアンロード（ダンプ）操作の対象となるソース・ファイルまたは宛先ファイルです。datafile または outfile を指定しないと、OLLOAD は出力を画面に表示します。

## オプション

**-sep character**

フィールド・セパレータです。このオプションを指定しないと、セパレータ文字はカンマ (,) であると解釈されます。

**-quote character**

引用符文字です。このオプションを指定しないと、引用符文字は一重引用符 (') であると解釈されます。

**-file filename**

このオプションは、データのロードおよびアンロード時にソース・ファイル名または宛先ファイル名を指定するために使用します。データのロード時には、*filename* は Oracle Lite データベースにロードされるソース・ファイルを示します。データのアンロード（ダンプ）時には、アンロードされるデータの宛先ファイルを示します。

---

---

**重要：** Oracle Lite データベースからデータをアンロードし、このデータを別の Oracle Lite データベースにロードする（つまりパイプ処理する）場合は、このオプションにファイル名を指定してはいけません。構文の例は、「例」にある 2 番目の例を参照してください。

---

---

**-log logfile**

このオプションは、OLLOAD がロード中に挿入できなかった行をリストしたログ・ファイルを生成する場合に指定します。ログ・ファイルを指定しないと、ロードは最初のエラーで停止します。

**-passwd passwd**

暗号化されているデータベースの接続パスワードです。ロードとアンロードを実行するには、このパスワードを指定する必要があります。

**-nosingle**

このオプションは、シングル・ユーザー・モードを使用しない場合に指定します。このオプションを指定するとパフォーマンスが低下しますが、データベースに対して他の接続が可能になります。

**-readonly**

このオプションは、読取り専用の Oracle Lite データベース（たとえば CD-ROM 上のデータベース）からデータをアンロードするときに指定します。

**-commit count**

このオプションは、指定された数の行を処理した後に OLOAD でコミットを実行する場合に使用します。デフォルトは 10000 です。OLOAD は、指定された数の行をコミットするたびに、画面にアスタリスク（\*）を出力します。コミット操作を使用禁止にするには、「0」を指定します。

**-mark count**

このオプションは、指定された数のレコードを処理した後に OLOAD で画面にドットを出力する場合に使用します。デフォルトは 1000 です。この機能を使用禁止にするには、「0」を指定します。

コメント

**データ解析**

OLOAD のデータ解析の例を次に示します。

入力	データ	説明
'Redwood Shores, CA'	Redwood Shores, CA	エスケープ・シーケンス（2 個の一重引用符）で 1 個の一重引用符を表します。
'O'Brien'	O'Brien	入力文字列を引用符で囲むと、文字列内の空白と句読点が保持されます。
fire fly	firefly	引用符で囲まれていないデータ内の空白は無視されます。
,	NULL,NULL	空のフィールドは NULL です。
1,,3,	1,NULL,3,NULL	空のフィールドは NULL です。
	[ 行が挿入されていない ]	完全に空の行は無視されます。

データベースの列数を超えた列の値は無視されます。行の終わりに値がない場合は、NULL に設定されます。

**OLOAD ユーティリティの制約**

OLOAD は、タブで区切られた入力ファイルと LONG データ型はサポートしません。

**例**

```
olload -quote ¥" -file p_kakaku.csv -load c:¥orant¥oldb40¥polite.odb skkm01
```

```
olload -dump c:¥orant¥oldb40¥polite.odb emp empno ename | olload -load myfile.odb  
myemp
```

**Oracle8 Navigator**

Oracle8 Navigator は、Oracle Lite データベースを管理するための GUI ツールです。Oracle8 Navigator は、データベース接続やデータベース・オブジェクトへのアクセスの他、さらに次のことを行うために使用します。

- ビュー、表、順序、シノニムおよびスナップショットを作成する。
- データベースを暗号化および復号化する。
- データを入力する。
- バックアップおよび復元する。
- 既存データをロードする。
- データベース間または同一データベース内で表をコピー・アンド・ペーストする。
- 索引を作成する。
- レプリケーションを実行する。

