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:

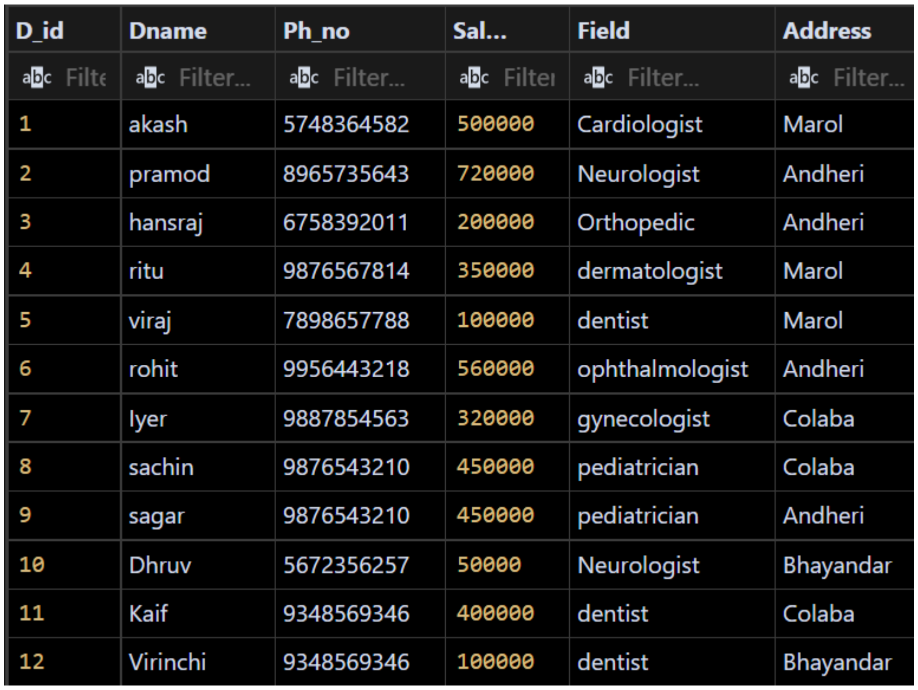
The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

HAVING Syntax:

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

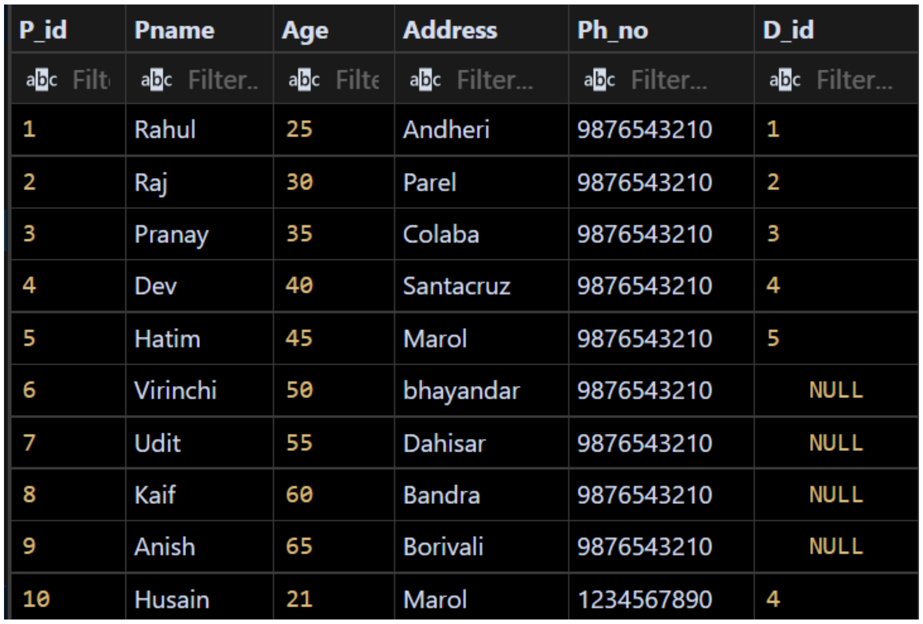
SCHEMA:

Doctor Table:



D_id	Dname	Ph_no	Sal...	Field	Address
1	akash	5748364582	500000	Cardiologist	Marol
2	pramod	8965735643	720000	Neurologist	Andheri
3	hansraj	6758392011	200000	Orthopedic	Andheri
4	ritu	9876567814	350000	dermatologist	Marol
5	viraj	7898657788	100000	dentist	Marol
6	rohit	9956443218	560000	ophthalmologist	Andheri
7	Iyer	9887854563	320000	gynecologist	Colaba
8	sachin	9876543210	450000	pediatrician	Colaba
9	sagar	9876543210	450000	pediatrician	Andheri
10	Dhruv	5672356257	50000	Neurologist	Bhayandar
11	Kaif	9348569346	400000	dentist	Colaba
12	Virinchi	9348569346	100000	dentist	Bhayandar

Patient Table:



P_id	Pname	Age	Address	Ph_no	D_id
1	Rahul	25	Andheri	9876543210	1
2	Raj	30	Parel	9876543210	2
3	Pranay	35	Colaba	9876543210	3
4	Dev	40	Santacruz	9876543210	4
5	Hatim	45	Marol	9876543210	5
6	Virinchi	50	bhayandar	9876543210	NULL
7	Udit	55	Dahisar	9876543210	NULL
8	Kaif	60	Bandra	9876543210	NULL
9	Anish	65	Borivali	9876543210	NULL
10	Husain	21	Marol	1234567890	4

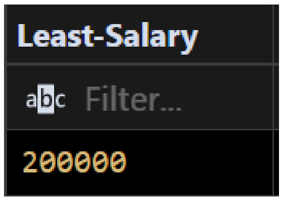
QUERIES:

MIN() Queries:

1. Least paid doctor with id less than 5:

SELECT MIN(Salary) AS "Least-Salary" FROM doctor WHERE D_id<5;

Result:

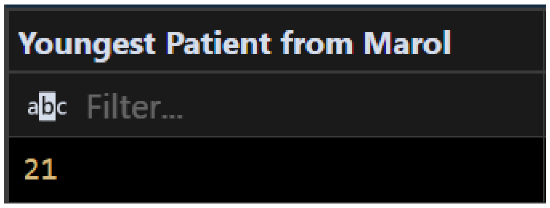


```
Least-Salary
Filter...
200000
```

2. Youngest Patient who lives in Marol:

SELECT MIN(Age) AS "Youngest Patient from Marol" FROM patient WHERE Address="Marol";

Result:



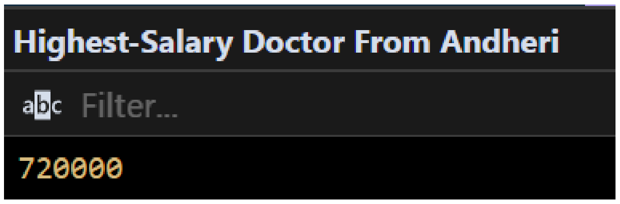
```
Youngest Patient from Marol
Filter...
21
```

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:

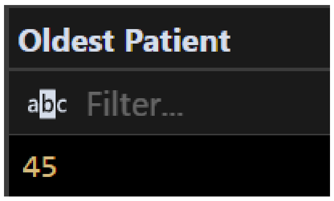


```
Highest-Salary Doctor From Andheri
Filter...
720000
```

2. Age of Oldest patient without a doctor assigned:

SELECT MAX(Age) AS "Oldest Patient" FROM patient WHERE D_id>0;

Result:



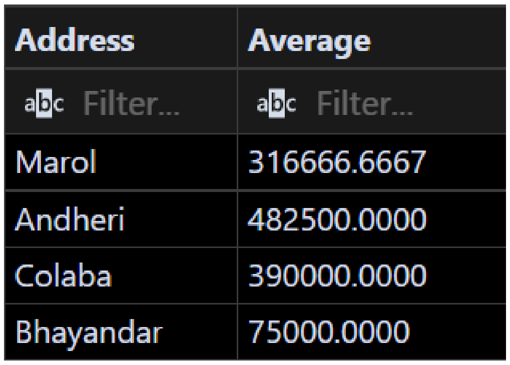
```
Oldest Patient
Filter...
45
```

AVG() Queries:

1. Average Salary of doctors sorted location-wise:

```
SELECT Address,AVG(Salary) AS "Average" FROM doctor GROUP BY Address;
```

Result:



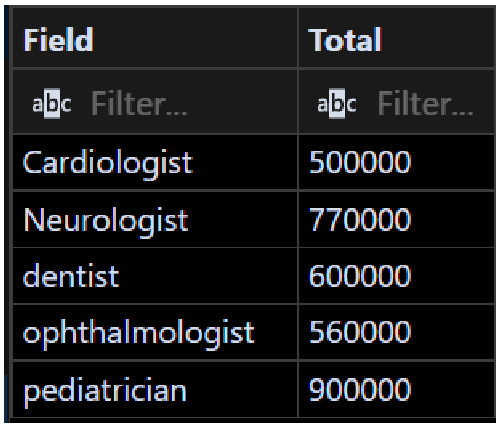
Address	Average
Filter...	Filter...
Marol	316666.6667
Andheri	482500.0000
Colaba	390000.0000
Bhayandar	75000.0000

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:

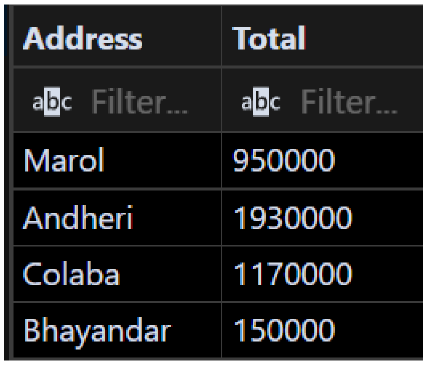


Field	Total
Filter...	Filter...
Cardiologist	500000
Neurologist	770000
dentist	600000
ophthalmologist	560000
pediatrician	900000

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:



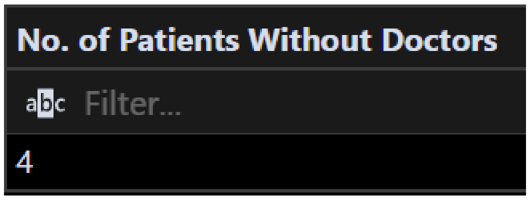
Address	Total
Marol	950000
Andheri	1930000
Colaba	1170000
Bhayandar	150000

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:

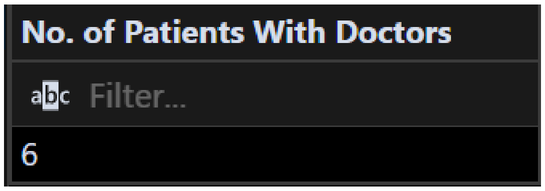


No. of Patients Without Doctors
4

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:

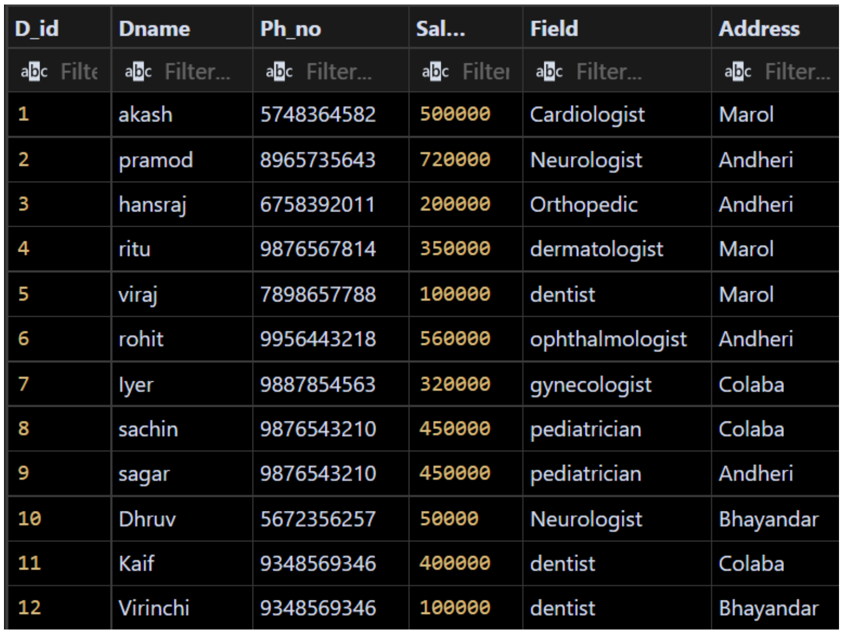


No. of Patients With Doctors
6

RESULT:

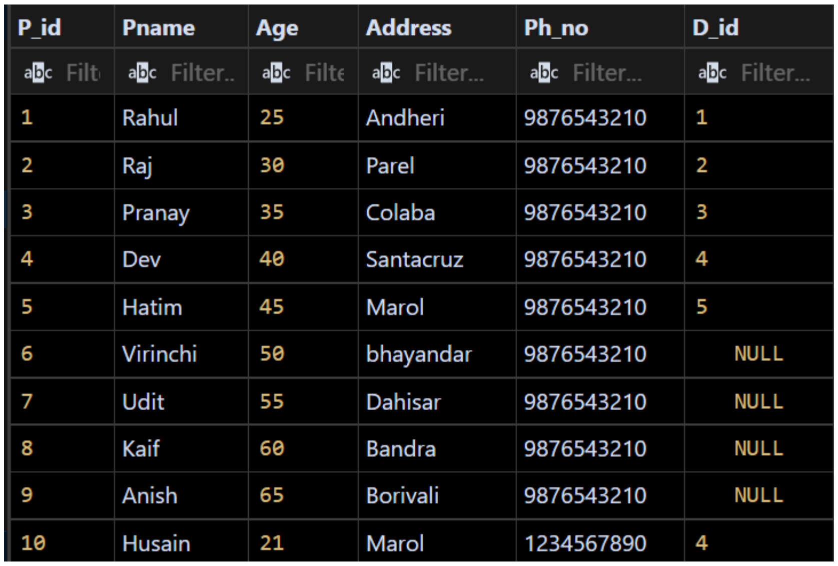
After Manipulation:

Doctor table:



D_id	Dname	Ph_no	Sal...	Field	Address
Filte	Filter...	Filter...	Filter	Filter...	Filter...
1	akash	5748364582	500000	Cardiologist	Marol
2	pramod	8965735643	720000	Neurologist	Andheri
3	hansraj	6758392011	200000	Orthopedic	Andheri
4	ritu	9876567814	350000	dermatologist	Marol
5	viraj	7898657788	100000	dentist	Marol
6	rohit	9956443218	560000	ophthalmologist	Andheri
7	Iyer	9887854563	320000	gynecologist	Colaba
8	sachin	9876543210	450000	pediatrician	Colaba
9	sagar	9876543210	450000	pediatrician	Andheri
10	Dhruv	5672356257	50000	Neurologist	Bhayandar
11	Kaif	9348569346	400000	dentist	Colaba
12	Virinchi	9348569346	100000	dentist	Bhayandar

Patient table:



P_id	Pname	Age	Address	Ph_no	D_id
Filt	Filter..	Filte	Filter...	Filter...	Filter...
1	Rahul	25	Andheri	9876543210	1
2	Raj	30	Parel	9876543210	2
3	Pranay	35	Colaba	9876543210	3
4	Dev	40	Santacruz	9876543210	4
5	Hatim	45	Marol	9876543210	5
6	Virinchi	50	bhayandar	9876543210	NULL
7	Udit	55	Dahisar	9876543210	NULL
8	Kaif	60	Bandra	9876543210	NULL
9	Anish	65	Borivali	9876543210	NULL
10	Husain	21	Marol	1234567890	4

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.