ASSIGNMENT MYSQL AGGREGATE FUNCTIONS with OPERATIONS

Nimra Ashraf

110829

Submitted to:

Mam Sehrish Khan

06/05/2025

USE OF BETWEEN & IN

The keyword **BETWEEN** is used to show the values/contents of the table between a given limit.

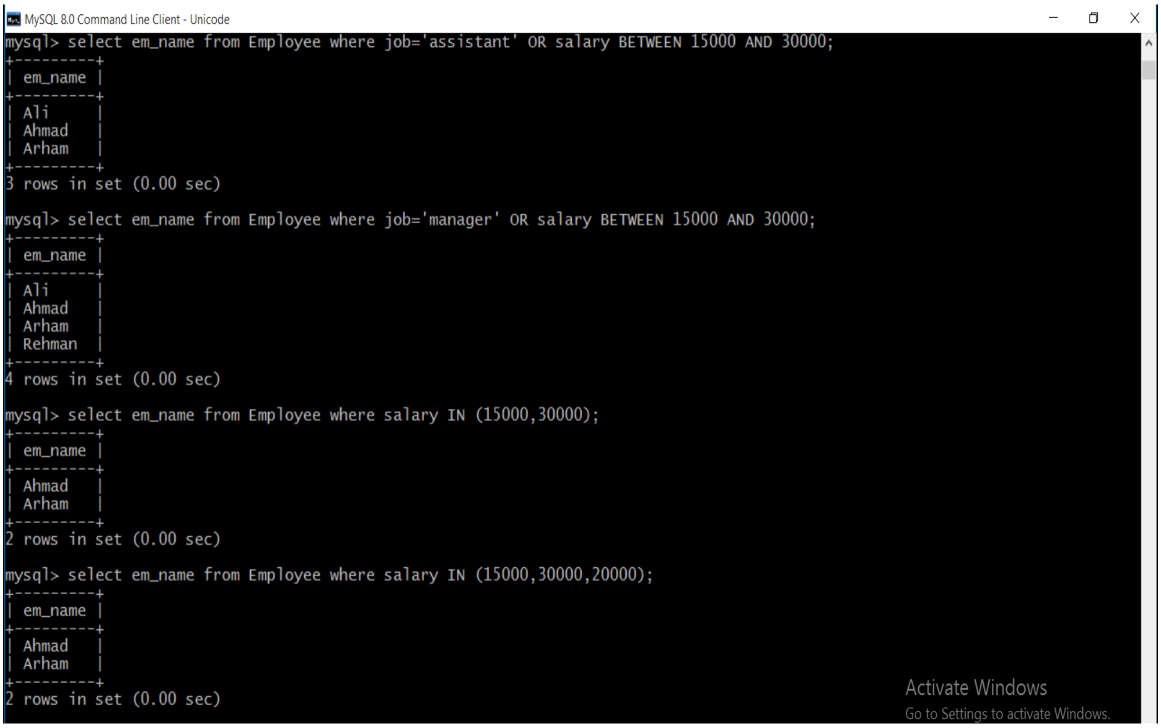
Syntax:

select attribute_name from table_name where attribute='value' OR attribute_value BETWEEN … AND … ;

IN keyword is used to show the contents who have the given values.

Syntax:

select attribute_name from table_name where attribute_value IN (value, value…);


MySQL 8.0 Command Line Client - Unicode
Activate Windows
Go to Settings to activate Windows.

```
mysql> select em_name from Employee where job='assistant' OR salary BETWEEN 15000 AND 30000;
+---------+
| em_name |
+---------+
| Ali     |
| Ahmad   |
| Arham   |
+---------+
3 rows in set (0.00 sec)

mysql> select em_name from Employee where job='manager' OR salary BETWEEN 15000 AND 30000;
+---------+
| em_name |
+---------+
| Ali     |
| Ahmad   |
| Arham   |
| Rehman  |
+---------+
4 rows in set (0.00 sec)

mysql> select em_name from Employee where salary IN (15000,30000);
+---------+
| em_name |
+---------+
| Ahmad   |
| Arham   |
+---------+
2 rows in set (0.00 sec)

mysql> select em_name from Employee where salary IN (15000,30000,20000);
+---------+
| em_name |
+---------+
| Ahmad   |
| Arham   |
+---------+
2 rows in set (0.00 sec)
```

