CS 3710 : Database Systems Lab

Car Dealer Company

Team Members

Libin N. George	111501015
Sooraj Tom	111501036

Table of contents

Introduction

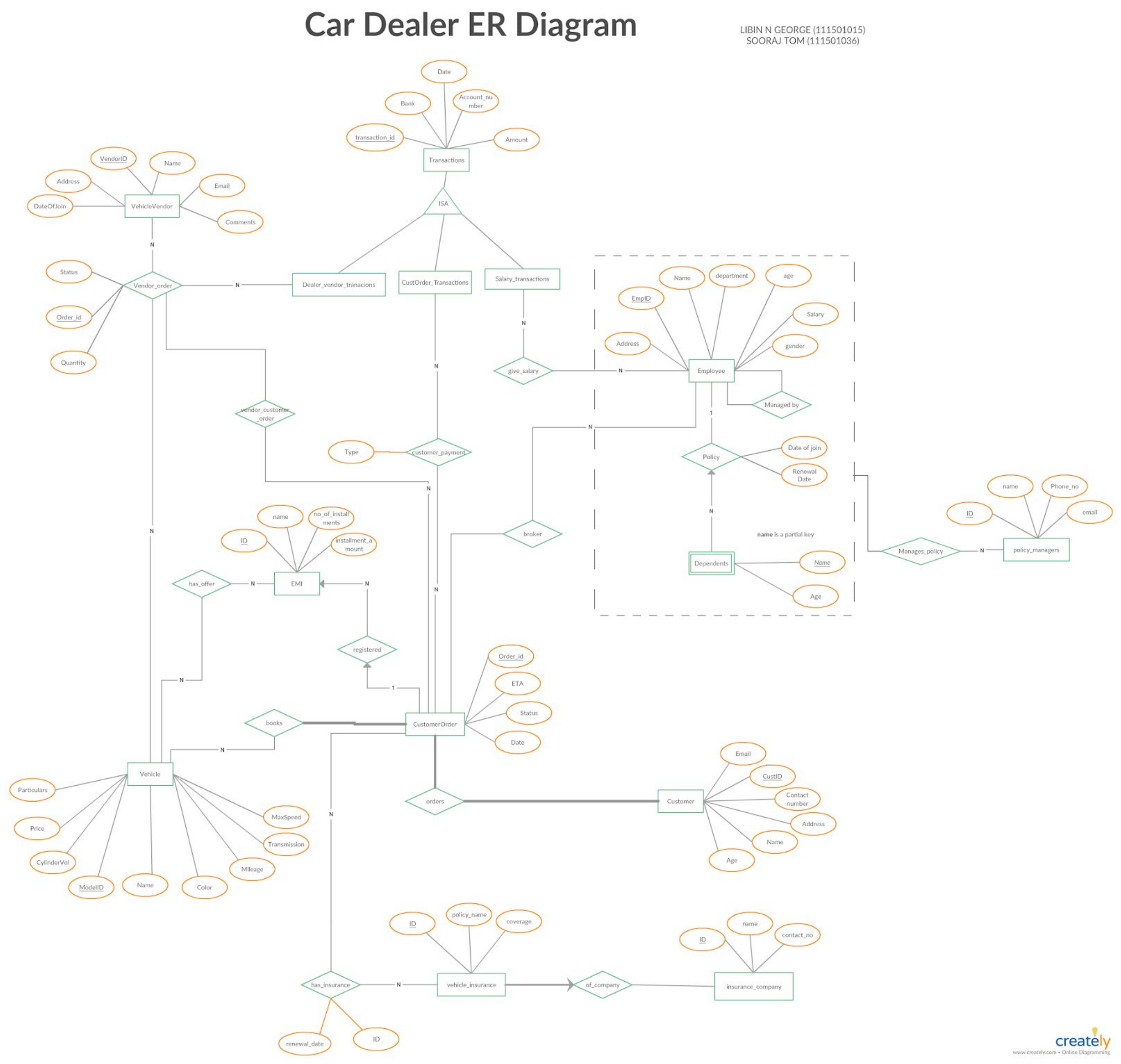
In this project, we designed and implemented a database for a car dealer company. The database boasts of many features such as an array of triggers, stored functions and procedures, in-built validation of data entries, etc. The database is capable of storing details about vehicles, their corresponding vendors, customers, their orders, employees along with their insurance details and all financial transactions made. The database also keeps track of an order from its generation until it is delivered. It also maintains details about EMI schemes if it is availed by the customer. The project is available at https://github.com/soorajtom/car_dealer_database.

Car Dealer Database Requirements

1. Should be able to make orders for customers.
2. Customers can avail special financial assists such as EMI payment options.
3. Customers can also choose to register for insurance policy for the vehicle.
4. Registering employees into different sections like broker, accountant, public relations, axillary staff, supervisor, admin etc.
5. Vehicles are provided by different companies.
6. Should be able to register verified Vehicle vendors.
7. Should be able to give vehicle order to vehicle vendors.
8. Database should be able to manage the transactions to vehicle vendors.
9. Salary transactions for each employee should be maintained. Each salary transaction should stored.
10. All Employees should take one policy.
11. Each policy taken by an employee have list of dependents.
12. Each policy taken by an employee is managed by a policy manager.

Design

Entity Relationship Diagram



RAIDS

We plans to implement RAID 1 which uses mirroring and 100 % redundancy. RAID 1 is enough as we are dealing with a car dealer company in which disk failure can be replaced easily.

Contributions

Libin N. George

- Requirement Analysis
- Design of ER diagram
- Design of schema
- Design of triggers
- Design of functions and procedures

Sooraj Tom

- Requirement Analysis
- Design of ER diagram
- Design of schema
- Design of functions and procedures
- Database seeding and testing
- Design of front end

Description of Implementation

Triggers

1. **email_validate_customer** : validates the email id for customer
2. **phone_number_validate_customer**: validates phone number and formats it to the international format.
3. **phone_number_validate_insurance_company** : validates phone number and formats it to the international format.
4. **email_validate_vehicle_vendor** : validates the email id for a vendor.
5. **email_validate_policy_manager** : validates email id for policy managers.
6. **phone_number_validate_policy_manager** : validates phone number and formats it to the international format.
7. **has_offer_validations** : validates that the product of installment amount and number of installments are consistent with the vehicle price.
8. **validate_age** : verifies that the customer is at least 18 years old.
9. **emi_register_check** : checks if the EMI request for a vehicle is allowed.
10. **emi_avalibility_check** : checks if an EMI offer is available for the vehicle when the transaction is made under emi mode.
11. **change_order_status** : Changes the status to READY in the customer_order when the vehicle arrives the showroom, ie the vendor delivered the vehicle. (ON UPDATE)

12. **check _for_normal_payment** : Check if vehicle has arrived or not before payment (normal or emi).
13. **change_order_status2** : Changes the status to READY in the customer_order when the vehicle arrives the showroom, ie the vendor delivered the vehicle. (ON INSERT)
14. **change_order_status1** : Changes the status to IN_TRANSIT in the customer_order when order is given to vehicle vendor.
15. **change_order_status3** : Changes the status to DELIVERED when customer pays (emi or normal).

Functions and Procedures

1. **FUNCTION check_phone_num** : Runs a regular expression check to validate phone number and converts it to international format
2. **PROCEDURE check_email** : Runs a regular expression check to validate email address
3. **PROCEDURE insert_customer_payment** : Inserts a payment made by a customer into the transaction table and links it to customer payment table. If it an EMI payment, the order is verified with the emi registration table.
4. **PROCEDURE insert_salary_payment** : Inserts a salary transaction, and links it to the give_salary table. An error is raised if the salary paid is less than the base salary.
5. **PROCEDURE salary_summary** : Generates a summary of all salary paid to a given employee.
6. **PROCEDURE pending_emi_payments** : Lists out customers who are enrolled for EMI payment option but hadn't paid in the last month.
7. **FUNCTION inc_by_percent** : Increments the given parameter by a given percentage.
8. **PROCEDURE salary_increment** : Increment the salary of given employee by a given percentage.
9. **PROCEDURE salary_increment_all** : Increments the salaries of all employees by a given percentage.
10. **PROCEDURE salary_decrement** : Decrements the salary of a given employee by a given percentage.
11. **FUNCTION get_emi** : given a order_id gets emi_name if available

12. **PROCEDURE create_user** : given username, role, and password creates user and gives role.
13. **PROCEDURE INITEMPLOYEE** : creates users for employee with username as password. Creates user only if user does not exist already.
14. **PROCEDURE INIT_INSURANCE_AGENT**: creates users for insureance_company with appropriate privilege. Creates user only if user does not exist already.
15. **PROCEDURE INIT_VEHICLE_DEALER** : creates users for vehicle_vendor with appropriate privileges. Creates user only if user does not exist already.

Views

1. **customer_order_view** (order id, customer name, ETA, status, vehicle id, vehicle name, order_date)
2. **customer_payment_view** (customer_name, transaction_id, emi_name, bank, account_number, payment_date, amount)
3. **employee_view** (id, name, address, salary, gender, age, department, manager)
4. **salary_payment_view** (transaction_id, employee_name, bank, account_number, payment_date, amount)
5. **vehicles_not_delivered** (shows orders which are not yet delivered)
6. **vehicle_emi_view** (combined view for vehicle and emi showing possible emi for each vehicle)

Transactions and Concurrency

Transactions are used in two procedures insert_customer_payment**,** insert_salary_payment. These procedures have to insert to multiple tables and partial insertion to some tables can cause inconsistency in the database. These procedures can be used concurrently by different users as it is implemented as transactions. The procedures mainly insert data in corresponding tables for some billing or money transactions. Isolation level used in the transactions are Repeatable-Read (innodb default). The deadlock detection and recovery is also enabled so that database system can detect and recover from some accidental deadlock.

Command to backup the database:

```
mysqldump --databases car_dealer -u root -p --result-file=backup.sql
```

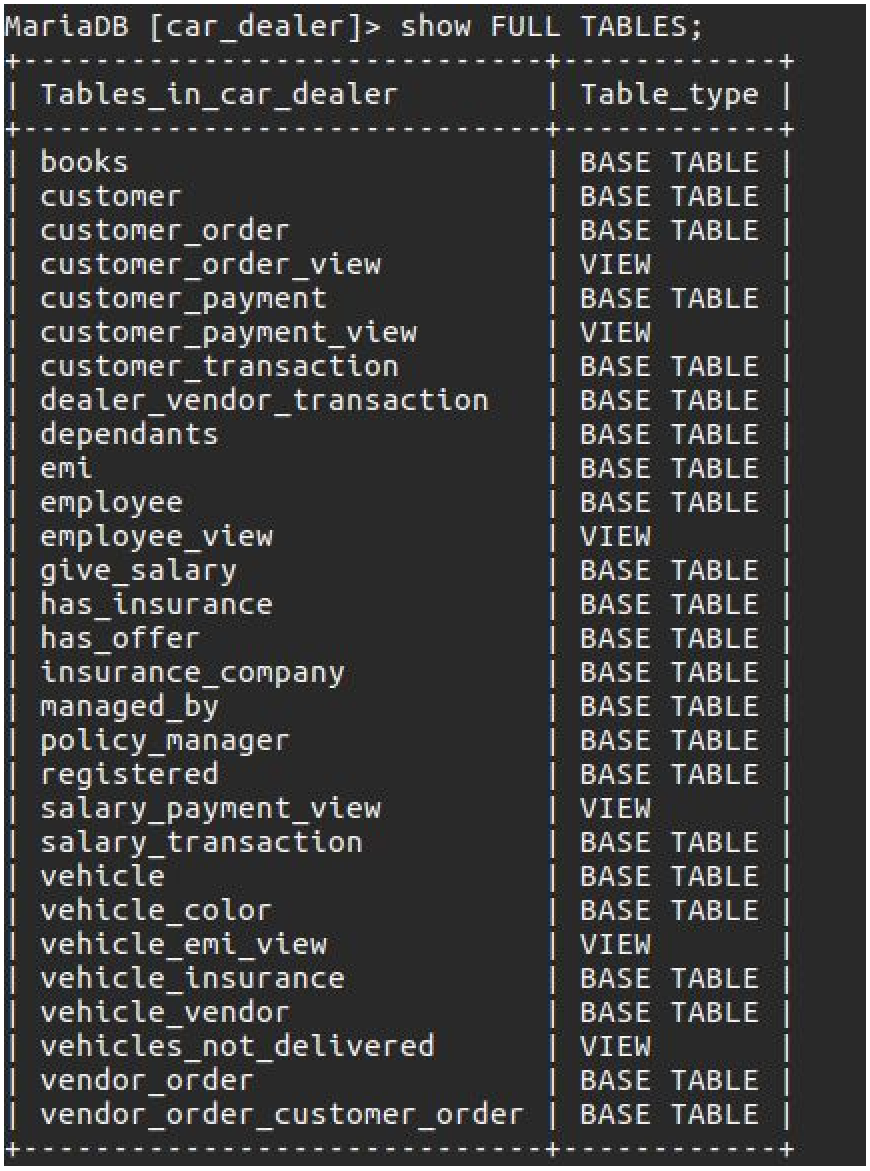
Roles

Role	Permission	Tables
DBA	ALL	%
Manager	SELECT, INSERT, UPDATE, DELETE	%
Accountant	SELECT, INSERT, UPDATE, DELETE	%transaction, give_salary, customer_payment, emi, registered
Broker	SELECT, INSERT, UPDATE	customer, customer_order, has_offer, has_insurance,vehicle, books, registered
	SELECT	insurance_company, vehicle_insurance, emi, customer_payment, customer_transaction
Insurance_agent	SELECT, INSERT, UPDATE, DELETE	insurance_company, vehicle_insurance, has_insurance
	SELECT	vehicle, customer, customer_order, books
Vendor_company _dealer	SELECT, INSERT, UPDATE, DELETE	vehicle, vehicle_vendor, vehicle_order
	SELECT	dealer_vendor_transaction

