Q.1) Car Rental Management System Description: Develop a database to manage car rentals, customer details, car availability, and billing.

1. Database Table Design

**Table: Customers**

This table holds customer details.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| customer\_id | INT (PK) | Unique identifier for each customer |
| first\_name | VARCHAR(50) | Customer's first name |
| last\_name | VARCHAR(50) | Customer's last name |
| phone\_number | VARCHAR(15) | Customer's contact number |
| email | VARCHAR(100) | Customer's email address |
| address | VARCHAR(255) | Customer's address |

**Table: Cars**

This table stores car information.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| car\_id | INT (PK) | Unique identifier for each car |
| make | VARCHAR(50) | Car manufacturer |
| model | VARCHAR(50) | Car model |
| year | INT | Year of manufacture |
| car\_type | VARCHAR(50) | Type (e.g., sedan, SUV) |
| available | BOOLEAN | Availability status |
| rate\_per\_day | DECIMAL(10,2) | Daily rental rate |

**Table: Rentals**

This table records details of rentals.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| rental\_id | INT (PK) | Unique identifier for each rental |
| customer\_id | INT (FK) | References Customers |
| car\_id | INT (FK) | References Cars |
| pickup\_date | DATE | Rental start date |
| dropoff\_date | DATE | Rental end date |
| total\_amount | DECIMAL(10,2) | Total rental cost |

**Table: Billing**

This table manages billing information.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| billing\_id | INT (PK) | Unique billing record |
| rental\_id | INT (FK) | References Rentals |
| billing\_date | DATE | Date of billing |
| amount | DECIMAL(10,2) | Amount billed |
| payment\_status | VARCHAR(20) | Status of payment (e.g., paid/unpaid) |

**Table: Additional\_Services**

This table logs additional services like GPS, insurance, etc.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| service\_id | INT (PK) | Unique identifier for each service |
| service\_name | VARCHAR(50) | Name of service (e.g., GPS) |
| service\_rate | DECIMAL(10,2) | Rate for the service |
| description | VARCHAR(255) | Service description |

**Table: Rental\_Services**

This table links Rentals with additional services.

| **Column** | **Data Type** | **Description** |
| --- | --- | --- |
| rental\_id | INT (FK) | References Rentals |
| service\_id | INT (FK) | References Additional\_Services |

### 2. Constraints for Referential Integrity

* **Foreign Keys**: Enforce relationships:
  + customer\_id in Rentals references Customers(customer\_id)
  + car\_id in Rentals references Cars(car\_id)
  + rental\_id in Billing references Rentals(rental\_id)
  + rental\_id in Rental\_Services references Rentals(rental\_id)
  + service\_id in Rental\_Services references Additional\_Services(service\_id)
* **Primary Keys**: Each table has a primary key for unique identification.
* **Check Constraints**: Ensure data consistency, such as non-negative rates and valid dates.

### 3. Stored Procedures

#### a. Check Car Availability

sql

Copy code

CREATE PROCEDURE CheckCarAvailability(IN carId INT, IN checkDate DATE)

BEGIN

DECLARE isAvailable BOOLEAN;

SELECT available INTO isAvailable FROM Cars WHERE car\_id = carId;

IF isAvailable THEN

SELECT 'Car is available';

ELSE

SELECT 'Car is not available';

END IF;

END;

#### b. Make a Rental

sql

Copy code

CREATE PROCEDURE MakeRental(IN custId INT, IN carId INT, IN pickup DATE, IN dropoff DATE)

BEGIN

DECLARE total DECIMAL(10,2);

DECLARE dailyRate DECIMAL(10,2);

SELECT rate\_per\_day INTO dailyRate FROM Cars WHERE car\_id = carId;

SET total = DATEDIFF(dropoff, pickup) \* dailyRate;

INSERT INTO Rentals (customer\_id, car\_id, pickup\_date, dropoff\_date, total\_amount)

VALUES (custId, carId, pickup, dropoff, total);

UPDATE Cars SET available = FALSE WHERE car\_id = carId;

END;

#### c. Process Return

sql

Copy code

CREATE PROCEDURE ProcessReturn(IN rentalId INT)

BEGIN

DECLARE carId INT;

SELECT car\_id INTO carId FROM Rentals WHERE rental\_id = rentalId;

UPDATE Cars SET available = TRUE WHERE car\_id = carId;

END;

### 4. Triggers

#### a. Trigger on Rental Pickup (Update Car Availability)

sql

Copy code

CREATE TRIGGER BeforeRentalPickup

BEFORE INSERT ON Rentals

FOR EACH ROW

BEGIN

UPDATE Cars SET available = FALSE WHERE car\_id = NEW.car\_id;

END;

#### b. Trigger on Rental Drop-off (Update Car Availability)

sql

Copy code

CREATE TRIGGER AfterRentalDropoff

AFTER UPDATE ON Rentals

FOR EACH ROW

WHEN NEW.dropoff\_date IS NOT NULL

BEGIN

UPDATE Cars SET available = TRUE WHERE car\_id = NEW.car\_id;

END;

### 5. SQL Queries for Reports

#### a. Generate Report on Car Usage

sql

Copy code

SELECT car\_id, COUNT(rental\_id) AS usage\_count

FROM Rentals

GROUP BY car\_id

ORDER BY usage\_count DESC;

#### b. Generate Report on Popular Services

sql

Copy code

SELECT service\_name, COUNT(rental\_id) AS usage\_count

FROM Rental\_Services rs

JOIN Additional\_Services s ON rs.service\_id = s.service\_id

GROUP BY service\_name

ORDER BY usage\_count DESC;

#### c. Generate Revenue by Car Type

sql

Copy code

SELECT car\_type, SUM(total\_amount) AS total\_revenue

FROM Rentals r

JOIN Cars c ON r.car\_id = c.car\_id

GROUP BY car\_type

ORDER BY total\_revenue DESC;