**Oracle8 Navigator の使用方法**

- Oracle8 Navigator の左のペインにある任意のオブジェクトをクリックすると、選択したオブジェクトの内容が右のペインに表示されます。
- 左のペインにあるボタン（オブジェクトまたはフォルダの横に表示される「+」記号または「-」記号）をクリックすると、オブジェクトやフォルダが拡張または縮小され、そこに含まれるオブジェクトが表示されます。
- 右のペインにある任意のオブジェクトをダブルクリックすると、選択したオブジェクトの内容が表示されるか、デフォルト操作（オブジェクトのオープンなど）がオブジェクトに対して実行されます。
- 右のペインにある項目をドラッグし、左のペインにあるオブジェクトまたはフォルダ上にドロップすると、選択された項目が左のペインにあるオブジェクトまたはフォルダ内にコピーされます。
- 左のペインまたは右のペインにある任意のオブジェクトを右ボタンでクリックすると、そのオブジェクトに対して実行できるアクションのリストが表示されます。

## Oracle8 Navigator による EMP 表の検査

Oracle8 Navigator を使用して EMP 表を検査するには、次の手順に従います。

1. Oracle8 Navigator が起動していない場合は、起動します。
2. 「Oracle Lite データベース」フォルダをダブルクリックします。
3. 「デフォルト Oracle Lite データベース」フォルダをダブルクリックします。
4. 「表」フォルダをダブルクリックします。
5. 右のウィンドウにある EMP 表のアイコンを右ボタンでクリックします。
6. 「オープン」メニュー・オプションを選択して、EMP 表のデータを表示します。

または

「プロパティ」メニュー・オプションを選択して、EMP 表のプロパティを表示します。

「EMP プロパティ」ダイアログ・ボックスの「一般」タブまたは「設計」タブを選択することにより、EMP 表のプロパティをさらに検査できます。

7. 区切り付きの書式でファイルにデータをエクスポートする場合は、「ファイルにエクスポート」メニュー・オプションを選択します。

詳細は、Oracle8 Navigator のオンライン・ヘルプを参照してください。

---

---

**重要：** Oracle8 Navigator のオンライン・ヘルプでは、Personal Oracle に言及する箇所が頻繁に出てきます。Personal Oracle で使用できる機能の中には、Oracle Lite では使用できないものがあります。

---

---

## Oracle Lite Designer

Oracle Lite Designer は、データベース・オブジェクトを表示し管理するための Java GUI ツールです。このツールは、『Oracle Lite Java 開発者ガイド』に詳しく説明されています。

## REMOVEDB

### 説明

データベースを削除するためのユーティリティです。

### 構文

`REMOVEDB DataSourceName DatabaseName`

## キーワードとパラメータ

### **DataSourceName**

削除するデータベースのデータ・ソース名です。

### **DatabaseName**

削除するデータベースの名前です。フルパス名、またはデータベース名のみを指定できます。データベース名のみを指定すると、データベースは、**ODBC.INI** ファイルで指定されたデータ・ソース名のデータ・ディレクトリから削除されます。

## コメント

REMOVEDB で削除されたデータベースは、引き続き Oracle8 Navigator に表示されます。Oracle8 Navigator に表示される削除されたデータベースに接続しようとする、かわりにデフォルトのデータベースに接続されます。Oracle8 Navigator を使用してデータベースを削除することをお勧めします。

## 例

```
removedb polite db1  
removedb polite c:¥testdir¥db2.odbc
```



---

## パフォーマンス・チューニング

この付録では、SQL 問合せのパフォーマンスを向上させるためのヒントを示します。説明する内容は次のとおりです。

- 単一表問合せの最適化
- 結合問合せの最適化
- ORDER BY および GROUP BY 句による最適化

例では、次のデータベース・スキーマを使用します。

表	列	主キー	外部キー
LOCATION	LOC# LOC_NAME	LOC#	
EMP	SS# NAME JOB_TITLE WORKS_IN	SS#	WORKS_IN は DEPT (DEPT#) を参照
DEPT	DEPT# NAME BUDGET LOC MGR	DEPT#	LOC は LOCATION (LOC#) を参照 MGR は EMP (SS#) を参照

