

CPS 510 - Assignment 4

Shayaan Kirubakaran (Section 5)

Kirusanth Palakanthan (Section 10)

Ahmed Hasan (Section 5)

Assignment 4 Part 1:*RidePickUpQueries.sql*

```
-- CPS510 – Assignment 4 (Part 1)
-- Ride & Pickup DBMS – Queries
-- Authors: Shayaan Kirubakaran, Kirusanth Palakanthan, Ahmed Hasan
```

```
-- Q1. Address Table: Toronto addresses
```

```
SELECT Address_ID, Street_Address AS Street, City, Province, Postal_Code
FROM Address
WHERE City = 'Toronto';
```

```
-- Q2. Customer Table: Balances with status
```

```
SELECT Customer_ID,
       First_Name || ' ' || Last_Name AS Customer_Name,
       Balance,
       CASE WHEN Balance > 0 THEN 'Active' ELSE 'Inactive' END AS Status
FROM Customer
ORDER BY Balance DESC;
```

```
-- Q3. Driver Table: With insurance info
```

```
SELECT Driver_ID,
       First_Name || ' ' || Last_Name AS Driver_Name,
       License_Info,
       Insurance_Info
FROM Driver
WHERE Insurance_Info IS NOT NULL;
```

```
-- Q4. Vehicle Table: Manufactured after 2015
```

```
SELECT Vehicle_ID, License_Plate, Make, Model, Year, Vehicle_Type
FROM Vehicle
WHERE Year > 2015;
```

```
-- Q5. Merchant Table: Merchants and their city
```

```
SELECT DISTINCT M.Merchant_ID, M.Name AS Merchant_Name, A.City, A.Province
FROM Merchant M, Address A
WHERE M.Address_ID = A.Address_ID;
```

```
-- Q6. Location Table: Ontario locations
```

```
SELECT Location_ID, Street_Address, City, Province, Postal_Code
FROM Location
WHERE Province = 'Ontario';
```

```
-- Q7. Service Orders: Completed > $20
```

```
SELECT Order_ID, Customer_ID, Driver_ID, Fare, Status
FROM Service_Order
WHERE Status = 'Completed'
AND Fare > 20;
```

```
-- Q8. Payment Table: Count of credit card payments
```

```
SELECT COUNT(*) AS Credit_Payments
FROM Payment
WHERE Payment_Type = 'Credit';
```

```
-- Q9. Ratings: Low (< 3 stars)
```

```
SELECT Rating_ID, Order_ID, Customer_Stars, Customer_Feedback
FROM Rating_System
WHERE Customer_Stars < 3;
```

```
-- Q10. Join: Completed orders with customer & driver names
```

```
SELECT so.Order_ID,
       c.First_Name || ' ' || c.Last_Name AS Customer_Name,
       d.First_Name || ' ' || d.Last_Name AS Driver_Name,
       so.Fare
  FROM Service_Order so, Customer c, Driver d
 WHERE so.Customer_ID = c.Customer_ID
   AND so.Driver_ID = d.Driver_ID
```

```
AND so.Status = 'Completed'  
ORDER BY so.Fare DESC;
```

```
-- Q11. Join: Payments with order info & customer name
```

```
SELECT p.Payment_ID, p.Payment_Type, p.Amount,  
    so.Order_ID,  
    c.First_Name || ' ' || c.Last_Name AS Customer_Name  
FROM Payment p, Service_Order so, Customer c  
WHERE p.Order_ID = so.Order_ID  
AND so.Customer_ID = c.Customer_ID  
ORDER BY p.Amount DESC;
```

```
-- Q12. Aggregation: Total orders per customer
```

```
SELECT c.Customer_ID,  
    c.First_Name || ' ' || c.Last_Name AS Customer_Name,  
    COUNT(so.Order_ID) AS Total_Orders  
FROM Customer c LEFT JOIN Service_Order so  
    ON c.Customer_ID = so.Customer_ID  
GROUP BY c.Customer_ID, c.First_Name || ' ' || c.Last_Name  
ORDER BY Total_Orders DESC;
```

```
-- Q13. Aggregation: Average fare per driver
```

```
SELECT d.Driver_ID,  
    d.First_Name || ' ' || d.Last_Name AS Driver_Name,  
    ROUND(AVG(so.Fare),2) AS Avg_Fare  
FROM Driver d JOIN Service_Order so  
    ON d.Driver_ID = so.Driver_ID  
WHERE so.Status = 'Completed'  
GROUP BY d.Driver_ID, d.First_Name || ' ' || d.Last_Name  
ORDER BY Avg_Fare DESC;
```

