DBMS MINI PROJECT
Title: Vehicle Insurance Management System

By:
Dhruv Thakur (PES2UG23CS175)
Dhruv Sudhan Naik (PES2UG23CS174)

Introduction

The CARINS system is a comprehensive database management system designed to streamline and organize vehicle insurance operations. It captures essential entities such as customers, policies, vehicles, claims, payments, agents, and service centers, using a well-structured Entity-Relationship (ER) model and relational schema to ensure data integrity, scalability, and efficient querying. The system enables automated tracking of policy renewals, claim processing, premium calculations, and customer interactions, facilitating seamless management of insurance workflows.

This system helps insurance companies analyze policy performance, evaluate claim patterns, generate detailed reports, and identify potential fraud or operational bottlenecks in claim settlement. As a DBMS project, it demonstrates practical skills in database design, data modeling, transaction management, and business process automation, offering a robust and real-world solution for managing comprehensive car insurance operations.

User Requirement Specification

Objectives

- Provide a centralized database to store insurance-related information including policies, claims, and customer data
- Enable automated policy renewal notifications and premium calculation
- Facilitate efficient claim processing and settlement tracking
- Offer tools for analyzing claim patterns, policy performance, and agent productivity

Functional Requirements

Customer Management

- Register and maintain customer profiles with personal and contact details
- Track customer policy history and claim records
- Assign unique customer IDs for identification

Policy Management

- Create and store policy details (policy number, type, coverage, premium, validity dates)
- Link policies to customers and vehicles
- Track policy status (active, expired, cancelled)
- Calculate premium based on vehicle type, age, and coverage

Vehicle Registration

- Record vehicle details (registration number, make, model, year, value)
- Associate vehicles with customers and policies

Claim Processing

- Log claim details (claim ID, date, type, amount, status)
- Track claim approval/rejection workflow
- Link claims to policies and customers
- Monitor claim settlement timelines

Payment Tracking

- Record premium payments with dates and amounts
- Track payment methods and transaction status
- Generate payment receipts and invoices

Agent Management

- Maintain agent profiles and commission records
- Track policies sold by each agent
- Monitor agent performance metrics

Reporting & Analytics

- Generate activity summaries (claims per month, policies issued)
- View policy renewal schedules and expiry notifications
- Analyze claim settlement rates and fraud patterns

Non-Functional Requirements

Performance

- The system should process policy searches and claim queries within seconds
- Support concurrent access for multiple users (agents, administrators, customers)

Scalability

- Must support thousands of customers, policies, and claims simultaneously
- Database design should accommodate growth in transaction volume

Security

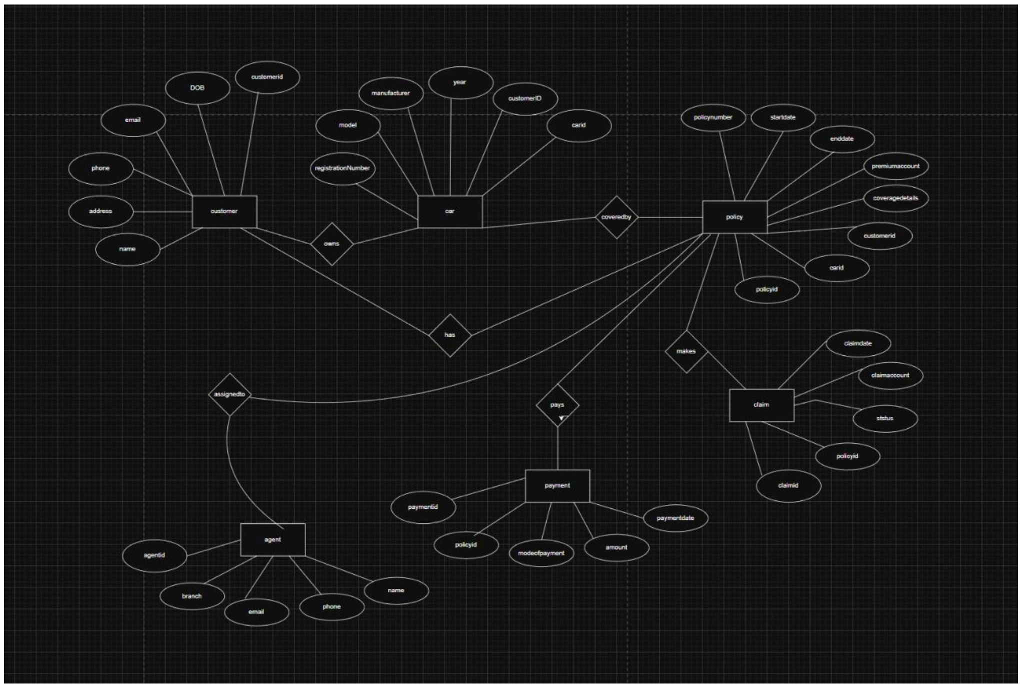
- Implement role-based access control (admin, agent, customer)
- Encrypt sensitive customer and payment information
- Maintain audit logs for all critical operations

Reliability

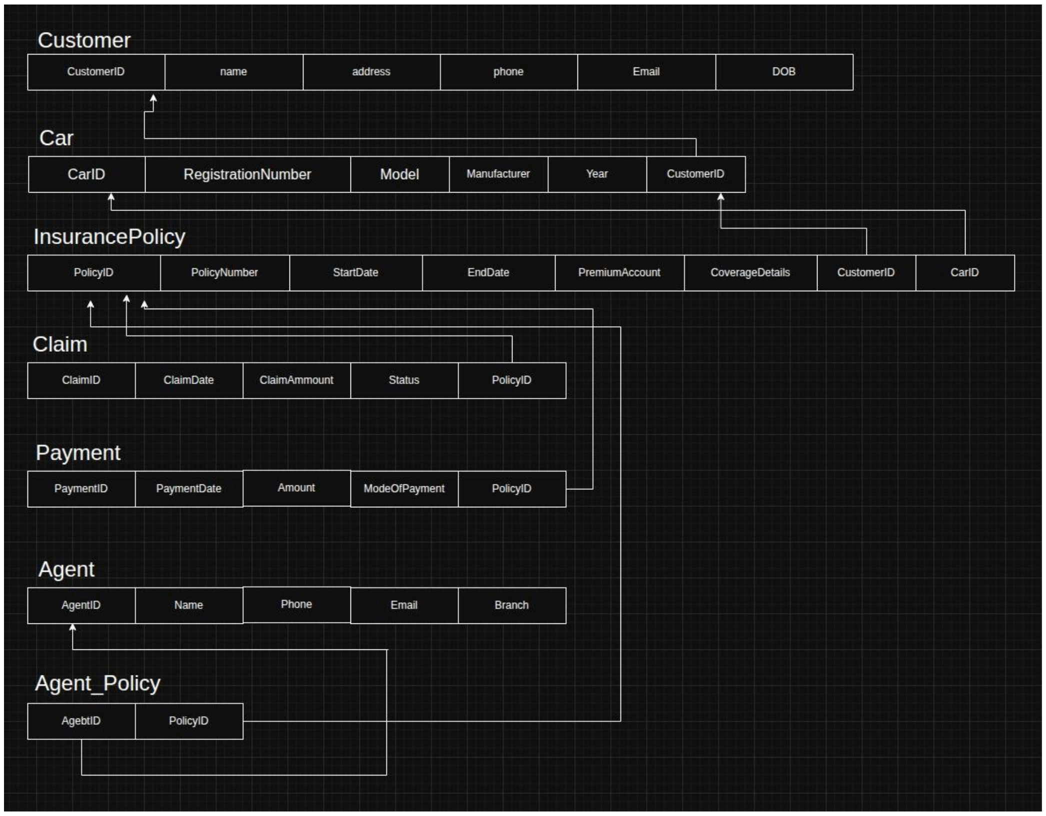
- Ensure data consistency through ACID properties
- Implement backup and recovery mechanisms

Category	**Tools/Technologies Used**
Frontend (UI)	HTML, CSS, JavaScript
Backend	Node.js
Database	MySQL, SQL CLI
Languages	SQL, Python
Version Control	GitHub