単一表問合せの最適化

特定の列の値を基に表の行を選択する問合せのパフォーマンスを向上するには、その列の索引を作成します。たとえば、次の問合せは、EMP 表の NAME 列に索引が作成されている方が、パフォーマンスが良くなります。

```
SELECT *  
FROM EMP  
WHERE NAME = 'Smith';
```

結合問合せの最適化

次のようにすると、結合問合せ（FROM 句に複数の表参照が使用されている問合せ）のパフォーマンスが向上します。

内部表の結合列に対する索引の作成

次の例では、結合問合せの内部表は DEPT で、DEPT の結合列は DEPT# です。DEPT.DEPT# の索引により、問合せのパフォーマンスが向上します。この例では、DEPT# が DEPT の主キーであるため、その索引はすでに暗黙的に作成されています。

```
SELECT e.SS#, e.NAME, d.BUDGET  
FROM EMP e, DEPT d  
WHERE e.WORKS_IN = DEPT.DEPT#  
AND e.JOB_TITLE = 'Manager';
```



**FROM 句における表の順序の設定（問合せオプティマイザをバイパスする場合）**

Oracle Lite の問合せオプティマイザにより、表を結合する最適な順序が決定されます。ただし、オプティマイザをバイパスする場合は、**POLITE.INI** ファイル内の **ReverseJoinOrder** パラメータを使用して、FROM 句内の順序に関して、表の結合順序を指定できます。詳細は、**付録 A「POLITE.INI のデータベース・パラメータ」**を参照してください。

FROM 句の最初の表は、問合せにおけるすべての表の中で、最も少ない行数を選択する表になります。FROM 句の 2 番目の表は、最初の表に結合される表です。同様に、3 番目の表は 2 番目の表に結合される、というように続きます。

たとえば、各部門とその管理者の名前を選択する場合は、次の 2 通りの問合せを作成できます。

```
SELECT d.NAME, e.NAME
FROM DEPT d, EMP e
WHERE d.MGR = e.SS#
```

または

```
SELECT d.NAME, e.NAME
FROM EMP e, DEPT d
WHERE d.MGR = e.SS#
```

部門数が 10、従業員数が 1000 名で、各問合せの内部表の結合列には索引が作成されているとします。最初の問合せでは、最初の表から 10 行の該当行が生成されます（この場合は表全体）。2 番目の問合せでは、最初の表から 1000 行の該当行が生成されます。このように、EMP 表の SS# 列に索引がある場合は、最初の問合せがより高速に実行されます。

別の例として、ニューヨークなど、特定の場所における社会保障番号と従業員名を取り出すための問合せを考えてみます。このサンプルのスキーマによると、問合せでは FROM 句に 3 種類の表参照があります。これら 3 種類の表は、6 通りの順序に設定できます。どの順序を選んでも結果は同じになりますが、パフォーマンスに大きな違いが生じる可能性があります。

ニューヨーク内の場所は少数（1 または 2）であると想定します。上記の規則を基にすると、LOCATION 表が FROM 句の最初の表になります。LOCATION は DEPT と結合する必要があるため、DEPT は 2 番目の表で、DEPT の LOC 列に索引が必要です。同様に、3 番目の表は EMP で、EMP# に索引が必要です。この問合せは次のように作成できます。

```
SELECT e.SS#, e.NAME
FROM LOCATION l, DEPT d, EMP e
WHERE l.LOC_NAME = 'New York' AND
l.LOC# = d.LOC AND
d.DEPT# = e.WORKS_IN;
```

## ORDER BY および GROUP BY 句による最適化

SELECT 文の実行速度の向上とメモリー・キャッシュの消費量の低減を目標に、様々なパフォーマンス改善が行われてきました。GROUP BY および ORDER BY 句は、適切な索引が利用できる場合は、ソートを回避しようとしています。

