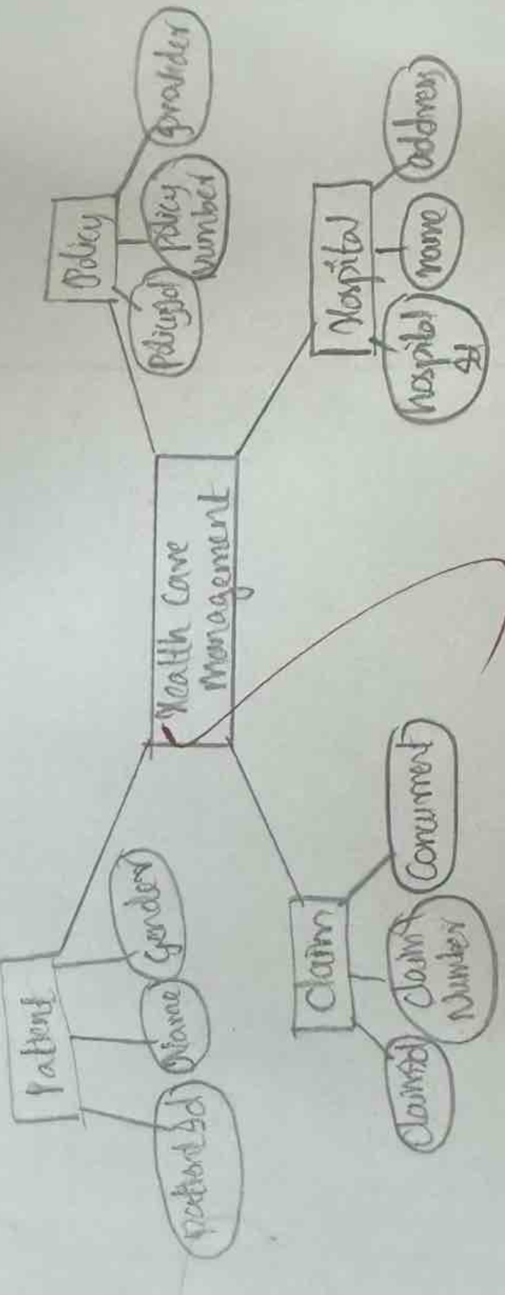
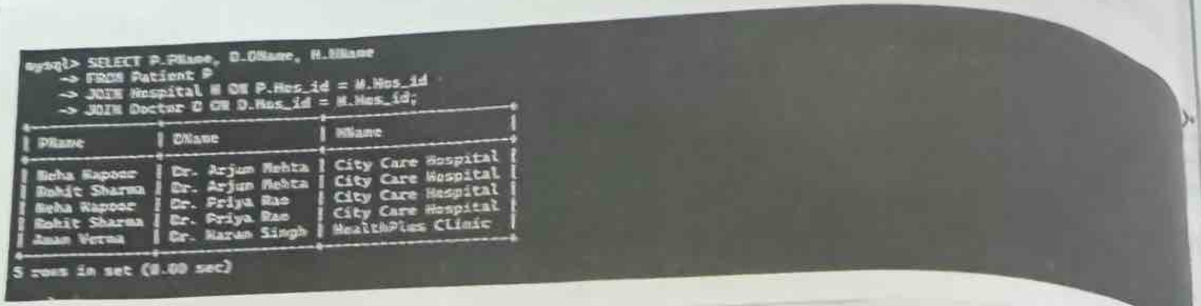
Task 12

