EXPERIMENT - IV
AGGREGATE FUNCTIONS

July 29, 2019

ADITHYA D RAJAGOPAL
ROLL NO : 9
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM

AIM

To study the SQL aggregate functions:

1. AVG()
2. MAX()
3. MIN()
4. COUNT()
5. SUM()

AVG()

The AVG() aggregate function is used to get the average of a field.

SYNTAX

AVG(fieldname) - Returns average of the field.

MAX()

The MAX() aggregate function is used to get the maximum value in a field.

SYNTAX

MAX(fieldname) - Returns the maximum value of the field.

MIN()

The MIN() aggregate function is used to get the minimum value in a field.

SYNTAX

MIN(fieldname) - Returns the minimum value of the field.

COUNT()

The COUNT() aggregate function is used to count the number of rows in a table.

SYNTAX

1. COUNT(*) - Returns the number of rows in the table including duplicates and those with null values.

2. COUNT(fieldname) - Returns the number of rows where field value is not null.

3. Count(All) : Returns the total number of rows.

SUM()

The SUM() aggregate function is used to get the total sum of a field.

SYNTAX

SUM(fieldname) - Returns the total sum of the field.

QUESTIONS

Create a table named student and populate the table as shown in the table.
The table contains the marks of 10 students for 3 subjects(Physics, Chemistry, Mathematics).The total marks for physics and chemistry is 25.while for mathematics it is 50.The pass mark for physics and chemistry is 12 and for mathematics it is 25.A student is awarded a 'Pass' if he has passed all the subjects.

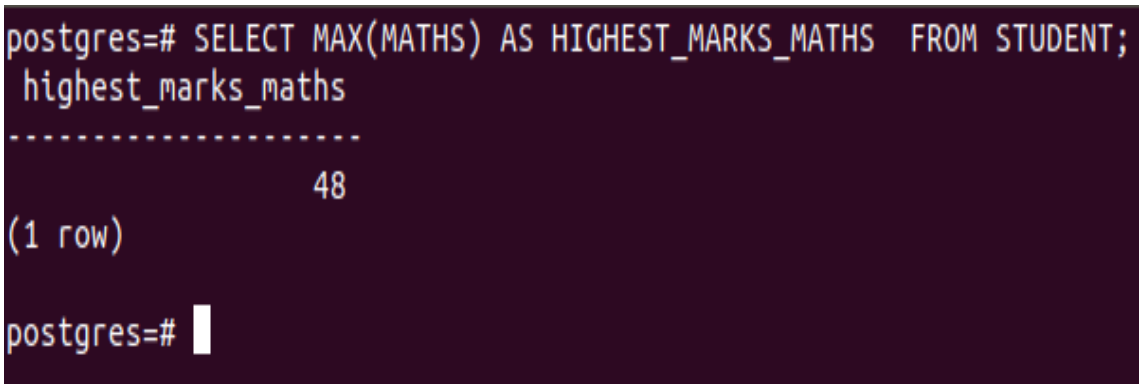
ROLLNO	NAME	PHYSICS	CHEMISTRY	MATHS
1	Adam	20	20	33
2	Bob	18	9	41
3	Bright	22	7	31
4	Duke	13	21	20
5	Elvin	14	22	23
6	Fletcher	2	10	48
7	Georgina	22	12	22
8	Mary	24	14	31
9	Tom	19	15	24
10	Zack	8	20	36

```
postgres=# SELECT * FROM STUDENT;
 rollno |   name    | physics | chemistry | maths
--------+-----------+---------+-----------+-------
      1 | Adam      |      20 |        20 |    33
      2 | Bob       |      18 |         9 |    41
      3 | Bright    |      22 |         7 |    31
      4 | Duke      |      13 |        21 |    20
      5 | Elvin     |      14 |        22 |    23
      6 | Fletcher  |       2 |        10 |    48
      7 | Georgina  |      22 |        12 |    22
      8 | Mary      |      24 |        14 |    31
      9 | Tom       |      19 |        15 |    24
     10 | Zack      |       8 |        20 |    36
(10 rows)

postgres=#
```

1. Find the class average for the subject âĂŸPhysicsâĂŹ

```
postgres=# SELECT AVG(PHYSICS) FROM STUDENT;
         avg
---------------------
 16.2000000000000000
(1 row)

postgres=#
```

