Keyword Operator

Keyword operator are:

LIKE
DISTINCT
IS NULL
IN
BETWEEN AND

Let us consider the table below to understand the keyword operator

Employee table:

emp_id	first_name	last_name	email	hire_date	salary	dept_id
1	kelly	davis	davis@gmail.com	2021-01-22	78000	80
2	tom	taylor	tom@gmail.com	2020-09-22	84200	70
3	mike	whalen	mike@gmail.com	2021-06-30	98200	50
4	andy	lumb	andy@gmail.com	2021-02-27	42200	80
5	anjel	nair	anj@gmail.com	2019-09-26	42200	40
6	ram	kumar	ram@gmail.com	2018-12-26	64200	40
7	rohan	sharma	ro@gmail.com	2018-11-24	84200	20
8	john	king	j0@gmail.com	2021-02-09	124200	20

DISTINCT:

The **SELECT DISTINCT** statement is used to return only distinct (different) values.

SELECT DISTINCT Syntax

SELECT DISTINCT column1, column2, ... **FROM** table_name;

Scenario 1: write a query to display the distinct salary from employee

```
SELECT DISTINCT salary FROM  sql_notes.employee;
```

Output:

salary
78000
84200
98200
42200
64200
124200

Scenario 2: write a query to update the hire date of the employee to ‘2021-02-09’ where employee id is 7.

UPDATE sql_notes.employee SET hire_date = '2021-02-09' WHERE emp_id = 7;

To verify weather it is updated or not you can execute the following query

SELECT * FROM sql_notes.employee;

Output:

Employee table:

emp_id	**first_name**	**last_name**	**email**	**hire_date**	**salary**	**dept_id**
1	kelly	davis	davis@gmail.com	2021-01-22	78000	80
2	tom	taylor	tom@gmail.com	2020-09-22	84200	70
3	mike	whalen	mike@gmail.com	2021-06-30	98200	50
4	andy	lumb	andy@gmail.com	2021-02-27	42200	80
5	anjel	nair	anj@gmail.com	2019-09-26	42200	40
6	ram	kumar	ram@gmail.com	2018-12-26	64200	40
7	rohan	sharma	ro@gmail.com	2021-02-09	84200	20
8	john	king	j0@gmail.com	2021-02-09	124200	20

BETWEEN AND Operator:

The **BETWEEN** command is used to select values within a given range. The values can be numbers, text, or dates.

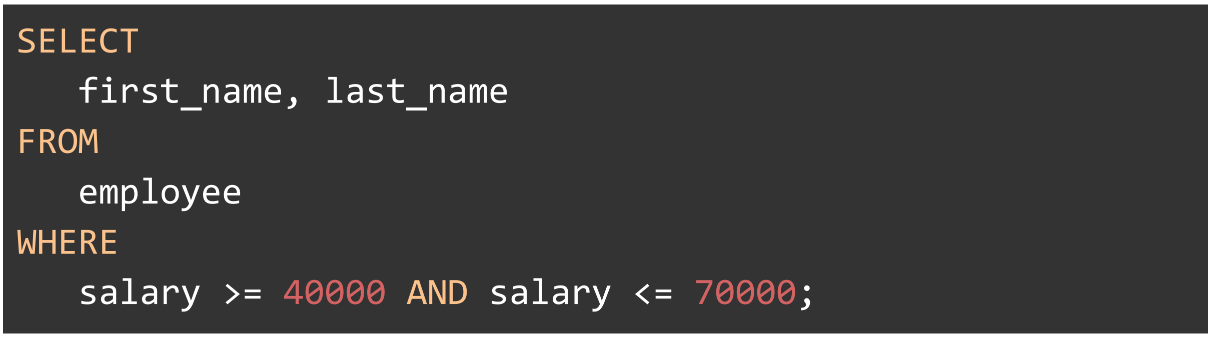
The **BETWEEN** command is inclusive: begin and end values are included.