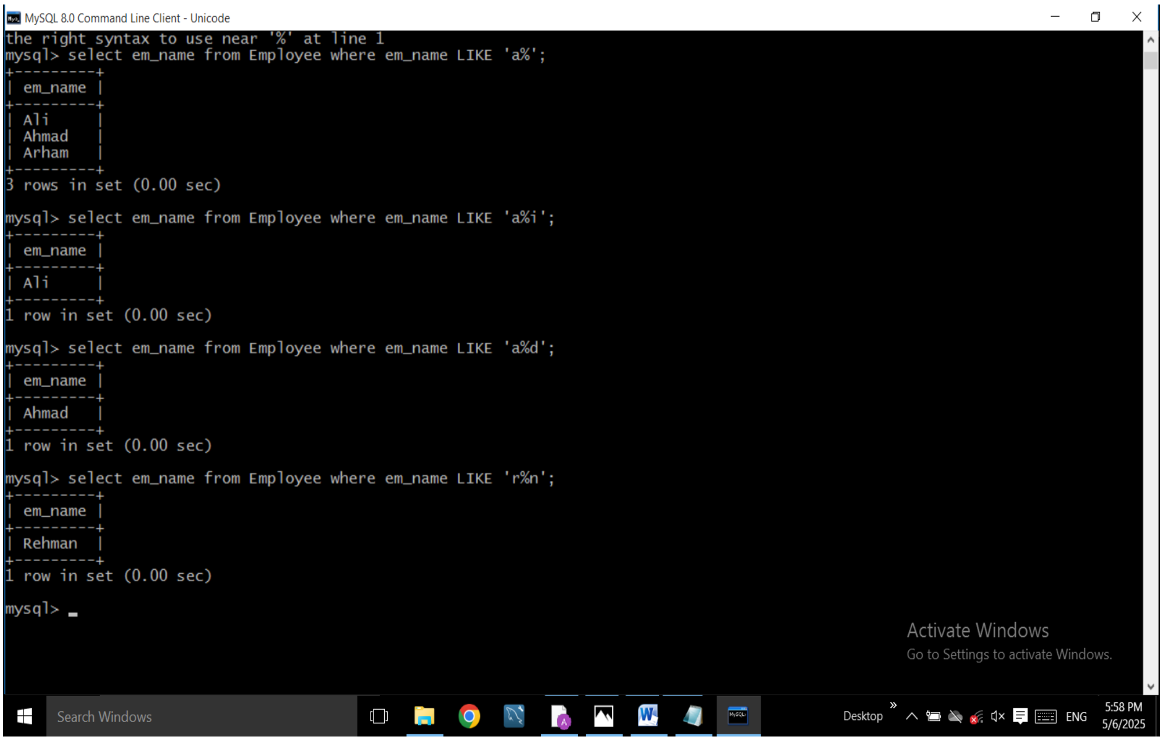
USE OF LIKE KEYWORD

This keyword is used when we want to show the name of a person or an attribute whose name start from the character given.

For example the names of employees start from A or whose name contains character r at 2nd or 3rd place.

Syntaxes:

1. **a%** The 1st character is a.
2. **_%a** Whatever the 1st character is, the last character should be a.
3. **a%_a** 1st and last characters should be a and the center characters whatever are.



