1

Introduction to Get Started with Oracle Database Development

An Oracle Database developer is responsible for creating or maintaining the database components of an application that uses the Oracle technology stack. Oracle Database developers either develop applications or convert existing applications to run in the Oracle Database environment.

> **See Also:**
>
> *Oracle Database Concepts* for more information about the duties of Oracle Database developers

About This Document

This document is the entry into the Oracle Database documentation set for application developers.

This document does the following:

- Explains the basic concepts behind development with Oracle Database
- Shows, with tutorials and examples, how to use basic features of SQL and PL/SQL
- Provides references to detailed information about subjects that it introduces
- Shows how to develop and deploy a simple Oracle Database application

Introduction to Get Started with Oracle Database Development (this chapter) describes the reader for whom this document is intended, outlines the organization of this document, introduces important Oracle Database concepts, and describes the sample schema used in the tutorials and examples in this document.

Connecting to Oracle Database and Exploring It explains how to connect to Oracle Database, how to view schema objects and the properties and data of Oracle Database tables, and how to use queries to retrieve data from an Oracle Database table.

About DML Statements and Transactions introduces data manipulation language (DML) statements and transactions. DML statements add, change, and delete Oracle Database table data. A transaction is a sequence of one or more SQL statements that Oracle Database treats as a unit: either all of the statements are performed, or none of them are.

Creating and Managing Schema Objects introduces data definition language (DDL) statements, which create, change, and drop schema objects.

Developing Stored Subprograms and Packages introduces stored subprograms and packages, which can be used as building blocks for many different database applications.

Using Triggers introduces triggers, which are stored PL/SQL units that automatically run ("fire") in response to specified events.

Working in a Global Environment introduces globalization support—National Language Support (NLS) parameters and Unicode-related features of SQL and PL/SQL.

Building Effective Applications explains how to build scalable applications and use recommended programming and security practices.

Developing a Simple Oracle Database Application shows how to develop a simple Oracle Database application.

Deploying an Oracle Database Application explains how to deploy an Oracle Database application—that is, how to install it in one or more environments where other users can run it—using the application developed in Developing a Simple Oracle Database Application as an example.

About Oracle Database

Oracle Database groups related information into logical structures called **schemas**. The logical structures contain **schema objects**.

When you connect to the database by providing your user name and password, you specify the schema and indicate that you are its owner. In Oracle Database, the user name and the name of the schema to which the user connects are the same.

About Schema Objects

Every object in an Oracle Database belongs to only one schema, and has a unique name with that schema.

Some of the objects that schemas can contain are:

- **Tables**

  Tables are the basic units of data storage in Oracle Database. Tables hold all user-accessible data. Each table contains **rows** that represent individual data **records**. Rows are composed of **columns** that represent the **fields** of the records.

- **Indexes**

  Indexes are optional objects that can improve the performance of data retrieval from tables. Indexes are created on one or more columns of a table, and are automatically maintained in the database.

- **Views**

  You can create a view that combines information from several different tables into a single presentation. A view can rely on information from both tables and other views.

- **Sequences**

  When all records of a table must be distinct, you can use a sequence to generate a serial list of unique integers for numeric columns, each of which represents the ID of one record.

- **Synonyms**

Synonyms are aliases for schema objects. You can use synonyms for security and convenience; for example, to hide the ownership of an object or to simplify SQL statements.

- **Stored subprograms**

  Stored subprograms (also called **schema-level subprograms**) are procedures and functions that are stored in the database. They can be invoked from client applications that access the database.

  Triggers are stored subprograms that are automatically run by the database when specified events occur in a particular table or view. Triggers can restrict access to specific data and perform logging.

- **Packages**

  A package is a group of related subprograms, along with the explicit cursors and variables they use, stored in the database as a unit, for continued use. Like stored subprograms, package subprograms can be invoked from client applications that access the database.

Typically, the objects that an application uses belong to the same schema.

See Also:

- *Oracle Database Concepts* for a comprehensive introduction to schema objects
- Creating and Managing Tables
- Managing Indexes
- Creating and Managing Views
- Creating and Managing Sequences
- Creating and Managing Synonyms
- Developing Stored Subprograms and Packages
- Using Triggers

About Oracle Database Access

You can access Oracle Database only through a client program, such as SQL*Plus or SQL Developer.

The client program's interface to Oracle Database is Structured Query Language (SQL). Oracle provides an extension to SQL called Procedural Language/SQL (PL/SQL).

About SQL*Plus

SQL*Plus (pronounced *sequel plus*) is an interactive and batch query tool that is installed with every Oracle Database installation. It has a command-line user interface that acts as the client when connecting to the database.