Improvements after testing phase

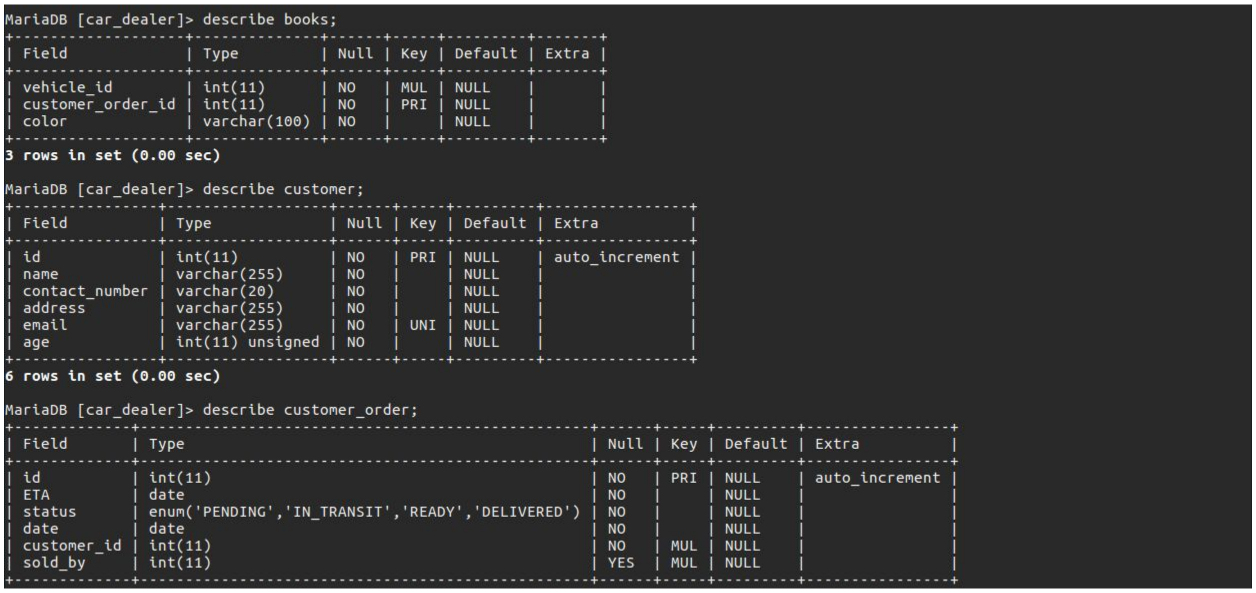
1. Restricting customers below 18 years old
2. Creating separate roles for vendor dealer to see vehicle order and vehicle details.
3. Changing status for customer order on creating vendor order, updating status in vendor order and on customer payment.
4. Validation on availing emi on vehicles.

Schema and Implementation

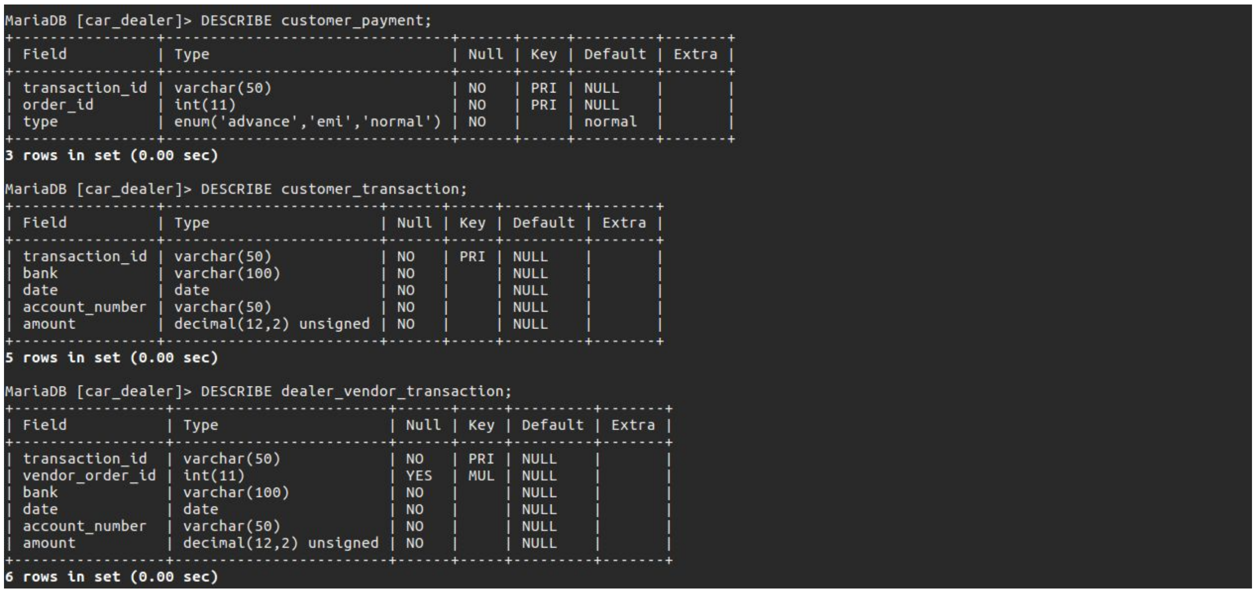


```
MariaDB [car_dealer]> show FULL TABLES;
+-----------------------------+------------+
| Tables_in_car_dealer        | Table_type |
+-----------------------------+------------+
| books                       | BASE TABLE |
| customer                    | BASE TABLE |
| customer_order              | BASE TABLE |
| customer_order_view         | VIEW       |
| customer_payment            | BASE TABLE |
| customer_payment_view       | VIEW       |
| customer_transaction        | BASE TABLE |
| dealer_vendor_transaction   | BASE TABLE |
| dependants                  | BASE TABLE |
| emi                         | BASE TABLE |
| employee                    | BASE TABLE |
| employee_view               | VIEW       |
| give_salary                 | BASE TABLE |
| has_insurance               | BASE TABLE |
| has_offer                   | BASE TABLE |
| insurance_company           | BASE TABLE |
| managed_by                  | BASE TABLE |
| policy_manager              | BASE TABLE |
| registered                  | BASE TABLE |
| salary_payment_view         | VIEW       |
| salary_transaction          | BASE TABLE |
| vehicle                     | BASE TABLE |
| vehicle_color               | BASE TABLE |
| vehicle_emi_view            | VIEW       |
| vehicle_insurance           | BASE TABLE |
| vehicle_vendor              | BASE TABLE |
| vehicles_not_delivered      | VIEW       |
| vendor_order                | BASE TABLE |
| vendor_order_customer_order | BASE TABLE |
+-----------------------------+------------+
```

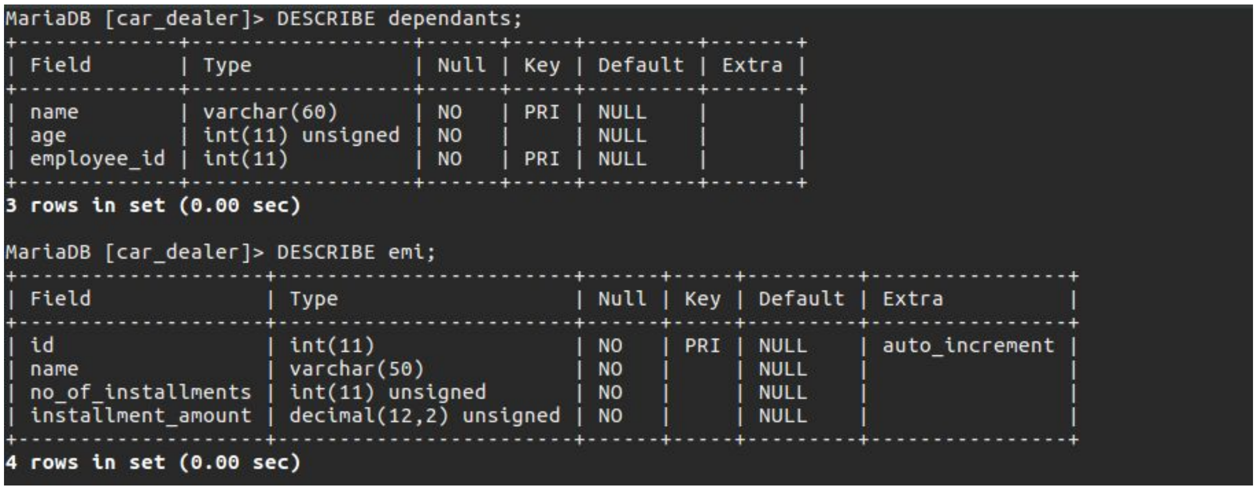
Schema of tables



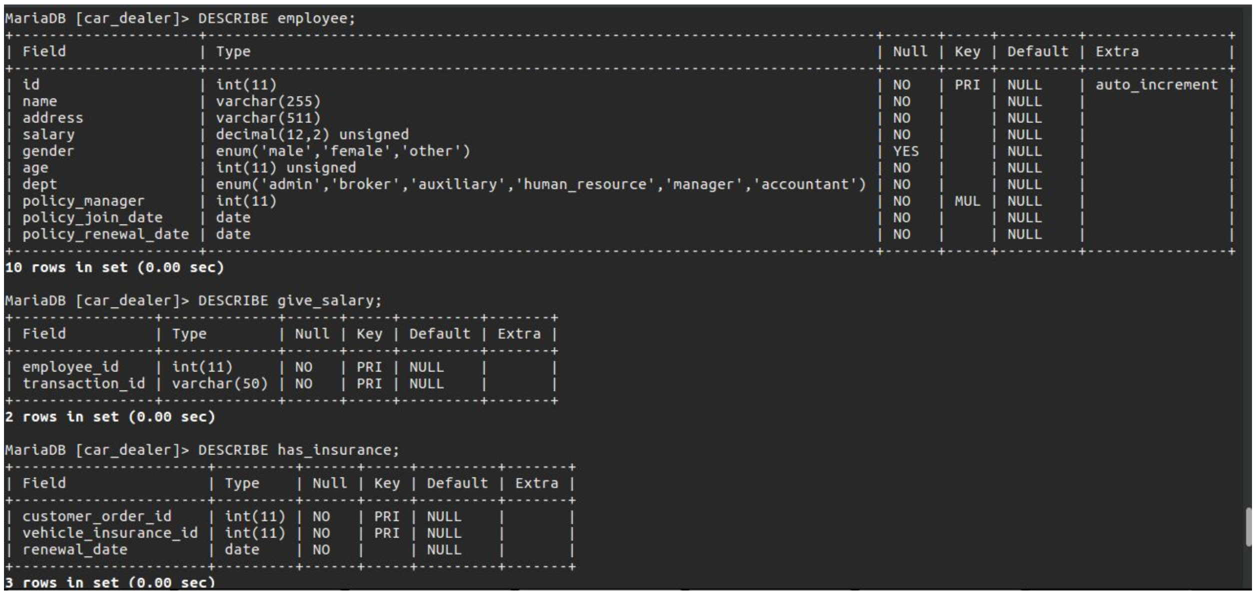
```
MariaDB [car_dealer]> describe books;
+-------------------+--------------+------+-----+---------+-------+
| Field             | Type         | Null | Key | Default | Extra |
+-------------------+--------------+------+-----+---------+-------+
| vehicle_id        | int(11)      | NO   | MUL | NULL    |       |
| customer_order_id | int(11)      | NO   | PRI | NULL    |       |
| color             | varchar(100) | NO   |     | NULL    |       |
+-------------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

MariaDB [car_dealer]> describe customer;
+----------------+------------------+------+-----+---------+----------------+
| Field          | Type             | Null | Key | Default | Extra          |
+----------------+------------------+------+-----+---------+----------------+
| id             | int(11)          | NO   | PRI | NULL    | auto_increment |
| name           | varchar(255)     | NO   |     | NULL    |                |
| contact_number | varchar(20)      | NO   |     | NULL    |                |
| address        | varchar(255)     | NO   |     | NULL    |                |
| email          | varchar(255)     | NO   | UNI | NULL    |                |
| age            | int(11) unsigned | NO   |     | NULL    |                |
+----------------+------------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

MariaDB [car_dealer]> describe customer_order;
+-------------+------------------------------------------------+------+-----+---------+----------------+
| Field       | Type                                           | Null | Key | Default | Extra          |
+-------------+------------------------------------------------+------+-----+---------+----------------+
| id          | int(11)                                        | NO   | PRI | NULL    | auto_increment |
| ETA         | date                                           | NO   |     | NULL    |                |
| status      | enum('PENDING','IN_TRANSIT','READY','DELIVERED') | NO   |     | NULL    |                |
| date        | date                                           | NO   |     | NULL    |                |
| customer_id | int(11)                                        | NO   | MUL | NULL    |                |
| sold_by     | int(11)                                        | YES  | MUL | NULL    |                |
+-------------+------------------------------------------------+------+-----+---------+----------------+
```



