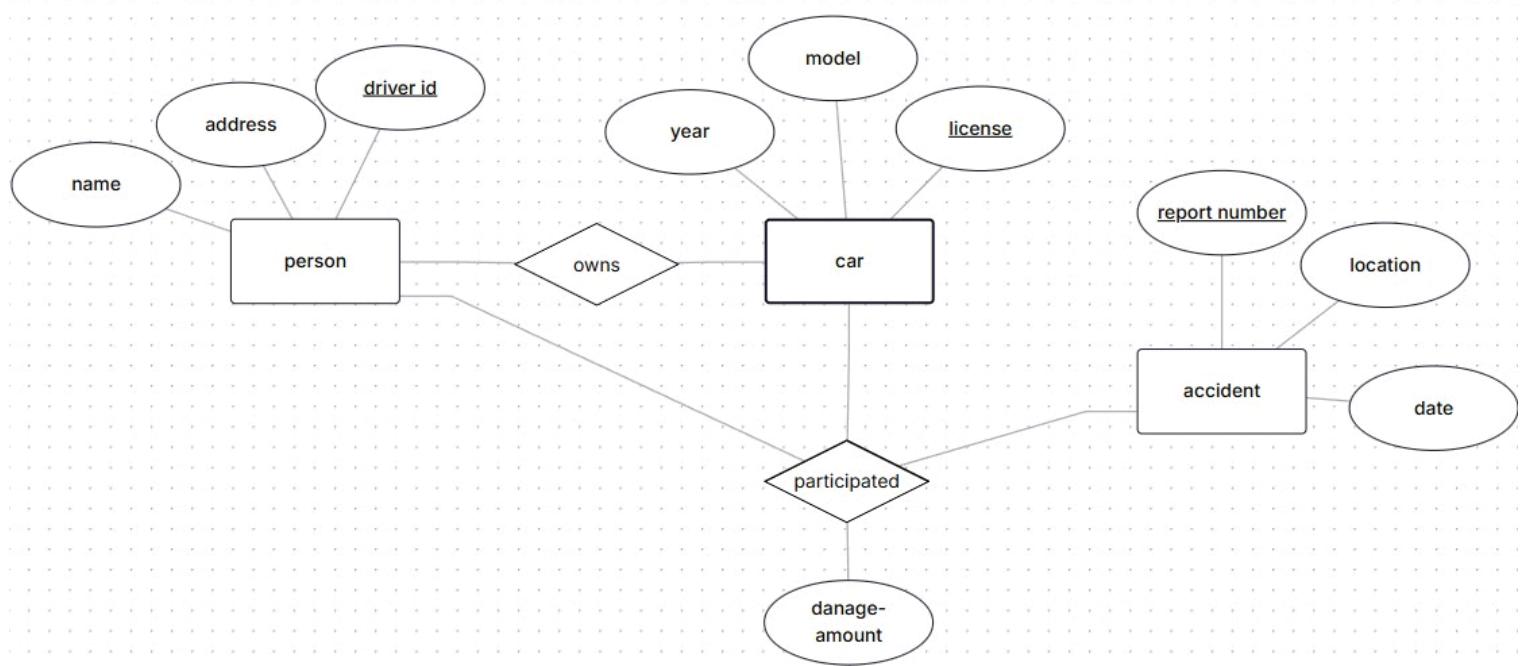


Case Study-03

Construct an ER diagram for a car insurance company whose customers own one or more cars. Each car has associated with a zero to any of recorded accidents

**Case study
group-09**

G.Harsha teja(2451-24-733-073)
G.Revanth(2451-24-733-076)
P.Hemanth(2451-24-733-309)
K.Siddharth reddy(2451-24-733-121)



```
mysql> use case_study;
Database changed
mysql> -- Table for person
mysql> CREATE TABLE person (
    ->     driver_id INT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     address VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql>
mysql> -- Table for car
mysql> CREATE TABLE car (
    ->     license VARCHAR(20) PRIMARY KEY,
    ->     model VARCHAR(50),
    ->     year INT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> -- Table for accident
mysql> CREATE TABLE accident (
    ->     report_number INT PRIMARY KEY,
    ->     date DATE,
    ->     location VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> -- Table for participated (relationship between person, car, and accident)
mysql> CREATE TABLE participated (
    ->     driver_id INT,
    ->     license VARCHAR(20),
    ->     report_number INT,
    ->     damage_amount DECIMAL(10,2),
    ->     PRIMARY KEY (driver_id, license, report_number),
    ->     FOREIGN KEY (driver_id) REFERENCES person(driver_id),
    ->     FOREIGN KEY (license) REFERENCES car(license),
    ->     FOREIGN KEY (report_number) REFERENCES accident(report_number)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

