Oracle SQL SELECT Statement

SQL Queries, SELECT Statement

Use a SELECT statement or subquery to retrieve data from one or more tables, object tables, views, object views, or materialized views

For example to retrieve all rows from emp table.

SQL> select empno, ename, sal from emp;

Or (if you want to see all the columns values

You can also give * which means all columns)

SQL> select * from emp;

If you want to see only employee names and their salaries then you can type the following statement

SQL> select name, sal from emp;

Filtering Information using Where Conditions

You can filter information using where conditions like suppose you want to see only those employees whose salary is above 5000 then you can type the following query with where condition

SQL>select * from emp where sal > 5000;

To see those employees whose salary is less than 5000 then the query will be

SQL> select * from emp where sal < 5000;

SQL SELECT Statement

The most commonly used SQL command is SELECT statement. SQL SELECT statement is used to query or retrieve data from a table in the database. A query may retrieve information from specified columns or from all of the columns in the table. To create a simple SQL SELECT Statement, you must specify the column(s) name and the table name. The whole query is called SQL SELECT Statement.

Syntax of SQL SELECT Statement:

```
SELECT column_list FROM table-name
[WHERE Clause]
[GROUP BY clause]
[HAVING clause]
[ORDER BY clause];
```

- *table-name* is the name of the table from which the information is retrieved.
- *column_list* includes one or more columns from which data is retrieved.
- The code within the brackets is optional.

database table student_details;

id	first_name	last_name	age	subject	games
100	Rahul	Sharma	10	Science	Cricket
101	Anjali	Bhagwat	12	Maths	Football
102	Stephen	Fleming	09	Science	Cricket
103	Shekar	Gowda	18	Maths	Badminton
104	Priya	Chandra	15	Economics	Chess

NOTE: These database tables are used here for better explanation of SQL commands. In reality, the tables can have different columns and different data.

For example, consider the table student_details. To select the first name of all the students the query would be like:

```
SELECT first_name FROM student_details;
```

SELECT Statements: Syntax

NOTE: SQL commands are not case sensitive. The above SELECT statement can also be written as

"select first_name from students_details;"

You can also retrieve data from more than one column. For example, to select first name and last name of all the students.

```
SELECT first_name, last_name FROM student_details;
```

You can also use clauses like WHERE, GROUP BY, HAVING, ORDER BY with SELECT statement. We will discuss these commands in coming chapters.

NOTE: In a SQL SELECT statement only SELECT and FROM statements are mandatory. Other clauses like WHERE, ORDER BY, GROUP BY, HAVING are optional.

How to use expressions in SQL SELECT Statement?

Expressions combine many arithmetic operators, they can be used in SELECT, WHERE and ORDER BY Clauses of the SQL SELECT Statement.

Here we will explain how to use expressions in the SQL SELECT Statement. About using expressions in WHERE and ORDER BY clause, they will be explained in their respective sections.

The operators are evaluated in a specific order of precedence, when more than one arithmetic operator is used in an expression. The order of evaluation is: parentheses, division, multiplication, addition, and subtraction. The evaluation is performed from the left to the right of the expression.

SELECT Statement Example?

If we want to display the first and last name of an employee combined together, the SQL Select Statement would be like

```
SELECT first_name + ' ' + last_name FROM employee;
```

Output:

first_name + '' + last_name

Rahul Sharma
Anjali Bhagwat
Stephen Fleming
Shekar Gowda
Priya Chandra

You can also provide aliases as below.

```
SELECT first_name + ' ' + last_name AS emp_name FROM employee;
```

Output:

```
emp_name
-------------
Rahul Sharma
Anjali Bhagwat
Stephen Fleming
Shekar Gowda
Priya Chandra
```