Share on your Social Media

Oracle SQL Tutorial

Published On: June 28, 2024

Oracle SQL Tutorial

Data can be stored, altered, and retrieved from databases using SQL, a standard query language. With no prior database expertise necessary, this Oracle SQL tutorial will help you learn SQL databases swiftly and efficiently from scratch.

Oracle SQL Tutorial PDF

Introduction to Oracle SQL

SQL is capable of running queries on a database. We can insert, modify, and delete records in SQL and we can also retrieve data from it.

We can create new databases, tables, views, and stored procedures using SQL. It is also used to set permissions on tables, procedures, and views for security purposes.

Why Oracle SQL?

You'll need the following to create a website that displays data from a database:

- A database management system (RDBMS – SQL).

- To employ a scripting language server-side, such as ASP or PHP.
- To obtain the desired data using SQL,
- To style the page using HTML and CSS.

SQL Syntax

Easy-to-understand keywords make up SQL statements.

A table called “Customers” has all of its records returned by the SQL statement that follows:

Choose each record in the Customers table:

*Select * FROM Customers;*

Database and Table

Most frequently, a database has one or more tables. Every table has a name (such as “Customers” or “Orders”) and is made up of records (rows) that hold data.

Example: Customer table

Cust_ID	Cust_Name	City	Country
0021	Hiral	Mumbai	India
0022	Tiron	Berlin	Germany
0023	Amy	Chennai	India
0024	Fahad	London	UK

The above table is named ‘Customer Table’ and it has four records and four columns.

Important Note: SQL keywords do not depend on the case: SELECT and select are the same.

Semicolon: Every SQL statement must terminate with a semicolon in certain database systems. In database systems that allow the execution of several SQL statements in a single request to the server, semicolons are typically used to divide each

C and C++ Tutorial
Published On: August 1, 2024

C and C++ Tutorial C is a high-level, procedural, general-purpose programming language. Whereas C++, a...

ASP DOTNET Tutorial
Published On: July 31, 2024

ASP DOTNET Tutorial Microsoft created the web framework known as ASP.NET. It is employed in...

Artificial Intelligence Tutorial
Published On: July 30, 2024

Artificial Intelligence Tutorial Artificial intelligence (AI) is significant since it enhances many facets of society...

Appium Testing Tutorial
Published On: July 30, 2024

Appium Testing Tutorial Designed to make the UI automation of many app platforms easier, Appium...

SQL statement.

Oracle SQL Interview Questions

SQL Commands

SELECT retrieves information from a database.

UPDATE: modifies database data

DELETE: removes information from a database.

INSERT INTO: This command adds new information to a database.

DATABASE CREATE: establishes a new database

ALTER DATABASE: makes changes to a database.

CREATE TABLE: This command adds a new table.

ALTER TABLE: makes changes to a table

TABLE DELETE: This removes a table.

CREATE INDEX generates a search key for an index.

INDEX DROP: This removes an index.

Creating Tables

In an Oracle database, tables serve as the fundamental unit of data storage. The datatype, like in DATE, can specify the width in advance.

In place of width, specify precision and scale if the columns are of the NUMBER datatype. A row is an arrangement of column data that represents a single record.

Every column in a table can have its own set of rules called as integrity restrictions. A NOT NULL integrity requirement is one example. Due to this restriction, a value must appear in the column for each row.

Syntax

> *create table DEPARTMENTS (*

```
  deptno       number,
  name         varchar2(50) not null,
  location     varchar2(50),
  constraint pk_departments primary key (deptno)
);
```

Referential integrity is the term used to describe the declarative specification of relationships between tables by tables.

By adding a foreign key that references the DEPARTMENTS table in the EMPLOYEES table, we may make a “child” table of the DEPARTMENTS table and observe how this functions.

Example

```
create table EMPLOYEES (
  empno          number,
  name           varchar2(50) not null,
  job            varchar2(50),
  manager        number,
  hiredate       date,
  salary         number(7,2),
  commission     number(7,2),
  deptno         number,
  constraint pk_employees primary key (empno),
  constraint fk_employees_deptno foreign key (deptno)
     references DEPARTMENTS (deptno)
);
```

Since foreign keys need to relate to primary keys, a primary key in the “parent” table must exist to build a “child” table.

Insert Into Statement

Use the INSERT INTO statement to insert new records into a table.

Syntax

> *INSERT INTO table_name (column1, column2, column3, ...)*
>
> *VALUES (value1, value2, value3, ...);*

Updating Tables

The UPDATE statement can be used to modify the current records in a table.

> **Syntax**
>
> *UPDATE table_name*
>
> *SET column1 = value1, column2 = value2, ...*
>
> *WHERE condition;*

Example

UPDATE Customers

SET Cust_Name = 'Anna', City= 'Delhi'

WHERE CustomerID = 0023;

Output: Customer table

Cust_ID	Cust_Name	City	Country
0021	Hiral	Mumbai	India
0022	Tiron	Berlin	Germany
0023	Anna	Delhi	India
0024	Fahad	London	UK

SQL Delete Statement

To delete already-existing entries from a table, use the DELETE statement.