RidePickUpQueries_Data.sql

```
-- CPS510 – Assignment 4 (Part 1)
-- Ride & Pickup DBMS – Sample Data Inserts
-- Authors: Shayaan Kirubakaran, Kirusanth Palakanthan, Ahmed Hasan
```

```
-- Address Data
```

```
INSERT INTO Address VALUES (1, 'Canada', 'Ontario', 'Toronto', '123 King St', 'M5G1X5');
INSERT INTO Address VALUES (2, 'Canada', 'Ontario', 'Mississauga', '50 Lakeshore Rd',
'L5E2N2');
INSERT INTO Address VALUES (3, 'Canada', 'Ontario', 'Toronto', '321 Front St', 'M5V2T3');
INSERT INTO Address VALUES (4, 'Canada', 'Ontario', 'Oakville', '123 Oak St', 'L6J4L4');
```

```
-- Location Data
```

```
INSERT INTO Location VALUES (1, '456 Queen St', 'Toronto', 'Ontario', 'M5H2N2', 'Canada',
NULL);
INSERT INTO Location VALUES (2, '789 Bay St', 'Toronto', 'Ontario', 'M5B2C5', 'Canada',
NULL);
INSERT INTO Location VALUES (3, '200 Dundas St', 'Mississauga', 'Ontario', 'L5A1X2',
'Canada', NULL);
INSERT INTO Location VALUES (4, '120 Lakeshore Rd', 'Mississauga', 'Ontario', 'L5G4G2',
'Canada', NULL);
```

```
-- Customer Data
```

```
INSERT INTO Customer VALUES (1, 'John', 'Doe', '4161234567', 'john.doe@email.com',
TO_DATE('1990-05-20','YYYY-MM-DD'), SYSDATE, 50.00, 1);
INSERT INTO Customer VALUES (2, 'Alice', 'Wong', '4165551234', 'alice.wong@email.com',
TO_DATE('1992-07-15','YYYY-MM-DD'), SYSDATE, 100.00, 2);
INSERT INTO Customer VALUES (3, 'Bob', 'Singh', '6471112222', 'bob.singh@email.com',
TO_DATE('1988-09-10','YYYY-MM-DD'), SYSDATE, 0.00, 3);
INSERT INTO Customer VALUES (4, 'Mia', 'Kim', '6471234567', 'mia.kim@email.com',
TO_DATE('1995-04-25','YYYY-MM-DD'), SYSDATE, 75.00, 4);
```

```
-- Driver Data
```

```
INSERT INTO Driver VALUES (1, 'Jane', 'Smith', '4169876543',
TO_DATE('1985-10-10','YYYY-MM-DD'),
'LIC12345', 'InsureCo #123', 1);
INSERT INTO Driver VALUES (2, 'Mark', 'Brown', '6472229999',
TO_DATE('1990-03-12','YYYY-MM-DD'),
'LIC54321', NULL, 2);
INSERT INTO Driver VALUES (3, 'Ravi', 'Kumar', '9057778888',
TO_DATE('1982-01-25','YYYY-MM-DD'),
'LIC98765', 'SafeDrive #456', 3);
INSERT INTO Driver VALUES (4, 'Sarah', 'Jones', '4166667777',
TO_DATE('1989-11-14','YYYY-MM-DD'),
'LIC24680', 'BestDrive #789', 4);
```

-- Vehicle Data

```
INSERT INTO Vehicle VALUES (1, 'ABC123', 'Toyota', 'Camry', 'Black', 2020, 'Sedan', 1);
INSERT INTO Vehicle VALUES (2, 'XYZ987', 'Honda', 'Civic', 'Blue', 2018, 'Sedan', 2);
INSERT INTO Vehicle VALUES (3, 'LMN456', 'Ford', 'Escape', 'White', 2022, 'SUV', 3);
INSERT INTO Vehicle VALUES (4, 'DEF567', 'Nissan', 'Altima', 'Silver', 2021, 'Sedan', 4);
```

-- Merchant Data

```
INSERT INTO Merchant VALUES (1, 'Pizza Place', 1);
INSERT INTO Merchant VALUES (2, 'Coffee Corner', 2);
INSERT INTO Merchant VALUES (3, 'Sushi Express', 3);
INSERT INTO Merchant VALUES (4, 'Burger King', 4);
```

