ADBMS PROJECT E281

-- Table for Users (Drivers)

CREATE TABLE Users (

User\_ID INT PRIMARY KEY,

Username VARCHAR(255),

Password VARCHAR(255)

);

-- Table for Vehicles

CREATE TABLE Vehicles (

Vehicle\_ID INT PRIMARY KEY,

License\_Plate VARCHAR(15),

User\_ID INT

);

-- Table for Delivery Locations

CREATE TABLE DeliveryLocations (

Location\_ID INT PRIMARY KEY,

Name VARCHAR(255),

Latitude DECIMAL(9, 6),

Longitude DECIMAL(9, 6)

);

-- Table for Geofences

CREATE TABLE Geofences (

Geofence\_ID INT PRIMARY KEY,

Name VARCHAR(255)

);

-- Table for Routes

CREATE TABLE Routes (

Route\_ID INT PRIMARY KEY,

Name VARCHAR(255)

);

-- Table to Associate Delivery Locations with Routes

CREATE TABLE RouteDeliveryLocations (

Route\_ID INT,

Location\_ID INT,

PRIMARY KEY (Route\_ID, Location\_ID)

);

-- Table to Track Vehicle Locations

CREATE TABLE VehicleLocations (

Location\_ID INT PRIMARY KEY,

Vehicle\_ID INT,

Latitude DECIMAL(9, 6),

Longitude DECIMAL(9, 6),

Timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO Users (User\_ID, Username, Password)

VALUES (1, 'JohnDoe', 'password123');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (2, 'JaneSmith', 'securepass');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (3, 'MikeJohnson', 'mypass123');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (4, 'EmilyDavis', 'password456');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (5, 'RobertWilson', 'securepass123');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (6, 'LindaBrown', 'lindapass');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (7, 'JamesMiller', 'james123');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (8, 'SophiaClark', 'sophiapass');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (9, 'WilliamSmith', 'william123');

INSERT INTO Users (User\_ID, Username, Password)

VALUES (10, 'OliviaWhite', 'oliviapass');

-- Table for Vehicles

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (1, 'ABC123', 1);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (2, 'XYZ789', 2);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (3, 'LMN456', 3);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (4, 'PQR789', 4);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (5, 'DEF123', 5);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (6, 'GHI456', 6);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (7, 'JKL789', 7);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (8, 'MNO123', 8);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (9, 'UVW456', 9);

INSERT INTO Vehicles (Vehicle\_ID, License\_Plate, User\_ID)

VALUES (10, 'STU789', 10);

-- Table for Delivery Locations

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (1, 'Location A', 40.7128, -74.0060);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (2, 'Location B', 34.0522, -118.2437);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (3, 'Location C', 41.8781, -87.6298);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (4, 'Location D', 51.5074, -0.1278);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (5, 'Location E', 35.6895, 139.6917);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (6, 'Location F', 52.5200, 13.4050);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (7, 'Location G', 48.8566, 2.3522);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (8, 'Location H', 37.7749, -122.4194);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (9, 'Location I', 22.5726, 88.3639);

INSERT INTO DeliveryLocations (Location\_ID, Name, Latitude, Longitude)

VALUES (10, 'Location J', 41.3851, 2.1734);

-- Table for Geofences

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (1, 'Geofence 1');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (2, 'Geofence 2');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (3, 'Geofence 3');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (4, 'Geofence 4');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (5, 'Geofence 5');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (6, 'Geofence 6');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (7, 'Geofence 7');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (8, 'Geofence 8');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (9, 'Geofence 9');

INSERT INTO Geofences (Geofence\_ID, Name)

VALUES (10, 'Geofence 10');

-- Table for Routes

INSERT INTO Routes (Route\_ID, Name)

VALUES (1, 'Route 1');

INSERT INTO Routes (Route\_ID, Name)

VALUES (2, 'Route 2');

INSERT INTO Routes (Route\_ID, Name)

VALUES (3, 'Route 3');

INSERT INTO Routes (Route\_ID, Name)

VALUES (4, 'Route 4');

INSERT INTO Routes (Route\_ID, Name)

VALUES (5, 'Route 5');

INSERT INTO Routes (Route\_ID, Name)

VALUES (6, 'Route 6');

INSERT INTO Routes (Route\_ID, Name)

VALUES (7, 'Route 7');

INSERT INTO Routes (Route\_ID, Name)

VALUES (8, 'Route 8');

INSERT INTO Routes (Route\_ID, Name)

VALUES (9, 'Route 9');

INSERT INTO Routes (Route\_ID, Name)

VALUES (10, 'Route 10');

-- Table to Associate Delivery Locations with Routes

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (1, 1);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (2, 2);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (3, 3);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (4, 4);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (5, 5);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (6, 6);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (7, 7);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (8, 8);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (9, 9);

