

# Insurance Management & System Using SQL

---

## Project Documentation

Done By : Divakaran S
-----------------------

### 1. Project Overview

This project models and analyzes vehicle insurance data using SQL. It covers user information, vehicles, brokers, quotes, and premium transactions. The objective is to derive insights such as active customers, premium distribution, and broker activities.

### 2. Tools & Technologies Used

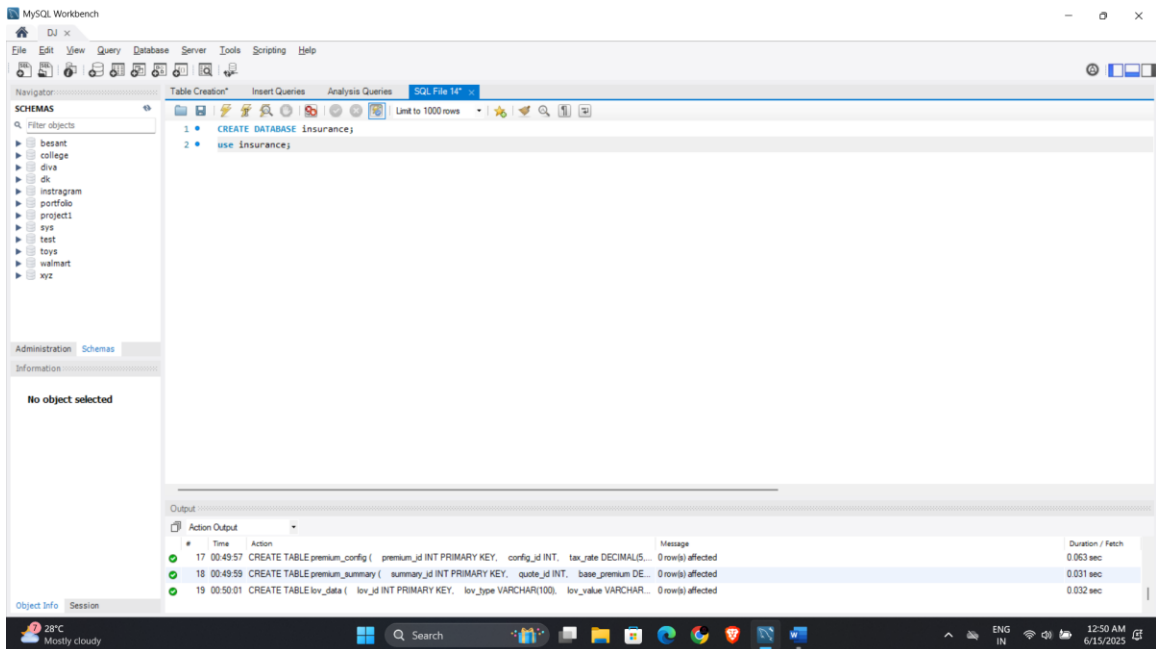
- MySQL
- SQL Workbench

### 3. Database Table Descriptions

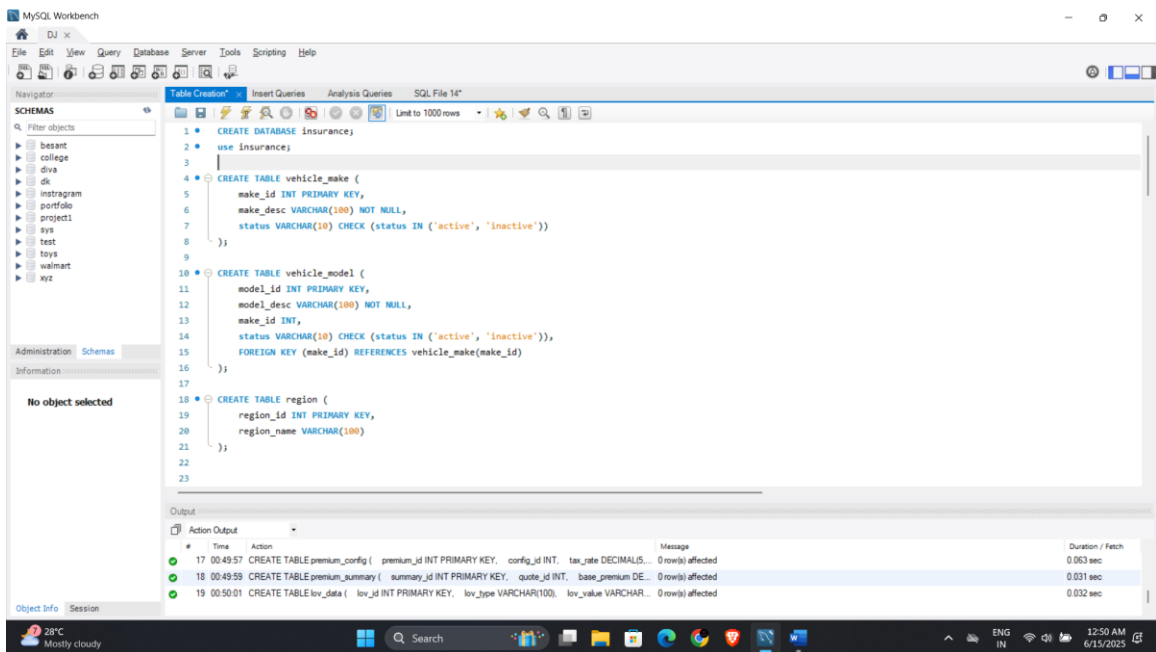
Table Name	Description
vehicle_make	Vehicle manufacturers
vehicle_model	Vehicle models linked to manufacturers
region	Geographical regions
state	Indian states linked to regions
city	Cities linked to states
product_config	Insurance products and premiums
user_personal_info	User profile and status
login_user	Login credentials and user status
broker_info	Insurance brokers and their organizations
quote_info	Quotes for insurance issued to users
premium_config	Tax and surcharge configuration
premium_summary	Premium calculations and totals
lov_data	List of values for dropdowns

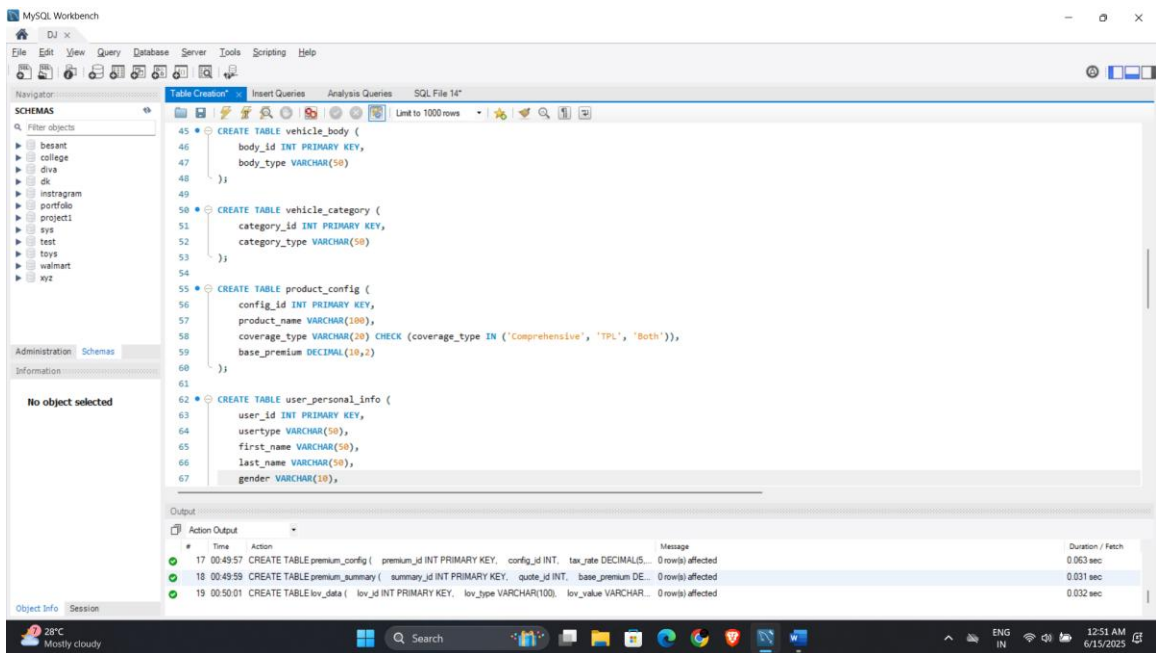
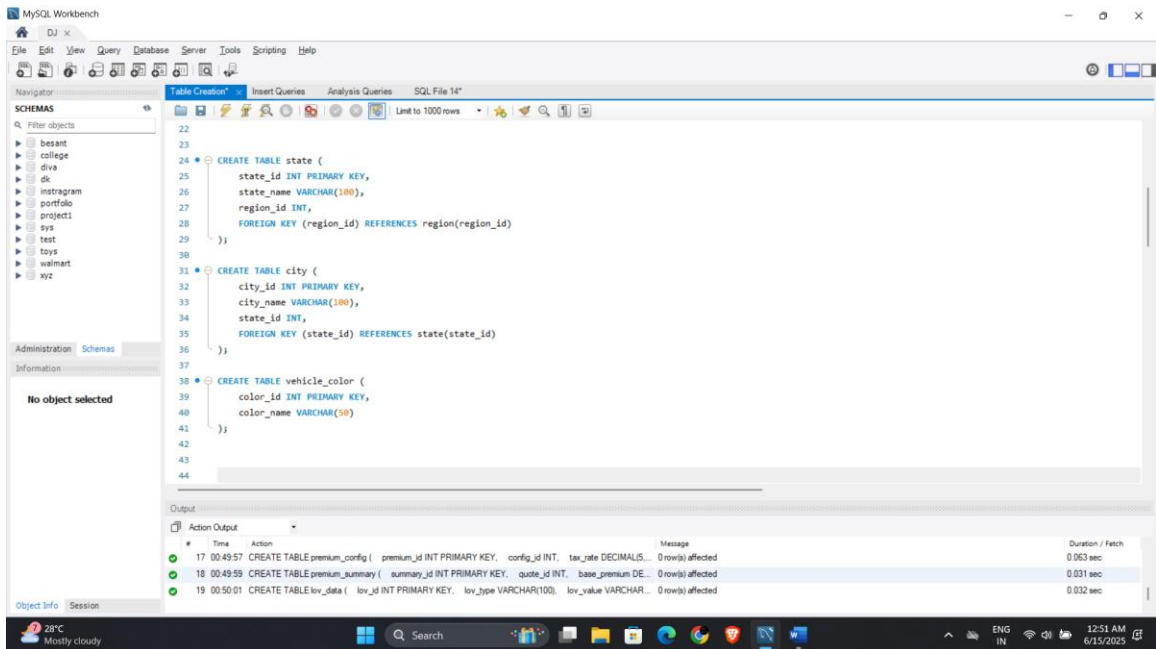
### 4. Database Schema

