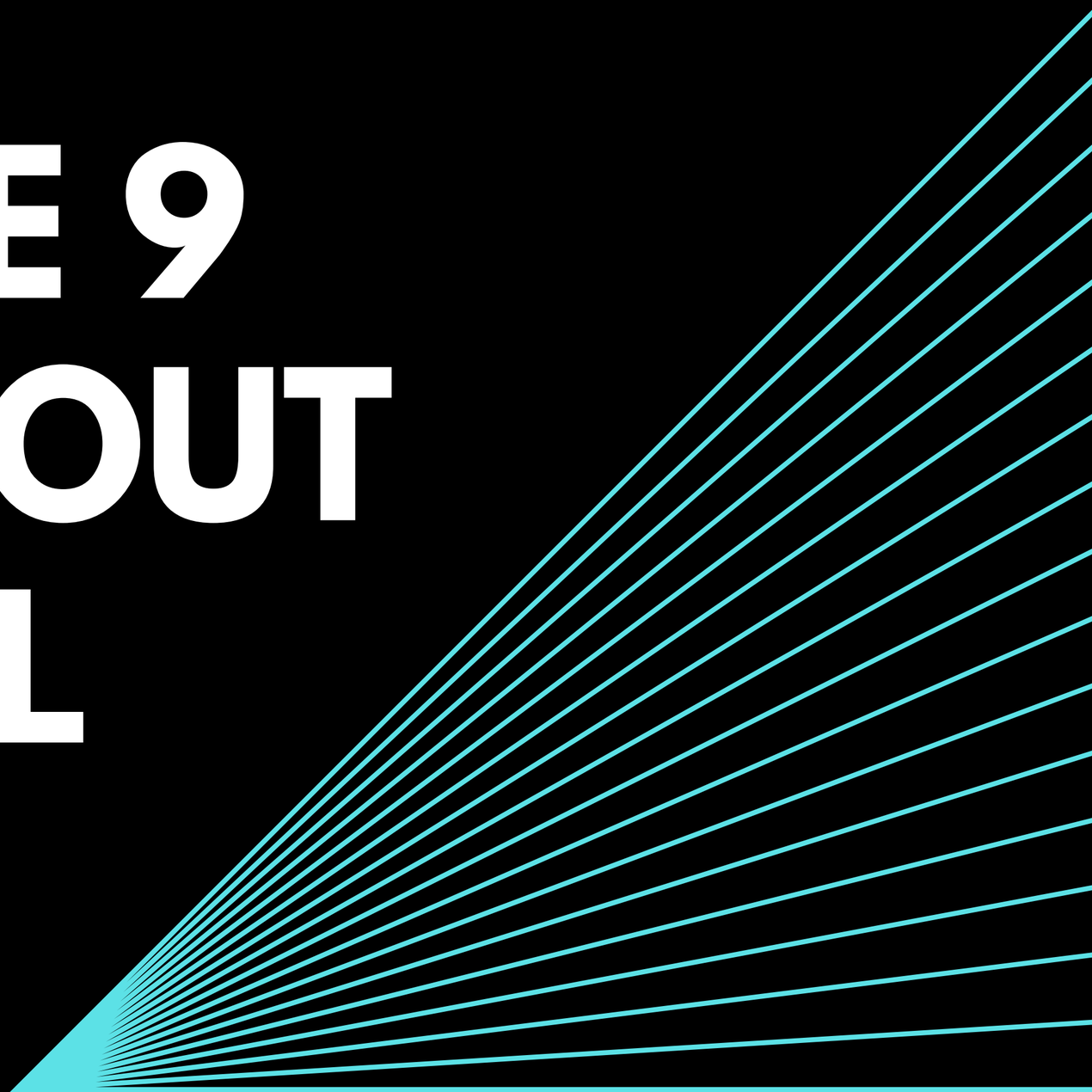
LECTURE 9
MORE ABOUT MYSQL



INITIAL TABLE USED FOR THE PRESENTATION

EMPLOYEE TABLE

E_ID	First_Name	Last_Name	salary	Curr_Loc	Department
1	Riya	Sharma	35000	Pune	Tech
2	Arnav	Shokeen	55000	Mumbai	Marketing
3	Arav	Jaiswal	45000	Delhi	HR
4	Mohini	Kapoor	65000	Banglore	Tech
5	Ram	Singhal	65000	Delhi	HR
6	Ganesh	Singh	35000	Punjab	Tech
7	Kiran	Khanna	45000	Pune	Marketing
8	Karan	Bansal	65000	Mumbai	Tech
9	Ritu	Khan	35000	Punjab	HR

9 rows in set (0.00 sec)

PROJECT TABLE

E_ID	P_ID	Project_Name	Pro_Loc
2	P1	IOT	Mumbai
1	P2	Big Data	Banglore
4	P3	Wind_Power	Punjab
6	P4	Android	Delhi
3	P5	Web	Mumbai
5	P6	Retail	Pune

6 rows in set (0.00 sec)

AGREEGATE FUNCTION

It includes: MAX, MIN, COUNT, AVERAGE, SUM

MAX FUNCTION

GET MAXIMUM SALARY

```
mysql> select MAX(salary) from employee;
+-------------+
| MAX(salary) |
+-------------+
|       65000 |
+-------------+
1 row in set (0.00 sec)
```