```

mysql> DESC PERSON;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| driver_id | int    | NO   | PRI | NULL    |
| name       | varchar(100) | YES  |     | NULL    |
| address    | varchar(255) | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.04 sec)

mysql> DESC CAR;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| license | varchar(20) | NO   | PRI | NULL    |
| model   | varchar(50)  | YES  |     | NULL    |
| year    | int    | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESC ACCIDENT;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| report_number | int    | NO   | PRI | NULL    |
| date         | date   | YES  |     | NULL    |
| location     | varchar(100) | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.03 sec)

mysql> DESC PARTICIPATED;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| driver_id | int    | NO   | PRI | NULL    |
| license   | varchar(20) | NO   | PRI | NULL    |
| report_number | int    | NO   | PRI | NULL    |
| damage_amount | decimal(10,2) | YES  |     | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

```

```

mysql> -- Insert into person table
mysql> INSERT INTO person (driver_id, name, address) VALUES
    -> (101, 'Amit Sharma', 'Delhi'),
    -> (102, 'Priya Mehta', 'Mumbai'),
    -> (103, 'Ravi Kumar', 'Hyderabad'),
    -> (104, 'Sneha Reddy', 'Chennai'),
    -> (105, 'Karan Singh', 'Bangalore');
Query OK, 5 rows affected (0.04 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert into car table
mysql> INSERT INTO car (license, model, year) VALUES
    -> ('MH12AB1234', 'Honda City', 2018),
    -> ('DL8CAF4321', 'Hyundai i20', 2020),
    -> ('TS09XY9876', 'Maruti Swift', 2017),
    -> ('KA03MN4567', 'Toyota Innova', 2019),
    -> ('TN10PQ6789', 'Ford EcoSport', 2021);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert into accident table
mysql> INSERT INTO accident (report_number, date, location) VALUES
    -> (201, '2023-01-15', 'Delhi'),
    -> (202, '2023-03-22', 'Mumbai'),
    -> (203, '2023-06-10', 'Hyderabad'),
    -> (204, '2023-08-05', 'Chennai'),
    -> (205, '2023-11-30', 'Bangalore');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql>
mysql> -- Insert into participated table
mysql> INSERT INTO participated (driver_id, license, report_number, damage_amount) VALUES
    -> (101, 'MH12AB1234', 201, 15000.00),
    -> (102, 'DL8CAF4321', 202, 8000.50),
    -> (103, 'TS09XY9876', 203, 12000.75),
    -> (104, 'KA03MN4567', 204, 5000.00),
    -> (105, 'TN10PQ6789', 205, 9500.25);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

```

