



Database System Mini Project

A report submitted to the

Department of Computer Engineering

Faculty of Engineering

University of Ruhuna Sri Lanka

On 11th September 2023

In completing an assignment for the module

EE4350 Database Systems

By

Group 55

Group members

Vindyani K.A.C.H. – EG/2020/4253

Virajani M.Y. – EG/2020/4254

Vithusan B. – EG/2020/4256

Content of Tables

Chapter 1 - Requirement Analysis	5
1.1 Functional requirements.....	5
1.2 Data requirements.....	6
Chapter 2 - Conceptual Design.....	8
Chapter 3 - Logical Design.....	9
3.1 First Normal Form.....	9
3.2 Second Normal Form	10
Chapter 4 – Implementation	10
4.1 Creating Schema with tables.	10
4.2 Demonstrating the Schema and Tables	12
4.3 Adding Constraints for Created Tables	13
4.4 Creating a trigger for the derived attribute “Age.”	14
4.5 Table Definitions	15
4.6 Inserting Data for Tables	21
4.7 Inserting, Updating and Deleting Operations for Existing Tables.	28
Chapter 5 – Transactions	42
5.1 Simple queries.....	42
5.2 Complex queries.....	46
Chapter 6 - Database Tuning.....	53

List of Figures

Figure 1: ER Diagram for Insurance Database System.....	8
Figure 2: The conceptual data model to representational data model mapping(1NF)	9

List of Tables

Table 1: Entity and Attributes **Error! Bookmark not defined.**

RELATIONAL DATABASE

Chapter 1 - Requirement Analysis

1.1 Functional requirements

Functional requirements, which are the actions that can be performed by the database system, are referred to as such. The goal at the end of this project is to ensure that the tasks provided can be carried out by the database, and this involves the assignment of performance and other limiting requirements to all functional levels. These include:

1. Data retrieval
2. The possibility of modifying the database content
3. The capability to fill possible data gaps or missing data
4. The ability to modify previously entered data in the database
5. The capacity to add comments on data when the data is missing or questionable
6. The capability to supply data with improved performance

This schema was established to represent an Insurance System database. Within this insurance schema, appropriate entities have been selected to represent the schema. Customers, Agents, Policy, Life Insurance, Vehicle Insurance, Property Insurance, and Payments are the areas covered by this schema. The data distributed across these areas can be accessed using this database. Additionally, the aforementioned functional requirements are fulfilled by this schema.

1.2 Data requirements

Table 0-1: Entities and Attributes

Entity	Attributes
Customer	Customer_ID (Not Null)
	First Name
	Last Name
	DOB
	Age
	Street
	City
	District
	Province
Agent	Contact No
	Agent_ID (Not Null)
	First Name
	Last Name
	Email
	Street
	City
	District
	Province
Policy	Contact No
	Policy_NO (Not null)
	Policy Name
	Policy Type
	Coverage
	Issued Date
Vehicle	Term Price
	Vehicle_ID(Not null)
	Policy Type
	Started Date
	End date

Car	Car Number(Not Null)
	Make
	Model
Bike	Bike Number(Not Null)
	Make
	Model
Property	Property_ID (Not Null)
	Cost
	Property Type
	Street
	City
	District
	Province
Life	Beneficiary Number (Not Null)
	First Name
	Last Name
	Relationship
	Percentage share
Accident	Report Number(Not Null)
	Date
	Time
	Venue
Payment	Date(Not Null)
	Payment method
	Amount

Chapter 2 - Conceptual Design

The Entity - Relationship (ER) diagram of the Ins_prj schema is drawn below using an appropriate tool according to the given instructions.

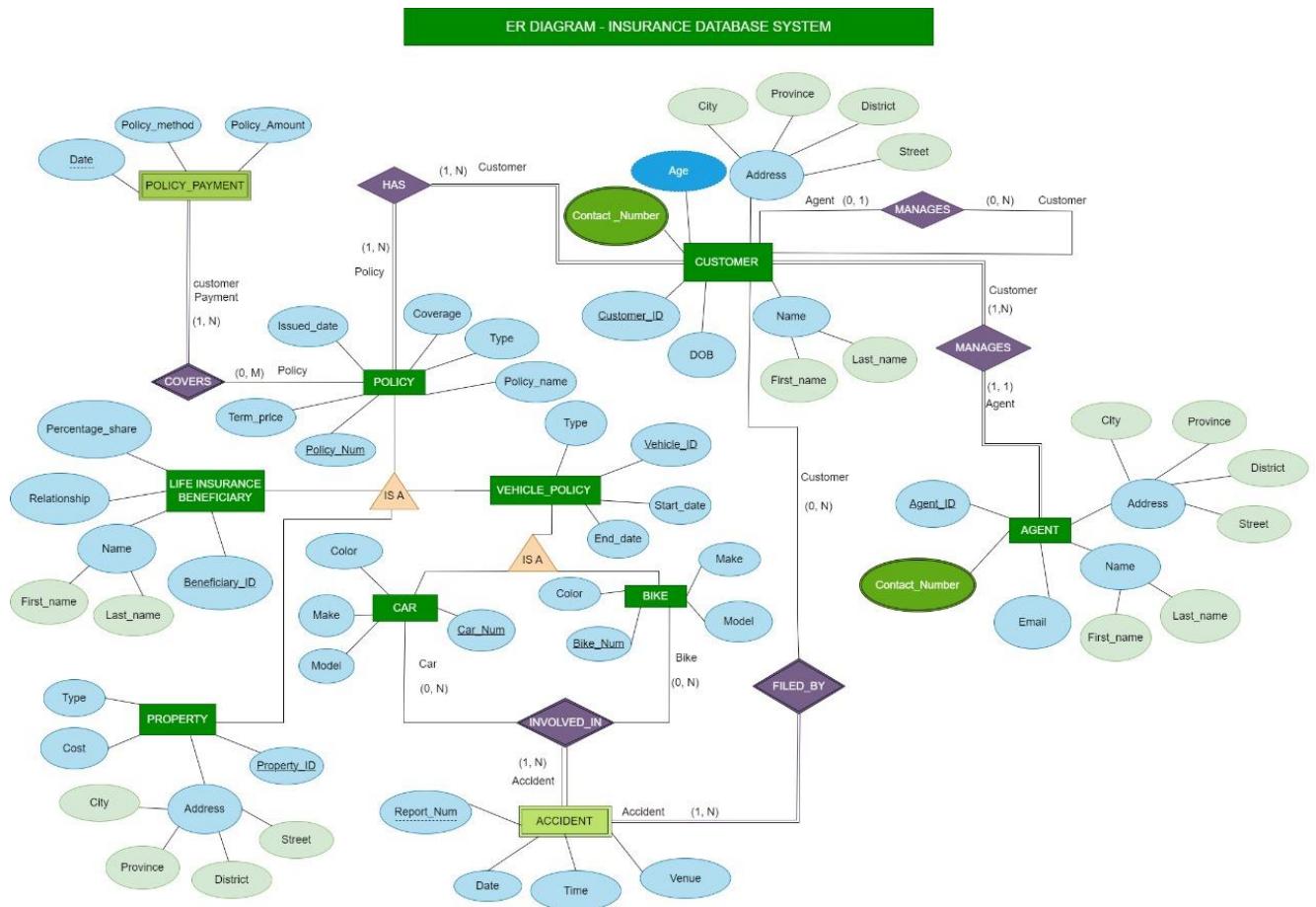


Figure 1: ER Diagram for Insurance Database System

Whether an attribute can be null or not is determined by the nature of the schema. Data that are not null are mentioned in the list above.

Brief analysis of some important relationships of this schema was done before creating ER diagram. They are,

1. Customer-Agent Relationship:
 - Customers can be associated with one or more agents.
 - Agents can serve multiple customers.
2. Customer-Policy Relationship:
 - Customers can have one or more insurance policies.
 - Each insurance policy is associated with a single customer.

3. Policy-Insurance Type Relationship:

- Insurance policies can be of different types such as Life Insurance, Vehicle Insurance, and Property Insurance.
- Each policy is linked to a specific insurance type.

4. Policy-Payment Relationship:

- Insurance policies involve payments, possibly premium payments.
- Each policy can have multiple payment records associated with it.

Chapter 3 - Logical Design

3.1 First Normal Form

ER diagram is mapped to the relational model in first normal by creating a separate table for “Contact_Numerbs” which is a multivalued attribute. Also, primary keys are highlighted because in 1NF it should be identified out of multiple candidate keys. All Other important keys, functional dependencies and referential integrities are represented by usual notation.

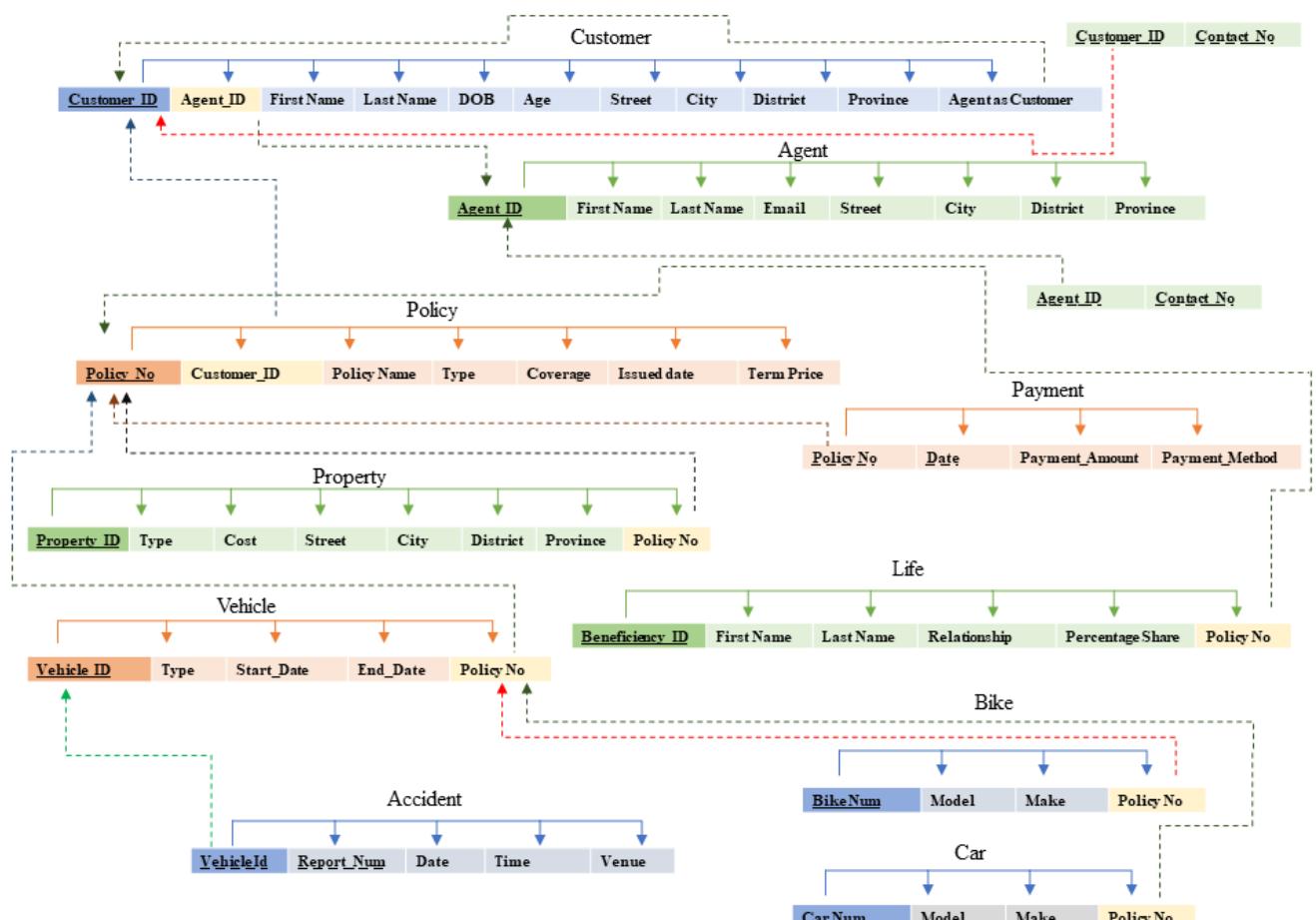


Figure 2: The conceptual data model to representational data model mapping(1NF)

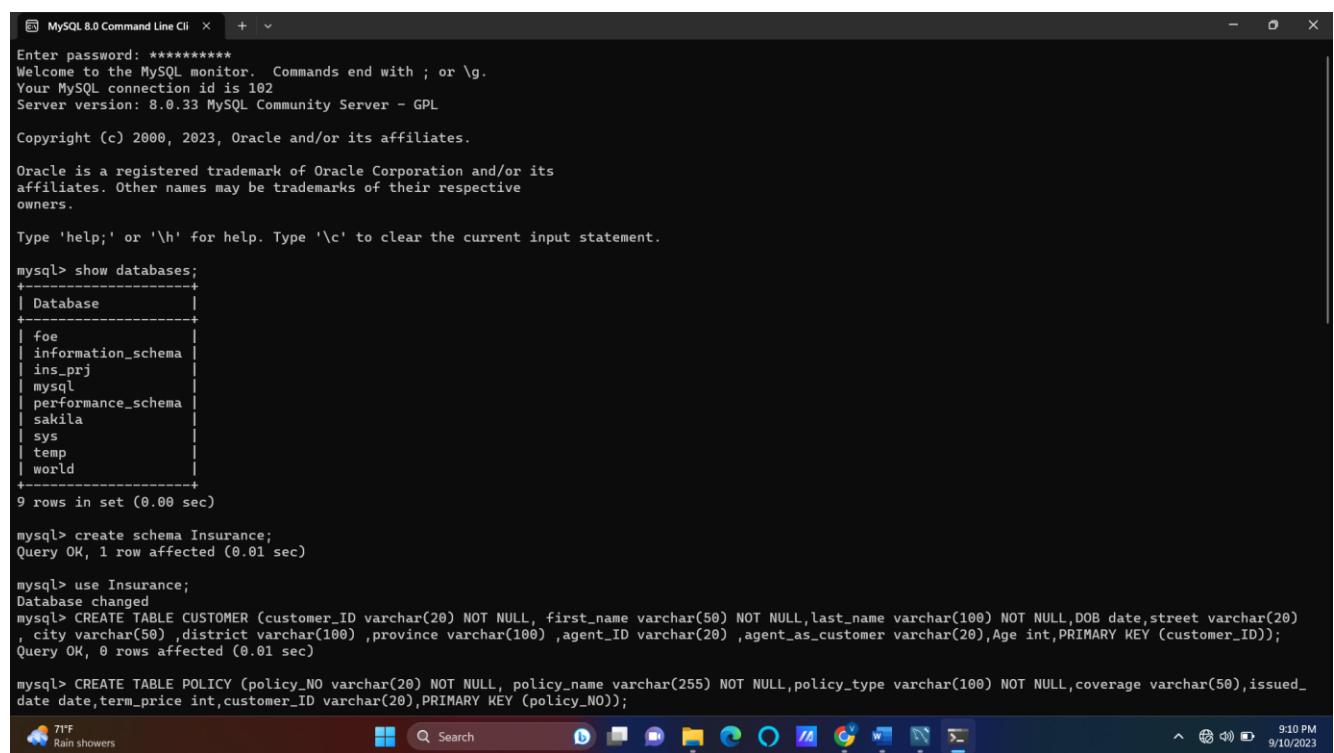
3.2 Second Normal Form

Based on Figure 2, it is observed that there are no partial functional dependencies in the transformation from the conceptual data model to the representational data model (1NF). Consequently, every non-prime attribute is fully functional dependency on any prime attribute. Due to this, the relational model depicted in Figure 2 is already in 2NF.

Chapter 4 – Implementation

In this section 2NF of the representational data model is implemented using MYSQL client and the MYSQL workbench. Checking currently available schemas, creating a new schema, defining tables and other important operations are included in the below screenshots.

4.1 Creating Schema with tables.



```
MySQL 8.0 Command Line Cli  +  X
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 102
Server version: 8.0.33 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| foo      |
| information_schema |
| ins_prj   |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| temp     |
| world   |
+-----+
9 rows in set (0.00 sec)

mysql> create schema Insurance;
Query OK, 1 row affected (0.01 sec)

mysql> use Insurance;
Database changed
mysql> CREATE TABLE CUSTOMER (customer_ID varchar(20) NOT NULL, first_name varchar(50) NOT NULL, last_name varchar(100) NOT NULL, DOB date, street varchar(20)
, city varchar(50), district varchar(100), province varchar(100), agent_ID varchar(20), agent_as_customer varchar(20), Age int, PRIMARY KEY (customer_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE POLICY (policy_NO varchar(20) NOT NULL, policy_name varchar(255) NOT NULL, policy_type varchar(100) NOT NULL, coverage varchar(50), issued_
date date, term_price int, customer_ID varchar(20), PRIMARY KEY (policy_NO));
Query OK, 0 rows affected (0.01 sec)
```

```
MySQL 8.0 Command Line Cli × + ▾
mysql> CREATE TABLE POLICY (policy_NO varchar(20) NOT NULL, policy_name varchar(255) NOT NULL,policy_type varchar(100) NOT NULL,coverage varchar(50),issued_date date,term_price int, customer_ID varchar(20),PRIMARY KEY (policy_NO));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE VEHICLE (vehicle_ID varchar(20) NOT NULL, policy_type varchar(100) NOT NULL,start_date date,end_date date,policy_NO varchar(20),PRIMARY KEY (vehicle_ID));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE LIFE (beneficiary_ID varchar(20) NOT NULL, first_name varchar(50) NOT NULL, last_name varchar(100) NOT NULL,relationship varchar(50),pece_nage_share int,policy_NO varchar(20),PRIMARY KEY (beneficiary_ID));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE PROPERTY (property_ID varchar(20) NOT NULL, cost int,property_type varchar(20),street varchar(20) , city varchar(50) ,district varchar(100) ,province varchar(100) ,policy_NO varchar(20),PRIMARY KEY (property_ID));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE AGENT (agent_ID varchar(20) NOT NULL,first_name varchar(50) NOT NULL,last_name varchar(100) NOT NULL,email varchar(50),street varchar(20) ) ,city varchar(50) ,district varchar(100) ,province varchar(100) ,PRIMARY KEY (agent_ID));
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE CAR (car_number varchar(20) NOT NULL, make varchar(50),model varchar(20),policy_NO varchar(20),PRIMARY KEY (car_number));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE BIKE (bike_number varchar(20) NOT NULL, make varchar(50),model varchar(20),policy_NO varchar(20),PRIMARY KEY (bike_number));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE ACCIDENT (vehicle_ID varchar(20) NOT NULL,report_num varchar(20) NOT NULL, report_date date NOT NULL,report_time time NOT NULL,venue varchar(50),PRIMARY KEY (report_num, vehicle_ID));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE PAYMENT (policy_NO varchar(20) NOT NULL, payment_date date NOT NULL,payment_method varchar(20) NOT NULL,payment_amount int,PRIMARY KEY (policy_NO, payment_date));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE AGENT_CONTACT (agent_ID varchar(20) NOT NULL , contact_number varchar(50) NOT NULL,PRIMARY KEY (agent_ID, contact_number));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE CUSTOMER_CONTACT (customer_ID varchar(20) , contact_number varchar(50) NOT NULL,PRIMARY KEY (customer_ID, contact_number));
Query OK, 0 rows affected (0.01 sec)
```

