DATABASE MANAGEMENT SYSTEMS

TICT2233

Aggregate Functions, Filtering and Limiting, Grouping

Create Employees table

- These are the fields include in Employees table

```
create table Employees(
Employee_ID decimal(6,0) not null,
Job_ID varchar(10) not null,
Salary decimal(8,2),
Department_ID decimal(4,0));
```

Aggregate Functions – MIN, MAX, COUNT, SUM, AVG

- MIN

Find the minimum salary in the Employees table

```
SELECT MIN(Salary) FROM Employee;
```

- MAX

Find the maximum salary in the Employees table

```
SELECT MAX(Salary) as FROM Employee;
```

- SUM

calculate the total salary of employees

```
SELECT SUM(Salary) as Total_Salay FROM Employee;
```

- AVG

calculate the average salary of employees

```
SELECT AVG(Salary) as Average_Salay FROM Employee;
```

- COUNT

Count the number of employees in the Employees table

```
SELECT COUNT(*) FROM Employees;
SELECT COUNT(Employee_ID) as No_of_Employees from Employees;
```

Filtering and Limiting

- LIMIT

Limits the number of rows returned by the query

Returns the first 5 rows

```
SELECT * FROM Employees LIMIT 5;
```

Using LIMIT with OFFSET

skips the first 10 rows, then return the next 5

```
SELECT * FROM Employees LIMIT 5 OFFSET 10;
```

- IN

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

```
SELECT * FROM Employees
WHERE Job_ID=“Developer”AND Job_ID=“Intern”

SELECT * FROM Employees
WHERE Job_ID  IN (“Developer”, “Intern”);

SELECT * FROM Employees
WHERE Job_ID  NOT IN (“Developer”, “Intern”);
```

- BETWEEN

The BETWEEN operator selects values within a given range.

Write a query to display all the data where the salary range between 40000 - 60000

```
SELECT * FROM Employees WHERE salary BETWEEN 40000 AND 60000;
```

Grouping

- GROUP BY

Group employees by their job roles

```
SELECT Job_ID as JobTitle
FROM Employees
GROUP BY Job_ID;
```

Group employees by their job roles and count the number of employees in each role.

```
SELECT Job_ID , COUNT(*) AS No_of_Jobs
FROM Employees GROUP BY Job_ID ;
```

Group employees by job roles and department.

```
SELECT Job_ID, Department_ID, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY Job_ID, Department_ID;
```

- Filtering Group with HAVING

Include groups with more than 5 employee.

```
SELECT Job_ID, COUNT(*) AS JobCount
FROM Employees
GROUP BY Job_ID
HAVING COUNT(*) > 5;
```

Find the minimum, maximum, and average salary for each job role.

```
 SELECT Job_ID,
MIN(Salary) AS MinSalary,
MAX(Salary) AS MaxSalary,
AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Job_ID ;
```

GROUP BY with ORDERING

Sort grouped results by the count of employees in descending order.

```
SELECT Job_ID, COUNT(*) AS JobCount
FROM Employees
GROUP BY Job_ID
ORDER BY JobCount DESC;
```

Exercise

Q1. Find the average salary of all employees in the company.

Q2. Find the highest and lowest salary in the company.

Q3. Find the minimum, maximum, and salary for each job title (Job_ID).

Q4. Find the job titles (Job_ID) where the number of employees is greater than 1 and the average salary is greater than 60,000.

Q5. List the top 3 job titles with the highest average salary, but only include job titles with an average salary greater than 70,000.

Q1. Find the average salary of all employees in the company.

```
SELECT
    AVG(Salary) AS AvgSalary
FROM Employees;
```

Q2. Find the highest and lowest salary in the company.

```
SELECT
    MAX(Salary) AS HighestSalary,
    MIN(Salary) AS LowestSalary
FROM Employees;
```

Q3. Find the minimum, maximum, and salary for each job title (Job_ID).

```
SELECT
    Job_ID,
    MIN(Salary) AS MinSalary,
    MAX(Salary) AS MaxSalary,
    AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Job_ID;
```

Q4. Find the job titles (Job_ID) where the number of employees is greater than 1 and the average salary is greater than 60,000.

```
SELECT
    Job_ID,
    COUNT(*) AS EmployeeCount,
    AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Job_ID
HAVING COUNT(*) > 1 AND AVG(Salary) > 60000;
```

Q5. List the top 3 job titles with the highest average salary, but only include job titles with an average salary greater than 70,000.

```
SELECT
    Job_ID,
    AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Job_ID
HAVING AVG(Salary) > 70000
ORDER BY AvgSalary DESC
LIMIT 3;
```