BETWEEN AND operator is also called as **Range Test Operator**

Syntax:

SELECT column1, column2.. **FROM** table_name **WHERE** column_name **BETWEEN** value 1 **AND** value 2;

Scenario 1: write a query to display the first_name, last_name of all the employees where salary is in the range of 40000 - 70000



```
SELECT
   first_name, last_name
FROM
   employee
WHERE
   salary >= 40000 AND salary <= 70000;
```

Output:

first_name	last_name
andy	lumb
anjel	nair
ram	kumar

Now let’s try to write the same query using BETWEEN operator

```
SELECT
    first_name, last_name
FROM
    employee
WHERE
   salary
BETWEEN 40000 AND 70000;
```

Output:

first_name	**last_name**
andy	lumb
anjel	nair
ram	kumar

Scenario 2: write a query to display the first_name, last_name of all the employees where hire date is in the range of **2018-01-01 to 2020-01-01**

```
SELECT
      first_name, last_name
FROM
       employee
WHERE
       hire_date
BETWEEN
       '2018-01-01' AND '2020-01-01';
```

Output:

first_name	last_name
anjel	nair
ram	kumar

Scenario 3: write a query to display the first_name, last_name of all the employees where first letter of first name is in between A and L

```
SELECT
        first_name, last_name
FROM
        employee
WHERE
        first_name
BETWEEN
        'A' AND 'L';
```

Output:

first_name	last_name
kelly	davis
andy	lumb
anjel	nair
john	king

Scenario 4: write a query to display the first_name, salary of all the employees

where salary is not in the range of 40000 to 70000

```
SELECT
    first_name, salary
FROM
    employee
WHERE
    salary
NOT BETWEEN 40000 AND 70000;
```

Output:

first_name	salary
kelly	78000
tom	84200
mike	98200
rohan	84200
john	124200

IN Operator:

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

Syntax:

```
SELECT column_name(s)
FROM table_name
```

```
WHERE column_name IN (value1, value2, ...);
```

Scenario 4: write a query to display the first_name, salary of all the employees where salary is equal to 78000 or 42200 or 64200

```
SELECT
    first_name, salary
FROM
    employee
WHERE
    salary
IN (78000, 42200, 64200);
```

Output:

first_name	**salary**
kelly	78000
andy	42200
anjel	42200
ram	64200

Scenario 5: write a query to display the first_name, salary of all the employees where salary is not equal to 78000 or 42200 or 64200

```sql
SELECT
     first_name, salary
FROM
     employee
WHERE
     salary
NOT IN (78000, 42200, 64200);
```

Output:

first_name	**salary**
tom	84200
mike	98200
rohan	84200
john	124200

Scenario 6: write a query to display the first_name, salary of all the employees where salary is not equal to 78000 or 42200 or 64200

```sql
SELECT
   first_name, salary
FROM
   employee
WHERE
   salary
NOT IN (78000, 42200, 64200);
```

Output:

first_name	**salary**
tom	84200
mike	98200
rohan	84200
john	124200

IS NULL Operator:

To understand the IS NULL operator let us set the dept_id of the employee where emp_id is greater than 6 by executing the following statement

```sql
UPDATE
    employee
SET
    dept_id = NULL
WHERE
    emp_id > 6;
```

To verify weather the values have been updated with NULL by executing the below query

```sql
SELECT * FROM employee;
```

Output:

emp_id	**first_name**	**last_name**	**email**	**hire_date**	**salary**	**dept_id**

1	kelly	davis	davis@gmail.com	2021-01-22	78000	80
2	tom	taylor	tom@gmail.com	2020-09-22	84200	70
3	mike	whalen	mike@gmail.com	2021-06-30	98200	50
4	andy	lumb	andy@gmail.com	2021-02-27	42200	80
5	anjel	nair	anj@gmail.com	2019-09-26	42200	40
6	ram	kumar	ram@gmail.com	2018-12-26	64200	40
7	rohan	sharma	ro@gmail.com	2021-02-09	84200	NULL
8	john	king	j0@gmail.com	2021-02-09	124200	NULL

Scenario 1: Write a query to display all the employee with dept_id is NULL

```
SELECT * FROM
    employee
WHERE
    dept_id
IS NULL;
```

Output:

emp_id	first_name	last_name	email	hire_date	salary	dept_id
7	rohan	sharma	ro@gmail.com	2021-02-09	84200	NULL
8	john	king	j0@gmail.com	2021-02-09	124200	NULL

Scenario 2: Write a query to display first name and salary of all the employee with dept_id is NOT NULL

```
SELECT
    first_name,salary
FROM
    employee
WHERE
    dept_id
IS NOT NULL;
```

Output:

first_name	**salary**
anjel	42200
ram	64200
mike	98200
tom	84200
kelly	78000
andy	42200