### IN 副問合せ変換

副問合せ内の選択リストに一意の索引が作成されたときに、IN 副問合せを結合文に変換します。

たとえば、次の IN 副問合せ文は対応する結合文に変換されます。ここで、c1 は表 t2 の主キーであるとしています。

```
SELECT c2 FROM t1 WHERE  
c2 IN (SELECT c1 FROM t2);
```

```
SELECT c2 FROM t1, t2 WHERE t1.c2 = t2.c1;
```

### GROUP BY を使用しない ORDER BY 最適化

次の条件がすべて満たされた場合、SELECT 文内の ORDER BY 句に対するソート手順が不要になります。

1. すべての ORDER BY 列が昇順または降順に並んでいる。
2. ORDER BY 句に列のみが指定されている。つまり、ORDER BY 句に式が使用されていない。
3. ORDER BY 列がベース表索引の接頭辞である。
4. 索引によるアクセスの方が、結果セットのソートよりも安価である。

### ORDER BY を使用しない GROUP BY 最適化

GROUP BY 列がベース表索引の接頭辞である場合、グループ設定のソート手順が不要になります。

### GROUP BY を使用した ORDER BY 最適化

ORDER BY 列が GROUP BY 列の接頭辞で、すべての列が昇順または降順に並んでいる場合、問合せ結果のソート手順が不要になります。GROUP BY 列がベース表索引の接頭辞である場合、グループ設定のソート手順も不要になります。

**副問合せ結果のキャッシュ**

副問合せで返される行数が少なく、問合せが相関していないとオプティマイザが判断した場合、パフォーマンスを向上させるために問合せ結果はメモリーにキャッシュされます。現在、行数は 2000 に設定されています。たとえば、次のとおりです。

```
select * from t1 where  
t1.c1 = (select sum(salary)  
from t2 where t2.deptno = 100);
```



---

# 索引

## A

---

ALL\_CONS\_COLUMNS  
システム・カタログ・オブジェクト, B-3  
ALL\_CONSTRAINTS  
システム・カタログ・オブジェクト, B-2  
ALL\_IND\_COLUMNS  
システム・カタログ・オブジェクト, B-4  
ALL\_INDEXES  
システム・カタログ・オブジェクト, B-3  
ALL\_OBJECTS  
システム・カタログ・オブジェクト, B-4  
ALL\_SEQUENCES  
システム・カタログ・オブジェクト, B-5  
ALL\_SYNONYMS  
システム・カタログ・オブジェクト, B-5  
ALL\_TAB\_COLUMNS  
システム・カタログ・オブジェクト, B-7  
ALL\_TAB\_COMMENTS  
システム・カタログ・オブジェクト, B-8  
ALL\_TABLES  
システム・カタログ・オブジェクト, B-5  
ALL\_USERS  
システム・カタログ・オブジェクト, B-8  
ALL\_VIEWS  
システム・カタログ・オブジェクト, B-8

## C

---

CAT  
システム・カタログ・オブジェクト, B-9  
COLUMN\_PRIVILEGES  
システム・カタログ・オブジェクト, B-9  
CREATEDB  
使用方法, C-2

## D

---

Delphi  
サンプル・アプリケーションの実行, 5-4  
DUAL  
システム・カタログ・オブジェクト, B-9

## J

---

Java アクセス・クラス (JAC), 2-4  
JDBC ドライバ, 2-3  
説明, 2-3

## M

---

MIGRATE  
使用方法, C-7

## N

---

Navigator  
機能, C-17

## O

---

OCA (Oracle Open Client Adapter)  
説明, 2-3  
OCI (Oracle Call Interface)  
説明, 2-3  
ODBC  
データ・ソース名, 3-2  
ODBC Administrator  
使用方法, C-8  
ODBC ドライバ, 2-3  
説明, 2-3

OLitRM40, C-13  
Oracle Call Interface (OCI)  
説明, 2-3  
Oracle Open Client Adapter (OCA)  
説明, 2-3