MySQL 8.0 Command Line Client - Unicode
Activate Windows
Go to Settings to activate Windows.
Search Windows
Desktop
ENG
5:58 PM
5/6/2025

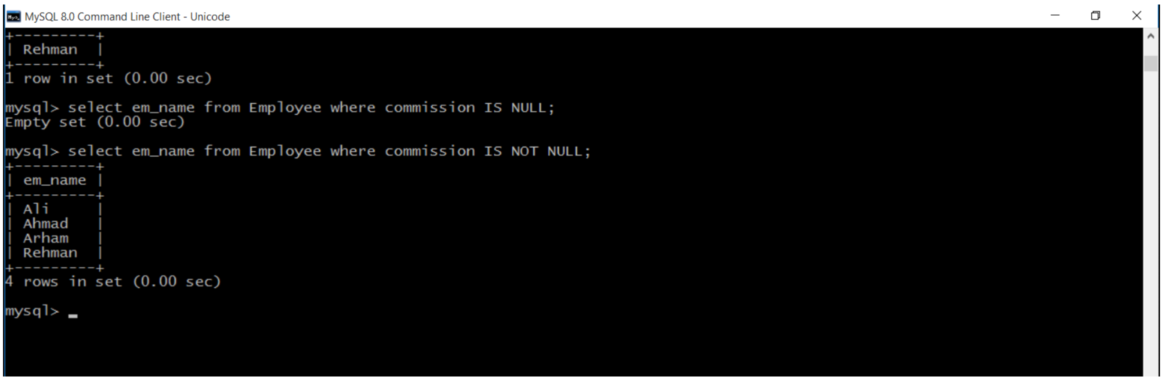
```
the right syntax to use near '%' at line 1
mysql> select em_name from Employee where em_name LIKE 'a%';
+---------+
| em_name |
+---------+
| Ali     |
| Ahmad   |
| Arham   |
+---------+
3 rows in set (0.00 sec)

mysql> select em_name from Employee where em_name LIKE 'a%i';
+---------+
| em_name |
+---------+
| Ali     |
+---------+
1 row in set (0.00 sec)

mysql> select em_name from Employee where em_name LIKE 'a%d';
+---------+
| em_name |
+---------+
| Ahmad   |
+---------+
1 row in set (0.00 sec)

mysql> select em_name from Employee where em_name LIKE 'r%n';
+---------+
| em_name |
+---------+
| Rehman  |
+---------+
1 row in set (0.00 sec)

mysql> _
```

USE OF “IS NULL” AND “IS NOT NULL”

These keywords are used for checking that the values of attributes checked are **NULL** or not.

Syntax Example:

1. **select attribute_name from table where attribute IS NULL**
2. **select attribute_name from table where attribute IS NOT NULL**



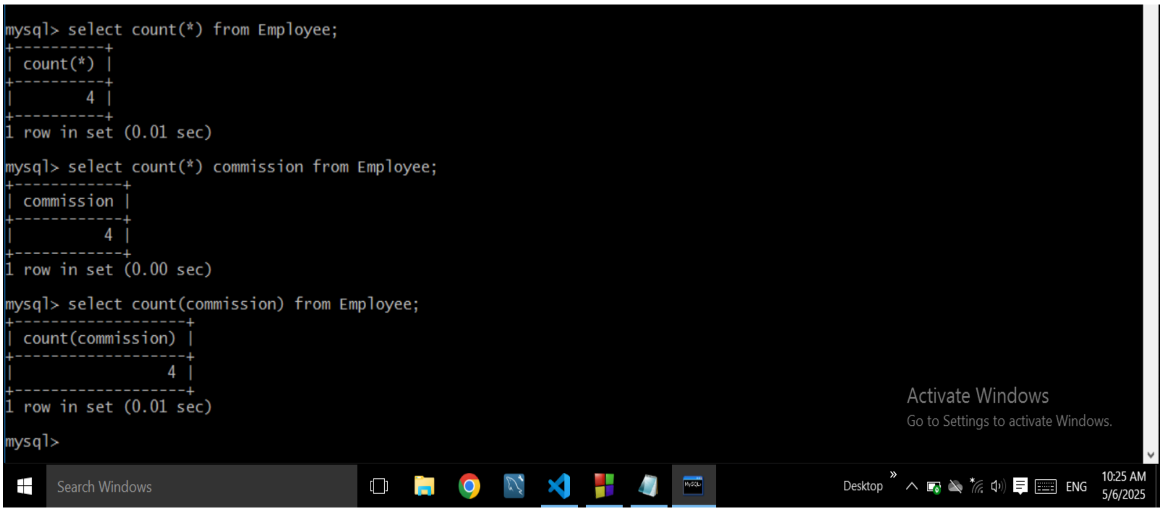
```
+---------+
| Rehman  |
+---------+
1 row in set (0.00 sec)

mysql> select em_name from Employee where commission IS NULL;
Empty set (0.00 sec)

mysql> select em_name from Employee where commission IS NOT NULL;
+---------+
| em_name |
+---------+
| Ali     |
| Ahmad   |
| Arham   |
| Rehman  |
+---------+
4 rows in set (0.00 sec)

mysql> _
```