ER Diagram



ER Schema



DDL Commands

```
CREATE DATABASE carinsurancedb;
USE carinsurancedb;

-----------------------------------------------------------
-- 1. Table: agent
-----------------------------------------------------------
CREATE TABLE agent (
   agentID INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   phone VARCHAR(20),
   email VARCHAR(100),
   branch VARCHAR(100)
);

-----------------------------------------------------------
-- 2. Table: agent_archive
-----------------------------------------------------------
CREATE TABLE agent_archive (
   archiveID INT AUTO_INCREMENT PRIMARY KEY,
   agentID INT NOT NULL,
   name VARCHAR(100),
   deleted_at DATETIME,
   FOREIGN KEY (agentID) REFERENCES agent(agentID)
      ON DELETE CASCADE
);

-----------------------------------------------------------
-- 3. Table: customer
-----------------------------------------------------------
CREATE TABLE customer (
   customerID INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(100) NOT NULL,
   DOB DATE,
   address VARCHAR(255),
   phone VARCHAR(20),
   email VARCHAR(100)
);

-----------------------------------------------------------
-- 4. Table: car
```

```
----------------------------------------------------------
CREATE TABLE car (
   carID INT AUTO_INCREMENT PRIMARY KEY,
   registrationNumber VARCHAR(50) UNIQUE,
   model VARCHAR(100),
   manufacturer VARCHAR(100),
   year INT,
   customerID INT,
   FOREIGN KEY (customerID) REFERENCES customer(customerID)
      ON DELETE CASCADE
);

----------------------------------------------------------
-- 5. Table: policy
----------------------------------------------------------
CREATE TABLE policy (
   policyID INT AUTO_INCREMENT PRIMARY KEY,
   policyNumber VARCHAR(50) UNIQUE NOT NULL,
   startDate DATE,
   endDate DATE,
   premiumAmount DECIMAL(10,2),
   coverageDetails TEXT,
   customerID INT,
   carID INT,
   FOREIGN KEY (customerID) REFERENCES customer(customerID)
      ON DELETE CASCADE,
   FOREIGN KEY (carID) REFERENCES car(carID)
      ON DELETE CASCADE
);

----------------------------------------------------------
-- 6. Table: claim
----------------------------------------------------------
CREATE TABLE claim (
   claimID INT AUTO_INCREMENT PRIMARY KEY,
   claimDate DATE,
   claimAmount DECIMAL(10,2),
   status VARCHAR(50),
   policyID INT,
   FOREIGN KEY (policyID) REFERENCES policy(policyID)
      ON DELETE CASCADE
);

---------------------------------------------------------
-- 7. Table: payment
---------------------------------------------------------
CREATE TABLE payment (
    paymentID INT AUTO_INCREMENT PRIMARY KEY,
    paymentDate DATE,
    modeOfPayment VARCHAR(50),
    amount DECIMAL(10,2),
    policyID INT,
    FOREIGN KEY (policyID) REFERENCES policy(policyID)
        ON DELETE CASCADE
);

---------------------------------------------------------
-- 8. Table: assignedto
-- (agent assigned to customer)
---------------------------------------------------------
CREATE TABLE assignedto (
    agentID INT,
    customerID INT,
    PRIMARY KEY (agentID, customerID),
    FOREIGN KEY (agentID) REFERENCES agent(agentID)
        ON DELETE CASCADE,
    FOREIGN KEY (customerID) REFERENCES customer(customerID)
        ON DELETE CASCADE
);
```

1. agent

```
INSERT INTO agent (name, phone, email, branch) VALUES
('Rohit Sharma', '9876543210', 'rohit.agent@example.com', 'Bangalore'),
('Aditi Verma', '9123456780', 'aditi.verma@example.com', 'Mumbai'),
('Karan Mehta', '9811122233', 'karan.mehta@example.com', 'Delhi');
```

2. agent_archive

```
INSERT INTO agent_archive (agentID, name, deleted_at) VALUES
(1, 'Rohit Sharma', '2024-10-15 14:30:00');
```

3. customer

```
INSERT INTO customer (name, DOB, address, phone, email) VALUES
```

```
('Dhruv Thakur', '2004-07-15', 'Indiranagar, Bangalore', '9902651000',
'dhruv@example.com'),
('Sneha Kapoor', '1999-03-11', 'Andheri West, Mumbai', '9876001122',
'sneha.k@example.com'),
('Arjun Rao', '1988-12-05', 'Koramangala, Bangalore', '9988776655',
'arjunrao@example.com');
```

4. car

```
INSERT INTO car (registrationNumber, model, manufacturer, year, customerID)
VALUES
('KA03AB1234', 'i20', 'Hyundai', 2020, 1),
('MH12XY9876', 'Swift', 'Maruti Suzuki', 2018, 2),
('KA05MN4567', 'City', 'Honda', 2019, 3);
```

5. policy

```
INSERT INTO policy (policyNumber, startDate, endDate, premiumAmount,
coverageDetails, customerID, carID) VALUES
('POL2024001', '2024-01-01', '2025-01-01', 15000.00, 'Comprehensive Coverage', 1, 1),
('POL2024002', '2024-02-15', '2025-02-15', 13000.00, 'Third-Party Coverage', 2, 2),
('POL2024003', '2024-03-10', '2025-03-10', 18000.00, 'Zero Depreciation', 3, 3);
```

6. claim

```
INSERT INTO claim (claimDate, claimAmount, status, policyID) VALUES
('2024-06-20', 12000.00, 'Pending', 1),
('2024-07-05', 8000.00, 'Approved', 2),
('2024-08-15', 5000.00, 'Rejected', 3);
```

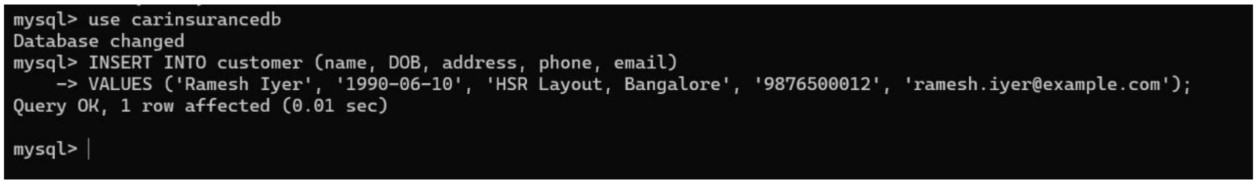
7. payment

```
INSERT INTO payment (paymentDate, modeOfPayment, amount, policyID) VALUES
('2024-01-01', 'Credit Card', 15000.00, 1),
('2024-02-15', 'UPI', 13000.00, 2),
('2024-03-10', 'Net Banking', 18000.00, 3);
```

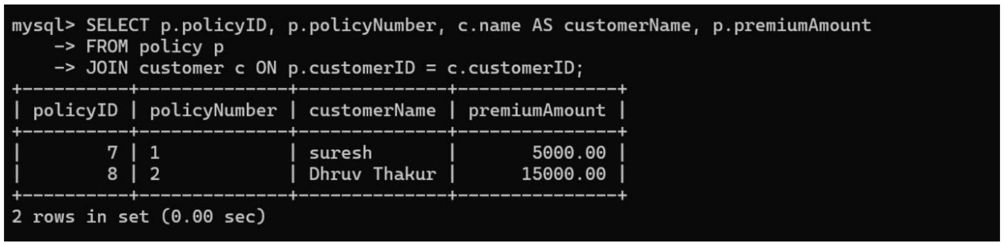
8. assignedto (Agent–Customer Mapping)

```
INSERT INTO assignedto (agentID, customerID) VALUES
(1, 1),
(1, 2),
(2, 3);
```

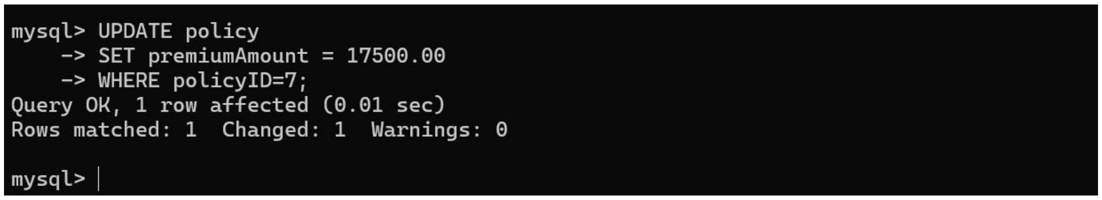
CRUD operation Screenshots



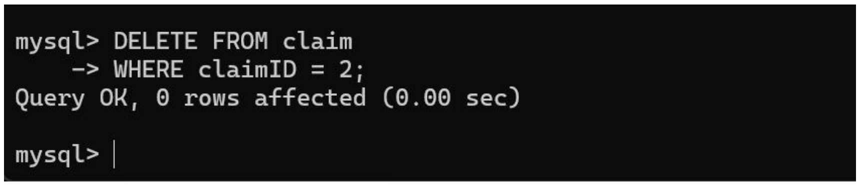
```
mysql> use carinsurancedb
Database changed
mysql> INSERT INTO customer (name, DOB, address, phone, email)
    -> VALUES ('Ramesh Iyer', '1990-06-10', 'HSR Layout, Bangalore', '9876500012', 'ramesh.iyer@example.com');
Query OK, 1 row affected (0.01 sec)

mysql>
```



```
mysql> SELECT p.policyID, p.policyNumber, c.name AS customerName, p.premiumAmount
    -> FROM policy p
    -> JOIN customer c ON p.customerID = c.customerID;
+----------+--------------+--------------+---------------+
| policyID | policyNumber | customerName | premiumAmount |
+----------+--------------+--------------+---------------+
|        7 | 1            | suresh       |       5000.00 |
|        8 | 2            | Dhruv Thakur |      15000.00 |
+----------+--------------+--------------+---------------+
2 rows in set (0.00 sec)
```