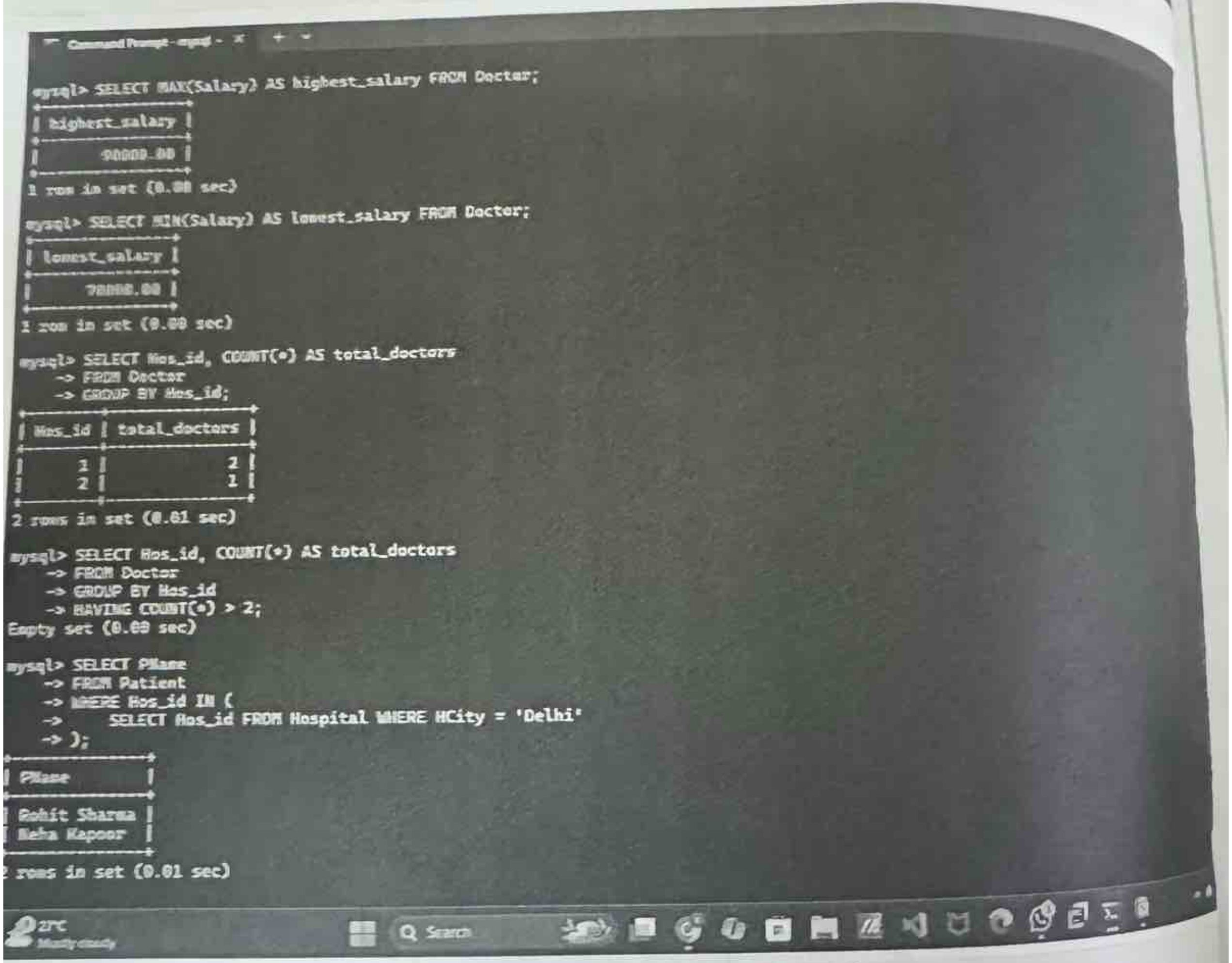
MINI PROJECT

20. E-R Diagram





```
mysql> SELECT P.PName, D.DName, H.HName
    -> FROM Patient P
    -> JOIN Hospital H ON P.Hos_id = H.Hos_id
    -> JOIN Doctor D ON D.Hos_id = H.Hos_id;
+--------------+-------------------+----------------------+
| PName        | DName             | HName                |
+--------------+-------------------+----------------------+
| Neha Kapoor  | Dr. Arjun Mehta   | City Care Hospital   |
| Rohit Sharma | Dr. Arjun Mehta   | City Care Hospital   |
| Neha Kapoor  | Dr. Priya Rao     | City Care Hospital   |
| Rohit Sharma | Dr. Priya Rao     | City Care Hospital   |
| Aman Verma   | Dr. Karan Singh   | HealthPlus Clinic    |
+--------------+-------------------+----------------------+
5 rows in set (0.00 sec)
```



Command Prompt - mysql

```
mysql> SELECT MAX(Salary) AS highest_salary FROM Doctor;
+----------------+
| highest_salary |
+----------------+
|       90000.00 |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT MIN(Salary) AS lowest_salary FROM Doctor;
+---------------+
| lowest_salary |
+---------------+
|      70000.00 |
+---------------+
1 row in set (0.00 sec)

mysql> SELECT Hos_id, COUNT(*) AS total_doctors
    -> FROM Doctor
    -> GROUP BY Hos_id;
+--------+---------------+
| Hos_id | total_doctors |
+--------+---------------+
|      1 |             2 |
|      2 |             1 |
+--------+---------------+
2 rows in set (0.01 sec)

mysql> SELECT Hos_id, COUNT(*) AS total_doctors
    -> FROM Doctor
    -> GROUP BY Hos_id
    -> HAVING COUNT(*) > 2;
Empty set (0.00 sec)

mysql> SELECT PName
    -> FROM Patient
    -> WHERE Hos_id IN (
    ->     SELECT Hos_id FROM Hospital WHERE HCity = 'Delhi'
    -> );
+--------------+
| PName        |
+--------------+
| Rohit Sharma |
| Neha Kapoor  |
+--------------+
2 rows in set (0.01 sec)
```

1. Normalization Steps :

- First Normal Form : Ensure atomic attribute values for each table.
- Second Normal Form ; Remove partial dependencies ; each non-key attributes must depend on the whole primary key.
- Third Normal Form : Remove transitive dependencies ; non-key attributes depend only on the primary key.

Candidate keys Example :

- Patient : PatientId (unique)
- Policy : PolicyNumber (unique)
- Claim : ClaimId (unique)
- Hospital : HospitalId (unique)

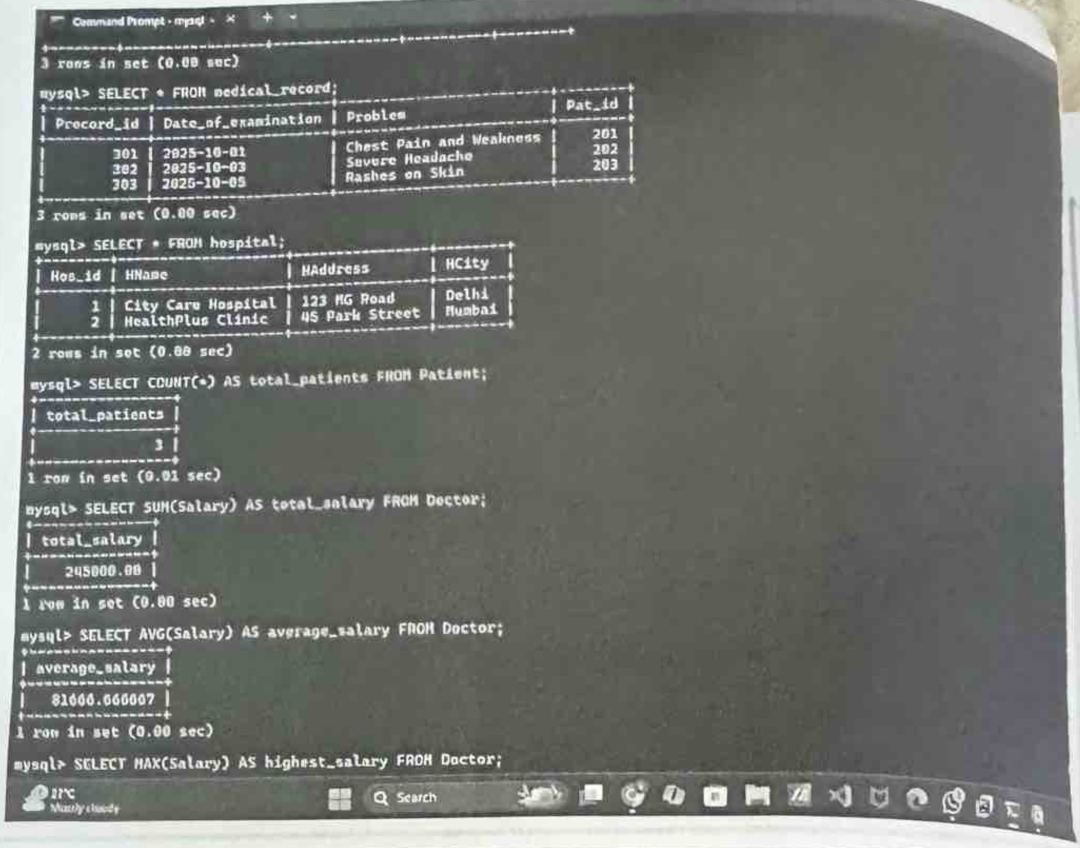
22. SQL Queries to process, approve and track insurance claims.