2. Find the highest marks for mathematics (To be displayed as highest_marks_maths).



3. Find the lowest marks for chemistry(To be displayed as lowest_mark_chemistry)

```
postgres=# SELECT MIN(CHEMISTRY) AS LOWEST_MARKS_CHEMISTRY FROM STUDENT;
 lowest_marks_chemistry
------------------------
                      7
(1 row)

postgres=#
```

4. Find the total number of students who has got a âĂŸpassâĂŹ in physics.

```
postgres=# SELECT COUNT(ROLLNO) FROM STUDENT
postgres-# WHERE PHYSICS>12;
 count
-------
     8
(1 row)

postgres=#
```

5. Generate the list of students who have passed in all the subjects

```
postgres=# SELECT * FROM STUDENT
postgres-# WHERE PHYSICS>12 AND CHEMISTRY>12 AND MATHS>25;
 rollno | name | physics | chemistry | maths
--------+------+---------+-----------+-------
      1 | Adam |      20 |        20 |    33
      8 | Mary |      24 |        14 |    31
(2 rows)

postgres=#
```

6. Generate a rank list for the class.Indicate Pass/Fail. Ranking based on total marks obtained by the students.

```
postgres=# ALTER TABLE STUDENT
postgres-# ADD TOTAL_MARKS INT;
ALTER TABLE
postgres=# ALTER TABLE STUDENT
postgres-# ADD RESULT TEXT;
ALTER TABLE
postgres=# UPDATE STUDENT
postgres-# SET TOTAL_MARKS=PHYSICS+CHEMISTRY+MATHS;
UPDATE 10
postgres=# UPDATE STUDENT
postgres-# SET RESULT='P'
postgres-# WHERE PHYSICS>=12 AND CHEMISTRY>=12 AND MATHS>=25;
UPDATE 2
```

```
postgres=# UPDATE STUDENT
SET RESULT='F'
WHERE PHYSICS<12 OR CHEMISTRY<12 OR MATHS<25;
UPDATE 8
```

```
postgres=# SELECT * FROM STUDENT
postgres-# ORDER BY TOTAL_MARKS DESC;
 rollno |   name   | physics | chemistry | maths | total_marks | result
--------+----------+---------+-----------+-------+-------------+--------
      1 | Adam     |      20 |        20 |    33 |          73 | P
      8 | Mary     |      24 |        14 |    31 |          69 | P
      2 | Bob      |      18 |         9 |    41 |          68 | F
     10 | Zack     |       8 |        20 |    36 |          64 | F
      6 | Fletcher |       2 |        10 |    48 |          60 | F
      3 | Bright   |      22 |         7 |    31 |          60 | F
      5 | Elvin    |      14 |        22 |    23 |          59 | F
      9 | Tom      |      19 |        15 |    24 |          58 | F
      7 | Georgina |      22 |        12 |    22 |          56 | F
      4 | Duke     |      13 |        21 |    20 |          54 | F
(10 rows)

postgres=#
```

7. Find pass percentage of the class for mathematics.

```
postgres=# SELECT 10*COUNT(*) AS PASS_PERCENTAGE_MATHS
postgres-# FROM STUDENT
postgres-# WHERE MATHS>=25;
 pass_percentage_maths
-----------------------
                    60
(1 row)

postgres=#
```

8. Find the overall pass percentage for all class.

```
postgres=# SELECT 10*COUNT(*) AS PASS_PERCENTAGE
postgres-# FROM STUDENT
postgres-# WHERE RESULT='P';
 pass_percentage
-----------------
              20
(1 row)

postgres=#
```

9. Find the class average.

```
postgres=# SELECT AVG(TOTAL_MARKS) AS CLASS_AVG
postgres-# FROM STUDENT;
      class_avg
---------------------
 62.1000000000000000
(1 row)

postgres=#
```

10. Find the total number of students who have got a Pass.

```
postgres=# SELECT COUNT(RESULT) FROM STUDENT
postgres-# WHERE RESULT='P';
 count
-------
     2
(1 row)

postgres=#
```

RESULT

The query was executed successfully and output was obtained.