```

mysql> SELECT * FROM CAR;
+-----+-----+-----+
| license | model | year |
+-----+-----+-----+
| DL8CAF4321 | Hyundai i20 | 2020 |
| KA03MN4567 | Toyota Innova | 2019 |
| MH12AB1234 | Honda City | 2018 |
| TN10PQ6789 | Ford EcoSport | 2021 |
| TS09XY9876 | Maruti Swift | 2017 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM PERSON;
+-----+-----+-----+
| driver_id | name | address |
+-----+-----+-----+
| 101 | Amit Sharma | Delhi |
| 102 | Priya Mehta | Mumbai |
| 103 | Ravi Kumar | Hyderabad |
| 104 | Sneha Reddy | Chennai |
| 105 | Karan Singh | Bangalore |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM ACCIDENT;
+-----+-----+-----+
| report_number | date | location |
+-----+-----+-----+
| 201 | 2023-01-15 | Delhi |
| 202 | 2023-03-22 | Mumbai |
| 203 | 2023-06-10 | Hyderabad |
| 204 | 2023-08-05 | Chennai |
| 205 | 2023-11-30 | Bangalore |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM PARTICIPATED;
+-----+-----+-----+-----+
| driver_id | license | report_number | damage_amount |
+-----+-----+-----+-----+
| 101 | MH12AB1234 | 201 | 15000.00 |
| 102 | DL8CAF4321 | 202 | 8000.50 |
| 103 | TS09XY9876 | 203 | 12000.75 |
| 104 | KA03MN4567 | 204 | 5000.00 |
| 105 | TN10PQ6789 | 205 | 9500.25 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> -- 1. List driver names with their cars involved in accidents
mysql> SELECT p.name, c.model, c.year
   > FROM person p
   > JOIN participated pa ON p.driver_id = pa.driver_id
   > JOIN car c ON pa.license = c.license;
+-----+-----+-----+
| name    | model  | year  |
+-----+-----+-----+
| Amit Sharma | Honda City | 2018 |
| Priya Mehta | Hyundai i20 | 2020 |
| Ravi Kumar | Maruti Swift | 2017 |
| Sneha Reddy | Toyota Innova | 2019 |
| Karan Singh | Ford EcoSport | 2021 |
+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> -- 2. Show accident details with driver and damage amount
mysql> SELECT a.report_number, a.date, a.location, p.name, pa.damage_amount
   > FROM accident a
   > JOIN participated pa ON a.report_number = pa.report_number
   > JOIN person p ON pa.driver_id = p.driver_id;
+-----+-----+-----+-----+-----+
| report_number | date      | location | name      | damage_amount |
+-----+-----+-----+-----+-----+
| 201 | 2023-01-15 | Delhi     | Amit Sharma | 15000.00 |
| 202 | 2023-03-22 | Mumbai    | Priya Mehta | 8000.50 |
| 203 | 2023-06-10 | Hyderabad | Ravi Kumar  | 12000.75 |
| 204 | 2023-08-05 | Chennai   | Sneha Reddy | 5000.00 |
| 205 | 2023-11-30 | Bangalore | Karan Singh | 9500.25 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> -- 3. Create a view showing driver and accident info
mysql> CREATE VIEW driver_accident_view AS
-> SELECT p.driver_id, p.name, a.report_number, a.date, a.location, pa.damage_amount
-> FROM person p
-> JOIN participated pa ON p.driver_id = pa.driver_id
-> JOIN accident a ON pa.report_number = a.report_number;
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> -- Query the view
mysql> SELECT * FROM driver_accident_view;
+-----+-----+-----+-----+-----+-----+
| driver_id | name      | report_number | date       | location   | damage_amount |
+-----+-----+-----+-----+-----+-----+
|    101 | Amit Sharma |        201 | 2023-01-15 | Delhi      |     15000.00 |
|    102 | Priya Mehta |        202 | 2023-03-22 | Mumbai     |      8000.50 |
|    103 | Ravi Kumar  |        203 | 2023-06-10 | Hyderabad |     12000.75 |
|    104 | Sneha Reddy |        204 | 2023-08-05 | Chennai    |      5000.00 |
|    105 | Karan Singh  |        205 | 2023-11-30 | Bangalore |     9500.25 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> -- 4. Create a view showing car and accident info
mysql> CREATE VIEW car_accident_view AS
-> SELECT c.license, c.model, c.year, a.report_number, a.date, pa.damage_amount
-> FROM car c
-> JOIN participated pa ON c.license = pa.license
-> JOIN accident a ON pa.report_number = a.report_number;
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> -- Query the view
mysql> SELECT * FROM car_accident_view;
+-----+-----+-----+-----+-----+-----+
| license | model      | year | report_number | date       | damage_amount |
+-----+-----+-----+-----+-----+-----+
| DL8CAF4321 | Hyundai i20 | 2020 |        202 | 2023-03-22 |     8000.50 |
| KA03MN4567 | Toyota Innova | 2019 |        204 | 2023-08-05 |     5000.00 |
| MH12AB1234 | Honda City   | 2018 |        201 | 2023-01-15 |    15000.00 |
| TN10PQ6789 | Ford EcoSport | 2021 |        205 | 2023-11-30 |     9500.25 |
| TS09XY9876 | Maruti Swift  | 2017 |        203 | 2023-06-10 |     12000.75 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

```

mysql> -- 5. Find drivers who had damage greater than average damage
mysql> SELECT name
   > FROM person
   > WHERE driver_id IN (
   >     SELECT driver_id
   >     FROM participated
   >     WHERE damage_amount > (SELECT AVG(damage_amount) FROM participated)
   > );
+-----+
| name      |
+-----+
| Amit Sharma |
| Ravi Kumar |
+-----+
2 rows in set (0.01 sec)

mysql>
mysql> -- 6. Find cars involved in accidents in the same location as accident #202
mysql> SELECT license, model
   > FROM car
   > WHERE license IN (
   >     SELECT license
   >     FROM participated
   >     WHERE report_number IN (
   >         SELECT report_number
   >         FROM accident
   >         WHERE location = (SELECT location FROM accident WHERE report_number = 202)
   >     )
   > );
+-----+-----+
| license | model    |
+-----+-----+
| DL8CAF4321 | Hyundai i20 |
+-----+-----+
1 row in set (0.00 sec)