- Insert new claim :

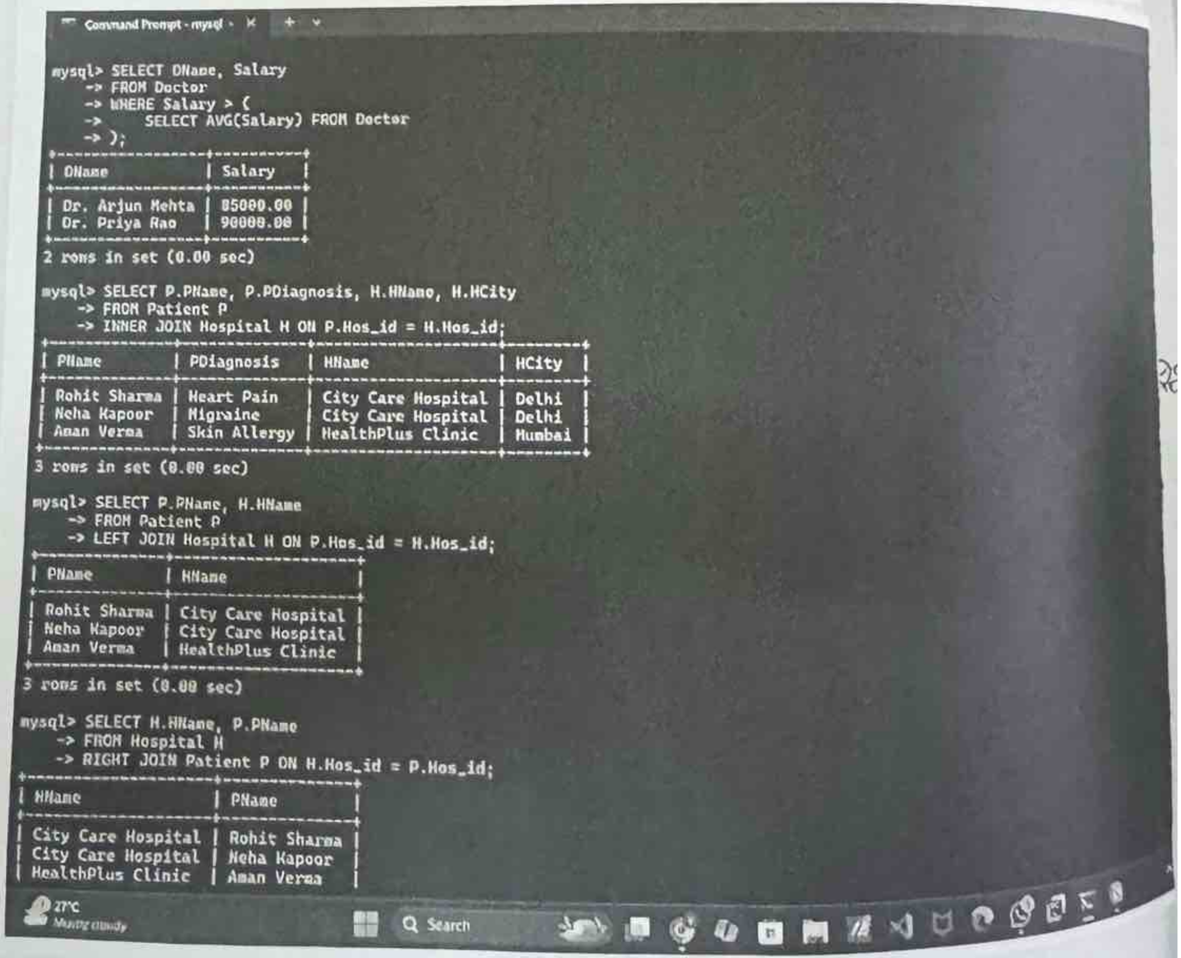
```
INSERT INTO claim( claimId, PolicyId, PatientId, Hospital Id, claimNumber,
                                  amount)
VALUES ( 1001, 'PN123', 1, 101, 22, 2000);
```