USE OF count (*) & count (attribute);

Use to find how many values an attribute have.

Syntax Example:

1. **select count (*) attribute_name from table;**
2. **select count (attribute_name) from table;**



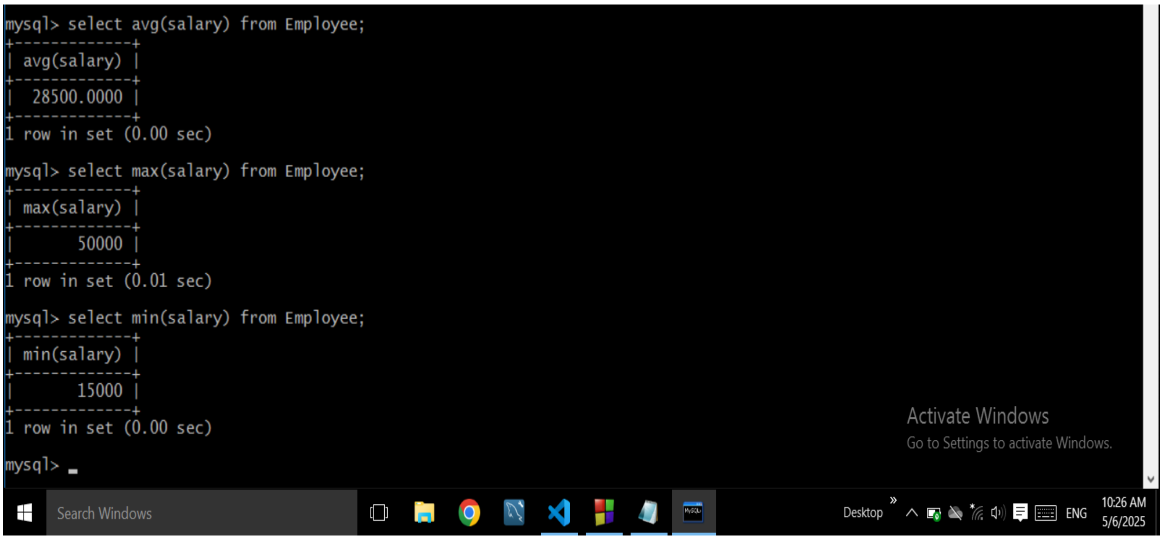
```
mysql> select count(*) from Employee;
+----------+
| count(*) |
+----------+
|        4 |
+----------+
1 row in set (0.01 sec)

mysql> select count(*) commission from Employee;
+------------+
| commission |
+------------+
|          4 |
+------------+
1 row in set (0.00 sec)

mysql> select count(commission) from Employee;
+-------------------+
| count(commission) |
+-------------------+
|                 4 |
+-------------------+
1 row in set (0.01 sec)

mysql>
```

FIND AVG, MAX & MIN

The keywords avg, max and min are used for finding the average, maximum and minimum of a given attribute.

