

MOTOR INSURANCE MANAGEMENT SYSTEM

NAME: RAMALAKSHMI V

PHONE NUMBER: 9943952545

EMAIL ID: ramalakshmi060903@gmail.com

PROJECT TITLE: MOTOR INSURANCE MANAGEMENT SYSTEM

ABSTRACT

The **Motor Insurance Management System** is designed to automate and manage the core processes involved in motor insurance operations, including vehicle information management, customer data handling, premium calculation, quote generation, and policy issuance. The system ensures accurate, consistent, and efficient handling of insurance-related data through a well-structured relational database.

This project emphasizes effective database design using **MySQL**, where master data such as vehicle make, model, body type, category, color, and location details are maintained separately from transactional data like user information, quotes, premium calculations, payments, and policies. **Primary key and foreign key constraints** are implemented to maintain data integrity, minimize redundancy, and support efficient data retrieval through relational joins.

The system supports multiple user roles such as **administrator, broker, and customer**, enabling controlled access to insurance operations. Premium calculation is performed based on vehicle category, selected coverage options, and predefined rate rules. The database design follows **normalization principles**, ensuring scalability, reliability, and ease of maintenance.

Overall, the Motor Insurance Management System provides a structured and efficient solution for managing motor insurance data and demonstrates the practical application of database concepts such as normalization, constraints, and relational mapping, making it suitable for real-world insurance systems and IT-based applications.

LIST OF TABLES:

1. make_list

Constraints

- PRIMARY KEY → make_id
- NOT NULL → make_desc, make_status

2. model_list

Constraints

- PRIMARY KEY → model_id
- FOREIGN KEY → make_id → make_list(make_id)
- DEFAULT → added_on = CURRENT_TIMESTAMP

3. motor_city

Constraints

- PRIMARY KEY → city_id
- UNIQUE → city_name

4. motor_bodytype

Constraints

- PRIMARY KEY → body_id
- NOT NULL → added_by

5. category_type

Constraints

- PRIMARY KEY → cat_id

6. motor_color

Constraints

- PRIMARY KEY → color_id

- UNIQUE → color_name
- FOREIGN KEY → model_id → model_list(model_id)
- NOT NULL → added_by

7. motor_configuration

Constraints

- PRIMARY KEY → config_id
- FOREIGN KEY →
 - make_id → make_list(make_id)
 - model_id → model_list(model_id)
 - cat_id → category_type(cat_id)
 - body_id → motor_bodytype(body_id)
 - color_id → motor_color(color_id)
 - city_name → motor_city(city_name)
- DEFAULT →
 - available_stock = 0
 - created_on = CURRENT_TIMESTAMP
 - updated_on = CURRENT_TIMESTAMP ON UPDATE

8. personal_information

Constraints

- PRIMARY KEY → user_id
- FOREIGN KEY → city_name → motor_city(city_name)
- NOT NULL → dob, email, phone

9. login_user

Constraints

- PRIMARY KEY → login_id
- FOREIGN KEY → user_id → personal_information(user_id)

10. broker_information

Constraints

- PRIMARY KEY → broker_id
- DEFAULT → added_on = CURRENT_TIMESTAMP

11. coverage_master

Constraints

- PRIMARY KEY → coverage_id

12. quote_information

Constraints

- PRIMARY KEY → quote_id
- FOREIGN KEY →
 - user_id → personal_information(user_id)
 - coverage_id → coverage_master(coverage_id)

13. premium_rate_calculation

Constraints

- PRIMARY KEY → rate_id
- FOREIGN KEY → coverage_id → coverage_master(coverage_id)

14. premium

Constraints

- PRIMARY KEY → premium_id
- FOREIGN KEY →
 - quote_id → quote_information(quote_id)
 - rate_id → premium_rate_calculation(rate_id)
- DEFAULT → calc_date = CURRENT_TIMESTAMP

15. broker_commission

Constraints

- PRIMARY KEY → commission_id
- FOREIGN KEY → coverage_id → coverage_master(coverage_id)

16. agent_application

Constraints

- PRIMARY KEY → app_id
- FOREIGN KEY →
 - user_id → personal_information(user_id)
 - quote_id → quote_information(quote_id)
- DEFAULT → submitted_on = CURRENT_TIMESTAMP

17. policy_master

Constraints

- PRIMARY KEY → policy_id
- FOREIGN KEY → app_id → agent_application(app_id)

18. payment_details

Constraints

- PRIMARY KEY → payment_id
- FOREIGN KEY → policy_id → policy_master(policy_id)
- DEFAULT → payment_date = CURRENT_TIMESTAMP

19. debit_credit_note

Constraints

- PRIMARY KEY → note_id
- FOREIGN KEY → policy_id → policy_master(policy_id)
 - ON DELETE CASCADE
 - ON UPDATE CASCADE
- CHECK → note_type IN ('Debit','Credit')
- NOT NULL → policy_id, note_amount
- DEFAULT → issued_date = CURRENT_TIMESTAMP

```
create database policy;
```

```
use policy;
```

1) CREATE TABLE make_list(

```
make_id INT PRIMARY KEY,  
make_desc VARCHAR(50) NOT NULL,  
make_status VARCHAR(20) NOT NULL,  
make_added_on DATE,  
make_added_by VARCHAR(50)  
);
```

```

INSERT INTO make_list (make_id, make_desc, make_status, make_added_on,
make_added_by) VALUES
(1,'Honda','Active','2025-01-01','Admin'),
(2,'Toyota','Active','2025-01-02','Admin'),
(3,'Suzuki','Active','2025-01-03','Admin');

```

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The code entered is:

```

11    make_added_by VARCHAR(50)
12  );
13  • INSERT INTO make_list (make_id, make_desc, make_status, make_added_on, make_added_by) VALUES
14  (1,'Honda','Active','2025-01-01','Admin'),
15  (2,'Toyota','Active','2025-01-02','Admin'),
16  (3,'Suzuki','Active','2025-01-03','Admin')
17
18 • select*from make_list;

```

The "Result Grid" tab shows the following data:

make_id	make_desc	make_status	make_added_on	make_added_by
1	Honda	Active	2025-01-01	Admin
2	Toyota	Active	2025-01-02	Admin
3	Suzuki	Active	2025-01-03	Admin
*	HULL	HULL	HULL	HULL

The "Output" tab shows the execution log:

#	Time	Action	Message	Duration / Fetch
2	11:40:55	create database policy	1 row(s) affected	0.235 sec
3	11:41:00	use policy	0 row(s) affected	0.000 sec
4	11:41:04	CREATE TABLE make_list(make_id INT PRIMARY KEY, make_desc VARCHAR(50) NOT NULL, make_status VARCHAR(20), added_on DATETIME DEFAULT CURRENT_TIMESTAMP, added_by VARCHAR(50), CONSTRAINT fk_model_make FOREIGN KEY(make_id) REFERENCES make_list(make_id)) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci	0 row(s) affected	0.422 sec
5	11:41:49	INSERT INTO make_list (make_id, make_desc, make_status, make_added_on, make_added_by) VALUES (1, 'Honda', 'Active', '2025-01-01', 'Admin')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.188 sec
6	11:42:04	select*from make_list LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

2) CREATE TABLE model_list(

```

model_id INT PRIMARY KEY,
model_desc VARCHAR(200) NOT NULL,
make_id INT,
model_status VARCHAR(20),
added_on DATETIME DEFAULT CURRENT_TIMESTAMP,
added_by VARCHAR(50),
CONSTRAINT fk_model_make FOREIGN KEY(make_id) REFERENCES
make_list(make_id)

```

);

```
INSERT INTO model_list (model_id, model_desc, make_id, model_status,
added_on, added_by) VALUES
(1,'Honda City',1,'Active','2025-01-05','Admin'),
(2,'Honda Amaze',1,'Active','2025-01-06','Admin'),
(3,'Honda Jazz',1,'Active','2025-01-07','Admin'),
(4,'Honda BR-V',1,'Active','2025-01-08','Admin'),
(5,'Honda WR-V',1,'Active','2025-01-09','Admin'),
(6,'Honda Civic',1,'Active','2025-01-10','Admin'),
(7,'Honda CR-V',1,'Active','2025-01-11','Admin'),
(8,'Honda Accord',1,'Active','2025-01-12','Admin'),
(9,'Honda City Hybrid',1,'Active','2025-01-13','Admin'),
(10,'Honda BR-V 2025',1,'Active','2025-01-14','Admin'),
(11,'Toyota Corolla',2,'Active','2025-01-15','Admin'),
(12,'Toyota Camry',2,'Active','2025-01-16','Admin'),
(13,'Toyota Fortuner',2,'Active','2025-01-17','Admin'),
(14,'Toyota Yaris',2,'Active','2025-01-18','Admin'),
(15,'Toyota Innova',2,'Active','2025-01-19','Admin'),
(16,'Toyota RAV4',2,'Active','2025-01-20','Admin'),
(17,'Toyota Hilux',2,'Active','2025-01-21','Admin'),
(18,'Toyota Vios',2,'Active','2025-01-22','Admin'),
(19,'Toyota Prius',2,'Active','2025-01-23','Admin'),
(20,'Toyota Urban Cruiser',2,'Active','2025-01-24','Admin'),
(21,'Suzuki Swift',3,'Active','2025-01-25','Admin'),
(22,'Suzuki Baleno',3,'Active','2025-01-26','Admin'),
```

```

(23,'Suzuki Vitara',3,'Active','2025-01-27','Admin'),
(24,'Suzuki Dzire',3,'Active','2025-01-28','Admin'),
(25,'Suzuki Celerio',3,'Active','2025-01-29','Admin'),
(26,'Suzuki Ertiga',3,'Active','2025-01-30','Admin'),
(27,'Suzuki XL7',3,'Active','2025-01-31','Admin'),
(28,'Suzuki Brezza',3,'Active','2025-02-01','Admin'),
(29,'Suzuki S-Cross',3,'Active','2025-02-02','Admin'),
(30,'Suzuki Grand Vitara',3,'Active','2025-02-03','Admin');

select*from model_list;

```

The screenshot shows the MySQL Workbench interface with a query editor window titled "Query 1". The query window contains the following SQL code:

```

53  (24,'Suzuki Dzire',3,'Active','2025-01-28','Admin'),
54  (25,'Suzuki Celerio',3,'Active','2025-01-29','Admin'),
55  (26,'Suzuki Ertiga',3,'Active','2025-01-30','Admin'),
56  (27,'Suzuki XL7',3,'Active','2025-01-31','Admin'),
57  (28,'Suzuki Brezza',3,'Active','2025-02-01','Admin'),
58  (29,'Suzuki S-Cross',3,'Active','2025-02-02','Admin'),
59  (30,'Suzuki Grand Vitara',3,'Active','2025-02-03','Admin');
60 • select*from model_list;

```

The "Result Grid" tab is selected, displaying the following data:

model_id	model_desc	make_id	model_status	added_on	added_by
1	Honda City	1	Active	2025-01-05 00:00:00	Admin
2	Honda Amaze	1	Active	2025-01-06 00:00:00	Admin
3	Honda Jazz	1	Active	2025-01-07 00:00:00	Admin
4	Honda BR-V	1	Active	2025-01-08 00:00:00	Admin
5	Honda WR-V	1	Active	2025-01-09 00:00:00	Admin
6	Honda Civic	1	Active	2025-01-10 00:00:00	Admin

The "Output" pane at the bottom shows the execution log:

Action	Time	Action	Message	Duration / Fetch
1	11:41:49	INSERT INTO make_list (make_id,make_desc,make_status,make_added_on,make_added_by) VA...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.188 sec
2	11:42:04	select*from make_list LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
3	11:42:56	CREATE TABLE model_list(model_id INT PRIMARY KEY, model_desc VARCHAR(200) NOT N...	0 row(s) affected	1.141 sec
4	11:43:11	INSERT INTO model_list (model_id,model_desc,make_id,model_status,added_on,added_by) VALU...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.219 sec
5	11:43:16	select*from model_list LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec

3) CREATE TABLE motor_city(

```

city_id INT PRIMARY KEY,
city_name VARCHAR(30) UNIQUE,
state_name VARCHAR(50)

);

```

```

INSERT INTO motor_city (city_id, city_name, state_name) VALUES
(1,'Chennai','Tamil Nadu'),
(2,'Coimbatore','Tamil Nadu'),
(3,'Madurai','Tamil Nadu'),
(4,'Trichy','Tamil Nadu'),
(5,'Erode','Tamil Nadu'),
(6,'Bangalore','Karnataka'),
(7,'Mysore','Karnataka'),
(8,'Hyderabad','Telangana'),
(9,'Mumbai','Maharashtra'),
(10,'Pune','Maharashtra');

```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80 (selected), dev, main, project, sys, work.
- Navigator:** Shows the schema structure under Schemas.
- Query Editor:** Contains the SQL code from the text block above.
- Result Grid:** Displays the data inserted into the motor_city table:

city_id	city_name	state_name
1	Chennai	Tamil Nadu
2	Coimbatore	Tamil Nadu
3	Madurai	Tamil Nadu
4	Trichy	Tamil Nadu
5	Erode	Tamil Nadu
6	Bangalore	Karnataka
7	Mysore	Karnataka
8	Hyderabad	Telangana
9	Mumbai	Maharashtra
10	Pune	Maharashtra
- Action Output:** Shows the log of executed statements and their durations:

#	Time	Action	Message	Duration / Fetch
8	11:43:11	INSERT INTO model_list (model_id, model_desc, make_id, model_status, added_on, added_by) VALUES	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.219 sec
9	11:43:16	select*from model_list LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
10	11:43:42	CREATE TABLE motor_city(city_id INT PRIMARY KEY, city_name VARCHAR(30) UNIQUE, s...)	0 row(s) affected	0.500 sec
11	11:43:52	INSERT INTO motor_city (city_id, city_name, state_name) VALUES (1,'Chennai','Tamil Nadu'), (2,'Coimbatore','Tamil Nadu'), (3,'Madurai','Tamil Nadu'), (4,'Trichy','Tamil Nadu'), (5,'Erode','Tamil Nadu'), (6,'Bangalore','Karnataka'), (7,'Mysore','Karnataka'), (8,'Hyderabad','Telangana'), (9,'Mumbai','Maharashtra'), (10,'Pune','Maharashtra');	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.250 sec
12	11:44:31	select*from motor_city LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

4) CREATE TABLE motor_bodytype(

```

body_id INT PRIMARY KEY,
body_desc VARCHAR(100),

```

```
body_status VARCHAR(50),
added_on DATE,
added_by VARCHAR(30) NOT NULL
);
INSERT INTO motor_bodytype (body_id, body_desc, body_status, added_on,
added_by) VALUES
(11,'Sedan','Active','2025-01-01','Admin'),
(12,'Hatchback','Active','2025-01-02','Admin'),
(13,'SUV','Active','2025-01-03','Admin'),
(14,'MPV','Active','2025-01-04','Admin'),
(15,'Crossover','Active','2025-01-05','Admin'),
(16,'Convertible','Active','2025-01-06','Admin'),
(17,'Coupe','Active','2025-01-07','Admin'),
(18,'Pickup','Active','2025-01-08','Admin'),
(19,'Van','Active','2025-01-09','Admin'),
(20,'Hybrid','Active','2025-01-10','Admin');
drop table motor_color;
```

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS dev main project sys work

Query 1

```

93   ('Crossover', 'Active', '2025-01-05', 'Admin'),
94   ('Convertible', 'Active', '2025-01-06', 'Admin'),
95   ('Coupe', 'Active', '2025-01-07', 'Admin'),
96   ('Pickup', 'Active', '2025-01-08', 'Admin'),
97   ('Van', 'Active', '2025-01-09', 'Admin'),
98   ('Hybrid', 'Active', '2025-01-10', 'Admin')
99
100 • select*from motor_bodytype;
  
```

	body_id	body_desc	body_status	added_on	added_by
1	Sedan	Active	2025-01-01	Admin	
2	Hatchback	Active	2025-01-02	Admin	
3	SUV	Active	2025-01-03	Admin	
4	MPV	Active	2025-01-04	Admin	
5	Crossover	Active	2025-01-05	Admin	
6	Convertible	Active	2025-01-06	Admin	

motor_bodytype4

Result Grid Form Editor Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
14	11:45:08	INSERT INTO motor_bodytype (body_id, body_desc, body_status, added_on, added_by) VALUES (1, 'Sedan', 'Active', '2025-01-01', 'Admin')	Error Code: 1146, Table 'policy.motor_bodytype' doesn't exist.	0.000 sec
15	11:45:40	CREATE TABLE motor_bodytype (body_id INT PRIMARY KEY, body_desc VARCHAR(100), body_status VARCHAR(100), added_on DATE, added_by VARCHAR(100))	b. 0 row(s) affected	0.484 sec
16	11:45:59	INSERT INTO motor_bodytype (body_id, body_desc, body_status, added_on, added_by) VALUES (1, 'Sedan', 'Active', '2025-01-01', 'Admin')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.235 sec
17	11:46:18	select*from motor_body LIMIT 0, 1000	Error Code: 1146, Table 'policy.motor_body' doesn't exist.	0.000 sec
18	11:46:31	select*from motor_bodytype LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