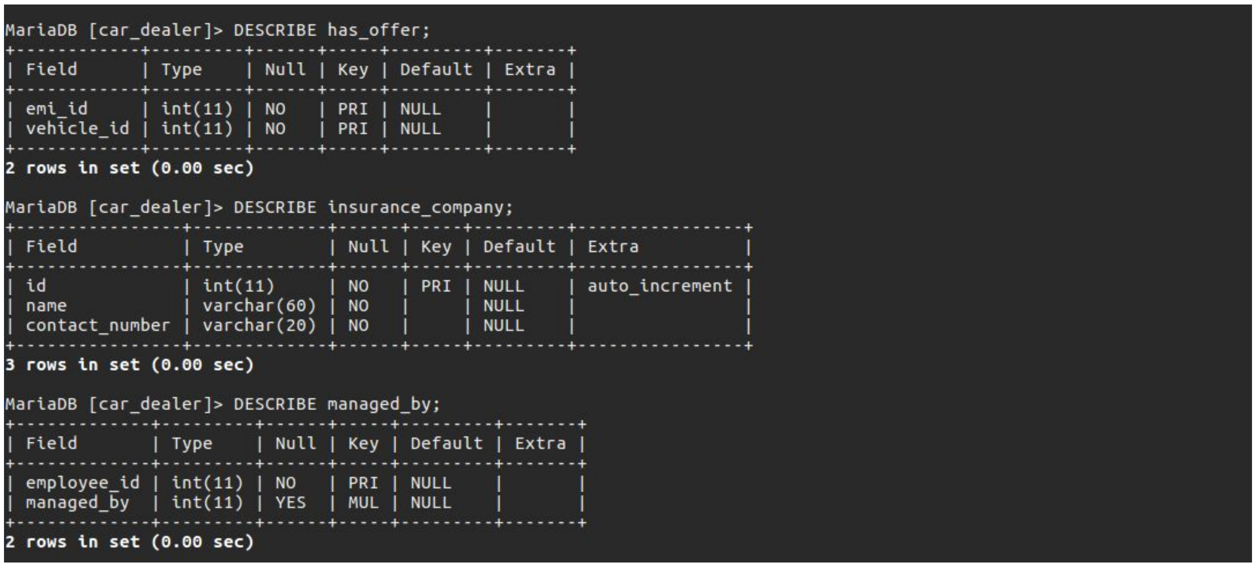
```
MariaDB [car_dealer]> DESCRIBE customer_payment;
+----------------+--------------------------------+------+-----+---------+-------+
| Field          | Type                           | Null | Key | Default | Extra |
+----------------+--------------------------------+------+-----+---------+-------+
| transaction_id | varchar(50)                    | NO   | PRI | NULL    |       |
| order_id       | int(11)                        | NO   | PRI | NULL    |       |
| type           | enum('advance','emi','normal') | NO   |     | normal  |       |
+----------------+--------------------------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE customer_transaction;
+----------------+------------------------+------+-----+---------+-------+
| Field          | Type                   | Null | Key | Default | Extra |
+----------------+------------------------+------+-----+---------+-------+
| transaction_id | varchar(50)            | NO   | PRI | NULL    |       |
| bank           | varchar(100)           | NO   |     | NULL    |       |
| date           | date                   | NO   |     | NULL    |       |
| account_number | varchar(50)            | NO   |     | NULL    |       |
| amount         | decimal(12,2) unsigned | NO   |     | NULL    |       |
+----------------+------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE dealer_vendor_transaction;
+-----------------+------------------------+------+-----+---------+-------+
| Field           | Type                   | Null | Key | Default | Extra |
+-----------------+------------------------+------+-----+---------+-------+
| transaction_id  | varchar(50)            | NO   | PRI | NULL    |       |
| vendor_order_id | int(11)                | YES  | MUL | NULL    |       |
| bank            | varchar(100)           | NO   |     | NULL    |       |
| date            | date                   | NO   |     | NULL    |       |
| account_number  | varchar(50)            | NO   |     | NULL    |       |
| amount          | decimal(12,2) unsigned | NO   |     | NULL    |       |
+-----------------+------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



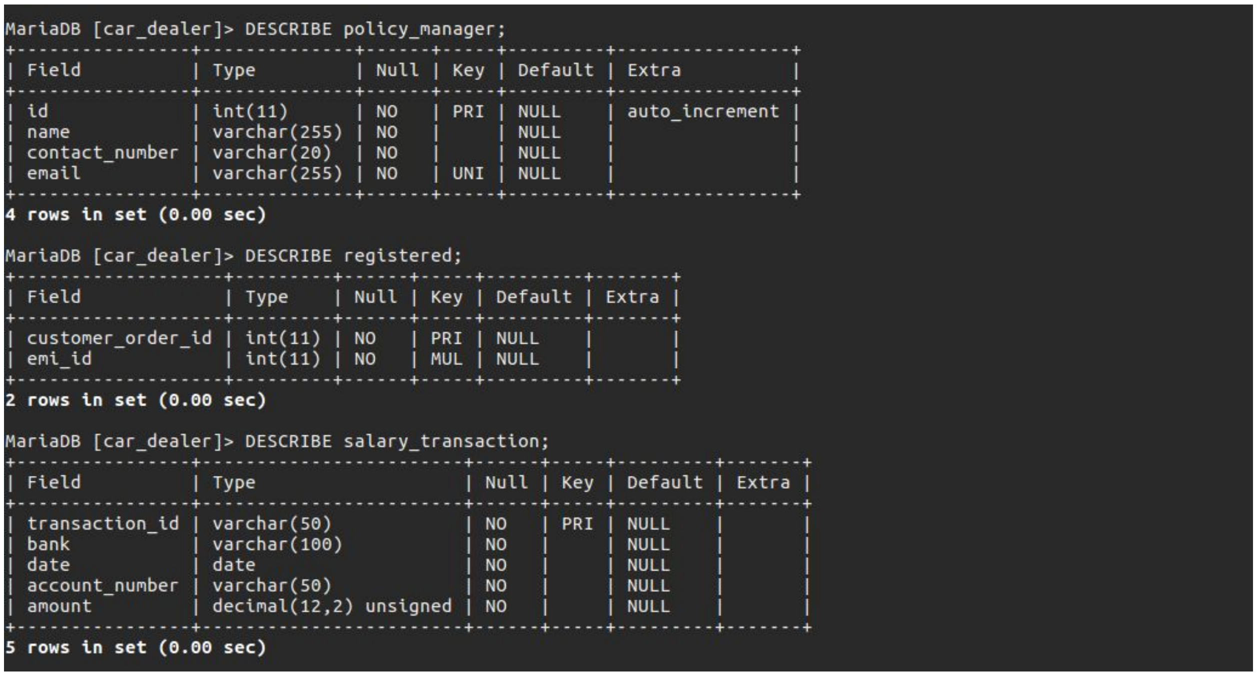
```
MariaDB [car_dealer]> DESCRIBE dependants;
+-------------+------------------+------+-----+---------+-------+
| Field       | Type             | Null | Key | Default | Extra |
+-------------+------------------+------+-----+---------+-------+
| name        | varchar(60)      | NO   | PRI | NULL    |       |
| age         | int(11) unsigned | NO   |     | NULL    |       |
| employee_id | int(11)          | NO   | PRI | NULL    |       |
+-------------+------------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE emi;
+--------------------+------------------------+------+-----+---------+----------------+
| Field              | Type                   | Null | Key | Default | Extra          |
+--------------------+------------------------+------+-----+---------+----------------+
| id                 | int(11)                | NO   | PRI | NULL    | auto_increment |
| name               | varchar(50)            | NO   |     | NULL    |                |
| no_of_installments | int(11) unsigned       | NO   |     | NULL    |                |
| installment_amount | decimal(12,2) unsigned | NO   |     | NULL    |                |
+--------------------+------------------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```



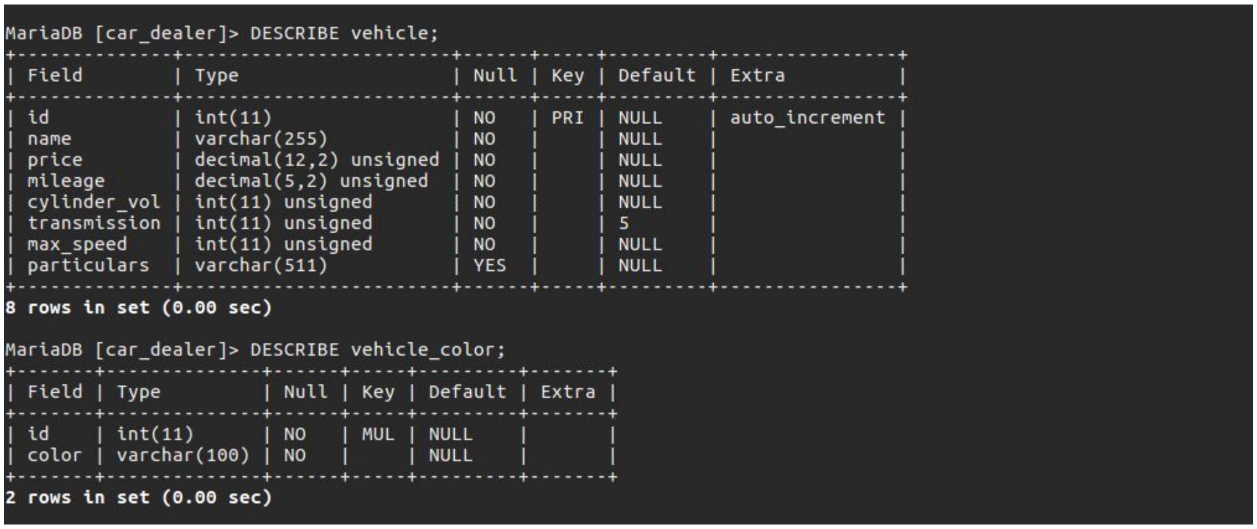
```
MariaDB [car_dealer]> DESCRIBE employee;
+---------------------+----------------------------------------------------------------------------+------+-----+---------+----------------+
| Field               | Type                                                                       | Null | Key | Default | Extra          |
+---------------------+----------------------------------------------------------------------------+------+-----+---------+----------------+
| id                  | int(11)                                                                    | NO   | PRI | NULL    | auto_increment |
| name                | varchar(255)                                                               | NO   |     | NULL    |                |
| address             | varchar(511)                                                               | NO   |     | NULL    |                |
| salary              | decimal(12,2) unsigned                                                     | NO   |     | NULL    |                |
| gender              | enum('male','female','other')                                              | YES  |     | NULL    |                |
| age                 | int(11) unsigned                                                           | NO   |     | NULL    |                |
| dept                | enum('admin','broker','auxiliary','human_resource','manager','accountant') | NO   |     | NULL    |                |
| policy_manager      | int(11)                                                                    | NO   | MUL | NULL    |                |
| policy_join_date    | date                                                                       | NO   |     | NULL    |                |
| policy_renewal_date | date                                                                       | NO   |     | NULL    |                |
+---------------------+----------------------------------------------------------------------------+------+-----+---------+----------------+
10 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE give_salary;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| employee_id    | int(11)     | NO   | PRI | NULL    |       |
| transaction_id | varchar(50) | NO   | PRI | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE has_insurance;
+----------------------+---------+------+-----+---------+-------+
| Field                | Type    | Null | Key | Default | Extra |
+----------------------+---------+------+-----+---------+-------+
| customer_order_id    | int(11) | NO   | PRI | NULL    |       |
| vehicle_insurance_id | int(11) | NO   | PRI | NULL    |       |
| renewal_date         | date    | NO   |     | NULL    |       |
+----------------------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```



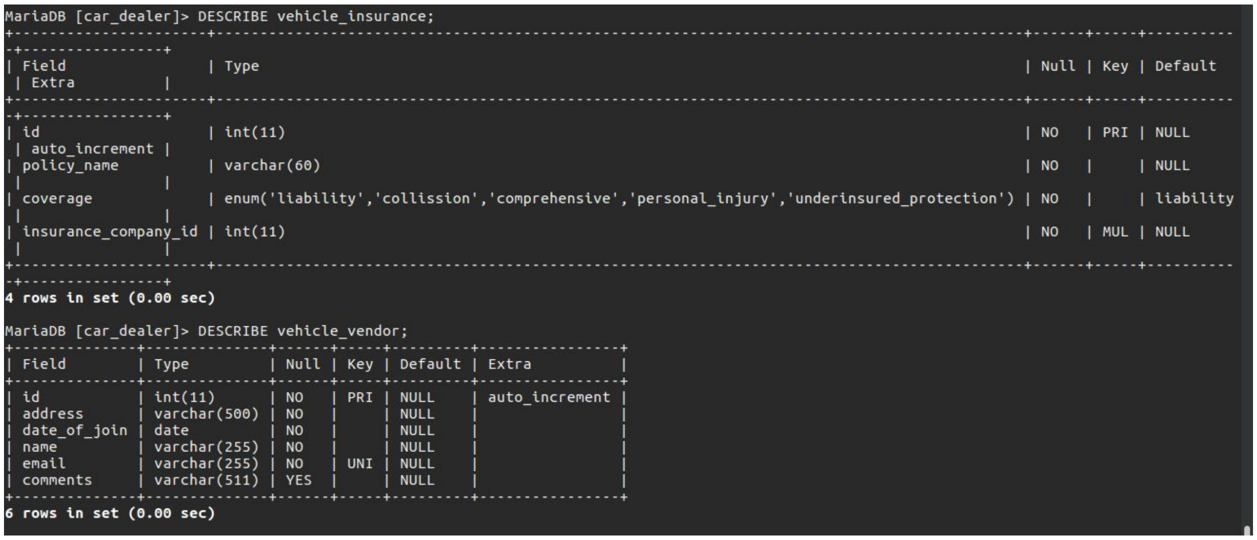
```
MariaDB [car_dealer]> DESCRIBE has_offer;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| emi_id     | int(11) | NO   | PRI | NULL    |       |
| vehicle_id | int(11) | NO   | PRI | NULL    |       |
+------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE insurance_company;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| id             | int(11)     | NO   | PRI | NULL    | auto_increment |
| name           | varchar(60) | NO   |     | NULL    |                |
| contact_number | varchar(20) | NO   |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE managed_by;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| employee_id | int(11) | NO   | PRI | NULL    |       |
| managed_by  | int(11) | YES  | MUL | NULL    |       |
+-------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