7°F Rain showers

Search



9:09 PM
9/10/2023

4.2 Demonstrating the Schema and Tables

```
MySQL 8.0 Command Line Cli × + ▾

mysql> show databases;
+-----+
| Database |
+-----+
| foo      |
| information_schema |
| ins_prj   |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
8 rows in set (0.00 sec)

mysql> use ins_prj;
Database changed
mysql> show tables;
+-----+
| Tables_in_ins_prj |
+-----+
| accident          |
| agent             |
| agent_contact     |
| bike              |
| car               |
| customer          |
| customer_contact  |
| life              |
| payment           |
| policy            |
| property          |
| vehicle           |
+-----+
11 rows in set (0.00 sec)

Cloud 22°C Mostly cloudy Search b F C O M X 12:07 PM
9/10/2023
```

4.3 Adding Constraints for Created Tables

```
MySQL 8.0 Command Line Cli + - X
mysql> ALTER TABLE CUSTOMER ADD CONSTRAINT FK_1 FOREIGN KEY(agent_ID) REFERENCES AGENT(agent_ID) ON DELETE cascade ON UPDATE cascade;
Query OK, 8 rows affected (0.04 sec)
Records: 8 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE POLICY ADD CONSTRAINT FK_2 FOREIGN KEY(customer_ID) REFERENCES CUSTOMER(customer_ID) ON DELETE cascade on UPDATE cascade;
Query OK, 11 rows affected (0.03 sec)
Records: 11 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE VEHICLE ADD CONSTRAINT FK_3 FOREIGN KEY(policy_NO) REFERENCES POLICY(policy_NO)ON DELETE CASCADE ON UPDATE CASCADE;
Query OK, 5 rows affected (0.04 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE CAR ADD CONSTRAINT FK_4 FOREIGN KEY(policy_NO) REFERENCES VEHICLE(policy_NO) ON DELETE cascade on update cascade;
Query OK, 2 rows affected (0.04 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE BIKE ADD CONSTRAINT FK_5 FOREIGN KEY(policy_NO) REFERENCES VEHICLE(policy_NO) ON DELETE cascade ON UPDATE cascade;
Query OK, 2 rows affected (0.04 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE LIFE ADD CONSTRAINT FK_6 FOREIGN KEY(policy_NO) REFERENCES POLICY(policy_NO) ON DELETE cascade ON UPDATE cascade;
Query OK, 2 rows affected (0.04 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE PROPERTY ADD CONSTRAINT FK_7 FOREIGN KEY(policy_NO) REFERENCES POLICY(policy_NO) ON DELETE cascade ON UPDATE cascade;
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE AGENT_CONTACT ADD CONSTRAINT FK_8 FOREIGN KEY(agent_ID) REFERENCES AGENT(agent_ID) ;
Query OK, 7 rows affected (0.03 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE CUSTOMER_CONTACT ADD CONSTRAINT FK_9 FOREIGN KEY(customer_ID) REFERENCES CUSTOMER(customer_ID) ;
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0

Cloudy 76°F Search b 🌐 w 10:51 PM 9/10/2023
```

```
MySQL 8.0 Command Line Cli + - X
mysql> ALTER TABLE PROPERTY ADD CONSTRAINT FK_7 FOREIGN KEY(policy_NO) REFERENCES POLICY(policy_NO) ON DELETE cascade ON UPDATE cascade;
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE AGENT_CONTACT ADD CONSTRAINT FK_8 FOREIGN KEY(agent_ID) REFERENCES AGENT(agent_ID) ;
Query OK, 7 rows affected (0.03 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE CUSTOMER_CONTACT ADD CONSTRAINT FK_9 FOREIGN KEY(customer_ID) REFERENCES CUSTOMER(customer_ID) ;
Query OK, 10 rows affected (0.03 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE accident ADD CONSTRAINT FK_10 FOREIGN KEY(vehicle_ID) REFERENCES policy(policy_NO) ;
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE payment ADD CONSTRAINT FK_11 FOREIGN KEY(policy_NO) REFERENCES policy(policy_NO) ;
Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0

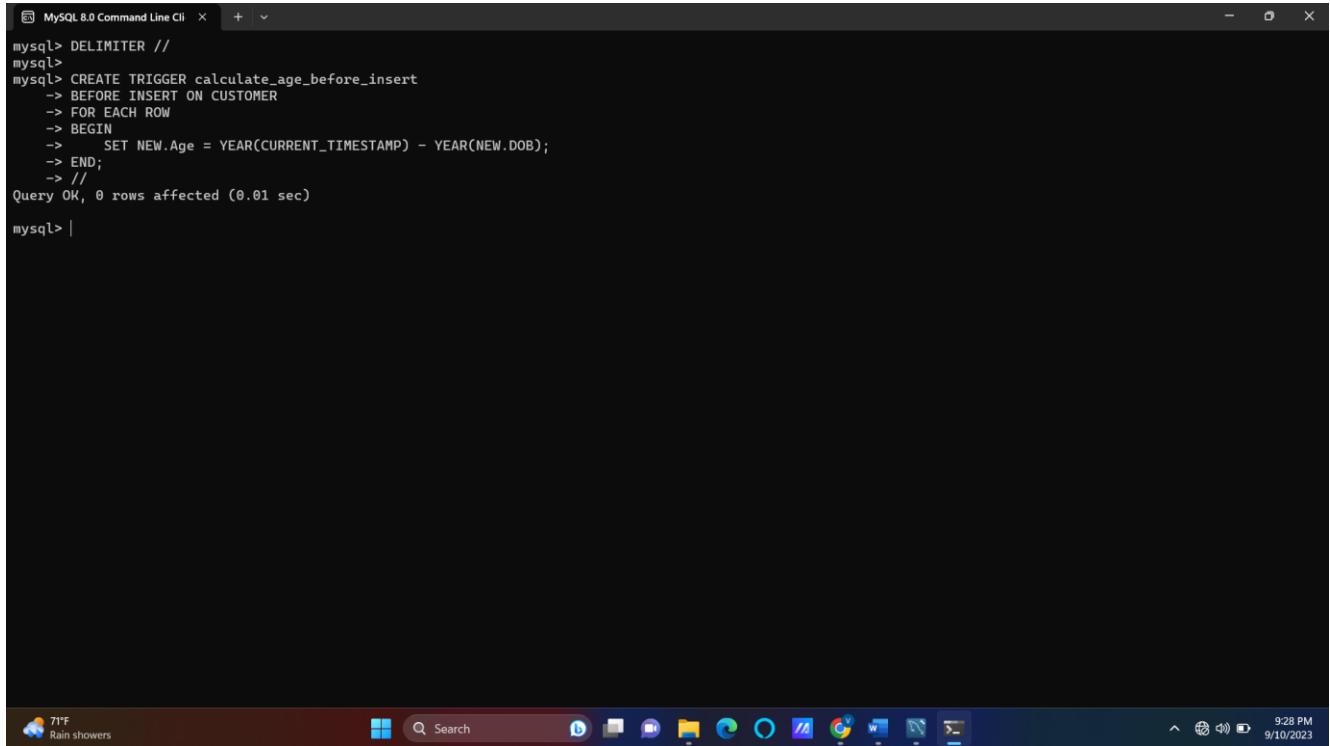
mysql> ALTER TABLE customer ADD CONSTRAINT FK_12 FOREIGN KEY(agent_as_customer) REFERENCES customer(customer_ID) ;
Query OK, 8 rows affected (0.05 sec)
Records: 8 Duplicates: 0 Warnings: 0

mysql>

Cloudy 76°F Search b 🌐 w 10:52 PM 9/10/2023
```

4.4 Creating a trigger for the derived attribute “Age.”

In order to make “Age” a derived attribute “Date_of_birth” attribute is added to the customer table. Age is derived in years from the date of birth of each customer. The screenshot of adding the relevant trigger is shown below.



The screenshot shows a terminal window titled "MySQL 8.0 Command Line Cli". The user has run the following SQL command to create a trigger:

```
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER calculate_age_before_insert
-> BEFORE INSERT ON CUSTOMER
-> FOR EACH ROW
-> BEGIN
->     SET NEW.Age = YEAR(CURRENT_TIMESTAMP) - YEAR(NEW.DOB);
-> END;
-> //
```

The command was successful, as indicated by the message "Query OK, 0 rows affected (0.01 sec)". The MySQL prompt "mysql>" is visible at the bottom of the window. The system tray at the bottom of the screen shows the weather as "71°F Rain showers" and the date and time as "9/10/2023 9:28 PM".

4.5 Table Definitions

■ Customer Table

```
MySQL 8.0 Command Line Cli x + v

+-----+
| customer | CREATE TABLE `customer` (
  `customer_ID` varchar(20) NOT NULL,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `DOB` date DEFAULT NULL,
  `Age` int DEFAULT NULL,
  `street` varchar(20) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `district` varchar(100) DEFAULT NULL,
  `province` varchar(100) DEFAULT NULL,
  `agent_ID` varchar(20) DEFAULT NULL,
  `agent_as_customer` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`customer_ID`),
  KEY `FK_1` (`agent_ID`),
  KEY `FK_12` (`agent_as_customer`),
  CONSTRAINT `FK_1` FOREIGN KEY (`agent_ID`) REFERENCES `agent` (`agent_ID`) ON DELETE SET NULL ON UPDATE CASCADE,
  CONSTRAINT `FK_12` FOREIGN KEY (`agent_as_customer`) REFERENCES `customer` (`customer_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
-----+
-----+
1 row in set (0.00 sec)

mysql> |
```



■ Agent Table

```
MySQL 8.0 Command Line Cli x + v

mysql> show create table agent;
+-----+
| Table | Create Table
+-----+
| agent | CREATE TABLE `agent` (
  `agent_ID` varchar(20) NOT NULL,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email` varchar(50) DEFAULT NULL,
  `street` varchar(20) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `district` varchar(100) DEFAULT NULL,
  `province` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`agent_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
-----+
1 row in set (0.00 sec)

mysql> |
```



■ Property Insurance Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table property;
+-----+-----+
| Table | Create Table
+-----+-----+
| property | CREATE TABLE `property` (
  `property_ID` varchar(20) NOT NULL,
  `cost` int DEFAULT NULL,
  `property_type` varchar(20) DEFAULT NULL,
  `street` varchar(20) DEFAULT NULL,
  `city` varchar(50) DEFAULT NULL,
  `district` varchar(100) DEFAULT NULL,
  `province` varchar(100) DEFAULT NULL,
  `policy_NO` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`property_ID`),
  KEY `FK_7` (`policy_NO`),
  CONSTRAINT `FK_7` FOREIGN KEY (`policy_NO`) REFERENCES `policy` (`policy_NO`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```



■ Policy Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table policy;
+-----+-----+
| Table | Create Table
+-----+-----+
| policy | CREATE TABLE `policy` (
  `policy_NO` varchar(20) NOT NULL,
  `policy_name` varchar(255) NOT NULL,
  `policy_type` varchar(100) NOT NULL,
  `coverage` varchar(50) DEFAULT NULL,
  `issued_date` date DEFAULT NULL,
  `term_price` int DEFAULT NULL,
  `customer_ID` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`policy_NO`),
  KEY `FK_2` (`customer_ID`),
  CONSTRAINT `FK_2` FOREIGN KEY (`customer_ID`) REFERENCES `customer` (`customer_ID`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```



■ Vehicle Insurance Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table vehicle;
+-----+
| Table | Create Table
+-----+
| vehicle | CREATE TABLE `vehicle` (
  `vehicle_ID` varchar(20) NOT NULL,
  `policy_type` varchar(100) NOT NULL,
  `start_date` date DEFAULT NULL,
  `end_date` date DEFAULT NULL,
  `policy_NO` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`vehicle_ID`),
  KEY `FK_3` (`policy_NO`),
  CONSTRAINT `FK_3` FOREIGN KEY (`policy_NO`) REFERENCES `policy` (`policy_NO`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```

■ Car Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table car;
+-----+
| Table | Create Table
+-----+
| car | CREATE TABLE `car` (
  `car_number` varchar(20) NOT NULL,
  `make` varchar(50) DEFAULT NULL,
  `model` varchar(20) DEFAULT NULL,
  `policy_NO` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`car_number`),
  KEY `FK_4` (`policy_NO`),
  CONSTRAINT `FK_4` FOREIGN KEY (`policy_NO`) REFERENCES `vehicle` (`policy_NO`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql> |
```

■ Bike Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table bike;
+-----+
| Table | Create Table
+-----+
| bike | CREATE TABLE `bike` (
  `bike_number` varchar(20) NOT NULL,
  `make` varchar(50) DEFAULT NULL,
  `model` varchar(20) DEFAULT NULL,
  `policy_NO` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`bike_number`),
  KEY `FK_5` (`policy_NO`),
  CONSTRAINT `FK_5` FOREIGN KEY (`policy_NO`) REFERENCES `vehicle` (`policy_NO`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```

■ Life Insurance Table

```
MySQL 8.0 Command Line Cli + x
mysql> show create table life;
+-----+
| Table | Create Table
+-----+
| life | CREATE TABLE `life` (
  `beneficiary_ID` varchar(20) NOT NULL,
  `first_name` varchar(50) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `relationship` varchar(50) DEFAULT NULL,
  `percentage_share` int DEFAULT NULL,
  `policy_NO` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`beneficiary_ID`),
  KEY `FK_6` (`policy_NO`),
  CONSTRAINT `FK_6` FOREIGN KEY (`policy_NO`) REFERENCES `policy` (`policy_NO`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```

■ Payment Table

```
MySQL 8.0 Command Line Cli + v
mysql> show create table payment;
+-----+
| Table | Create Table
+-----+
| payment | CREATE TABLE `payment` (
  `policy_NO` varchar(20) NOT NULL,
  `payment_date` date NOT NULL,
  `payment_method` varchar(20) NOT NULL,
  `payment_amount` int DEFAULT NULL,
  PRIMARY KEY (`policy_NO`),
  CONSTRAINT `FK_11` FOREIGN KEY (`policy_NO`) REFERENCES `policy` (`policy_NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql> |
```



■ Accident Table

```
MySQL 8.0 Command Line Cli + v
mysql> show create table accident;
+-----+
| Table | Create Table
+-----+
| accident | CREATE TABLE `accident` (
  `vehicle_ID` varchar(20) NOT NULL,
  `report_num` varchar(20) NOT NULL,
  `report_date` date NOT NULL,
  `report_time` time NOT NULL,
  `venue` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`report_num`),
  KEY `FK_10` (`vehicle_ID`),
  CONSTRAINT `FK_10` FOREIGN KEY (`vehicle_ID`) REFERENCES `policy` (`policy_NO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```



■ Agent_Contact Table

```
MySQL 8.0 Command Line Cli + v
mysql> show create table agent_contact;
+-----+
| Table      | Create Table
+-----+
| agent_contact | CREATE TABLE `agent_contact` (
  `agent_ID` varchar(20) NOT NULL,
  `contact_number` varchar(50) NOT NULL,
  PRIMARY KEY (`agent_ID`,`contact_number`),
  CONSTRAINT `FK_8` FOREIGN KEY (`agent_ID`) REFERENCES `agent` (`agent_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```



■ Customer_Contact Table

```
MySQL 8.0 Command Line Cli + v
mysql> show create table customer_contact;
+-----+
| Table      | Create Table
+-----+
| customer_contact | CREATE TABLE `customer_contact` (
  `customer_ID` varchar(20) NOT NULL,
  `contact_number` varchar(50) NOT NULL,
  PRIMARY KEY (`customer_ID`,`contact_number`),
  CONSTRAINT `FK_9` FOREIGN KEY (`customer_ID`) REFERENCES `customer` (`customer_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+
1 row in set (0.00 sec)

mysql>
```



4.6 Inserting Data for Tables

- Customer table

```
MySQL 8.0 Command Line Cli + - X
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1003', 'Raman', 'Udawala', '2000-09-10', 'New Road', 'Jaffna', 'Jaffna', 'Northern', 'AG02',null);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1004', 'Nuwan', 'Perera', '2000-09-10', 'Main Street', 'Suriyawewa', 'Hambantota', 'Southern', 'AG03',null);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1005', 'Pawan', 'Rajagiri', '2001-09-10', 'Main Street', 'Thanamalwila', 'Monaragala', 'Uva', 'AG04',null);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1006', 'Bimali', 'Sunil', '2002-09-10', 'No2', 'Kalmunai', 'Ampara', 'Uwa', 'AG05',null);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1007', 'Lakmali', 'Silva', '1999-09-10', 'Main Street', 'Homagama', 'Colombo', 'Western', 'AG05',null);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1008', 'Umalini', 'Srimali', '2000-09-10', 'Main Street', 'Hikkaduwa', 'Galle', 'Southern', 'AG05',null);
Query OK, 1 row affected (0.01 sec)

mysql> |
```

■ Agent table

```

MySQL 8.0 Command Line Cli  +  x

mysql> INSERT INTO AGENT VALUES ('AG05', 'Pawan', 'Weerasekara', 'Weerasekara456@gmail.com', 'Main Street', 'Koggala', 'Galle', 'Southern');
ERROR 1062 (23000): Duplicate entry 'AG05' for key 'agent.PRIMARY'
mysql> INSERT INTO AGENT VALUES ('AG04', 'Ruwan', 'Perera', 'ruwan123@gmail.com', 'Main Street', 'Jaffna', 'Jaffna', 'Northern');
ERROR 1062 (23000): Duplicate entry 'AG04' for key 'agent.PRIMARY'
mysql> INSERT INTO AGENT VALUES ('AG03', 'Piyal', 'Samarasinghe', 'piyal123@gmail.com', 'NO 5', 'Hapugala', 'Galle', 'Southern');
ERROR 1062 (23000): Duplicate entry 'AG03' for key 'agent.PRIMARY'
mysql> INSERT INTO AGENT VALUES ('AG02', 'Kumudu', 'Nimali', 'kumudunim23@gmail.com', 'Main Street', 'Dehiwala', 'Colombo', 'Western');
ERROR 1062 (23000): Duplicate entry 'AG02' for key 'agent.PRIMARY'
mysql> INSERT INTO AGENT VALUES ('AG01', 'Kamal', 'Jayasinghe', 'jayasinghe123@gmail.com', 'Main Street', 'Galle', 'Galle', 'Southern');
ERROR 1062 (23000): Duplicate entry 'AG01' for key 'agent.PRIMARY'
mysql>
mysql> select * from AGENT;
+-----+-----+-----+-----+-----+-----+-----+-----+
| agent_ID | first_name | last_name | email | street | city | district | province |
+-----+-----+-----+-----+-----+-----+-----+-----+
| AG01 | Kamal | Jayasinghe | jayasinghe123@gmail.com | Main Street | Galle | Galle | Southern |
| AG02 | Kumudu | Nimali | kumudunim23@gmail.com | Main Street | Dehiwala | Colombo | Western |
| AG03 | Piyal | Samarasinghe | piyal123@gmail.com | NO 5 | Hapugala | Galle | Southern |
| AG04 | Ruwan | Perera | ruwan123@gmail.com | Main Street | Jaffna | Jaffna | Northern |
| AG05 | Pawan | Weerasekara | Weerasekara456@gmail.com | Main Street | Koggala | Galle | Southern |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

The screenshot shows the MySQL command-line interface. The terminal window title is "MySQL 8.0 Command Line Cli". The command history shows multiple attempts to insert data into the "AGENT" table, each failing due to a primary key constraint violation (Duplicate entry 'AG0X' for key 'agent.PRIMARY'). Finally, the correct data is inserted, and the table is queried to show five rows of data. The operating system taskbar at the bottom indicates it's 9:52 PM on 9/10/2023.

■ Policy table

```

MySQL 8.0 Command Line Cli  +  x

mysql> INSERT INTO POLICY VALUES ('V-4257', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1001');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('L-4157', 'Family Security', 'Life', 'RS.10,000,000', '2023-09-01', 25000, 'CU1002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('P-4287', 'Home Safety', 'Property', 'Dwelling: Rs.1,50000 ', '2023-08-01', 6000, 'CU1003');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4259', 'Motorcycle Protection', 'Auto', 'Motorcycle Coverage', '2023-07-01', 8000, 'CU1002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-7257', 'Motorcycle Protection', 'Auto', 'Motorcycle Coverage', '2022-03-01', 8000, 'CU1004');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('P-4957', 'Renters Security', 'Property', 'Personal Belongings: Rs.40,000, Liability Coverage', '2023-04-01', 25000, 'CU1005');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('L-4237', 'Life Gaurd', 'Life', 'Rs.500,000', '2023-03-01', 25000, 'CU1006');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('P-1257', 'Home Safety', 'property', 'Dwelling: Rs.1000,000', '2023-03-03', 25000, 'CU1003');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4857', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-02', 50000, 'CU1007');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4157', 'Motorcycle Protection', 'Auto', 'Motorcycle Coverage', '2023-03-08', 75000, 'CU1008');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-6257', 'Children Future', 'Life', 'Rs.2500,000', '2023-03-01', 25000, 'CU1001');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM POLICY;
+-----+-----+-----+-----+-----+-----+-----+-----+
| policy_NO | policy_name | policy_type | coverage | issued_date | term_price | customer_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
| L-4157 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 25000 | CU1002 |
| L-4237 | Life Gaurd | Life | Rs.500,000 | 2023-03-01 | 25000 | CU1006 |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The screenshot shows the MySQL command-line interface. The terminal window title is "MySQL 8.0 Command Line Cli". The command history shows multiple attempts to insert data into the "POLICY" table, each failing due to primary key violations. Finally, the correct data is inserted, and the table is queried to show four rows of data. The operating system taskbar at the bottom indicates it's 9:56 PM on 9/10/2023.

```

MySQL 8.0 Command Line Cli + ▾

mysql> INSERT INTO POLICY VALUES ('L-4237', 'Life Gaurd', 'Life', 'Rs.500,000', '2023-03-01', 25000, 'CU1006');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('P-1257', 'Home Safety', 'property', 'Dwelling: Rs.1000,000', '2023-03-03', 25000, 'CU1003');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4857', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-02', 50000, 'CU1007');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4157', 'Motorcycle Protection', 'Auto', 'Motorcycle Coverage', '2023-03-08', 75000, 'CU1008');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-6257', 'Children Future', 'Life', 'Rs.2500,000', '2023-03-01', 25000, 'CU1001');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM POLICY;
+-----+-----+-----+-----+-----+-----+-----+
| policy_NO | policy_name | policy_type | coverage | issued_date | term_price | customer_ID |
+-----+-----+-----+-----+-----+-----+-----+
| L-4157 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 25000 | CU1002 |
| L-4237 | Life Gaurd | Life | Rs.500,000 | 2023-03-01 | 25000 | CU1006 |
| L-6257 | Children Future | Life | Rs.2500,000 | 2023-03-01 | 25000 | CU1001 |
| P-1257 | Home Safety | property | Dwelling: Rs.1000,000 | 2023-03-03 | 25000 | CU1003 |
| P-4287 | Home Safety | Property | Dwelling: Rs.1,50000 | 2023-08-01 | 6000 | CU1003 |
| P-4957 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 25000 | CU1005 |
| V-4157 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-03-08 | 75000 | CU1008 |
| V-4257 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1001 |
| V-4259 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-07-01 | 8000 | CU1002 |
| V-4857 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-02 | 50000 | CU1007 |
| V-7257 | Motorcycle Protection | Auto | Motorcycle Coverage | 2022-03-01 | 8000 | CU1004 |
+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql> |
```

9:57 PM
9/10/2023

■ Vehicle Insurance Table

```

MySQL 8.0 Command Line Cli + ▾

mysql> INSERT INTO VEHICLE VALUES ('ABC-123', 'Car', '2023-01-01', '2023-12-31', 'V-4257');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-223', 'Bike', '2023-01-01', '2023-12-31', 'V-4259');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('OLM-123', 'Bike', '2023-02-01', '2024-01-31', 'V-7257');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('XYZ-321', 'Car', '2023-01-01', '2023-12-31', 'V-4857');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('JKL-123', 'Bike', '2023-01-01', '2023-12-31', 'V-4157');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql>
mysql> SELECT * FROM VEHICLE;
+-----+-----+-----+-----+-----+
| vehicle_ID | policy_type | start_date | end_date | policy_NO |
+-----+-----+-----+-----+-----+
| ABC-123 | Car | 2023-01-01 | 2023-12-31 | V-4257 |
| JKL-123 | Bike | 2023-01-01 | 2023-12-31 | V-4157 |
| OLM-123 | Bike | 2023-02-01 | 2024-01-31 | V-7257 |
| PQR-223 | Bike | 2023-01-01 | 2023-12-31 | V-4259 |
| XYZ-321 | Car | 2023-01-01 | 2023-12-31 | V-4857 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

10:02 PM
9/10/2023

■ Car Table

```
MySQL 8.0 Command Line Cli + - X

mysql> INSERT INTO CAR VALUES ('ABC-123', 'Toyota', 'Toyota Highlander', 'V-4257');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CAR VALUES ('XYZ-321', 'BMW', 'BMW X5', 'V-4857');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM CAR;
+-----+-----+-----+-----+
| car_number | make | model | policy_NO |
+-----+-----+-----+-----+
| ABC-123 | Toyota | Toyota Highlander | V-4257 |
| XYZ-321 | BMW | BMW X5 | V-4857 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

69°F Rain showers 10:04 PM 9/10/2023

■ Bike Table

```
MySQL 8.0 Command Line Cli + - X

mysql> INSERT INTO BIKE VALUES ('JKL-123', 'Honda', 'Honda CRF25', 'V-4157');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO BIKE VALUES ('OLM-123', 'Yamaha', 'MT7', 'V-7257');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> SELECT * FROM BIKE;
+-----+-----+-----+-----+
| bike_number | make | model | policy_NO |
+-----+-----+-----+-----+
| JKL-123 | Honda | Honda CRF25 | V-4157 |
| OLM-123 | Yamaha | MT7 | V-7257 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

69°F Rain showers 10:05 PM 9/10/2023

■ Life Insurance Table

```
MySQL 8.0 Command Line Cli + - X
mysql> INSERT INTO LIFE VALUES ('B1001', 'Sunil', 'Perera', 'Child', 55, 'L-4157');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LIFE VALUES ('B1002', 'Yamaha', 'MT7', 'V-7257', 30, 'L-4237');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM LIFE;
+-----+-----+-----+-----+-----+
| beneficiary_ID | first_name | last_name | relationship | percentage_share | policy_NO |
+-----+-----+-----+-----+-----+
| B1001 | Sunil | Perera | Child | 55 | L-4157 |
| B1002 | Yamaha | MT7 | V-7257 | 30 | L-4237 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

69°F Rain showers 9/10/2023 10:07 PM

■ Property Insurance Table

```
MySQL 8.0 Command Line Cli + - X
mysql> INSERT INTO PROPERTY VALUES ('P1001', 25000, 'Single-family home', '234 Palm Lane', 'Weligama', 'Matara', 'Southern', 'P-4287');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PROPERTY VALUES ('P1002', 50000, 'Townhouse', 'Main road', 'Colombo', 'Colombo', 'Western', 'P-4957');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM PROPERTY;
+-----+-----+-----+-----+-----+-----+-----+
| property_ID | cost | property_type | street | city | district | province | policy_NO |
+-----+-----+-----+-----+-----+-----+-----+
| P1001 | 25000 | Single-family home | 234 Palm Lane | Weligama | Matara | Southern | P-4287 |
| P1002 | 50000 | Townhouse | Main road | Colombo | Colombo | Western | P-4957 |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

69°F Rain showers 9/10/2023 10:08 PM

■ Agent_Contact Table

```
MySQL 8.0 Command Line Cli + - x
mysql> INSERT INTO agent_contact VALUES ('AG01', '0711234567');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO agent_contact VALUES ('AG01', '0711236845');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO agent_contact VALUES ('AG02', '0721234567');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO agent_contact VALUES ('AG03', '0741234567');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO agent_contact VALUES ('AG03', '0749934567');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO agent_contact VALUES ('AG04', '0761234567');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO agent_contact VALUES ('AG05', '077123456');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> SELECT * FROM AGENT_CONTACT;
+-----+-----+
| agent_ID | contact_number |
+-----+-----+
| AG01     | 0711234567   |
| AG01     | 0711236845   |
| AG02     | 0721234567   |
| AG03     | 0741234567   |
| AG03     | 0749934567   |
| AG04     | 0761234567   |
| AG05     | 077123456    |
+-----+-----+
7 rows in set (0.00 sec)

mysql>
```



■ Customer_Contact Table

```
MySQL 8.0 Command Line Cli + - x
mysql> INSERT INTO customer_contact VALUES ('CU1001', '077133356');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1001', '077133396');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1002', '077133352');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1003', '077133350');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1004', '077970350');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1005', '074513356');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1005', '077133096');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1006', '077137352');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1007', '077134350');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO customer_contact VALUES ('CU1008', '077139350');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM CUSTOMER_CONTACT;
+-----+-----+
| customer_ID | contact_number |
+-----+-----+
| CU1001      | 077133356   |
| CU1001      | 077133396   |
| CU1002      | 077133352   |
| CU1003      | 077133350   |
| CU1004      | 077970350   |
| CU1005      | 074513356   |
| CU1005      | 077133096   |
| CU1006      | 077137352   |
| CU1007      | 077134350   |
| CU1008      | 077139350   |
+-----+-----+
10 rows in set (0.00 sec)

mysql>
```



■ Accident Table

```
MySQL 8.0 Command Line Cli × + ▾
mysql> INSERT INTO accident VALUES ('V-4257', 'ACC-40023', '2023-09-12', '13:07:00', 'Weligama');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO accident VALUES ('V-7257', 'ACC-40023', '2023-09-12', '13:07:00', 'Weligama');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO accident VALUES ('V-4857', 'ACC-41234', '2023-08-13', '11:08:00', 'Colombo');
Query OK, 1 row affected (0.01 sec)

mysql> select*from accident;
+-----+-----+-----+-----+-----+
| vehicle_ID | report_num | report_date | report_time | venue   |
+-----+-----+-----+-----+-----+
| V-4257    | ACC-40023  | 2023-09-12  | 13:07:00    | Weligama |
| V-7257    | ACC-40023  | 2023-09-12  | 13:07:00    | Weligama |
| V-4857    | ACC-41234  | 2023-08-13  | 11:08:00    | Colombo  |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> |
```

69°F Rain showers 10:20 PM 9/10/2023

■ Payment Table

```
MySQL 8.0 Command Line Cli × + ▾
mysql> INSERT INTO payment VALUES ('V-4857', '2023-02-10', 'Bank Transfer', 5000);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO payment VALUES ('V-7257', '2023-06-10', 'Credit', 4000);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO payment VALUES ('L-4237', '2023-08-10', 'Bank Transfer', 3000);
Query OK, 1 row affected (0.00 sec)

mysql> select*from payment;
+-----+-----+-----+-----+
| policy_NO | payment_date | payment_method | payment_amount |
+-----+-----+-----+-----+
| L-4237    | 2023-08-10  | Bank Transfer |      3000 |
| V-4857    | 2023-02-10  | Bank Transfer |      5000 |
| V-7257    | 2023-06-10  | Credit        |      4000 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

69°F Rain showers 10:22 PM 9/10/2023

4.7 Inserting, Updating and Deleting Operations for Existing Tables.

In this section insert operation is done again for future updating and deleting purposes.

Inserting

- Agent Table

```
MySQL 8.0 Command Line Cli < + >

mysql> insert into agent values ('AG06', 'Kamali', 'Jayasinghe', 'jayasinghe18883@gmail.com', 'Main Street', 'Thalpe', 'Galle', 'Southern');
Query OK, 1 row affected (0.01 sec)

mysql> insert into agent values ('AG07', 'bimal', 'ruwan', 'bimal12345@gmail.com', 'Main Street', 'Unwatuna', 'Galle', 'Southern');
Query OK, 1 row affected (0.01 sec)

mysql> insert into agent values ('AG08', 'Jayani', 'Jayasinghe', 'jayasinghe0011@gmail.com', 'No234', 'Thalpe', 'Galle', 'Southern');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM AGENT;
+-----+-----+-----+-----+-----+-----+-----+-----+
| agent_ID | first_name | last_name | email | street | city | district | province |
+-----+-----+-----+-----+-----+-----+-----+-----+
| AG01 | Kamal | Jayasinghe | jayasinghe123@gmail.com | Main Street | Galle | Galle | Southern
| AG02 | Kumudu | Nimali | kumudunim23@gmail.com | Main Street | Dehiwala | Colombo | Western
| AG03 | Piyal | Samarasinghe | piyal123@gmail.com | NO 5 | Hapugala | Galle | Southern
| AG04 | Ruwan | Perera | ruwan123@gmail.com | Main Street | Jaffna | Jaffna | Northern
| AG05 | Pawan | Weerasekara | Weerasekara456@gmail.com | Main Street | Koggala | Galle | Southern
| AG06 | Kamali | Jayasinghe | jayasinghe18883@gmail.com | Main Street | Thalpe | Galle | Southern
| AG07 | bimal | ruwan | bimal12345@gmail.com | Main Street | Unwatuna | Galle | Southern
| AG08 | Jayani | Jayasinghe | jayasinghe0011@gmail.com | No234 | Thalpe | Galle | Southern
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> |
```

- Customer Table

```
MySQL 8.0 Command Line Cli < + >

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU1009', 'Umal', 'Srimal', '2001-09-10', 'Main Street', 'Ginthota', 'Galle', 'Southern', 'AG06',null);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU10010', 'Kumudu', 'Rathnayaka', '2002-09-10', 'No234', 'Hikkaduwa', 'Galle', 'Southern', 'AG07',null);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO CUSTOMER (customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer) VALUES ('CU10011', 'Rayan', 'Srimal', '2000-09-25', 'No23', 'Unwatuna', 'Galle', 'Southern', 'AG08',null);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM COSTOMER;
ERROR 1146 (42S02): Table 'insurance.costomter' doesn't exist
mysql> SELECT * FROM CUSTOMER;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| customer_ID | first_name | last_name | DOB | street | city | district | province | agent_ID | agent_as_customer | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| CU1001 | Sumudu | Priyangani | 2000-09-10 | Main Street | Wakwella | Galle | Southern | AG01 | NULL | 23 |
| CU10010 | Kumudu | Rathnayaka | 2002-09-10 | No234 | Hikkaduwa | Galle | Southern | AG07 | NULL | 21 |
| CU10011 | Rayan | Srimal | 2000-09-25 | No23 | Unwatuna | Galle | Southern | AG08 | NULL | 23 |
| CU1002 | Piyal | Samarasinghe | 2001-09-10 | No5 | Hapugala | Galle | Southern | AG01 | CU1002 | 22 |
| CU1003 | Raman | Udawala | 2000-09-10 | New Road | Jaffna | Jaffna | Northern | AG02 | NULL | 23 |
| CU1004 | Nuwan | Perera | 2000-09-10 | Main Street | Suriyawewa | Hambantota | Southern | AG03 | NULL | 23 |
| CU1005 | Pawan | Rajagiri | 2001-09-10 | Main Street | Thananimalwila | Monaragala | Uva | AG04 | NULL | 22 |
| CU1006 | Bimal | Sunil | 2002-09-10 | NO2 | Kalmunai | Ampara | Uwa | AG05 | NULL | 21 |
| CU1007 | Lakmali | Silva | 1999-09-10 | Main Street | Homagama | Colombo | Western | AG06 | NULL | 24 |
| CU1008 | Umali | Srimali | 2000-09-10 | Main Street | Hikkaduwa | Galle | Southern | AG05 | NULL | 23 |
| CU1009 | Umal | Srimal | 2001-09-10 | Main Street | Ginthota | Galle | Southern | AG06 | NULL | 22 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> |
```

- Policy Table

```
MySQL 8.0 Command Line Cli + MySQL - _ X

mysql> INSERT INTO POLICY VALUES ('V-4000', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1009');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4001', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1009');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4002', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU10010');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4003', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU10011');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4004', 'Car Gaurd', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1009');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4005', 'Motorcycle Protection', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU10010');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4006', 'Motorcycle Protection', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1001');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('V-4007', 'Motorcycle Protection', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1009');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4008', 'Motorcycle Protection', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU1009');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('V-4009', 'Motorcycle Protection', 'Auto', 'Comprehensive Coverarage', '2023-03-01', 25000, 'CU10010');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-3000', 'Family Security', 'Life', 'RS.10,000,000', '2023-09-01', 50000, 'CU10011');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-3001', 'Family Security', 'Life', 'RS.10,000,000', '2023-09-01', 50000, 'CU10010');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-3002', 'Family Security', 'Life', 'RS.10,000,000', '2023-09-01', 50000, 'CU1009');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('L-3003', 'Family Security', 'Life', 'RS.10,000,000', '2023-09-01', 50000, 'CU1008');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO POLICY VALUES ('P-2000', 'Renter Security', 'Property', 'Personal Belongings: Rs.40,000, Liability Coverage', '2023-04-01', 75000, 'CU1009');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('P-2001', 'Renter Security', 'Property', 'Personal Belongings: Rs.40,000, Liability Coverage', '2023-04-01', 25000, 'CU10010');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('P-2002', 'Renter Security', 'Property', 'Personal Belongings: Rs.40,000, Liability Coverage', '2023-04-01', 75000, 'CU1008');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO POLICY VALUES ('P-2003', 'Renter Security', 'Property', 'Personal Belongings: Rs.40,000, Liability Coverage', '2023-04-01', 85000, 'CU1008');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM POLICY;
+-----+-----+-----+-----+-----+-----+-----+
| ID   | Type | Policy | Status | Premium | PolicyDate | PolicyNo |
+-----+-----+-----+-----+-----+-----+-----+
| 1    | Auto | Car Gaurd | Active | 25000 | 2023-03-01 | CU1009   |
| 2    | Auto | Car Gaurd | Active | 25000 | 2023-03-01 | CU1009   |
| 3    | Auto | Car Gaurd | Active | 25000 | 2023-03-01 | CU10010  |
| 4    | Auto | Car Gaurd | Active | 25000 | 2023-03-01 | CU10011  |
| 5    | Auto | Car Gaurd | Active | 25000 | 2023-03-01 | CU1009   |
| 6    | Auto | Motorcycle Protection | Active | 25000 | 2023-03-01 | CU10010  |
| 7    | Auto | Motorcycle Protection | Active | 25000 | 2023-03-01 | CU1001   |
| 8    | Auto | Motorcycle Protection | Active | 25000 | 2023-03-01 | CU1009   |
| 9    | Auto | Motorcycle Protection | Active | 25000 | 2023-03-01 | CU1009   |
| 10   | Auto | Motorcycle Protection | Active | 25000 | 2023-03-01 | CU10010  |
| 11   | Life | Family Security | Active | 50000 | 2023-09-01 | CU10011  |
| 12   | Life | Family Security | Active | 50000 | 2023-09-01 | CU10010  |
| 13   | Life | Family Security | Active | 50000 | 2023-09-01 | CU1009   |
| 14   | Life | Family Security | Active | 50000 | 2023-09-01 | CU1008   |
| 15   | Property | Renter Security | Active | 75000 | 2023-04-01 | CU1009   |
| 16   | Property | Renter Security | Active | 25000 | 2023-04-01 | CU10010  |
| 17   | Property | Renter Security | Active | 75000 | 2023-04-01 | CU1008   |
| 18   | Property | Renter Security | Active | 85000 | 2023-04-01 | CU1008   |
+-----+-----+-----+-----+-----+-----+-----+



20°C Mostly cloudy Search b w G 6:44 AM 9/11/2023
```

• Vehicle Insurance Table

```
MySQL 8.0 Command Line Cli + 
mysql> INSERT INTO VEHICLE VALUES ('ABC-001', 'car', '2023-01-01', '2023-12-31', 'V-4000');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('XYZ-002', 'car', '2023-01-01', '2023-12-31', 'V-4001');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('ABC-003', 'car', '2023-01-01', '2023-12-31', 'V-4002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('XYZ-004', 'car', '2023-01-01', '2023-12-31', 'V-4003');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('XYZ-005', 'car', '2023-01-01', '2023-12-31', 'V-4004');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-001', 'bike', '2023-01-01', '2023-12-31', 'V-4005');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-002', 'bike', '2023-01-01', '2023-12-31', 'V-4006');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-003', 'bike', '2023-01-01', '2023-12-31', 'V-4007');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-004', 'bike', '2023-01-01', '2023-12-31', 'V-4008');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO VEHICLE VALUES ('PQR-005', 'bike', '2023-01-01', '2023-12-31', 'V-4009');
Query OK, 1 row affected (0.01 sec)

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+
| vehicle_ID | policy_type | start_date | end_date | policy_NO |
+-----+-----+-----+-----+-----+
| ABC-001 | car | 2023-01-01 | 2023-12-31 | V-4000 |
| ABC-003 | car | 2023-01-01 | 2023-12-31 | V-4002 |
| ABC-123 | Car | 2023-01-01 | 2023-12-31 | V-4257 |
| JKL-123 | Bike | 2023-01-01 | 2023-12-31 | V-4157 |
| OLM-123 | Bike | 2023-02-01 | 2024-01-31 | V-7257 |
| PQR-001 | bike | 2023-01-01 | 2023-12-31 | V-4005 |
| PQR-002 | bike | 2023-01-01 | 2023-12-31 | V-4006 |
| PQR-003 | bike | 2023-01-01 | 2023-12-31 | V-4007 |
| PQR-004 | bike | 2023-01-01 | 2023-12-31 | V-4008 |
| PQR-005 | bike | 2023-01-01 | 2023-12-31 | V-4009 |
| PQR-223 | Bike | 2023-01-01 | 2023-12-31 | V-4259 |
| XYZ-002 | car | 2023-01-01 | 2023-12-31 | V-4001 |
| XYZ-004 | car | 2023-01-01 | 2023-12-31 | V-4003 |
| XYZ-005 | car | 2023-01-01 | 2023-12-31 | V-4004 |
| XYZ-321 | Car | 2023-01-01 | 2023-12-31 | V-4857 |
+-----+-----+-----+-----+-----+
21 rows in set (0.00 sec)

21°C Mostly cloudy Search 7:02 AM 9/11/2023
```

• Car Table

```
MySQL 8.0 Command Line Cli + 
mysql> INSERT INTO CAR VALUES ('ABC-001', 'Toyota', 'Toyota Highlander', 'V-4000');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CAR VALUES ('XYZ-002', 'BMW', 'BMW X5', 'V-4001');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CAR VALUES ('ABC-003', 'Toyota', 'Toyota Highlander', 'V-4002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CAR VALUES ('XYZ-004', 'BMW', 'BMW X5', 'V-4003');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO CAR VALUES ('XYZ-005', 'BMW', 'BMW X5', 'V-4004');
Query OK, 1 row affected (0.00 sec)

mysql> select * from car;
+-----+-----+-----+-----+
| car_number | make | model | policy_NO |
+-----+-----+-----+-----+
| ABC-001 | Toyota | Toyota Highlander | V-4000 |
| ABC-003 | Toyota | Toyota Highlander | V-4002 |
| ABC-123 | Toyota | Toyota Highlander | V-4257 |
| XYZ-002 | BMW | BMW X5 | V-4001 |
| XYZ-004 | BMW | BMW X5 | V-4003 |
| XYZ-005 | BMW | BMW X5 | V-4004 |
| XYZ-321 | BMW | BMW X5 | V-4857 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> |
```

- Life Insurance Table

```
MySQL 8.0 Command Line Cli + ▾
mysql> INSERT INTO LIFE VALUES ('B1003', 'Kamal', 'Perea', 'Child', 30, 'L-3000');Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO LIFE VALUES ('B1004', 'Raman', 'Perea', 'Child', 30, 'L-3001');Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LIFE VALUES ('B1005', 'Pamali', 'Perea', 'Child', 30, 'L-3002');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO LIFE VALUES ('B1006', 'Bimal', 'Perea', 'Child', 30, 'L-3003');Query OK, 1 row affected (0.00 sec)

mysql> select * from life;
+-----+-----+-----+-----+-----+-----+
| beneficiary_ID | first_name | last_name | relationship | percentage_share | policy_NO |
+-----+-----+-----+-----+-----+-----+
| B1001 | Sunil | Perea | Child | 55 | L-4157 |
| B1002 | Yamaha | MT7 | V-7257 | 30 | L-4237 |
| B1003 | Kamal | Perea | Child | 30 | L-3000 |
| B1004 | Raman | Perea | Child | 30 | L-3001 |
| B1005 | Pamali | Perea | Child | 30 | L-3002 |
| B1006 | Bimal | Perea | Child | 30 | L-3003 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> |
```

21°C Mostly cloudy 7:14 AM 9/11/2023

- Property Insurance Table

```
MySQL 8.0 Command Line Cli + ▾
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPERTY VALUES ('P1004', 75000, 'Townhouse', 'Main road', 'Colombo', 'Colombo', 'Western', 'P-2001');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO PROPERTY VALUES ('P1005', 75000, 'Townhouse', 'Main road', 'Colombo', 'Colombo', 'Western', 'P-2002');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO PROPERTY VALUES ('P1006', 75000, 'Townhouse', 'Main road', 'Colombo', 'Colombo', 'Western', 'P-2003');
Query OK, 1 row affected (0.01 sec)

mysql> select * from property;
+-----+-----+-----+-----+-----+-----+-----+-----+
| property_ID | cost | property_type | street | city | district | province | policy_NO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P1001 | 25000 | Single-family home | 234 Palm Lane | Weligama | Matara | Southern | P-4287 |
| P1002 | 50000 | Townhouse | Main road | Colombo | Colombo | Western | P-4957 |
| P1003 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2000 |
| P1004 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2001 |
| P1005 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2002 |
| P1006 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2003 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> |
```

21°C Mostly cloudy 7:18 AM 9/11/2023

- Accident Table

```
MySQL 8.0 Command Line Cli + MySQL> INSERT INTO accident VALUES ('V-4000', 'ACC-41234', '2023-08-13', '11:08:00', 'Colombo');  
Query OK, 1 row affected (0.01 sec)  
  
MySQL> INSERT INTO accident VALUES ('V-4002', 'ACC-41344', '2023-08-23', '15:08:00', 'Galle');  
Query OK, 1 row affected (0.00 sec)  
  
MySQL> INSERT INTO accident VALUES ('V-4003', 'ACC-41456', '2023-08-23', '15:08:00', 'Galle');  
Query OK, 1 row affected (0.01 sec)  
  
MySQL> select * from accident;  
+-----+-----+-----+-----+-----+  
| vehicle_ID | report_num | report_date | report_time | venue |  
+-----+-----+-----+-----+-----+  
| V-4257 | ACC-40023 | 2023-09-12 | 13:07:00 | Weligama |  
| V-7257 | ACC-40023 | 2023-09-12 | 13:07:00 | Weligama |  
| V-4000 | ACC-41234 | 2023-08-13 | 11:08:00 | Colombo |  
| V-4857 | ACC-41234 | 2023-08-13 | 11:08:00 | Colombo |  
| V-4002 | ACC-41344 | 2023-08-23 | 15:08:00 | Galle |  
| V-4003 | ACC-41456 | 2023-08-23 | 15:08:00 | Galle |  
+-----+-----+-----+-----+-----+  
6 rows in set (0.00 sec)  
  
MySQL>
```

- Payment Table

```
MySQL 8.0 Command Line Cli + MySQL> INSERT INTO payment VALUES ('L-3000', '2023-07-10', 'Bank Transfer', 6000);  
Query OK, 1 row affected (0.01 sec)  
  
MySQL> INSERT INTO payment VALUES ('L-3001', '2023-07-10', 'Credit', 7000);  
Query OK, 1 row affected (0.00 sec)  
  
MySQL> INSERT INTO payment VALUES ('L-3002', '2023-07-10', 'Bank Transfer', 6000);  
Query OK, 1 row affected (0.00 sec)  
  
MySQL> select * from payment;  
+-----+-----+-----+-----+  
| policy_NO | payment_date | payment_method | payment_amount |  
+-----+-----+-----+-----+  
| L-3000 | 2023-07-10 | Bank Transfer | 6000 |  
| L-3001 | 2023-07-10 | Credit | 7000 |  
| L-3002 | 2023-07-10 | Bank Transfer | 6000 |  
| L-4237 | 2023-08-10 | Bank Transfer | 3000 |  
| V-4857 | 2023-02-10 | Bank Transfer | 5000 |  
| V-7257 | 2023-06-10 | Credit | 4000 |  
+-----+-----+-----+-----+  
6 rows in set (0.00 sec)  
  
MySQL> |
```

Updating

- Customer Table

- Agent Table

• Policy Table

```

mysql> update policy set policy_name ='Motorcycle Protection Pro' where policy_NO = 'V-4000';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update policy set policy_type ='Auto Plus' where policy_NO = 'V-4001 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from policy;
+-----+-----+-----+-----+-----+-----+-----+
| policy_NO | policy_name | policy_type | coverage | issued_date | term_price | customer_ID |
+-----+-----+-----+-----+-----+-----+-----+
| L-3000 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU10011 |
| L-3001 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU10010 |
| L-3002 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU1009 |
| L-3003 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU1008 |
| L-4157 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 25000 | CU1002 |
| L-4237 | Life Gaurd | Life | Rs.500,000 | 2023-03-01 | 25000 | CU1006 |
| L-6257 | Children Future | Life | Rs.2500,000 | 2023-03-01 | 25000 | CU1001 |
| P-1257 | Home Safety | property | Dwelling: Rs.1000,000 | 2023-03-03 | 25000 | CU1003 |
| P-2000 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 75000 | CU1009 |
| P-2001 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 25000 | CU10010 |
| P-2002 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 75000 | CU1008 |
| P-2003 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 85000 | CU1008 |
| P-4287 | Home Safety | Property | Dwelling: Rs.1,50000 | 2023-08-01 | 6000 | CU1003 |
| P-4957 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 25000 | CU1005 |
| V-4000 | Motorcycle Protection Pro | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4001 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4002 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4003 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10011 |
| V-4004 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4005 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4006 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1001 |
| V-4007 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4008 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4009 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4157 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-03-08 | 75000 | CU1008 |
| V-4257 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1001 |
| V-4259 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-07-01 | 8000 | CU1002 |
| V-4857 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-02 | 50000 | CU1007 |
| V-7257 | Motorcycle Protection | Auto | Motorcycle Coverage | 2022-03-01 | 8000 | CU1004 |
+-----+-----+-----+-----+-----+-----+-----+

```

21°C Cloudy 7:54 AM 9/11/2023

• Vehicle Table

```

mysql> update VEHICLE set start_date ='2023-02-01' where policy_NO = 'V-4000';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update VEHICLE set end_date ='2023-11-30' where policy_NO = 'V-4002 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+
| vehicle_ID | policy_type | start_date | end_date | policy_NO |
+-----+-----+-----+-----+-----+
| ABC-001 | car | 2023-02-01 | 2023-12-31 | V-4000 |
| ABC-003 | car | 2023-01-01 | 2023-12-31 | V-4002 |
| ABC-123 | Car | 2023-01-01 | 2023-12-31 | V-4257 |
| JKL-123 | Bike | 2023-01-01 | 2023-12-31 | V-4157 |
| OLM-123 | Bike | 2023-02-01 | 2024-01-31 | V-7257 |
| PQR-001 | bike | 2023-01-01 | 2023-12-31 | V-4005 |
| PQR-002 | bike | 2023-01-01 | 2023-12-31 | V-4006 |
| PQR-003 | bike | 2023-01-01 | 2023-12-31 | V-4007 |
| PQR-004 | bike | 2023-01-01 | 2023-12-31 | V-4008 |
| PQR-005 | bike | 2023-01-01 | 2023-12-31 | V-4009 |
| PQR-223 | Bike | 2023-01-01 | 2023-12-31 | V-4259 |
| XYZ-002 | car | 2023-01-01 | 2023-12-31 | V-4001 |
| XYZ-004 | car | 2023-01-01 | 2023-12-31 | V-4003 |
| XYZ-005 | car | 2023-01-01 | 2023-12-31 | V-4004 |
| XYZ-321 | Car | 2023-01-01 | 2023-12-31 | V-4857 |
+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)

mysql>

```

25°C Haze 8:42 AM 9/11/2023

- Car Table

```
MySQL 8.0 Command Line Cli + X

mysql> update car set model ='xxxx' where car_number = 'ABC-001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update car set make ='yyyyy' where car_number = 'ABC-003 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from car;
+-----+-----+-----+-----+
| car_number | make | model | policy_NO |
+-----+-----+-----+-----+
| ABC-001 | Toyota | xxxx | V-4000 |
| ABC-003 | Toyota | Toyota Highlander | V-4002 |
| ABC-123 | Toyota | Toyota Highlander | V-4257 |
| XYZ-002 | BMW | BMW X5 | V-4001 |
| XYZ-004 | BMW | BMW X5 | V-4003 |
| XYZ-005 | BMW | BMW X5 | V-4004 |
| XYZ-321 | BMW | BMW X5 | V-4857 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> |
```

- Bike Table

```
MySQL 8.0 Command Line Cli + X

mysql> update bike set model ='bbbbbb' where bike_number = 'OLM-001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from bike;
+-----+-----+-----+-----+
| bike_number | make | model | policy_NO |
+-----+-----+-----+-----+
| JKL-123 | Honda | Honda CRF25 | V-4157 |
| OLM-001 | Yamaha | bbbbb | V-4005 |
| OLM-002 | Yamaha | MT7 | V-4006 |
| OLM-003 | Yamaha | MT7 | V-4007 |
| OLM-004 | Yamaha | MT7 | V-4008 |
| OLM-005 | Yamaha | MT6 | V-4009 |
| OLM-123 | Yamaha | MT7 | V-7257 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> update bike set make ='aaaaaa' where bike_number = 'OLM-003 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> update bike set make ='aaaaaa' where bike_number = 'OLM-002 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> |
```

- Life Insurance table

```
MySQL 8.0 Command Line Cli + MySQL OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update life set first_name ='Priyawardana' where beneficiary_ID = 'B1001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update life set relationship ='Gurdian' where beneficiary_ID = 'B1003 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from life;
+-----+-----+-----+-----+-----+
| beneficiary_ID | first_name | last_name | relationship | percentage_share | policy_NO |
+-----+-----+-----+-----+-----+
| B1001 | Priyawardana | Perera | Child | 55 | L-4157 |
| B1002 | Yamaha | MT7 | V-7257 | 30 | L-4237 |
| B1003 | Kamal | Perea | Child | 30 | L-3000 |
| B1004 | Raman | Perea | Child | 30 | L-3001 |
| B1005 | Pamali | Perea | Child | 30 | L-3002 |
| B1006 | Bimal | Perea | Child | 30 | L-3003 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> |
```

- Property Table

- Accident Table

```
MySQL 8.0 Command Line Cli + X

mysql> update accident set venue ='Matara' where vehicle_ID = 'V-4257';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update accident set venue ='Galle' where vehicle_ID = 'V-7257 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from accident;
+-----+-----+-----+-----+-----+
| vehicle_ID | report_num | report_date | report_time | venue |
+-----+-----+-----+-----+-----+
| V-4257 | ACC-40023 | 2023-09-12 | 13:07:00 | Matara |
| V-7257 | ACC-40023 | 2023-09-12 | 13:07:00 | Weligama |
| V-4000 | ACC-41234 | 2023-08-13 | 11:08:00 | Colombo |
| V-4857 | ACC-41234 | 2023-08-13 | 11:08:00 | Colombo |
| V-4002 | ACC-41344 | 2023-08-23 | 15:08:00 | Galle |
| V-4003 | ACC-41456 | 2023-08-23 | 15:08:00 | Galle |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

- Payment Table

```
MySQL 8.0 Command Line Cli + X

mysql> update payment set payment_method ='Debit' where policy_NO = 'L-3000';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update payment set payment_method ='Debit' where policy_NO = 'L-3001 ';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 0 Changed: 0 Warnings: 0

mysql> select * from payment;
+-----+-----+-----+-----+
| policy_NO | payment_date | payment_method | payment_amount |
+-----+-----+-----+-----+
| L-3000 | 2023-07-10 | Debit | 6000 |
| L-3001 | 2023-07-10 | Credit | 7000 |
| L-3002 | 2023-07-10 | Bank Transfer | 6000 |
| L-4237 | 2023-08-10 | Bank Transfer | 3000 |
| V-4857 | 2023-02-10 | Bank Transfer | 5000 |
| V-7257 | 2023-06-10 | Credit | 4000 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> |
```

Deleting

- Payment Table

```
MySQL 8.0 Command Line Cli < + >
mysql> delete from payment where policy_NO = 'L-3000';Query OK, 1 row affected (0.00 sec)

mysql> select * from payment;
+-----+-----+-----+-----+
| policy_NO | payment_date | payment_method | payment_amount |
+-----+-----+-----+-----+
| L-3001    | 2023-07-10 | Credit        |      7000 |
| L-3002    | 2023-07-10 | Bank Transfer |      6000 |
| L-4237    | 2023-08-10 | Bank Transfer |      3000 |
| V-4857    | 2023-02-10 | Bank Transfer |      5000 |
| V-7257    | 2023-06-10 | Credit        |      4000 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```



- Accident Table

```
MySQL 8.0 Command Line Cli < + >
mysql> delete from accident where vehicle_ID = 'V-4000';
Query OK, 1 row affected (0.00 sec)

mysql> select * from accident;
+-----+-----+-----+-----+-----+
| vehicle_ID | report_num | report_date | report_time | venue   |
+-----+-----+-----+-----+-----+
| V-4257     | ACC-40023  | 2023-09-12  | 13:07:00    | Matara  |
| V-7257     | ACC-40023  | 2023-09-12  | 13:07:00    | Weligama |
| V-4857     | ACC-41234  | 2023-08-13  | 11:08:00    | Colombo  |
| V-4002     | ACC-41344  | 2023-08-23  | 15:08:00    | Galle    |
| V-4003     | ACC-41456  | 2023-08-23  | 15:08:00    | Galle    |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```



- Property Table

```
MySQL 8.0 Command Line Cli < + >
mysql> delete from property where policy_NO = 'P-2003';
Query OK, 1 row affected (0.01 sec)

mysql> select * from proptry;
ERROR 1146 (42S02): Table 'insurance.proptry' doesn't exist
mysql> select * from property;
+-----+-----+-----+-----+-----+-----+-----+-----+
| property_ID | cost | property_type | street | city | district | province | policy_NO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P1001 | 100000 | Single-family home | 234 Palm Lane | Weligama | Matara | Southern | P-4287 |
| P1002 | 50000 | Townhouse | Main road | Colombo | Colombo | Western | P-4957 |
| P1003 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2000 |
| P1004 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2001 |
| P1005 | 75000 | Townhouse | Main road | Colombo | Colombo | Western | P-2002 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