INSERT INTO RouteDeliveryLocations (Route\_ID, Location\_ID)

VALUES (10, 10);

-- Table to Track Vehicle Locations

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (1, 1, 40.7128, -74.0060);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (2, 2, 34.0522, -118.2437);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (3, 1, 41.8781, -87.6298);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (4, 3, 51.5074, -0.1278);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (5, 2, 35.6895, 139.6917);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (6, 4, 52.5200, 13.4050);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (7, 3, 48.8566, 2.3522);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (8, 5, 37.7749, -122.4194);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (9, 5, 22.5726, 88.3639);

INSERT INTO VehicleLocations (Location\_ID, Vehicle\_ID, Latitude, Longitude)

VALUES (10, 1, 41.3851, 2.1734);

SELECT \* from Users;

SELECT \* from Vehicles;

SELECT \* from DeliveryLocations;

SELECT \* from Geofences;

SELECT \* from Routes;

SELECT \* from RouteDeliveryLocations;

SELECT \* from VehicleLocations;

-- Create a PL/SQL function to retrieve a username by User\_ID

CREATE OR REPLACE FUNCTION GetUsernameByUserID(p\_UserID IN INT) RETURN VARCHAR2 IS

v\_username VARCHAR2(255);

BEGIN

-- Retrieve the username for the provided User\_ID

SELECT Username INTO v\_username

FROM Users

WHERE User\_ID = p\_UserID;

RETURN v\_username;

END GetUsernameByUserID;

/

-- Create a PL/SQL function to retrieve a username by User\_ID

CREATE OR REPLACE FUNCTION GetUsernameByUserID(p\_UserID IN INT) RETURN VARCHAR2 IS

v\_username VARCHAR2(255);

BEGIN

-- Retrieve the username for the provided User\_ID

SELECT Username INTO v\_username

FROM Users

WHERE User\_ID = p\_UserID;

RETURN v\_username;

END GetUsernameByUserID;

/

CREATE TABLE UserChangeLog (

Log\_ID INT PRIMARY KEY,

Action VARCHAR(255),

User\_ID INT,

ChangeDate TIMESTAMP

);

-- Create a simplified table-level trigger to log any changes in the Users table

CREATE OR REPLACE TRIGGER UserLogTrigger

AFTER INSERT OR UPDATE OR DELETE ON Users

BEGIN

-- Log the action and the affected User\_ID for the entire table

INSERT INTO UserChangeLog (Action, User\_ID, ChangeDate)

SELECT 'CHANGE', User\_ID, SYSDATE FROM Users;

END;

/

-- Create a trigger to display a message for changes in the Users table

CREATE OR REPLACE TRIGGER UserLogTrigger

AFTER INSERT OR UPDATE OR DELETE ON Users

FOR EACH ROW

BEGIN

-- Display a message

DBMS\_OUTPUT.PUT\_LINE('A change has been made to the Users table.');

END;

/

-- Create a trigger to display old user information before an update

CREATE OR REPLACE TRIGGER DisplayOldUsersTrigger

BEFORE UPDATE ON Users

FOR EACH ROW

BEGIN

-- Display the old user information

DBMS\_OUTPUT.PUT\_LINE('Old User\_ID: ' || :old.User\_ID);

DBMS\_OUTPUT.PUT\_LINE('Old Username: ' || :old.Username);

DBMS\_OUTPUT.PUT\_LINE('Old Password: ' || :old.Password);

END;

/

CREATE OR REPLACE PROCEDURE FindUsersWithoutVehicles AS

BEGIN

-- Retrieve users without assigned vehicles

FOR UserRec IN (

SELECT U.User\_ID, U.Username, COUNT(V.Vehicle\_ID) AS Vehicle\_Count

FROM Users U

LEFT JOIN Vehicles V ON U.User\_ID = V.User\_ID

WHERE V.Vehicle\_ID IS NULL

GROUP BY U.User\_ID, U.Username

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('User ID: ' || UserRec.User\_ID);

DBMS\_OUTPUT.PUT\_LINE('Username: ' || UserRec.Username);

DBMS\_OUTPUT.PUT\_LINE('Vehicles Count: ' || UserRec.Vehicle\_Count);

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

END;

/

BEGIN

FindUsersWithoutVehicles;

END;

/

DECLARE

CURSOR UsersWithoutVehicles IS

SELECT U.User\_ID, U.Username, COUNT(V.Vehicle\_ID) AS Vehicle\_Count

FROM Users U

LEFT JOIN Vehicles V ON U.User\_ID = V.User\_ID

WHERE V.Vehicle\_ID IS NULL

GROUP BY U.User\_ID, U.Username;

BEGIN

FOR UserRec IN UsersWithoutVehicles

LOOP

DBMS\_OUTPUT.PUT\_LINE('User ID: ' || UserRec.User\_ID);