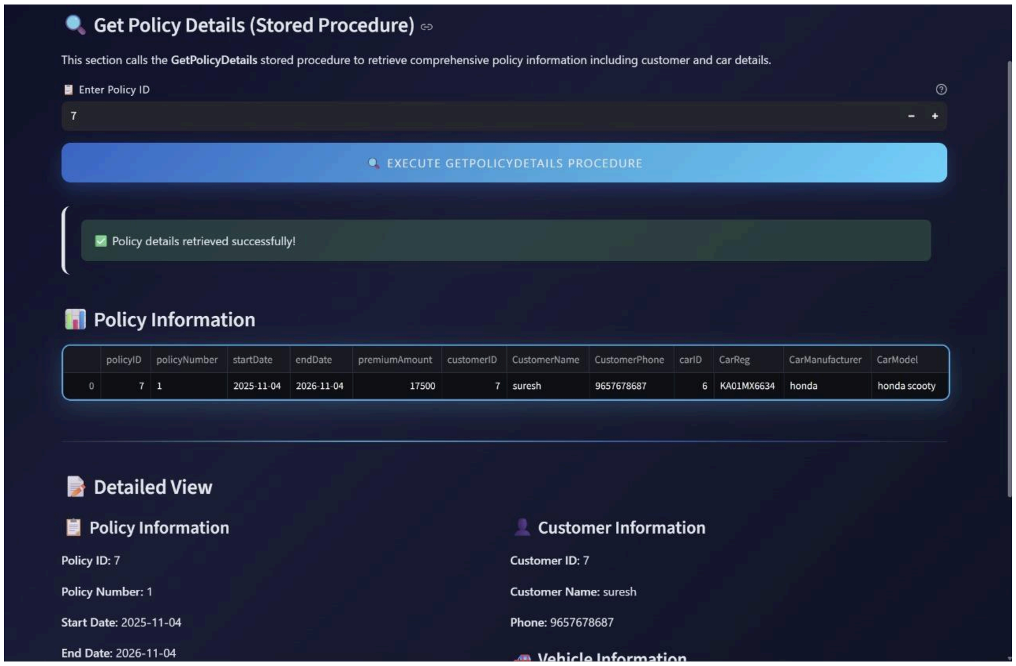


```
mysql> UPDATE policy
    -> SET premiumAmount = 17500.00
    -> WHERE policyID=7;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```



```
mysql> DELETE FROM claim
    -> WHERE claimID = 2;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Frontend


Get Policy Details (Stored Procedure)
Enter Policy ID
EXECUTE GETPOLICYDETAILS PROCEDURE

This section calls the **GetPolicyDetails** stored procedure to retrieve comprehensive policy information including customer and car details.

7

✅ Policy details retrieved successfully!

Policy Information

	policyID	policyNumber	startDate	endDate	premiumAmount	customerID	CustomerName	CustomerPhone	carID	CarReg	CarManufacturer	CarModel
0	7	1	2025-11-04	2026-11-04	17500	7	suresh	9657678687	6	KA01MX6634	honda	honda scooty

