|  |  |
| --- | --- |
| **Name** | **Hatim Yusuf Sawai** |
| **UID no.** | **2021300108** |
| **Experiment No.** | **5** |

|  |  |
| --- | --- |
| **AIM:** | **To implement aggregate functions on tables of a database** |
| **PROBLEM STATEMENT:** | To implement all types of aggregate functions including: MIN(), MAX(), COUNT(), AVG() & SUM() and also use GROUP BY & HAVING clauses on the aggregate functions to select specific data |
| **THEORY:** | **AGGREGATE FUNCTIONS:**  Info on all aggregate functions:  1. **COUNT** counts how many rows are in a particular column.  SUM adds together all the values in a particular column.  2. **MIN** & **MAX** return the lowest and highest values in a particular column, respectively.  3. **AVG** calculates the average of a group of selected values.  4. **SUM** calculates the total sum of a numeric column or specified values of the column.  Arithmetic operators only perform operations across rows. Aggregate functions are used to perform operations across entire columns (which could include millions of rows of data or more).    **1. COUNT FUNCTION:**  The COUNT() function returns the number of rows that matches a specified criterion.    **COUNT() Syntax:**  SELECT COUNT(column\_name)  FROM table\_name  WHERE condition;  **2. AVG FUNCTION:**  The AVG() function returns the average value of a numeric column.    **AVG() Syntax:**  SELECT AVG(column\_name)  FROM table\_name  WHERE condition;    **3. SUM FUNCTION:**  The SUM() function returns the total sum of a numeric column.    **SUM() Syntax:**  SELECT SUM(column\_name)  FROM table\_name  WHERE condition;    **4. MIN FUNCTION:**  The MIN() function returns the smallest value of the selected column.    **MIN() Syntax:**  SELECT MIN(column\_name)  FROM table\_name  WHERE condition;    **5. MAX FUNCTION:**  The MAX() function returns the largest value of the selected column.    **Max() Syntax:**  SELECT MAX(column\_name)  FROM table\_name  WHERE condition;  **GROUP BY CLAUSE:**  The GROUP BY statement groups rows that have the same values into summary rows, like "find the number of customers in each country".  The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.  **GROUP BY Syntax:**  SELECT column\_name(s)  FROM table\_name  WHERE condition  GROUP BY column\_name(s)  ORDER BY column\_name(s);    **HAVING CLAUSE:**  The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.    **HAVING Syntax:**  SELECT column\_name(s)  FROM table\_name  WHERE condition  GROUP BY column\_name(s)  HAVING condition  ORDER BY column\_name(s); |
| **SCHEMA:** | **Doctor Table:**  **Patient Table:** |
| **QUERIES:** | **MIN() Queries:**  1. Least paid doctor with id less than 5:  SELECT MIN(Salary) AS "Least-Salary" FROM doctor WHERE D\_id<5;  **Result:**  2. Youngest Patient who lives in Marol:  SELECT MIN(Age) AS "Youngest Patient from Marol" FROM patient WHERE Address="Marol";  **Result:**  **MAX() Queries:**  1. Highest paid doctor who resides in Andheri:  SELECT MAX(Salary) AS "Highest-Salary Doctor From Andheri" FROM doctor WHERE Address="Andheri";  **Result:**  2. Age of Oldest patient without a doctor assigned:  SELECT MAX(Age) AS "Oldest Patient" FROM patient WHERE D\_id>0;  **Result:**  **AVG() Queries:**  1. Average Salary of doctors sorted location-wise:  SELECT Address,AVG(Salary) AS "Average" FROM doctor GROUP BY Address;  **Result:**  **SUM() Queries:**  1. Display Sum of salaries of doctors for each field if the sum is above 4,00,000:  SELECT \* FROM (SELECT Field,SUM(Salary) AS "Total" FROM doctor GROUP BY Field) AS Employee WHERE Total>400000;  **Result:**  2. Display sum of salaries of doctors for each location where sum is above 4,00,00:  SELECT \* FROM (SELECT Address,SUM(Salary) AS "Total" FROM doctor GROUP BY Address) AS Employee WHERE Total>40000;  **Result:**  **COUNT() Queries:**  1. No. of patients without a doctor assigned:  SELECT COUNT(P\_id) AS "No. of Patients Without Doctors" FROM patient WHERE D\_id IS NULL;  **Result:**  2. No. of patients who have been assigned a doctor:  SELECT COUNT(P\_id) AS "No. of Patients With Doctors" FROM patient WHERE D\_id>0;  **Result:** |
| **RESULT:**  **After Manipulation:**  **Doctor table:**  **Patient table:** | |
| **CONCLUSION:** | In this experiment, we learned how to perform aggregate functions like min,max,count,av & sum on tables of the database to display specific results of the data. We also learned how to use the aggregate functions along with Group by & Having clauses to display specific sets of data. |