Database Creation



## Table Creation





MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

- besant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas

Information

No object selected

Object Info Session

Table Creation

```
80 CREATE TABLE login_user (  
81 login_id INT PRIMARY KEY,  
82 password VARCHAR(255),  
83 user_id INT,  
84 usertype VARCHAR(50) NOT NULL,  
85 status VARCHAR(10) CHECK (status IN ('active', 'inactive')),  
86 FOREIGN KEY (user_id) REFERENCES user_personal_info(user_id)  
87 );  
88  
89 CREATE TABLE broker_info (  
90 broker_id INT PRIMARY KEY,  
91 broker_name VARCHAR(100),  
92 broker_org_name VARCHAR(100),  
93 address VARCHAR(150),  
94 status VARCHAR(10) CHECK (status IN ('active', 'inactive'))  
95 );  
96  
97 CREATE TABLE quote_info (  
98 quote_id INT PRIMARY KEY,  
99 user_id INT,  
100 vehicle_id INT,  
101 config_id INT,  
102 quote_date DATE,  
103 final_premium DECIMAL(10,2),  
104 status VARCHAR(10) CHECK (status IN ('active', 'inactive')),  
105 FOREIGN KEY (user_id) REFERENCES user_personal_info(user_id),  
106 FOREIGN KEY (config_id) REFERENCES product_config(config_id)  
107 );  
108  
109 CREATE TABLE premium_config (  
110 premium_id INT PRIMARY KEY,  
111 config_id INT,  
112 tax_rate DECIMAL(5,2),  
113 surcharge DECIMAL(5,2),  
114 FOREIGN KEY (config_id) REFERENCES product_config(config_id)  
115 );  
116  
117 CREATE TABLE premium_summary (  
118 summary_id INT PRIMARY KEY,  
119 quote_id INT,  
120 base_premium DECIMAL(10,2),  
121 tax_amount DECIMAL(10,2),  
122 total_premium DECIMAL(10,2),  
123 FOREIGN KEY (quote_id) REFERENCES quote_info(quote_id)  
124 );  
125  
126 CREATE TABLE lov_data (  
127 lov_id INT PRIMARY KEY,  
128 lov_type VARCHAR(100),  
129 lov_value VARCHAR(100)  
130 );  
131
```

28°C Mostly cloudy 12:51 AM 6/15/2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHMAS

Filter objects

- besant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas

Information

No object selected

Object Info Session

Table Creation

```
102 quote_date DATE,  
103 final_premium DECIMAL(10,2),  
104 status VARCHAR(10) CHECK (status IN ('active', 'inactive')),  
105 FOREIGN KEY (user_id) REFERENCES user_personal_info(user_id),  
106 FOREIGN KEY (config_id) REFERENCES product_config(config_id)  
107 );  
108  
109 CREATE TABLE premium_config (  
110 premium_id INT PRIMARY KEY,  
111 config_id INT,  
112 tax_rate DECIMAL(5,2),  
113 surcharge DECIMAL(5,2),  
114 FOREIGN KEY (config_id) REFERENCES product_config(config_id)  
115 );  
116  
117 CREATE TABLE premium_summary (  
118 summary_id INT PRIMARY KEY,  
119 quote_id INT,  
120 base_premium DECIMAL(10,2),  
121 tax_amount DECIMAL(10,2),  
122 total_premium DECIMAL(10,2),  
123 FOREIGN KEY (quote_id) REFERENCES quote_info(quote_id)  
124 );  
125  
126 CREATE TABLE lov_data (  
127 lov_id INT PRIMARY KEY,  
128 lov_type VARCHAR(100),  
129 lov_value VARCHAR(100)  
130 );  
131
```

28°C Mostly cloudy 12:51 AM 6/15/2025

## Insert Queries

The screenshot shows the MySQL Workbench interface with a SQL script for inserting data into two tables: `vehicle_make` and `vehicle_model`. The script is as follows:

```
1 -- INSERTS FOR vehicle_make
2 * INSERT INTO vehicle_make VALUES (1, 'Tata Motors', 'active');
3 * INSERT INTO vehicle_make VALUES (2, 'Mahindra & Mahindra', 'active');
4 * INSERT INTO vehicle_make VALUES (3, 'Maruti Suzuki', 'active');
5 * INSERT INTO vehicle_make VALUES (4, 'Hyundai India', 'active');
6 * INSERT INTO vehicle_make VALUES (5, 'Honda Cars India', 'inactive');
7 * INSERT INTO vehicle_make VALUES (6, 'Toyota India', 'active');
8 * INSERT INTO vehicle_make VALUES (7, 'Kia Motors', 'active');
9 * INSERT INTO vehicle_make VALUES (8, 'Renault India', 'inactive');
10 * INSERT INTO vehicle_make VALUES (9, 'Nissan India', 'active');
11 * INSERT INTO vehicle_make VALUES (10, 'Volkswagen India', 'inactive');
12 * INSERT INTO vehicle_make VALUES (11, 'Skoda India', 'active');
13 * INSERT INTO vehicle_make VALUES (12, 'MG Motor', 'active');
14 * INSERT INTO vehicle_make VALUES (13, 'Ford India', 'inactive');
15 * INSERT INTO vehicle_make VALUES (14, 'Isuzu India', 'active');
16 * INSERT INTO vehicle_make VALUES (15, 'Jeep India', 'active');
17
18 -- INSERTS FOR vehicle_model
19 * INSERT INTO vehicle_model VALUES (1, 'Nexon', 1, 'active');
20 * INSERT INTO vehicle_model VALUES (2, 'Thar', 2, 'active');
21 * INSERT INTO vehicle_model VALUES (3, 'Swift', 3, 'active');
22 * INSERT INTO vehicle_model VALUES (4, 'Creta', 4, 'active');
```

The Output pane shows the execution results for the first four queries:

#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

The screenshot shows the MySQL Workbench interface with a SQL script for inserting data into three tables: `region`, `state`, and `vehicle_model`. The script is as follows:

```
28 * INSERT INTO vehicle_model VALUES (10, 'Polo', 10, 'active');
29 * INSERT INTO vehicle_model VALUES (11, 'Rapid', 11, 'active');
30 * INSERT INTO vehicle_model VALUES (12, 'Hector', 12, 'inactive');
31 * INSERT INTO vehicle_model VALUES (13, 'EcoSport', 13, 'inactive');
32 * INSERT INTO vehicle_model VALUES (14, 'D-Nav', 14, 'active');
33 * INSERT INTO vehicle_model VALUES (15, 'Compass', 15, 'active');
34
35 -- INSERTS FOR region
36 * INSERT INTO region VALUES (1, 'North');
37 * INSERT INTO region VALUES (2, 'South');
38 * INSERT INTO region VALUES (3, 'East');
39 * INSERT INTO region VALUES (4, 'West');
40
41
42 -- INSERTS FOR state
43 * INSERT INTO state VALUES (1, 'Tamil Nadu', 2);
44 * INSERT INTO state VALUES (2, 'Maharashtra', 4);
45 * INSERT INTO state VALUES (3, 'Delhi', 1);
46 * INSERT INTO state VALUES (4, 'Karnataka', 2);
47 * INSERT INTO state VALUES (5, 'West Bengal', 3);
48 * INSERT INTO state VALUES (6, 'Kerala', 2);
49 * INSERT INTO state VALUES (7, 'Uttar Pradesh', 1);
```