Detailed View

Policy Information

Policy ID: 7

Policy Number: 1

Start Date: 2025-11-04

End Date: 2026-11-04

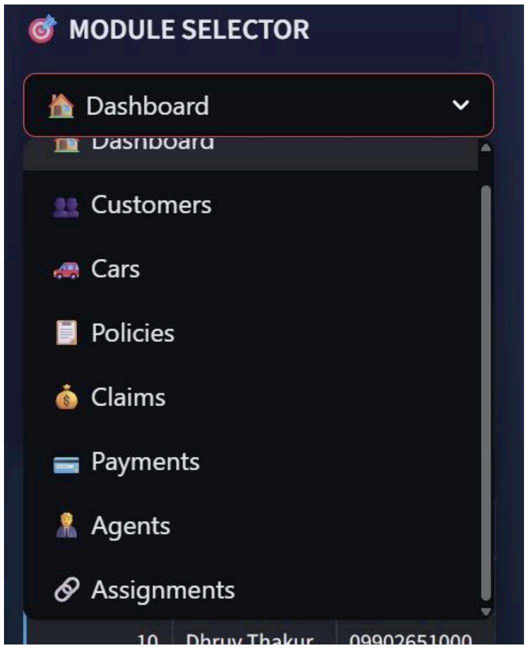
Customer Information

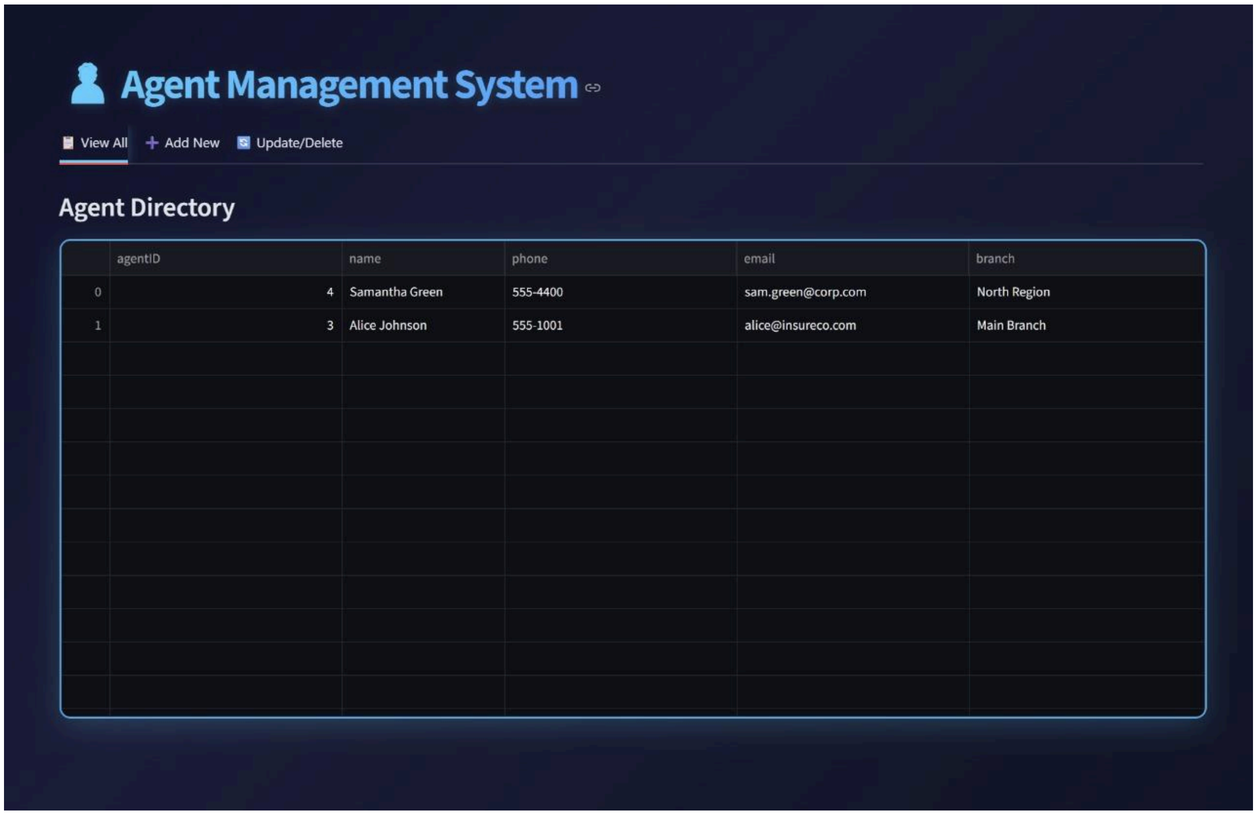
Customer ID: 7

Customer Name: suresh

Phone: 9657678687

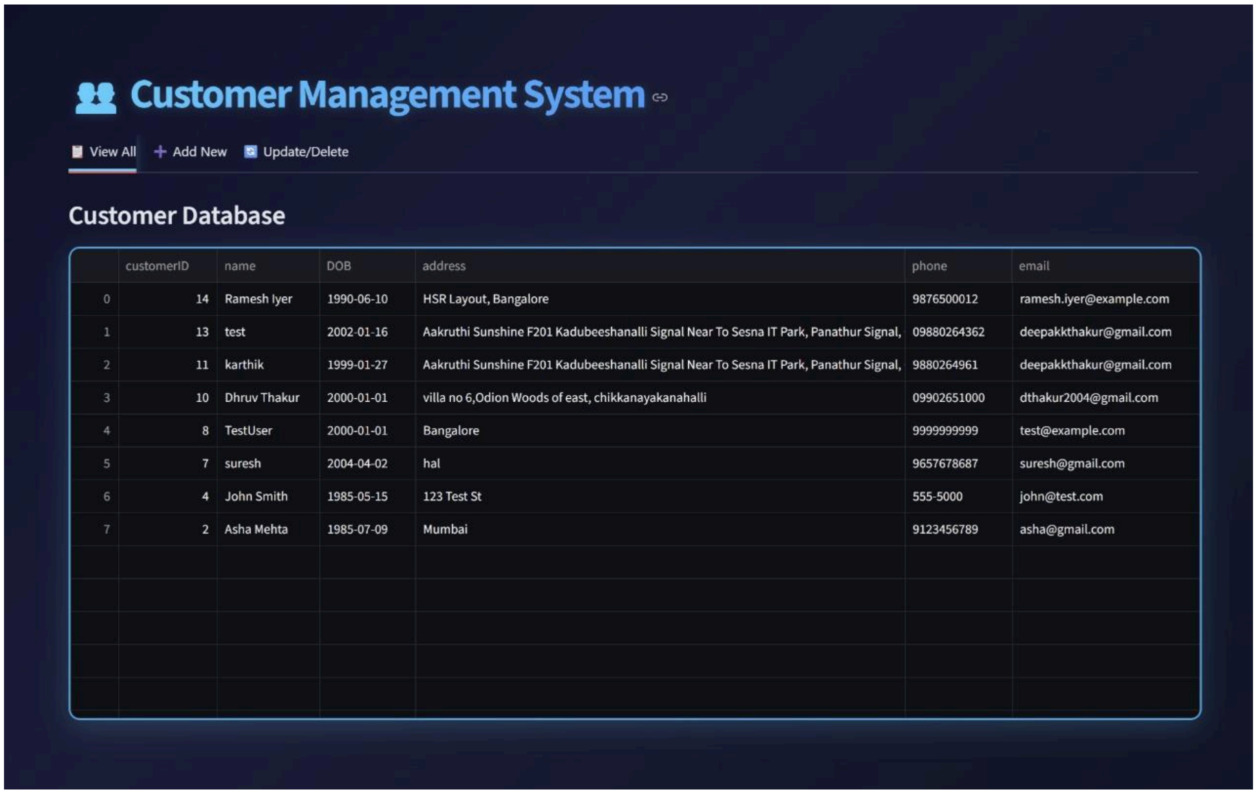
Vehicle Information


🎯 MODULE SELECTOR
🏠 Dashboard
🏠 Dashboard
👥 Customers
🚗 Cars
📋 Policies
💰 Claims
💳 Payments
👨‍💼 Agents
🔗 Assignments



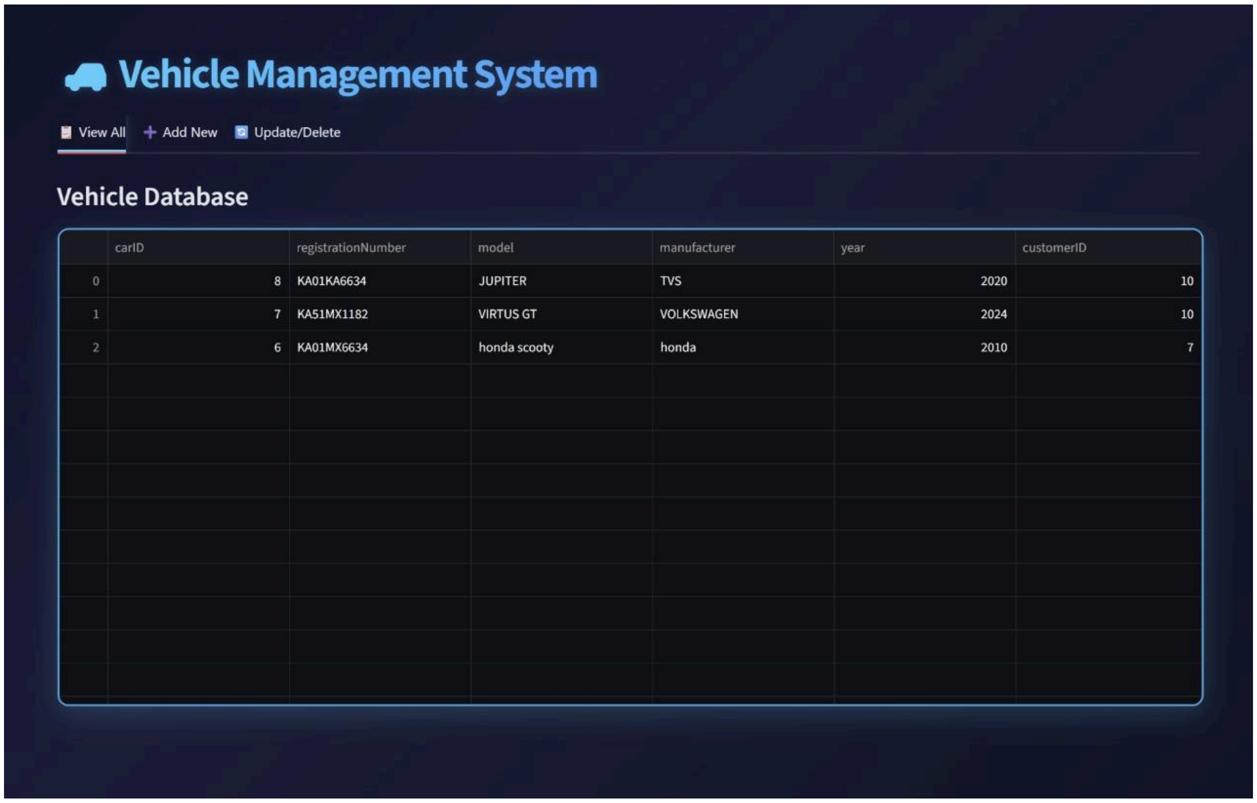
Agent Directory

	agentID	name	phone	email	branch
0	4	Samantha Green	555-4400	sam.green@corp.com	North Region
1	3	Alice Johnson	555-1001	alice@insureco.com	Main Branch



Customer Database

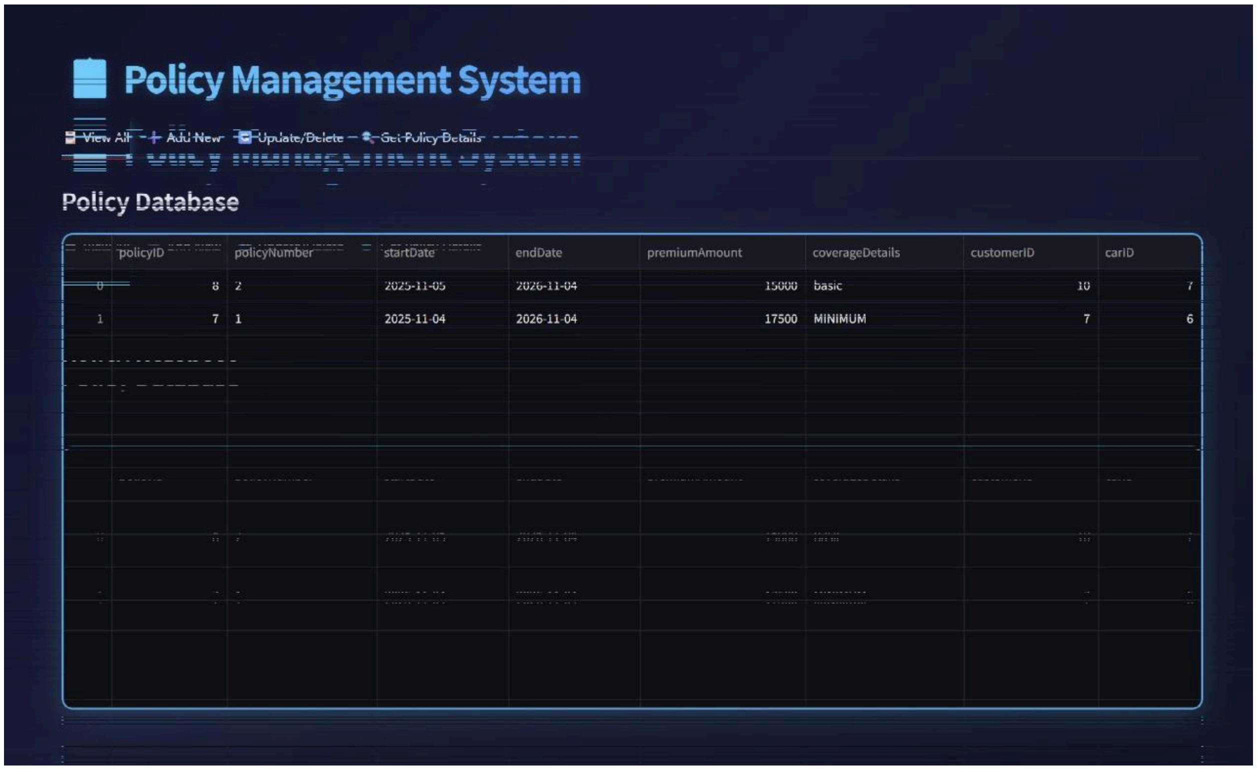
	customerID	name	DOB	address	phone	email
0	14	Ramesh Iyer	1990-06-10	HSR Layout, Bangalore	9876500012	ramesh.iyer@example.com
1	13	test	2002-01-16	Aakruthi Sunshine F201 Kadubeeshanalli Signal Near To Sesna IT Park, Panathur Signal,	09880264362	deepakkthakur@gmail.com
2	11	karthik	1999-01-27	Aakruthi Sunshine F201 Kadubeeshanalli Signal Near To Sesna IT Park, Panathur Signal,	9880264961	deepakkthakur@gmail.com
3	10	Dhruv Thakur	2000-01-01	villa no 6,Odion Woods of east, chikkanayakanahalli	09902651000	dthakur2004@gmail.com
4	8	TestUser	2000-01-01	Bangalore	9999999999	test@example.com
5	7	suresh	2004-04-02	hal	9657678687	suresh@gmail.com
6	4	John Smith	1985-05-15	123 Test St	555-5000	john@test.com
7	2	Asha Mehta	1985-07-09	Mumbai	9123456789	asha@gmail.com