- Life Insurance Table

```
MySQL 8.0 Command Line Cli < + >
mysql> delete from life where policy_NO = 'L-3003';
Query OK, 1 row affected (0.01 sec)

mysql> select * from life;
+-----+-----+-----+-----+-----+-----+
| beneficiary_ID | first_name | last_name | relationship | pecentage_share | policy_NO |
+-----+-----+-----+-----+-----+-----+
| B1001 | Priyawardana | Perera | Child | 55 | L-4157 |
| B1002 | Yamaha | MT7 | V-7257 | 30 | L-4237 |
| B1003 | Kamal | Pereea | Child | 30 | L-3000 |
| B1004 | Raman | Pereea | Child | 30 | L-3001 |
| B1005 | Pamali | Pereea | Child | 30 | L-3002 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> |
```

- Bike Table

```
MySQL 8.0 Command Line Cli + v
mysql> delete from bike where policy_NO = 'V-4005';
Query OK, 1 row affected (0.01 sec)

mysql> select * from bike;
+-----+-----+-----+-----+
| bike_number | make | model | policy_NO |
+-----+-----+-----+-----+
| JKL-123 | Honda | Honda CRF25 | V-4157 |
| OLM-002 | Yamaha | MT7 | V-4006 |
| OLM-003 | Yamaha | MT7 | V-4007 |
| OLM-004 | Yamaha | MT7 | V-4008 |
| OLM-005 | Yamaha | MT6 | V-4009 |
| OLM-123 | Yamaha | MT7 | V-7257 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

21°C Light rain 9:42 AM 9/11/2023

- Car Table

```
MySQL 8.0 Command Line Cli + v
mysql> delete from car where policy_NO = 'V-4001';
Query OK, 1 row affected (0.01 sec)

mysql> select * from car;
+-----+-----+-----+-----+
| car_number | make | model | policy_NO |
+-----+-----+-----+-----+
| ABC-001 | Toyota | xxxx | V-4000 |
| ABC-003 | Toyota | Toyota Highlander | V-4002 |
| ABC-123 | Toyota | Toyota Highlander | V-4257 |
| XYZ-004 | BMW | BMW X5 | V-4003 |
| XYZ-005 | BMW | BMW X5 | V-4004 |
| XYZ-321 | BMW | BMW X5 | V-4857 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

22°C Cloudy 9:43 AM 9/11/2023

- Vehicle Table

```
MySQL 8.0 Command Line Cli + 
mysql> delete from vehicle where policy_NO = 'V-4005';
Query OK, 1 row affected (0.01 sec)

mysql> select * from vehicle;
+-----+-----+-----+-----+-----+
| vehicle_ID | policy_type | start_date | end_date | policy_NO |
+-----+-----+-----+-----+-----+
| ABC-001 | car | 2023-02-01 | 2023-12-31 | V-4000 |
| ABC-003 | car | 2023-01-01 | 2023-12-31 | V-4002 |
| ABC-123 | Car | 2023-01-01 | 2023-12-31 | V-4257 |
| JKL-123 | Bike | 2023-01-01 | 2023-12-31 | V-4157 |
| OLM-123 | Bike | 2023-02-01 | 2024-01-31 | V-7257 |
| PQR-002 | bike | 2023-01-01 | 2023-12-31 | V-4006 |
| PQR-003 | bike | 2023-01-01 | 2023-12-31 | V-4007 |
| PQR-004 | bike | 2023-01-01 | 2023-12-31 | V-4008 |
| PQR-005 | bike | 2023-01-01 | 2023-12-31 | V-4009 |
| PQR-223 | Bike | 2023-01-01 | 2023-12-31 | V-4259 |
| XYZ-002 | car | 2023-01-01 | 2023-12-31 | V-4001 |
| XYZ-004 | car | 2023-01-01 | 2023-12-31 | V-4003 |
| XYZ-005 | car | 2023-01-01 | 2023-12-31 | V-4004 |
| XYZ-321 | Car | 2023-01-01 | 2023-12-31 | V-4857 |
+-----+-----+-----+-----+
14 rows in set (0.00 sec)

mysql>
```

22°C High winds soon 9:46 AM 9/11/2023

- Policy Table

```
MySQL 8.0 Command Line Cli + 
mysql> delete from policy where policy_NO = 'L-3000';
Query OK, 1 row affected (0.01 sec)

mysql> select * from policy;
+-----+-----+-----+-----+-----+-----+-----+-----+
| policy_NO | policy_name | policy_type | coverage | issued_date | term_price | customer_ID |
+-----+-----+-----+-----+-----+-----+-----+
| L-3001 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU10010 |
| L-3002 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU1009 |
| L-3003 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 50000 | CU1008 |
| L-4157 | Family Security | Life | RS.10,000,000 | 2023-09-01 | 25000 | CU1002 |
| L-4237 | Life Gaurd | Life | Rs.500,000 | 2023-03-01 | 25000 | CU1006 |
| L-6257 | Children Future | Life | Rs.2500,000 | 2023-03-01 | 25000 | CU1001 |
| P-1257 | Home Safety | property | Dwelling: Rs.1000,000 | 2023-03-03 | 25000 | CU1003 |
| P-2000 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 75000 | CU1009 |
| P-2001 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 25000 | CU10010 |
| P-2002 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 75000 | CU1008 |
| P-2003 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 85000 | CU1008 |
| P-4287 | Home Safety | Property | Dwelling: Rs.1,50000 | 2023-08-01 | 6000 | CU1003 |
| P-4957 | Renters Security | Property | Personal Belongings: Rs.40,000, Liability Coverage | 2023-04-01 | 25000 | CU1005 |
| V-4000 | Motorcycle Protection Pro | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4001 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4002 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4003 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10011 |
| V-4004 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4005 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4006 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1001 |
| V-4007 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4008 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1009 |
| V-4009 | Motorcycle Protection | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU10010 |
| V-4157 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-03-08 | 75000 | CU1008 |
| V-4257 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-01 | 25000 | CU1001 |
| V-4259 | Motorcycle Protection | Auto | Motorcycle Coverage | 2023-07-01 | 8000 | CU1002 |
| V-4857 | Car Gaurd | Auto | Comprehensive Coverarage | 2023-03-02 | 50000 | CU1007 |
| V-7257 | Motorcycle Protection | Auto | Motorcycle Coverage | 2022-03-01 | 8000 | CU1004 |
+-----+-----+-----+-----+-----+-----+-----+
28 rows in set (0.00 sec)

mysql>
```

22°C Heavy rain later 9:48 AM 9/11/2023

Chapter 5 – Transactions

Seven simple queries and thirteen complex queries is performed according to given instructions. Screenshots are shown below. Tables used for each transaction are retrieved first in order to give clear understanding about the transaction. After that query is performed.

5.1 Simple queries

- Select operation

The screenshot shows a Windows desktop environment with a MySQL 8.0 Command Line Cli window open. The command entered is `SELECT * FROM CUSTOMER;`. The resulting table has 11 rows and 13 columns, corresponding to the fields in the CUSTOMER table: customer_ID, first_name, last_name, DOB, street, city, district, province, agent_ID, agent_as_customer, and Age. The data includes various names like Umali, Priyangani, Kumudu, Rathnayaka, Rayan, Srimal, Piyal, Ariyarathna, Raman, Udawala, Nuwan, Perera, Pawan, Rajagiri, Bimal, Sunil, Lakmali, Silva, Umali, Srimali, and Omal, along with their respective details such as birth dates (e.g., 2000-09-10, 2002-09-10) and ages (e.g., 23, 21). The MySQL prompt mysql> is visible at the bottom.

customer_ID	first_name	last_name	DOB	street	city	district	province	agent_ID	agent_as_customer	Age
CU1001	Umali	Priyangani	2000-09-10	Main Street	Wakwella	Galle	Southern	AG01	NULL	23
CU10010	Kumudu	Rathnayaka	2002-09-10	No234	Hikkaduwa	Galle	Southern	AG07	NULL	21
CU10011	Rayan	Srimal	2000-09-25	No23	Unwatuna	Galle	Southern	AG08	NULL	23
CU1002	Piyal	Ariyarathna	2001-09-10	No5	Hapugala	Galle	Southern	AG01	CU1002	22
CU1003	Raman	Udawala	2000-09-10	New Road	Jaffna	Jaffna	Northern	AG02	NULL	23
CU1004	Nuwan	Perera	2000-09-10	Main Street	Suriyawewa	Hambantota	Southern	AG03	NULL	23
CU1005	Pawan	Rajagiri	2001-09-10	Main Street	Thanamalwila	Monaragala	Uva	AG04	NULL	22
CU1006	Bimal	Sunil	2002-09-10	NO2	Kalmunai	Ampara	Uwa	AG05	NULL	21
CU1007	Lakmali	Silva	1999-09-10	Main Street	Homagama	Colombo	Western	AG05	NULL	24
CU1008	Umali	Srimali	2000-09-10	Main Street	Hikkaduwa	Galle	Southern	AG05	NULL	23
CU1009	Omal	Srimal	2001-09-10	Main Street	Ginthota	Galle	Southern	AG06	NULL	22

- Project operation

The screenshot shows a Windows desktop environment with a MySQL 8.0 Command Line Cli window open. The command entered is `SELECT first_name, last_name FROM CUSTOMER;`. The resulting table has 11 rows and 2 columns, showing only the first_name and last_name for each customer. The data includes pairs like Umali, Priyangani, Kumudu, Rathnayaka, Rayan, Srimal, Piyal, Ariyarathna, Raman, Udawala, Nuwan, Perera, Pawan, Rajagiri, Bimal, Sunil, Lakmali, Silva, Umali, Srimali, and Omal. The MySQL prompt mysql> is visible at the bottom.

first_name	last_name
Umali	Priyangani
Kumudu	Rathnayaka
Rayan	Srimal
Piyal	Ariyarathna
Raman	Udawala
Nuwan	Perera
Pawan	Rajagiri
Bimal	Sunil
Lakmali	Silva
Umali	Srimali
Omal	Srimal

- Cartesian product operation

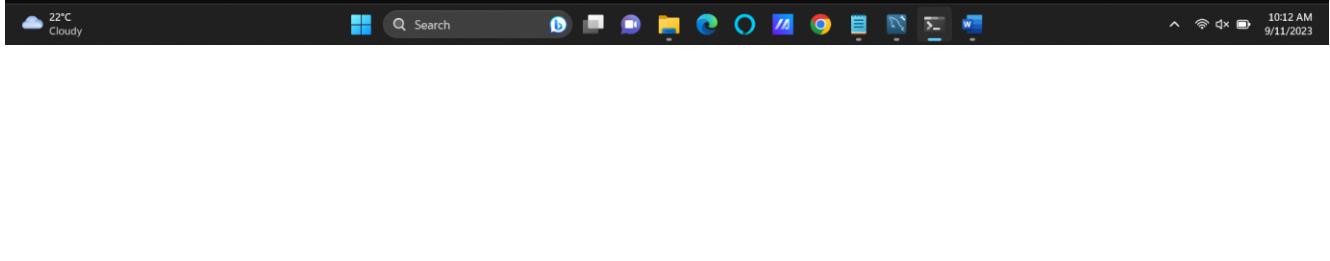
```
MySQL 8.0 Command Line Cli + 
mysql> SELECT C.first_name, C.last_name, P.policy_name
-> FROM CUSTOMER C
-> CROSS JOIN POLICY P;
+-----+-----+-----+
| first_name | last_name | policy_name |
+-----+-----+-----+
| Umali      | Srimali   | Family Security
| Umali      | Srimali   | Family Security
| Bimal      | Sunil     | Family Security
| Pawan      | Rajagiri   | Family Security
| Nuwan      | Perera    | Family Security
| Raman      | Udawala   | Family Security
| Piyal      | Ariyaratna | Family Security
| Kumudu     | Rathnayaka | Family Security
| Umali      | Priyangani | Family Security
| Umali      | Srimal    | Family Security
| Umali      | Srimali   | Family Security
| Bimal      | Sunil     | Family Security
| Pawan      | Rajagiri   | Family Security
| Nuwan      | Perera    | Family Security
| Raman      | Udawala   | Family Security
| Piyal      | Ariyaratna | Family Security
| Kumudu     | Rathnayaka | Family Security
| Umali      | Priyangani | Family Security
| Umali      | Srimal    | Family Security
| Umali      | Srimali   | Family Security
| Bimal      | Sunil     | Family Security
| Pawan      | Rajagiri   | Family Security
| Nuwan      | Perera    | Family Security
| Raman      | Udawala   | Family Security
| Piyal      | Ariyaratna | Family Security
| Kumudu     | Rathnayaka | Family Security
| Umali      | Priyangani | Family Security
| Umali      | Srimal    | Family Security
| Umali      | Srimali   | Family Security
| Bimal      | Sunil     | Family Security
| Pawan      | Rajagiri   | Family Security
| Nuwan      | Perera    | Family Security
| Raman      | Udawala   | Family Security
| Piyal      | Ariyaratna | Family Security
| Kumudu     | Rathnayaka | Family Security
| Umali      | Priyangani | Family Security
| Umali      | Srimal    | Life Gaurd
| Umali      | Srimali   | Life Gaurd
+-----+-----+-----+
```

- Creating a user view

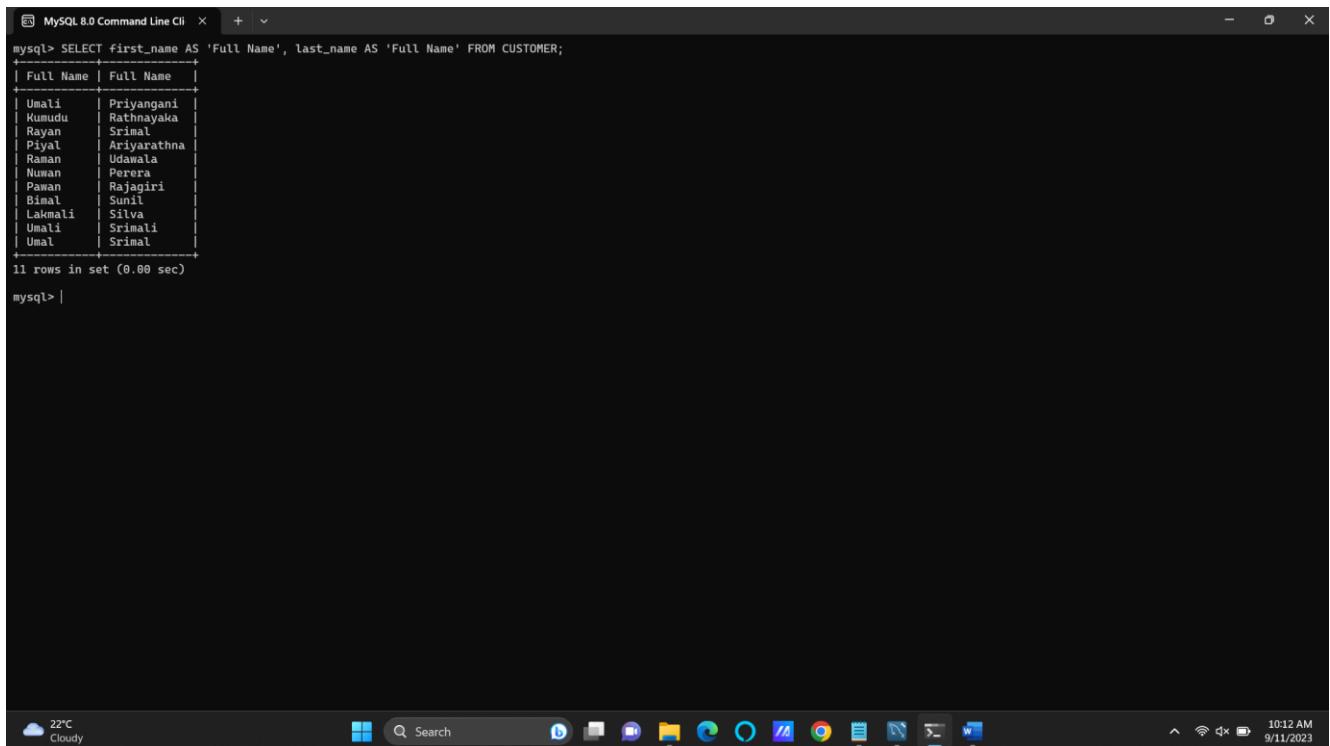
```
MySQL 8.0 Command Line Cli + 
mysql> CREATE VIEW CustomerAgentView AS SELECT customer_ID, first_name, last_name, agent_ID FROM CUSTOMER;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from CustomerAgentView;
+-----+-----+-----+-----+
| customer_ID | first_name | last_name | agent_ID |
+-----+-----+-----+-----+
| CU1001      | Umali      | Priyangani | AG01
| CU10010     | Kumudu     | Rathnayaka | AG07
| CU10011     | Rayan      | Srimal    | AG08
| CU1002      | Piyal      | Ariyaratna | AG01
| CU1003      | Raman      | Udawala   | AG02
| CU1004      | Nuwan      | Perera    | AG03
| CU1005      | Pawan      | Rajagiri   | AG04
| CU1006      | Bimal      | Sunil     | AG05
| CU1007      | Lakmali    | Silva     | AG05
| CU1008      | Umali      | Srimali   | AG05
| CU1009      | Umali      | Srimali   | AG06
+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```



- Rename Operation



MySQL 8.0 Command Line Cli

```
mysql> SELECT first_name AS 'Full Name', last_name AS 'Full Name' FROM CUSTOMER;
+-----+-----+
| Full Name | Full Name |
+-----+-----+
| Umalli     | Priyangani |
| Kumudu    | Rathnayaka |
| Rayan      | Srimal      |
| Piyal      | Ariyarathna |
| Raman      | Udawala    |
| Nuwan      | Perera     |
| Pawan      | Rajagiri   |
| Binjal    | Sunit      |
| Lakmali    | Silva      |
| Umali      | Srimali    |
| Umal       | Srimal     |
+-----+-----+
11 rows in set (0.00 sec)