## P

---

POLDEMO.SQL, 3-7  
PowerBuilder  
サンプル・アプリケーションの実行, 5-6

## R

---

Recovery Manager, C-13  
REMOVEDB  
使用方法, C-18

## S

---

SNAPSHOTS  
システム・カタログ・オブジェクト, B-10  
SQL\*Loader, 3-10

## T

---

TABLE\_PRIVILEGES  
システム・カタログ・オブジェクト, B-12

## U

---

USER\_OBJECTS  
システム・カタログ・オブジェクト, B-12

## V

---

Visual Basic  
サンプル・アプリケーションの実行, 5-8

## あ

---

アプリケーション  
サンプルの使用方法, 5-1  
チューニング, 2-8  
暗号化、データベース, 3-11

## い

---

インタフェース, 2-2

## え

---

永続性, 2-8  
エクスポート, 3-9  
エクスポートとインポート, 3-9

## お

---

オブジェクト・カーネル API (OKAPI), 2-4

## か

---

開発インタフェース, 2-2  
オブジェクト・データベース開発用, 2-2  
リレーショナル・データベース開発用, 2-2

## こ

---

更新可能スナップショット  
定義, 6-2  
高速リフレッシュ  
スナップショット・ログ, 6-3  
コピー・アンド・ペースト, 3-9

## さ

---

作成  
ODBC データ・ソース名, 3-2  
スナップショット, 6-5  
データベース, 3-2  
サンプル・アプリケーションの使用方法, 5-1

## し

---

システム・カタログ・オブジェクト  
ALL\_CONS\_COLUMNS, B-3  
ALL\_CONSTRAINTS, B-2  
ALL\_IND\_COLUMNS, B-4  
ALL\_INDEXES, B-3  
ALL\_OBJECTS, B-4  
ALL\_SEQUENCES, B-5  
ALL\_SYNONYMS, B-5  
ALL\_TAB\_COLUMNS, B-7

ALL\_TAB\_COMMENTS, B-8  
ALL\_TABLES, B-5  
ALL\_USERS, B-8  
ALL\_VIEWS, B-8  
CAT, B-9  
COLUMN\_PRIVILEGES, B-9  
DUAL, B-9  
SNAPSHOTS, B-10  
TABLE\_PRIVILEGES, B-12  
USER\_OBJECTS, B-12

## す

---

スナップショット  
  更新可能, 6-2  
  作成, 6-5  
  種類, 6-2  
  定義, 6-2  
  ベース表, 6-6  
  読取り専用, 6-2  
スナップショット・サイト  
  定義, 6-2  
スナップショット・ログ  
  作成, 6-3

## せ

---

制約  
  スナップショットへの追加, 6-6  
接続  
  新規データベース, 3-3  
  レプリケーションの設定, 6-4

## て

---

データ整合性制約  
  スナップショットへの追加, 6-6  
データ・ソース名  
  ODBC の作成, 3-2  
データベース  
  Oracle, 6-2  
  新規作成, 3-2  
  新規への接続, 3-3  
  スナップショット・サイト, 6-2  
  バックアップ, 3-10  
  マスター・サイト, 6-2

データベースへの値の移入  
  Navigator, 3-7  
  SQL\*Plus, 3-7  
データベース・ユーティリティ  
  CREATEDB, C-2  
  MIGRATE, C-7  
  ODBC Administrator, C-8  
  REMOVEDB, C-18  
デモ表の作成, 3-7

## と

---

問合せオブティマイザ, A-6, D-3  
トランザクション・サポート, 2-5

## は

---

パスワード, 3-11  
バックアップ、データベース, 3-10

## ひ

---

表  
  デモ, 3-7

## へ

---

ベース表  
  制約の追加, 6-6

## ま

---

マスター・サイト  
  接続, 6-4  
  定義, 6-2  
マスター表  
  定義, 6-2  
  レプリケーションの準備, 6-4

## よ

---

読取り専用スナップショット  
  定義, 6-2

## り

---

リモート・データベース, 6-2

## れ

---

レプリケーション

サーバー・サポートの確認, 6-3

接続の設定, 6-4

定義, 6-2

マスター表の準備, 6-4

レプリケーション・エンジン

設定, 6-3