SQL*Plus has its own commands and environment. In the SQL*Plus environment, you can enter and run SQL*Plus commands, SQL statements, PL/SQL statements, and operating system commands to perform tasks such as:

- Formatting, performing calculations on, storing, and printing query results
- Examining tables and object definitions
- Developing and running batch scripts
- Performing database administration

You can use SQL*Plus to generate reports interactively, to generate reports as batch processes, and to output the results to text file, to screen, or to HTML file for browsing on the Internet. You can generate reports dynamically using the HTML output facility.

You can use SQL*Plus in SQL Developer. For details, see *Oracle SQL Developer User's Guide*.

> **See Also:**
>
> - "Connecting to Oracle Database from SQL*Plus"
> - *SQL*Plus User's Guide and Reference* for information about SQL*Plus

About SQL Developer

SQL Developer (pronounced *sequel developer*) is a graphical user interface for Oracle Database, that is available in the default installation of Oracle Database and by free download from the Oracle Technology Network.

SQL Developer serves as a modern integrated development environment (IDE) for SQL and PL/SQL, and provides a graphical interface for managing database objects. You can also create reports, design data models, migrate third-party databases to Oracle, REST-enable tables and views, and deploy and manage Oracle REST Data Services. The SQL Worksheet allows you to enter and run SQL statements, PL/SQL statements, and SQL*Plus commands and scripts.

> **Note:**
>
> SQL Developer often offers several ways to do a task, but this document does not explain every possible way.

> **See Also:**
>
> - "Connecting to Oracle Database from SQL Developer"
> - *Oracle SQL Developer User's Guide* for information about SQL Developer

About Structured Query Language (SQL)

Structured Query Language (SQL) (pronounced *sequel*) is the set-based, high-level computer language with which all programs and users access data in Oracle Database.

SQL is a declarative, or nonprocedural, language; that is, it describes what to do, but not how. You specify the desired result set (for example, the names of current employees), but not how to get it.

See Also:

- *Oracle Database Concepts* for a complete overview of SQL
- *Oracle Database SQL Language Reference* for complete information about SQL

About Procedural Language/SQL (PL/SQL)

Procedural Language/SQL (PL/SQL) (pronounced *P L sequel*) is a native Oracle Database extension to SQL. It bridges the gap between declarative and imperative program control by adding procedural elements, such as conditional control and loops.

In PL/SQL, you can declare constants and variables, procedures and functions, types and variables of those types, and triggers. You can handle exceptions (runtime errors). You can create PL/SQL units—procedures, functions, packages, types, and triggers—that are stored in the database for reuse by applications that use any of the Oracle Database programmatic interfaces.

The basic unit of a PL/SQL source program is the block, which groups related declarations and statements. A block has an optional declarative part, a required executable part, and an optional exception-handling part.

See Also:

- *Oracle Database Concepts* for a complete overview of PL/SQL
- *Oracle Database PL/SQL Language Reference* for complete information about PL/SQL

About Other Client Programs, Languages, and Development Tools

Several other client programs, languages, and tools are available.

Note:

Some of the products on the preceding list do not ship with Oracle Database and must be downloaded separately.

See Also:

- *Oracle Database Concepts* for more information about tools for Oracle Database developers
- *Oracle Database Development Guide* for information about choosing a programming environment

Oracle Application Express

Oracle Application Express is an application development and deployment tool that enables you to quickly create secure and scalable web applications even if you have limited previous programming experience. The embedded Application Builder tool assembles an HTML interface or a complete application that uses schema objects, such as tables or stored procedures, into a collection of pages that are linked through tabs, buttons, or hypertext links.

See Also:

Oracle Application Express App Builder User's Guide for more information about Oracle Application Express

Oracle Java Database Connectivity (JDBC)

Oracle Java Database Connectivity (JDBC) is an API that enables Java to send SQL statements to an object-relational database, such as Oracle Database. Oracle Database JDBC provides complete support for the JDBC 3.0 and JDBC RowSet (JSR-114) standards, advanced connection caching for both XA and non-XA connections, exposure of SQL and PL/SQL data types to Java, and fast SQL data access.

See Also:

For more information about JDBC:

- *Oracle Database Concepts*
- *Oracle Database Development Guide*
- *Oracle Database Get Started with Java Development*

Hypertext Preprocessor (PHP)

The Hypertext Preprocessor (PHP) is a powerful interpreted server-side scripting language for quick web application development. PHP is an open source language that is distributed under a BSD-style license. PHP is designed for embedding database access requests directly into HTML pages.

Oracle Call Interface (OCI)

Oracle Call Interface (OCI) is the native C language API for accessing Oracle Database directly from C applications.

The OCI Software Development Kit is installed as part of the Oracle Instant Client, which enables you to run applications without installing the standard Oracle client or having an `ORACLE_HOME`. Your applications work without change, using significantly less disk space.