Vehicle Management System

Vehicle Database

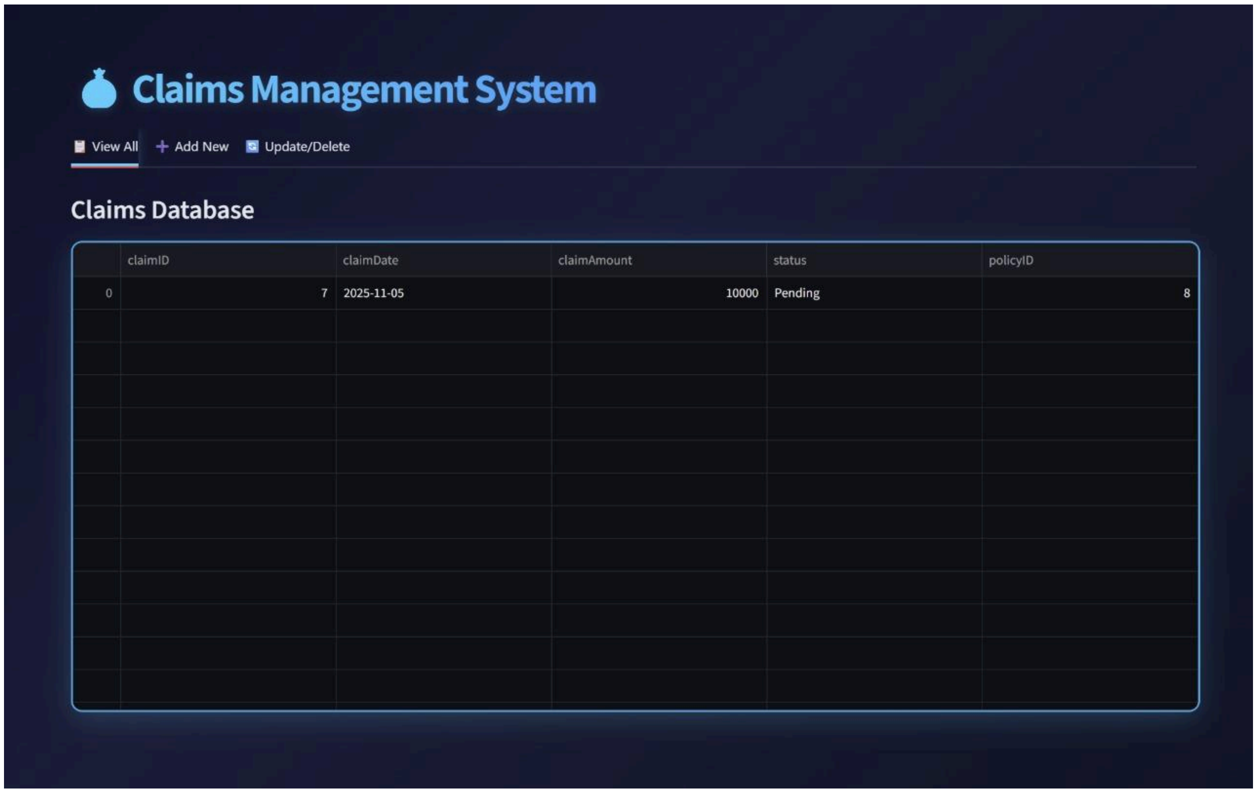
	carID	registrationNumber	model	manufacturer	year	customerID
0	8	KA01KA6634	JUPITER	TVS	2020	10
1	7	KA51MX1182	VIRTUS GT	VOLKSWAGEN	2024	10
2	6	KA01MX6634	honda scooty	honda	2010	7



Policy Management System

Policy Database

	policyID	policyNumber	startDate	endDate	premiumAmount	coverageDetails	customerID	carID
0	8	2	2025-11-05	2026-11-04	15000	basic	10	7
1	7	1	2025-11-04	2026-11-04	17500	MINIMUM	7	6

