**OLA Ride Database**

1. **CREATE TABLE DRIVERS**

CREATE TABLE Drivers (

DriverID INT PRIMARY KEY AUTO\_INCRIMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15) NOT NULL,

City VARCHAR(50) NOT NULL,

VehicleType VARCHAR(20) NOT NULL,

Rating DECIMAL(2,1) NOT NULL

);

1. **CREATE TABLE RIDERS**

CREATE TABLE Riders (

RiderID INT PRIMARY KEY AUTO\_INCRIMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15) NOT NULL,

City VARCHAR(50) NOT NULL,

JoinDate DATE NOT NULL

);

1. **CREATE TABLE RIDES**

CREATE TABLE Rides (

RideID INT PRIMARY KEY AUTO\_INCRIMENT,

RiderID INT NOT NULL,

DriverID INT NOT NULL,

RideDate DATE NOT NULL,

PickupLocation VARCHAR(100) NOT NULL,

DropLocation VARCHAR(100) NOT NULL,

Distance DECIMAL(5,2) NOT NULL,

Fare DECIMAL(10,2) NOT NULL,

RideStatus VARCHAR(20) NOT NULL,

FOREIGN KEY (RiderID) REFERENCES Riders(RiderID),

FOREIGN KEY (DriverID) REFERENCES Drivers(DriverID)

);

1. **CREATE TABLE PAYMENTS**

CREATE TABLE Payments (

PaymentID INT PRIMARY KEY AUTO\_INCRIMENT,

RideID INT NOT NULL,

PaymentMethod VARCHAR(20) NOT NULL,

Amount DECIMAL(10,2) NOT NULL,

PaymentDate DATE NOT NULL,

FOREIGN KEY (RideID) REFERENCES Rides(RideID)

);

5. Insert Sample Data (Indian Data)

INSERT INTO Drivers VALUES

(1, 'Rajesh', 'Kumar', '9876543210', 'Mumbai', 'Sedan', 4.7),

(2, 'Anil', 'Sharma', '9876543222', 'Delhi', 'SUV', 4.8),

(3, 'Sunita', 'Verma', '9876543233', 'Bangalore', 'Hatchback', 4.3);

INSERT INTO Riders VALUES

(1, 'Ravi', 'Patil', '9876543244', 'Mumbai', '2023-01-01'),

(2, 'Priya', 'Mehta', '9876543255', 'Delhi', '2023-02-01'),

(3, 'Arjun', 'Das', '9876543266', 'Chennai', '2023-03-01');

(4, 'Yogendra, 'Patidar, '9876543266', 'Nimatch, '2023-03-01');

(5, 'Sarthak, 'Sharma, '9876543266', 'Sagar', '2023-03-01');

INSERT INTO Rides VALUES

(1, 1, 1, '2023-04-01', 'Andheri', 'Bandra', 12.5, 250.00, 'Completed'),

(2, 2, 2, '2023-04-02', 'Connaught Place', 'Saket', 18.0, 400.00, 'Completed'),

(3, 3, 3, '2023-04-03', 'MG Road', 'Whitefield', 25.0, 600.00, 'Cancelled');

(4, 5, 1, '2023-04-01', 'Andheri', 'Bandra', 23.5, 320.00, 'Completed'),

INSERT INTO Payments VALUES

(1, 1, 'Card', 250.00, '2023-04-01'),

(2, 2, 'Cash', 400.00, '2023-04-02');

(3, 4, 'Cash', 320.00, '2023-04-02');

**QUERIES**

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

🡺SELECT firstName , lastName,rating from drivers WHERE rating > 4.5;

1. **Find the total number of rides completed by each driver.**

🡺 SELECT drivers.FirstName , drivers.LastName ,COUNT(rides.RideID) as total\_rides from rides JOIN drivers on drivers.DriverID = rides.DriverID

GROUP by rides.RideID

1. **List all riders who have never booked a ride.**

**🡺** SELECT riders.FirstName,riders.LastName FROM riders left JOIN rides on rides.RiderID = riders.RiderID WHERE rides.RiderID is null

1. **Calculate the total earnings of each driver from completed rides.**

SELECT drivers.FirstName, drivers.LastName,COUNT(rides.RideID) FROM drivers JOIN rides on drivers.DriverID = rides.DriverID WHERE rides.RideStatus = 'Completed' GROUP by drivers.FirstName;

1. **Retrieve the most recent ride for each rider**

SELECt rides.RideDate ,max(rides.RideID)as RecentRide from rides

1. **Count the number of rides taken in each city**.

SELECT drivers.City, COUNT(rides.RideID) from rides JOIN drivers on rides.DriverID = drivers.DriverID GROUP by drivers.City

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

SELECT riders.FirstName , riders.LastName , rides.Distance FROM riders JOIN rides on rides.RiderID = riders.RiderID WHERE rides.Distance > 20

1. **Identify the most preferred payment method**.

SELECT payments.PaymentMethod ,COUNT(payments.PaymentMethod) as maxPayment FROM payments JOIN rides ON rides.RideID = payments.RideID GROUP by payments.PaymentMethod DESC

1. **Find the top 3 highest-earning drivers.**

SELECT max(rides.Fare),COUNT(rides.RideID) FROM rides JOIN drivers on drivers.DriverID = rides.DriverID GROUP by drivers.DriverID DESC

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

SELECT riders.FirstName,riders.LastName,riders.Phone,riders.City,rides.Fare,rides.Distance,rides.RideStatus from riders JOIN rides on rides.RideID = riders.RiderID WHERE rides.RideStatus = 'Cancelled'