1).create DATABASE RideHailingApp;

2). CREATE TABLE Drivers (

DriverID INTEGER AUTO\_INCREMENT PRIMARY KEY,

FirstName Text(50),

LastName Text(50),

Phone Text(15),

City Text(50),

VehicleType ENUM NOT NULL,

Rating DECIMAL(2, 1) CHECK (Rating >= 0 AND Rating <= 5)

);

3). CREATE TABLE Riders(RiderID int AUTO\_INCREMENT PRIMARY KEY , FirstName Text (50), LastName Text (50),Phone Integer (15) not null,City Text (30),JoinDate date);

4). CREATE TABLE Rides (

RideID INTEGER AUTO\_INCREMENT PRIMARY KEY,

RiderID INTEGER,

DriverID INTEGER,

RideDate DATETIME,

PickupLocation Text(255),

DropLocation Text(255),

Distance DECIMAL(5, 2), -- Distance in km

Fare DECIMAL(10, 2),

RideStatus ENUM NOT NULL,

FOREIGN KEY (RiderID) REFERENCES Riders(RiderID),

FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)

);

5). CREATE TABLE Payments (

PaymentID INTEGER AUTO\_INCREMENT PRIMARY KEY,

RideID INTEGER,

PaymentMethod ENUM NOT NULL,

Amount DECIMAL(10, 2),

PaymentDate DATETIME,

FOREIGN KEY (RideID) REFERENCES Rides(RideID)

);

1). INSERT INTO Drivers (DriverID, FirstName, LastName, Phone, City, VehicleType, Rating)

VALUES

(1, 'Amit', 'Sharma', '9876543210', 'Delhi', 'Sedan', 4.5),

(2, 'Priya', 'Patel', '9876543211', 'Mumbai', 'SUV', 4.8),

(3, 'Rahul', 'Kumar', '9876543212', 'Bangalore', 'Hatchback', 4.2),

(4, 'Sneha', 'Iyer', '9876543213', 'Chennai', 'Sedan', 4.7),

(5, 'Vikram', 'Singh', '9876543214', 'Kolkata', 'SUV', 4.3),

(6, 'Neha', 'Mehta', '9876543215', 'Ahmedabad', 'Sedan', 4.9),

(7, 'Ravi', 'Nair', '9876543216', 'Hyderabad', 'Hatchback', 4.6),

(8, 'Kiran', 'Reddy', '9876543217', 'Pune', 'SUV', 4.4),

(9, 'Sita', 'Verma', '9876543218', 'Lucknow', 'Sedan', 4.8),

(10, 'Rohit', 'Joshi', '9876543219', 'Jaipur', 'Hatchback', 4.1);

2). INSERT INTO Riders (RiderID, FirstName, LastName, Phone, City, JoinDate)

VALUES

(101, 'Aarav', 'Gupta', '8888888881', 'Delhi', '2023-01-15'),

(102, 'Isha', 'Reddy', '8888888882', 'Mumbai', '2022-12-05'),

(103, 'Karan', 'Sharma', '8888888883', 'Bangalore', '2023-02-22'),

(104, 'Pooja', 'Patel', '8888888884', 'Ahmedabad', '2022-10-19'),

(105, 'Ravi', 'Kumar', '8888888885', 'Chennai', '2023-03-10'),

(106, 'Ananya', 'Singh', '8888888886', 'Kolkata', '2023-01-30'),

(107, 'Vishal', 'Verma', '8888888887', 'Hyderabad', '2022-11-12'),

(108, 'Simran', 'Joshi', '8888888888', 'Pune', '2023-03-25'),

(109, 'Neha', 'Mehta', '8888888889', 'Lucknow', '2023-04-01'),

(110, 'Amit', 'Bansal', '8888888890', 'Jaipur', '2023-02-14');

3). INSERT INTO Rides (RideID, RiderID, DriverID, RideDate, PickupLocation, DropLocation, Distance, Fare, RideStatus)

VALUES

(11, 101, 2, '2023-04-01 08:30:00', 'Connaught Place, Delhi', 'Indira Gandhi Airport, Delhi', 15, 300, 'Completed'),

(22, 102, 3, '2023-04-02 09:00:00', 'Colaba, Mumbai', 'Bandra, Mumbai', 10, 250, 'Completed'),

(33, 103, 4, '2023-04-03 10:15:00', 'MG Road, Bangalore', 'Kormanagala, Bangalore', 12, 275, 'Completed'),

(44, 104, 5, '2023-04-04 12:30:00', 'Ellis Bridge, Ahmedabad', 'Sarkhej, Ahmedabad', 18, 350, 'Cancelled'),

(55, 105, 6, '2023-04-05 14:00:00', 'T Nagar, Chennai', 'Mylapore, Chennai', 8, 200, 'Completed'),

(66, 106, 7, '2023-04-06 16:45:00', 'Salt Lake, Kolkata', 'Park Street, Kolkata', 6, 150, 'Ongoing'),

