

SELECT: WHERE Clause

WHERE clause can be used to filter records. So, we can use WHERE to get only those rows that fulfils a specified criterion.

Query Syntax:

```
SELECT column_1, column_2
FROM database_name
WHERE condition;
```

Table 1: employee					
Note: We'll refer this table for all the queries in this sheet					
employee_id [integer]	first_name [characters]	last_name [characters]	salary [integer]	department [characters]	city [characters]
1	John	Abraham	1000000	Banking	Delhi
2	Michael	Clarke	800000	Insurance	Bangalore
3	Roy	Thomas	700000	Banking	Gujarat
4	Tom	Jose	600000	Insurance	Delhi
5	Jerry	Pinto	650000	Insurance	Bangalore
6	Philip	Mathew	750000	Services	Chandigarh
7	Amir	Khan	650000	Services	Delhi

Example 1: Find an employee whose salary is equal or greater than or equal to 800000.

Query:

```
1 SELECT first_name, last_name
2 FROM employee
3 WHERE salary >= 800000;
```

Result:

```
+-----+-----+
| first_name | last_name |
+-----+-----+
| John      | Abraham   |
| Michael   | Clarke    |
+-----+-----+
2 rows in set (0.01 sec)
```

Example 2: Get employee details from employee table whose salary greater than 600000.

Query:

```
1 SELECT *
2 FROM employee
3 WHERE salary > 600000;
```

Result:

employee_id	first_name	last_name	salary	department	city
1	John	Abraham	1000000	Banking	Delhi
2	Michael	Clarke	800000	Insurance	Bangalore
3	Roy	Thomas	700000	Banking	Gujarat
5	Jerry	Pinto	650000	Insurance	Bangalore
6	Philip	Mathew	750000	Services	Chandigarh
7	Amir	Khan	650000	Services	Delhi

6 rows in set (0.00 sec)

Example 3: Get employee's department from employee table whose salary less than 800000.

Query:

```
1 SELECT department
2 FROM employee
3 WHERE salary < 800000;
```

Result:

department
Banking
Insurance
Insurance
Services
Services

5 rows in set (0.00 sec)

Example 4: Get employee names (first and last) from employee table whose salary is 1000000.

Query:

```
1 SELECT first_name, last_name
2 FROM employee
3 WHERE salary = 1000000;
```

Result:

first_name	last_name
John	Abraham

1 row in set (0.03 sec)

WHERE: AND, OR, NOT, BETWEEN

Example 5: Get employee details from employee table whose city is Delhi "and" salary is equal or greater than 800000.

Query:

```
1 SELECT *
2 FROM employee
3 WHERE city = 'Delhi' AND salary >= 800000;
```

Result:

```

+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | salary | department | city |
+-----+-----+-----+-----+-----+-----+
|          1 | John      | Abraham   | 1000000 | Banking    | Delhi |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Example 6: Get employee details from employee table whose city is delhi "or" salary is equal or greater than 800000.

Query:

```

1 SELECT *
2 FROM employee
3 WHERE city = 'Delhi' OR salary >= 800000;

```

Result:

```

+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | salary | department | city |
+-----+-----+-----+-----+-----+-----+
|          1 | John      | Abraham   | 1000000 | Banking    | Delhi |
|          2 | Michael   | Clarke    | 800000   | Insurance   | Bangalore |
|          4 | Tom       | Jose      | 600000   | Insurance   | Delhi |
|          7 | Amir     | Khan      | 650000   | Services    | Delhi |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

Example 7: Get employee first names from employee table whose salary between 650000 and 800000.

Query:

```

1 SELECT first_name
2 FROM employee
3 WHERE salary BETWEEN 650000 AND
  800000;

```

Result:

```

+-----+
| first_name |
+-----+
| Michael    |
| Roy        |
| Jerry      |
| Philip     |
| Amir      |
+-----+
5 rows in set (0.00 sec)

```

Example 8: Get employee first names from employee table whose salary does not fall between 650000 and 800000.