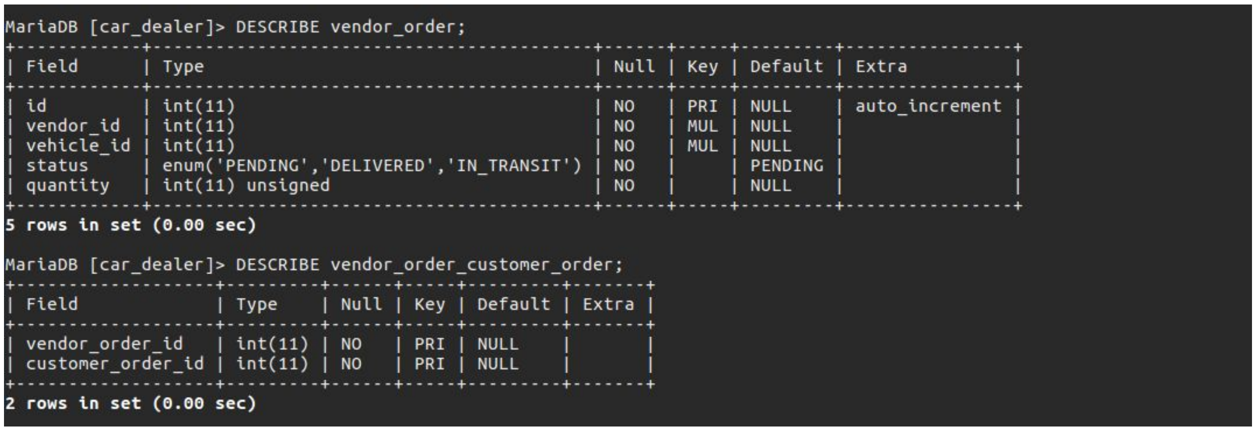
```
MariaDB [car_dealer]> DESCRIBE policy_manager;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id             | int(11)      | NO   | PRI | NULL    | auto_increment |
| name           | varchar(255) | NO   |     | NULL    |                |
| contact_number | varchar(20)  | NO   |     | NULL    |                |
| email          | varchar(255) | NO   | UNI | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE registered;
+-------------------+---------+------+-----+---------+-------+
| Field             | Type    | Null | Key | Default | Extra |
+-------------------+---------+------+-----+---------+-------+
| customer_order_id | int(11) | NO   | PRI | NULL    |       |
| emi_id            | int(11) | NO   | MUL | NULL    |       |
+-------------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE salary_transaction;
+----------------+------------------------+------+-----+---------+-------+
| Field          | Type                   | Null | Key | Default | Extra |
+----------------+------------------------+------+-----+---------+-------+
| transaction_id | varchar(50)            | NO   | PRI | NULL    |       |
| bank           | varchar(100)           | NO   |     | NULL    |       |
| date           | date                   | NO   |     | NULL    |       |
| account_number | varchar(50)            | NO   |     | NULL    |       |
| amount         | decimal(12,2) unsigned | NO   |     | NULL    |       |
+----------------+------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```



```
MariaDB [car_dealer]> DESCRIBE vehicle;
+--------------+------------------------+------+-----+---------+----------------+
| Field        | Type                   | Null | Key | Default | Extra          |
+--------------+------------------------+------+-----+---------+----------------+
| id           | int(11)                | NO   | PRI | NULL    | auto_increment |
| name         | varchar(255)           | NO   |     | NULL    |                |
| price        | decimal(12,2) unsigned | NO   |     | NULL    |                |
| mileage      | decimal(5,2) unsigned  | NO   |     | NULL    |                |
| cylinder_vol | int(11) unsigned       | NO   |     | NULL    |                |
| transmission | int(11) unsigned       | NO   |     | 5       |                |
| max_speed    | int(11) unsigned       | NO   |     | NULL    |                |
| particulars  | varchar(511)           | YES  |     | NULL    |                |
+--------------+------------------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE vehicle_color;
+-------+--------------+------+-----+---------+-------+
| Field | Type         | Null | Key | Default | Extra |
+-------+--------------+------+-----+---------+-------+
| id    | int(11)      | NO   | MUL | NULL    |       |
| color | varchar(100) | NO   |     | NULL    |       |
+-------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

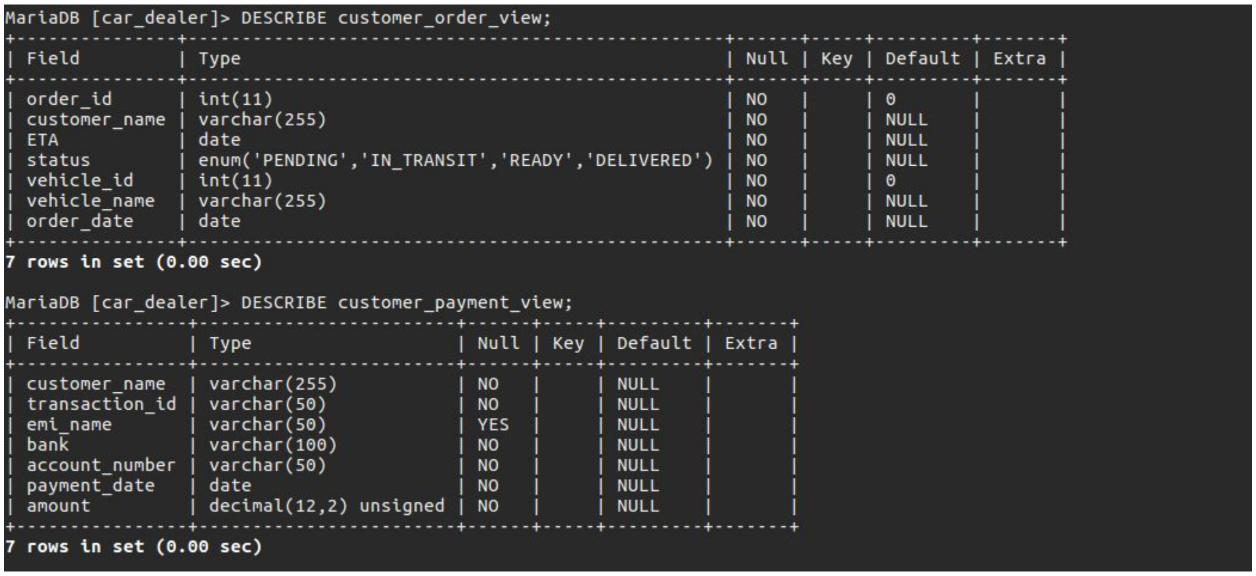


```
MariaDB [car_dealer]> DESCRIBE vehicle_insurance;
+----------------------+------------------------------------------------------------------------------------------+------+-----+---------
-+----------------+
| Field                | Type                                                                                     | Null | Key | Default
 | Extra          |
+----------------------+------------------------------------------------------------------------------------------+------+-----+---------
-+----------------+
| id                   | int(11)                                                                                  | NO   | PRI | NULL
 | auto_increment |
| policy_name          | varchar(60)                                                                              | NO   |     | NULL
 |                |
| coverage             | enum('liability','collission','comprehensive','personal_injury','underinsured_protection') | NO   |     | liability
 |                |
| insurance_company_id | int(11)                                                                                  | NO   | MUL | NULL
 |                |
+----------------------+------------------------------------------------------------------------------------------+------+-----+---------
-+----------------+
4 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE vehicle_vendor;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| address      | varchar(500) | NO   |     | NULL    |                |
| date_of_join | date         | NO   |     | NULL    |                |
| name         | varchar(255) | NO   |     | NULL    |                |
| email        | varchar(255) | NO   | UNI | NULL    |                |
| comments     | varchar(511) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

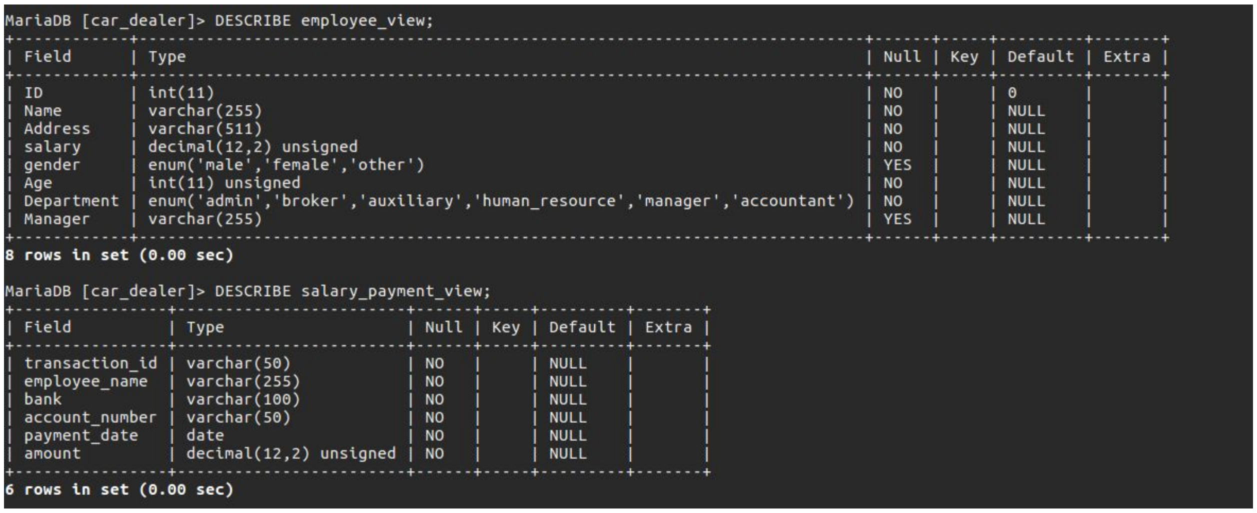


```
MariaDB [car_dealer]> DESCRIBE vendor_order;
+------------+-----------------------------------------+------+-----+---------+----------------+
| Field      | Type                                    | Null | Key | Default | Extra          |
+------------+-----------------------------------------+------+-----+---------+----------------+
| id         | int(11)                                 | NO   | PRI | NULL    | auto_increment |
| vendor_id  | int(11)                                 | NO   | MUL | NULL    |                |
| vehicle_id | int(11)                                 | NO   | MUL | NULL    |                |
| status     | enum('PENDING','DELIVERED','IN_TRANSIT') | NO   |     | PENDING |                |
| quantity   | int(11) unsigned                        | NO   |     | NULL    |                |
+------------+-----------------------------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE vendor_order_customer_order;
+-------------------+---------+------+-----+---------+-------+
| Field             | Type    | Null | Key | Default | Extra |
+-------------------+---------+------+-----+---------+-------+
| vendor_order_id   | int(11) | NO   | PRI | NULL    |       |
| customer_order_id | int(11) | NO   | PRI | NULL    |       |
+-------------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