-- Service Orders

-- Completed order (Customer 1, Driver 1)

```
INSERT INTO Service_Order VALUES (1, 1, 1, 1, 'Completed', CURRENT_TIMESTAMP, 25.50,
1, 2);
```

-- Pending order (Customer 2, Driver 2)

```
INSERT INTO Service_Order VALUES (2, 2, 2, 2, 'Pending', CURRENT_TIMESTAMP, 12.00,
2, 1);
```

-- Cancelled order (Customer 3, Driver 3)

```
INSERT INTO Service_Order VALUES (3, 3, 3, 3, 'Cancelled', CURRENT_TIMESTAMP, 30.00,
3, 1);
```

```
-- Completed order (Customer 2, Driver 1)
INSERT INTO Service_Order VALUES (4, 2, 1, 1, 'Completed', CURRENT_TIMESTAMP, 45.00,
1, 3);
```

```
-- Completed order (Customer 4, Driver 4)
INSERT INTO Service_Order VALUES (5, 4, 4, 4, 'Completed', CURRENT_TIMESTAMP, 40.00,
4, 2);
```

-- Payments

```
-- Payment for order 1
```

```
INSERT INTO Payment VALUES (1, 1, 'Credit', 25.50, 'Paid');
```

```
-- Payment for order 2
```

```
INSERT INTO Payment VALUES (2, 2, 'Debit', 12.00, 'Pending');
```

```
-- Payment for order 3
```

```
INSERT INTO Payment VALUES (3, 3, 'Balance', 30.00, 'Failed');
```

```
-- Payment for order 4
```

```
INSERT INTO Payment VALUES (4, 4, 'Credit', 45.00, 'Paid');
```

```
-- Payment for order 5
```

```
INSERT INTO Payment VALUES (5, 5, 'Debit', 40.00, 'Paid');
```

-- Ratings

```
INSERT INTO Rating_System VALUES (1, 1, 1, 1, 5, 'Great driver!', 5, 'Polite customer');
```

```
INSERT INTO Rating_System VALUES (2, 2, 2, 2, 2, 'Driver was late', 3, 'Customer was fine');
```

```
INSERT INTO Rating_System VALUES (3, 3, 3, 3, 3, 'Trip cancelled, no feedback', 3, 'No show');
```

```
INSERT INTO Rating_System VALUES (4, 4, 2, 1, 4, 'Fast delivery!', 5, 'Great customer');
```

```
INSERT INTO Rating_System VALUES (5, 5, 4, 4, 5, 'Awesome ride!', 5, 'Friendly driver');
```

```
COMMIT;
```

Assignment 4 Part 1 Output:

ADDRESS_ID STREET	CITY	PROVINCE	POSTAL_CODE
1 123 King St 3 321 Front St	Toronto Toronto	Ontario Ontario	M5G1X5 MSV2T3
CUSTOMER_ID CUSTOMER_NAME	BALANCE STATUS		
2 Alice Wong 4 Mia Kim 1 John Doe 3 Bob Singh	100 Active 75 Active 50 Active 0 Inactive		
DRIVER_ID DRIVER_NAME	LICENSE_INFO	INSURANCE_INFO	
1 Jane Smith 3 Ravi Kumar 4 Sarah Jones	LIC12345 LIC98765 LIC24680	InsureCo #123 SafeDrive #456 BestDrive #789	
VEHICLE_ID LICENSE_PLATE	MAKE	MODEL	YEAR VEHICLE_TYPE
1 ABC123 2 XYZ987 3 LMN456 4 DEF567	Toyota Honda Ford Nissan	Camry Civic Escape Altima	2020 Sedan 2018 Sedan 2022 SUV 2021 Sedan
MERCHANT_ID MERCHANT_NAME	CITY	PROVINCE	
4 Burger King 2 Coffee Corner 3 Sushi Express 1 Pizza Place	Oakville Mississauga Toronto Toronto	Ontario Ontario Ontario Ontario	M5R2N2 M5B2C5 L5A1X2 L5G4G2
LOCATION_ID STREET_ADDRESS	CITY	PROVINCE	POSTAL_CODE
1 456 Queen St 2 789 Bay St 3 200 Dundas St 4 120 Lakeshore Rd	Toronto Toronto Mississauga Mississauga	Ontario Ontario Ontario Ontario	M5R2N2 M5B2C5 L5A1X2 L5G4G2
ORDER_ID CUSTOMER_ID DRIVER_ID FARE STATUS			
1 1 1 25.5 Completed 4 2 1 45 Completed 5 4 4 40 Completed			
CREDIT_PAYMENTS			
2			
RATING_ID ORDER_ID CUSTOMER_STARS CUSTOMER_FEEDBACK			
2 2 2 Driver was late			
ORDER_ID CUSTOMER_NAME	DRIVER_NAME		FARE
4 Alice Wong 5 Mia Kim 1 John Doe	Jane Smith Sarah Jones Jane Smith		45 40 25.5
PAYOUT_ID PAYMENT_TYPE	AMOUNT	ORDER_ID CUSTOMER_NAME	
4 Credit 5 Debit 3 Balance 1 Credit 2 Debit	45 40 50 25.5 12	4 Alice Wong 5 Mia Kim 3 Bob Singh 1 John Doe 2 Alice Wong	
CUSTOMER_ID CUSTOMER_NAME	TOTAL ORDERS		
2 Alice Wong 4 Mia Kim 1 John Doe 3 Bob Singh	2 1 1 1		
DRIVER_ID DRIVER_NAME	AVG_FARE		
4 Sarah Jones 1 Jane Smith	40 35.25		
ADDRESS_ID STREET	CITY	PROVINCE	POSTAL_CODE
1 123 King St 3 321 Front St	Toronto Toronto	Ontario Ontario	M5G1X5 MSV2T3
CUSTOMER_ID CUSTOMER_NAME	BALANCE STATUS		
2 Alice Wong 4 Mia Kim 1 John Doe 3 Bob Singh	100 Active 75 Active 50 Active 0 Inactive		
DRIVER_ID DRIVER_NAME	LICENSE_INFO	INSURANCE_INFO	
1 Jane Smith 3 Ravi Kumar 4 Sarah Jones	LIC12345 LIC98765 LIC24680	InsureCo #123 SafeDrive #456 BestDrive #789	
VEHICLE_ID LICENSE_PLATE	MAKE	MODEL	YEAR VEHICLE_TYPE
1 ABC123 2 XYZ987 3 LMN456 4 DEF567	Toyota Honda Ford Nissan	Camry Civic Escape Altima	2020 Sedan 2018 Sedan 2022 SUV 2021 Sedan

