RECAP

Operator/Clause	Use Case	Features
BETWEEN…AND	Filters records in a range	Includes start and end values
IN	Matches multiple values	Better instead of multiple ORs
LIKE	Search for pattern	Uses % and _
IS NULL	Test for missing values	Checks for null
ORDER BY	Sort Results	Sorts in asc or desc order
LIMIT	Limit the number of rows in a resultset	Controls number of rows we can display

OFFSET:

It specifies how many rows to skip before starting to return the resultset.

SYNTAX:

```
SELECT column_name(s)
From table_name
LIMIT number_of_rows offset offset_value;
```

Ex.

```
mysql> select ename, job,sal
    -> from emp
    -> order by sal desc
    -> limit 5 offset 2;
+-------+----------+---------+
| ename | job      | sal     |
+-------+----------+---------+
| FORD  | ANALYST  | 3000.00 |
| JONES | MANAGER  | 2975.00 |
| BLAKE | MANAGER  | 2850.00 |
```

```
| CLARK | MANAGER   | 2450.00 |
| ALLEN | SALESMAN | 1600.00 |
+-------+----------+---------+
```

```
Select column_name(s)
From table_name
LIMIT offsetvalue,numberOfRows;
```

- Arithmetic Operators Examples

Ex. Add a fixed amount(400) to each employees salary and display the result.

```
mysql> SELECT ENAME,SAL, SAL+400 AS
NEW_SALARY
    -> From emp;
+--------+---------+------------+
| ENAME  | SAL     | NEW_SALARY |
+--------+---------+------------+
| SMITH  |  800.00 |    1200.00 |
| ALLEN  | 1600.00 |    2000.00 |
```

WARD	1250.00	1650.00
JONES	2975.00	3375.00
MARTIN	1250.00	1650.00
BLAKE	2850.00	3250.00
CLARK	2450.00	2850.00
SCOTT	3000.00	3400.00
KING	5000.00	5400.00
TURNER	1500.00	1900.00
ADAMS	1100.00	1500.00
JAMES	950.00	1350.00
FORD	3000.00	3400.00
MILLER	1300.00	1700.00

- Subtract a fixed amount(200) from each employees salary and display.

```
mysql> select ename,sal,sal-200 as 'Updated Salary'
    -> from emp;
```

ename	sal	Updated Salary

SMITH	800.00	600.00
ALLEN	1600.00	1400.00
WARD	1250.00	1050.00
JONES	2975.00	2775.00
MARTIN	1250.00	1050.00
BLAKE	2850.00	2650.00
CLARK	2450.00	2250.00
SCOTT	3000.00	2800.00
KING	5000.00	4800.00
TURNER	1500.00	1300.00
ADAMS	1100.00	900.00
JAMES	950.00	750.00
FORD	3000.00	2800.00
MILLER	1300.00	1100.00

Ex. Calculate a 10% bonus for each employee’s salary and display.

```
mysql> select ename,sal, sal*0.10 AS BONUS
    -> from emp;
+--------+---------+----------+
| ename  | sal     | BONUS    |
+--------+---------+----------+
| SMITH  |  800.00 |  80.0000 |
| ALLEN  | 1600.00 | 160.0000 |
| WARD   | 1250.00 | 125.0000 |
| JONES  | 2975.00 | 297.5000 |
| MARTIN | 1250.00 | 125.0000 |
| BLAKE  | 2850.00 | 285.0000 |
| CLARK  | 2450.00 | 245.0000 |
| SCOTT  | 3000.00 | 300.0000 |
| KING   | 5000.00 | 500.0000 |
| TURNER | 1500.00 | 150.0000 |
| ADAMS  | 1100.00 | 110.0000 |
| JAMES  |  950.00 |  95.0000 |
| FORD   | 3000.00 | 300.0000 |
| MILLER | 1300.00 | 130.0000 |
+--------+---------+----------+
```

Ex. Calculate the employee’s half salary.

```
mysql> SELECT ename, sal, sal/2 AS Half_Salary from emp;
+--------+---------+-------------+
| ename  | sal     | Half_Salary |
+--------+---------+-------------+
| SMITH  |  800.00 |  400.000000 |
| ALLEN  | 1600.00 |  800.000000 |
| WARD   | 1250.00 |  625.000000 |
| JONES  | 2975.00 | 1487.500000 |
| MARTIN | 1250.00 |  625.000000 |
| BLAKE  | 2850.00 | 1425.000000 |
| CLARK  | 2450.00 | 1225.000000 |
| SCOTT  | 3000.00 | 1500.000000 |
| KING   | 5000.00 | 2500.000000 |
| TURNER | 1500.00 |  750.000000 |
| ADAMS  | 1100.00 |  550.000000 |
| JAMES  |  950.00 |  475.000000 |
| FORD   | 3000.00 | 1500.000000 |
| MILLER | 1300.00 |  650.000000 |
+--------+---------+-------------+
14 rows in set (0.00 sec)
```

TASK: Add 100 rupees to salary, subtract 50, multiply by 2 and divide it by 3.

