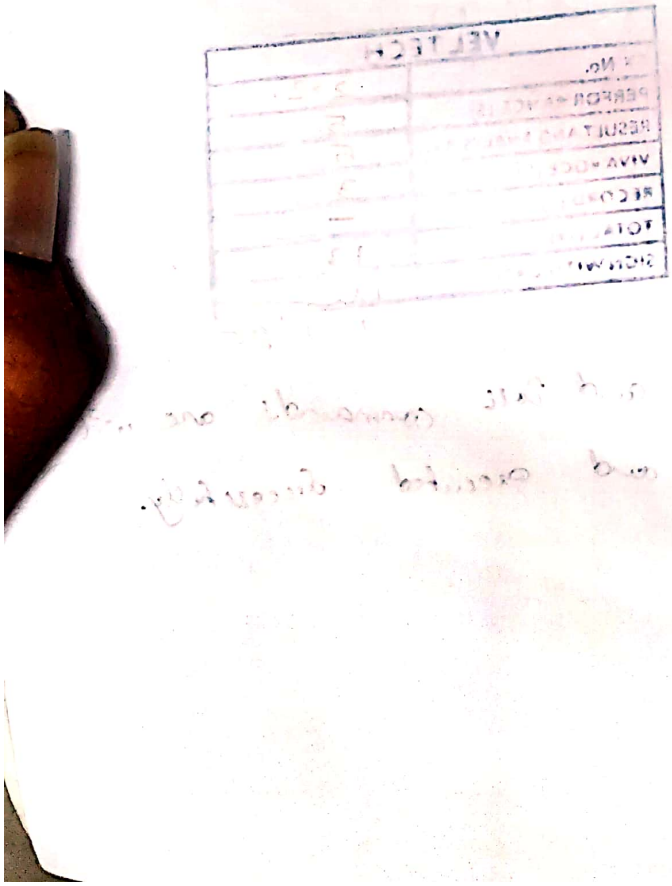
Table After insert :

Patient ID	Patient Name	Department	Gender
111	Arun	Cardiology	Male

table after UPDATE :

Patient ID	Patient name	Department	Gender
111	Kumar	Cardiology	Male



Task 3.1 :- DML Commands using clauses operators and functions in queries

Aim :- To implement of DML commands using clauses, operators and functions in queries.

Data manipulation language (DML) :-

The data manipulation language (DML) is using to retrieve, insert and modify database information these commands will be used by all database users during the routine operations of the database lets take a brief look at the basic look at the Basic DML command.

1. INSERT
2. UPDATE
3. DELETE

INSERT INTO :-

This is used to add records into a relation

Syntax :-

INSERT INTO <table_name> (field 1, field 2 ... field N) values (data 1, data 2 ... data n);

Example :-

SQL:

INSERT INTO Patient VALUES (111, 'Arun', 'Cardiology', 'Male');

UPDATE - SET - WHERE :-

This is used to update the content of a record in a relation.

SYNTAX :-

SQL:

UPDATE table_name SET field = data WHERE condition;

Example :-

UPDATE Patient SET patient_name = 'Kumar' WHERE Patient_ID = '110';

Appointments table after DELETE:

Appointment ID	Patient ID	Doctor ID	Appointment Date

Doctor table after DELETE:

Doctor ID	Doctor name	Department	Fees
201	Dr. Sharma	Cardiology	1000
203	Dr. Ahmed	Neurology	900
204	Dr. Rajesh	orthopedic	500
205	Dr. Neha	Dermatology	800

Patients table after TRUNCATE:

Patient ID	Patient name	Department	Gender

DELETE FROM:-

This is used to delete all records of a relation but it retains the structure.

SYNTAX:-

Sql:
DELETE FROM table_name;

Example:-

Sql:
DELETE FROM Appointments;

DELETE FROM WHERE:

This is used to delete specific records from a relation

Syntax:
Sql:-
DELETE FROM table_name WHERE condition

Example:
Sql:-
DELETE FROM Doctor where Doctor ID = 202;

TRUNCATE:-

This removes all data permanently but keeps the table structure

Syntax:-
Sql:-
TRUNCATE TABLE <table_names;

Example:-
Sql:
TRUNCATE TABLE patients;

Sample Queries and output:-

1. Retrive patient names ending with letter 'n' and patient no b/w 111 and 115

Patient name	Department	Gender
Arun	Cardiology	Male
Karan	orthopedic	Male
Rohan	Dermatology	Male

Doctor ID	Doctor name	Department	fees
202	Dr. priya	padiatric	700
205	Dr. Neha	Dermatology	800

Appointment ID	patient ID	Doctor ID	Appointment Date
302	112	203	2023-02-07
303	113	204	2023-02-09
304	114	202	2023-02-10
305	115	205	2023-02-12

Patient ID
111
112
113
114
115

Query:-

Sql-1:-

SELECT patient name, department, Gender from patients WHERE patient Name LIKE "%m" AND patient ID BETWEEN 111 AND 115.