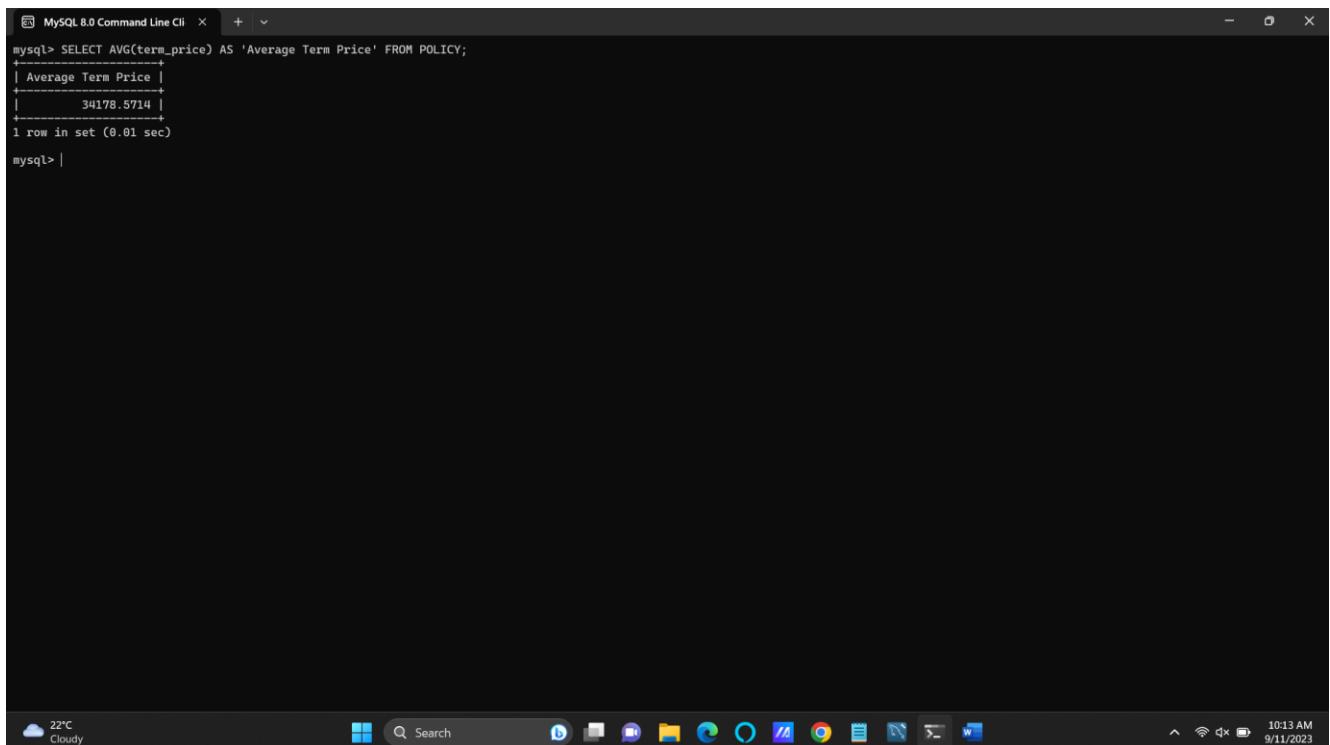
mysql> |
```

22°C Cloudy

Search

10:12 AM
9/11/2023

- Demonstrating the use of an aggregation function (Average)



MySQL 8.0 Command Line Cli

```
mysql> SELECT AVG(term_price) AS 'Average Term Price' FROM POLICY;
+-----+
| Average Term Price |
+-----+
|      34178.5714 |
+-----+
1 row in set (0.01 sec)

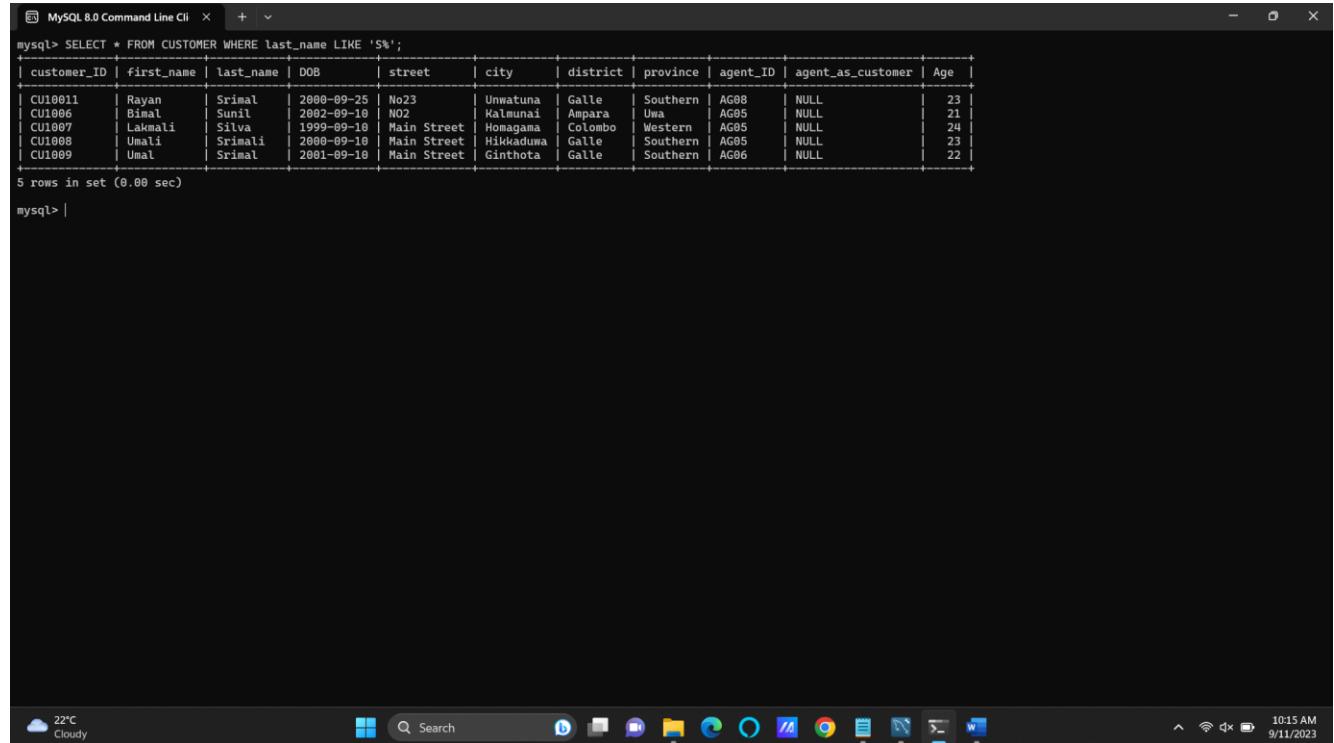
mysql> |
```

22°C Cloudy

Search

10:13 AM
9/11/2023

- Demonstrating the use of LIKE keyword



The screenshot shows a Windows desktop environment with a MySQL 8.0 Command Line Client window open. The window displays a query result from the 'CUSTOMER' table where the last name starts with 'S'. The results are as follows:

customer_ID	first_name	last_name	DOB	street	city	district	province	agent_ID	agent_as_customer	Age
CU10011	Rayan	Srimal	2000-09-25	No23	Unwatuna	Galle	Southern	AG08	NULL	23
CU1006	Binal	Sunil	2002-09-18	N02	Kalumnaia	Ampara	Uwa	AG05	NULL	21
CU1007	Lakmali	Silva	1999-09-18	Main Street	Homagama	Colombo	Western	AG05	NULL	24
CU1008	Umal	Srimali	2000-09-18	Main Street	Hikkaduwa	Galle	Southern	AG05	NULL	23
CU1009	Umal	Srimal	2001-09-18	Main Street	Ginthota	Galle	Southern	AG06	NULL	22

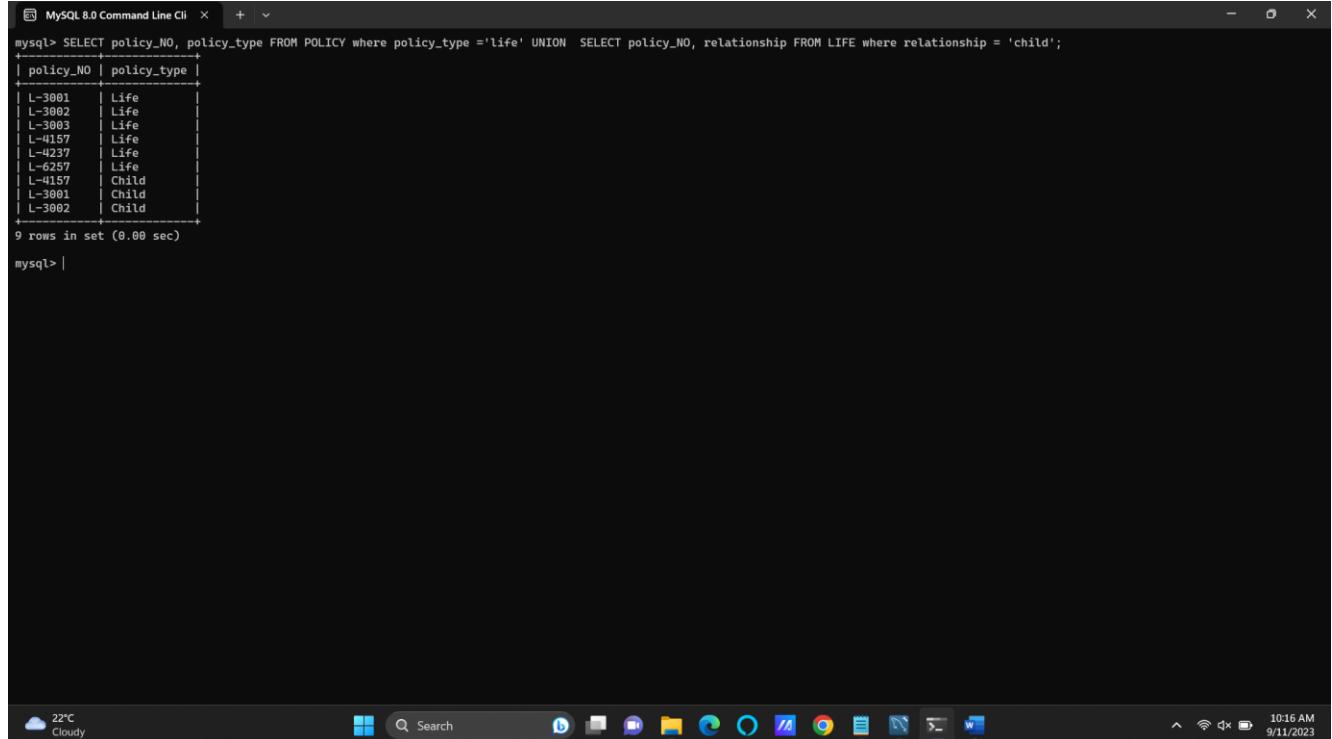
5 rows in set (0.00 sec)

mysql> |

The taskbar at the bottom of the screen shows various application icons, including Microsoft Edge, File Explorer, and other system utilities. The system tray indicates it's 22°C and cloudy, and the date and time are 9/11/2023 at 10:15 AM.

5.2 Complex queries

- Union operation



MySQL 8.0 Command Line Cli

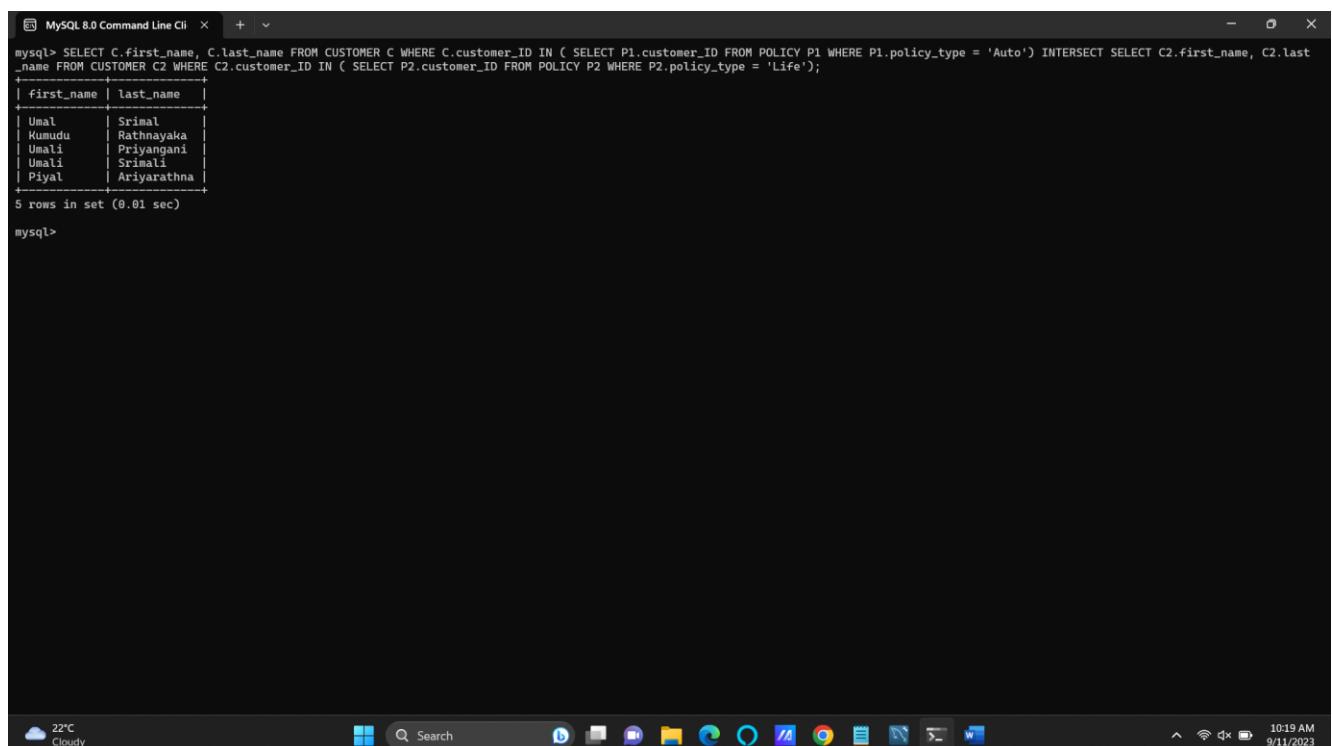
```
mysql> SELECT policy_NO, policy_type FROM POLICY where policy_type = 'life' UNION SELECT policy_NO, relationship FROM LIFE where relationship = 'child';
+-----+-----+
| policy_NO | policy_type |
+-----+-----+
| L-3001    | Life
| L-3002    | Life
| L-3003    | Life
| L-4157    | Life
| L-4237    | Life
| L-6257    | Life
| L-4157    | Child
| L-3001    | Child
| L-3002    | Child
+-----+-----+
9 rows in set (0.00 sec)

mysql> |
```

Windows taskbar at the bottom:

- Cloudy 22°C
- Search bar
- Icons for File, Print, Copy, Paste, Find, Replace, etc.
- System icons: battery, signal, volume, date/time (10:16 AM, 9/11/2023)

- Intersection Operation



MySQL 8.0 Command Line Cli

```
mysql> SELECT C.first_name, C.last_name FROM CUSTOMER C WHERE C.customer_ID IN ( SELECT P1.customer_ID FROM POLICY P1 WHERE P1.policy_type = 'Auto') INTERSECT SELECT C2.first_name, C2.last_name FROM CUSTOMER C2 WHERE C2.customer_ID IN ( SELECT P2.customer_ID FROM POLICY P2 WHERE P2.policy_type = 'Life');
+-----+-----+
| first_name | last_name |
+-----+-----+
| Umal       | Srimal
| Kumudu    | Rathnayaka
| Umali      | Priyangani
| Umali      | Srimali
| Piyal     | Ariyarathna
+-----+-----+
5 rows in set (0.01 sec)

mysql> |
```

Windows taskbar at the bottom:

- Cloudy 22°C
- Search bar
- Icons for File, Print, Copy, Paste, Find, Replace, etc.
- System icons: battery, signal, volume, date/time (10:19 AM, 9/11/2023)

- Set Difference Operation

```
MySQL 8.0 Command Line Cli + X
mysql> SELECT city,district FROM CUSTOMER where district ='Galle' EXCEPT SELECT policy_type, term_price FROM Policy where term_price = '25000';
+-----+-----+
| city | district |
+-----+-----+
| Wakwella | Galle
| Hikkaduwa | Galle
| Unawatuna | Galle
| Hapugala | Galle
| Ginthota | Galle
+-----+-----+
5 rows in set (0.00 sec)

mysql> |
```

22°C Rain coming Search b 📁 🚙 🎮 🌐 🌐 🌐 🌐 🌐 🌐 10:20 AM 9/11/2023

- Division Operation

```
MySQL 8.0 Command Line Cli + X
mysql> SELECT C.first_name, C.last_name FROM CUSTOMER C WHERE NOT EXISTS ( SELECT P.policy_type FROM POLICY P WHERE P.policy_type = 'Auto' AND NOT EXISTS ( SELECT CP.policy_type FROM POLICY CP WHERE CP.customer_ID = C.customer_ID AND CP.policy_type = P.policy_type));
+-----+-----+
| first_name | last_name |
+-----+-----+
| Umali | Priyangani
| Kumudu | Rathmaysaka
| Rayan | Srimal
| Piyal | Ariyarathna
| Numan | Perera
| Lakmali | Silva
| Umali | Srimali
| Umal | Srimal
+-----+-----+
8 rows in set (0.00 sec)

mysql> |
```

22°C Cloudy Search b 📁 🚙 🎮 🌐 🌐 🌐 🌐 🌐 🌐 10:23 AM 9/11/2023

- Inner Join Operation

- Natural Join Operation

```
MySQL 8.0 Command Line Cli + - X

mysql> SELECT C.first_name, C.last_name, P.policy_name FROM CUSTOMER C NATURAL JOIN POLICY P;
+-----+-----+-----+
| first_name | last_name | policy_name |
+-----+-----+-----+
| Umali      | Priyangani | Children Future
| Umali      | Priyangani | Motorcycle Protection
| Umali      | Priyangani | Car Gaurd
| Kumudu     | Rathnayaka | Family Security
| Kumudu     | Rathnayaka | Renters Security
| Kumudu     | Rathnayaka | Car Gaurd
| Kumudu     | Rathnayaka | Motorcycle Protection
| Kumudu     | Rathnayaka | Motorcycle Protection
| Rayan      | Srimal     | Car Gaurd
| Piyal      | Ariyarathna | Family Security
| Piyal      | Ariyarathna | Motorcycle Protection
| Raman      | Udawala    | Home Safety
| Raman      | Udawala    | Home Safety
| Nuwan      | Perera     | Motorcycle Protection
| Pawan      | Rajagiri   | Renters Security
| Bimal      | Sunil      | Life Gaurd
| Lakmali    | Silva      | Car Gaurd
| Umali      | Srimali    | Family Security
| Umali      | Srimali    | Renters Security
| Umali      | Srimali    | Renters Security
| Umali      | Srimali    | Motorcycle Protection
| Umali      | Srimali    | Family Security
| Umal       | Srimal     | Renters Security
| Umal       | Srimal     | Motorcycle Protection Pro
| Umal       | Srimal     | Car Gaurd
| Umal       | Srimal     | Car Gaurd
| Umal       | Srimal     | Motorcycle Protection
| Umal       | Srimal     | Motorcycle Protection
+-----+-----+-----+
28 rows in set (0.00 sec)

mysql>
```

- Left Outer Join Operation

```
MySQL 8.0 Command Line Cli x + v
mysql> SELECT C.first_name, C.last_name, P.property_type FROM PROPERTY P LEFT JOIN CUSTOMER C ON C.province = P.province;
+-----+-----+-----+
| first_name | last_name | property_type |
+-----+-----+-----+
| Umal      | Srimali   | Single-family home |
| Umal      | Srimali   | Single-family home |
| Numan     | Perera    | Single-family home |
| Piyal     | Ariyarathna | Single-family home |
| Rayan     | Srimal    | Single-family home |
| Kumudu    | Rathnayaka | Single-family home |
| Umal      | Priyanganani | Single-family home |
| Lakmali   | Silva     | Townhouse |
+-----+-----+-----+
11 rows in set (0.00 sec)

mysql>
```

22°C High winds soon 10:26 AM 9/11/2023

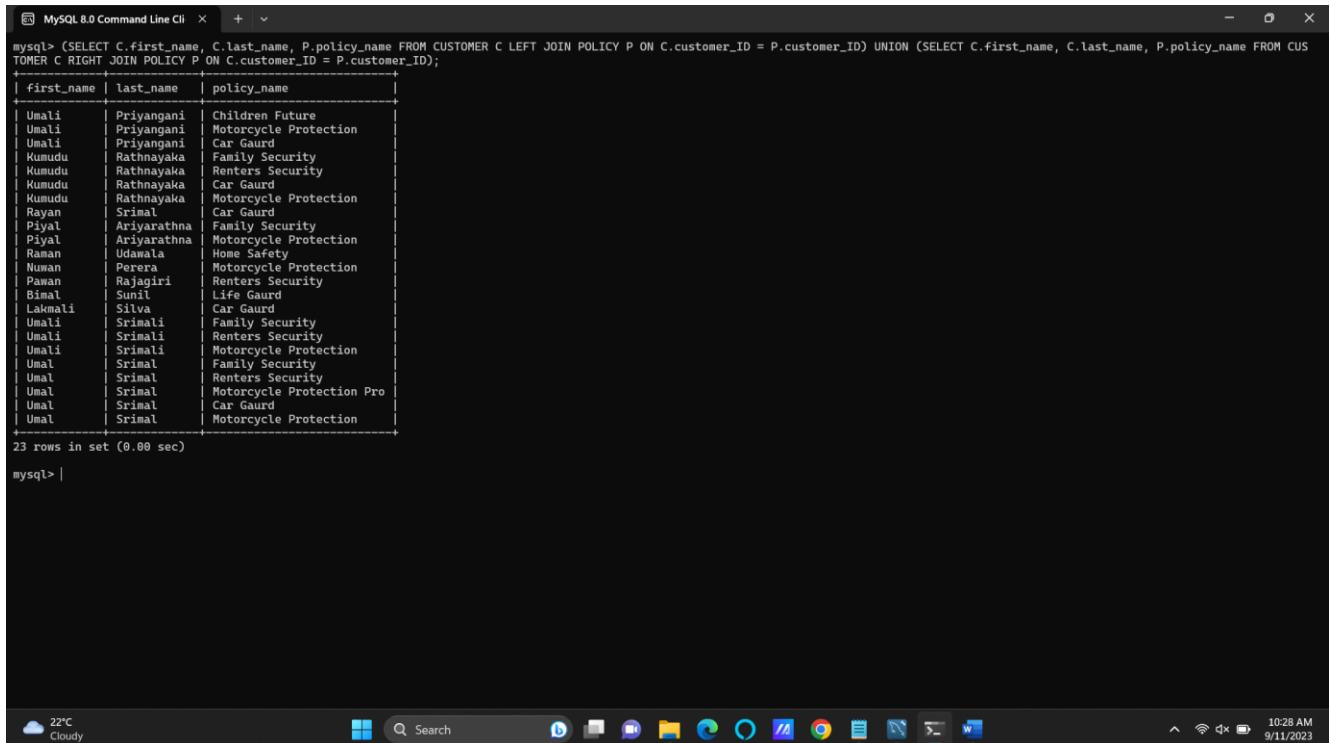
- Right Join Outer Operation