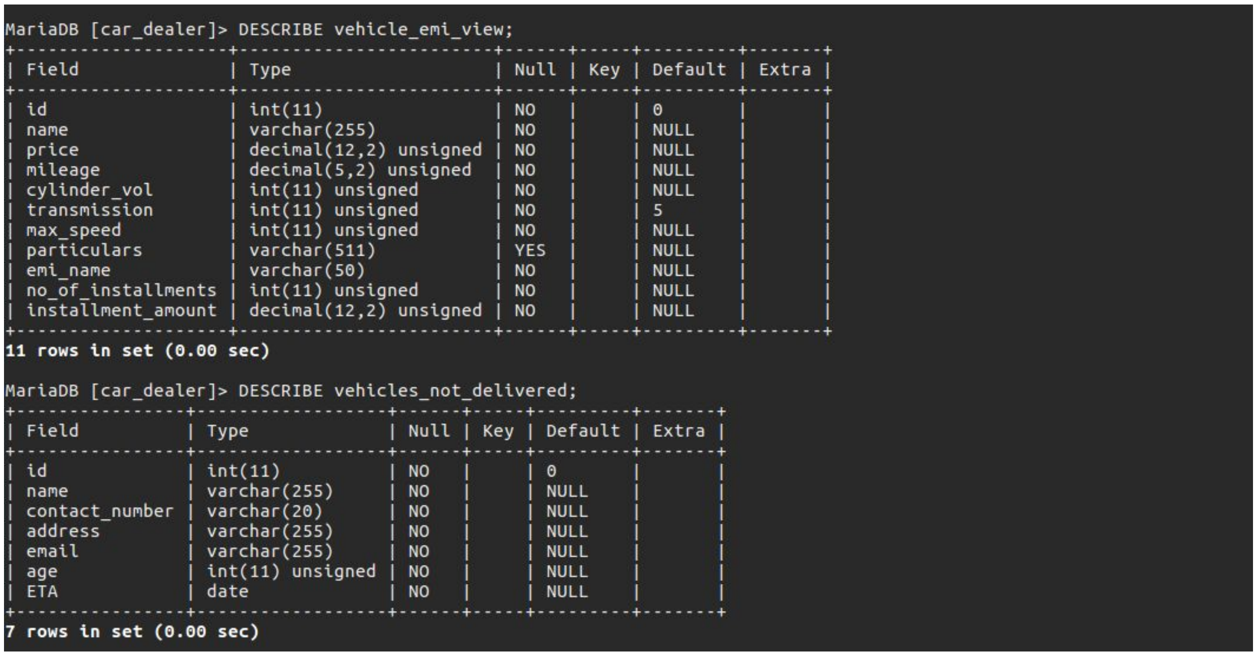
Schema of views



```
MariaDB [car_dealer]> DESCRIBE customer_order_view;
+---------------+-----------------------------------------------------+------+-----+---------+-------+
| Field         | Type                                                | Null | Key | Default | Extra |
+---------------+-----------------------------------------------------+------+-----+---------+-------+
| order_id      | int(11)                                             | NO   |     | 0       |       |
| customer_name | varchar(255)                                        | NO   |     | NULL    |       |
| ETA           | date                                                | NO   |     | NULL    |       |
| status        | enum('PENDING','IN_TRANSIT','READY','DELIVERED')    | NO   |     | NULL    |       |
| vehicle_id    | int(11)                                             | NO   |     | 0       |       |
| vehicle_name  | varchar(255)                                        | NO   |     | NULL    |       |
| order_date    | date                                                | NO   |     | NULL    |       |
+---------------+-----------------------------------------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE customer_payment_view;
+----------------+-------------------------+------+-----+---------+-------+
| Field          | Type                    | Null | Key | Default | Extra |
+----------------+-------------------------+------+-----+---------+-------+
| customer_name  | varchar(255)            | NO   |     | NULL    |       |
| transaction_id | varchar(50)             | NO   |     | NULL    |       |
| emi_name       | varchar(50)             | YES  |     | NULL    |       |
| bank           | varchar(100)            | NO   |     | NULL    |       |
| account_number | varchar(50)             | NO   |     | NULL    |       |
| payment_date   | date                    | NO   |     | NULL    |       |
| amount         | decimal(12,2) unsigned  | NO   |     | NULL    |       |
+----------------+-------------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

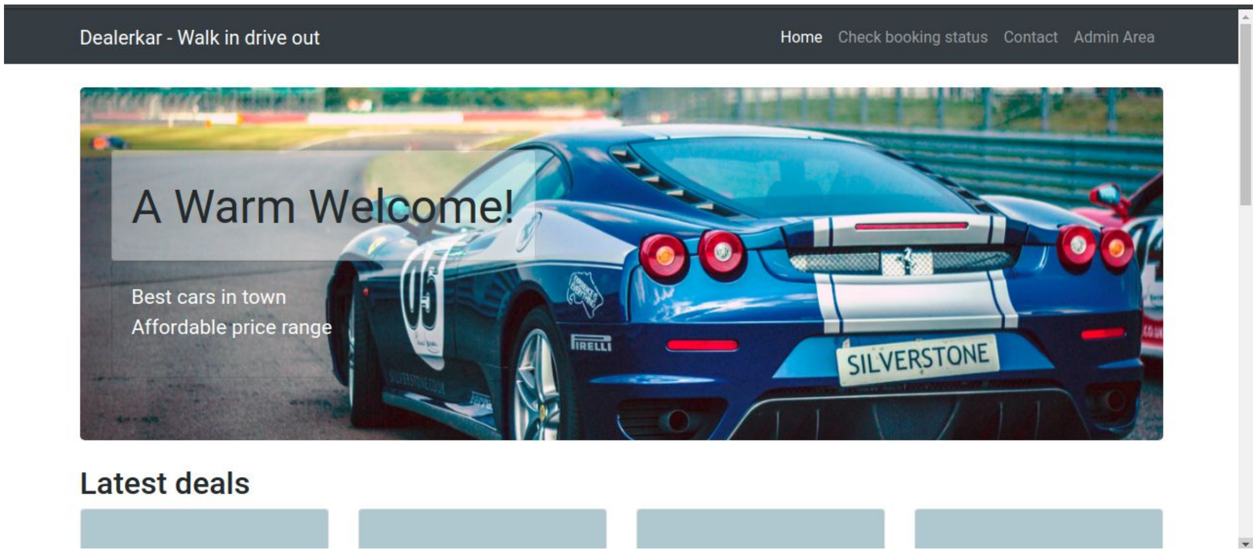