- Approve a claim :

```
Update claim SET status = 'Approval' , Approval Date = '2023-10-
 WHERE claimId = 1001 ;
```



```
+----------+-----------------------+--------------+--------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM medical_record;
+------------+---------------------+---------------------------+--------+
| Precord_id | Date_of_examination | Problem                   | Pat_id |
+------------+---------------------+---------------------------+--------+
|        301 | 2025-10-01          | Chest Pain and Weakness   |    201 |
|        302 | 2025-10-03          | Severe Headache           |    202 |
|        303 | 2025-10-05          | Rashes on Skin            |    203 |
+------------+---------------------+---------------------------+--------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM hospital;
+--------+---------------------+-----------------+--------+
| Hos_id | HName               | HAddress        | HCity  |
+--------+---------------------+-----------------+--------+
|      1 | City Care Hospital  | 123 MG Road     | Delhi  |
|      2 | HealthPlus Clinic   | 45 Park Street  | Mumbai |
+--------+---------------------+-----------------+--------+
2 rows in set (0.00 sec)

mysql> SELECT COUNT(*) AS total_patients FROM Patient;
+----------------+
| total_patients |
+----------------+
|              3 |
+----------------+
1 row in set (0.01 sec)

mysql> SELECT SUM(Salary) AS total_salary FROM Doctor;
+--------------+
| total_salary |
+--------------+
|    245000.00 |
+--------------+
1 row in set (0.00 sec)

mysql> SELECT AVG(Salary) AS average_salary FROM Doctor;
+----------------+
| average_salary |
+----------------+
|   81666.666667 |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT MAX(Salary) AS highest_salary FROM Doctor;
```



```
mysql> SELECT DName, Salary
    -> FROM Doctor
    -> WHERE Salary > (
    ->     SELECT AVG(Salary) FROM Doctor
    -> );
+-----------------+----------+
| DName           | Salary   |
+-----------------+----------+
| Dr. Arjun Mehta | 85000.00 |
| Dr. Priya Rao   | 90000.00 |
+-----------------+----------+
2 rows in set (0.00 sec)

mysql> SELECT P.PName, P.PDiagnosis, H.HName, H.HCity
    -> FROM Patient P
    -> INNER JOIN Hospital H ON P.Hos_id = H.Hos_id;
+--------------+--------------+--------------------+--------+
| PName        | PDiagnosis   | HName              | HCity  |
+--------------+--------------+--------------------+--------+
| Rohit Sharma | Heart Pain   | City Care Hospital | Delhi  |
| Neha Kapoor  | Migraine     | City Care Hospital | Delhi  |
| Aman Verma   | Skin Allergy | HealthPlus Clinic  | Mumbai |
+--------------+--------------+--------------------+--------+
3 rows in set (0.00 sec)

mysql> SELECT P.PName, H.HName
    -> FROM Patient P
    -> LEFT JOIN Hospital H ON P.Hos_id = H.Hos_id;
+--------------+--------------------+
| PName        | HName              |
+--------------+--------------------+
| Rohit Sharma | City Care Hospital |
| Neha Kapoor  | City Care Hospital |
| Aman Verma   | HealthPlus Clinic  |
+--------------+--------------------+
3 rows in set (0.00 sec)

mysql> SELECT H.HName, P.PName
    -> FROM Hospital H
    -> RIGHT JOIN Patient P ON H.Hos_id = P.Hos_id;
+--------------------+--------------+
| HName              | PName        |
+--------------------+--------------+
| City Care Hospital | Rohit Sharma |
| City Care Hospital | Neha Kapoor  |
| HealthPlus Clinic  | Aman Verma   |
```

• Track claims for a given policy:

```
SELECT * FROM claim WHERE PolicyNumber = 'PN123';
```

• Calculate total approved claims per patient:

```
SELECT patientId, SUM(Amount) AS TotalApproved
FROM claim
WHERE status = 'Approved'
GROUP BY PatientId;
```

Q3. Transaction Management during concurrent claim approvals

1. Optimistic concurrency control:

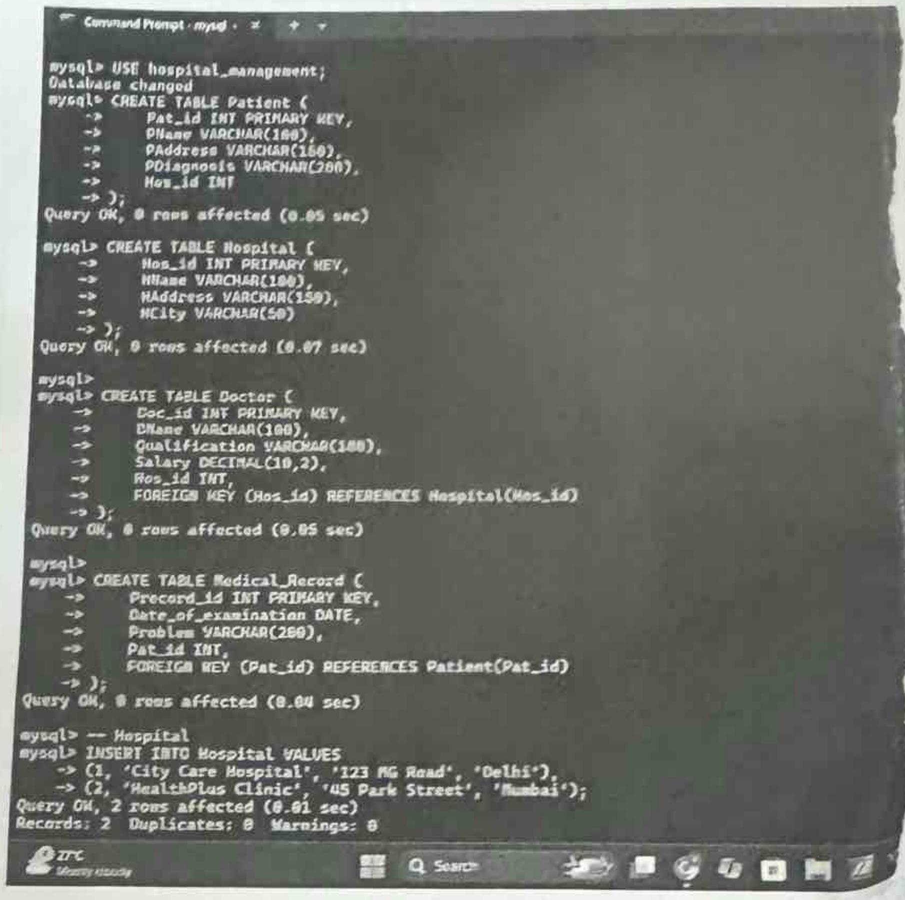
```
UPDATE claim
  SET status = 'Approved', version = version + 1
WHERE claim_Id = 'uuid_1234' AND status = 'IN_REVIEW' AND version
                        = 3;
```