Query:

```

1 SELECT first_name
2 FROM employee
3 WHERE salary NOT BETWEEN 650000
  AND 800000;

```

Result:

```

+-----+
| first_name |
+-----+
| John       |
| Tom        |
+-----+

```

```
+-----+
2 rows in set (0.00 sec)
```

WHERE: LIKE with wildcards - % and underscore (_)

We can use wildcards to create some really interesting filters. You can create queries like – all whose name start with 'S', names ending with 'P', all people with an 'f' in their name, people with names of exactly 4 characters, etc.

To achieve this we use wildcards to create advanced filters:

Wildcard	Description
%	Matches zero or more characters
_ (underscore)	Matches exactly one character

Example 9: Write SQL Query to find name of employee whose name Start with 'M'.

Query:

```
1 SELECT *
2 FROM employee
3 WHERE first_name LIKE 'M%';
```

Result:

```
+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | salary | department | city |
+-----+-----+-----+-----+-----+-----+
|          2 | Michael   | Clarke   | 800000 | Insurance   | Bangalore |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Example 10: Get all employee details from employee table whose first name is "John".

Query:

```
1 SELECT *
2 FROM employee
3 WHERE first_name = 'John';
```

Result:

```
+-----+-----+-----+-----+-----+-----+
| employee_id | first_name | last_name | salary | department | city |
+-----+-----+-----+-----+-----+-----+
|          1 | John      | Abraham   | 1000000 | Banking    | Delhi |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Example 11: Get employee names from employee table whose salary start with 7.

Query:

```
1 SELECT first_name
2 FROM employee
3 WHERE salary LIKE '7%';
```

Result:

```
+-----+
| first_name |
+-----+
| Roy        |
| Philip     |
+-----+
2 rows in set (0.02 sec)
```

Example 12: Get employee names from employee table whose name end with 'y'.

Query:

```
1 SELECT first_name
2 FROM employee
3 WHERE first_name LIKE '%y';
```

Result:

```
+-----+
| first_name |
+-----+
| Roy        |
| Jerry      |
+-----+
2 rows in set (0.00 sec)
```

Example 13: Get all cities from employee table whose name start with 'G' and end with 't'.

Query:

```
1 SELECT city
2 FROM employee
3 WHERE city LIKE 'G%t';
```

Result:

```
+-----+
| city      |
+-----+
| Gujarat   |
+-----+
1 row in set (0.00 sec)
```

Example 14: Get Department of employee from employee table whose Department name second letter is 'e', fifth letter is 'i' and last letter is 's'.

Query:

```
1 SELECT department
2 FROM employee
3 WHERE department LIKE '_ e _ _ i % s';
```

Result:

```
+-----+
| department |
+-----+
| Services    |
| Services    |
+-----+
2 rows in set (0.00 sec)
```

Example 15: Get names of employees whose name contain exactly three characters.

Query:

```

1  SELECT first_name
2  FROM employee
3  WHERE first_name LIKE '_ _ _';

```

Result:

```

+-----+
| first_name |
+-----+
| Roy        |
| Tom        |
+-----+
2 rows in set (0.00 sec)

```

Example 16: Get employee names whose name contains letter 'e' at any position.

Query:

```

1  SELECT first_name
2  FROM employee
3  WHERE first_name LIKE '%e%';

```

Result:

```

+-----+
| first_name |
+-----+
| Michael    |
| Jerry      |
+-----+
2 rows in set (0.00 sec)

```

TABLE: UPDATE

To update data in the table, we use UPDATE statement. The UPDATE command specifies the row to be changed using the WHERE clause, and the new data using the SET keyword. Note: without the where clause UPDATE command will change all rows.

Query Syntax: To Update all rows

```

UPDATE table_name
SET
    column_1 = value_1,
    column_1 = value_2,
    column_1 = value_3,
    ...
;

```

Query Syntax: To Update specific rows

```

UPDATE table_name
SET
    column_1 = value_1,
    column_1 = value_2,
    column_1 = value_3,
    ...
WHERE condition;

```

Example 17: Update name of employees with "Jones" whose name is "Roy".

