1

Introduction to Oracle SQL

Structured Query Language (SQL) is the set of statements with which all programs and users access data in an Oracle Database. Application programs and Oracle tools often allow users access to the database without using SQL directly, but these applications in turn must use SQL when executing the user's request. This chapter provides background information on SQL as used by most database systems.

This chapter contains these topics:

History of SQL

Dr. E. F. Codd published the paper, "A Relational Model of Data for Large Shared Data Banks", in June 1970 in the Association of Computer Machinery (ACM) journal, *Communications of the ACM*. Codd's model is now accepted as the definitive model for relational database management systems (RDBMS). The language, Structured English Query Language (SEQUEL) was developed by IBM Corporation, Inc., to use Codd's model. SEQUEL later became SQL (still pronounced "sequel"). In 1979, Relational Software, Inc. (now Oracle) introduced the first commercially available implementation of SQL. Today, SQL is accepted as the standard RDBMS language.

SQL Standards

Oracle strives to comply with industry-accepted standards and participates actively in SQL standards committees. Industry-accepted committees are the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO), which is affiliated with the International Electrotechnical Commission (IEC). Both ANSI and the ISO/IEC have accepted SQL as the standard language for relational databases. When a new SQL standard is simultaneously published by these organizations, the names of the standards conform to conventions used by the organization, but the standards are technically identical.

> **See Also:**
>
> Oracle and Standard SQL for a detailed description of Oracle Database conformance to the SQL standard

How SQL Works

The strengths of SQL provide benefits for all types of users, including application programmers, database administrators, managers, and end users. Technically speaking, SQL

is a data sublanguage. The purpose of SQL is to provide an interface to a relational database such as Oracle Database, and all SQL statements are instructions to the database. In this SQL differs from general-purpose programming languages like C and BASIC. Among the features of SQL are the following:

- It processes sets of data as groups rather than as individual units.
- It provides automatic navigation to the data.
- It uses statements that are complex and powerful individually, and that therefore stand alone. Flow-control statements, such as begin-end, if-then-else, loops, and exception condition handling, were initially not part of SQL and the SQL standard, but they can now be found in ISO/IEC 9075-4 - Persistent Stored Modules (SQL/PSM). The PL/SQL extension to Oracle SQL is similar to PSM.

SQL lets you work with data at the logical level. You need to be concerned with the implementation details only when you want to manipulate the data. For example, to retrieve a set of rows from a table, you define a condition used to filter the rows. All rows satisfying the condition are retrieved in a single step and can be passed as a unit to the user, to another SQL statement, or to an application. You need not deal with the rows one by one, nor do you have to worry about how they are physically stored or retrieved. All SQL statements use the **optimizer**, a part of Oracle Database that determines the most efficient means of accessing the specified data. Oracle also provides techniques that you can use to make the optimizer perform its job better.

SQL provides statements for a variety of tasks, including:

- Querying data
- Inserting, updating, and deleting rows in a table
- Creating, replacing, altering, and dropping objects
- Controlling access to the database and its objects
- Guaranteeing database consistency and integrity

SQL unifies all of the preceding tasks in one consistent language.

Common Language for All Relational Databases

All major relational database management systems support SQL, so you can transfer all skills you have gained with SQL from one database to another. In addition, all programs written in SQL are portable. They can often be moved from one database to another with very little modification.

Using Enterprise Manager

Many of the operations you can accomplish using SQL syntax can be done much more easily using Enterprise Manager. For more information, see the Oracle Enterprise Manager documentation set, *Oracle Database 2 Day DBA*, or any of the Oracle Database *2 Day* + books.

Lexical Conventions

The following lexical conventions for issuing SQL statements apply specifically to the Oracle Database implementation of SQL, but are generally acceptable in other SQL implementations.

When you issue a SQL statement, you can include one or more tabs, carriage returns, spaces, or comments anywhere a space occurs within the definition of the statement. Thus, Oracle Database evaluates the following two statements in the same manner:

```
SELECT last_name,salary*12,MONTHS_BETWEEN(SYSDATE,hire_date)
  FROM employees
  WHERE department_id = 30
  ORDER BY last_name;

SELECT last_name,
  salary * 12,
        MONTHS_BETWEEN( SYSDATE, hire_date )
FROM employees
WHERE department_id=30
ORDER BY last_name;
```

Case is insignificant in reserved words, keywords, identifiers, and parameters. However, case is significant in text literals and quoted names. Refer to Text Literals for a syntax description of text literals.

> **Note:**
>
> SQL statements are terminated differently in different programming environments. This documentation set uses the default SQL*Plus character, the semicolon (;).

Tools Support

Oracle provides a number of utilities to facilitate your SQL development process:

- Oracle SQL Developer is a graphical tool that lets you browse, create, edit, and delete (drop) database objects, edit and debug PL/SQL code, run SQL statements and scripts, manipulate and export data, and create and view reports.

  Using SQL Developer, you can connect to any target Oracle Database schema using standard Oracle Database authentication. DBAs can also use SQL Developer to administer and monitor their database, with interfaces for Data Pump, RMAN, and Auditing also included.

  Once connected, you can perform operations on objects in the database. You can also connect to schemas for selected databases, such as MySQL, Microsoft SQL Server, and Amazon Redshift, view metadata and data in these databases, and migrate these databases to Oracle Database.
- Oracle SQL Developer Command Line (SQLcl) is a free command line interface for Oracle Database. It allows you to interactively or batch execute SQL and PL/SQL.

  SQLcl offers integrated Oracle Cloud (OCI) support, client side scripting with JavaScript, custom commands, and updated SQL*Plus commands (INFO vs DESC). Additionally, SQLcl provides native vi or Emacs editing, statement completion, and persistent command recall for a feature-rich experience, all while supporting your previously written SQL*Plus scripts.
- Database Actions delivers your favorite Oracle Database desktop tool's features and experience to your web browser. Delivered as a single-page web application, Database Actions is powered by Oracle REST Data Services (ORDS).

  Database Actions offers a worksheet for running queries and scripts, the ability to manage and browse your data dictionary, a REST development environment for your REST APIs

and AUTOREST enabled objects, an interface for Oracle's JSON Document Store (SODA), a DBA console for managing the database, a data model reporting solution, and access to PerfHub. Database Actions is also available automatically for any Oracle Autonomous Database OCI Service.

- SQL*Plus is an interactive and batch query tool that is installed with every Oracle Database server or client installation. It has a command-line user interface.

> **See Also:**
>
> *SQL*Plus User's Guide and Reference* and *Oracle APEX App Builder User's Guide* for more information on these products

The Oracle Call Interface and Oracle precompilers let you embed standard SQL statements within a procedure programming language.

- The Oracle Call Interface (OCI) lets you embed SQL statements in C programs.
- The Oracle precompilers, Pro*C/C++ and Pro*COBOL, interpret embedded SQL statements and translate them into statements that can be understood by C/C++ and COBOL compilers, respectively.

> **See Also:**
>
> *Oracle C++ Call Interface Developer's Guide*, *Pro*COBOL Developer's Guide*, and *Oracle Call Interface Developer's Guide* for additional information on the embedded SQL statements allowed in each product

Most (but not all) Oracle tools also support all features of Oracle SQL. This reference describes the complete functionality of SQL. If the Oracle tool that you are using does not support this complete functionality, then you can find a discussion of the restrictions in the manual describing the tool, such as *SQL*Plus User's Guide and Reference*.