2. Pessimitic Locking

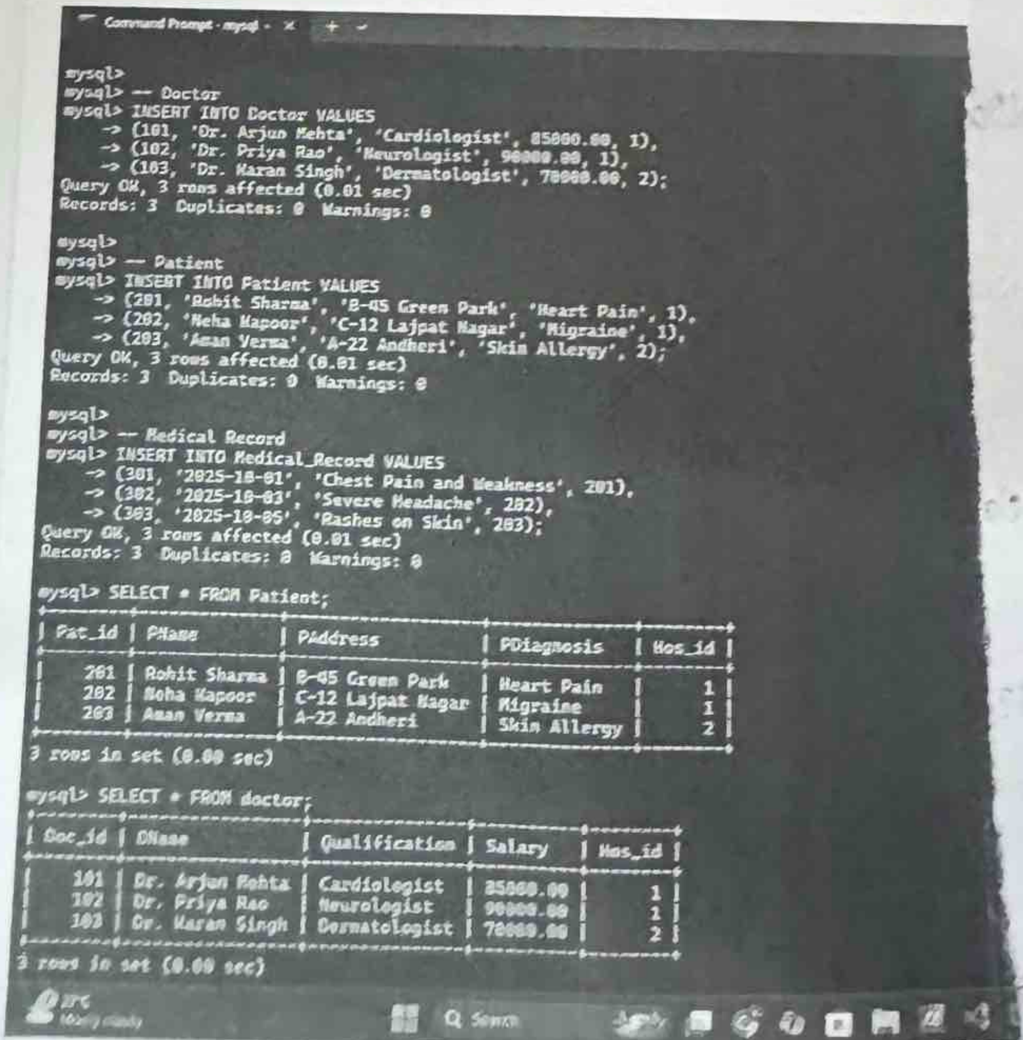
```
BEGIN;
SELECT status FROM claim WHERE claim_id = 'uuid_1234' FOR UPDATE
UPDATE claim SET status = 'Approved' WHERE claim_id = 'uuid_12
COMMIT;
```

3. Idempotency & business-level compensation

Make approval operations idempotent.



```
mysql> USE hospital_management;
Database changed
mysql> CREATE TABLE Patient (
    ->     Pat_id INT PRIMARY KEY,
    ->     PName VARCHAR(100),
    ->     PAddress VARCHAR(150),
    ->     PDiagnosis VARCHAR(200),
    ->     Hos_id INT
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE Hospital (
    ->     Hos_id INT PRIMARY KEY,
    ->     HName VARCHAR(100),
    ->     HAddress VARCHAR(150),
    ->     HCity VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.07 sec)

mysql>
mysql> CREATE TABLE Doctor (
    ->     Doc_id INT PRIMARY KEY,
    ->     DName VARCHAR(100),
    ->     Qualification VARCHAR(100),
    ->     Salary DECIMAL(10,2),
    ->     Hos_id INT,
    ->     FOREIGN KEY (Hos_id) REFERENCES Hospital(Hos_id)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql>
mysql> CREATE TABLE Medical_Record (
    ->     Precord_id INT PRIMARY KEY,
    ->     Date_of_examination DATE,
    ->     Problem VARCHAR(200),
    ->     Pat_id INT,
    ->     FOREIGN KEY (Pat_id) REFERENCES Patient(Pat_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> -- Hospital
mysql> INSERT INTO Hospital VALUES
    -> (1, 'City Care Hospital', '123 MG Road', 'Delhi'),
    -> (2, 'HealthPlus Clinic', '45 Park Street', 'Mumbai');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```