DBMS\_OUTPUT.PUT\_LINE('Username: ' || UserRec.Username);

DBMS\_OUTPUT.PUT\_LINE('Vehicles Count: ' || UserRec.Vehicle\_Count);

DBMS\_OUTPUT.NEW\_LINE;

END LOOP;

END;

/

CREATE TABLE UsersAudit (

Audit\_ID INT PRIMARY KEY,

User\_ID INT,

Username VARCHAR(255),

Password VARCHAR(255),

Change\_Type VARCHAR(10), -- 'INSERT', 'UPDATE', or 'DELETE'

Change\_Date TIMESTAMP

);

CREATE OR REPLACE TRIGGER UserAuditTrigger

AFTER INSERT OR UPDATE OR DELETE ON Users

FOR EACH ROW

BEGIN

-- Capture the change type

DECLARE

v\_change\_type VARCHAR(10);

BEGIN

IF INSERTING THEN

v\_change\_type := 'INSERT';

ELSIF UPDATING THEN

v\_change\_type := 'UPDATE';

ELSIF DELETING THEN

v\_change\_type := 'DELETE';

END IF;

-- Insert old and new values into the audit table

INSERT INTO UsersAudit (Audit\_ID, User\_ID, Username, Password, Change\_Type, Change\_Date)

VALUES (SEQ\_USERS\_AUDIT.NEXTVAL, :old.User\_ID, :old.Username, :old.Password, v\_change\_type, SYSTIMESTAMP);

-- For UPDATE, also capture the new values

IF UPDATING THEN

INSERT INTO UsersAudit (Audit\_ID, User\_ID, Username, Password, Change\_Type, Change\_Date)

VALUES (SEQ\_USERS\_AUDIT.NEXTVAL, :new.User\_ID, :new.Username, :new.Password, v\_change\_type, SYSTIMESTAMP);

END IF;

END;

END;

/

CREATE SEQUENCE SEQ\_USERS\_AUDIT;

CREATE SEQUENCE SEQ\_USERS\_AUDIT;

CREATE OR REPLACE TRIGGER UserAuditTrigger

AFTER INSERT OR UPDATE OR DELETE ON Users

FOR EACH ROW

BEGIN

-- Capture the change type

DECLARE

v\_change\_type VARCHAR(10);

BEGIN

IF INSERTING THEN

v\_change\_type := 'INSERT';

ELSIF UPDATING THEN

v\_change\_type := 'UPDATE';

ELSIF DELETING THEN

v\_change\_type := 'DELETE';

END IF;

-- Insert old and new values into the audit table

INSERT INTO UsersAudit (Audit\_ID, User\_ID, Username, Password, Change\_Type, Change\_Date)

VALUES (SEQ\_USERS\_AUDIT.NEXTVAL, :old.User\_ID, :old.Username, :old.Password, v\_change\_type, SYSTIMESTAMP);

-- For UPDATE, also capture the new values

IF UPDATING THEN

INSERT INTO UsersAudit (Audit\_ID, User\_ID, Username, Password, Change\_Type, Change\_Date)

VALUES (SEQ\_USERS\_AUDIT.NEXTVAL, :new.User\_ID, :new.Username, :new.Password, v\_change\_type, SYSTIMESTAMP);

END IF;

END;

END;

/

INSERT INTO Users (User\_ID, Username, Password)

VALUES (11, 'NewUser', 'NewPassword');

SELECT \* FROM UsersAudit;

DECLARE

-- Declare variables

v\_username VARCHAR(255);

v\_location\_name VARCHAR(255);

v\_geofence\_name VARCHAR(255);

v\_route\_name VARCHAR(255);

v\_location\_id INT;

v\_route\_id INT;

v\_vehicle\_id INT;

BEGIN

-- Retrieve a username from the Users table

SELECT Username INTO v\_username FROM Users WHERE User\_ID = 3;

DBMS\_OUTPUT.PUT\_LINE('Username: ' || v\_username);

-- Retrieve a location name from the DeliveryLocations table

SELECT Name INTO v\_location\_name FROM DeliveryLocations WHERE Location\_ID = 3;

DBMS\_OUTPUT.PUT\_LINE('Location Name: ' || v\_location\_name);

-- Retrieve a geofence name from the Geofences table

SELECT Name INTO v\_geofence\_name FROM Geofences WHERE Geofence\_ID = 2;

DBMS\_OUTPUT.PUT\_LINE('Geofence Name: ' || v\_geofence\_name);

-- Retrieve a route name from the Routes table

SELECT Name INTO v\_route\_name FROM Routes WHERE Route\_ID = 4;

DBMS\_OUTPUT.PUT\_LINE('Route Name: ' || v\_route\_name);