The Output pane shows the execution results for the first four queries (repeated from the previous screenshot):

#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- desant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas

Information

No object selected

Table Creation

```
-- INSERTS FOR city
58
59 INSERT INTO city VALUES (1, 'Chennai', 1);
60 INSERT INTO city VALUES (2, 'Mumbai', 2);
61 INSERT INTO city VALUES (3, 'Delhi', 3);
62 INSERT INTO city VALUES (4, 'Bangalore', 4);
63 INSERT INTO city VALUES (5, 'Kolkata', 5);
64 INSERT INTO city VALUES (6, 'Kochi', 6);
65 INSERT INTO city VALUES (7, 'Lucknow', 7);
66 INSERT INTO city VALUES (8, 'Patna', 8);
67 INSERT INTO city VALUES (9, 'Ahmedabad', 9);
68 INSERT INTO city VALUES (10, 'Jaipur', 10);
69 INSERT INTO city VALUES (11, 'Amaritson', 11);
70 INSERT INTO city VALUES (12, 'Gurgaon', 12);
71 INSERT INTO city VALUES (13, 'Vijayawada', 13);
72 INSERT INTO city VALUES (14, 'Bhubaneswar', 14);
73
74
75 -- INSERTS FOR vehicle_color
76 INSERT INTO vehicle_color VALUES (1, 'Red');
77 INSERT INTO vehicle_color VALUES (2, 'Blue');
78 INSERT INTO vehicle_color VALUES (3, 'Green');
79 INSERT INTO vehicle_color VALUES (4, 'Black');
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

Object Info Session

Hot days ahead 28°C

Search

ENG IN 12:58 AM 6/15/2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- desant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas

Information

No object selected

Table Creation

```
-- INSERTS FOR vehicle_body
82
83 INSERT INTO vehicle_body VALUES (1, 'Sedan');
84 INSERT INTO vehicle_body VALUES (2, 'SUV');
85 INSERT INTO vehicle_body VALUES (3, 'Hatchback');
86 INSERT INTO vehicle_body VALUES (4, 'Convertible');
87 INSERT INTO vehicle_body VALUES (5, 'Coupe');
88
89 -- INSERTS FOR vehicle_category
90 INSERT INTO vehicle_category VALUES (1, 'Private');
91 INSERT INTO vehicle_category VALUES (2, 'Commercial');
92 INSERT INTO vehicle_category VALUES (3, 'Luxury');
93 INSERT INTO vehicle_category VALUES (4, 'Transport');
94 INSERT INTO vehicle_category VALUES (5, 'Tourist');
95
96 -- INSERTS FOR product_config
97 INSERT INTO product_config VALUES (1, 'Standard Plan', 'Comprehensive', 4500.00);
98 INSERT INTO product_config VALUES (2, 'Economy Plan', 'TPL', 3200.00);
99 INSERT INTO product_config VALUES (3, 'Premium Plan', 'Both', 8000.00);
100 INSERT INTO product_config VALUES (4, 'Urban Plus', 'Comprehensive', 7000.00);
101 INSERT INTO product_config VALUES (5, 'Rural Saver', 'TPL', 2800.00);
102
103 -- INSERTS FOR lov_data
104
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

Object Info Session

Hot days ahead 28°C

Search

ENG IN 12:58 AM 6/15/2025



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- desant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas Information

No object selected

Table Creation

```
-- INSERTS FOR lov_data
103
104 INSERT INTO lov_data VALUES (1, 'MaritalStatus', 'Single');
105 INSERT INTO lov_data VALUES (2, 'MaritalStatus', 'Married');
106 INSERT INTO lov_data VALUES (3, 'Education', 'Graduate');
107 INSERT INTO lov_data VALUES (4, 'Education', 'Post-Graduate');
108 INSERT INTO lov_data VALUES (5, 'Education', 'Diploma');
109
110
111
112 -- INSERTS FOR user_personal_info
113 INSERT INTO user_personal_info VALUES (1, 'Customer', 'Ravi', 'Kumar', 'Male', '1995-06-15', 'ravi.kumar@example.com', '9876543210', '12 PG Road, Chennai', 1, 1, 1, 1234567890);
114 INSERT INTO user_personal_info VALUES (2, 'Customer', 'Anjali', 'Verma', 'Female', '1990-02-20', 'anjali.verma@example.com', '9876543211', '88 Park Street, Kolkata', 5, 5, 1, 2);
115 INSERT INTO user_personal_info VALUES (3, 'Customer', 'Vikram', 'Singh', 'Male', '1988-09-18', 'vikram.singh@example.com', '9876543212', '22 Residency Rd, Bangalore', 4, 4, 1, 1);
116 INSERT INTO user_personal_info VALUES (4, 'Customer', 'Neha', 'Sharma', 'Female', '1993-12-05', 'neha.sharma@example.com', '9876543213', '44 MI Lane, Delhi', 3, 3, 1, 456789012);
117 INSERT INTO user_personal_info VALUES (5, 'Customer', 'Amit', 'Patel', 'Male', '1992-03-11', 'amit.patel@example.com', '9876543214', '20 Ring Road, Ahmedabad', 9, 9, 1, 5678901);
118 INSERT INTO user_personal_info VALUES (6, 'Customer', 'Deepa', 'Joshi', 'Female', '1987-07-19', 'deepa.joshi@example.com', '9876543215', '32 Sector 18, Noida', 7, 7, 1, 6789012);
119 INSERT INTO user_personal_info VALUES (7, 'Customer', 'Rahul', 'Mishra', 'Male', '1991-01-25', 'rahul.mishra@example.com', '9876543216', '78 Rajaji Nagar, Bengaluru', 4, 4, 1, 1);
120 INSERT INTO user_personal_info VALUES (8, 'Customer', 'Sneha', 'Menon', 'Female', '1985-04-22', 'sneha.menon@example.com', '9876543217', '11 MG Road, Kochi', 6, 6, 1, 890123456);
121 INSERT INTO user_personal_info VALUES (9, 'Customer', 'Arjun', 'Deshmukh', 'Male', '1994-11-08', 'arjun.deshmukh@example.com', '9876543218', '25 Andheri East, Mumbai', 2, 2, 1, 1);
122 INSERT INTO user_personal_info VALUES (10, 'Customer', 'Meena', 'Pillai', 'Female', '1996-08-30', 'meena.pillai@example.com', '9876543219', '17 Kowdiar, Thiruvananthapuram', 6, 6, 1, 1);
123
124 -- INSERTS FOR login_user
125 INSERT INTO login_user VALUES (1, 'pass1234', 1, 'Customer', 'active');
```

Output

#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

Object Info Session

Hot days ahead 28°C

Search

ENG IN 12:58 AM 6/15/2025

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- desant
- college
- diva
- dk
- instagram
- portfolio
- project1
- sys
- test
- toys
- walmart
- xyz

Administration Schemas Information

No object selected

Table Creation

```
-- INSERTS FOR login_user
124
125 INSERT INTO login_user VALUES (1, 'pass1234', 1, 'Customer', 'active');
126 INSERT INTO login_user VALUES (2, 'pass2345', 2, 'Customer', 'active');
127 INSERT INTO login_user VALUES (3, 'pass3456', 3, 'Customer', 'inactive');
128 INSERT INTO login_user VALUES (4, 'pass4567', 4, 'Customer', 'active');
129 INSERT INTO login_user VALUES (5, 'pass5678', 5, 'Customer', 'active');
130 INSERT INTO login_user VALUES (6, 'pass6789', 1, 'Customer', 'active');
131 INSERT INTO login_user VALUES (7, 'pass7890', 2, 'Customer', 'active');
132 INSERT INTO login_user VALUES (8, 'pass8901', 3, 'Customer', 'inactive');
133 INSERT INTO login_user VALUES (9, 'pass9012', 4, 'Customer', 'active');
134 INSERT INTO login_user VALUES (10, 'pass0123', 5, 'Customer', 'active');
135
136 -- INSERTS FOR broker_info
137 INSERT INTO broker_info VALUES (1, 'Suresh Babu', 'SecureBrokers Pvt Ltd', '101 GST Road, Chennai', 'active');
138 INSERT INTO broker_info VALUES (2, 'Priya Mehta', 'Shield Insurance', '55 Nariman Point, Mumbai', 'active');
139 INSERT INTO broker_info VALUES (3, 'Alok Reddy', 'CoverSure India', '78 Brigade Road, Bangalore', 'inactive');
140 INSERT INTO broker_info VALUES (4, 'Divya Kapoor', 'MaxPolicy Agency', '19 CP Circle, Delhi', 'active');
141 INSERT INTO broker_info VALUES (5, 'Rajan Desai', 'TrustBrokers LLP', '16 CG Road, Ahmedabad', 'active');
142
143 -- INSERTS FOR quote_info
144 INSERT INTO quote_info VALUES (1, 1, 1, 1, '2024-06-01', 5200.00, 'active');
145 INSERT INTO quote_info VALUES (2, 2, 3, 2, '2024-06-03', 3400.00, 'active');
```