2. List doctors where consultation fees b/w 700 and 800

Query:-

Sql 1:-

SELECT * FROM Doctor WHERE fees BETWEEN 700 AND 800

3. find the record with minimum appointment duration

Query:-

Sql 1:

SELECT MIN(DURATION) FROM Appointment;

MIN (Duration)
20

4. find appointment with date > "2023-02-07"

Query:-

Sql 1:-

SELECT * FROM Appointment WHERE APPOINTMENT Date >= (2023-02-07)

5. List distinct IDs

Query

Sql 1:-

SELECT DISTINCT Patient ID FROM patient.

~~Patient ID~~

Output:-

Patient ID
1
2
3
4
6

Department	Gender	Total patients
Cardiology	Male	1
Neurology	female	1
orthopedics	Male	1
pediatrics	female	1
Dermatology	Male	1

Doctor Name	Department	Count
Dr. Ahmed	Neurology	1
Dr. Neha	Dermatology	1
Dr. Priya	pediatrics	1
Dr. Rajesh	orthopedics	1
Dr. Sharma	Cardiology	1

6. Combine Patient IDs from patients and Appointments (UNION)

Query :-

Sql :-

```
SELECT patientID FROM patient
UNION
SELECT patient ID FROM Appointments;
```

7. Group patients based on gender and department

Query :-

Sql :-

```
SELECT Department, Gender, COUNT(*) AS
total patients
FROM patients
GROUP By Department, Gender.
```

8. find doctors and their department details using GROUP By and ORDER By

Query :-

Sql :-

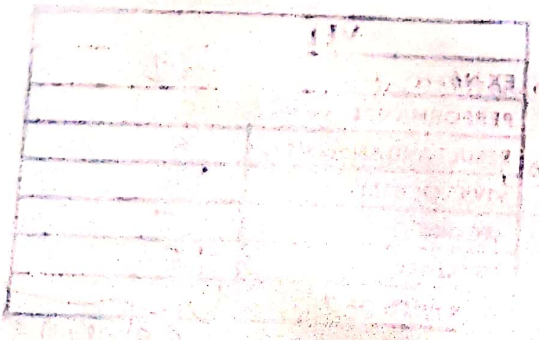
```
SELECT Doctor Name, Department, COUNT(*) AS
count
FROM Doctors
GROUP By Doctor Name, Department
ORDER BY Doctor Name;
```

VEL	
EXPERIMENT	31
PERFORMANCE	5
RESULT AND ANALYSIS	5
VIVA VOCE (5)	5
RECORD (5)	—
TOTAL (20)	15
N WITH DATE	25/08/25

Result :- the implementation of DML Commands using clauses, joins and functions in Queries executed successfully.

Example table : patients.

PatientID	Patient Name	Department	Bill Amount
101	Arun	Cardiology	2000
102	Sneha	Neurology	3500
103	Karan	Orthopedics	1500
104	Meera	pediatrics	4000
105	Rohan	Dermatology	



Task 3.2 Aggregate Functions (Multiple row operations)

Aim:-

To study and implement aggregate functions (count(), sum(), AVG(), MIN(), MAX()) on a sample patient database.

Procedure:-

1. Create a table named students.
2. Insert sample records.
3. write queries using aggregate functions.
4. observe and record the output.

COMMANDS WITH EXPLANATION:

~~EXAMPLE table: Pat~~

1. Count the total number of patient

Sql:

```
SELECT COUNT(*) As Total_Patients
FROM patients;
```

Output:-

Total_patients
5

2. find the highest bill amount

Sql:

```
SELECT MAX(Bill Amount) As Highest_Bill FROM patients.
```

Output:-

Highest_Bill
4000

3. find the average bill amount of patient

Sql:

SELECT AVG (Bill Amount) As Average_bill
from Patients;

Output:-

Average_Bill
2700

4. find the minimum bill amount among patients in Neurology department

Sql:-

SELECT MIN (Bill Amount) As Min_Neuro_Bill

Output:-

Min_Neuro_Bill
3500

5. find the total bill amount by each department

Sql:

SELECT Department, SUM (Bill Amount) As
Total_Bill
from Patients
Group By Department;

Output:-

Department	Total_Bill
Cardiology	2000
Neurology	3500
orthopedics	1500
pediatrics	4000
Dermatology	

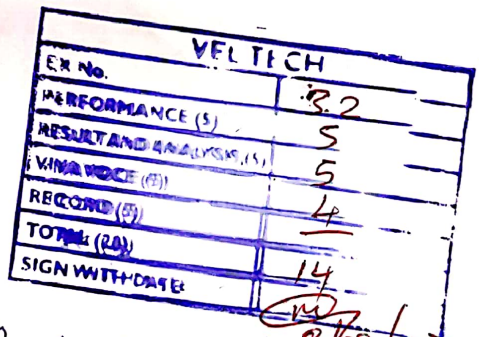
6. Find the average bill per department, ordered by average descending

Sql:-

```
SELECT Department, AVG(Bill Amount) As Aver Bill
FROM patients
Group By Department
ordered By Avg_Bill DESC;
```

Output:-

Department	Avg_Bill
pediatrics	4000
Neurology	3500
Dermatology	2500
Cardiology	200
orthopedic	1500



Result:- The implementation of Aggregate functions are executed Successfully.