GET MORE INFORMATION

```
mysql> select * from employee
    -> where salary = (select MAX(salary)from employee);
+------+------------+-----------+--------+----------+------------+
| E_ID | First_Name | Last_Name | salary | Curr_Loc | Department |
+------+------------+-----------+--------+----------+------------+
|    4 | Mohini     | Kapoor    |  65000 | Banglore | Tech       |
|    5 | Ram        | Singhal   |  65000 | Delhi    | HR         |
|    8 | Karan      | Bansal    |  65000 | Mumbai   | Tech       |
+------+------------+-----------+--------+----------+------------+
3 rows in set (0.00 sec)
```

Nested Query!

It will get the maximum salary from the inner query and filter the whole data according to the maximum salary found.

Since the maximum salary was found out from the first query, hence we have to write the whole query as an inner query.

salary = max(salary) WON'T WORK

MIN FUNCTION

Very smiliar to max salary calculation

```
mysql> select MIN(salary) from employee;
+-------------+
| MIN(salary) |
+-------------+
|       35000 |
+-------------+
1 row in set (0.00 sec)

mysql> select * from employee
    -> where salary = (select MIN(salary)from employee);
+------+------------+-----------+--------+----------+------------+
| E_ID | First_Name | Last_Name | salary | Curr_Loc | Department |
+------+------------+-----------+--------+----------+------------+
|    1 | Riya       | Sharma    |  35000 | Pune     | Tech       |
|    6 | Ganesh     | Singh     |  35000 | Punjab   | Tech       |
|    9 | Ritu       | Khan      |  35000 | Punjab   | HR         |
+------+------------+-----------+--------+----------+------------+
3 rows in set (0.00 sec)
```

AVERAGE FUNCTION

Average = (Sum of all) / (Total no.)

```
mysql> select AVG(salary) from employee;
+-------------+
| AVG(salary) |
+-------------+
|  49444.4444 |
+-------------+
1 row in set (0.00 sec)
```

GET SECOND MAXIMUM SALARY

```
mysql> select * from employee
    -> where salary = (select max(salary) from employee where salary != (select max(salary) from employee));
+------+------------+-----------+--------+----------+------------+
| E_ID | First_Name | Last_Name | salary | Curr_Loc | Department |
+------+------------+-----------+--------+----------+------------+
|    2 | Arnav      | Shokeen   |  55000 | Mumbai   | Marketing  |
+------+------------+-----------+--------+----------+------------+
1 row in set (0.00 sec)
```

COUNT FUNCTION

Count the no of records in a table

```
mysql> select count(*) from employee;
+----------+
| count(*) |
+----------+
|        9 |
+----------+
1 row in set (0.00 sec)
```

Counting distinct values in a particular column.

```
mysql> select COUNT(Distinct Salary) from employee;
+------------------------+
| COUNT(Distinct Salary) |
+------------------------+
|                      4 |
+------------------------+
1 row in set (0.00 sec)
```

GROUP BY

Counting People in each department

```
mysql> select Department,count(*)
    -> from employee
    -> group by department;
+------------+----------+
| Department | count(*) |
+------------+----------+
| Tech       |        4 |
| Marketing  |        2 |
| HR         |        3 |
+------------+----------+
3 rows in set (0.00 sec)
```

HAVING

It is basically condition on GROUP BY

Getting the names of the departments where more than 2 people are working

```
mysql> select Department,count(*)
    -> from employee
    -> group by department
    -> having count(*)>2;
+------------+----------+
| Department | count(*) |
+------------+----------+
| Tech       |        4 |
| HR         |        3 |
+------------+----------+
2 rows in set (0.00 sec)
```

Getting details of each person who has the maximum salary in each department.

```
mysql> select *
    -> from employee
    -> where salary in (Select Max(salary) from employee group by department);
+------+------------+-----------+--------+----------+------------+
| E_ID | First_Name | Last_Name | salary | Curr_Loc | Department |
+------+------------+-----------+--------+----------+------------+
|    2 | Arnav      | Shokeen   |  55000 | Mumbai   | Marketing  |
|    4 | Mohini     | Kapoor    |  65000 | Banglore | Tech       |
|    5 | Ram        | Singhal   |  65000 | Delhi    | HR         |
|    8 | Karan      | Bansal    |  65000 | Mumbai   | Tech       |
+------+------------+-----------+--------+----------+------------+
4 rows in set (0.00 sec)
```

Knowing which department is working on which project.

```
mysql> select Department, Project_Name
    -> from employee t1, project t2
    -> where t1.E_ID = t2.E_ID;
+------------+--------------+
| Department | Project_Name |
+------------+--------------+
| Marketing  | IOT          |
| Tech       | Big Data     |
| Tech       | Wind_Power   |
| Tech       | Android      |
| HR         | Web          |
| HR         | Retail       |
+------------+--------------+
6 rows in set (0.00 sec)
```

Getting data from two separate tables using the primary key of one table which acts as foreign key of the other table.

END OF LECTURE 9