> **See Also:**
>
> - *Oracle Database Development Guide* for more information about OCI
> - *Oracle Call Interface Programmer's Guide* for complete information about OCI

Oracle C++ Call Interface (OCCI)

Oracle C++ Call Interface (OCCI) is the native C++ language API for accessing Oracle Database directly from C++ applications. Like OCI, OCCI supports both relational and object-oriented programming paradigms.

The OCCI Software Development Kit is also installed as part of the Oracle Instant Client, which enables you to run applications without installing the standard Oracle client or having an `ORACLE_HOME`. Your applications work without change, using significantly less disk space.

> **See Also:**
>
> - *Oracle Database Development Guide* for more information about OCCI
> - *Oracle C++ Call Interface Programmer's Guide* for complete information about OCCI

Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) is a set of database access APIs that connect to the database, prepare, and then run SQL statements on the database. An application that uses an ODBC driver can access nonuniform data sources, such as spreadsheets and comma-delimited files.

The Oracle ODBC driver conforms to ODBC 3.51 specifications. It supports all core APIs and a subset of Level 1 and Level 2 functions. Microsoft supplies the Driver manager component for the Windows platform.

Like OCI, OCCI, and JDBC, ODBC is part of the Oracle Instant Client installation.

See Also:

- *Oracle Database Concepts*
- *Oracle Services for Microsoft Transaction Server Developer's Guide for Microsoft Windows* for information about using the Oracle ODBC driver with Windows
- *Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems* for information about using Oracle ODBC driver on Linux

Pro*C/C++ Precompiler

The Pro*C/C++ precompiler lets you embed SQL statements in a C or C++ source file. The precompiler accepts the source program as input, translates the embedded SQL statements into standard Oracle runtime library calls, and generates a modified source program that you can compile, link, and run.

See Also:

- *Oracle Database Concepts* for more information about Oracle precompilers
- *Oracle Database Development Guide* for more information about the Pro*C/C++ precompiler
- *Pro*C/C++ Programmer's Guide* for complete information about the Pro*C/C++ precompiler

Pro*COBOL Precompiler

The Pro*COBOL precompiler lets you embed SQL statements in a COBOL source file. The precompiler accepts the source program as input, translates the embedded SQL statements into standard Oracle runtime library calls, and generates a modified source program that you can compile, link, and run.

See Also:

- *Oracle Database Concepts* for more information about Oracle precompilers
- *Oracle Database Development Guide* for more information about the Pro*COBOL precompiler
- *Pro*COBOL Programmer's Guide* for complete information about the Pro*COBOL precompiler

Microsoft .NET Framework

The Microsoft .NET Framework is a multilanguage environment for building, deploying, and running applications and XML web services.

The main components of the Microsoft .NET Framework are:

- Common Language Runtime (CLR)

  The Common Language Runtime (CLR) is a language-neutral development and runtime environment that provides services that help manage running applications.

- Framework Class Libraries (FCL)

  The Framework Class Libraries (FCL) provide a consistent, object-oriented library of prepackaged functionality.

Oracle Data Provider for .NET (ODP.NET)

Oracle Data Provider for .NET (ODP.NET) provides fast and efficient ADO.NET data access from .NET applications to Oracle Database. ODP.NET allows developers to take advantage of advanced Oracle Database functionality that exists in Oracle Database, including SecureFiles, XML DB, and Advanced Queuing.

Oracle Developer Tools for Visual Studio (ODT)

Oracle Developer Tools for Visual Studio (ODT) is a set of application tools that integrate with the Visual Studio environment. These tools provide graphic user interface access to Oracle functionality, enable the user to perform a wide range of application development tasks, and improve development productivity and ease of use. Oracle Developer Tools supports the programming and implementation of .NET stored procedures using Visual Basic, C#, and other .NET languages.

Oracle Providers for ASP.NET

Oracle Providers for ASP.NET offer ASP.NET developers an easy way to store state common to web applications within Oracle Database. These providers are modeled on existing Microsoft ASP.NET providers, sharing similar schema and programming interfaces to provide .NET developers a familiar interface. Oracle supports the Membership, Profile, Role, and other providers.

> **See Also:**
>
> - *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows*
> - *Oracle Database Development Guide*

Oracle Provider for OLE DB (OraOLEDB)

Oracle Provider for OLE DB (OraOLEDB) is an open standard data access methodology that uses a set of Component Object Model (COM) interfaces for accessing and manipulating different types of data. These interfaces are available from various database providers.

See Also:

Oracle Provider for OLE DB Developer's Guide for Microsoft Windows for more information about OraOLEDB

About Sample Schema HR

The HR sample schema can be installed with Oracle Database. This schema contains information about employees—departments, locations, work histories, and related information. Like all schemas, HR has tables, views, indexes, procedures, functions, and other attributes. The examples and tutorials in this document use the schema.

See Also:

- *Oracle Database Sample Schemas* for a complete description of the HR schema
- "Connecting to Oracle Database as User HR" for instructions for connecting to Oracle Database as the user HR