MERCHANT_ID	MERCHANT_NAME	CITY	PROVINCE		
4	Burger King	Oakville	Ontario		
2	Coffee Corner	Mississauga	Ontario		
3	Sushi Express	Toronto	Ontario		
1	Pizza Place	Toronto	Ontario		
LOCATION_ID	STREET_ADDRESS	CITY	PROVINCE	POSTAL_CODE	
1	156 Queen St	Toronto	Ontario	M5H2M2	
2	789 Bay St	Toronto	Ontario	M5B3C5	
3	200 Dundas St	Mississauga	Ontario	L5R1X2	
4	120 Lakeshore Rd	Mississauga	Ontario	L5G4E2	
ORDER_ID	CUSTOMER_ID	DRIVER_ID	FARE STATUS		
1	1	1	25.5 Completed		
4	2	1	45 Completed		
5	4	4	40 Completed		
CREDIT_PAYMENTS					
	2				
RATING_ID	ORDER_ID	CUSTOMER_STARS	CUSTOMER_FEEDBACK		
2	2	2	Driver was late		
ORDER_ID	CUSTOMER_NAME	DRIVER_NAME	FARE		
4	Alice Wong	Jane Smith	45		
5	Mia Kim	Sarah Jones	40		
1	John Doe	Jane Smith	25.5		
PAYMENT_ID	PAYMENT_TYPE	AMOUNT	ORDER_ID	CUSTOMER_NAME	
4	Credit	45	4	Alice Wong	
5	Debit	40	5	Mia Kim	
3	Balance	30	3	Bob Singh	
1	Credit	25.5	1	John Doe	
2	Debit	12	2	Alice Wong	
CUSTOMER_ID	CUSTOMER_NAME	TOTAL_ORDERS			
2	Alice Wong	2			
4	Mia Kim	1			
1	John Doe	1			
3	Bob Singh	1			
DRIVER_ID	DRIVER_NAME	AVG_FARE			
4	Sarah Jones	40			
1	Jane Smith	35.25			

```
-- CPS510 – Assignment 4 (Part 2)
-- Ride & Pickup DBMS – 2 Views, Join Queries, Advanced Queries
-- Authors: Shayaan Kirubakaran, Kirusanth Palakanthan, Ahmed Hasan
```

```
-- ====== VIEWS ======
```

```
-- ======
```

```
-- VIEW 1: Active Customer Orders
```

```
-- Shows all active orders for customers with status 'Pending' or 'In Progress'
```

```
CREATE OR REPLACE VIEW Active_Customer_Orders AS
```

```
SELECT DISTINCT c.Customer_ID,
```

```
    c.First_Name || ' ' || c.Last_Name AS Customer_Name,
```

```
    so.Order_ID,
```

```
    so.Status,
```

```
    so.Fare
```

```
FROM Customer c
```

```
JOIN Service_Order so ON c.Customer_ID = so.Customer_ID
```

```
WHERE so.Status IN ('Pending', 'In Progress');
```

```
-- Display view results
```

```
SELECT * FROM Active_Customer_Orders;
```

```
-- ======
```

```
-- VIEW 2: Driver Earnings Summary
```

```
-- Shows total earnings for each driver from completed orders
```

```
CREATE OR REPLACE VIEW Driver_Earnings AS
```

```
SELECT d.Driver_ID,
```

```
    d.First_Name || ' ' || d.Last_Name AS Driver_Name,
```

```
    SUM(so.Fare) AS Total_Earnings
```

```
FROM Driver d
```

```
JOIN Service_Order so ON d.Driver_ID = so.Driver_ID
```

```
JOIN Payment p ON so.Order_ID = p.Order_ID
```

```
WHERE so.Status = 'Completed'
```

```
GROUP BY d.Driver_ID, d.First_Name, d.Last_Name
```

```
ORDER BY Total_Earnings DESC;
```

```
-- Display view results
```

```
SELECT * FROM Driver_Earnings;
```

```
-- ===== JOIN QUERIES =====
```

```
-- =====  
-- Join Query 1: Customer Feedback on Completed Orders  
-- Shows customer ratings and feedback with driver details  
-- =====
```

```
SELECT DISTINCT c.Customer_ID,  
    c.First_Name || ' ' || c.Last_Name AS Customer_Name,  
    so.Order_ID,  
    rs.Customer_Stars AS Customer_Rating,  
    rs.Customer_Feedback,  
    d.First_Name || ' ' || d.Last_Name AS Driver_Name  
FROM Service_Order so  
JOIN Customer c ON so.Customer_ID = c.Customer_ID  
JOIN Rating_System rs ON so.Order_ID = rs.Order_ID  
JOIN Driver d ON so.Driver_ID = d.Driver_ID  
WHERE so.Status = 'Completed'  
ORDER BY so.Order_ID;
```

```
-- =====  
-- Join Query 2: Merchants and Their Delivery Orders  
-- Shows which merchants fulfilled which orders with city and province  
-- =====
```

```
SELECT DISTINCT m.Merchant_ID,  
    m.Name AS Merchant_Name,  
    so.Order_ID,  
    so.Status  
FROM Merchant m  
JOIN Service_Order so ON m.Merchant_ID = so.Merchant_ID  
WHERE so.Status IN ('Completed', 'In Progress')  
ORDER BY m.Merchant_ID;
```

```
-- =====  
-- Join Query 3: Completed Orders with Customer & Driver Info  
-- =====
```

```
SELECT so.Order_ID,  