```

```
mysql> -- 7. Count how many accidents each driver participated in
mysql> SELECT p.name, COUNT(pa.report_number) AS accident_count
   -> FROM person p
   -> JOIN participated pa ON p.driver_id = pa.driver_id
   -> GROUP BY p.name;
+-----+-----+
| name | accident_count |
+-----+-----+
| Amit Sharma | 1 |
| Priya Mehta | 1 |
| Ravi Kumar | 1 |
| Sneha Reddy | 1 |
| Karan Singh | 1 |
+-----+
5 rows in set (0.03 sec)

mysql>
mysql> -- 8. Find the car models that participated in accidents after 2023-06-01
mysql> SELECT DISTINCT c.model
   -> FROM car c
   -> JOIN participated pa ON c.license = pa.license
   -> JOIN accident a ON pa.report_number = a.report_number
   -> WHERE a.date > '2023-06-01';
+-----+
| model |
+-----+
| Maruti Swift |
| Toyota Innova |
| Ford EcoSport |
+-----+
3 rows in set (0.00 sec)
```

```
mysql>
mysql> -- 9. Show the maximum damage amount per accident
mysql> SELECT a.report_number, MAX(pa.damage_amount) AS max_damage
   -> FROM accident a
   -> JOIN participated pa ON a.report_number = pa.report_number
   -> GROUP BY a.report_number;
+-----+-----+
| report_number | max_damage |
+-----+-----+
|      201      |    15000.00 |
|      202      |     8000.50 |
|      203      |    12000.75 |
|      204      |     5000.00 |
|      205      |    9500.25 |
+-----+-----+
5 rows in set (0.00 sec)

mysql>
mysql> -- 10. List all drivers and their addresses who have cars manufactured after 2018
mysql> SELECT DISTINCT p.name, p.address
   -> FROM person p
   -> JOIN participated pa ON p.driver_id = pa.driver_id
   -> JOIN car c ON pa.license = c.license
   -> WHERE c.year > 2018;
+-----+-----+
| name    | address   |
+-----+-----+
| Priya Mehta | Mumbai |
| Sneha Reddy | Chennai |
| Karan Singh | Bangalore |
+-----+-----+
3 rows in set (0.00 sec)

mysql> |
```

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

Query 1 new_procedure - Routine get_total_damage - Routine new_function - Routine get_driver_accidents - Routine

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

SCHEMAS

- attendance_system
- case_study**
 - Tables
 - accident
 - car
 - participated
 - person
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - Functions
- college
- cse073
- harsha
- rohan
- school
- sys

Name: get_driver_accidents

DDL:

```

CREATE DEFINER='root'@'localhost' PROCEDURE `get_driver_accidents` (IN driverId INT)
BEGIN
  SELECT a.report_number, a.date, a.location, pa.damage_amount
  FROM accident a
  JOIN participated pa ON a.report_number = pa.report_number
  WHERE pa.driver_id = driverId;
END
  
```

Routine

#	Time	Action	Message	Duration / Fetch
4	2 18:27:00	Apply changes to _SYNTAX_ERROR	Applying object changes ...	
4	3 18:27:57	Apply changes to _SYNTAX_ERROR	Applying object changes ...	
4	4 18:28:51	Apply changes to get_total_damage	Changes applied	