Output

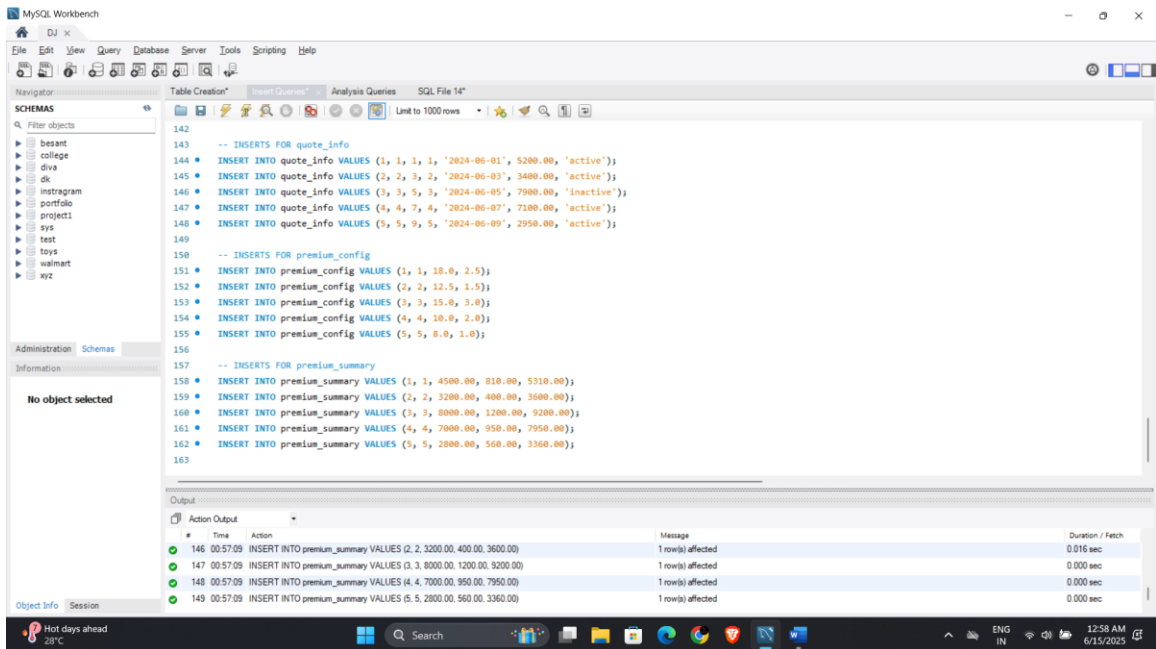
#	Time	Action	Message	Duration / Fetch
146	00:57:09	INSERT INTO premium_summary VALUES (2, 2, 3200.00, 400.00, 3600.00)	1 row(s) affected	0.016 sec
147	00:57:09	INSERT INTO premium_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00)	1 row(s) affected	0.000 sec
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec

Object Info Session

Hot days ahead 28°C

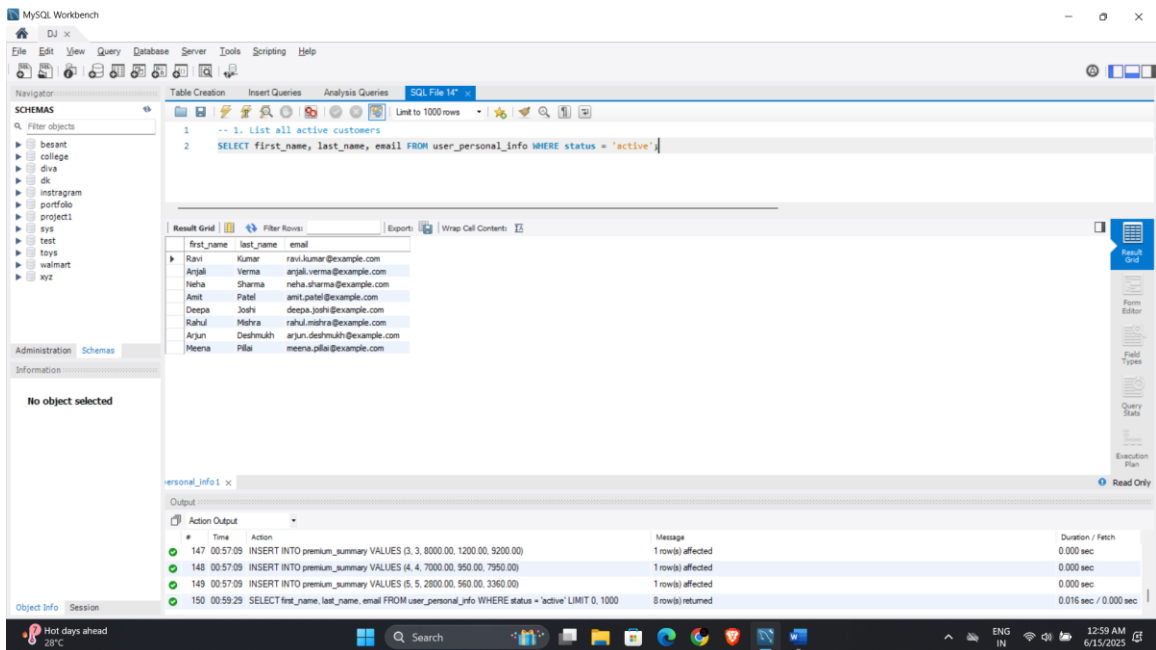
Search

ENG IN 12:58 AM 6/15/2025



## 5. SQL Queries & Insights

### List of All Active Customers





## Vehicle Models by a Specific Make

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 2. Fetch vehicle models by a specific make
SELECT
  vm.model_desc
FROM
  vehicle_model vm
JOIN
  vehicle_make vmk ON vm.make_id = vmk.make_id
WHERE
  vmk.make_desc = 'Tata Motors';
```

The Result Grid shows the output of the query:

model_desc
Nexon

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
148	00:57:09	INSERT INTO premium_summary VALUES (4, 4, 7000.00, 950.00, 7950.00)	1 row(s) affected	0.000 sec
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec
150	00:59:29	SELECT first_name, last_name, email FROM user_personal_info WHERE status = 'active' LIMIT 0, 1000	8 row(s) returned	0.016 sec / 0.000 sec
151	01:00:23	SELECT vm.model_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = v...	1 row(s) returned	0.000 sec / 0.000 sec

## Active Brokers

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 3. Show all brokers operating in active status
SELECT broker_name, broker_org_name FROM broker_info WHERE status = 'active';
```

The Result Grid shows the output of the query:

broker_name	broker_org_name
Suresh Babu	SecureBrokers Pvt.Ltd
Priya Mehta	Shield Insurance
Divya Kapoor	MaxPolicy Agency
Rajan Desai	TrustBrokers LLP

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
149	00:57:09	INSERT INTO premium_summary VALUES (5, 5, 2800.00, 560.00, 3360.00)	1 row(s) affected	0.000 sec
150	00:59:29	SELECT first_name, last_name, email FROM user_personal_info WHERE status = 'active' LIMIT 0, 1000	8 row(s) returned	0.016 sec / 0.000 sec
151	01:00:23	SELECT vm.model_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = v...	1 row(s) returned	0.000 sec / 0.000 sec
152	01:01:49	SELECT broker_name, broker_org_name FROM broker_info WHERE status = 'active' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

## Quotes for a Specific User

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 4. List all quotes for a particular user
SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = 1;
```

The Result Grid shows the following data:

quote_id	final_premium	quote_date
1	5200.00	2024-06-01

The Output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
150	00:59:29	SELECT first_name, last_name, email FROM user_personal_info WHERE status = 'active' LIMIT 0, 1000	8 row(s) returned	0.016 sec / 0.000 sec
151	01:00:23	SELECT vm_model_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = v...	1 row(s) returned	0.000 sec / 0.000 sec
152	01:01:49	SELECT broker_name, broker_org_name FROM broker_info WHERE status = 'active' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
153	01:02:24	SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## Total Premium Paid by Each User

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 5. Get total premium paid by each user
SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid
FROM premium_summary ps
JOIN quote_info qi ON qi.quote_id = ps.quote_id
JOIN user_personal_info up ON qi.user_id = up.user_id
GROUP BY up.user_id;
```

The Result Grid shows the following data:

user_id	first_name	total_paid
1	Ravi	\$310.00
2	Arjun	3600.00
3	Vikram	9200.00
4	Neha	7950.00
5	Amit	3360.00

The Output pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
152	01:01:49	SELECT broker_name, broker_org_name FROM broker_info WHERE status = 'active' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
153	01:02:24	SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
154	01:03:05	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.016 sec / 0.000 sec
155	01:03:20	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.000 sec / 0.000 sec