```
mysql>
mysql> -- Doctor
mysql> INSERT INTO Doctor VALUES
    -> (101, 'Dr. Arjun Mehta', 'Cardiologist', 85000.00, 1),
    -> (102, 'Dr. Priya Rao', 'Neurologist', 90000.00, 1),
    -> (103, 'Dr. Karan Singh', 'Dermatologist', 70000.00, 2);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Patient
mysql> INSERT INTO Patient VALUES
    -> (201, 'Rohit Sharma', 'B-45 Green Park', 'Heart Pain', 1),
    -> (202, 'Neha Kapoor', 'C-12 Lajpat Nagar', 'Migraine', 1),
    -> (203, 'Aman Verma', 'A-22 Andheri', 'Skin Allergy', 2);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Medical Record
mysql> INSERT INTO Medical_Record VALUES
    -> (301, '2025-10-01', 'Chest Pain and Weakness', 201),
    -> (302, '2025-10-03', 'Severe Headache', 202),
    -> (303, '2025-10-05', 'Rashes on Skin', 203);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Patient;
+--------+--------------+-------------------+--------------+--------+
| Pat_id | PName        | PAddress          | PDiagnosis   | Hos_id |
+--------+--------------+-------------------+--------------+--------+
|    201 | Rohit Sharma | B-45 Green Park   | Heart Pain   |      1 |
|    202 | Neha Kapoor  | C-12 Lajpat Nagar | Migraine     |      1 |
|    203 | Aman Verma   | A-22 Andheri      | Skin Allergy |      2 |
+--------+--------------+-------------------+--------------+--------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM doctor;
+--------+-----------------+---------------+----------+--------+
| Doc_id | DName           | Qualification | Salary   | Hos_id |
+--------+-----------------+---------------+----------+--------+
|    101 | Dr. Arjun Mehta | Cardiologist  | 85000.00 |      1 |
|    102 | Dr. Priya Rao   | Neurologist   | 90000.00 |      1 |
|    103 | Dr. Karan Singh | Dermatologist | 70000.00 |      2 |
+--------+-----------------+---------------+----------+--------+
3 rows in set (0.00 sec)
```

SUBMITTED → IN_REVIEW → Approved.

4. Isolation Levels

Use read committed typically, or REPEATABLE READ | SERIALIZE if strict serializibility is required.

4. Performing CRUD Operations in MongoDB to manage claim records.

• Basic MongoDB CRUD operations for claim records :

• Create ;

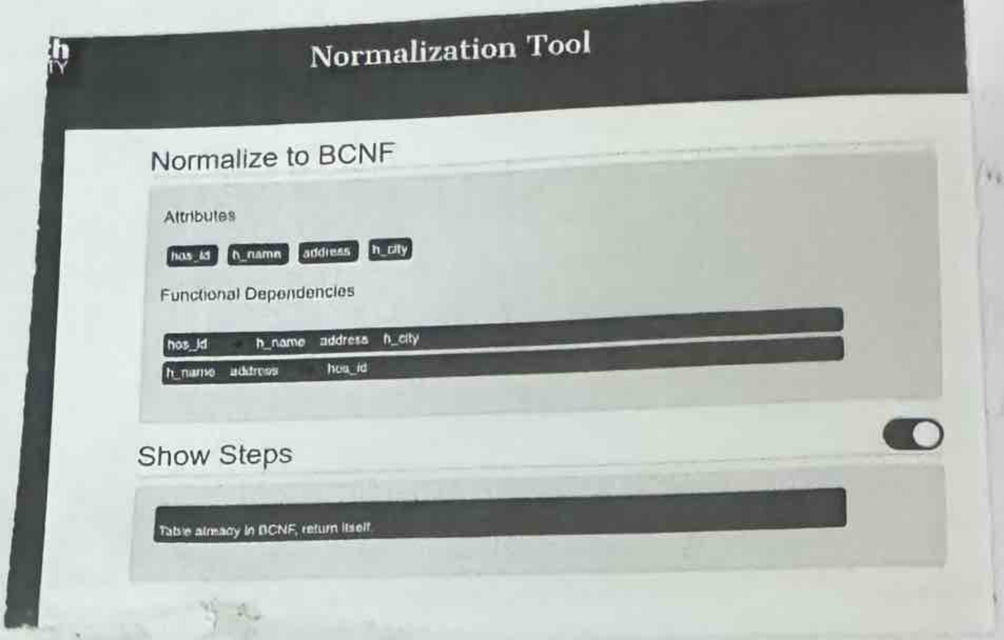
```
db.claims.insertOne({
    claimId : 1001,
 PolicyNumber : 'PN123',
  PatientID   : 1,
 HospitalId   : 101,
 claimdate    : '2023-10-01',
 Amount       : 2000,
 Status       : 'Submitted'
});
```

• Read ;

```
db.claims.find({policyNumber :'PN123'});
```

• Update :

```
db.claims.updateOne({
```



Normalization Tool

Normalize to BCNF

Attributes

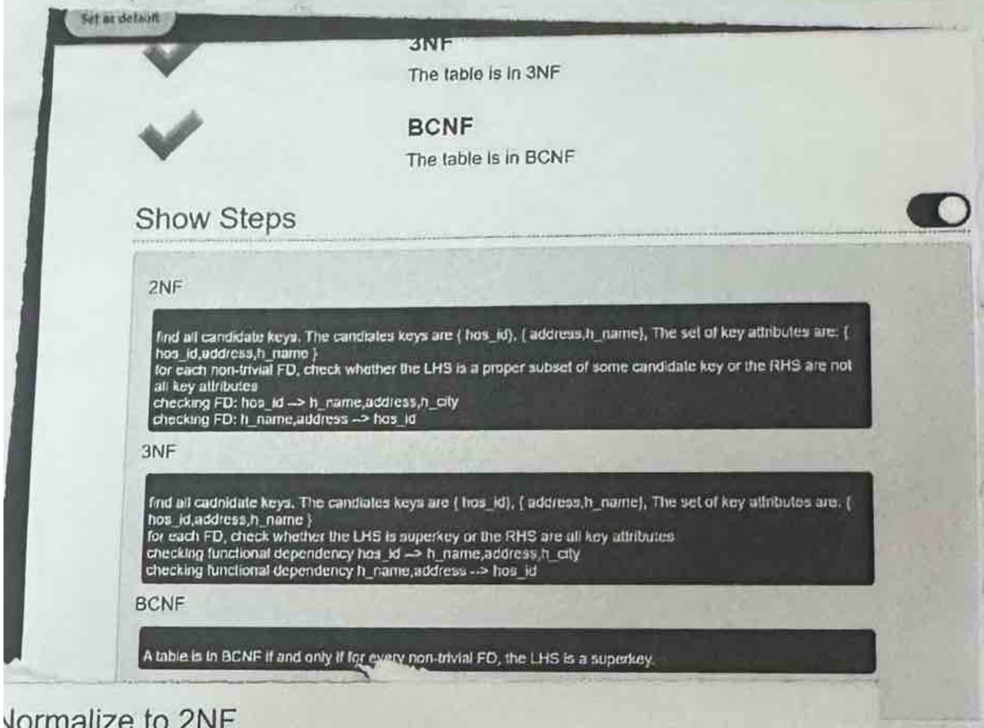
hos_id h_name address h_city

Functional Dependencies