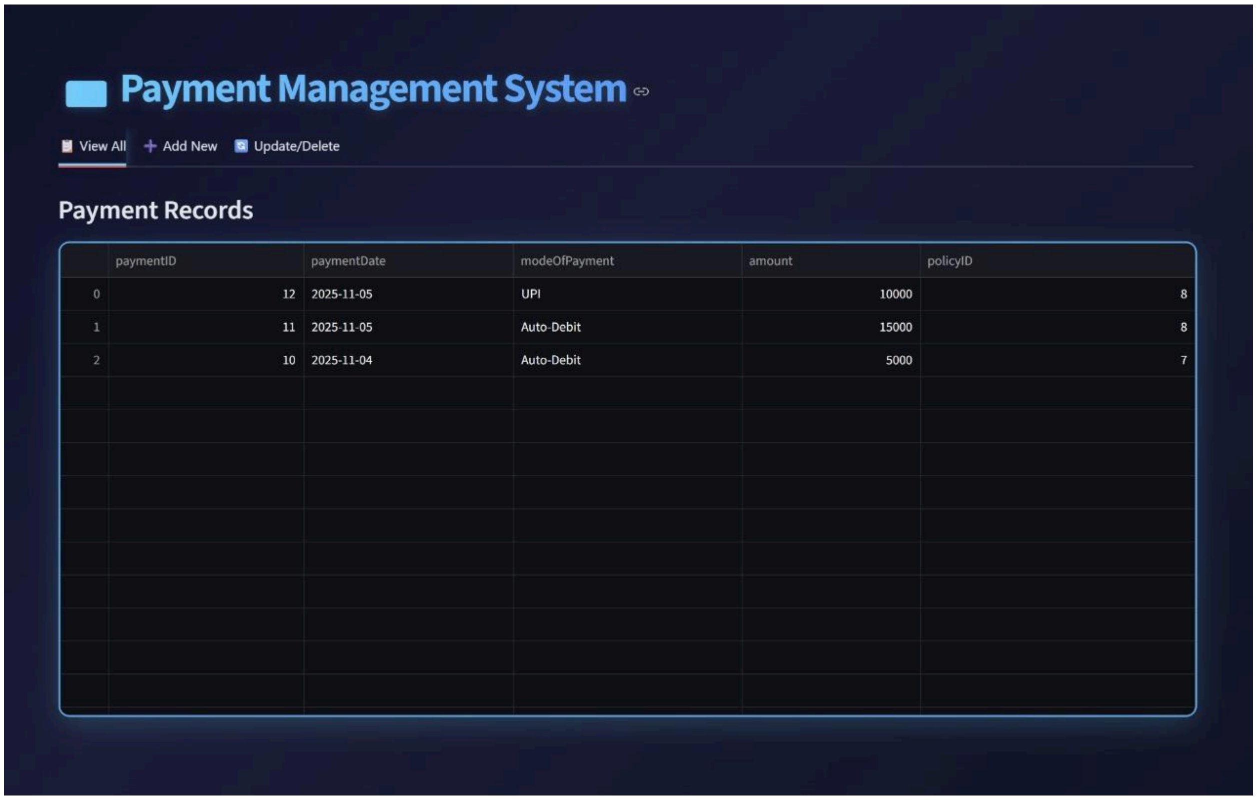


Claims Management System

View All | Add New | Update/Delete

Claims Database

	claimID	claimDate	claimAmount	status	policyID
0	7	2025-11-05	10000	Pending	8

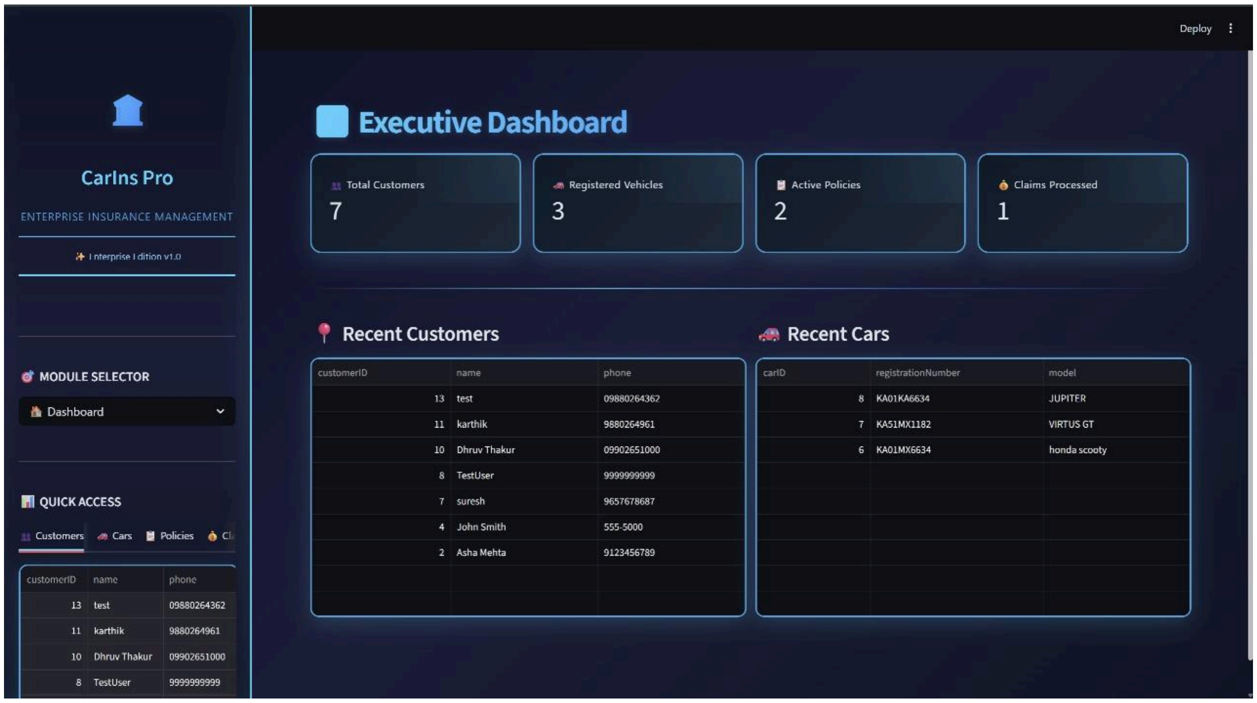


Payment Management System

View All | Add New | Update/Delete

Payment Records

	paymentID	paymentDate	modeOfPayment	amount	policyID
0	12	2025-11-05	UPI	10000	8
1	11	2025-11-05	Auto-Debit	15000	8
2	10	2025-11-04	Auto-Debit	5000	7



Deploy

CarIns Pro

ENTERPRISE INSURANCE MANAGEMENT

✨ Enterprise Edition v1.0

MODULE SELECTOR

Dashboard

QUICK ACCESS

Customers | Cars | Policies | Cl

customerID	name	phone
13	test	09880264362
11	karthik	9880264961
10	Dhruv Thakur	09902651000
8	TestUser	9999999999

Executive Dashboard

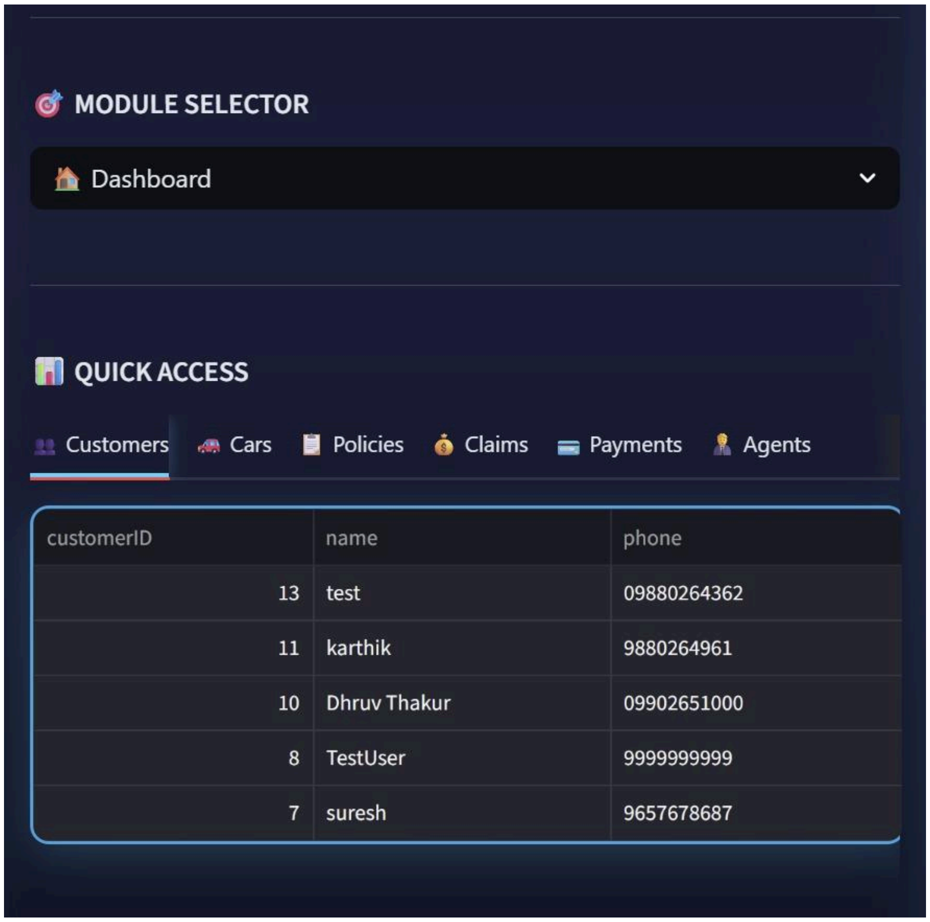
Total Customers	Registered Vehicles	Active Policies	Claims Processed
7	3	2	1

Recent Customers

customerID	name	phone
13	test	09880264362
11	karthik	9880264961
10	Dhruv Thakur	09902651000
8	TestUser	9999999999
7	suresh	9657678687
4	John Smith	555-5000
2	Asha Mehta	9123456789

Recent Cars

carID	registrationNumber	model
8	KA01KA6634	JUPITER
7	KA51MX1182	VIRTUS GT
6	KA01MX6634	honda scooty



MODULE SELECTOR

🏠 Dashboard

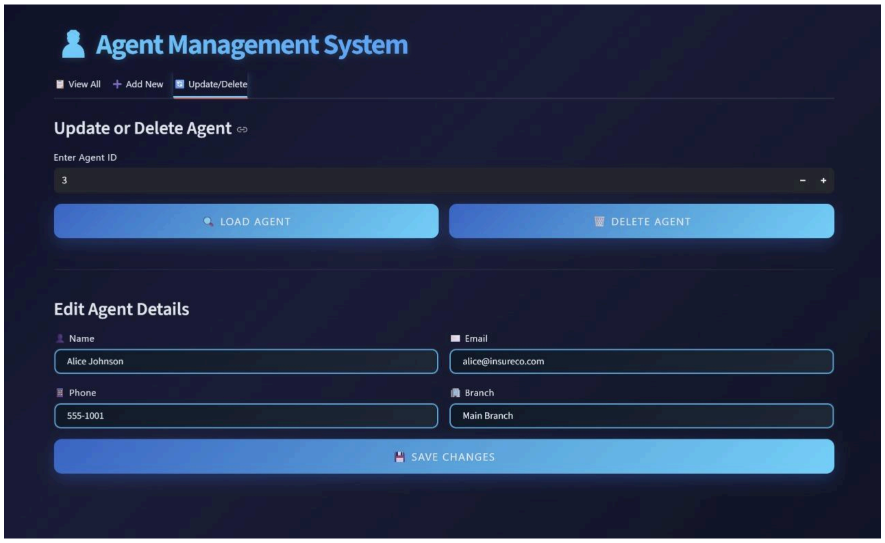
QUICK ACCESS

Customers | Cars | Policies | Claims | Payments | Agents

customerID	name	phone
13	test	09880264362
11	karthik	9880264961
10	Dhruv Thakur	09902651000
8	TestUser	9999999999
7	suresh	9657678687