## Inactive Vehicle Models

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 6. Find vehicles with inactive models
SELECT vm.model_desc FROM vehicle_model vm WHERE status = 'inactive';
```

The Results grid shows the following data:

model_desc
City
Magnite
Hector
EcoSport

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
153	01:02:24	SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
154	01:03:05	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.016 sec / 0.000 sec
155	01:03:20	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.000 sec / 0.000 sec
156	01:04:08	SELECT vm.model_desc FROM vehicle_model vm WHERE status = 'inactive' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

## Average Premium per Coverage Type

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 7. Get average premium for each plan type
SELECT coverage_type, AVG(base_premium) AS avg_base
FROM product_config
GROUP BY coverage_type;
```

The Results grid shows the following data:

coverage_type	avg_base
Comprehensive	5750.000000
TPL	3000.000000
Both	8000.000000

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
154	01:03:05	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.016 sec / 0.000 sec
155	01:03:20	SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid FROM premium_summary ps JOIN qu...	5 row(s) returned	0.000 sec / 0.000 sec
156	01:04:08	SELECT vm.model_desc FROM vehicle_model vm WHERE status = 'inactive' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
157	01:07:46	SELECT coverage_type, AVG(base_premium) AS avg_base FROM product_config GROUP BY coverage_typ...	3 row(s) returned	0.000 sec / 0.000 sec

## Users with Their City and State

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 8. List all users with their city and state
SELECT u.first_name, u.last_name, c.city_name, s.state_name
FROM user_personal_info u
JOIN city c ON u.city = c.city_id
JOIN state s ON u.state = s.state_id;
```

The Result Grid displays the following data:

first_name	last_name	city_name	state_name
Ravi	Kumar	Chennai	Tamil Nadu
Anjali	Verma	Kolkata	West Bengal
Vikram	Singh	Bangalore	Karnataka
Neha	Sharma	Delhi	Delhi
Amit	Patel	Ahmedabad	Gujarat
Deepa	Joshi	Ludhrow	Uttar Pradesh
Rahul	Mishra	Bangalore	Karnataka
Snaha	Menon	Kochi	Kerala
Arun	Debnath	Mumbai	Maharashtra
Meena	Pillai	Kochi	Kerala

The Action Output pane shows the execution progress of the query.

## Users Who Paid Above-Average Premium

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

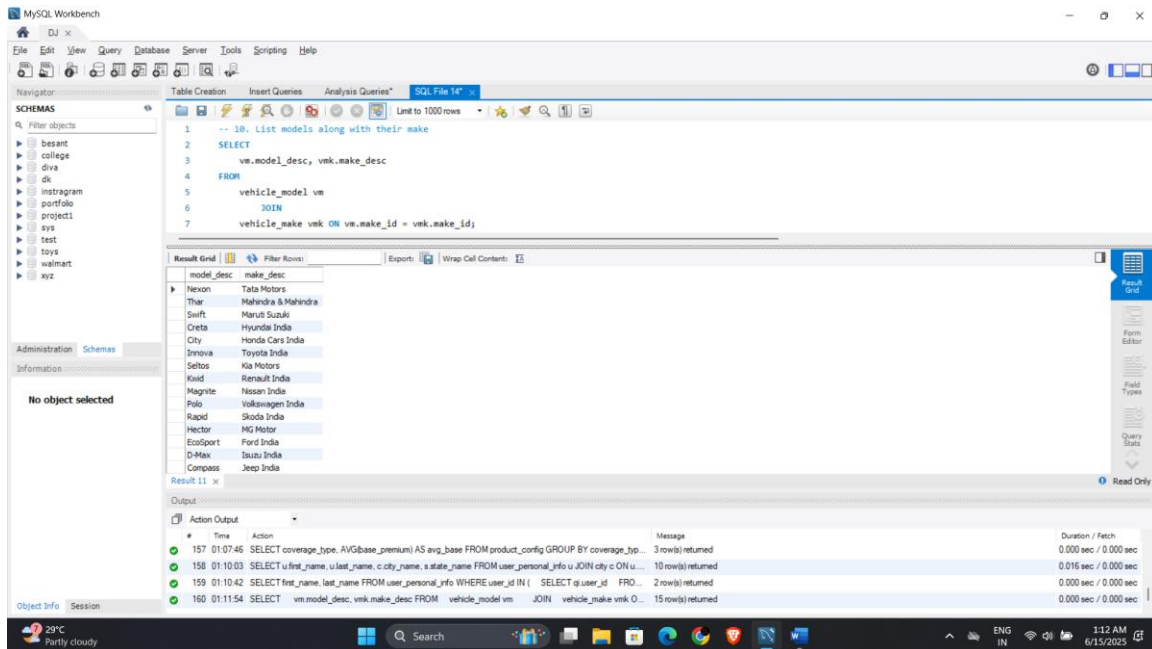
```
-- 9. Users who Paid Above-Average Premium
SELECT first_name, last_name
FROM user_personal_info
WHERE user_id IN (
  SELECT qi.user_id
  FROM quote_info qi
  JOIN premium_summary ps ON qi.quote_id = ps.quote_id
  GROUP BY qi.user_id
  HAVING SUM(ps.total_premium) > (
    SELECT AVG(total_premium) FROM premium_summary
  )
);
```

The Result Grid displays the following data:

first_name	last_name
Vikram	Singh
Neha	Sharma

The Action Output pane shows the execution progress of the query.

## Model and Make Combinations



The screenshot shows the MySQL Workbench interface. The left sidebar displays a list of schemas: besant, college, diva, dk, instagram, portfolio, project1, sys, test, walmart, and xyz. The main editor window contains the following SQL query:

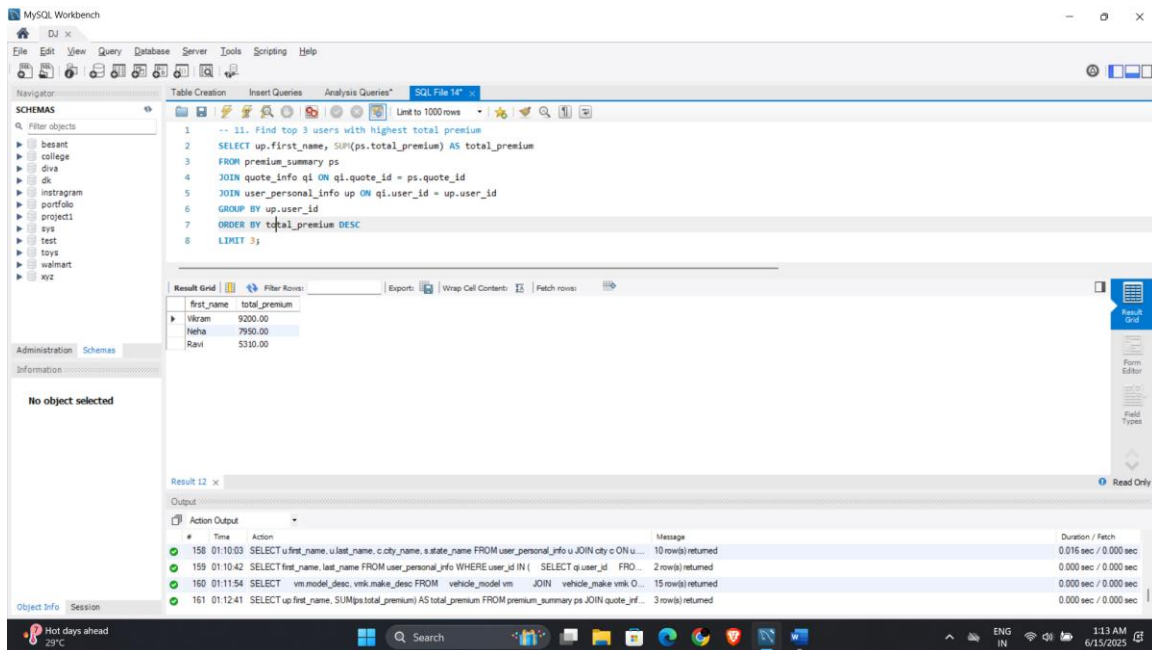
```
-- 10. List models along with their make
SELECT
  vm.model_desc, vmk.make_desc
FROM
  vehicle_model vm
JOIN
  vehicle_make vmk ON vm.make_id = vmk.make_id;
```

The query results are displayed in a table with two columns: model\_desc and make\_desc. The results are as follows:

model_desc	make_desc
Nexon	Tata Motors
Thar	Mahindra & Mahindra
Swift	Maruti Suzuki
Creta	Hyundai India
City	Honda Cars India
Innova	Toyota India
Seltos	Kia Motors
Kwid	Renault India
Magnite	Nissan India
Polo	Volkswagen India
Rapid	Skoda India
Hector	MG Motor
EcoSport	Ford India
D-Max	Tata India
Compass	Jeep India

The bottom status bar shows the system temperature as 29°C and the time as 1:12 AM on 6/15/2023.

## Top 3 Users with Highest Premium



The screenshot shows the MySQL Workbench interface. The left sidebar displays a list of schemas: besant, college, diva, dk, instagram, portfolio, project1, sys, test, walmart, and xyz. The main editor window contains the following SQL query:

```
-- 11. Find top 3 users with highest total premium
SELECT up.first_name, SUM(ps.total_premium) AS total_premium
FROM premium_summary ps
JOIN quote_info qi ON qi.quote_id = ps.quote_id
JOIN user_personal_info up ON qi.user_id = up.user_id
GROUP BY up.user_id
ORDER BY total_premium DESC
LIMIT 3;
```

The query results are displayed in a table with two columns: first\_name and total\_premium. The results are as follows:

first_name	total_premium
Vikram	9200.00
Neha	7950.00
Ravi	5310.00

The bottom status bar shows the system temperature as 29°C and the time as 1:13 AM on 6/15/2023.

## Quotes with Final Premium

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 12 List all quotes along with product config
SELECT q.quote_id, q.final_premium, pc.product_name
FROM quote_info q
JOIN product_config pc ON q.config_id = pc.config_id;
```

The Result Grid shows the following data:

quote_id	final_premium	product_name
1	5200.00	Standard Plan
2	3400.00	Economy Plan
3	7900.00	Premium Plan
4	7300.00	Urban Plus
5	2950.00	Rural Server

The Action Output shows the following messages:

#	Time	Action	Message	Duration / Fetch
159	01:10:42	SELECT first_name, last_name FROM user_personal_info WHERE user_id IN ( SELECT q.user_id FROM quote_info q JOIN product_config pc ON q.config_id = pc.config_id )	2 row(s) returned	0.000 sec / 0.000 sec
160	01:11:54	SELECT vm.model_desc, vm.make_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = vmk.make_id	15 row(s) returned	0.000 sec / 0.000 sec
161	01:12:41	SELECT up.first_name, SUM(p.total_premium) AS total_premium FROM premium_summary ps JOIN quote_info q ON q.config_id = ps.config_id	3 row(s) returned	0.000 sec / 0.000 sec
162	01:13:40	SELECT q.quote_id, q.final_premium, pc.product_name FROM quote_info q JOIN product_config pc ON q.config_id = pc.config_id	5 row(s) returned	0.000 sec / 0.000 sec

## City with the Highest Number of Users

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
-- 13 City with the Highest Number of Users
SELECT c.city_name, COUNT(u.user_id) AS user_count
FROM user_personal_info u
JOIN city c ON u.city = c.city_id
GROUP BY c.city_name
ORDER BY user_count DESC
LIMIT 1;
```

The Result Grid shows the following data:

city_name	user_count
Bangalore	2

The Action Output shows the following messages:

#	Time	Action	Message	Duration / Fetch
160	01:11:54	SELECT vm.model_desc, vm.make_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = vmk.make_id	15 row(s) returned	0.000 sec / 0.000 sec
161	01:12:41	SELECT up.first_name, SUM(p.total_premium) AS total_premium FROM premium_summary ps JOIN quote_info q ON q.config_id = ps.config_id	3 row(s) returned	0.000 sec / 0.000 sec
162	01:13:40	SELECT q.quote_id, q.final_premium, pc.product_name FROM quote_info q JOIN product_config pc ON q.config_id = pc.config_id	5 row(s) returned	0.000 sec / 0.000 sec
163	01:15:32	SELECT c.city_name, COUNT(u.user_id) AS user_count FROM user_personal_info u JOIN city c ON u.city = c.city_id	1 row(s) returned	0.000 sec / 0.000 sec



## Procedure: Get quotes by user id

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```
-- 14.Procedure: Get all quotes by user
DELIMITER //
CREATE PROCEDURE GetQuotesByUser(IN uid INT)
BEGIN
    SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = uid;
END //
DELIMITER ;

CALL GetQuotesByUser(5);
```

The Result Grid shows the output of the procedure call:

quote_id	final_premium	quote_date
5	2950.00	2024-06-09

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
163	01:15:32	SELECT c.city_name, COUNT(u.user_id) AS user_count FROM user_personal_info u JOIN city c ON u.city = c...	1 row(s) returned	0.000 sec / 0.000 sec
164	01:17:00	CREATE PROCEDURE GetQuotesByUser(IN uid INT) BEGIN SELECT quote_id, final_premium, quote_date...	0 row(s) affected	0.015 sec
165	01:17:03	CALL GetQuotesByUser(1)	1 row(s) returned	0.000 sec / 0.000 sec
166	01:17:07	CALL GetQuotesByUser(5)	1 row(s) returned	0.015 sec / 0.000 sec

## Function: Calculate premium with tax

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

```
-- 15. Function: Calculate premium with tax
DELIMITER //
CREATE FUNCTION CalculateTotalPremium(base DECIMAL(10,2), tax DECIMAL(5,2))
RETURNS DECIMAL(10,2)
DETERMINISTIC
BEGIN
    RETURN base + (base * tax / 100);
END //
DELIMITER ;

SELECT CalculateTotalPremium(5000.00, 18.0) as Total_Premium;
```

The Result Grid shows the output of the function call:

Total_Premium
5900.00

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
166	01:17:07	CALL GetQuotesByUser(5)	1 row(s) returned	0.015 sec / 0.000 sec
167	01:18:38	CREATE FUNCTION CalculateTotalPremium(base DECIMAL(10,2), tax DECIMAL(5,2)) RETURNS DECIMAL...	0 row(s) affected	0.015 sec
168	01:18:41	SELECT CalculateTotalPremium(5000.00, 18.0) LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
169	01:19:01	SELECT CalculateTotalPremium(5000.00, 18.0) as Total_Premium LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

## 6. Conclusion

This SQL project demonstrates the creation and analysis of a comprehensive vehicle insurance database. It showcases schema design, data population, and analytical querying to derive valuable insights.

## 7. Appendix

### Table Creation

```
CREATE DATABASE insurance;
```

```
use insurance;
```

```
CREATE TABLE vehicle_make (  
    make_id INT PRIMARY KEY,  
    make_desc VARCHAR(100) NOT NULL,  
    status VARCHAR(10) CHECK (status IN ('active', 'inactive'))  
);
```

```
CREATE TABLE vehicle_model (  
    model_id INT PRIMARY KEY,  
    model_desc VARCHAR(100) NOT NULL,  
    make_id INT,  
    status VARCHAR(10) CHECK (status IN ('active', 'inactive')),  
    FOREIGN KEY (make_id) REFERENCES vehicle_make(make_id)  
);
```

```
CREATE TABLE region (  
    region_id INT PRIMARY KEY,
```

```
    region_name VARCHAR(100)
);
```

```
CREATE TABLE state (
    state_id INT PRIMARY KEY,
    state_name VARCHAR(100),
    region_id INT,
    FOREIGN KEY (region_id) REFERENCES region(region_id)
);
```

```
CREATE TABLE city (
    city_id INT PRIMARY KEY,
    city_name VARCHAR(100),
    state_id INT,
    FOREIGN KEY (state_id) REFERENCES state(state_id)
);
```

```
CREATE TABLE vehicle_color (
    color_id INT PRIMARY KEY,
    color_name VARCHAR(50)
);
```

```
CREATE TABLE vehicle_body (  
    body_id INT PRIMARY KEY,  
    body_type VARCHAR(50)  
);
```

```
CREATE TABLE vehicle_category (  
    category_id INT PRIMARY KEY,  
    category_type VARCHAR(50)  
);
```

```
CREATE TABLE product_config (  
    config_id INT PRIMARY KEY,  
    product_name VARCHAR(100),  
    coverage_type VARCHAR(20) CHECK (coverage_type IN ('Comprehensive', 'TPL',  
'Both')),  
    base_premium DECIMAL(10,2)  
);
```

```
CREATE TABLE user_personal_info (  
    user_id INT PRIMARY KEY,  
    usertype VARCHAR(50),  
    first_name VARCHAR(50),  
    last_name VARCHAR(50),  
    gender VARCHAR(10),  
    dob DATE NOT NULL,
```

```
email VARCHAR(100) NOT NULL,  
phone VARCHAR(20) NOT NULL,  
addr1 VARCHAR(100),  
city INT,  
state INT,  
country INT,  
national_id BIGINT NOT NULL,  
nationality INT NOT NULL,  
status VARCHAR(10) CHECK (status IN ('active', 'inactive'))  
);
```

```
CREATE TABLE login_user (  
    login_id INT PRIMARY KEY,  
    password VARCHAR(255),  
    user_id INT,  
    usertype VARCHAR(50) NOT NULL,  
    status VARCHAR(10) CHECK (status IN ('active', 'inactive')),  
    FOREIGN KEY (user_id) REFERENCES user_personal_info(user_id)  
);
```

```
CREATE TABLE broker_info (  
    broker_id INT PRIMARY KEY,  
    broker_name VARCHAR(100),  
    broker_org_name VARCHAR(100),  
    address VARCHAR(150),
```

```
status VARCHAR(10) CHECK (status IN ('active', 'inactive'))
);
```

```
CREATE TABLE quote_info (
    quote_id INT PRIMARY KEY,
    user_id INT,
    vehicle_id INT,
    config_id INT,
    quote_date DATE,
    final_premium DECIMAL(10,2),
    status VARCHAR(10) CHECK (status IN ('active', 'inactive')),
    FOREIGN KEY (user_id) REFERENCES user_personal_info(user_id),
    FOREIGN KEY (config_id) REFERENCES product_config(config_id)
);
```

```
CREATE TABLE premium_config (
    premium_id INT PRIMARY KEY,
    config_id INT,
    tax_rate DECIMAL(5,2),
    surcharge DECIMAL(5,2),
    FOREIGN KEY (config_id) REFERENCES product_config(config_id)
);
```

```
CREATE TABLE premium_summary (
    summary_id INT PRIMARY KEY,
```



```
quote_id INT,  
base_premium DECIMAL(10,2),  
tax_amount DECIMAL(10,2),  
total_premium DECIMAL(10,2),  
FOREIGN KEY (quote_id) REFERENCES quote_info(quote_id)  
);
```

```
CREATE TABLE lov_data (  
    lov_id INT PRIMARY KEY,  
    lov_type VARCHAR(100),  
    lov_value VARCHAR(100)  
);
```