Syntax Example:

1. select avg (attribute) from table;
2. select max (attribute) from table;
3. select min (attribute) from table;



```
mysql> select avg(salary) from Employee;
+-------------+
| avg(salary) |
+-------------+
|  28500.0000 |
+-------------+
1 row in set (0.00 sec)

mysql> select max(salary) from Employee;
+-------------+
| max(salary) |
+-------------+
|       50000 |
+-------------+
1 row in set (0.01 sec)

mysql> select min(salary) from Employee;
+-------------+
| min(salary) |
+-------------+
|       15000 |
+-------------+
1 row in set (0.00 sec)

mysql> _
```

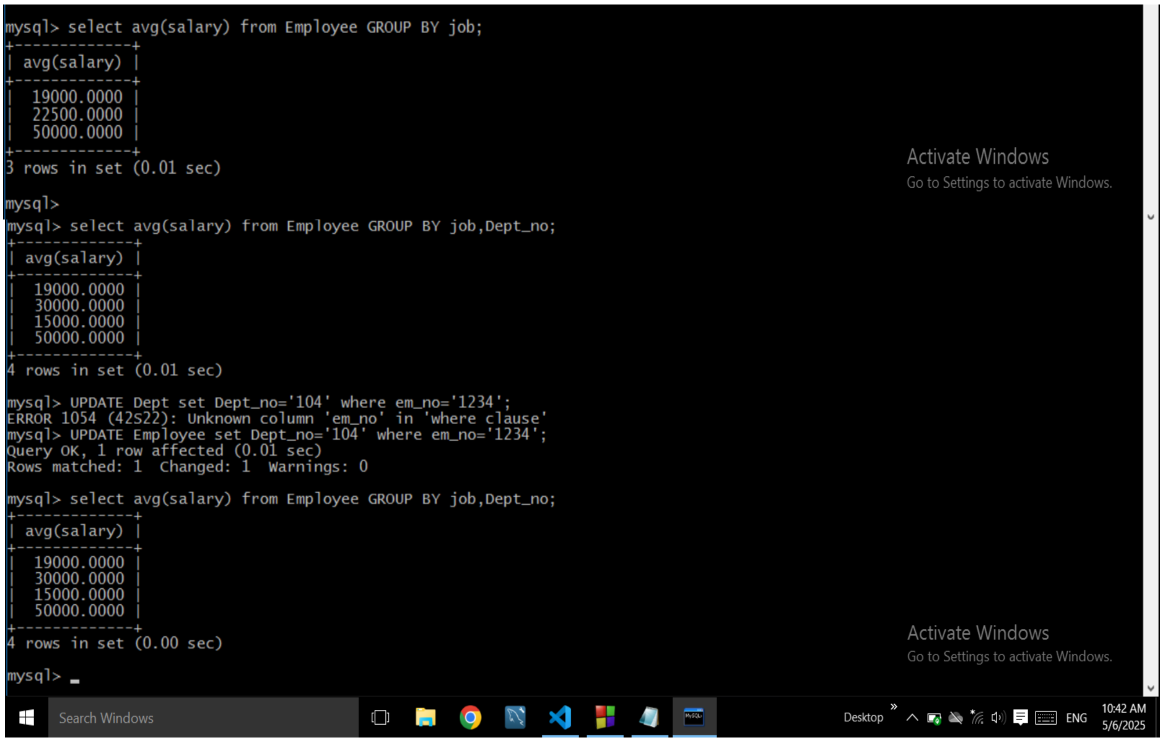
USE GROUP BY

GROUP BY clause is very important used to group rows from a table based on the values of one or more column. It is used with aggregate functions like **AVG, MAX, MIN, SUM and COUNT** to perform calculations on grouped data.

Also we can perform operations on group within the group.