    c.First_Name || ' ' || c.Last_Name AS Customer_Name,  
    d.First_Name || ' ' || d.Last_Name AS Driver_Name,  
    so.Fare  
FROM Service_Order so  
JOIN Customer c ON so.Customer_ID = c.Customer_ID  
JOIN Driver d ON so.Driver_ID = d.Driver_ID
```

```
WHERE so.Status = 'Completed'  
ORDER BY so.Fare DESC;
```

```
-- ===== ADVANCED / AGGREGATION QUERIES =====
```

```
-- =====  
-- Advanced Query 1: Total Orders by Customer  
-- =====  
SELECT c.Customer_ID,  
       c.First_Name || ' ' || c.Last_Name AS Customer_Name,  
       COUNT(so.Order_ID) AS Total_Orders  
FROM Customer c  
LEFT JOIN Service_Order so ON c.Customer_ID = so.Customer_ID  
GROUP BY c.Customer_ID, c.First_Name, c.Last_Name  
ORDER BY Total_Orders DESC;
```

```
-- =====  
-- Advanced Query 2: Average Fare per Driver  
-- =====  
SELECT d.Driver_ID,  
       d.First_Name || ' ' || d.Last_Name AS Driver_Name,  
       ROUND(AVG(so.Fare), 2) AS Avg_Fare  
FROM Driver d  
JOIN Service_Order so ON d.Driver_ID = so.Driver_ID  
WHERE so.Status = 'Completed'  
GROUP BY d.Driver_ID, d.First_Name, d.Last_Name  
ORDER BY Avg_Fare DESC;
```

```
-- =====  
-- Advanced Query 3: Total Earnings per Merchant  
-- =====  
SELECT m.Merchant_ID,  
       m.Name AS Merchant_Name,  
       SUM(so.Fare) AS Total_Earnings  
FROM Merchant m  
JOIN Service_Order so ON m.Merchant_ID = so.Merchant_ID  
WHERE so.Status = 'Completed'  
GROUP BY m.Merchant_ID, m.Name  
ORDER BY Total_Earnings DESC;
```

```
-- =====  
-- Advanced Query 4: Service Orders with Customer, Driver, Payment Info
```

```
-- =====
SELECT so.Order_ID,
       c.First_Name || ' ' || c.Last_Name AS Customer_Name,
       d.First_Name || ' ' || d.Last_Name AS Driver_Name,
       p.Amount AS Payment_Amount,
       so.Fare AS Service_Fare,
       so.Status AS Order_Status
  FROM Service_Order so
 JOIN Customer c ON so.Customer_ID = c.Customer_ID
 JOIN Driver d ON so.Driver_ID = d.Driver_ID
 JOIN Payment p ON so.Order_ID = p.Order_ID
 WHERE so.Status = 'Completed'
 ORDER BY so.Order_ID;
```

```
-- =====
-- Advanced Query 5: Total and Average Fare per Customer
-- =====
SELECT c.Customer_ID,
       c.First_Name || ' ' || c.Last_Name AS Customer_Name,
       COUNT(so.Order_ID) AS Total_Orders,
       SUM(so.Fare) AS Total_Fare,
       ROUND(AVG(so.Fare), 2) AS Avg_Fare
  FROM Customer c
 JOIN Service_Order so ON c.Customer_ID = so.Customer_ID
 WHERE so.Status = 'Completed'
 GROUP BY c.Customer_ID, c.First_Name, c.Last_Name
 ORDER BY Total_Fare DESC;
```

```
-- =====
-- Advanced Query 6: Driver's Completed Orders and Total Earnings
-- =====
SELECT d.Driver_ID,
       d.First_Name || ' ' || d.Last_Name AS Driver_Name,
       COUNT(so.Order_ID) AS Total_Completed_Orders,
       SUM(so.Fare) AS Total_Earnings
  FROM Driver d
 JOIN Service_Order so ON d.Driver_ID = so.Driver_ID
 WHERE so.Status = 'Completed'
 GROUP BY d.Driver_ID, d.First_Name, d.Last_Name
 ORDER BY Total_Earnings DESC;
```

Assignment 4 Part 2 Views Output Screenshot:

Columns Data Grants Dependencies Details Triggers SQL Errors

Actions...    

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	CUSTOMER_ID	NUMBER	No	(null)	1	(null)	NO	NO	NO
2	CUSTOMER_NAME	VARCHAR2(101)	Yes	(null)	2	(null)	NO	NO	NO
3	ORDER_ID	NUMBER	No	(null)	3	(null)	YES	YES	YES
4	STATUS	VARCHAR2(20)	Yes	(null)	4	(null)	YES	YES	YES
5	FARE	NUMBER(10,2)	Yes	(null)	5	(null)	YES	YES	YES

assignment_4_part2.sql ridepickup_data.sql DRIVER_EARNINGS

Columns Data Grants Dependencies Details Triggers SQL Errors

Actions...    

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	INSERTABLE	UPDATABLE	DELETABLE
1	DRIVER_ID	NUMBER	No	(null)	1	(null)	NO	NO	NO
2	DRIVER_NAME	VARCHAR2(101)	Yes	(null)	2	(null)	NO	NO	NO
3	TOTAL_EARNINGS	NUMBER	Yes	(null)	3	(null)	NO	NO	NO

