# Data Access

A general requirement for a DBMS is to adhere to accepted industry standards for a data access language.

## Structured Query Language (SQL)

42

SQL is a set-based declarative language that provides an interface to an RDBMS such as Oracle Database.

Procedural languages such as C describe *how* things should be done. SQL is nonprocedural and describes *what* should be done.

SQL is the ANSI standard language for relational databases. All operations on the data in an Oracle database are performed using SQL statements. For example, you use SQL to create tables and query and modify data in tables.

A SQL statement can be thought of as a very simple, but powerful, computer program or instruction. Users specify the result that they want (for example, the names of employees), not how to derive it. A SQL statement is a string of SQL text such as the following:

```
SELECT first_name, last_name FROM employees;
```

SQL statements enable you to perform the following tasks:

- Query data
- Insert, update, and delete rows in a table
- Create, replace, alter, and drop objects
- Control access to the database and its objects
- Guarantee database consistency and integrity

SQL unifies the preceding tasks in one consistent language. Oracle SQL is an implementation of the ANSI standard. Oracle SQL supports numerous features that extend beyond standard SQL.

> **See Also:**
>
> "SQL" to learn more about SQL standards and the main types of SQL statements

## PL/SQL and Java

42

**PL/SQL** is a procedural extension to Oracle SQL.

PL/SQL is integrated with Oracle Database, enabling you to use all of the Oracle Database SQL statements, functions, and data types. You can use PL/SQL to control the flow of a SQL program, use variables, and write error-handling procedures.

You can place SQL statements in source code and submit it to a precompiler or Java translator before compilation. Alternatively, you can eliminate the precompilation step and use an API such as Java Database Connectivity (JDBC) or Oracle Call Interface (OCI) to enable the application to interact with the database.

- Use server-side programming to develop data logic that resides in the database

  An application can explicitly invoke stored subprograms (procedures and functions), written in PL/SQL (pronounced *P L sequel*) or Java. You can also create a trigger, which is named program unit that is stored in the database and invoked in response to a specified event.

This chapter explains the second approach. The principal benefit of server-side programming is that functionality built into the database can be deployed anywhere. The database and not the application determines the best way to perform tasks on a given operating system. Also, subprograms increase scalability by centralizing application processing on the server, enabling clients to reuse code. Because subprogram calls are quick and efficient, a single call can start a compute-intensive stored subprogram, reducing network traffic.

You can use the following languages to store data logic in Oracle Database:

- PL/SQL

  PL/SQL is the Oracle Database procedural extension to SQL. PL/SQL is integrated with the database, supporting all Oracle SQL statements, functions, and data types. Applications written in database APIs can invoke PL/SQL stored subprograms and send PL/SQL code blocks to the database for execution.

- Java

  Oracle Database also provides support for developing, storing, and deploying Java applications. Java stored subprograms run in the database and are independent of programs that run in the middle tier. Java stored subprograms interface with SQL using a similar execution model to PL/SQL.

**See Also:**

- "Client-Side Database Programming" to learn about embedding SQL with precompilers and APIs
- *Oracle Database 2 Day Developer's Guide* for an introduction to Oracle Database application development
- *Oracle Database Development Guide* to learn how to choose a programming environment

42

# Overview of PL/SQL

PL/SQL provides a server-side, stored procedural language that is easy-to-use, seamless with SQL, robust, portable, and secure. You can access and manipulate database data using procedural objects called *PL/SQL units*.

PL/SQL units generally are categorized as follows:

The solution consists of client-side and server-side programmatic interfaces, tools to support Java development, and a Java Virtual Machine integrated with Oracle Database. All these products are compatible with Java standards.

The Java programming environment consists of the following additional features:

- Java stored procedures as the Java equivalent and companion for PL/SQL. Java stored procedures are tightly integrated with PL/SQL. You can call Java stored procedures from PL/SQL packages and procedures from Java stored procedures.
- The JDBC and SQLJ programming interfaces for accessing SQL data.
- Tools and scripts that assist in developing, loading, and managing classes.

## Java Stored Procedures

42

A **Java stored procedure** is a Java method published to SQL and stored in the database.

Like a PL/SQL subprogram, a Java procedure can be invoked directly with products like SQL*Plus or indirectly with a trigger. You can access it from any Oracle Net client —OCI, precompiler, or JDBC.

To publish Java methods, you write call specifications, which map Java method names, parameter types, and return types to their SQL counterparts. When called by client applications, a Java stored procedure can accept arguments, reference Java classes, and return Java result values.

Applications calling the Java method by referencing the name of the call specification. The run-time system looks up the call specification definition in the Oracle data dictionary and runs the corresponding Java method.

In addition, you can use Java to develop powerful programs independently of PL/SQL. Oracle Database provides a fully compliant implementation of the Java programming language and JVM.

> **See Also:**
>
> *Oracle Database Java Developer's Guide* explains how to write stored procedures in Java, how to access them from PL/SQL, and how to access PL/SQL functionality from Java

## Java and PL/SQL Integration

You can call existing PL/SQL programs from Java and Java programs from PL/SQL. This solution protects and leverages your PL/SQL and Java code.

Oracle Database offers two different approaches for accessing SQL data from Java: JDBC and SQLJ. JDBC is available on both client and server, whereas SQLJ is available only on the client.

### JDBC Drivers

JDBC is a database access protocol that enables you to connect to a database and run SQL statements and queries to the database.

based Java code. At run time, the program can communicate with multi-vendor databases using standard JDBC drivers.

The following example shows a simple SQLJ executable statement:

```
String name;
#sql  { SELECT first_name INTO :name FROM employees WHERE
employee_id=112 };
System.out.println("Name is " + name + ", employee number = " +
employee_id);
```

**See Also:**

- "SQLJ"
- *Oracle Database SQLJ Developer's Guide*

# Overview of Triggers

42

A database **trigger** is a compiled stored program unit, written in either PL/SQL or Java, that Oracle Database invokes ("fires") automatically in certain situations.

A trigger fires whenever one of the following operations occurs:

1. DML statements on a particular table or view, issued by any user

   DML statements modify data in schema objects. For example, inserting and deleting rows are DML operations.

2. DDL statements issued either by a particular user or any user

   DDL statements define schema objects. For example, creating a table and adding a column are DDL operations.

3. Database events

   User login or logoff, errors, and database startup or shutdown are events that can invoke triggers.

Triggers are schema objects that are similar to subprograms but differ in the way they are invoked. A subprogram is explicitly run by a user, application, or trigger. Triggers are implicitly invoked by the database when a triggering event occurs.

**See Also:**

- "Overview of SQL Statements" to learn about DML and DDL
- "Overview of Database Instance Startup and Shutdown"

See also dimension table; fact table.

**state object**

A session-level structure that contains metadata about the status of database resources such as processes, sessions, and transactions in the SGA.

**statement trigger**

A trigger that is fired once on behalf of the triggering statement, regardless of the number of rows affected by the triggering statement.

**statement-level atomicity**

The characteristic of a SQL statement as an atomic unit of work that either completely succeeds or completely fails.

**statement-level read consistency**

The guarantee that data returned by a single query is committed and consistent for a single point in time.

**statement-level rollback**

A database operation in which the effects of an unsuccessful SQL statement are rolled back because the statement caused an error during execution.

**stored procedure**

42

A named PL/SQL block or Java program that Oracle Database stores in the database. Applications can call stored procedures by name.

**Streams pool**

A memory pool that stores buffered queue messages.

**Structured Query Language (SQL)**

See SQL.

**subquery**

A query nested within another SQL statement. Unlike implicit queries, subqueries use a `SELECT` statement to retrieve data.