

SQL Where Clause

Prerequisites for Query execution

Create the data base:

Create database Dataset;

Use the Data Base:

use Dataset;

Create the following Table:

```
CREATE TABLE EMPLOYEE
```

```
(
  Emp_id    INT,
  FName     VARCHAR(15),
  LName     VARCHAR(15),
  Job_Id     VARCHAR(45),
  Manager    CHAR(4),
  HireDate   DATE,
  Salary     INT,
  Commission INT,
  DEPT_Id    INT)
;
```

Insert the records into the table as given below using the given command:

```
INSERT INTO EMPLOYEE
```

```
VALUES (9369, 'Neha', 'Sharma', 'SOFTWARE ENGINEER', 7902, '2022-12-17', 28000,0,20);
```

Enter following record similarly.

Sample Data:-

Emp_id	FName	LName	Job_Id	Manager	HireDate	Salary	Commission	DEPT_Id
9369	Neha	Sharma	SOFTWARE ENGINEER	7902	2022-12-17	28000	0	20
9499	Nisha	Joshi	SALESMAN	7698	2022-02-20	16000	300	30
9566	Akash	Chaudhary	MANAGER	7839	2022-04-02	35070	0	90
9654	Samiksha	Deshpande	SALESMAN	7698	2022-09-28	12500	1400	30
9782	Lina	Umre	MANAGER	7839	2022-06-09	29400	0	90
9788	Lisha	Dhule	ANALYST	7566	2022-12-09	30000	0	20
9839	Avi	Patil	PRESIDENT	7566	2022-11-17	95000	0	10
9844	Kartik	Dev	SALESMAN	7698	2022-09-08	15000	0	30
9876	Swara	Meshram	SOFTWARE ENGINEER	7788	2022-01-12	30100	0	20
9900	Rahul	Sing	TECHNICAL LEAD	7698	2022-12-03	52950	0	60
9902	Dev	Ladhha	ANALYST	7566	2022-12-03	30000	0	10
9934	Kishan	Mathur	SOFTWARE ENGINEER	7782	2022-01-23	3300	0	20
9591	Swarada	B	SALESMAN	7698	2022-02-22	25000	0	100
9698	Vivek	B	MANAGER	7839	2022-05-01	34200	0	30
9777	Nilay	Ramekar	ANALYST	7839	2022-05-01	50000	200	NULL
9860	Prakash	Dive	ANALYST	7839	2022-06-21	70000	100	50
9861	Pooja	Joshi	ANALYST	7839	2022-07-01	50000	100	50

Where clause

The basic form of the SELECT statement is SELECT-FROM-WHERE block. In a SELECT statement, WHERE clause is optional. Using SELECT without a WHERE clause is useful for browsing data from tables.

In a WHERE clause, you can specify a search condition (logical expression) that has one or more conditions. When the condition (logical expression) evaluates to true the WHERE clause filter unwanted rows from the result.

Here is the syntax:

Syntax:

```
SELECT <column_list>  
FROM <table name>  
WHERE <condition>;
```

Display the emp_id, fname, lname, dept_id of employees whose dept_id=20 :

```
SELECT emp_id, fname,  
lname, dept_id  
FROM employee  
WHERE dept_id=20;
```

Display the emp_id, job_id, salary of employees whose last_name='Deshpande'.

Note : Character strings are enclosed in quotation marks. Character values are case-sensitive for some database.

```
SELECT emp_id, job_id, salary  
FROM employee  
WHERE lname = 'Deshpande';
```

WHERE clause using comparison conditions in SQL

Display the employee_id, fname, lname and salary of employees whose salary is greater than or equal to 4000 :

```
SELECT emp_id, fname, lname, salary  
FROM employee  
WHERE salary >= 4000;
```

WHERE clause using expression in SQL

Display the first_name, last_name, salary and (salary+(salary*commission_pct)) as Net Salary of employees whose Net Salary is in the range 10000 and 15000 and who gets atleast a percentage of commission_pct.

```
SELECT fname,lname,salary,
(salary+(salary*commission)) "Net Salary"
FROM employee
WHERE
(salary+(salary*commission))
BETWEEN 10000 AND 15000
AND commission>0;
```

WHERE clause using BETWEEN condition in SQL

Display the employee_id, first_name, last_name and salary of employees whose salary is greater than or equal to 40000 and less than equal to 60000 where 40000 is the lower limit and 60000 is the upper limit of the salary.

```
SELECT emp_id, fname, lname, salary
FROM employee
WHERE salary BETWEEN 4000 AND 6000;
```

WHERE clause using IN condition in SQL

The IN condition is used to test for values in a list.

Display the employee_id, first_name, last_name, department_id and salary of employees whose department_id 60, 90 or 100.

```
SELECT emp_id, fname, lname,
dept_id, salary
FROM employee
WHERE dept_id IN(60,90,100);
```

WHERE clause using LIKE condition in SQL

The LIKE condition is used to perform wildcard searches of string values. The search condition can contain either numbers or literal characters, _ denotes one character and % denotes zero or many characters.

Display the employee_id, first_name, last_name and salary of employees whose first_name starting with 'S'.

```
SELECT emp_id, fname, lname,
dept_id, salary
FROM employee
WHERE fname LIKE('S%');
```

WHERE clause using NULL condition in SQL

IS NULL operator is used to test for nulls.

Display the employee_id, first_name, last_name and salary of employees whose department_id is null.

```
SELECT emp_id, fname, lname,  
dept_id, salary  
FROM employee  
WHERE dept_id IS NULL;
```

WHERE clause using Logical Conditions in SQL

WHERE clause using the AND operator in SQL

Display the employee_id, first_name, last_name and salary of employees whose first_name starting with 'S' and salary greater than or equal to 40000.

```
SELECT emp_id, fname, lname,  
dept_id, salary  
FROM employee  
WHERE fname LIKE('S%')  
AND salary >= 40000;
```

WHERE clause using the OR operator in SQL

Display the employee_id, first_name, last_name and salary of employees whose first_name starting with 'S' or 'A'.

```
SELECT emp_id, fname, lname,  
dept_id, salary  
FROM employee  
WHERE fname LIKE('S%')  
OR fname LIKE('A%');
```

WHERE clause using the NOT operator in SQL

Display the employee_id, first_name, last_name and salary of employees except the department_id 90, 60 or 100 :

```
SELECT emp_id, fname, lname,  
dept_id, salary  
FROM employee  
WHERE dept_id  
NOT IN (90, 60, 100);
```