```
hos_id  → h_name  address  h_city
h_name  address  → hos_id
```

Show Steps

```
Table already in BCNF, return itself.
```



Set as default

3NF
The table is in 3NF

BCNF
The table is in BCNF

Show Steps

2NF

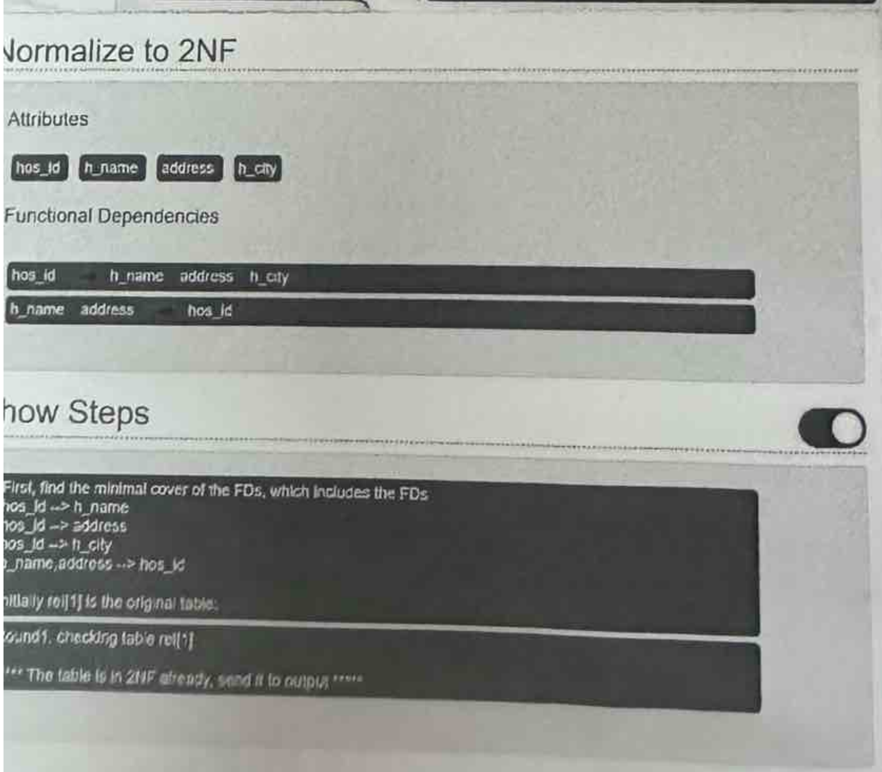
```
find all candidate keys. The candiates keys are { hos_id}, { address,h_name}, The set of key attributes are: {
hos_id,address,h_name }
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not
all key attributes
checking FD: hos_id --> h_name,address,h_city
checking FD: h_name,address --> hos_id
```

3NF

```
find all cadnidate keys. The candiates keys are { hos_id}, { address,h_name}, The set of key attributes are: {
hos_id,address,h_name }
for each FD, check whether the LHS is superkey or the RHS are all key attributes
checking functional dependency hos_id --> h_name,address,h_city
checking functional dependency h_name,address --> hos_id
```

BCNF

```
A table is in BCNF if and only if for every non-trivial FD, the LHS is a superkey.
```



Normalize to 2NF

Attributes

hos_id h_name address h_city

Functional Dependencies

```
hos_id  → h_name  address  h_city
h_name  address  → hos_id
```

Show Steps

```
First, find the minimal cover of the FDs, which includes the FDs
hos_id --> h_name
hos_id --> address
hos_id --> h_city
h_name,address --> hos_id

Initially rel[1] is the original table.

Round1, checking table rel[1]