### **Insert Queries**

```
-- INSERTS FOR vehicle_make  
  
INSERT INTO vehicle_make VALUES (1, 'Tata Motors', 'active');  
INSERT INTO vehicle_make VALUES (2, 'Mahindra & Mahindra', 'active');  
INSERT INTO vehicle_make VALUES (3, 'Maruti Suzuki', 'active');  
INSERT INTO vehicle_make VALUES (4, 'Hyundai India', 'active');  
INSERT INTO vehicle_make VALUES (5, 'Honda Cars India', 'inactive');  
INSERT INTO vehicle_make VALUES (6, 'Toyota India', 'active');  
INSERT INTO vehicle_make VALUES (7, 'Kia Motors', 'active');  
INSERT INTO vehicle_make VALUES (8, 'Renault India', 'inactive');  
INSERT INTO vehicle_make VALUES (9, 'Nissan India', 'active');
```

```
INSERT INTO vehicle_make VALUES (10, 'Volkswagen India', 'inactive');
INSERT INTO vehicle_make VALUES (11, 'Skoda India', 'active');
INSERT INTO vehicle_make VALUES (12, 'MG Motor', 'active');
INSERT INTO vehicle_make VALUES (13, 'Ford India', 'inactive');
INSERT INTO vehicle_make VALUES (14, 'Isuzu India', 'active');
INSERT INTO vehicle_make VALUES (15, 'Jeep India', 'active');
```

```
-- INSERTS FOR vehicle_model
```

```
INSERT INTO vehicle_model VALUES (1, 'Nexon', 1, 'active');
INSERT INTO vehicle_model VALUES (2, 'Thar', 2, 'active');
INSERT INTO vehicle_model VALUES (3, 'Swift', 3, 'active');
INSERT INTO vehicle_model VALUES (4, 'Creta', 4, 'active');
INSERT INTO vehicle_model VALUES (5, 'City', 5, 'inactive');
INSERT INTO vehicle_model VALUES (6, 'Innova', 6, 'active');
INSERT INTO vehicle_model VALUES (7, 'Seltos', 7, 'active');
INSERT INTO vehicle_model VALUES (8, 'Kwid', 8, 'active');
INSERT INTO vehicle_model VALUES (9, 'Magnite', 9, 'inactive');
INSERT INTO vehicle_model VALUES (10, 'Polo', 10, 'active');
INSERT INTO vehicle_model VALUES (11, 'Rapid', 11, 'active');
INSERT INTO vehicle_model VALUES (12, 'Hector', 12, 'inactive');
INSERT INTO vehicle_model VALUES (13, 'EcoSport', 13, 'inactive');
INSERT INTO vehicle_model VALUES (14, 'D-Max', 14, 'active');
INSERT INTO vehicle_model VALUES (15, 'Compass', 15, 'active');
```

```
-- INSERTS FOR region
```

```
INSERT INTO region VALUES (1, 'North');
```

```
INSERT INTO region VALUES (2, 'South');
```

```
INSERT INTO region VALUES (3, 'East');
```

```
INSERT INTO region VALUES (4, 'West');
```

```
-- INSERTS FOR state
```

```
INSERT INTO state VALUES (1, 'Tamil Nadu', 2);
```

```
INSERT INTO state VALUES (2, 'Maharashtra', 4);
```

```
INSERT INTO state VALUES (3, 'Delhi', 1);
```

```
INSERT INTO state VALUES (4, 'Karnataka', 2);
```

```
INSERT INTO state VALUES (5, 'West Bengal', 3);
```

```
INSERT INTO state VALUES (6, 'Kerala', 2);
```

```
INSERT INTO state VALUES (7, 'Uttar Pradesh', 1);
```

```
INSERT INTO state VALUES (8, 'Bihar', 3);
```

```
INSERT INTO state VALUES (9, 'Gujarat', 4);
```

```
INSERT INTO state VALUES (10, 'Rajasthan', 1);
```

```
INSERT INTO state VALUES (11, 'Punjab', 1);
```

```
INSERT INTO state VALUES (12, 'Haryana', 1);
```

```
INSERT INTO state VALUES (13, 'Andhra Pradesh', 2);
```

```
INSERT INTO state VALUES (14, 'Odisha', 3);
```

```
-- INSERTS FOR city
```

```
INSERT INTO city VALUES (1, 'Chennai', 1);
```

```
INSERT INTO city VALUES (2, 'Mumbai', 2);
```

```
INSERT INTO city VALUES (3, 'Delhi', 3);
INSERT INTO city VALUES (4, 'Bangalore', 4);
INSERT INTO city VALUES (5, 'Kolkata', 5);
INSERT INTO city VALUES (6, 'Kochi', 6);
INSERT INTO city VALUES (7, 'Lucknow', 7);
INSERT INTO city VALUES (8, 'Patna', 8);
INSERT INTO city VALUES (9, 'Ahmedabad', 9);
INSERT INTO city VALUES (10, 'Jaipur', 10);
INSERT INTO city VALUES (11, 'Amritsar', 11);
INSERT INTO city VALUES (12, 'Gurgaon', 12);
INSERT INTO city VALUES (13, 'Vijayawada', 13);
INSERT INTO city VALUES (14, 'Bhubaneswar', 14);
```

```
-- INSERTS FOR vehicle_color
```

```
INSERT INTO vehicle_color VALUES (1, 'Red');
INSERT INTO vehicle_color VALUES (2, 'Blue');
INSERT INTO vehicle_color VALUES (3, 'Green');
INSERT INTO vehicle_color VALUES (4, 'Black');
INSERT INTO vehicle_color VALUES (5, 'White');
```

```
-- INSERTS FOR vehicle_body
```

```
INSERT INTO vehicle_body VALUES (1, 'Sedan');
INSERT INTO vehicle_body VALUES (2, 'SUV');
INSERT INTO vehicle_body VALUES (3, 'Hatchback');
```

```
INSERT INTO vehicle_body VALUES (4, 'Convertible');
```

```
INSERT INTO vehicle_body VALUES (5, 'Coupe');
```

```
-- INSERTS FOR vehicle_category
```

```
INSERT INTO vehicle_category VALUES (1, 'Private');
```

```
INSERT INTO vehicle_category VALUES (2, 'Commercial');
```

```
INSERT INTO vehicle_category VALUES (3, 'Luxury');
```

```
INSERT INTO vehicle_category VALUES (4, 'Transport');
```

```
INSERT INTO vehicle_category VALUES (5, 'Tourist');
```

```
-- INSERTS FOR product_config
```

```
INSERT INTO product_config VALUES (1, 'Standard Plan', 'Comprehensive',  
4500.00);
```

```
INSERT INTO product_config VALUES (2, 'Economy Plan', 'TPL', 3200.00);
```

```
INSERT INTO product_config VALUES (3, 'Premium Plan', 'Both', 8000.00);
```

```
INSERT INTO product_config VALUES (4, 'Urban Plus', 'Comprehensive', 7000.00);
```

```
INSERT INTO product_config VALUES (5, 'Rural Saver', 'TPL', 2800.00);
```

```
-- INSERTS FOR lov_data
```

```
INSERT INTO lov_data VALUES (1, 'MaritalStatus', 'Single');
```

```
INSERT INTO lov_data VALUES (2, 'MaritalStatus', 'Married');
```

```
INSERT INTO lov_data VALUES (3, 'Education', 'Graduate');
```

```
INSERT INTO lov_data VALUES (4, 'Education', 'Post-Graduate');
```

```
INSERT INTO lov_data VALUES (5, 'Education', 'Diploma');
```