```
MariaDB [car_dealer]> DESCRIBE employee_view;
+------------+-------------------------------------------------------------------------------+------+-----+---------+-------+
| Field      | Type                                                                          | Null | Key | Default | Extra |
+------------+-------------------------------------------------------------------------------+------+-----+---------+-------+
| ID         | int(11)                                                                       | NO   |     | 0       |       |
| Name       | varchar(255)                                                                  | NO   |     | NULL    |       |
| Address    | varchar(511)                                                                  | NO   |     | NULL    |       |
| salary     | decimal(12,2) unsigned                                                        | NO   |     | NULL    |       |
| gender     | enum('male','female','other')                                                 | YES  |     | NULL    |       |
| Age        | int(11) unsigned                                                              | NO   |     | NULL    |       |
| Department | enum('admin','broker','auxiliary','human_resource','manager','accountant')    | NO   |     | NULL    |       |
| Manager    | varchar(255)                                                                  | YES  |     | NULL    |       |
+------------+-------------------------------------------------------------------------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE salary_payment_view;
+----------------+------------------------+------+-----+---------+-------+
| Field          | Type                   | Null | Key | Default | Extra |
+----------------+------------------------+------+-----+---------+-------+
| transaction_id | varchar(50)            | NO   |     | NULL    |       |
| employee_name  | varchar(255)           | NO   |     | NULL    |       |
| bank           | varchar(100)           | NO   |     | NULL    |       |
| account_number | varchar(50)            | NO   |     | NULL    |       |
| payment_date   | date                   | NO   |     | NULL    |       |
| amount         | decimal(12,2) unsigned | NO   |     | NULL    |       |
+----------------+------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



```
MariaDB [car_dealer]> DESCRIBE vehicle_emi_view;
+--------------------+-----------------------+------+-----+---------+-------+
| Field              | Type                  | Null | Key | Default | Extra |
+--------------------+-----------------------+------+-----+---------+-------+
| id                 | int(11)               | NO   |     | 0       |       |
| name               | varchar(255)          | NO   |     | NULL    |       |
| price              | decimal(12,2) unsigned | NO   |     | NULL    |       |
| mileage            | decimal(5,2) unsigned | NO   |     | NULL    |       |
| cylinder_vol       | int(11) unsigned      | NO   |     | NULL    |       |
| transmission       | int(11) unsigned      | NO   |     | 5       |       |
| max_speed          | int(11) unsigned      | NO   |     | NULL    |       |
| particulars        | varchar(511)          | YES  |     | NULL    |       |
| emi_name           | varchar(50)           | NO   |     | NULL    |       |
| no_of_installments | int(11) unsigned      | NO   |     | NULL    |       |
| installment_amount | decimal(12,2) unsigned | NO   |     | NULL    |       |
+--------------------+-----------------------+------+-----+---------+-------+
11 rows in set (0.00 sec)

MariaDB [car_dealer]> DESCRIBE vehicles_not_delivered;
+----------------+------------------+------+-----+---------+-------+
| Field          | Type             | Null | Key | Default | Extra |
+----------------+------------------+------+-----+---------+-------+
| id             | int(11)          | NO   |     | 0       |       |
| name           | varchar(255)     | NO   |     | NULL    |       |
| contact_number | varchar(20)      | NO   |     | NULL    |       |
| address        | varchar(255)     | NO   |     | NULL    |       |
| email          | varchar(255)     | NO   |     | NULL    |       |
| age            | int(11) unsigned | NO   |     | NULL    |       |
| ETA            | date             | NO   |     | NULL    |       |
+----------------+------------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Front end views



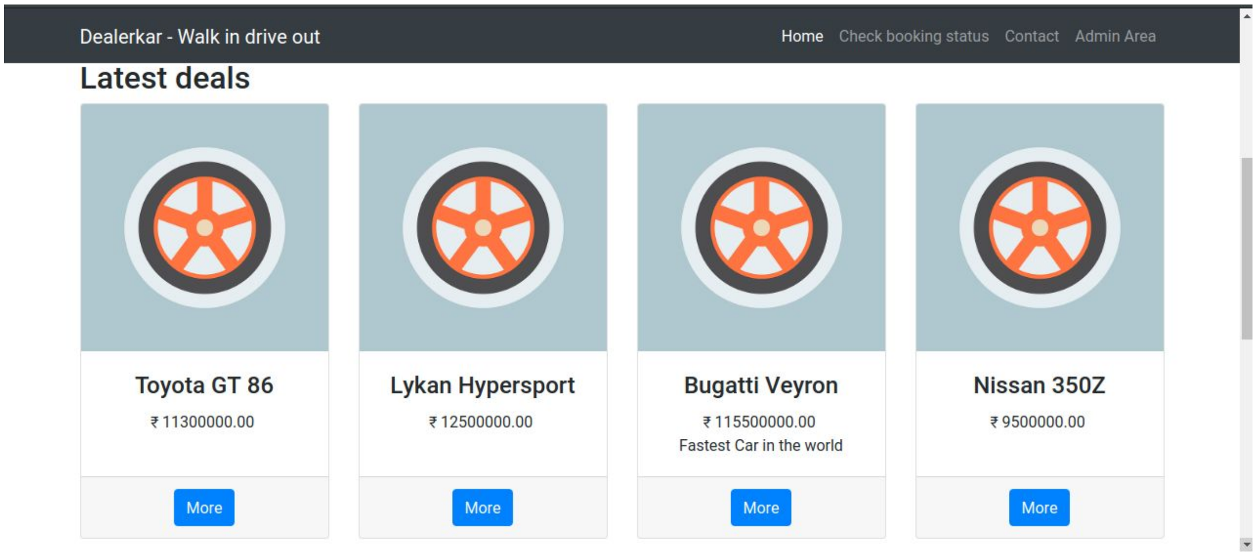
Dealerkar - Walk in drive out

A Warm Welcome!

Best cars in town

Affordable price range

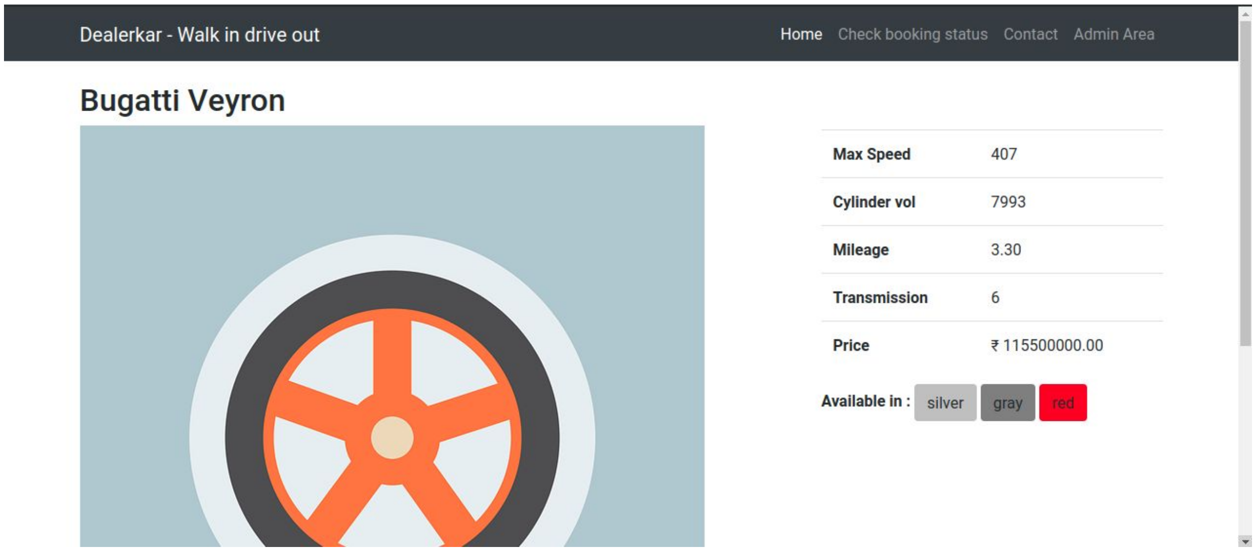
Latest deals



Latest deals

Toyota GT 86	Lykan Hypersport	Bugatti Veyron	Nissan 350Z
₹ 11300000.00	₹ 12500000.00	₹ 115500000.00	₹ 9500000.00
		Fastest Car in the world	

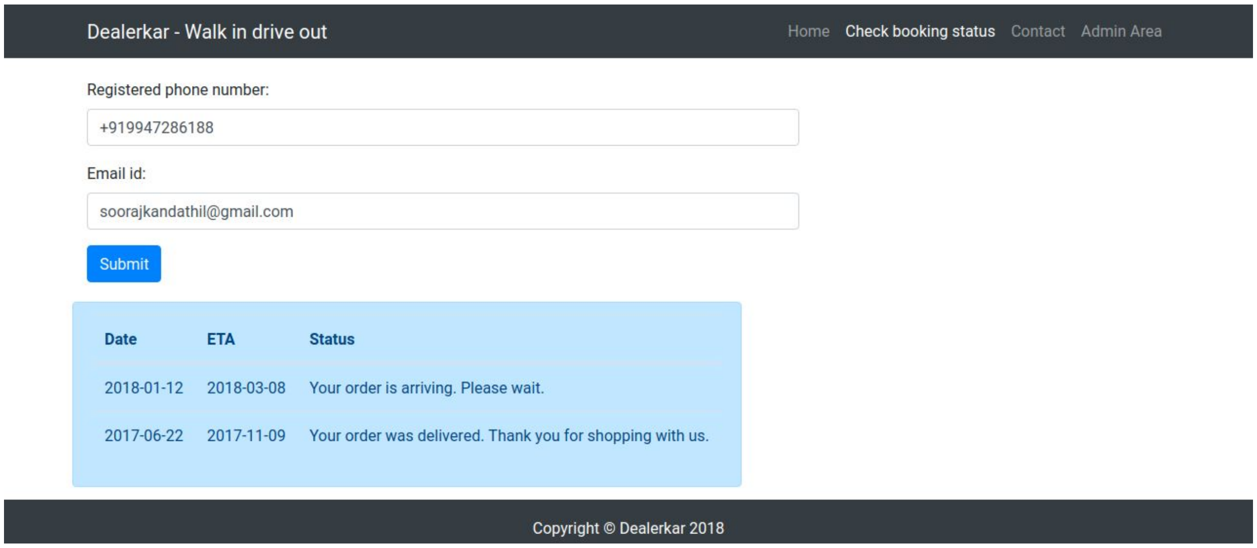
The following diagram shows what happens when you press more...



Bugatti Veyron

Max Speed	407
Cylinder vol	7993
Mileage	3.30
Transmission	6
Price	₹ 115500000.00

Available in : silver gray red



Registered phone number:

+919947286188

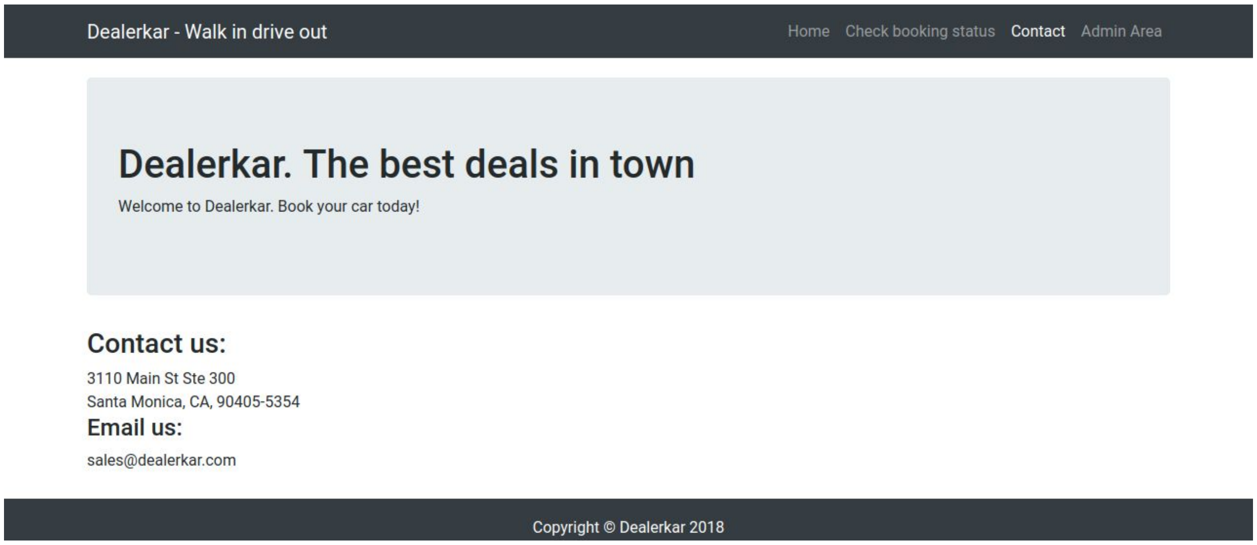
Email id:

soorajkandathil@gmail.com

Submit

Date	ETA	Status
2018-01-12	2018-03-08	Your order is arriving. Please wait.
2017-06-22	2017-11-09	Your order was delivered. Thank you for shopping with us.

Copyright © Dealerkar 2018



Dealerkar. The best deals in town

Welcome to Dealerkar. Book your car today!

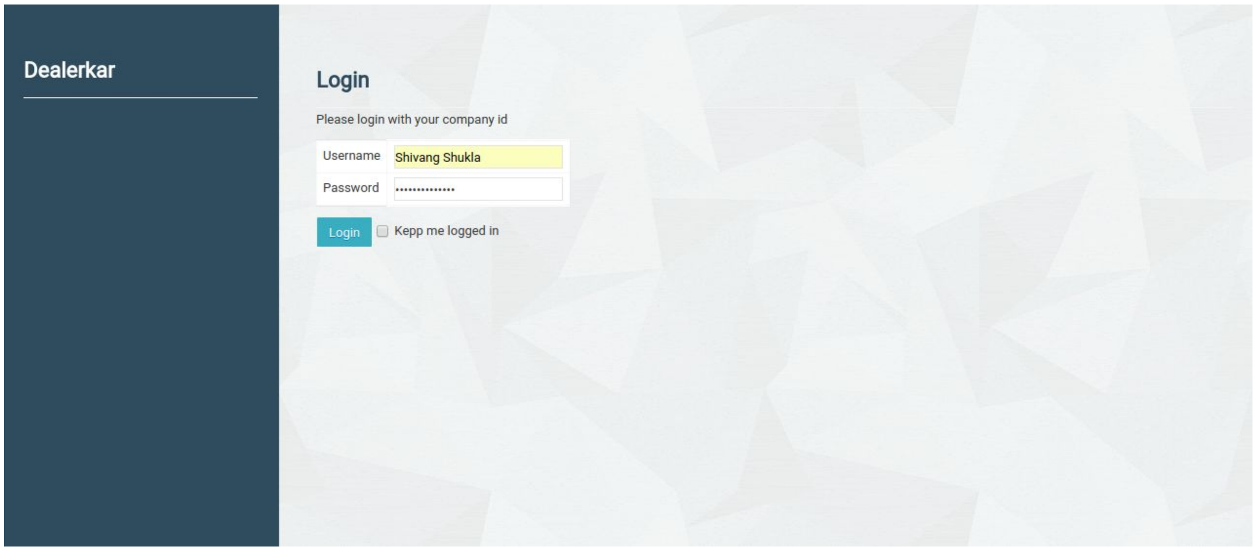
Contact us:

3110 Main St Ste 300
Santa Monica, CA, 90405-5354

Email us:

sales@dealerkar.com

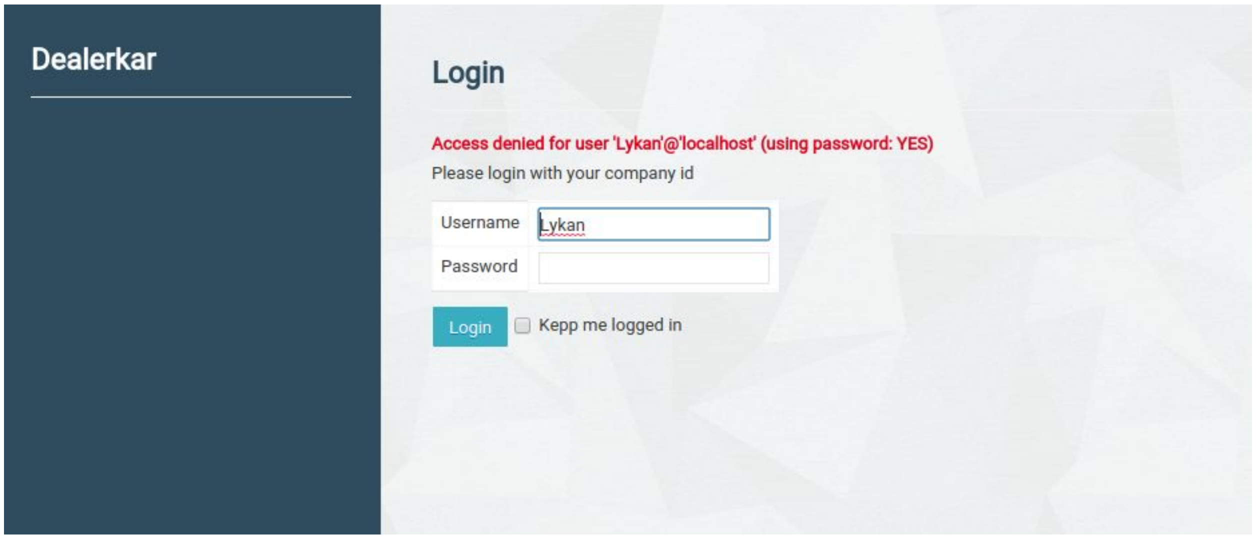
Copyright © Dealerkar 2018



Please login with your company id

Username	Shivang Shukla
Password	••••••••••••••

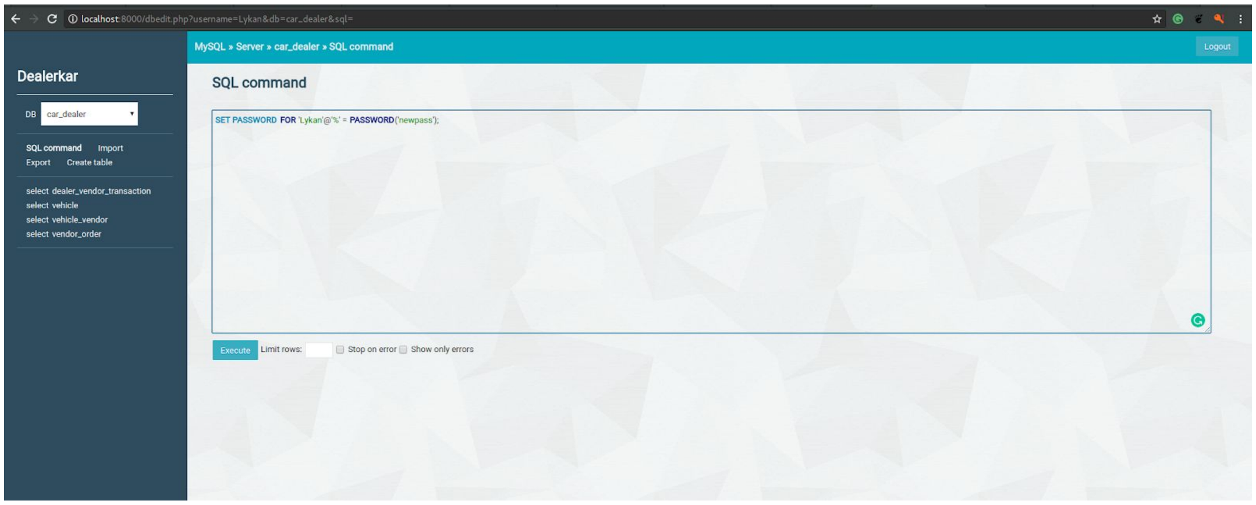
When password is wrong user is told to login with id given by company



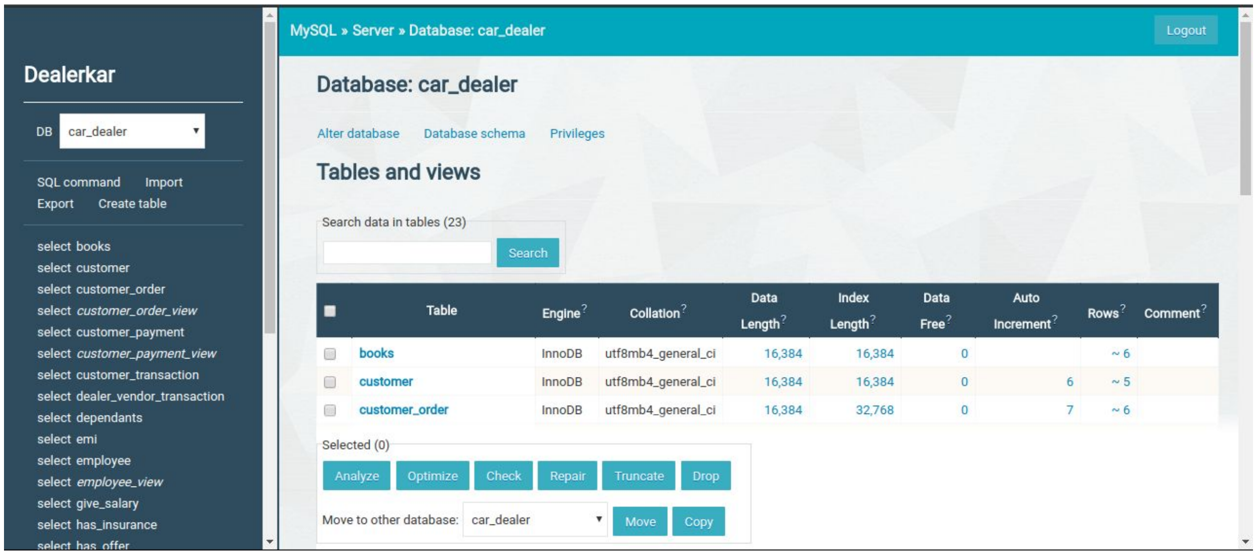
Access denied for user 'Lykan'@'localhost' (using password: YES)
Please login with your company id

Username: Lykan

Changing Password