Query:

```
1 SELECT employee_id, first_name, last_name
2 FROM employee
3 WHERE first_name = 'Roy';
4
5 UPDATE employee
6 SET
7     first_name = 'Jones'
8 WHERE first_name = 'Roy';
9
10 SELECT employee_id, first_name, last_name
11 FROM employee
12 WHERE first_name = 'Jones';
```

Result:

employee_id	first_name	last_name
3	Roy	Thomas

Query OK, 1 row affected (0.32 sec)
Rows matched: 1 Changed: 1 Warnings: 0

employee_id	first_name	last_name
3	Jones	Thomas

NOTE: You can see that 1 row affected, that means change has been done.

Example 18: Delete record of the employee whose name is "Jerry".

Query:

```
1 DELETE FROM employee
2 WHERE first_name = 'Jerry';
```

Result:

Query OK, 1 row affected (0.09 sec)

NOTE: You can see that 1 row affected, that means we deletion has been performed.

Problem 1	Refer the table structure given below and answer the questions that follows.																
Table Structure																	
<table><tr><th colspan="6">Table 2: student</th></tr><tr><td>student_id [integer]</td><td>first_name [characters]</td><td>last_name [characters]</td><td>age [integer]</td><td>city [characters]</td><td>fees [integer]</td></tr></table>						Table 2: student						student_id [integer]	first_name [characters]	last_name [characters]	age [integer]	city [characters]	fees [integer]
Table 2: student																	
student_id [integer]	first_name [characters]	last_name [characters]	age [integer]	city [characters]	fees [integer]												
Problem 1.1	Write a query for selecting students whose age is not 14.																
Problem 1.2	Select those student details whose fees is either less than 4500 or greater than 6000.																
Problem 1.3	Get student details of those students whose age does not lie between 12 and 20.																
Problem 1.4	Get student details of those students having name started with any vowels ('a', 'e', 'i', 'o', 'u').																
Problem 1.5	Get student details whose first name and city both starts with same letter 'd'.																
Problem 1.6	Get student_id and first_name of those students having their first_name starting with 'g' and the third char is 'r'.																
Problem 1.7	Get first_name of all those students whose first_name ends with 'k' and age is greater than 14 and they reside in Delhi.																

Problem 2	Refer table structure given below.																												
Table Structure																													
<table><tr><th colspan="6">Table 3: student</th></tr><tr><th>student_id [integer]</th><th>first_name [characters]</th><th>last_name [characters]</th><th>age [integer]</th><th>city [characters]</th><th>fees [integer]</th></tr><tr><td>1</td><td>Vikas</td><td>Kumar</td><td>15</td><td>Delhi</td><td>2000</td></tr><tr><td>2</td><td>Neha</td><td>Jain</td><td>14</td><td>Alwar</td><td>4500</td></tr></table>						Table 3: student						student_id [integer]	first_name [characters]	last_name [characters]	age [integer]	city [characters]	fees [integer]	1	Vikas	Kumar	15	Delhi	2000	2	Neha	Jain	14	Alwar	4500
Table 3: student																													
student_id [integer]	first_name [characters]	last_name [characters]	age [integer]	city [characters]	fees [integer]																								
1	Vikas	Kumar	15	Delhi	2000																								
2	Neha	Jain	14	Alwar	4500																								
Problem 2.1	Write a query for following output.																												
<pre>+-----+-----+-----+-----+-----+-----+ student_id first_name last_name age city fees +-----+-----+-----+-----+-----+-----+ 1 Vikas Kumar 15 Delhi 2000 +-----+-----+-----+-----+-----+-----+ 2 Neha Jain 17 Gurugram 4500 +-----+-----+-----+-----+-----+-----+</pre>																													
Problem 2.2	Write a query for deletion of all records where name ends with 'y'.																												