Triggers, Procedures/Functions, Nested query, Join, Aggregate queries

Triggers


Agent Management System

Update or Delete Agent

Enter Agent ID: 3

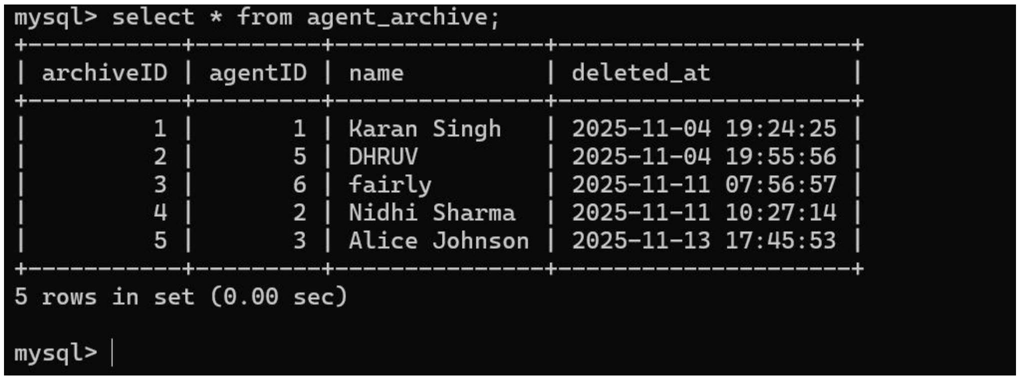
Edit Agent Details

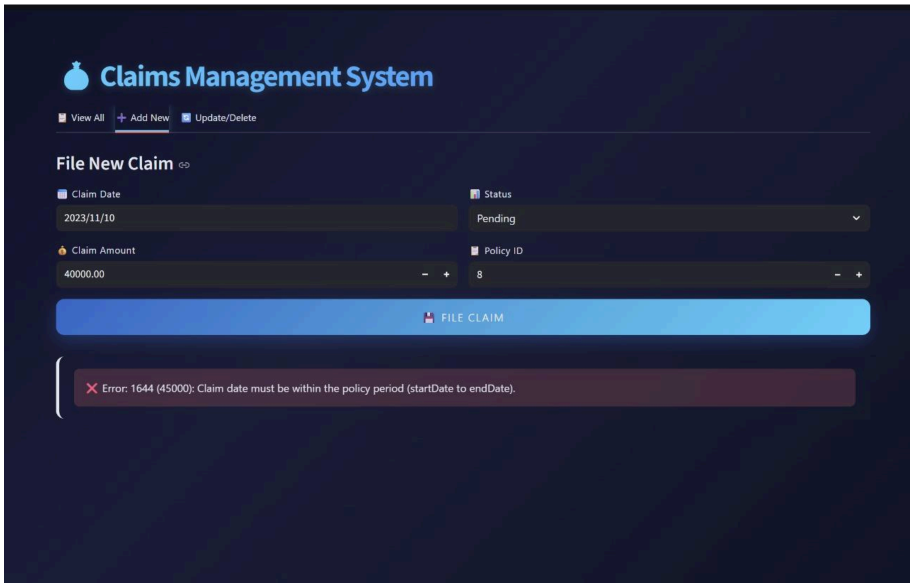
Name: Alice Johnson

Email: alice@insureco.com

Phone: 555-1001

Branch: Main Branch


```
mysql> select * from agent_archive;
+-----------+---------+---------------+---------------------+
| archiveID | agentID | name          | deleted_at          |
+-----------+---------+---------------+---------------------+
|         1 |       1 | Karan Singh   | 2025-11-04 19:24:25 |
|         2 |       5 | DHRUV         | 2025-11-04 19:55:56 |
|         3 |       6 | fairly        | 2025-11-11 07:56:57 |
|         4 |       2 | Nidhi Sharma  | 2025-11-11 10:27:14 |
|         5 |       3 | Alice Johnson | 2025-11-13 17:45:53 |
+-----------+---------+---------------+---------------------+
5 rows in set (0.00 sec)

mysql> |
```


Claims Management System

File New Claim

Claim Date: 2023/11/10

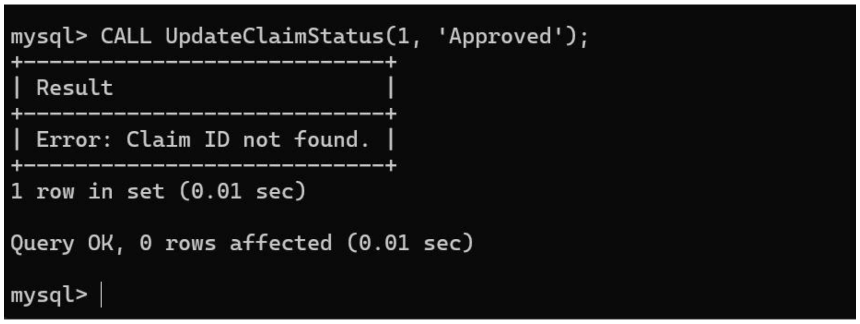
Status: Pending

Claim Amount: 40000.00

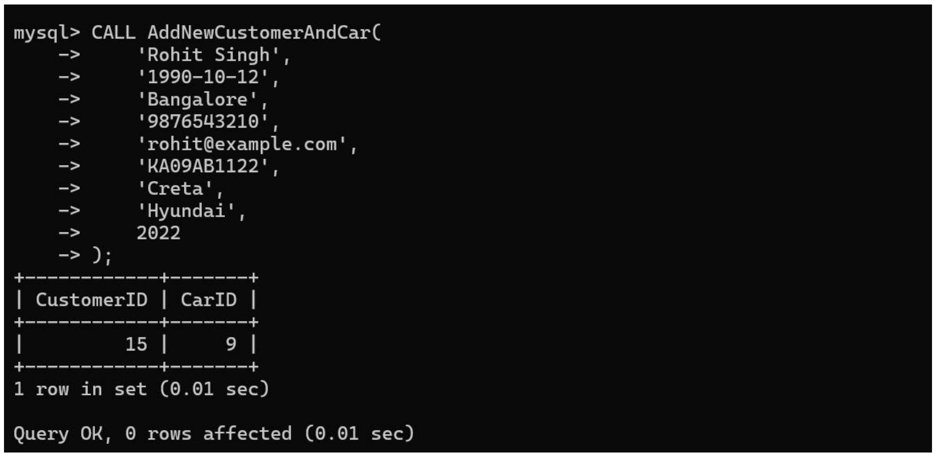
Policy ID: 8

Error: 1644 (45000): Claim date must be within the policy period (startDate to endDate).

Procedure

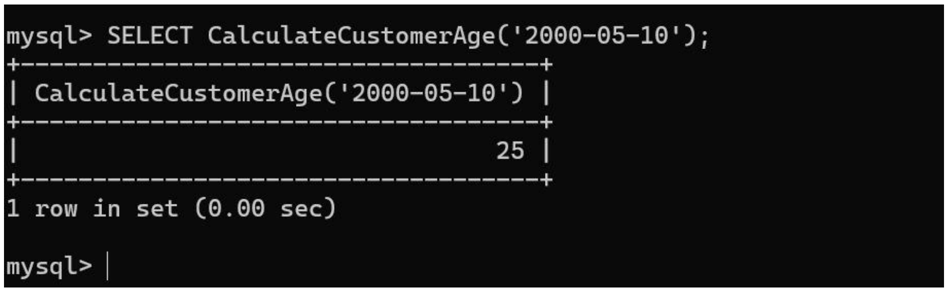


```
mysql> CALL UpdateClaimStatus(1, 'Approved');
+-----------------------------+
| Result                      |
+-----------------------------+
| Error: Claim ID not found. |
+-----------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

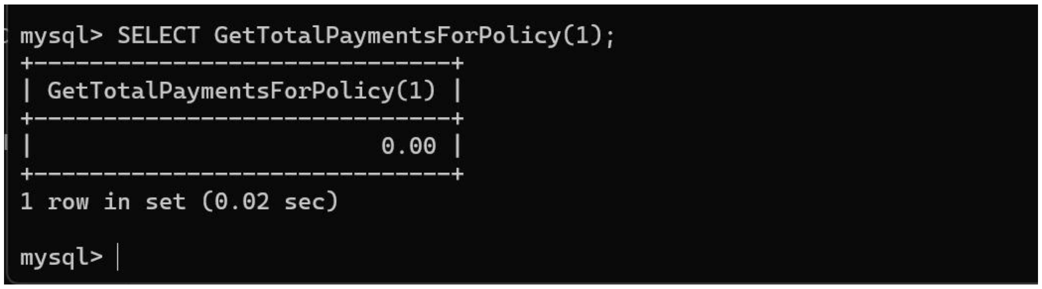


```
mysql> CALL AddNewCustomerAndCar(
    ->     'Rohit Singh',
    ->     '1990-10-12',
    ->     'Bangalore',
    ->     '9876543210',
    ->     'rohit@example.com',
    ->     'KA09AB1122',
    ->     'Creta',
    ->     'Hyundai',
    ->     2022
    -> );
+------------+-------+
| CustomerID | CarID |
+------------+-------+
|         15 |     9 |
+------------+-------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

Function

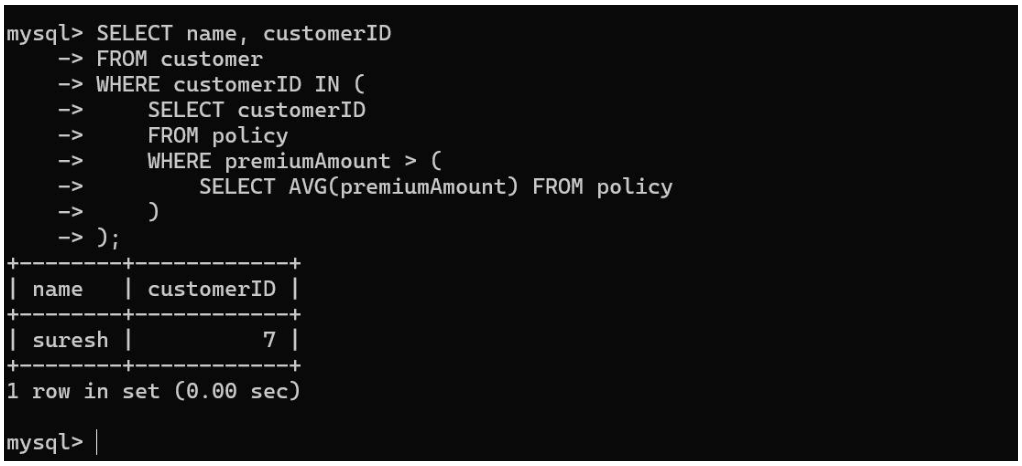


```
mysql> SELECT CalculateCustomerAge('2000-05-10');
+------------------------------------+
| CalculateCustomerAge('2000-05-10') |
+------------------------------------+
|                                 25 |
+------------------------------------+
1 row in set (0.00 sec)

mysql>
```