(77, 107, 8, '2023-04-07 18:00:00', 'Gachibowli, Hyderabad', 'Banjara Hills, Hyderabad', 14, 280, 'Completed'),

(88, 108, 9, '2023-04-08 09:30:00', 'Aundh, Pune', 'Koregaon Park, Pune', 20, 400, 'Completed'),

(99, 109, 10, '2023-04-09 11:15:00', 'Hazratganj, Lucknow', 'Alambagh, Lucknow', 10, 225, 'Ongoing'),

(100, 110, 1, '2023-04-10 07:30:00', 'Jaipur Railway Station, Jaipur', 'Hawa Mahal, Jaipur', 5, 150, 'Cancelled');

4). INSERT INTO `payments` (`PaymentID`, `RideID`, `PaymentMethod`, `Amount`, `PaymentDate`) VALUES

(1, 11, 'Card', 300.00, '2023-04-01 09:00:00'),

(2, 22, 'Cash', 250.00, '2023-04-02 09:30:00'),

(3, 33, 'Wallet', 275.00, '2023-04-03 10:30:00'),

(4, 44, 'Card', 350.00, '2023-04-04 13:00:00'),

(5, 55, 'Cash', 200.00, '2023-04-05 14:30:00'),

(6, 66, 'Wallet', 150.00, '2023-04-06 17:00:00'),

(7, 77, 'Card', 280.00, '2023-04-07 18:30:00'),

(8, 88, 'Cash', 400.00, '2023-04-08 10:00:00'),

(9, 99, 'Card', 225.00, '2023-04-09 12:00:00'),

(10, 100, 'Wallet', 150.00, '2023-04-10 08:00:00');

// 10 queries:

1. Retrieve the names and contact details of all drivers with a rating of 4.5 or higher.

SELECT FirstName, LastName, Phone, City, VehicleType, Rating

FROM Drivers

WHERE Rating >= 4.5;

2. Find the total number of rides completed by each driver.

SELECT d.DriverID, d.FirstName, d.LastName, COUNT(r.RideID) AS TotalRides

FROM Drivers d

JOIN Rides r ON d.DriverID = r.DriverID

WHERE r.RideStatus = 'Completed'

GROUP BY d.DriverID;

3. List all riders who have never booked a ride.

SELECT r.FirstName, r.LastName, r.Phone, r.City

FROM Riders r

LEFT JOIN Rides ri ON r.RiderID = ri.RiderID

WHERE ri.RideID IS NULL;

4. Calculate the total earnings of each driver from completed rides.

SELECT d.DriverID, d.FirstName, d.LastName, SUM(p.Amount) AS TotalEarnings

FROM Drivers d

JOIN Rides r ON d.DriverID = r.DriverID

JOIN Payments p ON r.RideID = p.RideID

WHERE r.RideStatus = 'Completed'

GROUP BY d.DriverID;

5. Retrieve the most recent ride for each rider.

SELECT r.RiderID, r.FirstName, r.LastName, ri.RideID, ri.RideDate,ri.PickupLocation, ri.DropLocation

FROM Riders r

JOIN Rides ri ON r.RiderID = ri.RiderID

WHERE ri.RideDate = (SELECT MAX(RideDate) FROM Rides WHERE RiderID = r.RiderID);

6. Count the number of rides taken in each city.

SELECT r.City, COUNT(ri.RideID) AS NumberOfRides

FROM Riders r

JOIN Rides ri ON r.RiderID = ri.RiderID

GROUP BY r.City;

7. List all rides where the distance was greater than 20 km.

SELECT ri.RideID, ri.RiderID, ri.DriverID, ri.RideDate, ri.Distance

FROM Rides ri

WHERE ri.Distance > 20;

8. Identify the most preferred payment method.

SELECT p.PaymentMethod, COUNT(p.PaymentID) AS PaymentCount

FROM Payments p

GROUP BY p.PaymentMethod

ORDER BY PaymentCount DESC

LIMIT 1;

9. Find the top 3 highest-earning drivers.

SELECT d.DriverID, d.FirstName, d.LastName, SUM(p.Amount) AS TotalEarnings

FROM Drivers d

JOIN Rides r ON d.DriverID = r.DriverID

JOIN Payments p ON r.RideID = p.RideID

WHERE r.RideStatus = 'Completed'

GROUP BY d.DriverID

ORDER BY TotalEarnings DESC

LIMIT 3;

10. Retrieve details of all cancelled rides along with the rider's and driver's names.

SELECT ri.RideID, r.FirstName AS RiderFirstName, r.LastName AS RiderLastName,

d.FirstName AS DriverFirstName, d.LastName AS DriverLastName, ri.RideDate, ri.PickupLocation, ri.DropLocation

FROM Rides ri

JOIN Riders r ON ri.RiderID = r.RiderID

JOIN Drivers d ON ri.DriverID = d.DriverID

WHERE ri.RideStatus = 'Cancelled';