Administration Schemas Information

Output

Schema: case_study

SQLEditor

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents

CHEMAS

Filter objects

case_study

- Tables
 - accident
 - car
 - participated
 - person
- Views
 - car_accident_view
 - driver_accident_view
- Stored Procedures
 - get_driver_accidents

Functions

college

cse073

harsha

rohan

school

sys

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

report_number	date	location	damage_amount
201	2023-01-15	Delhi	15000.00

Result 1

File Edit View Query Database Server Tools Scripting Help

Navigator: **case_study**

SCHEMAS

- Filter objects
- attendance_system
- case_study**
 - Tables
 - accident
 - car
 - participated
 - person
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - Functions
- college
- cse073
- harsha
- rohan
- school
- sys

Query 1 new_procedure - Routine get_total_damage - Routine new_function - Routine get_driver_accidents - Routine

Name: **get_total_damage**

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```

CREATE DEFINER='root'@'localhost' FUNCTION `get_total_damage`(driverId INT) RETURNS decimal(10,2)
  DETERMINISTIC
BEGIN
  DECLARE total DECIMAL(10,2);

  SELECT SUM(damage_amount)
  INTO total
  FROM participated
  WHERE driver_id = driverId;

  RETURN IFNULL(total, 0);
END
  
```

Routine

Administration Schemas

Information

Schema: **case_study**

Action Output

#	Time	Action	Message	Duration / Fetch
4	2 18:27:00	Apply changes to _SYNTAX_ERROR	Applying object changes ...	
4	3 18:27:57	Apply changes to _SYNTAX_ERROR	Applying object changes ...	
4	4 18:28:51	Apply changes to get_total_damage	Changes applied	
5	5 18:29:48	Apply changes to get_total_damage	No changes detected	
6	6 18:32:55	Apply changes to get_driver_accidents	Changes applied	

The screenshot shows a database management system interface with the following details:

- Navigator:** On the left, it displays the schema structure. Under the **case_study** schema, there are tables (accident, car, participated, person), views (car_accident_view, driver_accident_view), stored procedures (get_driver_accidents), and a function (get_total_damage). Other schemas listed include attendance_system, college, cse073, harsha, rohan, school, and sys.
- Query Editor:** The main area shows a query window titled "Query 1" with the following SQL code:

```
1 • select case_study.get_total_damage(101);  
2
```
- Result Grid:** Below the query editor, the results of the executed query are displayed in a grid:

case_study.get_total_damage(101)
15000.00
- Bottom Navigation:** At the bottom, there are tabs for "Administration", "Schemas", and "Result 1".

Schemas: attendance_system, case_study

Tables: accident, car, participated, person

Views: car_accident_view, driver_accident_view

Stored Procedures: get_driver_accidents, get_driver_damage, list_driver_damage

Functions: get_total_damage

Objects: college, cse073, harsha, rohan, school, sys

Procedure: list_driver_damage