```
SET PASSWORD FOR 'Lykan'@'%' = PASSWORD('newpass');
```

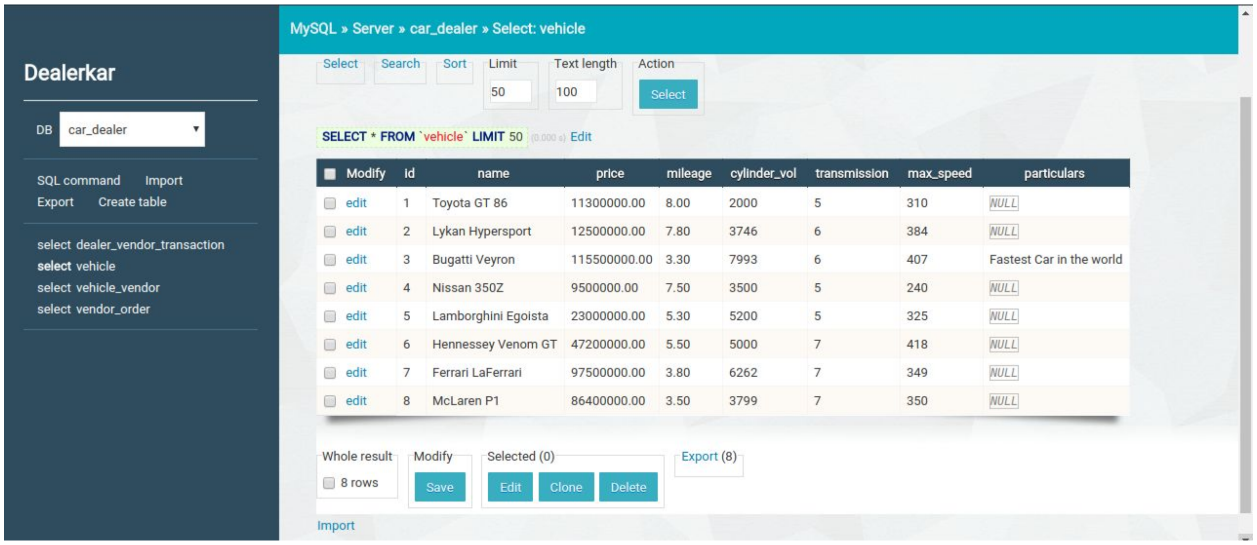


Database: car_dealer

Tables and views

	Table	Engine?	Collation?	Data Length?	Index Length?	Data Free?	Auto Increment?	Rows?	Comment?
☐	books	InnoDB	utf8mb4_general_ci	16,384	16,384	0		~ 6	
☐	customer	InnoDB	utf8mb4_general_ci	16,384	16,384	0	6	~ 5	
☐	customer_order	InnoDB	utf8mb4_general_ci	16,384	32,768	0	7	~ 6	

For a user with privilege manager.



```
SELECT * FROM `vehicle` LIMIT 50 (0.000 s)
```

	Modify	id	name	price	mileage	cylinder_vol	transmission	max_speed	particulars
☐	edit	1	Toyota GT 86	11300000.00	8.00	2000	5	310	NULL
☐	edit	2	Lykan Hypersport	12500000.00	7.80	3746	6	384	NULL
☐	edit	3	Bugatti Veyron	115500000.00	3.30	7993	6	407	Fastest Car in the world
☐	edit	4	Nissan 350Z	9500000.00	7.50	3500	5	240	NULL
☐	edit	5	Lamborghini Egoista	23000000.00	5.30	5200	5	325	NULL
☐	edit	6	Hennessey Venom GT	47200000.00	5.50	5000	7	418	NULL
☐	edit	7	Ferrari LaFerrari	97500000.00	3.80	6262	7	349	NULL
☐	edit	8	McLaren P1	86400000.00	3.50	3799	7	350	NULL

For a user with privilege vender_company_dealer. (lower privilege).

Source : triggers.sql

```sql
DELIMITER //
--
-- CUSTOMER TABLE TRIGGERS
--
DROP TRIGGER IF EXISTS `email_validate_customer` //
CREATE TRIGGER `email_validate_customer`
    BEFORE INSERT
    ON `customer`
    FOR EACH ROW
BEGIN
    CALL check_email(new.email, 'customer');
END;//

DROP TRIGGER IF EXISTS `phone_number_validate_customer` //
CREATE TRIGGER `phone_number_validate_customer`
    BEFORE INSERT
    ON `customer`
    FOR EACH ROW
BEGIN
    SET NEW.contact_number = check_phone_num(NEW.contact_number,
'customer');
END;//
--
-- INSURANCE COMPANY TABLE TRIGGERS
--
DROP TRIGGER IF EXISTS `phone_number_validate_insurance_company` //
CREATE TRIGGER `phone_number_validate_insurance_company`
    BEFORE INSERT
    ON `insurance_company`
    FOR EACH ROW
BEGIN
    SET NEW.contact_number = check_phone_num(NEW.contact_number,
'insurance_company');
END;//
--
-- VEHICLE VENDOR TABLE TRIGGERS
--
```

```
DROP TRIGGER IF EXISTS `email_validate_vehicle_vendor` //
CREATE TRIGGER `email_validate_vehicle_vendor`
    BEFORE INSERT
    ON `vehicle_vendor`
    FOR EACH ROW
BEGIN
    CALL check_email(new.email, 'vehicle_vendor');
END;//

--
-- triggers for policy_manager
--

DROP TRIGGER IF EXISTS `email_validate_policy_manager` //
CREATE TRIGGER `email_validate_policy_manager`
    BEFORE INSERT
    ON `policy_manager`
    FOR EACH ROW
BEGIN
    CALL check_email(new.email, 'policy_manager');
END;//

DROP TRIGGER IF EXISTS `phone_number_validate_policy_manager` //
CREATE TRIGGER `phone_number_validate_policy_manager`
    BEFORE INSERT
    ON `policy_manager`
    FOR EACH ROW
BEGIN
    SET NEW.contact_number = check_phone_num(NEW.contact_number,
'policy_manager');
END;//

-- -
-- - TRIGGER FOR VALIDATING EMI OFFER FOR VEHICLES
-- -
DROP TRIGGER IF EXISTS `has_offer_validations` //
CREATE TRIGGER `has_offer_validations`
    BEFORE INSERT
    ON `has_offer`
    FOR EACH ROW
BEGIN
DECLARE amount DECIMAL(12,2);
DECLARE price DECIMAL(12,2);
SET amount = (SELECT no_of_installments*installment_amount
           FROM emi
     WHERE id=new.emi_id);
SET price = (SELECT price
           FROM vehicle
     WHERE id=new.vehicle_id);
    IF amount >= price THEN
     SIGNAL SQLSTATE VALUE '45000'
     SET MESSAGE_TEXT = '[table : has_offer] - total price of
vehicle not matching emi aggregate';
    END IF;
END;//

-- -
-- - TRIGGER FOR VALIDATING CUSTOMER
-- -

DROP TRIGGER IF EXISTS `validate_age` //
CREATE TRIGGER `validate_age`
    BEFORE INSERT
    ON `customer`
    FOR EACH ROW
BEGIN
    IF new.age < 18 THEN
     SIGNAL SQLSTATE VALUE '45000'
     SET MESSAGE_TEXT = '[table : customer] - customer should be
atleast 18 years old';
    END IF;
END;//


-- -
-- - TRIGGER FOR VALIDATING EMI REGISTRATION
-- -
```

```
DROP TRIGGER IF EXISTS `emi_register_check` //
CREATE TRIGGER `emi_register_check`
    BEFORE INSERT
    ON `registered`
    FOR EACH ROW
BEGIN
DECLARE v_id INT;
SET v_id = (SELECT vehicle_id FROM books WHERE customer_order_id =
new.customer_order_id);
    IF new.emi_id NOT IN (SELECT emi_id FROM has_offer WHERE
vehicle_id = v_id) THEN
      SIGNAL SQLSTATE VALUE '45000'
      SET MESSAGE_TEXT = '[table : registered] - emi_id entered is
invalid for given vehicle';
    END IF;
END;//


DROP TRIGGER IF EXISTS `emi_avalibility_check` //
CREATE TRIGGER `emi_avalibility_check`
    BEFORE INSERT
    ON `customer_payment`
    FOR EACH ROW
BEGIN
DECLARE v_id INT;
SET v_id = (SELECT vehicle_id FROM books WHERE customer_order_id =
new.order_id);
      IF new.type = 'emi' THEN
    IF NOT EXISTS (SELECT emi_id
                       FROM has_offer
                       WHERE vehicle_id = v_id) THEN
      SIGNAL SQLSTATE VALUE '45000'
      SET MESSAGE_TEXT = '[table : customer_payment] - emi plan does
not exist for given vehicle';
    END IF;
      END IF;
END;//

--
```

```
-- CHANGE STATUS IN customer order on reciving Vendor order status
change
--
DROP TRIGGER IF EXISTS `change_order_status` //
CREATE TRIGGER `change_order_status`
    AFTER UPDATE
    ON `vendor_order`
    FOR EACH ROW
BEGIN
DECLARE order_id INT;
SET order_id = (SELECT customer_order_id FROM
vendor_order_customer_order WHERE vendor_order_id=new.id);
      IF new.status = 'DELIVERED' THEN
       UPDATE customer_order SET status='READY' WHERE id=order_id;
      END IF;
END;//

--
-- CHANGE STATUS IN customer order on receiving Vendor order status
change
--
DROP TRIGGER IF EXISTS `change_order_status2` //
CREATE TRIGGER `change_order_status2`
    AFTER INSERT
    ON `vendor_order`
    FOR EACH ROW
BEGIN
DECLARE order_id INT;
SET order_id = (SELECT customer_order_id FROM
vendor_order_customer_order WHERE vendor_order_id=new.id);
      IF new.status = 'DELIVERED' THEN
       UPDATE customer_order SET status='READY' WHERE id=order_id;
      END IF;
END;//

--
-- CHANGE STATUS IN customer order on inserting Vendor order
--
DROP TRIGGER IF EXISTS `change_order_status1` //
```

```
CREATE TRIGGER `change_order_status1`
    AFTER INSERT
    ON `vendor_order_customer_order`
    FOR EACH ROW
BEGIN
    DECLARE status_ enum('PENDING','IN_TRANSIT','READY','DELIVERED');
    SET status_ = (SELECT status FROM customer_order WHERE
id=new.customer_order_id);
    IF (status_='PENDING' ) THEN
      UPDATE customer_order SET status='IN_TRANSIT' WHERE
id=new.customer_order_id;
    END IF;
END;//

--
-- CHANGE STATUS in customer order on inserting customer payment
--
DROP TRIGGER IF EXISTS `change_order_status3` //
CREATE TRIGGER `change_order_status3`
    AFTER INSERT
    ON `customer_payment`
    FOR EACH ROW
BEGIN
DECLARE status_ enum('PENDING','IN_TRANSIT','READY','DELIVERED')    ;
DECLARE amount_paid DECIMAL(12,2);
DECLARE v_price DECIMAL(12,2);
SET status_ = (SELECT status FROM customer_order WHERE
id=new.order_id);
SET amount_paid = (SELECT SUM(T.amount)
              FROM customer_transaction AS T,
        customer_payment AS P
        WHERE P.order_id = new.order_id
        AND P.transaction_id = T.transaction_id);
SET v_price = (SELECT V.price FROM books AS B, vehicle AS V ,
customer_order AS CO
          WHERE V.id = B.vehicle_id and CO.id = B.customer_order_id
and CO.id = new.order_id );
    IF (status_='READY') THEN
     IF new.type = 'emi' OR (amount_paid=v_price) THEN
```

```
          UPDATE customer_order SET status='DELIVERED' WHERE
id=new.order_id;
      END IF;
    END IF;
END;//

--
-- CHECK STATUS of customer order on inserting customer payment (type
ready)
--

DROP TRIGGER IF EXISTS `check_for_normal_payment` //
CREATE TRIGGER `check_for_normal_payment`
    BEFORE INSERT
    ON `customer_payment`
    FOR EACH ROW
BEGIN
    DECLARE status_ enum('PENDING','IN_TRANSIT','READY','DELIVERED')
;
    SET status_ = (SELECT status FROM customer_order WHERE
id=new.order_id);
    IF ((new.type = 'normal') AND (NOT status_='READY')) THEN
      SIGNAL SQLSTATE VALUE '45000'
           SET MESSAGE_TEXT = '[table : customer_payment] - Your Car
has not reached showroom yet';
    END IF;
END;//

DELIMITER ;
-- show triggers \G;
```

Source : functions_procedures.sql

```
DELIMITER //
--
-- FUNCTION TO INCREMENT BY PERCENT
--
DROP FUNCTION IF EXISTS inc_by_percent;

CREATE FUNCTION inc_by_percent(salary DECIMAL(12,2), percent
DECIMAL(5,2)) RETURNS DECIMAL(12,2)