TASK: Show employees whose updated salary(after adding 400) is greater than 2000.

Distinct Keyword

- It is used to ensure that duplicate rows are removed from our resultset.
- It returns only the unique rows.

SYNTAX:

```
select DISTINCT column1, column2
From table_name;
```

EX. Finding the unique job roles in emp table.

```
mysql> SELECT DISTINCT JOB
    -> From emp;
+-----------+
| JOB       |
+-----------+
| CLERK     |
| SALESMAN  |
| MANAGER   |
| ANALYST   |
| PRESIDENT |
+-----------+
```

Ex.To get distinct department numbers

```
SELECT DISTINCT DEPTNO
From emp;
```

Group By Clause

It is used to group rows that share the same values in specified columns into summary rows.

It is commonly used with aggregate functions like COUNT(), SUM(), AVG(), MIN(), MAX().

SYNTAX:

```
Select column1, aggregate_function(column2)
From table_name
Group by column;
```

IMPs:

Group by comes after the where clause and before the order by clause(if using it).

Ex. Total employees but with department.

```
mysql> SELECT deptno,count(*) as
total_employees
    -> From emp
    -> Group by deptno;
+--------+-----------------+
| deptno | total_employees |
+--------+-----------------+
|     20 |               5 |
|     30 |               6 |
|     10 |               3 |
+--------+-----------------+
```

Aggregate Functions:

- It is used to perform calculations on set of values
- It returns a summarized result.

Function	Description
COUNT()	Count the number of rows
SUM()	Returns the total sum of numeric values in a column
MIN()	Finds the smallest value in a column
MAX()	Finds the largest value in the column
AVG()	Calculates the average of numeric values

1. COUNT()

- Counts the number of rows in a specified column

Ex. Select COUNT(*) AS total_employees

From emp;

```
mysql> Select COUNT(*) AS total_employees
    -> From emp;
+-----------------+
```

```
| total_employees |
+-----------------+
|              14 |
+-----------------+
```

Ex. Count employees in each department.

```
Select deptno, count(*) as Total_employees
From emp
Group by deptno;
mysql> Select deptno, count(*) as Total_employees
    -> From emp
    -> Group by deptno;
+--------+-----------------+
| deptno | Total_employees |
+--------+-----------------+
|     20 |               5 |
|     30 |               6 |
|     10 |               3 |
+--------+-----------------+
```

Ex. Count distinct job roles.

```
Select count(DISTINCT JOB) As Unique_jobs from emp;
```

```
mysql> Select count(DISTINCT JOB) As Unique_jobs from emp;
+-------------+
| Unique_jobs |
+-------------+
|           5 |
+-------------+
```

2.SUM()

- It calculates the total sum of numeric values in a column;

Ex. To get the total salary of employees.

```
Select SUM(sal) AS total_salary
From emp;
```

```
mysql> Select SUM(sal) AS total_salary
    -> From emp;
+--------------+
| total_salary |
+--------------+
```

```
|     29025.00 |
+--------------+
```

Ex. Total salary by department:

```
SELECT DEPTNO, SUM(sal) As Total_salary
From emp
Group by deptno;
mysql> SELECT DEPTNO, SUM(sal) As Total_salary
    -> From emp
    -> Group by deptno;
+--------+--------------+
| DEPTNO | Total_salary |
+--------+--------------+
|     20 |     10875.00 |
|     30 |      9400.00 |
|     10 |      8750.00 |
+--------+--------------+
```

Groups employees by dept and calculates total salary paid in each department.

3. Avg()

- To calculate the average of numeric values in a column.

  Ex. To get the average salary of employees.

```
Select avg(sal) As AVG_salary
From emp;
mysql> Select avg(sal) As AVG_salary
    -> From emp;
+-------------+
| AVG_salary  |
+-------------+
| 2073.214286 |
+-------------+
```

Ex. To get average salary by job.

```
SELECT job, avg(sal) as avg_salary
From emp
Group by job;
mysql> SELECT job, avg(sal) as avg_salary
    -> From emp
    -> Group by job;
+-----------+-------------+
| job       | avg_salary  |
+-----------+-------------+
| CLERK     | 1037.500000 |
| SALESMAN  | 1400.000000 |
```

```
| MANAGER   | 2758.333333 |
| ANALYST   | 3000.000000 |
| PRESIDENT | 5000.000000 |
+-----------+-------------+
```

4.Min() & Max()

MIN() -> Finds the smallest value in a column
MAX() -> Finds the largest value in a column

Ex. To get the minimum and maximum salary.
SELECT MIN(Sal) as Minimum_salary, MAX(SAL) as Maximum_salary from emp;

Ex. To get the minimum and maximum salary by department.

```
SELECT deptno, min(sal) as min_salary, max(sal) as max_salary
from emp
group by deptno;
```

```
mysql> SELECT deptno, min(sal) as min_salary, max(sal) as max_salary
    -> from emp
    -> group by deptno;
```

deptno	min_salary	max_salary
20	800.00	3000.00
30	950.00	2850.00
10	1300.00	5000.00