```

CREATE DEFINER='root'@'localhost' PROCEDURE `list_driver_damage`()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE v_driverId INT;
    DECLARE v_name VARCHAR(100);
    DECLARE v_totalDamage DECIMAL(10,2);
    DECLARE cur CURSOR FOR
        SELECT p.driver_id, p.name
        FROM person p;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN cur;
    read_loop: LOOP
        FETCH cur INTO v_driverId, v_name;
        IF done = 1 THEN
            LEAVE read_loop;
        END IF;
        SELECT IFNULL(SUM(damage_amount),0)
        INTO v_totalDamage
        FROM participated
        WHERE driver_id = v_driverId;
        SELECT v_driverId AS DriverID, v_name AS DriverName, v_totalDamage AS TotalDamage;
    END LOOP;
    CLOSE cur;
END

```

Navigator

SCHEMAS

Filter objects

case_study

- Tables
 - accident
 - car
 - participated
 - person
- Views
 - car_accident_view
 - driver_accident_view
- Stored Procedures
 - get_driver_accidents
 - list_driver_damage
- Functions
 - f1 get_total_damage

attendance_system

cse073

harsha

rohan

school

sys

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage - Routine list_driver_damage

1 • call case_study.list_driver_damage();
2

Result Grid | Filter Rows: Export: Wrap Cell Content:

DriverID	DriverName	TotalDamage
105	Karan Singh	9500.25

Administration Schemas Result 1 Result 2 Result 3 Result 4 Result 5

Schemas

Filter objects

attendance_system

case_study

- Tables
 - accident
 - car
 - participated
 - person
- Views
 - car_accident_view
 - driver_accident_view
- Stored Procedures
 - get_driver_accidents
 - list_driver_damage
- Functions
 - f() get_total_damage

college

cse073

harsha

pavan

rohan

school

sys

Administration Schemas

Information

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage

1 • call case_study.list_driver_damage();
2

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content:

DriverID	DriverName	TotalDamage
101	Amit Sharma	15000.00

Result 1 Result 2 Result 3 Result 4 Result 5

File Edit View Query Database Server Tools Scripting Help

Navigator: Filter objects

Schemas

- attendance_system
- case_study
 - Tables
 - accident
 - car
 - participated
 - person
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - get_driver_accidents
 - list_driver_damage
 - Functions
 - f0_get_total_damage
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Administration Schemas

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage

1 • call case_study.list_driver_damage();
2

Result Grid | Filter Rows: Export: Wrap Cell Content: □

DriverID	DriverName	TotalDamage
102	Priya Mehta	8000.50

MANQMA X

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage

SCHEMAS

Filter objects

case_study

- Tables
 - accident
 - car
 - participated
 - person
- Views
 - car_accident_view
 - driver_accident_view
- Stored Procedures
 - get_driver_accidents
 - list_driver_damage
- Functions
 - f0 get_total_damage

Administration Schemas

Information Result 1 Result 2 Result 3 Result 4 Result 5

Query 1

```
1 •    call case_study.list_driver_damage();
2
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

DriverID	DriverName	TotalDamage
103	Ravi Kumar	12000.75

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- attendance_system
- case_study
 - Tables
 - accident
 - car
 - participated
 - person
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - get_driver_accidents
 - list_driver_damage
 - Functions
 - get_total_damage
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Administration Schemas Information

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage

1 • call case_study.list_driver_damage();
2

Result Grid Filter Rows: Export: Wrap Cell Content:

DriverID	DriverName	TotalDamage
104	Sneha Reddy	5000.00

Result 1 Result 2 Result 3 Result 4 Result 5

Result Grid Form Editor Field Types Read Only

The screenshot shows a database management interface with a toolbar at the top, a navigation pane on the left, and a main workspace on the right. The workspace contains a query editor with a single line of SQL code, a results grid displaying one row of data, and a toolbar with various icons for managing the result set.

Query Editor:

```
1 • call case_study.list_driver_damage();
2
```

Results Grid:

DriverID	DriverName	TotalDamage
104	Sneha Reddy	5000.00

Toolbar:

- Result Grid
- Form Editor
- Field Types
- Read Only

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- attendance_system
- case_study
 - Tables
 - accident
 - car
 - participated
 - person
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - get_driver_accidents
 - list_driver_damage
 - Functions
 - get_total_damage
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Administration Schemas

Information

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage

1 • call case_study.list_driver_damage();
2

Result Grid Filter Rows: Export: Wrap Cell Content:

DriverID	DriverName	TotalDamage
104	Sneha Reddy	5000.00

Result 1 Result 2 Result 3 Result 4 Result 5

Result Grid Form Editor Field Types Read Only

The screenshot shows a database management interface with a toolbar at the top, a navigation pane on the left, and a main workspace with a query editor and a results grid.

Toolbar: File, Edit, View, Query, Database, Server, Tools, Scripting, Help.

Navigation: Navigator, SCHEMAS, Administration, Schemas, Information.

Query Editor: Shows a single query in Query 1:

```
1 • call case_study.list_driver_damage();
2
```

Results Grid: Displays the output of the query:

DriverID	DriverName	TotalDamage
104	Sneha Reddy	5000.00

Status Bar: Result 1, Result 2, Result 3, Result 4, Result 5.

Right Sidebar: Result Grid, Form Editor, Field Types, Read Only.