```
MySQL 8.0 Command Line Cli x + v
mysql> SELECT C.first_name, C.last_name, P.policy_name FROM CUSTOMER C RIGHT JOIN POLICY P ON C.customer_ID = P.customer_ID;
+-----+-----+-----+
| first_name | last_name | policy_name |
+-----+-----+-----+
| Kumudu    | Rathnayaka | Family Security |
| Umal      | Srimal    | Family Security |
| Umal      | Srimali   | Family Security |
| Piyal     | Ariyarathna | Family Security |
| Bimal     | Sunil     | Life Gaurd |
| Umal      | Priyanganani | Children Future |
| Raman     | Udawala   | Home Safety |
| Umal      | Srimal    | Renters Security |
| Kumudu    | Rathnayaka | Renters Security |
| Umal      | Srimali   | Renters Security |
| Umal      | Srimali   | Renters Security |
| Raman     | Udawala   | Home Safety |
| Panan     | Rajagiri  | Renters Security |
| Umal      | Srimal    | Motorcycle Protection Pro |
| Umal      | Srimal    | Car Gaurd |
| Kumudu    | Rathnayaka | Car Gaurd |
| Rayan     | Srimal    | Car Gaurd |
| Umal      | Srimal    | Car Gaurd |
| Kumudu    | Rathnayaka | Motorcycle Protection |
| Umal      | Priyanganani | Motorcycle Protection |
| Umal      | Srimal    | Motorcycle Protection |
| Umal      | Srimal    | Motorcycle Protection |
| Kumudu    | Rathnayaka | Motorcycle Protection |
| Umal      | Srimali   | Motorcycle Protection |
| Umal      | Priyanganani | Motorcycle Protection |
| Piyal     | Ariyarathna | Motorcycle Protection |
| Lakmali   | Silva     | Car Gaurd |
| Numan     | Perera    | Motorcycle Protection |
+-----+-----+-----+
28 rows in set (0.00 sec)

mysql> |
```

Cloudy 10:26 AM 9/11/2023

- Full Outer Join Operation



MySQL 8.0 Command Line Cli

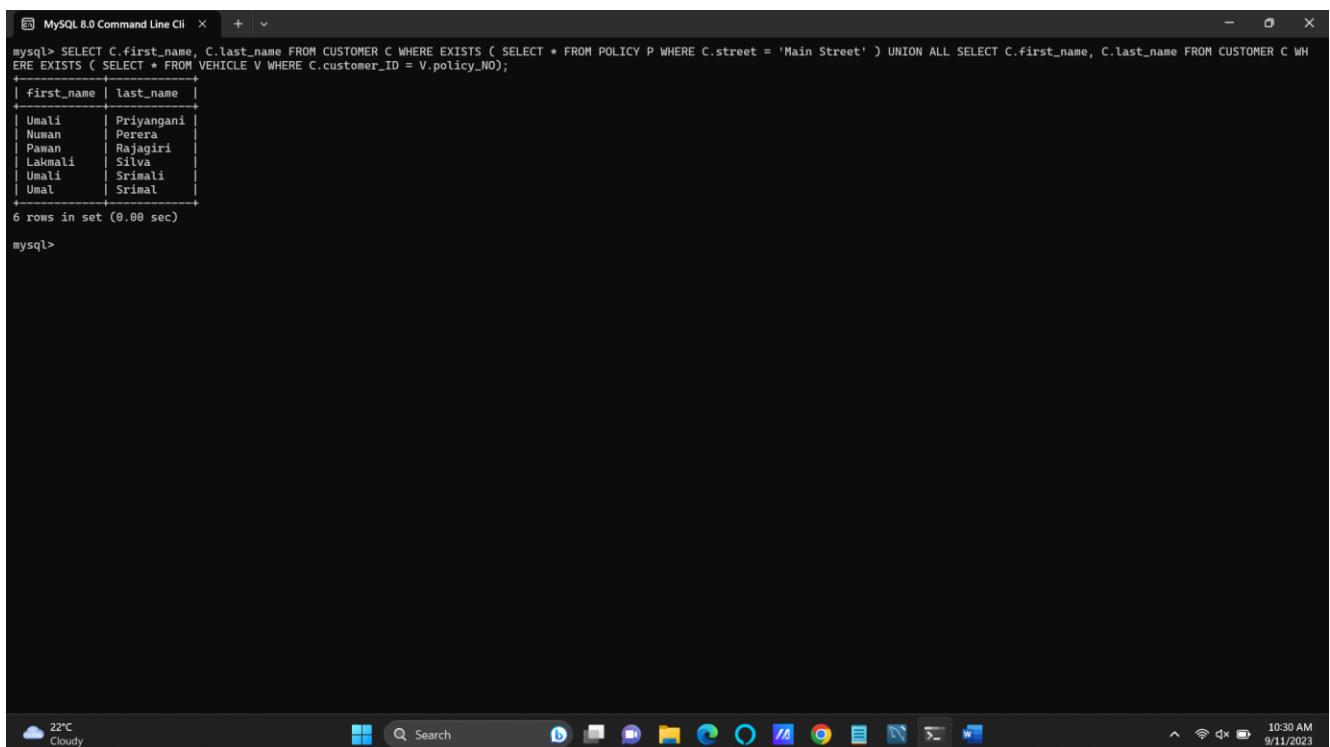
```
mysql> (SELECT C.first_name, C.last_name, P.policy_name FROM CUSTOMER C LEFT JOIN POLICY P ON C.customer_ID = P.customer_ID) UNION (SELECT C.first_name, C.last_name, P.policy_name FROM CUSTOMER C RIGHT JOIN POLICY P ON C.customer_ID = P.customer_ID);
+-----+-----+-----+
| first_name | last_name | policy_name |
+-----+-----+-----+
| Umali      | Priyangani | Children Future
| Umali      | Priyangani | Motorcycle Protection
| Umali      | Priyangani | Car Gaurd
| Kumudu     | Rathnayaka | Family Security
| Kumudu     | Rathnayaka | Renters Security
| Kumudu     | Rathnayaka | Car Gaurd
| Kumudu     | Rathnayaka | Motorcycle Protection
| Rayan      | Srimal      | Car Gaurd
| Piyal      | Ariyarathna | Family Security
| Piyal      | Ariyarathna | Motorcycle Protection
| Raman      | Udawala    | Home Safety
| Nuwan      | Perera     | Motorcycle Protection
| Pawan      | Rajagiri   | Renters Security
| Bimal      | Sunil       | Life Gaurd
| Lakmali    | Silva       | Car Gaurd
| Umali      | Srimali    | Family Security
| Umali      | Srimali    | Renters Security
| Umali      | Srimali    | Motorcycle Protection
| Umal       | Srimal     | Family Security
| Umal       | Srimal     | Renters Security
| Umal       | Srimal     | Motorcycle Protection Pro
| Umal       | Srimal     | Car Gaurd
| Umal       | Srimal     | Motorcycle Protection
23 rows in set (0.00 sec)

mysql>
```

Windows taskbar at the bottom:

- Cloudy 22°C
- Search bar
- Icons for File, Print, Copy, Paste, Find, Replace, etc.
- System icons: battery, signal, volume, date/time (10:28 AM, 9/11/2023)

- Outer Union



MySQL 8.0 Command Line Cli

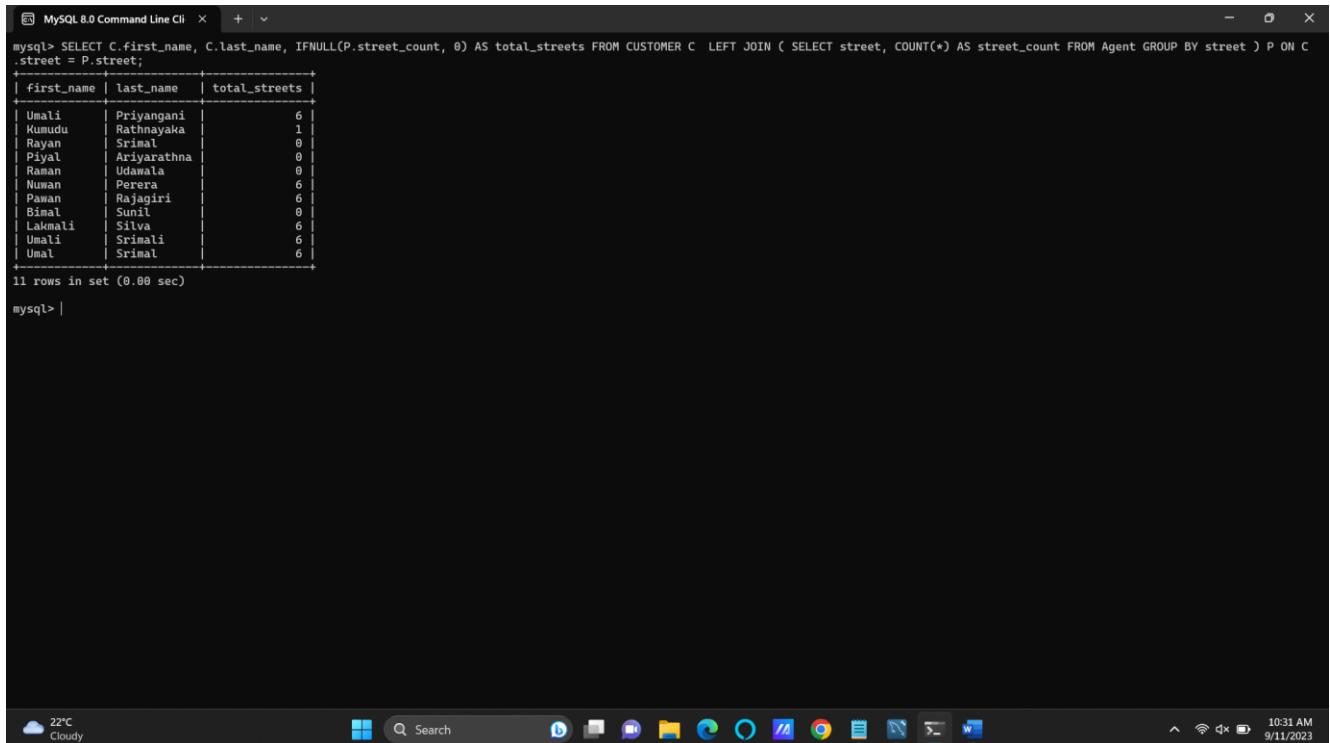
```
mysql> SELECT C.first_name, C.last_name FROM CUSTOMER C WHERE EXISTS ( SELECT * FROM POLICY P WHERE C.street = 'Main Street' ) UNION ALL SELECT C.first_name, C.last_name FROM CUSTOMER C WHERE NOT EXISTS ( SELECT * FROM VEHICLE V WHERE C.customer_ID = V.policy_NO );
+-----+-----+
| first_name | last_name |
+-----+-----+
| Umali      | Priyangani
| Nuwan      | Perera
| Pawan      | Rajagiri
| Lakmali    | Silva
| Umali      | Srimali
| Umal       | Srimal
6 rows in set (0.00 sec)

mysql>
```

Windows taskbar at the bottom:

- Cloudy 22°C
- Search bar
- Icons for File, Print, Copy, Paste, Find, Replace, etc.
- System icons: battery, signal, volume, date/time (10:30 AM, 9/11/2023)

- Nested Subquery with Aggregation

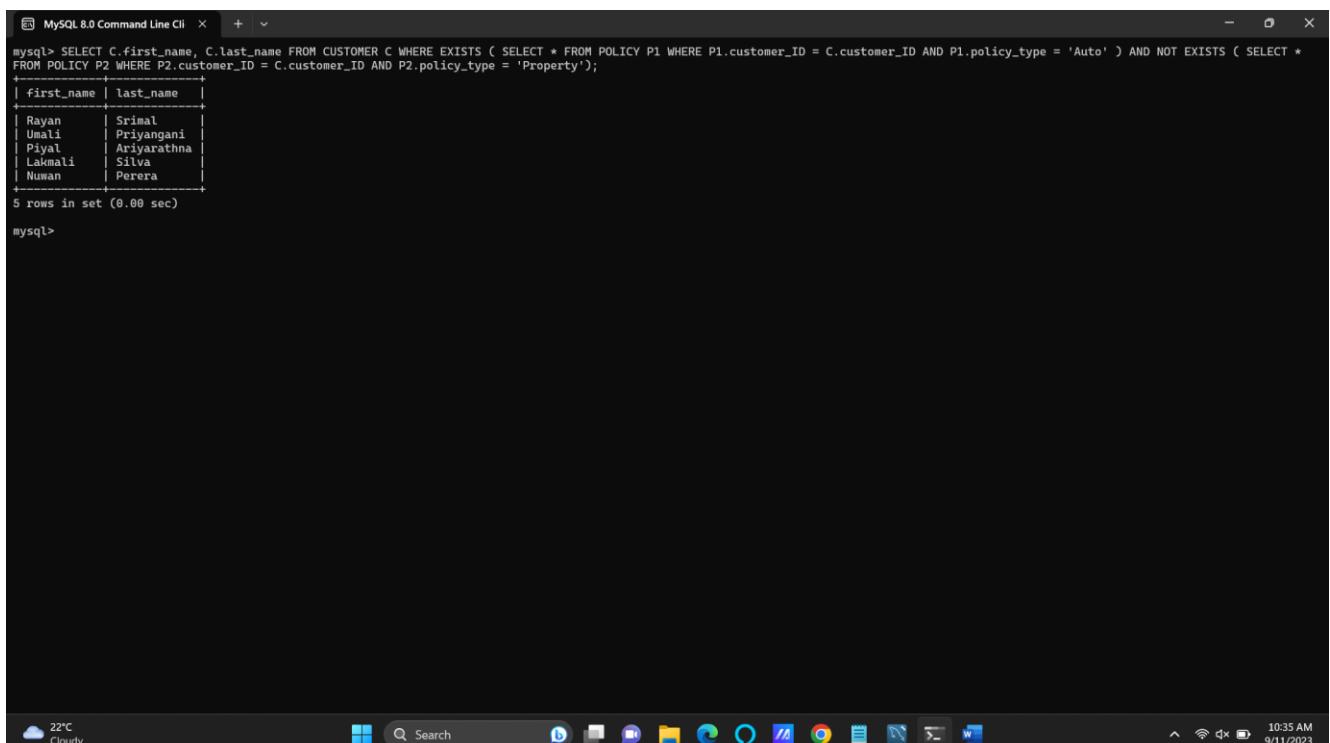


```
MySQL 8.0 Command Line Cli + ×
mysql> SELECT C.first_name, C.last_name, IFNULL(P.street_count, 0) AS total_streets FROM CUSTOMER C LEFT JOIN ( SELECT street, COUNT(*) AS street_count FROM Agent GROUP BY street ) P ON C.street = P.street;
+-----+-----+-----+
| first_name | last_name | total_streets |
+-----+-----+-----+
| Umali      | Priyangan |          6 |
| Kumudu     | Rathnayaka |        1 |
| Rayan      | Srimal    |          0 |
| Piyal      | Ariyarathna |        0 |
| Raman      | Udwala   |          0 |
| Nuwan      | Perera   |          6 |
| Pawan      | Rajagiri  |          6 |
| Bimal      | Sunil    |          0 |
| Lakmali    | Silva    |          6 |
| Umali      | Srimal   |          6 |
| Umal       | Srimal   |          6 |
+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> |
```

The screenshot shows the MySQL 8.0 Command Line Client interface. A query is being run to select customer names and the count of streets they are associated with. The results show 11 rows, indicating each customer has 6 streets. The system tray at the bottom shows it's 10:31 AM on 9/11/2023.

- Nested Subquery with Set Difference



```
MySQL 8.0 Command Line Cli + ×
mysql> SELECT C.first_name, C.last_name FROM CUSTOMER C WHERE EXISTS ( SELECT * FROM POLICY P1 WHERE P1.customer_ID = C.customer_ID AND P1.policy_type = 'Auto' ) AND NOT EXISTS ( SELECT * FROM POLICY P2 WHERE P2.customer_ID = C.customer_ID AND P2.policy_type = 'Property' );
+-----+-----+
| first_name | last_name |
+-----+-----+
| Rayan      | Srimal    |
| Umali      | Priyangan |
| Piyal      | Ariyarathna |
| Lakmali    | Silva    |
| Nuwan      | Perera   |
+-----+-----+
5 rows in set (0.00 sec)

mysql> |
```

The screenshot shows the MySQL 8.0 Command Line Client interface. A query is being run to find customers who have an 'Auto' policy but no 'Property' policy. The results show 5 rows, corresponding to the customers Rayan, Umali, Piyal, Lakmali, and Nuwan. The system tray at the bottom shows it's 10:35 AM on 9/11/2023.

- Nested Subquery with Union

```
MySQL 8.0 Command Line Cli x + v
mysql> SELECT C.first_name, C.last_name, C.street, C.province FROM CUSTOMER C WHERE C.first_name IN ( SELECT A.first_name FROM Agent A WHERE A.street ='Main Street' UNION SELECT P2.first_name FROM Agent P2 WHERE P2.province = 'Southern');
+-----+-----+-----+-----+
| first_name | last_name | street | province |
+-----+-----+-----+-----+
| Kumudu | Rathnayaka | No234 | Southern |
| Piyal | Ariyarathna | No5 | Southern |
| Pawan | Rajagiri | Main Street | Uva |
| Bimal | Sunil | NO2 | Uwa |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```



Chapter 6 - Database Tuning

Tuning of 10 selected complex queries that have written in chapter 5 is tuned using indexing technique. Following steps are taken in query tuning for each complex query in order to provide a clear understanding.