Object Info

Type here to search

5) CREATE TABLE category_type(

```

cat_id INT PRIMARY KEY,
cat_desc VARCHAR(30),
cat_description VARCHAR(100),
added_on DATE,
added_by VARCHAR(100)
);

INSERT INTO category_type (cat_id, cat_desc, cat_description, added_on, added_by) VALUES
(1,'Basic','Basic Model','2025-01-01','Admin'),
(2,'Premium','Premium Model','2025-01-02','Admin'),
(3,'Luxury','Luxury Model','2025-01-03','Admin'),
(4,'Sport','Sport Model','2025-01-04','Admin'),
(5,'Economy','Economy Model','2025-01-05','Admin'),
(6,'Electric','Electric Model','2025-01-06','Admin'),

```

```
(7,'Diesel','Diesel Model','2025-01-07','Admin'),
(8,'Petrol','Petrol Model','2025-01-08','Admin'),
(9,'Hybrid','Hybrid Model','2025-01-09','Admin'),
(10,'Offroad','Offroad Model','2025-01-10','Admin');
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (dev, main, project, sys, work) and a list of tables (category_type, category_type5).
- Query Editor:** Contains the following SQL code:


```
114 (15,'Economy','Economy Model','2025-01-05','Admin'),
115 (16,'Electric','Electric Model','2025-01-06','Admin'),
116 (17,'Diesel','Diesel Model','2025-01-07','Admin'),
117 (18,'Petrol','Petrol Model','2025-01-08','Admin'),
118 (19,'Hybrid','Hybrid Model','2025-01-09','Admin'),
119 (20,'Offroad','Offroad Model','2025-01-10','Admin');

120
121 • select*from category_type;
```
- Result Grid:** Displays the data from the category_type table:

cat_id	cat_desc	cat_description	added_on	added_by
11	Basic	Basic Model	2025-01-01	Admin
12	Premium	Premium Model	2025-01-02	Admin
13	Luxury	Luxury Model	2025-01-03	Admin
14	Sport	Sport Model	2025-01-04	Admin
15	Economy	Economy Model	2025-01-05	Admin
16	Electric	Electric Model	2025-01-06	Admin
- Action Output:** Shows the history of actions taken:

#	Time	Action	Message	Duration / Fetch
17	11:46:18	select*from motor_body LIMIT 0, 1000	Error Code: 1146. Table 'policy.motor_body' doesn't exist	0.000 sec
18	11:46:31	select*from motor_bodytype LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
19	11:46:55	CREATE TABLE category_type(cat_id INT PRIMARY KEY, cat_desc VARCHAR(30), cat_de...)	0 row(s) affected	1.015 sec
20	11:47:22	INSERT INTO category_type (cat_id, cat_desc, cat_description, added_on, added_by) VALUES (11, 'Basic Model', 'Basic Model', '2025-01-01', 'Admin')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.219 sec
21	11:47:43	select*from category_type LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

6) CREATE TABLE motor_color(

```
color_id INT PRIMARY KEY,
color_name VARCHAR(20) UNIQUE,
model_id INT,
added_on INT,
added_by VARCHAR(40) NOT NULL,
CONSTRAINT fk_color_model FOREIGN KEY(model_id) REFERENCES
model_list(model_id)
);
rollback;
```

```
INSERT INTO motor_color (color_id, color_name, model_id, added_on, added_by) VALUES  
(1,'Red',1,20250101,'Admin'),  
(2,'Blue',2,20250102,'Admin'),  
(3,'Black',3,20250103,'Admin'),  
(4,'White',4,20250104,'Admin'),  
(5,'Grey',5,20250105,'Admin'),  
(6,'Silver',6,20250106,'Admin'),  
(7,'Green',7,20250107,'Admin'),  
(8,'Yellow',8,20250108,'Admin'),  
(9,'Orange',9,20250109,'Admin'),  
(10,'Brown',10,20250110,'Admin'),  
(11,'cyan',11,20250111,'Admin'),  
(12,'magenta',12,20250112,'Admin'),  
(13,'lime',13,20250113,'Admin'),  
(14,'White-grey',14,20250114,'Admin'),  
(15,'Grey-white',15,20250115,'Admin'),  
(16,'indigo',16,20250116,'Admin'),  
(17,'violet',17,20250117,'Admin'),  
(18,'marron',18,20250118,'Admin'),  
(19,'Olive green',19,20250119,'Admin'),  
(20,'olive Brown',20,20250120,'Admin'),  
(21,'olive Red',21,20250121,'Admin'),  
(22,'olive Blue',22,20250122,'Admin'),  
(23,'olive Black',23,20250123,'Admin'),
```

```

(24,'olive White',24,20250124,'Admin'),
(25,'olive Grey',25,20250125,'Admin'),
(26,'olive Silver',26,20250126,'Admin'),
(27,'teal Green',27,20250127,'Admin'),
(28,'gold Yellow',28,20250128,'Admin'),
(29,'corl orange',29,20250129,'Admin'),
(30,'teal Brown',30,20250130,'Admin');

select*from motor_color;

```

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

157     (25,'olive Grey',25,20250125,'Admin'),
158     (26,'olive Silver',26,20250126,'Admin'),
159     (27,'teal Green',27,20250127,'Admin'),
160     (28,'gold Yellow',28,20250128,'Admin'),
161     (29,'corl orange',29,20250129,'Admin'),
162     (30,'teal Brown',30,20250130,'Admin');

164 • select*from motor_color;

```

The result grid displays the following data:

color_id	color_name	model_id	added_on	added_by
1	Red	1	20250101	Admin
2	Blue	2	20250102	Admin
3	Black	3	20250103	Admin
4	White	4	20250104	Admin
5	Grey	5	20250105	Admin
6	Silver	6	20250106	Admin

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
20	11:47:22	INSERT INTO category_type (cat_id, cat_desc, cat_description, added_on, added_by) VALUES (11, 'Motor Configuration', 'Motor Configuration', '2025-01-11 11:47:22', 'Admin')	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.219 sec
21	11:47:43	select*from category_type LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
22	11:48:04	CREATE TABLE motor_color(color_id INT PRIMARY KEY, color_name VARCHAR(20) UNIQUE, model_id INT, added_on DATETIME, added_by VARCHAR(20))	0 row(s) affected	0.829 sec
23	11:48:19	INSERT INTO motor_color (color_id, color_name, model_id, added_on, added_by) VALUES (1, 'Red', 1, '2025-01-11 11:48:19', 'Admin')	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.156 sec
24	11:48:24	select*from motor_color LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec

7) CREATE TABLE motor_configuration (

```

config_id INT PRIMARY KEY,
make_id INT NOT NULL,
model_id INT NOT NULL,
cat_id INT NOT NULL,
body_id INT,

```

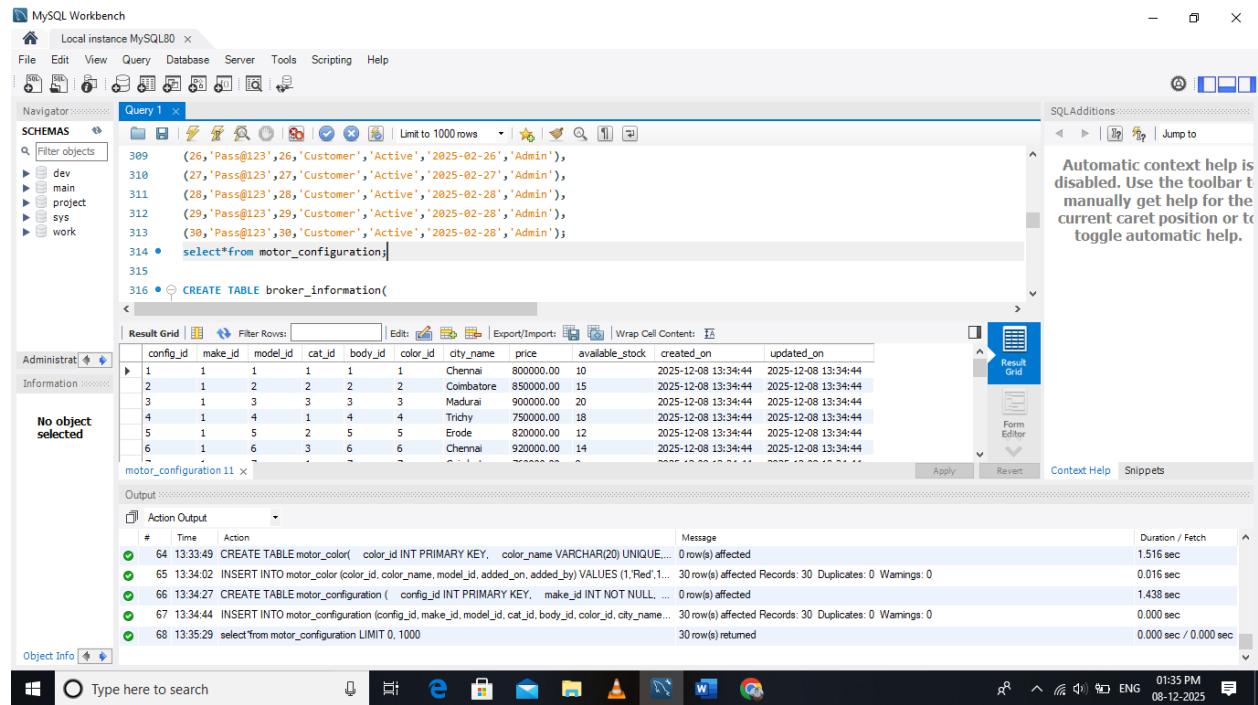
```

color_id INT,
city_name VARCHAR(30),
price DECIMAL(10,2),
available_stock INT DEFAULT 0,
created_on DATETIME DEFAULT CURRENT_TIMESTAMP,
updated_on DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
CONSTRAINT fk_make FOREIGN KEY (make_id) REFERENCES
make_list(make_id),
CONSTRAINT fk_model FOREIGN KEY (model_id) REFERENCES
model_list(model_id),
CONSTRAINT fk_cat FOREIGN KEY (cat_id) REFERENCES
category_type(cat_id),
CONSTRAINT fk_body FOREIGN KEY (body_id) REFERENCES
motor_bodytype(body_id),
CONSTRAINT fk_color FOREIGN KEY (color_id) REFERENCES
motor_color(color_id),
CONSTRAINT fk_city FOREIGN KEY (city_name) REFERENCES
motor_city(city_name)
);
INSERT INTO motor_configuration (config_id, make_id, model_id, cat_id,
body_id, color_id, city_name, price, available_stock) VALUES
(1,1,1,1,1,'Chennai',800000,10),
(2,1,2,2,2,'Coimbatore',850000,15),
(3,1,3,3,3,'Madurai',900000,20),
(4,1,4,1,4,'Trichy',750000,18),

```

(5,1,5,2,5,5,'Erode',820000,12),
(6,1,6,3,6,6,'Chennai',920000,14),
(7,1,7,1,7,7,'Coimbatore',760000,9),
(8,1,8,2,8,8,'Madurai',880000,13),
(9,1,9,3,9,9,'Trichy',910000,17),
(10,1,10,1,10,10,'Erode',770000,11),
(11,2,11,2,1,11,'Chennai',950000,12),
(12,2,12,3,2,12,'Coimbatore',980000,14),
(13,2,13,1,3,13,'Madurai',990000,10),
(14,2,14,2,4,14,'Trichy',1020000,8),
(15,2,15,3,5,15,'Erode',970000,11),
(16,2,16,1,6,16,'Chennai',1010000,9),
(17,2,17,2,7,17,'Coimbatore',960000,13),
(18,2,18,3,8,18,'Madurai',1030000,7),
(19,2,19,1,9,19,'Trichy',1000000,10),
(20,2,20,2,10,20,'Erode',1050000,6),
(21,3,21,3,1,21,'Chennai',1100000,8),
(22,3,22,1,2,22,'Coimbatore',1150000,6),
(23,3,23,2,3,23,'Madurai',1180000,9),
(24,3,24,3,4,24,'Trichy',1130000,7),
(25,3,25,1,5,25,'Erode',1170000,5),
(26,3,26,2,6,26,'Chennai',1200000,8),
(27,3,27,3,7,27,'Coimbatore',1220000,6),
(28,3,28,1,8,28,'Madurai',1190000,7),

(29,3,29,2,9,29,'Trichy',1240000,4),
 (30,3,30,3,10,30,'Erode',1210000,6);



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** dev, main, project, sys, work.
- Query Editor:** Contains the SQL code for creating the `motor_configuration` table and inserting data into it.
- Result Grid:** Displays the data inserted into the `motor_configuration` table.
- Action Output:** Shows the log of actions taken, including the creation of the `motor_color` table and the insertion of data into it.
- System Bar:** Shows the Windows taskbar with various pinned icons and the system clock indicating 01:35 PM on 08-12-2025.

```

309  (26,'Pass@123',26,'Customer','Active','2025-02-26','Admin'),
310  (27,'Pass@123',27,'Customer','Active','2025-02-27','Admin'),
311  (28,'Pass@123',28,'Customer','Active','2025-02-28','Admin'),
312  (29,'Pass@123',29,'Customer','Active','2025-02-28','Admin'),
313  (30,'Pass@123',30,'Customer','Active','2025-02-28','Admin')
314 • select*from motor_configuration;
315
316 • CREATE TABLE broker_information(
  config_id int,
  make_id int,
  model_id int,
  cat_id int,
  body_id int,
  color_id int,
  city_name varchar(20),
  price double,
  available_stock int,
  created_on datetime,
  updated_on datetime
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
  
```

config_id	make_id	model_id	cat_id	body_id	color_id	city_name	price	available_stock	created_on	updated_on
1	1	1	1	1	1	Chennai	800000.00	10	2025-12-08 13:34:44	2025-12-08 13:34:44
2	1	2	2	2	2	Combatore	850000.00	15	2025-12-08 13:34:44	2025-12-08 13:34:44
3	1	3	3	3	3	Madurai	900000.00	20	2025-12-08 13:34:44	2025-12-08 13:34:44
4	1	4	1	4	4	Trichy	750000.00	18	2025-12-08 13:34:44	2025-12-08 13:34:44
5	1	5	2	5	5	Erode	820000.00	12	2025-12-08 13:34:44	2025-12-08 13:34:44
6	1	6	3	6	6	Chennai	920000.00	14	2025-12-08 13:34:44	2025-12-08 13:34:44

8) CREATE TABLE personal_information(

user_id INT PRIMARY KEY,
 user_type VARCHAR(50),
 first_name VARCHAR(20),
 last_name VARCHAR(20),
 gender VARCHAR(10),
 dob DATE NOT NULL,
 email VARCHAR(100) NOT NULL,
 marital_status INT,
 education VARCHAR(30),
 phone BIGINT NOT NULL,
 address_1 VARCHAR(300),

```

permanent_address VARCHAR(300),
city_name VARCHAR(30),
status_update VARCHAR(30),
added_on DATE,
added_by VARCHAR(40),
FOREIGN KEY (city_name) REFERENCES motor_city(city_name)
);

INSERT INTO personal_information (user_id, user_type, first_name, last_name, gender, dob, email, marital_status, education, phone, address_1, permanent_address, city_name, status_update, added_on, added_by) VALUES
(1,'Customer','Ramalakshmi','V','Female','2002-05-03','ramall1@example.com',0,'B.Tech',9000000001,'Address 1','Permanent 1','Chennai','Active','2025-02-01','Admin'),
(2,'Customer','Saranya','R','Female','2002-06-15','saranya2@example.com',0,'B.Tech',9000000002,'Address 2','Permanent 2','Coimbatore','Active','2025-02-02','Admin'),
(3,'Customer','Anitha','K','Female','2002-07-20','anitha3@example.com',0,'B.Tech',9000000003,'Address 3','Permanent 3','Madurai','Active','2025-02-03','Admin'),
(4,'Customer','Priya','L','Female','2002-08-11','priya4@example.com',0,'B.Tech',9000000004,'Address 4','Permanent 4','Trichy','Active','2025-02-04','Admin'),
(5,'Customer','Divya','M','Female','2002-09-05','divya5@example.com',0,'B.Tech',9000000005,'Address 5','Permanent 5','Erode','Active','2025-02-05','Admin'),
(6,'Customer','Meena','N','Female','2002-10-22','meena6@example.com',0,'B.Tech',9000000006,'Address 6','Permanent 6','Chennai','Active','2025-02-06','Admin'),

```

(7,'Customer','Latha','O','Female','2002-11-30','latha7@example.com',0,'B.Tech',9000000007,'Address 7','Permanent 7','Coimbatore','Active','2025-02-07','Admin'),

(8,'Customer','Kavya','P','Female','2002-12-12','kavya8@example.com',0,'B.Tech',9000000008,'Address 8','Permanent 8','Madurai','Active','2025-02-08','Admin'),

(9,'Customer','Nisha','Q','Female','2003-01-20','nisha9@example.com',0,'B.Tech',9000000009,'Address 9','Permanent 9','Trichy','Active','2025-02-09','Admin'),

(10,'Customer','Swathi','R','Female','2003-02-14','swathi10@example.com',0,'B.Tech',9000000010,'Address 10','Permanent 10','Erode','Active','2025-02-10','Admin'),

(11,'Customer','Raja','S','Male','2002-03-01','raja11@example.com',0,'B.Tech',9000000011,'Address 11','Permanent 11','Chennai','Active','2025-02-11','Admin'),

(12,'Customer','Arun','T','Male','2002-04-02','arun12@example.com',0,'B.Tech',9000000012,'Address 12','Permanent 12','Coimbatore','Active','2025-02-12','Admin'),

(13,'Customer','Karthik','U','Male','2002-05-10','karthik13@example.com',0,'B.Tech',9000000013,'Address 13','Permanent 13','Madurai','Active','2025-02-13','Admin'),

(14,'Customer','Vikram','V','Male','2002-06-18','vikram14@example.com',0,'B.Tech',9000000014,'Address 14','Permanent 14','Trichy','Active','2025-02-14','Admin'),

(15,'Customer','Sathya','W','Male','2002-07-25','sathya15@example.com',0,'B.Tech',9000000015,'Address 15','Permanent 15','Erode','Active','2025-02-15','Admin'),

(16,'Customer','Harish','X','Male','2002-08-05','harish16@example.com',0,'B.Tech',9000000016,'Address 16','Permanent 16','Chennai','Active','2025-02-16','Admin'),

(17,'Customer','Ramesh','Y','Male','2002-09-14','ramesh17@example.com',0,'B.Tech',9000000017,'Address 17','Permanent 17','Coimbatore','Active','2025-02-17','Admin'),

(18,'Customer','Vishal','Z','Male','2002-10-22','vishal18@example.com',0,'B.Tech',9000000018,'Address 18','Permanent 18','Madurai','Active','2025-02-18','Admin'),

(19,'Customer','Mani','A','Male','2002-11-30','mani19@example.com',0,'B.Tech',9000000019,'Address 19','Permanent 19','Trichy','Active','2025-02-19','Admin'),

(20,'Customer','Arjun','B','Male','2002-12-12','arjun20@example.com',0,'B.Tech',9000000020,'Address 20','Permanent 20','Erode','Active','2025-02-20','Admin'),

(21,'Customer','Ajith','C','Male','2003-01-01','ajith21@example.com',0,'B.Tech',9000000021,'Address 21','Permanent 21','Chennai','Active','2025-02-21','Admin'),

(22,'Customer','Kumar','D','Male','2003-01-10','kumar22@example.com',0,'B.Tech',9000000022,'Address 22','Permanent 22','Coimbatore','Active','2025-02-22','Admin'),

(23,'Customer','Ranjith','E','Male','2003-01-20','ranjith23@example.com',0,'B.Tech',9000000023,'Address 23','Permanent 23','Madurai','Active','2025-02-23','Admin'),

(24,'Customer','Suresh','F','Male','2003-02-01','suresh24@example.com',0,'B.Tech',9000000024,'Address 24','Permanent 24','Trichy','Active','2025-02-24','Admin'),

(25,'Customer','Vijay','G','Male','2003-02-10','vijay25@example.com',0,'B.Tech',9000000025,'Address 25','Permanent 25','Erode','Active','2025-02-25','Admin'),

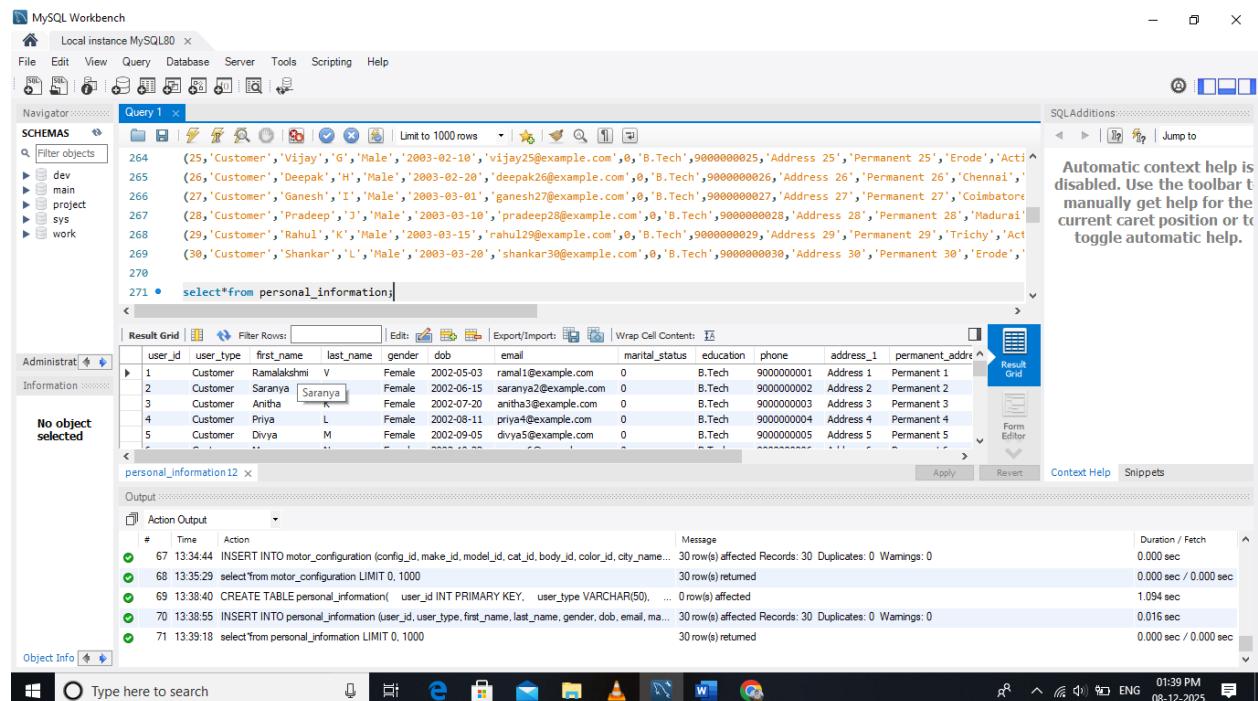
(26,'Customer','Deepak','H','Male','2003-02-20','deepak26@example.com',0,'B.Tech',9000000026,'Address 26','Permanent 26','Chennai','Active','2025-02-26','Admin'),

(27,'Customer','Ganesh','I','Male','2003-03-01','ganesh27@example.com',0,'B.Tech',9000000027,'Address 27','Permanent 27','Coimbatore','Active','2025-02-27','Admin'),

(28,'Customer','Pradeep','J','Male','2003-03-10','pradeep28@example.com',0,'B.Tech',9000000028,'Address 28','Permanent 28','Madurai','Active','2025-02-28','Admin'),

(29,'Customer','Rahul','K','Male','2003-03-15','rahul29@example.com',0,'B.Tech',9000000029,'Address 29','Permanent 29','Trichy','Active','2025-02-28','Admin'),

(30,'Customer','Shankar','L','Male','2003-03-20','shankar30@example.com',0,'B.Tech',9000000030,'Address 30','Permanent 30','Erode','Active','2025-02-28','Admin');



The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for New Connection, Open Connection, Save, Print, Copy, Paste, Find, Replace, and others.
- Navigator:** Shows the schema tree with 'dev', 'main', 'project', 'sys', and 'work' databases.
- SQL Editor:** Contains the query: `select * from personal_information;`
- Result Grid:** Displays the data from the personal_information table.

user_id	user_type	first_name	last_name	gender	dob	email	marital_status	education	phone	address_1	permanent_address
1	Customer	Ramalakshmi	V	Female	2002-05-03	ramal1@example.com	0	B.Tech	9000000001	Address 1	Permanent 1
2	Customer	Saranya	Saranya	Female	2002-06-15	saranya2@example.com	0	B.Tech	9000000002	Address 2	Permanent 2
3	Customer	Anitha	K	Female	2002-07-20	anitha3@example.com	0	B.Tech	9000000003	Address 3	Permanent 3
4	Customer	Priya	L	Female	2002-08-11	priya4@example.com	0	B.Tech	9000000004	Address 4	Permanent 4
5	Customer	Divya	M	Female	2002-09-05	divya5@example.com	0	B.Tech	9000000005	Address 5	Permanent 5

- Output Tab:** Shows the execution history of the query.

#	Time	Action	Message	Duration / Fetch
67	13:34:44	INSERT INTO motor_configuration (config_id, make_id, model_id, cat_id, body_id, color_id, city_name)	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
68	13:35:29	select * from motor_configuration LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
69	13:38:40	CREATE TABLE personal_information(user_id INT PRIMARY KEY, user_type VARCHAR(50), ...)	0 row(s) affected	1.094 sec
70	13:38:55	INSERT INTO personal_information (user_id, user_type, first_name, last_name, gender, dob, email, ma...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.016 sec
71	13:39:18	select * from personal_information LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec

- System Tray:** Shows the Windows taskbar with various icons and the system clock indicating 01:39 PM on 08-12-2025.

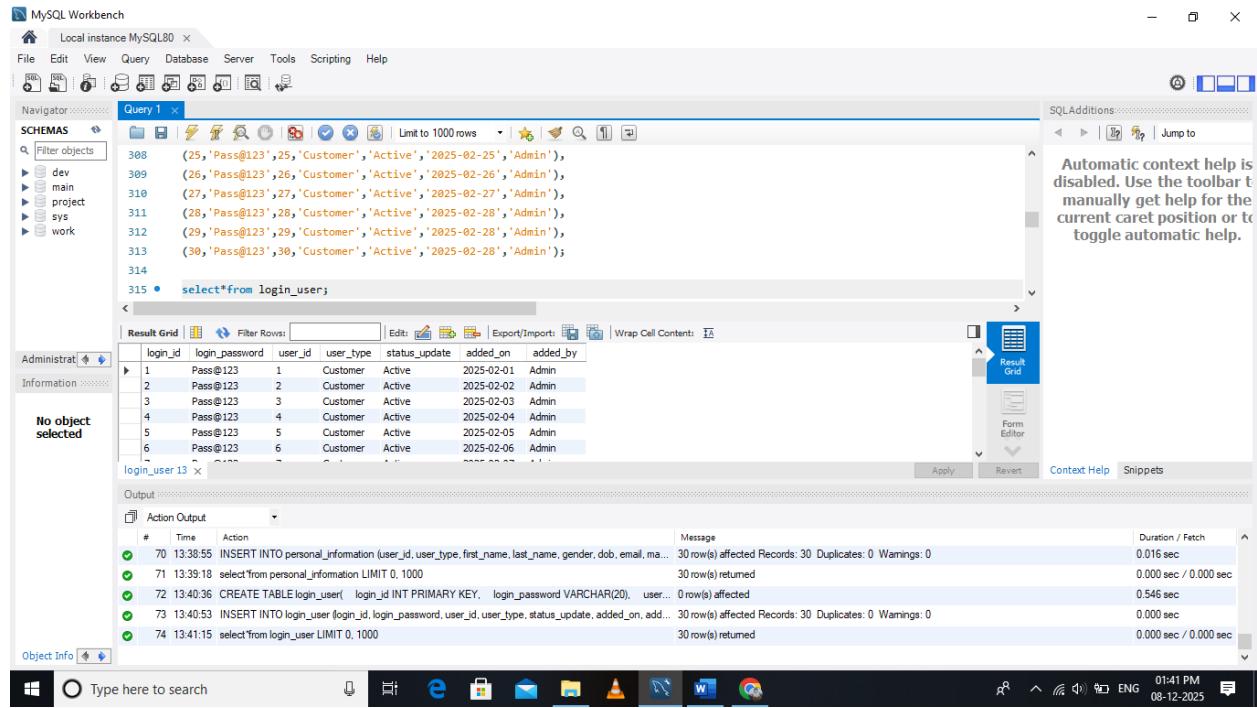
9) CREATE TABLE login_user(

```
login_id INT PRIMARY KEY,  
login_password VARCHAR(20),  
user_id INT,  
user_type VARCHAR(30),  
status_update VARCHAR(20),  
added_on DATE,  
added_by VARCHAR(50),  
FOREIGN KEY(user_id) REFERENCES personal_information(user_id)  
);
```

```
INSERT INTO login_user (login_id, login_password, user_id, user_type,  
status_update, added_on, added_by) VALUES
```

```
(1,'Pass@123',1,'Customer','Active','2025-02-01','Admin'),  
(2,'Pass@123',2,'Customer','Active','2025-02-02','Admin'),  
(3,'Pass@123',3,'Customer','Active','2025-02-03','Admin'),  
(4,'Pass@123',4,'Customer','Active','2025-02-04','Admin'),  
(5,'Pass@123',5,'Customer','Active','2025-02-05','Admin'),  
(6,'Pass@123',6,'Customer','Active','2025-02-06','Admin'),  
(7,'Pass@123',7,'Customer','Active','2025-02-07','Admin'),  
(8,'Pass@123',8,'Customer','Active','2025-02-08','Admin'),  
(9,'Pass@123',9,'Customer','Active','2025-02-09','Admin'),  
(10,'Pass@123',10,'Customer','Active','2025-02-10','Admin'),  
(11,'Pass@123',11,'Customer','Active','2025-02-11','Admin'),
```

(12,'Pass@123',12,'Customer','Active','2025-02-12','Admin'),
(13,'Pass@123',13,'Customer','Active','2025-02-13','Admin'),
(14,'Pass@123',14,'Customer','Active','2025-02-14','Admin'),
(15,'Pass@123',15,'Customer','Active','2025-02-15','Admin'),
(16,'Pass@123',16,'Customer','Active','2025-02-16','Admin'),
(17,'Pass@123',17,'Customer','Active','2025-02-17','Admin'),
(18,'Pass@123',18,'Customer','Active','2025-02-18','Admin'),
(19,'Pass@123',19,'Customer','Active','2025-02-19','Admin'),
(20,'Pass@123',20,'Customer','Active','2025-02-20','Admin'),
(21,'Pass@123',21,'Customer','Active','2025-02-21','Admin'),
(22,'Pass@123',22,'Customer','Active','2025-02-22','Admin'),
(23,'Pass@123',23,'Customer','Active','2025-02-23','Admin'),
(24,'Pass@123',24,'Customer','Active','2025-02-24','Admin'),
(25,'Pass@123',25,'Customer','Active','2025-02-25','Admin'),
(26,'Pass@123',26,'Customer','Active','2025-02-26','Admin'),
(27,'Pass@123',27,'Customer','Active','2025-02-27','Admin'),
(28,'Pass@123',28,'Customer','Active','2025-02-28','Admin'),
(29,'Pass@123',29,'Customer','Active','2025-02-28','Admin'),
(30,'Pass@123',30,'Customer','Active','2025-02-28','Admin');



10) CREATE TABLE broker_information(

```

broker_id INT PRIMARY KEY,
broker_name VARCHAR(30),
broker_organisation_name VARCHAR(100),
contact_info VARCHAR(200),
status_update VARCHAR(50),
added_on DATETIME DEFAULT CURRENT_TIMESTAMP,
added_by VARCHAR(30)
);

```

```

INSERT INTO broker_information (broker_id, broker_name,
broker_organisation_name, contact_info, status_update, added_on, added_by)
VALUES

```

```

(1,'Arun','BrokerOne','1234567890','Active','2025-01-01','Admin'),
(2,'Ravi','BrokerTwo','1234567891','Active','2025-01-02','Admin'),

```

```
(3,'Kumar','BrokerThree','1234567892','Active','2025-01-03','Admin'),
(4,'Suresh','BrokerFour','1234567893','Active','2025-01-04','Admin'),
(5,'Mani','BrokerFive','1234567894','Active','2025-01-05','Admin'),
(6,'Vijay','BrokerSix','1234567895','Active','2025-01-06','Admin'),
(7,'Rahul','BrokerSeven','1234567896','Active','2025-01-07','Admin'),
(8,'Arjun','BrokerEight','1234567897','Active','2025-01-08','Admin'),
(9,'Deepak','BrokerNine','1234567898','Active','2025-01-09','Admin'),
(10,'Ganesh','BrokerTen','1234567899','Active','2025-01-10','Admin');
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL code for inserting data into the `broker_information` table.
- Result Grid:** Displays the 10 inserted rows from the `broker_information` table.
- Action Output:** Shows the log of actions taken, including the creation of the `coverage_master` table and the execution of the insert statements.

broker_id	broker_name	broker_organisation_name	contact_info	status_update	added_on	added_by
1	Arun	BrokerOne	1234567890	Active	2025-01-01 00:00:00	Admin
2	Ravi	BrokerTwo	1234567891	Active	2025-01-02 00:00:00	Admin
3	Kumar	BrokerThree	1234567892	Active	2025-01-03 00:00:00	Admin
4	Suresh	BrokerFour	1234567893	Active	2025-01-04 00:00:00	Admin
5	Mani	BrokerFive	1234567894	Active	2025-01-05 00:00:00	Admin
6	Vijay	BrokerSix	1234567895	Active	2025-01-06 00:00:00	Admin
7	Rahul	BrokerSeven	1234567896	Active	2025-01-07 00:00:00	Admin
8	Arjun	BrokerEight	1234567897	Active	2025-01-08 00:00:00	Admin
9	Deepak	BrokerNine	1234567898	Active	2025-01-09 00:00:00	Admin
10	Ganesh	BrokerTen	1234567899	Active	2025-01-10 00:00:00	Admin

11) CREATE TABLE coverage_master(

```
coverage_id INT PRIMARY KEY,
coverage_type VARCHAR(100),
coverage_desc VARCHAR(200)
);
```

```

INSERT INTO coverage_master (coverage_id, coverage_type, coverage_desc)
VALUES
(1,'Health','Health Insurance'),
(2,'Vehicle','Vehicle Insurance'),
(3,'Property','Property Insurance'),
(4,'Life','Life Insurance'),
(5,'Travel','Travel Insurance'),
(6,'Pet','Pet Insurance'),
(7,'Marine','Marine Insurance'),
(8,'Crop','Crop Insurance'),
(9,'Accident','Accident Insurance'),
(10,'Liability','Liability Insurance');

```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL code for inserting data into the `coverage_master` table.
- Result Grid:** Displays the 10 inserted rows in a tabular format.
- Action Output:** Shows the log of executed statements, including the creation of the table and the insertion of 10 rows.
- System Tray:** Shows the Windows taskbar with various icons and system status.

coverage_id	coverage_type	coverage_desc
1	Health	Health Insurance
2	Vehicle	Vehicle Insurance
3	Property	Property Insurance
4	Life	Life Insurance
5	Travel	Travel Insurance
6	Pet	Pet Insurance
7	Marine	Marine Insurance
8	Crop	Crop Insurance
9	Accident	Accident Insurance
10	Liability	Liability Insurance

```

+----+-----+-----+
| 1  | Health | Health Insurance |
| 2  | Vehicle | Vehicle Insurance |
| 3  | Property | Property Insurance |
| 4  | Life | Life Insurance |
| 5  | Travel | Travel Insurance |
| 6  | Pet | Pet Insurance |
| 7  | Marine | Marine Insurance |
| 8  | Crop | Crop Insurance |
| 9  | Accident | Accident Insurance |
| 10 | Liability | Liability Insurance |
+---+-----+-----+

```

Action Output Log:

- 73 13:40:53 INSERT INTO login_user (login_id, login_password, user_id, user_type, status_update, added_on, addd_by) VALUES ('1', '123456', '1', 'User', '0', '2023-12-08 13:40:53', '1')
- 74 13:41:15 select from login_user LIMIT 0, 1000
- 75 13:42:25 CREATE TABLE coverage_master(coverage_id INT PRIMARY KEY, coverage_type VARCHAR(255), coverage_desc VARCHAR(255))
- 76 13:42:35 INSERT INTO coverage_master (coverage_id, coverage_type, coverage_desc) VALUES (1,'Health','Health Insurance')
- 77 13:42:59 select from coverage_master LIMIT 0, 1000

12) CREATE TABLE quote_information(

```
quote_id INT PRIMARY KEY,  
user_id INT,  
coverage_id INT,  
quote_date DATETIME,  
status_update VARCHAR(20),  
FOREIGN KEY(user_id) REFERENCES personal_information(user_id),  
FOREIGN KEY(coverage_id) REFERENCES coverage_master(coverage_id)  
);
```

```
INSERT INTO quote_information (quote_id, user_id, coverage_id, quote_date,  
status_update) VALUES
```

```
(1,1,2,'2025-03-01','Pending'),  
(2,2,1,'2025-03-02','Pending'),  
(3,3,3,'2025-03-03','Pending'),  
(4,4,2,'2025-03-04','Pending'),  
(5,5,4,'2025-03-05','Pending'),  
(6,6,2,'2025-03-06','Pending'),  
(7,7,5,'2025-03-07','Pending'),  
(8,8,2,'2025-03-08','Pending'),  
(9,9,1,'2025-03-09','Pending'),  
(10,10,3,'2025-03-10','Pending'),  
(11,11,2,'2025-03-11','Pending'),  
(12,12,4,'2025-03-12','Pending'),  
(13,13,2,'2025-03-13','Pending'),  
(14,14,5,'2025-03-14','Pending'),
```

```
(15,15,2,'2025-03-15','Pending'),
(16,16,3,'2025-03-16','Pending'),
(17,17,2,'2025-03-17','Pending'),
(18,18,1,'2025-03-18','Pending'),
(19,19,2,'2025-03-19','Pending'),
(20,20,4,'2025-03-20','Pending'),
(21,21,2,'2025-03-21','Pending'),
(22,22,5,'2025-03-22','Pending'),
(23,23,2,'2025-03-23','Pending'),
(24,24,3,'2025-03-24','Pending'),
(25,25,2,'2025-03-25','Pending'),
(26,26,1,'2025-03-26','Pending'),
(27,27,2,'2025-03-27','Pending'),
(28,28,4,'2025-03-28','Pending'),
(29,29,2,'2025-03-29','Pending'),
(30,30,3,'2025-03-30','Pending');
```

The screenshot shows the MySQL Workbench interface. The top navigation bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The main area has tabs for Navigator, Schemas, and SQLAdditions. The SQLAdditions tab displays a note about context help. The Query 1 tab contains the following SQL code:

```
396 (28,28,4,'2025-03-28','Pending'),
397 (29,29,2,'2025-03-29','Pending'),
398 (30,30,3,'2025-03-30','Pending');
399
400 • select*from quote_information;
401
402 • CREATE TABLE premium_rate_calculation (
    rate_id IN PRIMARY KEY,
403
<--
```

The Results Grid shows the following data:

quote_id	user_id	coverage_id	quote_date	status_update
1	1	2	2025-03-01 00:00:00	Pending
2	2	1	2025-03-02 00:00:00	Pending
3	3	3	2025-03-03 00:00:00	Pending
4	4	2	2025-03-04 00:00:00	Pending
5	5	4	2025-03-05 00:00:00	Pending
6	6	2	2025-03-06 00:00:00	Pending

The Action Output pane at the bottom lists the following actions:

- 79 13:43:42 INSERT INTO broker_information (broker_id, broker_name, broker_organisation_name, contact_info, s... 10 rows) affected Records: 10 Duplicates: 0 Warnings: 0 Duration / Fetch: 0.000 sec / 0.000 sec
- 80 13:44:02 select from broker_information LIMIT 0, 1000 10 rows) returned 0.000 sec / 0.000 sec
- 81 13:45:00 CREATE TABLE quote_information (quote_id INT PRIMARY KEY, user_id INT, coverage_id INT, quote_d... 0 rows) affected 0.796 sec
- 82 13:45:11 INSERT INTO quote_information (quote_id, user_id, coverage_id, quote_date, status_update) VALUE... 30 rows) affected Records: 30 Duplicates: 0 Warnings: 0 0.000 sec
- 83 13:45:38 select from quote_information LIMIT 0, 1000 30 rows) returned 0.000 sec / 0.000 sec

```
13) CREATE TABLE premium_rate_calculation (
    rate_id INT PRIMARY KEY,
    coverage_id INT,
    premium_rate DECIMAL(10,2),
    effective_from DATE,
    FOREIGN KEY (coverage_id) REFERENCES coverage_master(coverage_id)
);

INSERT INTO premium_rate_calculation (rate_id, coverage_id, premium_rate,
effective_from) VALUES
(1,1,5000.00,'2025-01-01'),
(2,2,7500.00,'2025-01-01'),
(3,3,6000.00,'2025-01-01'),
(4,4,8000.00,'2025-01-01'),
(5,5,5500.00,'2025-01-01'),
(6,6,4500.00,'2025-01-01'),
(7,7,7000.00,'2025-01-01'),
(8,8,6500.00,'2025-01-01'),
(9,9,5000.00,'2025-01-01'),
(10,10,6000.00,'2025-01-01');
```

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator Query 1 x
SCHEMAS Filter objects
dev main project sys work
414 (5,5,5500.00, '2025-01-01'),
415 (6,6,4500.00, '2025-01-01'),
416 (7,7,7000.00, '2025-01-01'),
417 (8,8,6500.00, '2025-01-01'),
418 (9,9,5000.00, '2025-01-01'),
419 (10,10,6000.00, '2025-01-01');
420
421 • select*from premium_rate_calculation;

```

rate_id	coverage_id	premium_rate	effective_from
1	1	5000.00	2025-01-01
2	2	7500.00	2025-01-01
3	3	6000.00	2025-01-01
4	4	8000.00	2025-01-01
5	5	5500.00	2025-01-01
6	6	4500.00	2025-01-01
7	7	7000.00	2025-01-01
8	8	6500.00	2025-01-01
9	9	5000.00	2025-01-01
10	10	6000.00	2025-01-01

No object selected

Object Info

Action Output

#	Time	Action	Message	Duration / Fetch
82	13:45:11	INSERT INTO quote_information (quote_id, user_id, coverage_id, quote_date, status_update) VALUE...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
83	13:45:38	select*from quote_information LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
84	13:46:21	CREATE TABLE premium_rate_calculation (rate_id INT PRIMARY KEY, coverage_id INT, pre...	0 row(s) affected	0.656 sec
85	13:46:31	INSERT INTO premium_rate_calculation (rate_id, coverage_id, premium_rate, effective_from) VALUE...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
86	13:47:04	select*from premium_rate_calculation LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

14) CREATE TABLE premium (

```

premium_id INT PRIMARY KEY,
quote_id INT,
rate_id INT,
premium_amount DECIMAL(10,2),
calc_date DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (quote_id) REFERENCES quote_information(quote_id),
FOREIGN KEY (rate_id) REFERENCES premium_rate_calculation(rate_id)
);

```

INSERT INTO premium (premium_id, quote_id, rate_id, premium_amount, calc_date) VALUES

```

(1,1,2,7500.00,'2025-03-02'),
(2,2,1,5000.00,'2025-03-03'),

```

(3,3,3,6000.00,'2025-03-04'),
(4,4,2,7500.00,'2025-03-05'),
(5,5,4,8000.00,'2025-03-06'),
(6,6,2,7500.00,'2025-03-07'),
(7,7,5,5500.00,'2025-03-08'),
(8,8,2,7500.00,'2025-03-09'),
(9,9,1,5000.00,'2025-03-10'),
(10,10,3,6000.00,'2025-03-11'),
(11,11,2,7500.00,'2025-03-12'),
(12,12,4,8000.00,'2025-03-13'),
(13,13,2,7500.00,'2025-03-14'),
(14,14,5,5500.00,'2025-03-15'),
(15,15,2,7500.00,'2025-03-16'),
(16,16,3,6000.00,'2025-03-17'),
(17,17,2,7500.00,'2025-03-18'),
(18,18,1,5000.00,'2025-03-19'),
(19,19,2,7500.00,'2025-03-20'),
(20,20,4,8000.00,'2025-03-21'),
(21,21,2,7500.00,'2025-03-22'),
(22,22,5,5500.00,'2025-03-23'),
(23,23,2,7500.00,'2025-03-24'),
(24,24,3,6000.00,'2025-03-25'),
(25,25,2,7500.00,'2025-03-26'),
(26,26,1,5000.00,'2025-03-27'),

```
(27,27,2,7500.00,'2025-03-28'),  

(28,28,4,8000.00,'2025-03-29'),  

(29,29,2,7500.00,'2025-03-30'),  

(30,30,3,6000.00,'2025-03-31');
```

The screenshot shows the MySQL Workbench interface. In the Query Editor pane, several SQL statements are listed, including INSERT operations and a SELECT statement:

```
459 (26,26,1,5000.00,'2025-03-27'),  

460 (27,27,2,7500.00,'2025-03-28'),  

461 (28,28,4,8000.00,'2025-03-29'),  

462 (29,29,2,7500.00,'2025-03-30'),  

463 (30,30,3,6000.00,'2025-03-31');  

464  

465 • select*from premium;  

466 |
```

The Result Grid displays the data inserted into the 'premium' table:

premium_id	quote_id	rate_id	premium_amount	calc_date
1	2	7500.00		2025-03-02 00:00:00
2	2	1	5000.00	2025-03-03 00:00:00
3	3	3	6000.00	2025-03-04 00:00:00
4	4	2	7500.00	2025-03-05 00:00:00
5	5	4	8000.00	2025-03-06 00:00:00
6	6	2	7500.00	2025-03-07 00:00:00

The Output pane shows the execution log:

- # 85 13:46:31 INSERT INTO premium_rate_calculation (rate_id, coverage_id, premium_rate, effective_from) VALUES (1, 1, 5000.00, '2025-03-02 00:00:00')
- # 86 13:47:04 select from premium_rate_calculation LIMIT 0, 1000
- # 87 13:47:54 CREATE TABLE premium (premium_id INT PRIMARY KEY, quote_id INT, rate_id INT, pre...
- # 88 13:48:07 INSERT INTO premium (premium_id, quote_id, rate_id, premium_amount, calc_date) VALUES (1,1,2,7500.00, '2025-03-02 00:00:00')
- # 89 13:48:29 select from premium LIMIT 0, 1000

15) CREATE TABLE broker_commission (

```
commission_id INT PRIMARY KEY,  

coverage_id INT,  

commission_percent DECIMAL(5,2),  

FOREIGN KEY (coverage_id) REFERENCES coverage_master(coverage_id)  

);
```

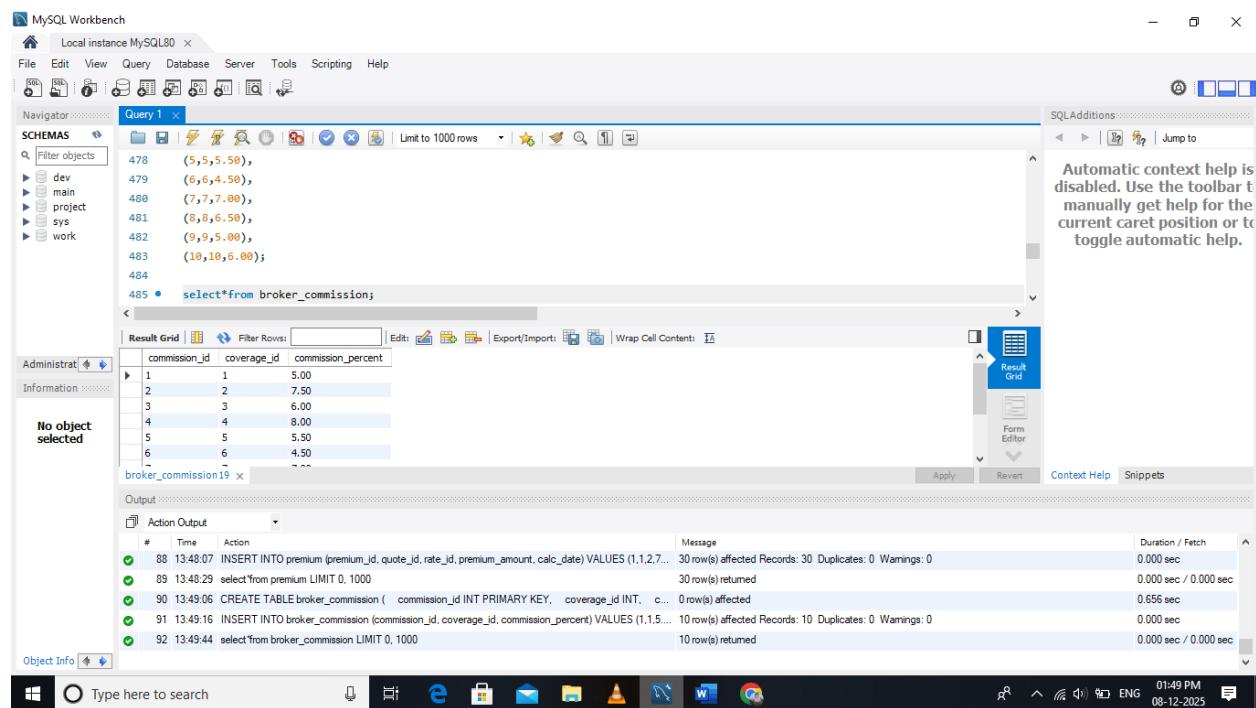
```
INSERT INTO broker_commission (commission_id, coverage_id,  

commission_percent) VALUES
```

```
(1,1,5.00),  

(2,2,7.50),
```

(3,3,6.00),
(4,4,8.00),
(5,5,5.50),
(6,6,4.50),
(7,7,7.00),
(8,8,6.50),
(9,9,5.00),
(10,10,6.00);



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** dev, main, project, sys, work
- Query Editor:** Contains the following SQL code:

```

478      (5,5,5.50),
479      (6,6,4.50),
480      (7,7,7.00),
481      (8,8,6.50),
482      (9,9,5.00),
483      (10,10,6.00);

484
485 • select*from broker_commission;

```
- Result Grid:** Displays the data from the broker_commission table:

commission_id	coverage_id	commission_percent
1	1	5.00
2	2	7.50
3	3	6.00
4	4	8.00
5	5	5.50
6	6	4.50
- Action Output:** Shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
88	13:48:07	INSERT INTO premium (premium_id, quote_id, rate_id, premium_amount, calc_date) VALUES (1,1,2,7,...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
89	13:48:29	select*from premium LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
90	13:49:06	CREATE TABLE broker_commission (commission_id INT PRIMARY KEY, coverage_id INT, c...	0 rows affected	0.656 sec
91	13:49:16	INSERT INTO broker_commission (commission_id, coverage_id, commission_percent) VALUES (1,1,5...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
92	13:49:44	select*from broker_commission LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

16) CREATE TABLE agent_application (

app_id INT PRIMARY KEY,
user_id INT,
quote_id INT,
submitted_on DATETIME DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES personal_information(user_id),

```
FOREIGN KEY (quote_id) REFERENCES quote_information(quote_id)
);

INSERT INTO agent_application (app_id, user_id, quote_id, submitted_on)
VALUES

(1,1,1,'2025-03-03'),
(2,2,2,'2025-03-04'),
(3,3,3,'2025-03-05'),
(4,4,4,'2025-03-06'),
(5,5,5,'2025-03-07'),
(6,6,6,'2025-03-08'),
(7,7,7,'2025-03-09'),
(8,8,8,'2025-03-10'),
(9,9,9,'2025-03-11'),
(10,10,10,'2025-03-12'),
(11,11,11,'2025-03-13'),
(12,12,12,'2025-03-14'),
(13,13,13,'2025-03-15'),
(14,14,14,'2025-03-16'),
(15,15,15,'2025-03-17'),
(16,16,16,'2025-03-18'),
(17,17,17,'2025-03-19'),
(18,18,18,'2025-03-20'),
(19,19,19,'2025-03-21'),
(20,20,20,'2025-03-22'),
```

```
(21,21,21,'2025-03-23'),  

(22,22,22,'2025-03-24'),  

(23,23,23,'2025-03-25'),  

(24,24,24,'2025-03-26'),  

(25,25,25,'2025-03-27'),  

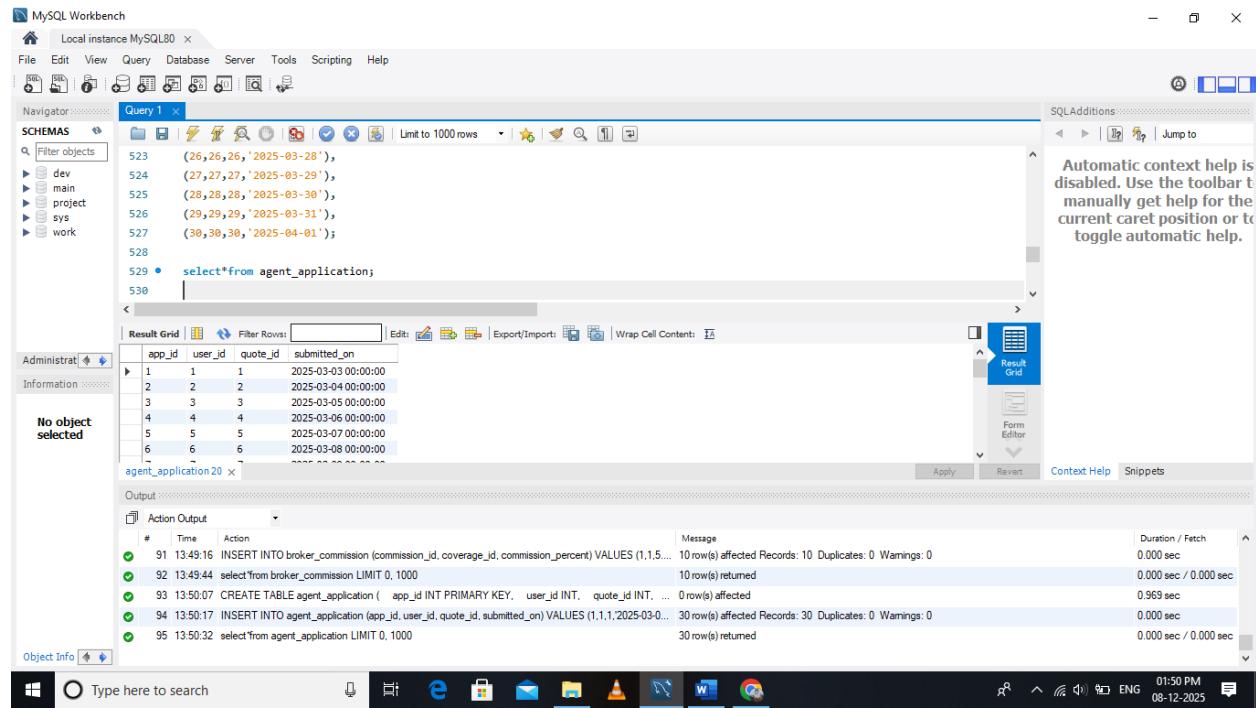
(26,26,26,'2025-03-28'),  

(27,27,27,'2025-03-29'),  

(28,28,28,'2025-03-30'),  

(29,29,29,'2025-03-31'),  

(30,30,30,'2025-04-01');
```



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the following SQL code:

```

523     ('26,26,26,'2025-03-28'),
524     ('27,27,27,'2025-03-29'),
525     ('28,28,28,'2025-03-30'),
526     ('29,29,29,'2025-03-31'),
527     ('30,30,30,'2025-04-01');

528
529 • select*from agent_application;
530

```
- Result Grid:** Displays the results of the `agent_application` query:

app_id	user_id	quote_id	submitted_on
1	1	1	2025-03-03 00:00:00
2	2	2	2025-03-04 00:00:00
3	3	3	2025-03-05 00:00:00
4	4	4	2025-03-06 00:00:00
5	5	5	2025-03-07 00:00:00
6	6	6	2025-03-08 00:00:00
- Action Output:** Shows the log of actions performed:

#	Time	Action	Message	Duration / Fetch
91	13:49:16	INSERT INTO broker_commission (commission_id, coverage_id, commission_percent) VALUES (1,1,5)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
92	13:49:44	select*from broker_commission LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
93	13:50:07	CREATE TABLE agent_application (app_id INT PRIMARY KEY, user_id INT, quote_id INT, submitted_on DATETIME)	0 row(s) affected	0.969 sec
94	13:50:17	INSERT INTO agent_application (app_id, user_id, quote_id, submitted_on) VALUES (1,1,1,2025-03-03 00:00:00)	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
95	13:50:32	select*from agent_application LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec

17) CREATE TABLE policy_master (

```
policy_id INT PRIMARY KEY,  

app_id INT,
```

```
policy_start DATE,  
policy_end DATE,  
debit_credit_note VARCHAR(50),  
FOREIGN KEY (app_id) REFERENCES agent_application(app_id)  
);
```

```
INSERT INTO policy_master (policy_id, app_id, policy_start, policy_end,  
debit_credit_note) VALUES  
(1,1,'2025-04-01','2026-03-31','Debit'),  
(2,2,'2025-04-02','2026-04-01','Credit'),  
(3,3,'2025-04-03','2026-04-02','Debit'),  
(4,4,'2025-04-04','2026-04-03','Credit'),  
(5,5,'2025-04-05','2026-04-04','Debit'),  
(6,6,'2025-04-06','2026-04-05','Credit'),  
(7,7,'2025-04-07','2026-04-06','Debit'),  
(8,8,'2025-04-08','2026-04-07','Credit'),  
(9,9,'2025-04-09','2026-04-08','Debit'),  
(10,10,'2025-04-10','2026-04-09','Credit'),  
(11,11,'2025-04-11','2026-04-10','Debit'),  
(12,12,'2025-04-12','2026-04-11','Credit'),  
(13,13,'2025-04-13','2026-04-12','Debit'),  
(14,14,'2025-04-14','2026-04-13','Credit'),  
(15,15,'2025-04-15','2026-04-14','Debit'),  
(16,16,'2025-04-16','2026-04-15','Credit'),  
(17,17,'2025-04-17','2026-04-16','Debit'),  
(18,18,'2025-04-18','2026-04-17','Credit'),
```

```
(19,19,'2025-04-19','2026-04-18','Debit'),
(20,20,'2025-04-20','2026-04-19','Credit'),
(21,21,'2025-04-21','2026-04-20','Debit'),
(22,22,'2025-04-22','2026-04-21','Credit'),
(23,23,'2025-04-23','2026-04-22','Debit'),
(24,24,'2025-04-24','2026-04-23','Credit'),
(25,25,'2025-04-25','2026-04-24','Debit'),
(26,26,'2025-04-26','2026-04-25','Credit'),
(27,27,'2025-04-27','2026-04-26','Debit'),
(28,28,'2025-04-28','2026-04-27','Credit'),
(29,29,'2025-04-29','2026-04-28','Debit'),
(30,30,'2025-04-30','2026-04-29','Credit');
```

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

dev main project sys work

Query 1

```
566   (25,25,'2025-04-25','2026-04-24','Debit'),
567   (26,26,'2025-04-26','2026-04-25','Credit'),
568   (27,27,'2025-04-27','2026-04-26','Debit'),
569   (28,28,'2025-04-28','2026-04-27','Credit'),
570   (29,29,'2025-04-29','2026-04-28','Debit'),
571   (30,30,'2025-04-30','2026-04-29','Credit');

573 • select*from policy_master;
```

Result Grid

policy_id	app_id	policy_start	policy_end	debit_credit_note
1	1	2025-04-01	2026-03-31	Debit
2	2	2025-04-02	2026-04-01	Credit
3	3	2025-04-03	2026-04-02	Debit
4	4	2025-04-04	2026-04-03	Credit
5	5	2025-04-05	2026-04-04	Debit
6	6	2025-04-06	2026-04-05	Credit
7	7	2025-04-07	2026-04-06	Debit
8	8	2025-04-08	2026-04-07	Credit
9	9	2025-04-09	2026-04-08	Debit
10	10	2025-04-10	2026-04-09	Credit
11	11	2025-04-11	2026-04-10	Debit
12	12	2025-04-12	2026-04-11	Credit
13	13	2025-04-13	2026-04-12	Debit
14	14	2025-04-14	2026-04-13	Credit
15	15	2025-04-15	2026-04-14	Debit
16	16	2025-04-16	2026-04-15	Credit
17	17	2025-04-17	2026-04-16	Debit
18	18	2025-04-18	2026-04-17	Credit
19	19	2025-04-19	2026-04-18	Debit
20	20	2025-04-20	2026-04-19	Credit
21	21	2025-04-21	2026-04-20	Debit
22	22	2025-04-22	2026-04-21	Credit
23	23	2025-04-23	2026-04-22	Debit
24	24	2025-04-24	2026-04-23	Credit
25	25	2025-04-25	2026-04-24	Debit
26	26	2025-04-26	2026-04-25	Credit
27	27	2025-04-27	2026-04-26	Debit
28	28	2025-04-28	2026-04-27	Credit
29	29	2025-04-29	2026-04-28	Debit
30	30	2025-04-30	2026-04-29	Credit

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

#	Time	Action	Message	Duration / Fetch
94	13:50:17	INSERT INTO agent_application (app_id, user_id, quote_id, submitted_on) VALUES (1,1,1,2025-03-0...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
95	13:50:32	select*from agent_application LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
96	13:50:59	CREATE TABLE policy_master (policy_id INT PRIMARY KEY, app_id INT, policy_start DATE, policy_end DATE, debit_credit_note VARCHAR(50))	0 row(s) affected	0.687 sec
97	13:51:12	INSERT INTO policy_master (policy_id, app_id, policy_start, policy_end, debit_credit_note) VALUES (...)	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
98	13:51:27	select*from policy_master LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec

Object Info

Type here to search

Windows Taskbar: File Explorer, Edge, Mail, File, VLC, Word, Google Chrome

System tray: Battery, Network, ENG 01:51 PM, 08-12-2025

```
18) CREATE TABLE payment_details (
    payment_id INT PRIMARY KEY,
    policy_id INT,
    payment_mode VARCHAR(20),
    amount DECIMAL(10,2),
    payment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (policy_id) REFERENCES policy_master(policy_id)
);
```

```
INSERT INTO payment_details (payment_id, policy_id, payment_mode, amount,
payment_date) VALUES
(1,1,'Online',7500.00,'2025-04-01'),
(2,2,'Cash',5000.00,'2025-04-02'),
(3,3,'Online',6000.00,'2025-04-03'),
(4,4,'Cash',7500.00,'2025-04-04'),
(5,5,'Online',8000.00,'2025-04-05'),
(6,6,'Cash',7500.00,'2025-04-06'),
(7,7,'Online',5500.00,'2025-04-07'),
(8,8,'Cash',7500.00,'2025-04-08'),
(9,9,'Online',5000.00,'2025-04-09'),
(10,10,'Cash',6000.00,'2025-04-10'),
(11,11,'Online',7500.00,'2025-04-11'),
(12,12,'Cash',8000.00,'2025-04-12'),
(13,13,'Online',7500.00,'2025-04-13'),
(14,14,'Cash',5500.00,'2025-04-14'),
(15,15,'Online',7500.00,'2025-04-15'),
```

```
(16,16,'Cash',6000.00,'2025-04-16'),
(17,17,'Online',7500.00,'2025-04-17'),
(18,18,'Cash',5000.00,'2025-04-18'),
(19,19,'Online',7500.00,'2025-04-19'),
(20,20,'Cash',8000.00,'2025-04-20'),
(21,21,'Online',7500.00,'2025-04-21'),
(22,22,'Cash',5500.00,'2025-04-22'),
(23,23,'Online',7500.00,'2025-04-23'),
(24,24,'Cash',6000.00,'2025-04-24'),
(25,25,'Online',7500.00,'2025-04-25'),
(26,26,'Cash',5000.00,'2025-04-26'),
(27,27,'Online',7500.00,'2025-04-27'),
(28,28,'Cash',8000.00,'2025-04-28'),
(29,29,'Online',7500.00,'2025-04-29'),
(30,30,'Cash',6000.00,'2025-04-30');
```

The screenshot shows the MySQL Workbench interface with a query editor containing the provided SQL code. The results grid displays the following data:

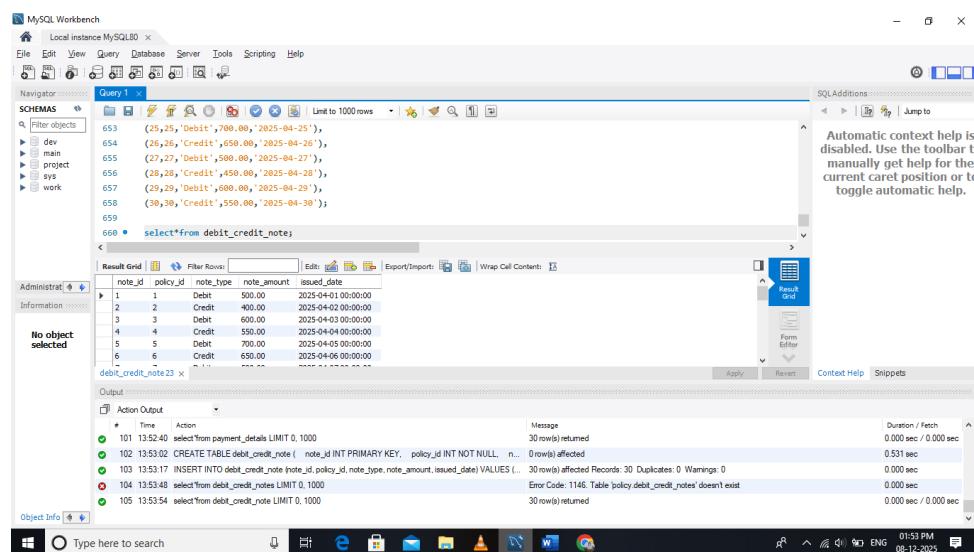
	payment_id	policy_id	payment_mode	amount	payment_date
1	1	Online	7500.00	2025-04-01 00:00:00	
2	2	Cash	5000.00	2025-04-02 00:00:00	
3	3	Online	6000.00	2025-04-03 00:00:00	
4	4	Cash	7500.00	2025-04-04 00:00:00	
5	5	Online	8000.00	2025-04-05 00:00:00	
6	6	Cash	7500.00	2025-04-06 00:00:00	

The status bar at the bottom shows the time as 01:52 PM and the date as 08-12-2025.

```
19) CREATE TABLE debit_credit_note (
    note_id INT PRIMARY KEY,
    policy_id INT NOT NULL,
    note_type VARCHAR(10) CHECK(note_type IN ('Debit','Credit')),
    note_amount DECIMAL(10,2) NOT NULL,
    issued_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY(policy_id) REFERENCES policy_master(policy_id) ON
    DELETE CASCADE ON UPDATE CASCADE
);

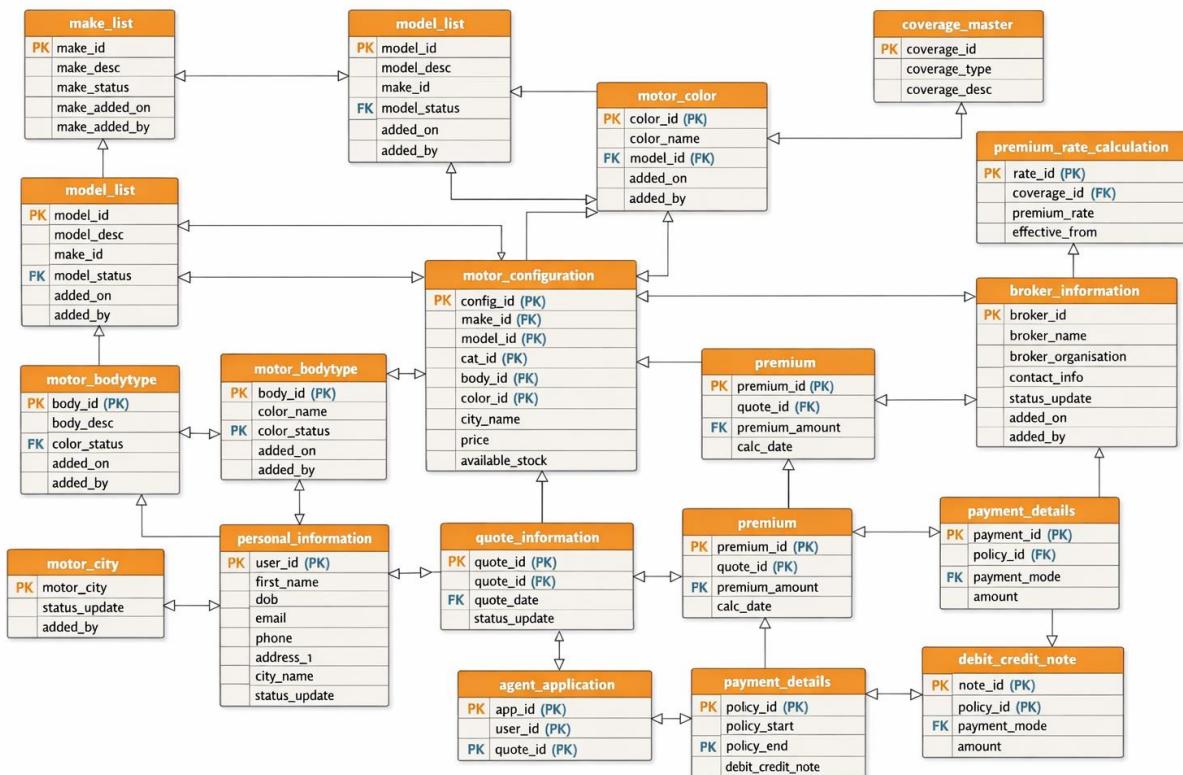
INSERT INTO debit_credit_note (note_id, policy_id, note_type, note_amount,
issued_date) VALUES
(1,1,'Debit',500.00,'2025-04-01'),
(2,2,'Credit',400.00,'2025-04-02'),
(3,3,'Debit',600.00,'2025-04-03'),
(4,4,'Credit',550.00,'2025-04-04'),
(5,5,'Debit',700.00,'2025-04-05'),
(6,6,'Credit',650.00,'2025-04-06'),
(7,7,'Debit',500.00,'2025-04-07'),
(8,8,'Credit',450.00,'2025-04-08'),
(9,9,'Debit',600.00,'2025-04-09'),
(10,10,'Credit',550.00,'2025-04-10'),
(11,11,'Debit',500.00,'2025-04-11'),
(12,12,'Credit',400.00,'2025-04-12'),
(13,13,'Debit',600.00,'2025-04-13'),
(14,14,'Credit',550.00,'2025-04-14'),
```

(15,15,'Debit',700.00,'2025-04-15'),
(16,16,'Credit',650.00,'2025-04-16'),
(17,17,'Debit',500.00,'2025-04-17'),
(18,18,'Credit',450.00,'2025-04-18'),
(19,19,'Debit',600.00,'2025-04-19'),
(20,20,'Credit',550.00,'2025-04-20'),
(21,21,'Debit',500.00,'2025-04-21'),
(22,22,'Credit',400.00,'2025-04-22'),
(23,23,'Debit',600.00,'2025-04-23'),
(24,24,'Credit',550.00,'2025-04-24'),
(25,25,'Debit',700.00,'2025-04-25'),
(26,26,'Credit',650.00,'2025-04-26'),
(27,27,'Debit',500.00,'2025-04-27'),
(28,28,'Credit',450.00,'2025-04-28'),
(29,29,'Debit',600.00,'2025-04-29'),
(30,30,'Credit',550.00,'2025-04-30');



ER DIAGRAM:

Entity – Relationship Diagram



QUERIES:

1) SELECT ALL

QUERY:

```
select*from motor_configuration;
```

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab contains the following SQL code:

```
676 JOIN motor_color clr ON mot.color_id = clr.color_id
677 WHERE mot.price > (0.5 * (SELECT MAX(price) FROM motor_configuration WHERE cat_id = mot.cat_id))
678 GROUP BY mc.make_desc, md1.model_desc
679 HAVING AVG(mot.price) > 500000
680 ORDER BY avg_price DESC
681 LIMIT 10 OFFSET 3;
682
683 • select*from motor_configuration;
```

The 'Result Grid' pane displays the results of the query, which is empty. The 'Output' pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
1	16:12:00	select*from motor_configuration LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in ...	0.109 sec
2	16:12:12	use policy	0 row(s) affected	0.000 sec
3	16:12:20	select*from motor_configuration LIMIT 0, 1000	30 row(s) returned	0.218 sec / 0.016 sec

2) SELECT SPECIFY COLUMN:

QUERY:

```
SELECT
```

```
make_id, make_desc, make_added_on
```

```
FROM
```

```
make_list;
```

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
dev
main
policy
Information
No object selected
Query 1 < SQL File 3*
680 ORDER BY avg_price DESC
681 LIMIT 10 OFFSET 3;
682
683 • select*from motor_configuration;
684
685 • select make_id,make_desc,make_added_on from make_list;
686
687
Result Grid | Filter Rows | Export/Import | Wrap Cell Content: 
make_id make_desc make_added_on
1 Honda 2025-01-01
2 Toyota 2025-01-02
3 Suzuki 2025-01-03
4 null null null
make_list 2 < Output
Action Output
# Time Action Message Duration / Fetch
1 16:12:00 select*from motor_configuration LIMIT 0, 1000 Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in ... 0.109 sec
0 row(s) affected
30 row(s) returned
0.218 sec / 0.016 sec
3 row(s) returned
0.172 sec / 0.000 sec
4 16:20:25 select make_id,make_desc,make_added_on from make_list LIMIT 0, 1000
Object Info

```

3) ARITHMATIC OPERATION:

QUERY:

SELECT

config_id, city_name, price, price + 20000

FROM

motor_configuration;

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
dev
main
policy
Information
No object selected
Query 1 < SQL File 3*
682
683 • SELECT *
684 FROM
685 motor_configuration;
686
687
688 • SELECT
689 make_id, make_desc, make_added_on
Result Grid | Filter Rows | Export/Import | Wrap Cell Content: 
config_id city_name price price+
1 Chennai 80000.00 820000.00
2 Comptore 850000.00 870000.00
3 Madurai 90000.00 920000.00
4 Trichy 750000.00 770000.00
5 Erode 820000.00 840000.00
6 Chennai 600000.00 620000.00
Result 3 < Output
Action Output
# Time Action Message Duration / Fetch
1 16:12:00 select*from motor_configuration LIMIT 0, 1000 Error Code: 1046. No database selected. Select the default DB to be used by double-clicking its name in ... 0.109 sec
0 row(s) affected
30 row(s) returned
0.218 sec / 0.016 sec
3 row(s) returned
0.172 sec / 0.000 sec
5 16:23:40 select config_id,city_name,price,price+20000 from motor_configuration LIMIT 0, 1000 30 row(s) returned
0.031 sec / 0.000 sec
Object Info

```

4) ALIAS NAME:

QUERY:

SELECT

```
make_id, make_desc, make_added_on AS ADDED_ON  
FROM  
make_list;
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database is set to 'Local instance MySQL80'. The main area displays a query window with the following SQL code:

```
684 *  
685 FROM  
686 motor_configuration;  
687  
688 • SELECT  
689 make_id, make_desc, make_added_on AS ADDED_ON  
690 FROM  
691 make_list;
```

The results grid shows the following data:

make_id	make_desc	ADDED_ON
1	Honda	2025-01-01
2	Toyota	2025-01-02
3	Suzuki	2025-01-03

Below the results, the 'Action Output' pane displays the following log entries:

#	Time	Action	Message	Duration / Fetch
2	16:12:12	use policy	0 row(s) affected	0.000 sec
3	16:12:20	select from motor_configuration LIMIT 0, 1000	30 row(s) returned	0.218 sec / 0.016 sec
4	16:20:25	select make_id,make_desc,make_added_on from make_list LIMIT 0, 1000	3 row(s) returned	0.172 sec / 0.000 sec
5	16:23:40	select config_id,city_name,price,price+ 20000 from motor_configuration LIMIT 0, 1000	30 row(s) returned	0.031 sec / 0.000 sec
6	16:27:14	SELECT make_id,make_desc,make_added_on AS ADDED_ON FROM make_list LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

5) WHERE CONDITION:

QUERY:

SELECT

```
config_id, city_name, price, price + 20000
```

FROM

```
motor_configuration
```

WHERE

```
City_name LIKE 'chennai';
```

```

690
691
692 • SELECT
693     config_id, city_name, price, price + 20000
694     FROM
695         motor_configuration
696     WHERE
697         City_name LIKE 'chennai';

```

config_id	city_name	price	price + 20000
1	Chennai	80000.00	82000.00
6	Chennai	92000.00	94000.00
11	Chennai	95000.00	97000.00
16	Chennai	101000.00	103000.00
21	Chennai	110000.00	112000.00
24	Chennai	120000.00	122000.00

Result 7 | Read Only | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
6	16:27:14	SELECT make_id,make_desc,make_added_on AS ADDED_ON FROM make_list LIMIT 0,1000	3 row(s) returned	0.000 sec / 0.000 sec
7	16:28:51	SELECT 'FROM motor_configuration LIMIT 0,1000'	30 row(s) returned	0.000 sec / 0.000 sec
8	16:29:04	SELECT 'FROM motor_configuration LIMIT 0,1000'	30 row(s) returned	0.000 sec / 0.000 sec
9	16:29:43	SELECT config_id,city_name,price,price + 20000 FROM motor_configuration WHERE CITY_...	Error Code: 1054. Unknown column 'CHENNAI' in 'where clause'	0.031 sec
10	16:30:27	SELECT config_id,city_name,price,price + 20000 FROM motor_configuration WHERE CITY_...	6 row(s) returned	0.093 sec / 0.000 sec

6) COMPARISON OPERATOR:

QUERY:

```

SELECT
    COVERAGE_ID, PREMIUM_RATE
    FROM
        PREMIUM_RATE_CALCULATION
    WHERE
        PREMIUM_RATE > 5550;

```

```

697
698
699 • SELECT
700     COVERAGE_ID, PREMIUM_RATE
701     FROM
702         PREMIUM_RATE_CALCULATION
703     WHERE
704         PREMIUM_RATE > 5550;

```

COVERAGE_ID	PREMIUM_RATE
2	7500.00
3	6000.00
4	8000.00
7	7000.00
8	6500.00
10	6000.00

PREMIUM_RATE_CALCULATION... | Read Only | Context Help | Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
9	16:29:43	SELECT config_id,city_name,price,price + 20000 FROM motor_configuration WHERE CITY_...	Error Code: 1054. Unknown column 'CHENNAI' in 'where clause'	0.031 sec
10	16:30:27	SELECT config_id,city_name,price,price + 20000 FROM motor_configuration WHERE CITY_...	6 row(s) returned	0.093 sec / 0.000 sec
11	16:35:01	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PRE...	0 row(s) returned	0.125 sec / 0.000 sec
12	16:35:08	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PRE...	0 row(s) returned	0.000 sec / 0.000 sec
13	16:35:14	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PRE...	6 row(s) returned	0.000 sec / 0.000 sec

7) IN:

QUERY:

SELECT

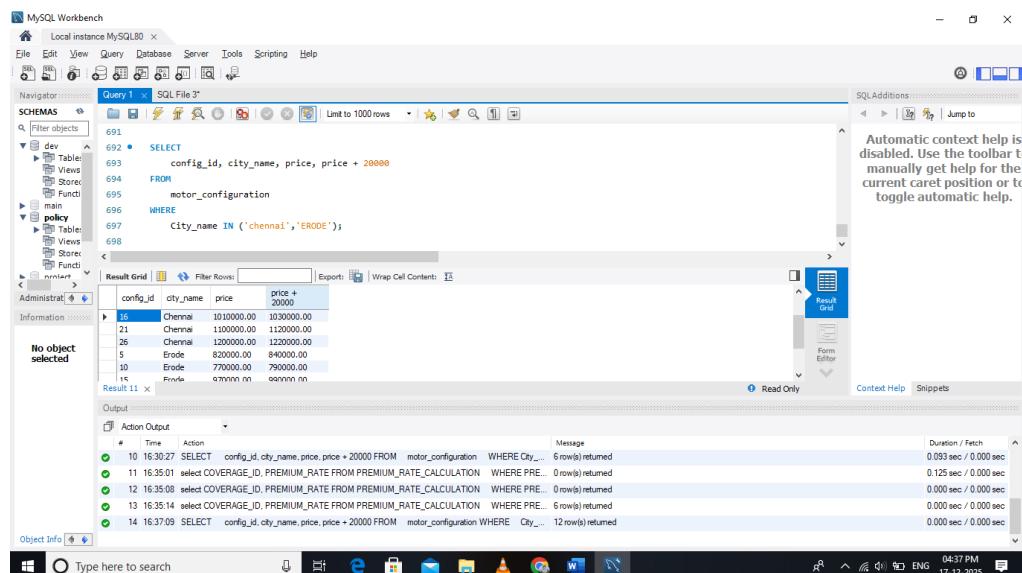
config id, city name, price, price + 20000

FROM

motor configuration

WHERE

```
City name IN ('chennai','ERODE');
```



8) NOT IN:

QUERY:

SELECT

config id, city name, price, price + 20000

FROM

motor configuration

WHERE

City_name NOT IN ('chennai','ERODE');

```

SELECT config_id, city_name, price, price + 20000
FROM motor_configuration
WHERE city_name NOT IN ('chennai', 'ERODE');

```

Result Grid:

config_id	city_name	price	price + 20000
2	Combabore	850000.00	870000.00
3	Medurai	900000.00	920000.00
4	Tirchi	750000.00	770000.00
7	Combabore	760000.00	780000.00
8	Medurai	880000.00	900000.00
9	Tirhu	910000.00	930000.00
10	Tiru	910000.00	930000.00
11	Tiru	910000.00	930000.00

Action Output:

Time	Action	Message	Duration / Fetch
11 16:35:01	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE > 5550;	0 row(s) returned	0.125 sec / 0.000 sec
12 16:35:08	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE > 5550;	0 row(s) returned	0.000 sec / 0.000 sec
13 16:35:14	select COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE > 5550;	0 row(s) returned	0.000 sec / 0.000 sec
14 16:37:09	SELECT config_id, city_name, price, price + 20000 FROM motor_configuration WHERE city_name NOT IN ('chennai', 'ERODE');	12 row(s) returned	0.000 sec / 0.000 sec
15 16:38:51	SELECT config_id, city_name, price, price + 20000 FROM motor_configuration WHERE city_name NOT IN ('chennai', 'ERODE');	18 row(s) returned	0.015 sec / 0.000 sec

9) LIKE OPERATOR:

QUERY:

select FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION

WHERE FIRST_NAME LIKE 'S_R__Y%';

```

SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION
WHERE FIRST_NAME LIKE 'S_R__Y';

```

Result Grid:

FIRST_NAME	EDUCATION	CITY_NAME	USER_ID	GENDER
Saranya	B.Tech	Combabore	2	Female

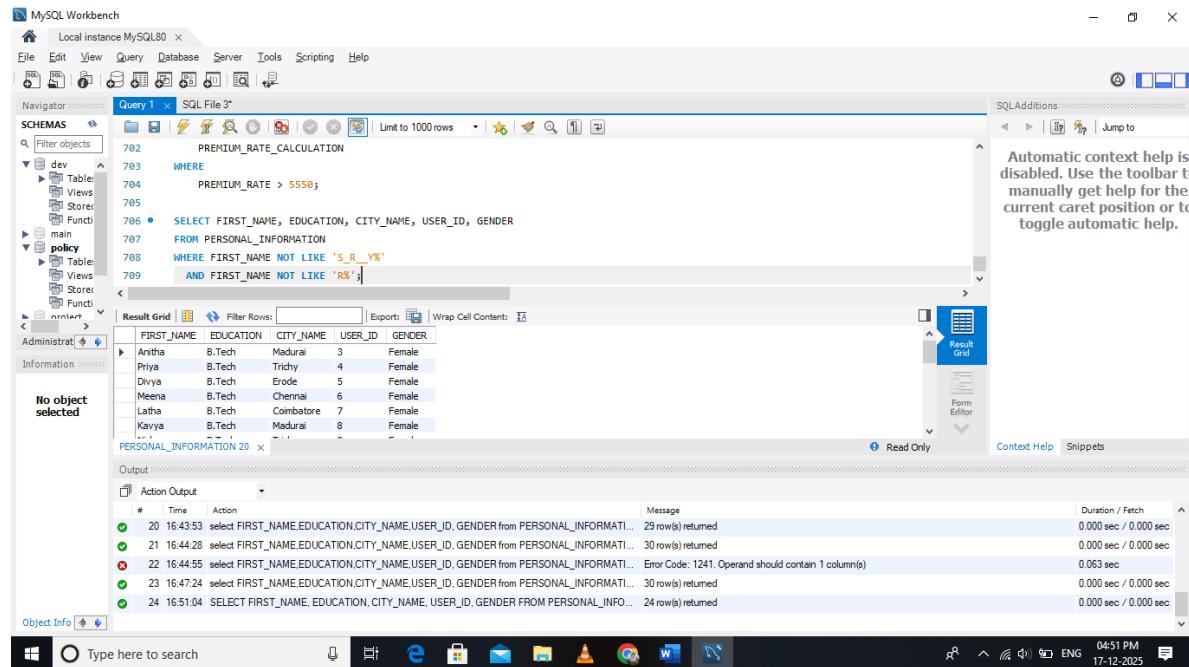
Action Output:

Time	Action	Message	Duration / Fetch
15 16:38:51	SELECT config_id, city_name, price, price + 20000 FROM motor_configuration WHERE city_name NOT IN ('chennai', 'ERODE');	18 row(s) returned	0.015 sec / 0.000 sec
16 16:42:13	select FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION WHERE FIRST_NAME LIKE 'S_R__Y';	5 row(s) returned	0.109 sec / 0.000 sec
17 16:42:18	select FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION WHERE FIRST_NAME LIKE 'S_R__Y';	5 row(s) returned	0.000 sec / 0.000 sec
18 16:42:45	select FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION WHERE FIRST_NAME LIKE 'S_R__Y';	0 row(s) returned	0.000 sec / 0.000 sec
19 16:42:59	select FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER from PERSONAL_INFORMATION WHERE FIRST_NAME LIKE 'S_R__Y';	1 row(s) returned	0.000 sec / 0.000 sec

10) NOT LIKE AND LOGICAL OPERATOR:

QUERY:

```
SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER  
FROM PERSONAL_INFORMATION  
WHERE FIRST_NAME NOT LIKE 'S_R__Y%'  
AND FIRST_NAME NOT LIKE 'R%';
```



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

```
PREMIUM_RATE_CALCULATION  
WHERE  
PREMIUM_RATE > 5550;  
SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER  
FROM PERSONAL_INFORMATION  
WHERE FIRST_NAME NOT LIKE 'S_R__Y%'  
AND FIRST_NAME NOT LIKE 'R%';
```

The results grid displays the following data:

FIRST_NAME	EDUCATION	CITY_NAME	USER_ID	GENDER
Anitha	B.Tech	Madurai	3	Female
Priya	B.Tech	Tiru	4	Female
Divya	B.Tech	Erode	5	Female
Meena	B.Tech	Chennai	6	Female
Latha	B.Tech	Combatore	7	Female
Kavya	B.Tech	Madurai	8	Female
...

The status bar at the bottom right indicates the results were read-only and returned 29 rows.

11) BETWEEN AND:

QUERY:

```
SELECT  
COVERAGE_ID, PREMIUM_RATE  
FROM  
PREMIUM_RATE_CALCULATION
```

WHERE

PREMIUM_RATE between 5500 AND 8000 ;

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
697 City_name NOT IN ('chennai','ERODE');
698 
699 •   SELECT
700     COVERAGE_ID, PREMIUM_RATE
701   FROM
702     PREMIUM_RATE_CALCULATION
703   WHERE
704     PREMIUM_RATE between 5500 AND 8000 ;
```

The result grid displays the following data:

COVERAGE_ID	PREMIUM_RATE
2	7500.00
3	6000.00
4	8000.00
5	5500.00
7	7000.00
8	6500.00

The status bar at the bottom right shows the time as 04:54 PM and the date as 17-12-2025.

12) NOT BETWEEN AND:

QUERY:

```
SELECT
    COVERAGE_ID, PREMIUM_RATE
  FROM
    PREMIUM_RATE_CALCULATION
 WHERE
    PREMIUM_RATE NOT between 5500 AND 8000 ;
```

MySQL Workbench - Local instance MySQL80

Query 1 SQL File 3

```

697 City_name NOT IN ('chennai', 'ERODE');
698 • SELECT COVERAGE_ID, PREMIUM_RATE
699 FROM PREMIUM_RATE_CALCULATION
700 WHERE PREMIUM_RATE NOT between 5500 AND 8000;
701
702
703
704
    
```

Result Grid

COVERAGE_ID	PREMIUM_RATE
1	5000.00
6	4500.00
9	5000.00

Action Output

#	Time	Action	Message	Duration / Fetch
24	16:51:04	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFORMATION	24 row(s) returned	0.000 sec / 0.000 sec
25	16:53:53	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	0 row(s) returned	0.000 sec / 0.000 sec
26	16:54:04	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	7 row(s) returned	0.000 sec / 0.000 sec
27	16:54:54	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
28	16:55:35	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec

Object Info

Type here to search

Output

Action Output

#	Time	Action	Message	Duration / Fetch
25	16:53:53	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	0 row(s) returned	0.000 sec / 0.000 sec
26	16:54:04	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	7 row(s) returned	0.000 sec / 0.000 sec
27	16:54:54	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
28	16:55:35	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec

04:59 PM 17-12-2025

13) ORDER BY:

QUERY:

```

SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
FROM PERSONAL_INFORMATION
WHERE FIRST_NAME NOT LIKE 'S_R__Y%'
AND FIRST_NAME NOT LIKE 'R%'
order by FIRST_NAME desc;
    
```

MySQL Workbench - Local instance MySQL80

Query 1 SQL File 3

```

703 WHERE PREMIUM_RATE NOT between 5500 AND 8000;
704
705
706 • SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
707 FROM PERSONAL_INFORMATION
708 WHERE FIRST_NAME NOT LIKE 'S_R__Y%'
709 AND FIRST_NAME NOT LIKE 'R%';
710 order by FIRST_NAME desc;
    
```

Result Grid

FIRST_NAME	EDUCATION	CITY_NAME	USER_ID	GENDER
Vishal	B.Tech	Medura	18	Male
Vikram	B.Tech	Trichy	14	Male
Vijay	B.Tech	Erode	25	Male
Swathi	B.Tech	Erode	10	Female
Surendra	B.Tech	Trichy	24	Male
Shankar	B.Tech	Erode	30	Male

Action Output

#	Time	Action	Message	Duration / Fetch
25	16:53:53	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	0 row(s) returned	0.000 sec / 0.000 sec
26	16:54:04	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	7 row(s) returned	0.000 sec / 0.000 sec
27	16:54:54	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
28	16:55:35	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
29	16:59:25	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFORMATION	24 row(s) returned	0.063 sec / 0.000 sec

Object Info

Type here to search

Output

Action Output

#	Time	Action	Message	Duration / Fetch
25	16:53:53	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	0 row(s) returned	0.000 sec / 0.000 sec
26	16:54:04	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	7 row(s) returned	0.000 sec / 0.000 sec
27	16:54:54	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
28	16:55:35	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE PREMIUM_RATE NOT between 5500 AND 8000;	3 row(s) returned	0.000 sec / 0.000 sec
29	16:59:25	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFORMATION	24 row(s) returned	0.063 sec / 0.000 sec

04:59 PM 17-12-2025

14) BACKUP TABLE:

QUERY:

CREATE TABLE INFORMATION AS

```
select*FROM PERSONAL_INFORMATION;
```

```
select*FROM INFORMATION;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** dev, main, policy.
- Tables:** INFORMATION.
- Query Editor:** Shows the SQL code for creating the table and inserting data.
- Result Grid:** Displays the data from the INFORMATION table.
- Action Output:** Shows the log of actions taken, including the creation of the table and the insertion of 30 rows.

user_id	user_type	first_name	last_name	gender	dob	email	marital_status	education	phone	address_1	permanent_addr
1	Customer	Ramalakshmi	V	Female	2002-05-03	ramali@example.com	0	B.Tech	9000000001	Address 1	Permanent 1
2	Customer	Saranya	R	Female	2002-06-15	saranya@example.com	0	B.Tech	9000000002	Address 2	Permanent 2
3	Customer	Anitha	K	Female	2002-07-20	anitha3@example.com	0	B.Tech	9000000003	Address 3	Permanent 3
4	Customer	Priya	L	Female	2002-09-11	priya@example.com	0	B.Tech	9000000004	Address 4	Permanent 4
5	Customer	Divya	M	Female	2002-09-05	divya5@example.com	0	B.Tech	9000000005	Address 5	Permanent 5

15) LIMIT:

QUERY:

```
SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
```

```
FROM PERSONAL_INFORMATION
```

```
WHERE FIRST_NAME NOT LIKE 'S_R__Y%'
```

```
AND FIRST_NAME NOT LIKE 'R%'
```

```
order by FIRST_NAME desc
```

```
limit 10;
```

```

SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
FROM PERSONAL_INFORMATION
WHERE FIRST_NAME NOT LIKE 'S_R_%'
AND FIRST_NAME NOT LIKE 'R%'
order by FIRST_NAME desc
limit 10;

```

The screenshot shows the MySQL Workbench interface. In the Navigator panel, the schema 'dev' is selected, showing tables like 'PERSONAL_INFORMATION'. The Query Editor contains the provided SQL code. The Results Grid displays the query results:

FIRST_NAME	EDUCATION	CITY_NAME	USER_ID	GENDER
Vishal	B.Tech	Madurai	18	Male
Vikram	B.Tech	Trichy	14	Male
Vijay	B.Tech	Erode	25	Male
Swathi	B.Tech	Erode	10	Female
Suresh	B.Tech	Trichy	24	Male
Shankar	B.Tech	Erode	30	Male
			..	

The Action Output pane shows the following log:

#	Time	Action	Message	Duration / Fetch
28	16:55:35	SELECT COVERAGE_ID, PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION WHERE...	3 row(s) returned	0.000 sec / 0.000 sec
29	16:59:25	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	24 row(s) returned	0.063 sec / 0.000 sec
30	17:02:56	CREATE TABLE INFORMATION AS select*FROM PERSONAL_INFORMATION	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	1.532 sec
31	17:03:34	select*FROM INFORMATION LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
32	17:05:33	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	10 row(s) returned	0.000 sec / 0.000 sec

16) OFFSET:

QUERY:

```

SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
FROM PERSONAL_INFORMATION
WHERE FIRST_NAME LIKE 'S%'
order by FIRST_NAME ASC
limit 5
offset 3

```

```

SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER
FROM PERSONAL_INFORMATION
WHERE FIRST_NAME LIKE 'S%'
order by FIRST_NAME ASC
limit 5
offset 3;

```

FIRST_NAME	EDUCATION	CITY_NAME	USER_ID	GENDER
Suresh	B.Tech	Trichy	24	Male
Swathi	B.Tech	Erode	10	Female

PERSONAL_INFORMATION 33 ×

Action Output

#	Time	Action	Message	Duration / Fetch
34	17:07:10	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	10 row(s) returned	0.000 sec / 0.000 sec
35	17:07:22	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	0 row(s) returned	0.000 sec / 0.000 sec
36	17:07:36	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	0 row(s) returned	0.000 sec / 0.000 sec
37	17:08:00	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	5 row(s) returned	0.000 sec / 0.000 sec
38	17:08:19	SELECT FIRST_NAME, EDUCATION, CITY_NAME, USER_ID, GENDER FROM PERSONAL_INFO...	2 row(s) returned	0.016 sec / 0.000 sec

17) INFORMATION SCHEMA:

QUERY:

```

SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'POLICY';

```

```

CREATE TABLE INFORMATION AS
select*FROM PERSONAL_INFORMATION;

select*FROM INFORMATION;

SELECT TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'POLICY';

```

TABLE_NAME
agent_application
broker_commission
broker_information
category_type
coverage_master
debt_credit_note
etc...

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:29:24	SELECT TABLE_NAME FROM INFORMATION_SCHEMA TABLES WHERE TABLE_SCHEMA = 'POLI...' 20 row(s) returned		0.000 sec / 0.000 sec

18) AGGREGATE FUNCTION:

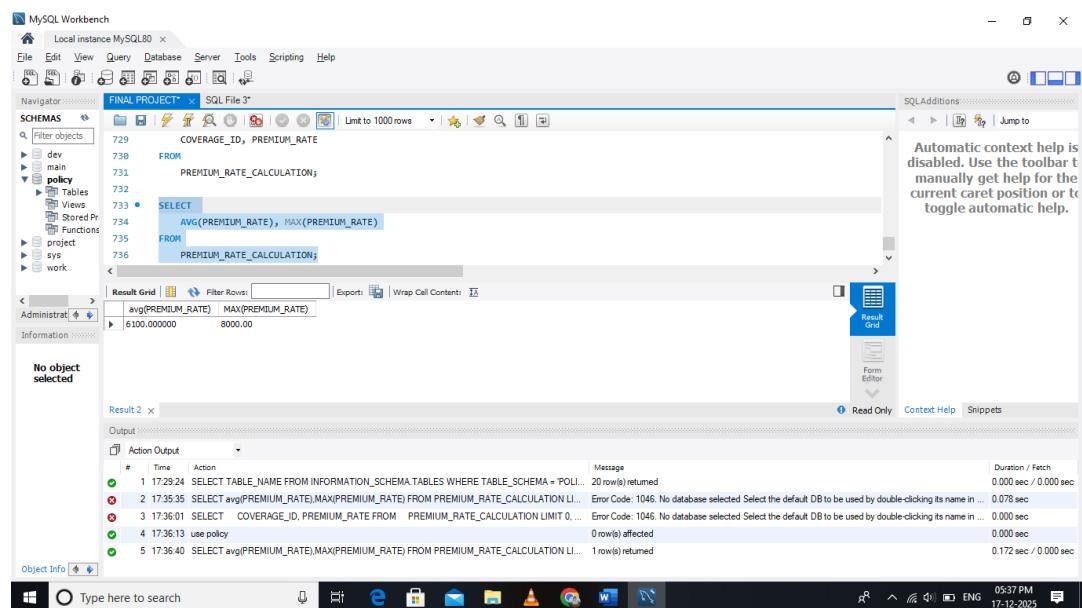
QUERY:

SELECT

AVG(PREMIUM_RATE), MAX(PREMIUM_RATE)

FROM

PREMIUM_RATE_CALCULATION;



```
MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator FINAL PROJECT* SQL File 3*
SCHEMAS dev main policy
Tables Views Stored Procedures Functions project sys work
Information No object selected
Result Grid avg(PREMIUM_RATE) MAX(PREMIUM_RATE)
6100.000000 8000.00
Result 2
Output Action Output
# Time Action Message Duration / Fetch
1 17:29:24 SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'POLI...' 20 row(s) returned 0.000 sec / 0.000 sec
2 17:35:35 SELECT avg(PREMIUM_RATE),MAX(PREMIUM_RATE) FROM PREMIUM_RATE_CALCULATION L... Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in ... 0.078 sec
3 17:36:01 SELECT COVERAGE_ID,PREMIUM_RATE FROM PREMIUM_RATE_CALCULATION LIMIT 0... Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in ... 0.000 sec
4 17:36:13 use policy 0 row(s) affected 0.000 sec
5 17:36:40 SELECT avg(PREMIUM_RATE),MAX(PREMIUM_RATE) FROM PREMIUM_RATE_CALCULATION L... 1 row(s) returned 0.172 sec / 0.000 sec
Object Info
Type here to search
R ENG 09:37 PM 17-12-2025
```

19) GROUP BY:

QUERY:

SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION

group by CITY_NAME;

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area displays a query editor titled 'FINAL PROJECT*' containing the following SQL code:

```

736 PREMIUM_RATE_CALCULATION;
737
738
739 •    select*FROM MOTOR_CONFIGURATION;
740
741 •    SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION
742     group by CITY_NAME;
743

```

The results grid shows the following data:

Avg(price)	City_name
996666.666667	Chennai
986666.666667	Combatore
998333.333333	Erode
1028333.333333	Madurai
1008333.333333	Trichy

The 'Output' section below the results grid displays the execution log:

#	Time	Action	Message	Duration / Fetch
17	17:46:35	select*FROM MOTOR_CONFIGURATION LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
18	17:46:52	SELECT MAKE_ID, AVG(PRICE).UPDATED_ON FROM MOTOR_CONFIGURATION group by CRE...	Error Code: 1055. Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregat...	0.000 sec
19	17:47:42	SELECT AVG(PRICE) FROM MOTOR_CONFIGURATION group by CREATED_ON LIMIT 0, 1000	1 row(s) returned	0.063 sec / 0.000 sec
20	17:47:56	SELECT AVG(PRICE) FROM MOTOR_CONFIGURATION group by CITY_NAME LIMIT 0, 1000	5 row(s) returned	0.125 sec / 0.000 sec
21	17:48:09	SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME LIMI...	5 row(s) returned	0.000 sec / 0.000 sec

20)HAVING:

QUERY:

SELECT

CITY_NAME,

AVG(PRICE) AS AVG_PRICE

FROM MOTOR_CONFIGURATION

GROUP BY CITY_NAME

HAVING MAX(CAT_ID) = 3;

```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: FINAL PROJECT* x SQL File 3*
SCHEMAS Filter objects
dev main policy project sys work
Tables Views Stored Pr Functions
730 • select*FROM MOTOR_CONFIGURATION;
740
741 • | SELECT
742 | CITY_NAME,
743 | AVG(PRICE) AS AVG_PRICE
744 | FROM MOTOR_CONFIGURATION
745 | GROUP BY CITY_NAME
746 | HAVING MAX(CAT_ID) = 3;
747 |
Result Grid Filter Rows: Export: Wrap Cell Content: □
CITY_NAME AVG_PRICE
Chennai 996666.666667
Combbatore 986666.666667
Erode 998333.333333
Madurai 1028333.333333
Trichy 1008333.333333
Result 23 x
Output:
Action Output
# Time Action Message Duration / Fetch
35 17:54:16 SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME H... 0 row(s) returned 0.000 sec / 0.000 sec
36 17:54:19 SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME H... 0 row(s) returned 0.000 sec / 0.000 sec
37 17:54:27 SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME H... 0 row(s) returned 0.000 sec / 0.000 sec
38 17:55:41 SELECT CITY_NAME, AVG(PRICE) AS AVG_PRICE FROM MOTOR_CONFIGURATION GRO... 0 row(s) returned 0.078 sec / 0.000 sec
39 20:09:57 SELECT CITY_NAME, AVG(PRICE) AS AVG_PRICE FROM MOTOR_CONFIGURATION GRO... 5 row(s) returned 0.000 sec / 0.000 sec
Object Info

```

21) JOINS:

QUERY:

SELECT

```

mc.make_desc AS make,
ml.model_desc AS model,
clr.color_name AS color,
mb.body_desc AS body_type,
cfg.city_name,
cfg.price,
cfg.available_stock

```

FROM motor_configuration cfg

JOIN make_list mc ON cfg.make_id = mc.make_id

JOIN model_list ml ON cfg.model_id = ml.model_id

JOIN motor_color clr ON cfg.color_id = clr.color_id

```

JOIN motor_bodytype mb ON cfg.body_id = mb.body_id
ORDER BY cfg.price DESC;

```

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

744 FROM MOTOR_CONFIGURATION
745 GROUP BY CITY_NAME
746 HAVING MAX(CAT_ID) = 3;
747
748 • SELECT
749     mc.make_desc AS make,
750     ml.model_desc AS model,
751     clr.color_name AS color,

```

The result grid displays the following data:

make	model	color	body_type	city_name	price	available_stock
Suzuki	Suzuki S-Cross	corl orange	Van	Trichy	1240000.00	4
Suzuki	Suzuki XL7	teal Green	Coupe	Coimbatore	1220000.00	6
Suzuki	Suzuki Grand Vitara	teal Brown	Hybrid	Erode	1210000.00	6
Suzuki	Suzuki Ertiga	olive Silver	Convertible	Chennai	1200000.00	8
Suzuki	Suzuki Brezza	gold Yellow	Pickup	Madurai	1190000.00	7
Suzuki	Suzuki Vitara	olive Black	SUV	Madurai	1180000.00	9

The output pane shows the execution log:

```

# Time Action Duration / Fetch
36 17:54:19 SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME HAVING COUNT(CAT_ID)=3 0.000 sec / 0.000 sec
37 17:54:27 SELECT AVG(PRICE),CITY_NAME FROM MOTOR_CONFIGURATION group by CITY_NAME HAVING COUNT(CAT_ID)=3 0.000 sec / 0.000 sec
38 17:55:41 SELECT CITY_NAME, AVG(PRICE) AS AVG_PRICE FROM MOTOR_CONFIGURATION GROUP BY CITY_NAME 0.078 sec / 0.000 sec
39 20:09:57 SELECT CITY_NAME, AVG(PRICE) AS AVG_PRICE FROM MOTOR_CONFIGURATION GROUP BY CITY_NAME 0.000 sec / 0.000 sec
40 21:44:24 SELECT mc.make_desc AS make, ml.model_desc AS model, clr.color_name AS color, mb.body_id AS body_id, mc.price AS price, mc.availab... 0.109 sec / 0.000 sec

```

22) SUBQUERY:

QUERY:

SELECT

user_id,

first_name,

email,

city_name

FROM personal_information

WHERE user_id IN (

SELECT user_id

FROM quote_information

WHERE coverage_id = (

```

SELECT coverage_id
FROM coverage_master
WHERE coverage_type = 'Vehicle'
)
);

```

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the stored procedure code.
- Result Grid:** Displays the output of the query, showing 13 rows of user information.
- Output Panel:** Shows the execution log with 5 entries, each detailing a SELECT query and its duration.
- System Tray:** Shows the date and time as 09:46 PM on 17-12-2025.

user_id	first_name	email	city_name
1	Ramalakshmi	ramal1@example.com	Chennai
4	Priya	priya4@example.com	Trichy
6	Meena	meen6@example.com	Chennai
8	Kavya	kavya8@example.com	Madurai
11	Raja	raja11@example.com	Chennai
13	Karthik	karthik13@example.com	Madurai
...

23) PROCEDURE:

QUERY:

DELIMITER \$\$

```
CREATE PROCEDURE get_customers_by_city(IN p_city VARCHAR(30))
```

```
BEGIN
```

```
SELECT
```

```
user_id,
```

```

CONCAT(first_name,'',last_name) AS customer_name,
email,
phone,
city_name

FROM personal_information

WHERE city_name = p_city;

END $$
```

DELIMITER ;

```
CALL get_customers_by_city('chennai');
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** FINAL PROJECT -> SQL File 3*. Shows the schema structure with tables like dev, main, policy, project, sys, work, and a newly created table named policy.
- SQL Editor:** Displays the stored procedure code. The code includes a delimiter \$\$, a create procedure statement with parameters p_city VARCHAR(30), a begin block, and a select statement returning user_id, first_name, email, phone, and city_name.
- Result Grid:** Shows the output of the SELECT query. The results are as follows:

	user_id	customer_name	email	phone	city_name
1	1	Ramalakshmi V	ramal1@example.com	9000000001	Chennai
6	6	Meena N	meenat@example.com	9000000006	Chennai
11	11	Raja S	raja11@example.com	9000000011	Chennai
16	16	Harish X	harish16@example.com	9000000016	Chennai
21	21	Ajith C	ajith21@example.com	9000000021	Chennai
26	26	Deepak H	deepak26@example.com	9000000026	Chennai

- Action Output:** Shows the history of actions taken. The log includes:
 - 41 21:45:52 SELECT user_id, first_name, email, city_name FROM personal_information WHERE user_id...
 - 42 21:48:05 CREATE PROCEDURE get_customers_by_city(IN p_city VARCHAR(30)) BEGIN SELECT us... 0 rows affected
 - 43 21:50:04 CALL get_customers_by_city('Chennai')
 - 44 21:50:19 CALL get_customers_by_city('erode')
 - 45 21:51:03 CALL get_customers_by_city('chennai')
- System Bar:** Shows the Windows taskbar with various application icons and system status.

24) FUNCTIONS:

QUERY:

DELIMITER \$\$

```
CREATE FUNCTION calculate_age(p_dob DATE)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, p_dob, CURDATE());
END $$
```

DELIMITER ;

```
SELECT user_id, calculate_age(dob) AS age, CITY_NAME
FROM personal_information;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is 'main'. Other schemas listed include 'de', 'main', 'policy', 'temp', 'sys', and 'work'.
- SQL Editor:** The SQL tab contains the following code:

```
794
795 • CALL get_customers_by_city('chennai');
796
797 DELIMITER $$
```

```
798
799 • CREATE FUNCTION calculate_age(p_dob DATE)
800 RETURNS INT
801 DETERMINISTIC
```
- Result Grid:** The Result Grid shows the output of the SELECT query:

user_id	age	CITY_NAME
1	23	Chennai
2	23	Coimbatore
3	23	Madurai
4	23	Trichy
5	23	Erode
6	23	Chennai
- Action Output:** The Action Output pane shows the history of actions taken in the session, including the creation of the function and its execution.

25) TRIGGERS:

QUERY:

```
CREATE TABLE premium_history (
    history_id INT AUTO_INCREMENT PRIMARY KEY,
    premium_id INT,
    quote_id INT,
    old_premium_amount DECIMAL(10,2),
    old_calc_date DATE,
    updated_on DATETIME
);
```

```
DELIMITER $$
```

```
CREATE TRIGGER trg_premium_backup_before_update
BEFORE UPDATE ON premium
FOR EACH ROW
BEGIN
    INSERT INTO premium_history
    (
        premium_id,
        quote_id,
        old_premium_amount,
        old_calc_date,
```

```
        updated_on  
    )  
VALUES  
(  
    OLD.premium_id,  
    OLD.quote_id,  
    OLD.premium_amount,  
    OLD.calc_date,  
    NOW()  
);  
END $$
```

DELIMITER ;

```
UPDATE premium  
SET premium_amount = 5500  
WHERE premium_id = 1;
```

```
SELECT * FROM premium_history;
```

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema structure under 'SCHEMAS'. A query editor window titled 'FINAL PROJECT*' contains the following SQL code:

```

DELIMITER ;
UPDATE premium
SET premium_amount = 5500
WHERE premium_id = 1;
SELECT * FROM premium_history;

```

The results grid shows one row of data from the 'premium_history' table:

history_id	premium_id	quote_id	old_premium_amount	old_calc_date	updated_on
1	1	7500.00	NULL	2025-03-02	2025-12-17 22:17:28

The bottom pane shows the 'Action History' with the following log entries:

#	Time	Action	Message	Duration / Fetch
56	22:10:52	DROP trigger trg_quote_default_status	0 row(s) affected	0.235 sec
57	22:16:57	CREATE TABLE premium_history (history_id INT AUTO_INCREMENT PRIMARY KEY, premium...)	0 row(s) affected	0.609 sec
58	22:17:08	CREATE TRIGGER trg_premium_backup_before_update BEFORE UPDATE ON premium FOR EACH...	0 row(s) affected	0.266 sec
59	22:17:28	UPDATE premium SET premium_amount = 5500 WHERE premium_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.265 sec
60	22:17:39	SELECT * FROM premium_history LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

26) VIEW:

QUERY:

CREATE VIEW vw_customer_details AS

SELECT

```

user_id,
CONCAT(first_name, ' ', last_name) AS customer_name,
email,
city_name

```

FROM personal_information;

SELECT * FROM vw_customer_details; CREATE VIEW vw_customer_details AS

SELECT

```

user_id,
CONCAT(first_name,' ',last_name) AS customer_name,
email,
city_name
FROM personal_information;

```

```
SELECT * FROM vw_customer_details;
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** dev, main, policy (selected), project, sys, work.
- SQL Editor:** FINAL PROJECT - SQL File 3 contains the following code:

```

861
862 • UPDATE premium
863     SET premium_amount = 5500
864     WHERE premium_id = 1;
865
866 • SELECT * FROM premium_history;
867
868 • CREATE VIEW vw_customer_details AS

```
- Result Grid:** Displays the results of the SELECT query:

#	user_id	customer_name	email	city_name
1	Ramalakshmi V	ramal1@example.com	Chennai	
2	Saranya R	saranya2@example.com	Coimbatore	
3	Anitha K	anitha3@example.com	Madurai	
4	Priya L	priya4@example.com	Trichy	
5	Divya M	divya5@example.com	Erode	
6	Meena N	meena6@example.com	Chennai	
- Output:** Shows the log of actions taken:

#	Action	Time	Message	Duration / Fetch	
58	CREATE TRIGGER	22:17:08	trg_premium_backup_before_update BEFORE UPDATE ON premium FOR EACH...	0.265 sec	
59	UPDATE	22:17:28	premium SET premium_amount = 5500 WHERE premium_id = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.265 sec
60	SELECT	22:17:39	* FROM premium_history LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
61	CREATE VIEW	22:24:04	vw_customer_details AS SELECT user_id, CONCAT(first_name,' ',last_name) AS...	0 row(s) affected	0.140 sec
62	SELECT	22:24:15	* FROM vw_customer_details LIMIT 0, 1000	30 row(s) returned	0.016 sec / 0.000 sec

27) INDEX:

QUERY:

```

CREATE INDEX idx_city
ON personal_information(city_name);

```

```
SELECT *
```

```
FROM personal_information  
WHERE city_name = 'Chennai';
```

```
EXPLAIN  
SELECT *  
FROM personal_information  
WHERE city_name = 'Chennai';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** policy
- Queries:** FINAL PROJECT* -> SQL File 3
- Code:**

```
871    CONCAT(first_name, ' ', last_name) AS customer_name,  
872    email,  
873    city_name  
874    FROM personal_information;  
875  
876 •     SELECT * FROM vw_customer_details;  
877  
878 •     CREATE INDEX idx_city
```
- Result Grid:** Shows the creation of an index for the column 'city_name'.
- Action Output:** Displays the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
61	22:24:04	CREATE VIEW vw_customer_details AS SELECT user_id, CONCAT(first_name, ' ', last_name) AS customer_name, email, city_name FROM personal_information;	0 row(s) affected	0.140 sec
62	22:24:15	SELECT * FROM vw_customer_details LIMIT 0, 1000	30 row(s) returned	0.016 sec / 0.000 sec
63	22:28:34	CREATE INDEX idx_city ON personal_information(city_name)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	1.484 sec
64	22:28:50	SELECT * FROM personal_information WHERE city_name = 'Chennai' LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
65	22:29:12	EXPLAIN SELECT * FROM personal_information WHERE city_name = 'Chennai'	1 row(s) returned	0.079 sec / 0.000 sec

28) OVERALL QUERY:

```
CREATE VIEW vw_customer_premium_summary AS  
SELECT DISTINCT  
    p.user_id AS customer_id,  
    CONCAT(p.first_name, ' ', p.last_name) AS customer_name,  
    c.coverage_type AS coverage,
```

```
COUNT(q.quote_id) AS total_quotes,  
SUM(pr.premium_amount) AS total_premium  
FROM personal_information p  
JOIN quote_information q ON p.user_id = q.user_id  
JOIN coverage_master c ON q.coverage_id = c.coverage_id  
JOIN premium pr ON q.quote_id = pr.quote_id  
WHERE  
    p.city_name LIKE 'C%'  
    AND p.email IS NOT NULL  
    AND pr.premium_amount BETWEEN 1000 AND 10000  
    AND q.coverage_id IN (  
        SELECT coverage_id  
        FROM coverage_master  
        WHERE coverage_type LIKE '%Vehicle%'  
    )  
GROUP BY  
    p.user_id, p.first_name, p.last_name, c.coverage_type  
HAVING  
    COUNT(q.quote_id) >= 1;  
  
SELECT * FROM vw_customer_premium_summary;
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area displays a query editor titled 'FINAL PROJECT*' containing the following SQL code:

```

967     )
968   GROUP BY
969   p.user_id, p.first_name, p.last_name, c.coverage_type
970
971   HAVING
972     COUNT(q.quote_id) >= 1;
973
974   SELECT * FROM vw_customer_premium_summary;

```

Below the code, the 'Result Grid' shows the output of the query:

customer_id	customer_name	coverage	total_quotes	total_premium
1	Ramalakshmi V	Vehicle	1	5500.00
6	Meena N	Vehicle	1	7500.00
11	Raja S	Vehicle	1	7500.00
17	Ramesh Y	Vehicle	1	7500.00
21	Ajith C	Vehicle	1	7500.00
27	Ganesh I	Vehicle	1	7500.00

The 'Output' section at the bottom shows the 'Action Output' log:

#	Time	Action	Message	Duration / Fetch
69	22:45:37	CREATE OR REPLACE VIEW vehicle_payments_view AS SELECT ...	Error Code: 1054. Unknown column 'mc.mc_id' in 'on clause'	0.015 sec
70	22:46:51	SELECT DISTINCT mk.make_desc AS vehicle_make, ...	Error Code: 1054. Unknown column 'pm.config_id' in 'on clause'	0.000 sec
71	22:48:21	SELECT DISTINCT mk.make_desc AS vehicle_make, ...	Error Code: 1054. Unknown column 'pm.config_id' in 'on clause'	0.000 sec
72	22:48:53	CREATE VIEW vw_customer_premium_summary AS SELECT DISTINCT ...	0 row(s) affected	0.156 sec
73	22:49:08	SELECT * FROM vw_customer_premium_summary LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

CONCLUSION:

This project successfully designed and implemented a **relational database system for an Insurance Policy Management System** using SQL. The database structure efficiently manages customer details, vehicle information, insurance coverage, quotations, premium calculations, policies, payments, and broker commissions through well-defined tables and relationships.

By applying **normalization principles, primary keys, and foreign key constraints**, data redundancy was minimized and data integrity was ensured. Advanced SQL concepts such as **JOINS, subqueries, views, indexes, stored procedures, functions, triggers, and access control** were effectively used to retrieve, secure, and maintain data efficiently.

The system enables accurate premium calculation, policy generation, payment tracking, and reporting, making it reliable and scalable for real-world insurance operations. Overall, this project demonstrates the effective use of database design and SQL techniques to build a robust, secure, and user-friendly insurance management system.