-- Retrieve a location ID from the RouteDeliveryLocations table

SELECT Location\_ID INTO v\_location\_id FROM RouteDeliveryLocations WHERE Route\_ID = 2;

DBMS\_OUTPUT.PUT\_LINE('Location ID: ' || v\_location\_id);

-- Retrieve a route ID from the RouteDeliveryLocations table

SELECT Route\_ID INTO v\_route\_id FROM RouteDeliveryLocations WHERE Location\_ID = 4;

DBMS\_OUTPUT.PUT\_LINE('Route ID: ' || v\_route\_id);

-- Retrieve a vehicle ID from the Vehicles table

SELECT User\_ID INTO v\_vehicle\_id FROM Vehicles WHERE Vehicle\_ID = 6;

DBMS\_OUTPUT.PUT\_LINE('Vehicle ID: ' || v\_vehicle\_id);

END;

DECLARE

-- Declare variables

v\_username VARCHAR(255);

v\_location\_name VARCHAR(255);

v\_geofence\_name VARCHAR(255);

v\_route\_name VARCHAR(255);

v\_vehicle\_id INT;

BEGIN

-- Retrieve a username from the Users table

SELECT Username INTO v\_username FROM Users WHERE User\_ID = 3;

DBMS\_OUTPUT.PUT\_LINE('Username: ' || v\_username);

-- Retrieve a location name from the DeliveryLocations table

SELECT Name INTO v\_location\_name FROM DeliveryLocations WHERE Location\_ID = 3;

DBMS\_OUTPUT.PUT\_LINE('Location Name: ' || v\_location\_name);

-- Retrieve a geofence name from the Geofences table

SELECT Name INTO v\_geofence\_name FROM Geofences WHERE Geofence\_ID = 2;

DBMS\_OUTPUT.PUT\_LINE('Geofence Name: ' || v\_geofence\_name);

-- Retrieve a route name from the Routes table

SELECT Name INTO v\_route\_name FROM Routes WHERE Route\_ID = 4;

DBMS\_OUTPUT.PUT\_LINE('Route Name: ' || v\_route\_name);

-- Retrieve a vehicle ID from the Vehicles table

SELECT User\_ID INTO v\_vehicle\_id FROM Vehicles WHERE Vehicle\_ID = 6;

DBMS\_OUTPUT.PUT\_LINE('Vehicle ID: ' || v\_vehicle\_id);

END;

CREATE OR REPLACE FUNCTION GetUsernameByUserID(p\_UserID IN INT) RETURN VARCHAR2 IS

v\_username VARCHAR2(255);

BEGIN

-- Retrieve the username for the provided User\_ID

SELECT Username INTO v\_username

FROM Users

WHERE User\_ID = p\_UserID;

RETURN v\_username;

END GetUsernameByUserID;

SELECT GetUsernameByUserID(2) FROM DUAL

SELECT

V.Vehicle\_ID,

R.Route\_ID,

COUNT(\*) AS UsageCount

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

GROUP BY

V.Vehicle\_ID, R.Route\_ID

ORDER BY

UsageCount DESC;

WITH VehicleMaxRoute AS (

SELECT

V.Vehicle\_ID,

R.Route\_ID,

COUNT(\*) AS UsageCount

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

GROUP BY

V.Vehicle\_ID, R.Route\_ID

)

SELECT

VMR.Vehicle\_ID,

R.Name AS RouteName,

VMR.UsageCount

FROM

VehicleMaxRoute VMR

JOIN

Routes R ON VMR.Route\_ID = R.Route\_ID

WHERE

VMR.UsageCount = (SELECT MAX(UsageCount) FROM VehicleMaxRoute WHERE Vehicle\_ID = VMR.Vehicle\_ID);

SELECT

V.Vehicle\_ID,

R.Name AS RouteName,

COUNT(\*) AS UsageCount

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

WHERE

V.Vehicle\_ID = 1

GROUP BY

V.Vehicle\_ID, R.Name

HAVING

COUNT(\*) = (

SELECT

MAX(COUNT(\*))

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

WHERE

V.Vehicle\_ID = 1

GROUP BY

V.Vehicle\_ID, R.Name

);

SELECT

COUNT(\*) AS Vehicle1MostTraveledRouteCount

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

WHERE

V.Vehicle\_ID = 1

GROUP BY

V.Vehicle\_ID

HAVING

COUNT(\*) = (

SELECT

MAX(COUNT(\*))

FROM

Vehicles V

JOIN

VehicleLocations VL ON V.Vehicle\_ID = VL.Vehicle\_ID

JOIN

RouteDeliveryLocations RL ON VL.Location\_ID = RL.Location\_ID

JOIN

Routes R ON RL.Route\_ID = R.Route\_ID

WHERE

V.Vehicle\_ID = 1

GROUP BY

V.Vehicle\_ID

);