Syntax Example:

1. **select aggreagate_function (attribute) from table GROUP BY attributes;**



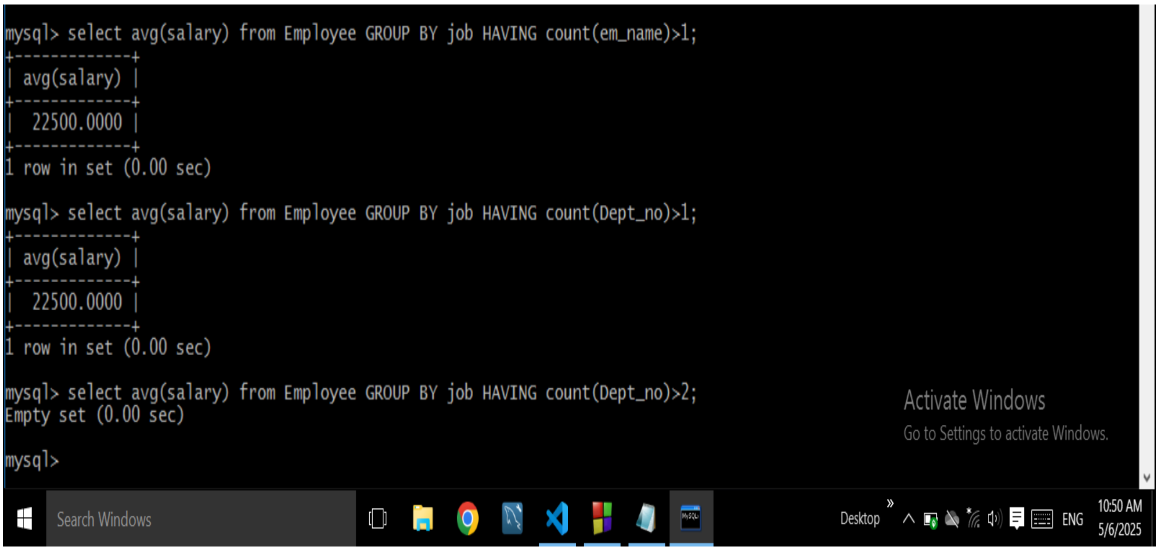
```
mysql> select avg(salary) from Employee GROUP BY job;
+-------------+
| avg(salary) |
+-------------+
|  19000.0000 |
|  22500.0000 |
|  50000.0000 |
+-------------+
3 rows in set (0.01 sec)

mysql>
mysql> select avg(salary) from Employee GROUP BY job,Dept_no;
+-------------+
| avg(salary) |
+-------------+
|  19000.0000 |
|  30000.0000 |
|  15000.0000 |
|  50000.0000 |
+-------------+
4 rows in set (0.01 sec)

mysql> UPDATE Dept set Dept_no='104' where em_no='1234';
ERROR 1054 (42S22): Unknown column 'em_no' in 'where clause'
mysql> UPDATE Employee set Dept_no='104' where em_no='1234';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select avg(salary) from Employee GROUP BY job,Dept_no;
+-------------+
| avg(salary) |
+-------------+
|  19000.0000 |
|  30000.0000 |
|  15000.0000 |
|  50000.0000 |
+-------------+
4 rows in set (0.00 sec)

mysql> _
```

USE OF HAVING

The SQL HAVING clause is used to filter the results of a GROUP BY query based on the result of an aggregate function. It is similar to the WHERE clause but is specifically applied after grouping and aggregation, allowing you to filter on the results of aggregate functions like COUNT, SUM, AVG, and others.

Syntax Example:

1. **select aggreagate_function (attribute) from table GROUP BY attributes HAVING count (attribute) >1;**



```
mysql> select avg(salary) from Employee GROUP BY job HAVING count(em_name)>1;
+-------------+
| avg(salary) |
+-------------+
|  22500.0000 |
+-------------+
1 row in set (0.00 sec)

mysql> select avg(salary) from Employee GROUP BY job HAVING count(Dept_no)>1;
+-------------+
| avg(salary) |
+-------------+
|  22500.0000 |
+-------------+
1 row in set (0.00 sec)

mysql> select avg(salary) from Employee GROUP BY job HAVING count(Dept_no)>2;
Empty set (0.00 sec)

mysql>
```