-- INSERTS FOR user\_personal\_info

```
INSERT INTO user_personal_info VALUES (1, 'Customer', 'Ravi', 'Kumar', 'Male',  
'1995-06-15', 'ravi.kumar@example.com', '9876543210', '12 MG Road, Chennai', 1,  
1, 1, 123456789012, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (2, 'Customer', 'Anjali', 'Verma', 'Female',  
'1990-02-20', 'anjali.verma@example.com', '9876543211', '88 Park Street, Kolkata',  
5, 5, 1, 234567890123, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (3, 'Customer', 'Vikram', 'Singh', 'Male',  
'1988-09-10', 'vikram.singh@example.com', '9876543212', '22 Residency Rd,  
Bangalore', 4, 4, 1, 345678901234, 1, 'inactive');
```

```
INSERT INTO user_personal_info VALUES (4, 'Customer', 'Neha', 'Sharma', 'Female',  
'1993-12-05', 'neha.sharma@example.com', '9876543213', '44 MI Lane, Delhi', 3, 3,  
1, 456789012345, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (5, 'Customer', 'Amit', 'Patel', 'Male',  
'1992-03-11', 'amit.patel@example.com', '9876543214', '20 Ring Road,  
Ahmedabad', 9, 9, 1, 567890123456, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (6, 'Customer', 'Deepa', 'Joshi', 'Female',  
'1987-07-19', 'deepa.joshi@example.com', '9876543215', '32 Sector 18, Noida', 7, 7,  
1, 678901234567, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (7, 'Customer', 'Rahul', 'Mishra', 'Male',  
'1991-01-25', 'rahul.mishra@example.com', '9876543216', '70 Rajaji Nagar,  
Bengaluru', 4, 4, 1, 789012345678, 1, 'active');
```

```
INSERT INTO user_personal_info VALUES (8, 'Customer', 'Sneha', 'Menon', 'Female',  
'1985-04-22', 'sneha.menon@example.com', '9876543217', '11 MG Road, Kochi', 6,  
6, 1, 890123456789, 1, 'inactive');
```

```
INSERT INTO user_personal_info VALUES (9, 'Customer', 'Arjun', 'Deshmukh', 'Male',  
'1994-11-08', 'arjun.deshmukh@example.com', '9876543218', '25 Andheri East,  
Mumbai', 2, 2, 1, 901234567890, 1, 'active');
```



```
INSERT INTO user_personal_info VALUES (10, 'Customer', 'Meena', 'Pillai', 'Female',  
'1996-08-30', 'meena.pillai@example.com', '9876543219', '17 Kowdiar,  
Thiruvananthapuram', 6, 6, 1, 912345678901, 1, 'active');
```

```
-- INSERTS FOR login_user
```

```
INSERT INTO login_user VALUES (1, 'pass1234', 1, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (2, 'pass2345', 2, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (3, 'pass3456', 3, 'Customer', 'inactive');
```

```
INSERT INTO login_user VALUES (4, 'pass4567', 4, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (5, 'pass5678', 5, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (6, 'pass6789', 1, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (7, 'pass7890', 2, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (8, 'pass8901', 3, 'Customer', 'inactive');
```

```
INSERT INTO login_user VALUES (9, 'pass9012', 4, 'Customer', 'active');
```

```
INSERT INTO login_user VALUES (10, 'pass0123', 5, 'Customer', 'active');
```

```
-- INSERTS FOR broker_info
```

```
INSERT INTO broker_info VALUES (1, 'Suresh Babu', 'SecureBrokers Pvt Ltd', '101  
GST Road, Chennai', 'active');
```

```
INSERT INTO broker_info VALUES (2, 'Priya Mehta', 'Shield Insurance', '55 Nariman  
Point, Mumbai', 'active');
```

```
INSERT INTO broker_info VALUES (3, 'Alok Reddy', 'CoverSure India', '78 Brigade  
Road, Bangalore', 'inactive');
```

```
INSERT INTO broker_info VALUES (4, 'Divya Kapoor', 'MaxPolicy Agency', '19 CP  
Circle, Delhi', 'active');
```

```
INSERT INTO broker_info VALUES (5, 'Rajan Desai', 'TrustBrokers LLP', '16 CG Road,  
Ahmedabad', 'active');
```

-- INSERTS FOR quote\_info

INSERT INTO quote\_info VALUES (1, 1, 1, 1, '2024-06-01', 5200.00, 'active');

INSERT INTO quote\_info VALUES (2, 2, 3, 2, '2024-06-03', 3400.00, 'active');

INSERT INTO quote\_info VALUES (3, 3, 5, 3, '2024-06-05', 7900.00, 'inactive');

INSERT INTO quote\_info VALUES (4, 4, 7, 4, '2024-06-07', 7100.00, 'active');

INSERT INTO quote\_info VALUES (5, 5, 9, 5, '2024-06-09', 2950.00, 'active');

-- INSERTS FOR premium\_config

INSERT INTO premium\_config VALUES (1, 1, 18.0, 2.5);

INSERT INTO premium\_config VALUES (2, 2, 12.5, 1.5);

INSERT INTO premium\_config VALUES (3, 3, 15.0, 3.0);

INSERT INTO premium\_config VALUES (4, 4, 10.0, 2.0);

INSERT INTO premium\_config VALUES (5, 5, 8.0, 1.0);

-- INSERTS FOR premium\_summary

INSERT INTO premium\_summary VALUES (1, 1, 4500.00, 810.00, 5310.00);

INSERT INTO premium\_summary VALUES (2, 2, 3200.00, 400.00, 3600.00);

INSERT INTO premium\_summary VALUES (3, 3, 8000.00, 1200.00, 9200.00);

INSERT INTO premium\_summary VALUES (4, 4, 7000.00, 950.00, 7950.00);

INSERT INTO premium\_summary VALUES (5, 5, 2800.00, 560.00, 3360.00);

## Analysis Queries

-- USE CASE QUERIES

-- 1. List all active customers

```
SELECT first_name, last_name, email FROM user_personal_info WHERE status = 'active';
```

-- 2. Fetch vehicle models by a specific make

```
SELECT vm.model_desc FROM vehicle_model vm JOIN vehicle_make vmk ON vm.make_id = vmk.make_id WHERE vmk.make_desc = 'Tata Motors';
```

-- 3. Show all brokers operating in active status

```
SELECT broker_name, broker_org_name FROM broker_info WHERE status = 'active';
```

-- 4. List all quotes for a particular user

```
SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id = 1;
```

-- 5. Get total premium paid by each user

```
SELECT up.user_id, up.first_name, SUM(ps.total_premium) AS total_paid  
FROM premium_summary ps  
JOIN quote_info qi ON qi.quote_id = ps.quote_id  
JOIN user_personal_info up ON qi.user_id = up.user_id  
GROUP BY up.user_id;
```

-- 6. Find vehicles with inactive models

```
SELECT vm.model_desc FROM vehicle_model vm WHERE status = 'inactive';
```

-- 7. Get average premium for each plan type

```
SELECT coverage_type, AVG(base_premium) AS avg_base  
FROM product_config  
GROUP BY coverage_type;
```

-- 8. List all users with their city and state

```
SELECT u.first_name, u.last_name, c.city_name, s.state_name  
FROM user_personal_info u  
JOIN city c ON u.city = c.city_id  
JOIN state s ON u.state = s.state_id;
```

-- 9. Users Who Paid Above-Average Premium

```
SELECT first_name, last_name  
FROM user_personal_info  
WHERE user_id IN (  
    SELECT qi.user_id  
    FROM quote_info qi  
    JOIN premium_summary ps ON qi.quote_id = ps.quote_id  
    GROUP BY qi.user_id  
    HAVING SUM(ps.total_premium) > (  
        SELECT AVG(total_premium) FROM premium_summary
```

```
)  
);
```

-- 10. List models along with their make

```
SELECT vm.model_desc, vmk.make_desc FROM vehicle_model vm JOIN vehicle_make  
vmk ON vm.make_id = vmk.make_id;
```

-- 11. Find top 3 users with highest total premium

```
SELECT up.first_name, SUM(ps.total_premium) AS total_premium  
FROM premium_summary ps  
JOIN quote_info qi ON qi.quote_id = ps.quote_id  
JOIN user_personal_info up ON qi.user_id = up.user_id  
GROUP BY up.user_id  
ORDER BY total_premium DESC  
LIMIT 3;
```

-- 12 List all quotes along with product config

```
SELECT q.quote_id, q.final_premium, pc.product_name  
FROM quote_info q  
JOIN product_config pc ON q.config_id = pc.config_id;
```

-- 13.City with the Highest Number of Users

```
SELECT c.city_name, COUNT(u.user_id) AS user_count  
FROM user_personal_info u
```

```
JOIN city c ON u.city = c.city_id
```

```
GROUP BY c.city_name
```

```
ORDER BY user_count DESC
```

```
LIMIT 1;
```

```
-- 14.Procedure: Get all quotes by user
```

```
DELIMITER //
```

```
CREATE PROCEDURE GetQuotesByUser(IN uid INT)
```

```
BEGIN
```

```
    SELECT quote_id, final_premium, quote_date FROM quote_info WHERE user_id =  
uid;
```

```
END //
```

```
DELIMITER ;
```

```
CALL GetQuotesByUser(5);
```

```
-- 15. Function: Calculate premium with tax
```

```
DELIMITER //
```

```
CREATE FUNCTION CalculateTotalPremium(base DECIMAL(10,2), tax  
DECIMAL(5,2))
```

```
RETURNS DECIMAL(10,2)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN base + (base * tax / 100);
```

END //

DELIMITER ;

SELECT CalculateTotalPremium(5000.00, 18.0);