***** The table is in 2NF already, send it to output *****
```

UBMITTED

Isolation L

se read

erforming

Basic Mo

Create ;

db.clai

clai

PolicyN

Patient

Hospital

claimda

Amoun

Status

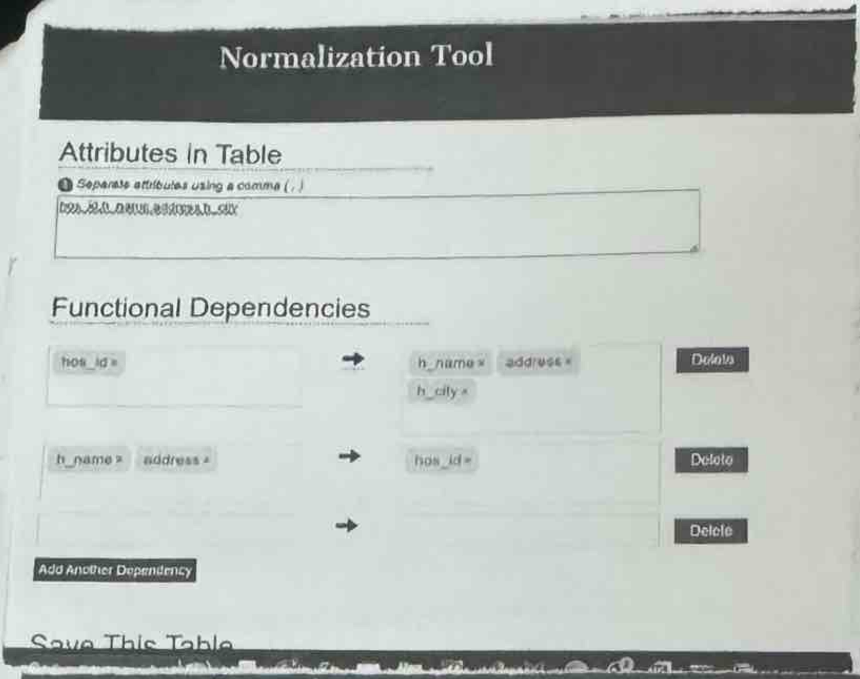
});

Read ;

db.c

Update

d



Normalization Tool

Attributes in Table

Separate attributes using a comma (,)

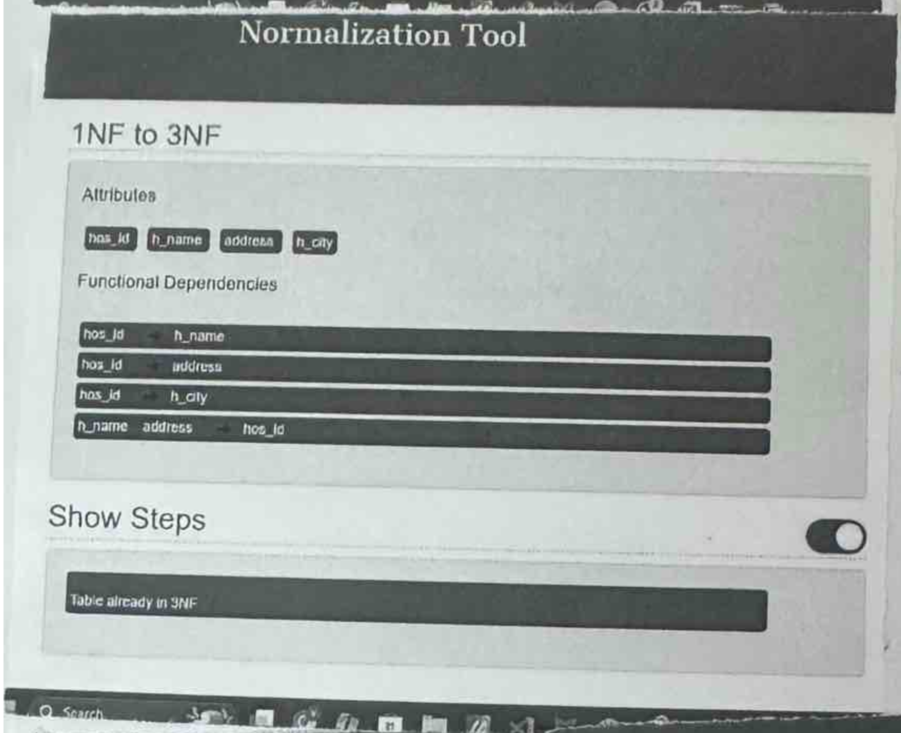
hos_id,h_name,address,h_city

Functional Dependencies

hos_id ×	→	h_name × address × h_city ×	Delete
h_name × address ×	→	hos_id ×	Delete
	→		Delete

Add Another Dependency

Save This Table



Normalization Tool

1NF to 3NF

Attributes

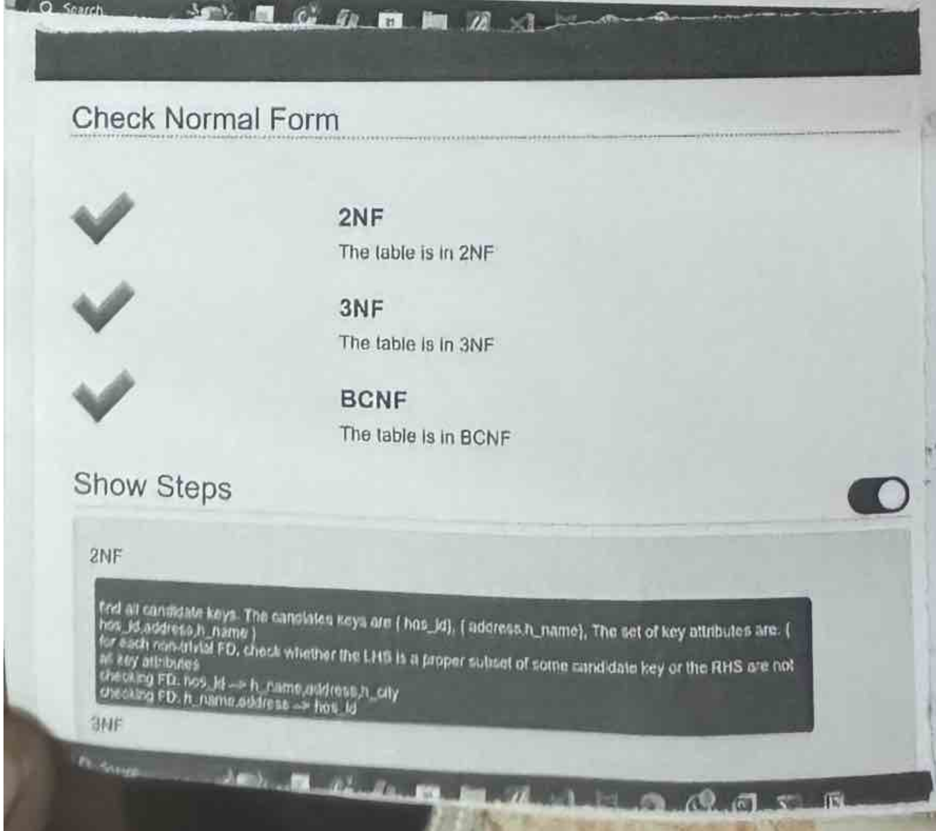
hos_id h_name address h_city

Functional Dependencies

```
hos_id → h_name
hos_id → address
hos_id → h_city
h_name address → hos_id
```

Show Steps

Table already in 3NF



Check Normal Form

✓ **2NF**
The table is in 2NF

✓ **3NF**
The table is in 3NF

✓ **BCNF**
The table is in BCNF

Show Steps

2NF

```
find all candidate keys. The candidates keys are { hos_id}, { address,h_name}, The set of key attributes are: {
hos_id,address,h_name }
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key or the RHS are not
all key attributes
checking FD: hos_id --> h_name,address,h_city
checking FD: h_name,address --> hos_id
```

3NF

```
claimId : 1001 },
{ set: { status : 'Approved', approvaldate : '2023-10-10' }
;
```

Delete :

```
db.claims.deleteOne({claimId : 1001 });
```

VEL TECH - CSE	
EX NO.	12
PERFORMANCE (5)	5
RESULT AND ANALYSIS (3)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	15
SIGN WITH DATE	[illegible]

VEL TECH	
EX NO.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	

Result : Thus the mini project is successfully verified and executed