Syntax

> *DELETE FROM table_name WHERE condition;*

Example

DELETE FROM Customers WHERE Cust_Name='Tiron';

Output: Customer table

Cust_ID	Cust_Name	City	Country
0021	Hiral	Mumbai	India
0023	Anna	Delhi	India
0024	Fahad	London	UK

Oracle SQL Syllabus PDF

SQL Aggregate Functions

Aggregate functions can be used to add up data in tables. The null value function (NVL) will be used to enable us to correctly sum columns containing null values, and column aliases will be used to rename columns for readability.

The SQL aggregation functions that are most frequently used are:

- The function **MIN()** yields the lowest value present in the chosen column.
- In the chosen column, the **MAX()** method yields the maximum value.
- **COUNT()** yields the number of rows within a set.
- **SUM()** yields a numerical column's total sum.
- The function **AVG()** yields the mean value of a numerical column.

> **Example**
>
> *select*
>
> *count(*) customer_count,*
>
> *sum(profit) total_profit,*
>
> *sum(tax) total_tax,*
>
> *min(profit + nvl(tax,0)) min_compensation,*
>
> *max(profit + nvl(tax,0)) max_compensation*
>
> *from customers;*

SQL Stored Procedures

A stored procedure is a prepared SQL code that can be saved and used again.

Syntax

CREATE PROCEDURE procedure_name

AS

sql_statement

GO;

Running a stored procedure

EXEC procedure_name;

Example

CREATE PROCEDURE SelectAllCustomers @City nvarchar(30)

AS

*SELECT * FROM Customers WHERE City = @City*

GO;

Execution

EXEC SelectAllCustomers @City = 'London';

SQL Views

A view in SQL is a virtual table created from a SQL statement's result set.

A view has rows and columns, exactly like a table in the real world. A view contains fields that come from one or more actual database tables.

A view can have SQL statements and functions added to it so that the data is displayed as though it is from a single table. The CREATE VIEW statement creates a view.

> **Syntax**
>
> *CREATE VIEW view_name AS*

> *SELECT column1, column2, ...*
>
> *FROM table_name*
>
> *WHERE condition;*

> **Example**
>
> *CREATE VIEW [India Customers] AS*
>
> *SELECT Cust_Name*
>
> *FROM Customers*
>
> *WHERE Country = India;*

SQL Injection

One code injection method that could wipe out your database is SQL injection. One of the most popular methods for web hacking is SQL injection.

The act of inserting harmful code into SQL statements through web page input is known as SQL injection.

SQL in Web Pages

When you ask a user for input, such as their username or user ID, and they respond with a SQL statement that you will unintentionally execute on your database, this is known as SQL injection.

Take a look at the following example, which adds a variable (txtUserId) to a select string to generate a SELECT query. The variable (getRequestString) is retrieved from user input:

Example

txtUserId = getRequestString("UserId");

*txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;*

SQL Hosting

Your web server has to have access to a database system that employs the SQL language if you want your website to be able to store and retrieve data from a database.

Oracle, MySQL, MS Access, and MS SQL Server are the most widely used SQL hosting databases.

SQL Data Types

A column's data type indicates the types of values it can store, including integers, characters, currencies, dates and times, binary, and so forth.

Popular Oracle SQL Data Types

MySQL supports the following three main types of data: string, numeric, and date and time.

String Data Types

CHAR(size): A string with a fixed length that may include special characters, numerals, and letters. The values range from 0 to 255. By default, 1

VARCHAR(size): A variable-length string that is capable of holding special characters, numerals, and letters with values ranging from 0 to 65,535.

BINARY(size): Similar to CHAR(), except that binary byte strings are stored. The size option specifies the column length in bytes. By default, 1.

VARBINARY(size): Similar to VARCHAR(); except, binary byte strings are stored. The maximum column length in bytes is specified using the size argument.

TINYBLOB: Regarding Binary Large Objects (BLOBs). Maximum byte length: 255

TINYTEXT: carries a string that can contain up to 255 characters.

TEXT(size): retains a string up to 65,535 bytes in length.

BLOB(size) for Binary Large Objects, or BLOBs, carries a maximum of 65,535 bytes of data.

MEDIUMTEXT contains a string that can contain up to 16,777,215 characters.

MEDIUMBLOB: Binary Large Objects, or BLOBs, carry 16,777,215 bytes of data.

LONGTEXT stores a string that can include up to 4,294,967,295 characters.

SET(Value1, Value2, Value3,...): An object that is a string and that can have zero or more values selected from a list of feasible values. A SET list can include up to 64 values.

LONGBLOB for Binary Large Objects, or BLOBs, carries 4,294,967,295 bytes of data in its memory.

ENUM(val1, val2, val3,...): A string object with a single, limited value that can be selected from a range of options. An ENUM list can contain up to 65535 values.

Numeric Data Types

BIT(size): Size specifies how many bits there are in each value. (1 to 64)

TINYINT(size): A tiny integer. The signed range is -128 through 127. The unsigned range spans 0 through 255.