- Drop existing indexes that are externally created in used tables
- Show indexes of table before creating suitable index (using SHOW INDEX command)
- Show number of accessed rows before creating suitable index (using EXPLAIN command)
- Create suitable index in order to tune the query
- Show indexes of table after creating suitable index (using SHOW INDEX command)
- Show number of accessed rows after creating suitable index (using EXPLAIN command)

Tunning of a query is shown by comparing number of accessed rows in explain table, before creating suitable index and after creating that index. If there is a decrement of accessed rows after creating index, we can say that the query is tuned properly. In all screenshots that are included below satisfy that requirement.

- Union Operation

The screenshot shows a MySQL 8.0 Command Line Client window. The session starts with the command 'use ins_prj;'. Then, it runs 'show index from policy;' which displays the following table:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
policy	0	PRIMARY	1	policy_NO	A	11	NULL	NULL	YES	BTREE			YES	NULL
policy	1	FV_2	1	customer_ID	A	8	NULL	NULL	YES	BTREE			YES	NULL

Next, the command 'explain(SELECT policy_NO, policy_type FROM POLICY where policy_type = 'life' > UNION > SELECT policy_NO, relationship FROM LIFE where relationship = 'child');' is run, resulting in the following EXPLAIN output:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	POLICY	NULL	ALL	NULL	NULL	NULL	NULL	11	10.00	Using where
2	UNION	LIFE	NULL	ALL	NULL	NULL	NULL	NULL	2	50.00	Using where
3	UNION RESULT	<union1,2>	NULL	ALL	NULL	NULL	NULL	NULL	NULL	NULL	Using temporary

Finally, the command 'create index type_policy on policy(policy_type);' is run, followed by another 'explain' command which shows a significant reduction in the number of rows scanned (from 60 to 3), indicating the query has been tuned successfully.

• Intersection Operation

```

MySQL> show index from policy;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| policy |          0 | PRIMARY  |          1 | policy_ID   | A           |          8 |          1 |      YES |      YES | BTREE    |          |          | YES | NULL   |
| policy |          1 | FK_2     |          1 | customer_ID | A           |          8 |          1 |      YES |      YES | BTREE    |          |          | YES | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> explain SELECT C.first_name, C.last_name
-- FROM CUSTOMER C
-- WHERE C.customer_ID IN (
-- SELECT P1.customer_ID
-- FROM POLICY P1
-- WHERE P1.policy_type = 'Auto'
-- );
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where | |
| 1 | PRIMARY     | C         | NULL       | eq_ref | PRIMARY     | PRIMARY | 8      | <subquery>.customer_ID | 1     | 100.00 | Using where |
| 2 | MATERIALIZED | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 3 | INTERSECT    | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 3 | INTERSECT    | C2        | NULL       | eq_ref  | PRIMARY     | PRIMARY | 8      | <subquery>.customer_ID | 1     | 100.00 | Using where |
| 4 | MATERIALIZED | C2        | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 5 | INTERSECT RESULT | <intersect1> | NULL       | ALL    | NULL        | NULL  | NULL    | NULL | 100.00 | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set, 1 warning (0.00 sec)

mysql> create index type_policy on policy(policy_type);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain SELECT C.first_name, C.last_name
-- FROM CUSTOMER C
-- WHERE C.customer_ID IN (
-- SELECT P1.customer_ID
-- FROM POLICY P1
-- WHERE P1.policy_type = 'Auto'
-- );
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where | |
| 1 | PRIMARY     | C         | NULL       | eq_ref | PRIMARY     | PRIMARY | 8      | <subquery>.customer_ID | 1     | 100.00 | Using where |
| 2 | MATERIALIZED | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 3 | INTERSECT    | <subquery> | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 3 | INTERSECT    | C2        | NULL       | eq_ref  | PRIMARY     | PRIMARY | 8      | <subquery>.customer_ID | 1     | 100.00 | Using where |
| 4 | MATERIALIZED | C2        | NULL       | ALL    | PRIMARY     | NULL  | NULL    | NULL | 100.00 | Using where |
| 5 | INTERSECT RESULT | <intersect1> | NULL       | ALL    | NULL        | NULL  | NULL    | NULL | 100.00 | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set, 1 warning (0.00 sec)

mysql>

```

• Set Difference

```

MySQL> show index from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| customer |          0 | PRIMARY  |          1 | customer_ID | A           |          8 |          1 |      YES |      YES | BTREE    |          |          | YES | NULL   |
| customer |          1 | FK_1     |          1 | agent_ID    | A           |          5 |          1 |      YES |      YES | BTREE    |          |          | YES | NULL   |
| customer |          1 | FK_12    |          1 | agent_as_customer | A |          2 |          1 |      YES |      YES | BTREE    |          |          | YES | NULL   |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain SELECT city,district  FROM CUSTOMER where district ='Galle'
-- EXCEPT
-- SELECT policy_type, term_price FROM Policy where term_price = '25000';
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | CUSTOMER | NULL       | ALL    | NULL        | NULL  | NULL    | NULL | 8      | 12.50  | Using where |
| 2 | EXCEPT      | Policy   | NULL       | ALL    | NULL        | NULL  | NULL    | NULL | 11     | 10.00  | Using where |
| 3 | EXCEPT RESULT | <except1> | NULL       | ALL    | NULL        | NULL  | NULL    | NULL | NULL   | NULL   | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> create index customer_district on customer(district);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain(SELECT city,district  FROM CUSTOMER where district ='Galle'
-- EXCEPT
-- SELECT policy_type, term_price FROM Policy where term_price = '25000');
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | CUSTOMER | NULL       | ref   | customer_district | customer_district | 403 | const  | 3     | 100.00 | NULL   |
| 2 | EXCEPT      | Policy   | NULL       | ALL   | NULL        | NULL  | NULL    | NULL | 11     | 10.00  | Using where |
| 3 | EXCEPT RESULT | <except1> | NULL       | ALL   | NULL        | NULL  | NULL    | NULL | NULL   | NULL   | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.01 sec)

mysql>

```

- Division Operation

```
MySQL 8.0 Command Line Cli + X

mysql> show index from policy;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| policy |          0 | PRIMARY |            1 | policy_ID | A           |       11 |        1 |      YES | BTREE |          |          |          | YES |      MALL |
| policy |          1 | FN_2     |            1 | customer_ID | A           |        8 |        1 |      YES | BTREE |          |          |          | YES |      MALL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> explain SELECT C.first_name, C.last_name
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | C      |          1 | ALL    | NULL         | NULL |      100 |      | 100.00 | Using where | |
| 2 | DEPENDENT SUBQUERY | P      |          1 | ALL    | NULL         | NULL |      100 |      | 10.00  | Using where |
| 3 | DEPENDENT SUBQUERY | CP     |          1 | ALL    | ref          | P     |      83 | ins_prj.C.customer_ID | 1 | 100.00 | Using where; Not exists |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 3 warnings (0.00 sec)

mysql> create index type_idx on policy(policy_type);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain SELECT C.first_name, C.last_name
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | C      |          1 | ALL    | NULL         | NULL |      100 |      | 100.00 | Using where | |
| 2 | DEPENDENT SUBQUERY | P      |          1 | ALL    | NULL         | NULL |      100 |      | 10.00  | Using where |
| 3 | DEPENDENT SUBQUERY | CP     |          1 | ALL    | ref          | P     |      83 | ins_prj.C.customer_ID | 1 | 100.00 | Using where; Not exists |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql>
```

- Inner Join

```
MySQL 8.0 Command Line Cli + X

mysql> show index from Agent;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| agent |          0 | PRIMARY |            1 | agent_ID | A           |       5 |        1 |      YES | BTREE |          |          |          | YES |          |
| agent |          1 | NULL     |            1 | NULL     | NULL        |      100 |        1 |      YES | BTREE |          |          |          | YES |          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql> explain(SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> INNER JOIN AGENT A ON C.province = A.province);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | A      |          1 | ALL    | NULL         | NULL |      100 |      | 100.00 | NULL |
| 1 | SIMPLE      | C      |          1 | ALL    | NULL         | NULL |      100 |      | 12.50  | Using where; Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> create index agent_province on agent(province);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> explain(SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> INNER JOIN AGENT A ON C.province = A.province);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | C      |          1 | ALL    | NULL         | NULL |      100 |      | 100.00 | Using where |
| 1 | SIMPLE      | A      |          1 | ref    | agent_province | agent_province | 403 | ins_prj.C.province | 1 | 100.00 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

- Left Outer Join

```
MySQL> show index from property;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| property | 0 | PRIMARY | 1 | property_ID | A | 2 | NULL | NULL | YES | BTREE |  |  | YES | NULL |
| property | 1 | FK_7 | 1 | policy_NO | A | 2 | NULL | NULL | YES | BTREE |  |  | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MySQL> explain(SELECT C.first_name, C.last_name, P.property_type
    -> FROM PROPERTY P
    -> LEFT JOIN CUSTOMER C ON C.province = P.province);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | P | NULL | ALL | NULL | NULL | NULL | NULL | 2 | 100.00 | NULL |
| 1 | SIMPLE | C | NULL | ALL | NULL | NULL | NULL | NULL | 8 | 100.00 | Using where; Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

MySQL> create index pro_pro on property(province);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

MySQL> explain(SELECT C.first_name, C.last_name, P.property_type
    -> FROM PROPERTY P
    -> LEFT JOIN CUSTOMER C ON C.province = P.province);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | P | NULL | ALL | NULL | NULL | NULL | NULL | 2 | 100.00 | NULL |
| 1 | SIMPLE | C | NULL | ALL | NULL | NULL | NULL | NULL | 8 | 100.00 | Using where; Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

MySQL>
```

- Outer Union

```
MySQL> show index from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| customer | 0 | PRIMARY | 1 | customer_ID | A | 8 | NULL | NULL | YES | BTREE |  |  | YES | NULL |
| customer | 1 | FK_3 | 1 | agent_ID | A | 5 | NULL | NULL | YES | BTREE |  |  | YES | NULL |
| customer | 1 | FK_12 | 1 | agent_as_customer | A | 2 | NULL | NULL | YES | BTREE |  |  | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

MySQL> explain(SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> WHERE EXISTS (
    ->     SELECT *
    ->     FROM POLICY P
    ->     WHERE C.street = 'Main Street'
    -> )
    -> UNION ALL
    -> SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> WHERE EXISTS (
    ->     SELECT *
    ->     FROM VEHICLE V
    ->     WHERE C.customer_ID = V.policy_NO
    -> ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | C | NULL | ALL | NULL | NULL | NULL | NULL | 8 | 12.50 | Using where |
| 1 | PRIMARY | P | NULL | index | FK_2 | 83 | NULL | NULL | 11 | 100.00 | Using index; FirstMatch(C); Using join buffer (hash join) |
| 3 | UNION | C | NULL | index | FK_3 | 83 | NULL | NULL | 5 | 100.00 | Using where; Using index; LooseScan |
| 3 | UNION | V | NULL | eq_ref | PRIMARY | 82 | ins_prj.V.policy_NO | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

MySQL> create index cus_street on customer(street);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

MySQL> explain(SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> WHERE EXISTS (
    ->     SELECT *
    ->     FROM POLICY P
    ->     WHERE C.street = 'Main Street'
    -> )
    -> UNION ALL
    -> SELECT C.first_name, C.last_name
    -> FROM CUSTOMER C
    -> WHERE EXISTS (
    ->     SELECT *
    ->     FROM VEHICLE V
    ->     WHERE C.customer_ID = V.policy_NO
    -> ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | C | NULL | ALL | <subquery> cus_street | cus_street | 1 | 1 | 5 | 100.00 | Using index condition | |
| 1 | PRIMARY | <subquery> | NULL | eq_ref | <auto_distinct_key> | auto_distinct_key | 46 | ins_prj.C.street | 1 | 1 | 100.00 | NULL |
| 2 | MATERIALIZED | P | NULL | index | NUL | 83 | NULL | NULL | 11 | 100.00 | Using index |
| 3 | UNION | V | NULL | index | FK_3 | 83 | NULL | NULL | 5 | 100.00 | Using where; Using index; LooseScan |
| 3 | UNION | C | NULL | eq_ref | PRIMARY | 82 | ins_prj.V.policy_NO | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 3 warnings (0.00 sec)

MySQL>
```

- Nested Subquery with Set Difference

```
mysql> show index from policy;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| policy |          0 | PRIMARY |            1 | policy_ID | A           |       11 |          1 |      YES | BTREE |          |          |          | YES | NULL |
| policy |          1 | FK_2    |            1 | customer_ID | A           |        5 |          1 |      YES | BTREE |          |          |          | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> explain(SELECT C.first_name, C.last_name
   > FROM CUSTOMER C
   > WHERE EXISTS (
   >   > SELECT *
   >   > FROM POLICY P1
   >   > WHERE P1.customer_ID = C.customer_ID AND P1.policy_type = 'Auto'
   >   ) AND NOT EXISTS (
   >   > SELECT *
   >   > FROM POLICY P2
   >   > WHERE P2.customer_ID = C.customer_ID AND P2.policy_type = 'Property'
   >   ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | <subquery2> | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 11   | 100.00    | Using where |
| 1 | SIMPLE     | C       | NULL    | eq_ref | PRIMARY     | PRIMARY | 82     | <subquery2>.customer_ID | 1   | 100.00    | NULL |
| 1 | SIMPLE     | P2     | NULL    | ref  | FK_2        | FK_2   | 83     | <subquery2>.customer_ID | 1   | 100.00    | Using where; Not exists |
| 2 | MATERIALIZED| P1     | NULL    | ALL  | FK_2        | NULL   | NULL    | NULL | 11   | 10.00     | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql> create index type_policy on policy(policy_type);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> explain(SELECT C.first_name, C.last_name
   > FROM CUSTOMER C
   > WHERE EXISTS (
   >   > SELECT *
   >   > FROM POLICY P1
   >   > WHERE P1.customer_ID = C.customer_ID AND P1.policy_type = 'Auto'
   >   ) AND NOT EXISTS (
   >   > SELECT *
   >   > FROM POLICY P2
   >   > WHERE P2.customer_ID = C.customer_ID AND P2.policy_type = 'Property'
   >   ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE     | <subquery2> | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 11   | 100.00    | Using where |
| 1 | SIMPLE     | C       | NULL    | eq_ref | PRIMARY     | PRIMARY | 82     | <subquery2>.customer_ID | 1   | 100.00    | NULL |
| 1 | SIMPLE     | P2     | NULL    | ref  | FK_2        | FK_2   | 83     | <subquery2>.customer_ID | 1   | 100.00    | Using where; Not exists |
| 2 | MATERIALIZED| P1     | NULL    | ALL  | FK_2        | NULL   | NULL    | CONST | 5    | 100.00    | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 3 warnings (0.00 sec)

mysql>
```

- Nested Subquery with Union

```
mysql> show index from customer;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| customer |          0 | PRIMARY |            1 | customer_ID | A           |       8 |          1 |      YES | BTREE |          |          |          | YES | NULL |
| customer |          1 | FK_1    |            1 | agent_ID    | A           |        5 |          1 |      YES | BTREE |          |          |          | YES | NULL |
| customer |          1 | FK_12   |            1 | agent_as_customer | A           |        2 |          1 |      YES | BTREE |          |          |          | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain(SELECT C.first_name, C.last_name, C.street, C.province
   > FROM CUSTOMER C
   > WHERE C.first_name IN (
   >   > SELECT A.first_name
   >   > FROM Agent A
   >   > WHERE A.street = 'Main Street'
   >   UNION
   >   > SELECT P2.first_name
   >   > FROM Agent P2
   >   > WHERE P2.province = 'Southern'
   >   ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | C       | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 8    | 100.00    | Using where |
| 2 | DEPENDENT SUBQUERY | A       | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 5    | 20.00     | Using where |
| 3 | DEPENDENT UNION  | P2     | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 5    | 20.00     | Using where |
| 4 | UNION RESULT  | <union2,3> | NULL    | ALL  | NULL         | NULL | NULL    | NULL | NULL | NULL     | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql> create index f_name on customer(first_name);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> explain(SELECT C.first_name, C.last_name, C.street, C.province
   > FROM CUSTOMER C
   > WHERE C.first_name IN (
   >   > SELECT A.first_name
   >   > FROM Agent A
   >   > WHERE A.street = 'Main Street'
   >   UNION
   >   > SELECT P2.first_name
   >   > FROM Agent P2
   >   > WHERE P2.province = 'Southern'
   >   ));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY     | C       | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 8    | 100.00    | Using where |
| 2 | DEPENDENT SUBQUERY | A       | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 5    | 20.00     | Using where |
| 3 | DEPENDENT UNION  | P2     | NULL    | ALL  | NULL         | NULL | NULL    | NULL | 5    | 20.00     | Using where |
| 4 | UNION RESULT  | <union2,3> | NULL    | ALL  | NULL         | NULL | NULL    | NULL | NULL | NULL     | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set, 1 warning (0.00 sec)

mysql>
```

- Nested Subquery with Aggregation

```

MySQL 8.0 Command Line Cli  +  X

mysql> show index from agent;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| agent |          0 | PRIMARY |          1 | agent_ID | A           |          5 |        NULL |    YES |    YES | BTREE   |          |          | YES |      NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> explain(SELECT C.first_name, C.last_name, IFNULL(P.street_count, 0) AS total_streets
   >   FROM CUSTOMER C
   >   LEFT JOIN (
   >     >   SELECT street, COUNT(*) AS street_count
   >     >   FROM Agent
   >     >   GROUP BY street
   >   ) P ON C.street = P.street);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY    | C     |          1 | ALL  | <derived2> | <auto_key0> |       83 | NULL |    8 |    100.00 | NULL |
| 1 | PRIMARY    | <derived2> |          1 | ref   | <auto_key0> | <auto_key0> |       83 | ins_prj.C.street |    2 |    100.00 | NULL |
| 2 | DERIVED    | Agent |          1 | ALL  | <derived2> | <auto_key0> |       83 | NULL |    5 |    100.00 | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> create index a_street on agent(street);
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> explain(SELECT C.first_name, C.last_name, IFNULL(P.street_count, 0) AS total_streets
   >   FROM CUSTOMER C
   >   LEFT JOIN (
   >     >   SELECT street, COUNT(*) AS street_count
   >     >   FROM Agent
   >     >   GROUP BY street
   >   ) P ON C.street = P.street);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY    | C     |          1 | ALL  | <derived2> | <auto_key0> |       83 | NULL |    8 |    100.00 | NULL |
| 1 | PRIMARY    | <derived2> |          1 | ref   | <auto_key0> | <auto_key0> |       83 | ins_prj.C.street |    2 |    100.00 | NULL |
| 2 | DERIVED    | Agent |          1 | index | a_street    | a_street  |       83 | NULL |    5 |    100.00 | Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> |

```