

PES UNIVERSITY

DBMS LAB

WEEK 6

SQL – Aggregate functions.

Name : Suhan B Revankar

SRN : PES2UG19CS412

Section : G Section

PROBLEM STATEMENT:

Write the SQL query using aggregate functions for the following.

1. Show the resulting salaries if every employee working on the 'ProductX' project is given 10% raise.
2. Find the sum of the salaries of all employees of the 'Research' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
3. Count the number of distinct salary values in the database.
4. Retrieve the names of all employees who have two or more dependents.
5. For each department, retrieve the department number, the number of employees in the department, and their average salary.
6. Retrieve the names of employees who make at least \$10,000 more than the employee who is paid the least in the company.
7. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.
8. Count the total number of employees whose salaries exceed \$40,000 in each department.

PREREQUISITE:

Create the Company database containing all the required schema:

```
C:\Program Files\PostgreSQL\13\bin>psql -U postgres -f "C:\Users\91944\Documents\FIFTH SEMESTER\DBMS\companyddl.sql"
Password for user postgres:
DROP DATABASE
CREATE DATABASE
You are now connected to database "company" as user "postgres".
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
ALTER TABLE
```

Insert Values into all the tables:

```
C:\Program Files\PostgreSQL\13\bin>psql -U postgres -f "C:\Users\91944\Documents\FIFTH SEMESTER\DBMS\company_insert.sql"
Password for user postgres:
You are now connected to database "company" as user "postgres".
ALTER TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

```
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
psql:C:\Users\91944\Documents\FIFTH SEMESTER\DBMS\company_insert.sql:46: ERROR:  null value in column "hours" of relation "works_on" violates not-null constraint
DETAIL:  Failing row contains (888665555, 20, null).
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
ALTER TABLE
```

EXECUTION OF THE QUERIES:

1. Show the resulting salaries if every employee working on the 'ProductX' project is given a 10% raise.

```
postgres=# \c company;
You are now connected to database "company" as user "postgres".
company=# SELECT FNAME, MINIT, LNAME, 1.1*SALARY AS "RESULTING SALARY"
company-# FROM EMPLOYEE, WORKS_ON, PROJECT
company-# WHERE SSN = ESSN AND PNO = PNUMBER AND PNAME = 'ProductX';
 fname | minit | lname | RESULTING SALARY
-----+-----+-----+-----
 John  | B     | Smith |          33000.000
 Joyce | A     | English |          27500.000
(2 rows)
```

2. Find the sum of the salaries of all employees of the 'Research' department, as well as the maximum salary, the minimum salary, and the average salary in this department.

```
company=# SELECT SUM (SALARY), MAX (SALARY), MIN (SALARY), AVG (SALARY)
company-# FROM EMPLOYEE, DEPARTMENT
company-# WHERE DNO = DNUMBER AND DNAME = 'Research';
      sum       |      max       |      min       |      avg
-----+-----+-----+-----
 133000.00 |  40000.00 |  25000.00 | 33250.000000000000
(1 row)
```

3. Count the number of distinct salary values in the database.

```
company=# SELECT COUNT(DISTINCT SALARY) FROM EMPLOYEE;
count
-----
      6
(1 row)
```

4. Retrieve the names of all employees who have two or more dependents.

```
company=# SELECT FNAME, MINIT, LNAME
company-# FROM EMPLOYEE
company-# WHERE (SELECT COUNT (*)
company-# FROM DEPENDENT
company-# WHERE SSN = ESSN) >= 2;
  fname      | minit | lname
-----+-----+-----
  John       | B     | Smith
Franklin    | T     | Wong
(2 rows)
```

5. For each department, retrieve the department number, the number of employees in the department, and their average salary.

```
company=# SELECT DNO, COUNT (*), AVG (SALARY)
company-# FROM EMPLOYEE
company-# GROUP BY DNO;
 dno | count |          avg
-----+-----+-----
   5 |     4 | 33250.000000000000
   4 |     3 | 31000.000000000000
   1 |     1 | 55000.000000000000
(3 rows)
```

6. Retrieve the names of employees who make at least \$10,000 more than the employee who is paid the least in the company

```
company=# SELECT FNAME, MINIT, LNAME
company-# FROM EMPLOYEE
company-# WHERE SALARY >= 10000 + (SELECT MIN(SALARY) FROM EMPLOYEE);
  fname   | minit | lname
-----+-----+-----
James     | E     | Borg
Franklin  | T     | Wong
Jennifer  | S     | Wallace
Ramesh    | K     | Narayan
(4 rows)
```

7. Retrieve the names of all employees who work in the department that has the employee with the highest salary among all employees.

```
company=# SELECT FNAME, MINIT, LNAME FROM EMPLOYEE WHERE DNO =
company-# ( SELECT DNO FROM EMPLOYEE WHERE SALARY =
company-# ( SELECT MAX(SALARY) FROM EMPLOYEE ) );
  fname   | minit | lname
-----+-----+-----
James     | E     | Borg
(1 row)
```

8. Count the total number of employees whose salaries exceed \$40,000 in each department.

```
company=# SELECT Dname, COUNT (*)
company-# FROM DEPARTMENT, EMPLOYEE
company-# WHERE Dnumber=Dno AND Salary>40000
company-# GROUP BY Dname;
  dname      | count
-----+-----
Headquarters |      1
Administration |      1
(2 rows)
```