BOOL: Nonzero values are regarded as true, and zero as false.

BOOLEAN: Equal to bool.

SMALLINT(size): A little integer. The signed range lies between 32768 and 32767. The range of unsigned numbers is 0 to 65535.

MEDIUM(size): A medium number. The range that is signed is -8388608 to 8388607. The unsigned range is 16777215-2067215.

INT(size): The signed interval is -2147483648–2147483647. The range of unsigned numbers is 0 to 4294967295.

FLOAT(size, d): An integer with floating points. The

size of all the digits is defined. The d option specifies how many digits come after the decimal point.

DOUBLE(size, d): A floating-point number of typical size. The size of all the digits is defined.

DECIMAL (d, size): A precise fixed-point figure. The size of all the digits is defined.

Date and Time Data Types

DATE: A specific date. YYYY-MM-DD is the format. '1000-01-01' to '9999-12-31' is the supported range.

DATETIME (fsp): A combination of the date and time. Style: DD/MM/YYYY hh:mm:ss.

TIMESTAMP(fsp): A date and time. The seconds since the Unix epoch ('1970-01-01 00:00:00' UTC) are used to store TIMESTAMP values. YYYY-MM-DD hh:mm:ss is the format.

TIME (fsp): A moment. The format is hh:mm:ss. The range that is supported is '-838:59:59' to '838:59:59'

YEAR: It is represented in 4 digits. (4-digit format are 0000 and 1901 to 2155)

Create Triggers

Procedures known as triggers are kept in the database and are invoked, or implicitly run, when certain events take place.

Table primary keys can be automatically populated with triggers; an example trigger for this purpose is provided in the trigger examples below.

We'll get a globally unique identifier, or GUID, by using a built-in function.

Example

```
create or replace trigger  DEPARTMENTS_BIU
  before insert or update on DEPARTMENTS
  for each row
```

```
begin
   if inserting and :new.deptno is null then
      :new.deptno := to_number(sys_guid(),
       'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
   end if;
end;
/
create or replace trigger EMPLOYEES_BIU
   before insert or update on EMPLOYEES
   for each row
begin
   if inserting and :new.empno is null then
      :new.empno := to_number(sys_guid(),
        'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');
   end if;
end;
/
```

Querying Oracle SQL Data Dictionary

The Oracle data dictionary provides access to table metadata. The queries that follow demonstrate how to query the tables in the data dictionary.

Example

```
select table_name, tablespace_name, status
from user_tables
where table_Name = 'EMPLOYEES';
select column_id, column_name , data_type
from user_tab_columns
where table_Name = 'EMPLOYEES'
order by column_id;
```

Aggregate Queries

Aggregate functions can be used to add up data in tables. The null value function (NVL) will be used to enable us to correctly sum columns containing null values, and column aliases will be used to rename columns for readability.

Example

```
select
    count(*) employee_count,
     sum(salary) total_salary,
     sum(commission) total_commission,
     min(salary + nvl(commission,0))
min_compensation,
     max(salary + nvl(commission,0))
max_compensation
from employees;
```

Compressing Data in SQL

Table compression minimizes memory consumption in the buffer cache and conserves disk space. Moreover, table compression helps expedite query performance when reading data.

It can also be utilized in online transaction processing (OLTP) systems, it is particularly helpful in online analytical processing (OLAP) systems, which include extensive read-only activities.

- The COMPRESS clause of the CREATE TABLE statement is used to specify table compression.
- Using this clause in an ALTER TABLE command allows you to enable compression for a table that already exists.
- Here, the data that is added or altered once compression is enabled is the only data that is compressed.
- In a similar vein, you may use the ALTER TABLE...

NOCOMPRESS statement to stop table compression for an already compressed table.

- Here, newly added data is placed uncompressed, while all previously compressed data is kept compressed.

Syntax

alter table EMPLOYEES compress for oltp;

alter table DEPARTMENTS compress for oltp;

Dropping Tables

The SQL DROP command can be used to drop tables. When a table is dropped, all of its rows along with its sub-objects, such as its triggers and indexes, are also gone.

You can drop database tables in any order by using the optional cascade constraints clause, which will drop and remove constraints.

> **Syntax**
>
> *drop table departments cascade constraints;*
>
> *drop table employees cascade constraints;*

Un-dropping Tables

This table will go in the recycling bin if the RECYCLEBIN initialization parameter is set to ON, which is the default value in 10g.

> **Syntax**
>
> *select object_name,*
>
> *original_name, type,*
>
> *can_undrop,*
>
> *can_purge*
>
> *from recyclebin;*

Example

flashback table DEPARTMENTS to before drop;

flashback table EMPLOYEES to before drop;

select count(*) departments

from departments;

select count(*) employees

from employees;

Bottom Line

Get hands-on exposure to the topics you have learned in this **Oracle SQL tutorial** by enrolling in our **Oracle SQL training in Chennai.**

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas

Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development
- DevOps

Social Media Links

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed
- Software Suggest
- Sitejabber