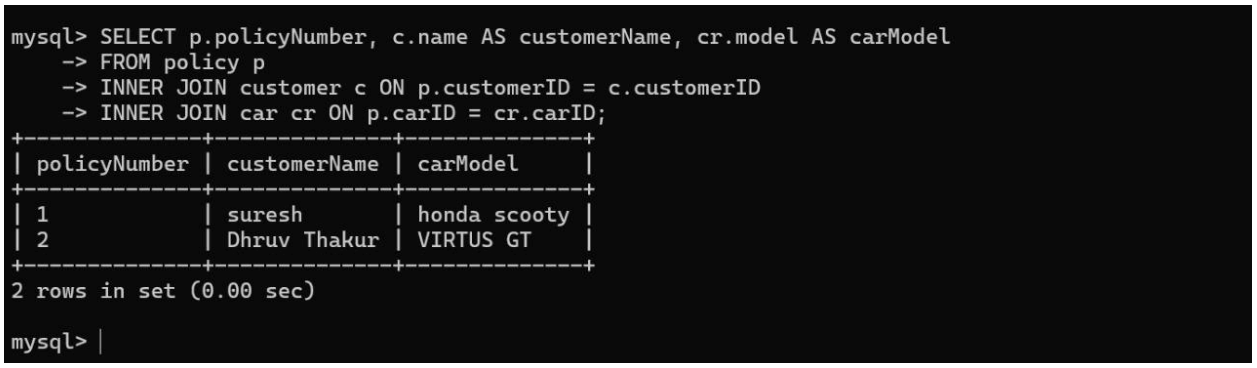
```
mysql> SELECT GetTotalPaymentsForPolicy(1);
+------------------------------+
| GetTotalPaymentsForPolicy(1) |
+------------------------------+
|                         0.00 |
+------------------------------+
1 row in set (0.02 sec)

mysql>
```

Nested Query



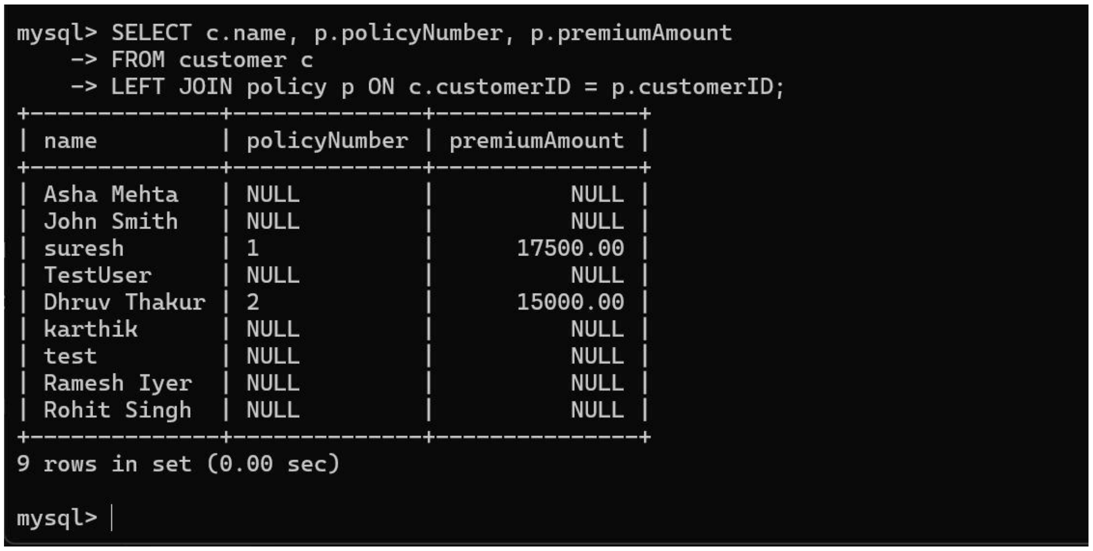
```
mysql> SELECT name, customerID
    -> FROM customer
    -> WHERE customerID IN (
    ->     SELECT customerID
    ->     FROM policy
    ->     WHERE premiumAmount > (
    ->         SELECT AVG(premiumAmount) FROM policy
    ->     )
    -> );
+--------+------------+
| name   | customerID |
+--------+------------+
| suresh |          7 |
+--------+------------+
1 row in set (0.00 sec)

mysql> 
```

Left Join



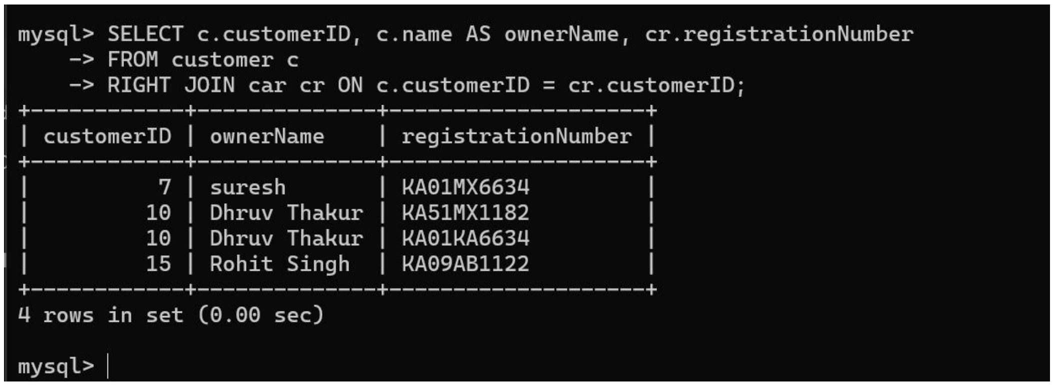
```
mysql> SELECT p.policyNumber, c.name AS customerName, cr.model AS carModel
    -> FROM policy p
    -> INNER JOIN customer c ON p.customerID = c.customerID
    -> INNER JOIN car cr ON p.carID = cr.carID;
+--------------+--------------+--------------+
| policyNumber | customerName | carModel     |
+--------------+--------------+--------------+
| 1            | suresh       | honda scooty |
| 2            | Dhruv Thakur | VIRTUS GT    |
+--------------+--------------+--------------+
2 rows in set (0.00 sec)

mysql> 
```

Right Join



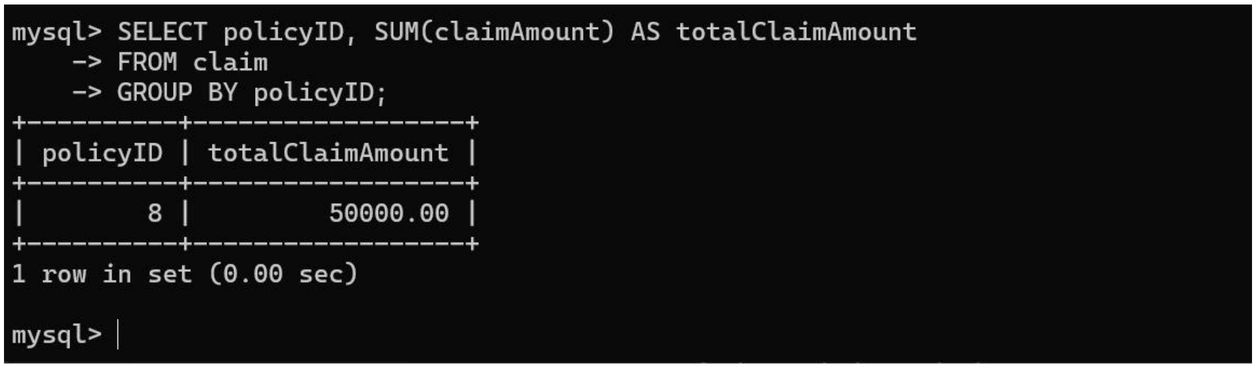
```
mysql> SELECT c.name, p.policyNumber, p.premiumAmount
    -> FROM customer c
    -> LEFT JOIN policy p ON c.customerID = p.customerID;
+--------------+--------------+---------------+
| name         | policyNumber | premiumAmount |
+--------------+--------------+---------------+
| Asha Mehta   | NULL         |          NULL |
| John Smith   | NULL         |          NULL |
| suresh       | 1            |      17500.00 |
| TestUser     | NULL         |          NULL |
| Dhruv Thakur | 2            |      15000.00 |
| karthik      | NULL         |          NULL |
| test         | NULL         |          NULL |
| Ramesh Iyer  | NULL         |          NULL |
| Rohit Singh  | NULL         |          NULL |
+--------------+--------------+---------------+
9 rows in set (0.00 sec)

mysql> 
```

Inner Join



```
mysql> SELECT c.customerID, c.name AS ownerName, cr.registrationNumber
    -> FROM customer c
    -> RIGHT JOIN car cr ON c.customerID = cr.customerID;
+------------+--------------+--------------------+
| customerID | ownerName    | registrationNumber |
+------------+--------------+--------------------+
|          7 | suresh       | KA01MX6634         |
|         10 | Dhruv Thakur | KA51MX1182         |
|         10 | Dhruv Thakur | KA01KA6634         |
|         15 | Rohit Singh  | KA09AB1122         |
+------------+--------------+--------------------+
4 rows in set (0.00 sec)

mysql>
```

Aggregate



```
mysql> SELECT policyID, SUM(claimAmount) AS totalClaimAmount
    -> FROM claim
    -> GROUP BY policyID;
+----------+------------------+
| policyID | totalClaimAmount |
+----------+------------------+
|        8 |         50000.00 |
+----------+------------------+
1 row in set (0.00 sec)

mysql>
```