BEGIN
    SET salary = salary + salary*percent;
    RETURN salary;
END //

--
-- Validation Functions and Errors
--

DROP FUNCTION IF EXISTS check_phone_num;
CREATE FUNCTION check_phone_num (phone_no VARCHAR(20), t_name
VARCHAR(20)) RETURNS  VARCHAR(20)
BEGIN
    DECLARE msg VARCHAR (120);
    SET msg = CONCAT('[table:', t_name, '] - `contact_number` column
is not valid phone number');
    IF phone_no NOT RLIKE '^(\\+?[0-9]{1,3})?[0-9]{10}$' THEN
      SIGNAL SQLSTATE VALUE '45000'
           SET MESSAGE_TEXT = msg;
    ELSE
    BEGIN
      IF (LENGTH(phone_no)=10) THEN
          SET phone_no = CONCAT('+91', phone_no);
      END IF;
      IF (phone_no NOT LIKE '+%') AND NOT (LENGTH(phone_no)=10) THEN
               SET phone_no = CONCAT('+', phone_no);
      END IF;
    END;
    END IF;
    RETURN phone_no;
END//

DROP PROCEDURE IF EXISTS check_email;
CREATE PROCEDURE check_email(IN email VARCHAR(255),
                                       IN t_name VARCHAR(20))
BEGIN
    DECLARE msg VARCHAR(120);
    SET msg = CONCAT('[table:', t_name, '] - `email` column is not
valid');
    IF email NOT LIKE '%_@%_.__%' THEN
      SIGNAL SQLSTATE VALUE '45000'
              SET MESSAGE_TEXT = msg;
    END IF;

END//

--
-- PROCEDURE TO INSERT CUSTOMER PAYMENT emi_id is NULL if type is not
emi
--
DROP PROCEDURE IF EXISTS insert_customer_payment;
CREATE PROCEDURE insert_customer_payment(
                   IN order_id INT,
                   IN transaction_id VARCHAR(50),
        IN type enum('advance','emi','normal'),
        IN emi_id INT,
        IN bank VARCHAR(100),
        IN account_number VARCHAR(50),
        IN payment_date DATE,
        IN amount DECIMAL(12,2))
BEGIN
DECLARE _date DATE ;
DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLWARNING
BEGIN
      ROLLBACK;
END;
START TRANSACTION;
```

```
SET _date = IF(ISNULL(payment_date), CURDATE(), payment_date);
INSERT INTO customer_transaction
      (transaction_id,
      bank,
      date,
      account_number,
      amount) VALUES (transaction_id,
                            bank,
            _date,
                      account_number,
                      amount);
INSERT INTO customer_payment(transaction_id, order_id, type)
      VALUES (transaction_id, order_id, type);
      IF (type='emi' AND (NOT EXISTS (SELECT customer_order_id FROM
registered))) THEN
              INSERT INTO registered(customer_order_id, emi_id)
       VALUES (order_id, emi_id);
      END IF;
COMMIT;
END //

--
-- PROCEDURE FOR INSERTING SALARY PAYMENT
--
DROP PROCEDURE IF EXISTS insert_salary_payment;
CREATE PROCEDURE insert_salary_payment(
                   IN employee_id INT,
                   IN transaction_id VARCHAR(50),
        IN bank VARCHAR(100),
        IN account_number VARCHAR(50),
        IN payment_date DATE,
        IN amount DECIMAL(12,2))
BEGIN
DECLARE salary DECIMAL(12,2) ;
DECLARE _date DATE ;
DECLARE EXIT HANDLER FOR SQLEXCEPTION, SQLWARNING
BEGIN
      ROLLBACK;
END;
```

```sql
START TRANSACTION;
SET salary = (SELECT E.salary
           FROM employee AS E
      WHERE E.id=employee_id);
SET _date = IF(ISNULL(payment_date), CURDATE(), payment_date);
      IF salary > amount THEN
      SIGNAL SQLSTATE VALUE '45000'
      SET MESSAGE_TEXT = 'THE EMPLOYEE IS PAID SALARY LESS THAN THE
BASE SALARY';
      END IF;
INSERT INTO salary_transaction
      (transaction_id,
      bank,
      date,
      account_number,
      amount) VALUES (transaction_id,
                           bank, _date,
                     account_number,
                     amount);
INSERT INTO give_salary(employee_id, transaction_id)
      VALUES (employee_id, transaction_id);
COMMIT;
END //

--
-- PROCEDURE FOR GETTING SUMMARY OF SALARY PAID TO AN EMPLOYER
--
DROP PROCEDURE IF EXISTS salary_summary;
CREATE PROCEDURE salary_summary(IN empid INTEGER)
BEGIN
    SELECT E.name, ST.date, ST.amount FROM `employee` AS E,
`give_salary` AS GS, `salary_transaction` AS ST
    WHERE E.id=empid AND GS.employee_id=empid AND
ST.transaction_id=GS.transaction_id;
END//

--
-- PROCEDURE TO FIND CUSTOMER WHO OPTED FOR emi AND NOT PAYED ANY
PAYMENTS IN LAST MONTH
```

```
--

DROP PROCEDURE IF EXISTS pending_emi_payments;
CREATE PROCEDURE pending_emi_payments()
BEGIN
DECLARE cur_date DATE;
DECLARE past_month DATE;
SET cur_date = CURDATE();
SET past_month = DATE_SUB( cur_date, INTERVAL 1 MONTH);
     SELECT C.name AS name,
          CO.id as order_id
     FROM customer AS C  ,
     customer_order AS CO,
     customer_payment AS CP,
     customer_transaction AS CT,
     emi as E,
     registered AS R
     WHERE C.id = CO.customer_id
     and CP.order_id = CO.id
     and CP.type = 'emi'
     and CT.transaction_id = CP.transaction_id
     and CT.date NOT BETWEEN cur_date and past_month
     and E.id = R.emi_id
     and R.customer_order_id = CO.id
     and NOT ((E.no_of_installments*E.installment_amount) = (SELECT
                      SUM(T.amount)
           FROM customer_transaction AS T
           WHERE transaction_id = CT.transaction_id));
END //

--
-- PROCEDURE FOR INCREMENTING SALARY FOR PARTICULAR EMPLOYEE
--
DROP PROCEDURE IF EXISTS salary_increment;

CREATE PROCEDURE salary_increment(IN empid INTEGER,
                                        IN percent DECIMAL(12,2))
BEGIN
    UPDATE employee SET salary = inc_by_percent(salary,percent) WHERE
```

```
id=empid;
END //

--
-- PROCEDURE FOR INCREMENTING SALARY FOR ALL EMPLOYEES
--
DROP PROCEDURE IF EXISTS salary_increment_all;

CREATE PROCEDURE salary_increment_all(IN percent DECIMAL(12,2))
BEGIN
    UPDATE employee SET salary = inc_by_percent(salary,percent);
END //

--
-- PROCEDURE FOR DECREMENTING SALARY FOR PARTICULAR EMPLOYEE
--
DROP PROCEDURE IF EXISTS salary_decrement;

CREATE PROCEDURE salary_decrement(IN empid INTEGER,
                                            IN percent DECIMAL(12,2))
BEGIN
    UPDATE employee SET salary = inc_by_percent(salary,(-percent))
WHERE id=empid;
END //

DELIMITER ;
-- show procedure status \G;
-- show function status \G
```

Source : create_roles.sql

```
CREATE OR REPLACE ROLE DBA;
GRANT ALL PRIVILEGES ON car_dealer.* TO DBA WITH GRANT OPTION;

CREATE OR REPLACE ROLE manager;
GRANT SELECT, DELETE, UPDATE, INSERT ON car_dealer.* TO manager WITH GRANT
OPTION;
```

```
CREATE OR REPLACE ROLE accountant;
GRANT EXECUTE ON PROCEDURE car_dealer.insert_customer_payment TO
accountant;
GRANT EXECUTE ON PROCEDURE car_dealer.insert_salary_payment TO accountant;
GRANT EXECUTE ON PROCEDURE car_dealer.pending_emi_payments TO accountant;
GRANT SELECT, UPDATE, DELETE ON car_dealer.customer_payment_view TO
accountant;
GRANT SELECT, UPDATE, DELETE ON car_dealer.salary_payment_view TO
accountant;
GRANT SELECT ON car_dealer.emi TO accountant;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.registered TO
accountant;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer_transaction TO
accountant;
GRANT SELECT, UPDATE, DELETE, INSERT ON
car_dealer.dealer_vendor_transaction TO accountant;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.salary_transaction TO
accountant;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer_payment TO
accountant;
GRANT SELECT ON car_dealer.customer_order_view TO accountant;
GRANT SELECT ON car_dealer.vehicle_emi_view TO accountant;
GRANT SELECT ON car_dealer.vendor_order TO accountant;

CREATE OR REPLACE ROLE broker;
GRANT SELECT ON car_dealer.customer_order_view TO broker;
GRANT SELECT ON car_dealer.vehicle_emi_view TO broker;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer TO broker;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer_order TO
broker;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.books TO broker;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer_order TO
broker;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.customer_order TO
broker;
GRANT SELECT ON car_dealer.vehicle_insurance TO broker;
GRANT SELECT ON car_dealer.has_offer TO broker;
GRANT SELECT ON car_dealer.has_insurance TO broker;
GRANT SELECT ON car_dealer.vehicle TO broker;
GRANT SELECT ON car_dealer.registered TO broker;
GRANT SELECT ON car_dealer.insurance_company TO broker;
GRANT SELECT ON car_dealer.vehicle_insurance TO broker;
GRANT SELECT ON car_dealer.emi TO broker;
GRANT SELECT ON car_dealer.customer_payment TO broker;
GRANT SELECT ON car_dealer.customer_transaction TO broker;

CREATE OR REPLACE ROLE insurance_agent;
GRANT SELECT ON car_dealer.insurance_company TO insurance_agent;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.vehicle_insurance TO
insurance_agent;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.has_insurance TO
insurance_agent;
GRANT SELECT ON car_dealer.vehicle TO insurance_agent;
GRANT SELECT ON car_dealer.customer TO insurance_agent;
GRANT SELECT ON car_dealer.customer_order TO insurance_agent;
GRANT SELECT ON car_dealer.books TO insurance_agent;

CREATE OR REPLACE ROLE vendor_company_dealer;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.vehicle TO
vendor_company_dealer;
GRANT SELECT ON car_dealer.vehicle_vendor TO vendor_company_dealer;
GRANT SELECT, UPDATE, DELETE, INSERT ON car_dealer.vendor_order TO
vendor_company_dealer;
GRANT SELECT ON car_dealer.dealer_vendor_transaction TO
vendor_company_dealer;




--
-- CREATING USER
--

DELIMITER //

CREATE OR REPLACE PROCEDURE create_user( IN username VARCHAR(30),
                           IN `password` VARCHAR(30), IN role VARCHAR(30))
BEGIN
```

```
DECLARE stmt VARCHAR(100);
prepare stmt FROM CONCAT('CREATE USER IF NOT EXISTS ''', username,
'''@''%'' IDENTIFIED BY ''',`password`, '''');
execute stmt;
DEALLOCATE PREPARE stmt;
prepare stmt FROM CONCAT('GRANT ', role , ' TO ''', username, '''');
execute stmt;
DEALLOCATE PREPARE stmt;
prepare stmt FROM CONCAT('SET DEFAULT ROLE ' , role,' FOR ''',  username,
'''@''%''');
execute stmt;
DEALLOCATE PREPARE stmt;
END //


CREATE OR REPLACE PROCEDURE INITEMPLOYEE()
BEGIN
DECLARE n INT DEFAULT 0;
DECLARE i INT DEFAULT 0;
DECLARE username VARCHAR(30);
DECLARE role VARCHAR(30);
DECLARE _dept VARCHAR(20);
SELECT COUNT(*) FROM employee INTO n;
SET i = 0;
WHILE i<n DO
  SET username = (SELECT name FROM employee LIMIT i,1);
  SET _dept = (SELECT dept FROM employee LIMIT i,1);
      IF (_dept = 'manager') THEN
       SET role = 'manager';
    END IF;
    IF (_dept = 'accountant') THEN
       SET role = 'accountant';
    END IF;
    IF (_dept = 'broker') THEN
       SET role = 'broker';
    END IF;
    IF (_dept = 'admin') THEN
       SET role = 'DBA';
    END IF;
  CALL create_user(username, username, role);
  SET i = i + 1;
END WHILE;
END //

CREATE OR REPLACE PROCEDURE INIT_VEHICLE_DEALER()
BEGIN
DECLARE n INT DEFAULT 0;
DECLARE i INT DEFAULT 0;
DECLARE username VARCHAR(30);
DECLARE role VARCHAR(30);
SELECT COUNT(*) FROM vehicle_vendor INTO n;
SET role = 'vendor_company_dealer';
SET i = 0;
WHILE i<n DO
  SET username = (SELECT name FROM vehicle_vendor LIMIT i,1);
  CALL create_user(username, username, role);
  SET i = i + 1;
END WHILE;
END //


CREATE OR REPLACE PROCEDURE INIT_INSURANCE_AGENT()
BEGIN
DECLARE n INT DEFAULT 0;
DECLARE i INT DEFAULT 0;
DECLARE username VARCHAR(30);
DECLARE role VARCHAR(30);
SELECT COUNT(*) FROM insurance_company INTO n;
SET role = 'insurance_agent';
SET i = 0;
WHILE i<n DO
  SET username = (SELECT name FROM insurance_company LIMIT i,1);
  CALL create_user(username, username, role);
  SET i = i + 1;
END WHILE;
END //

DELIMITER ;
```