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- attendance_system
- case_study
 - Tables
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - get_driver_accidents
 - list_driver_damage
 - Functions
 - get_total_damage
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage SQL File 10* SQL File 11*

```

1 DELIMITER $$

2

3 • CREATE TRIGGER after_damage_update
4 AFTER UPDATE ON participated
5 FOR EACH ROW
6 BEGIN
7   INSERT INTO accident_log (report_number, message)
8   VALUES (NEW.report_number, CONCAT('Damage updated to: ', NEW.damage_amount));
9 END $$

10

11 DELIMITER ;

```

Administration Schemas Information

No object selected

Action Output

#	Time	Action	Message	Duration / Fetch
2	23:43:08	CREATE TRIGGER after_accident_Insert AFTER INSERT ON accident FOR EACH ROW BEGIN INSERT ... 0 row(s) affected		0.016 sec
3	23:43:43	CREATE TRIGGER after_damage_update AFTER UPDATE ON participated FOR EACH ROW BEGIN INS... 0 row(s) affected		0.016 sec
4	23:44:15	UPDATE participated SET damage_amount = 20000 WHERE driver_id = 101 AND report_number = 201	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.032 sec
5	23:44:15	SELECT * FROM accident_log LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- attendance_system
- case_study**
 - Tables
 - Views
 - car_accident_view
 - driver_accident_view
 - Stored Procedures
 - get_driver_accidents
 - list_driver_damage
 - Functions
 - get_total_damage
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Query 1 | get_driver_accidents | get_driver_accidents | get_driver_accidents | get_total_damage | list_driver_damage | get_driver_accidents | list_driver_damage | SQL File 10* | SQL File 11*

```

1 -- First create a log table to store trigger messages
2 • CREATE TABLE accident_log (
3   log_id INT AUTO_INCREMENT PRIMARY KEY,
4   report_number INT,
5   message VARCHAR(255),
6   log_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
7 );
8
9 DELIMITER $$

10
11 • CREATE TRIGGER after_accident_insert
12   AFTER INSERT ON accident
13   FOR EACH ROW
14   BEGIN
15     INSERT INTO accident_log (report_number, message)
16     VALUES (NEW.report_number, CONCAT('Accident added at location: ', NEW.location));
17   END $$

18
19 DELIMITER ;

```

Administration Schemas Information

No object selected

Output

Action	Time	Action	Message	Duration / Fetch
2 23:43:08	CREATE TRIGGER after_accident_insert AFTER INSERT ON accident FOR EACH ROW BEGIN	INSERT ...	0 row(s) affected	0.016 sec
3 23:43:43	CREATE TRIGGER after_damage_update AFTER UPDATE ON participated FOR EACH ROW BEGIN	INS...	0 row(s) affected	0.016 sec
4 23:44:15	UPDATE participated SET damage_amount = 20000 WHERE driver_id = 101 AND report_number = 201		1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.032 sec
5 23:44:15	SELECT * FROM accident_log LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- attendance_system
- case_study**
- Tables
- Views
- Stored Procedures
- Functions
- college
- cse073
- harsha
- pavan
- rohan
- school
- sys

Query 1 get_driver_accidents get_driver_accidents get_driver_accidents get_total_damage list_driver_damage get_driver_accidents list_driver_damage SQL File 10* SQL File 11*

```

1 • UPDATE participated
2   SET damage_amount = 20000
3   WHERE driver_id = 101 AND report_number = 201;
4
5 • SELECT * FROM accident_log;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Contents: |

log_id	report_number	message	log_time
1	201	Damage updated to: 20000.00	2025-12-09 23:44:15
*	NULL	NULL	NULL

Administration Schemas

Information

No object selected

accident_log 1 x

Action Output

#	Time	Action	Message	Duration / Fetch
2	23:43:08	CREATE TRIGGER after_accident_insert AFTER INSERT ON accident FOR EACH ROW BEGIN INSERT ...	0 row(s) affected	0.016 sec
3	23:43:43	CREATE TRIGGER after_damage_update AFTER UPDATE ON participated FOR EACH ROW BEGIN INS...	0 row(s) affected	0.016 sec
4	23:44:15	UPDATE participated SET damage_amount = 20000 WHERE driver_id = 101 AND report_number = 201	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.032 sec
5	23:44:15	SELECT * FROM accident_log LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec