Danny Vu

1001045960

CSE-3330-003

Car Rental Database 2019

HONOR CODE

I pledge, on my honor, to uphold UT Arlington’s tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Students are required to not share any of the project related documents and solution with others in any way or form even after the completion of the project. Students may, however, show their projects to interviewers.

**TASK 1:**

**QUERY 1:**

ALTER TABLE RENTAL

ADD COLUMN Returned INT;

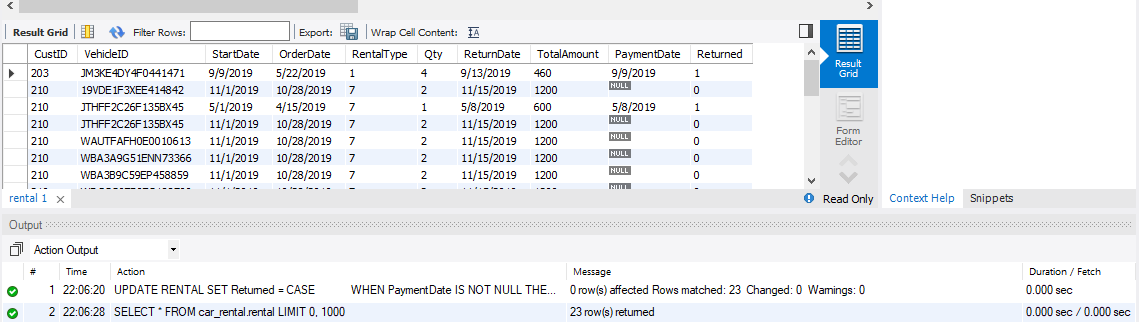
UPDATE RENTAL

SET Returned = CASE

WHEN PaymentDate IS NOT NULL THEN 1

ELSE 0

END;



**QUERY 2:**

SELECT RL.OrderDate, RL.StartDate, RL.ReturnDate, DATEDIFF(RL.ReturnDate, RL.StartDate) as TotalDays,

V.VehicleID as VIN, V.Description as Vehicle,

CASE

WHEN R.Category = '0' THEN 'BASIC'

WHEN R.Category = '1' THEN 'LUXUR'

END as Category,

CASE

WHEN R.Type = '1' THEN 'COMPACT'

WHEN R.Type = '2' THEN 'MEDIUM'

WHEN R.Type = '3' THEN 'LARGE'

WHEN R.Type = '4' THEN 'SUV'

WHEN R.Type = '5' THEN 'TRUCK'

WHEN R.Type = '6' THEN 'VAN'

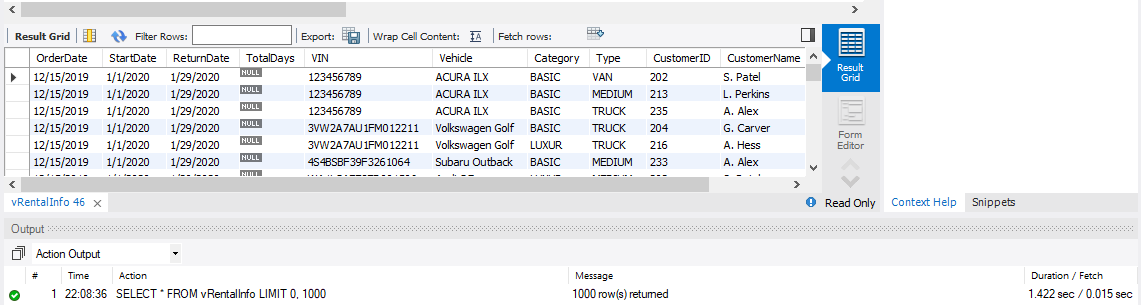
END as Type,

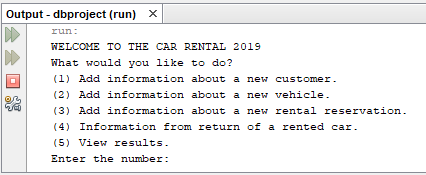
C.CustID as CustomerID, C.Name as CustomerName, RL.TotalAmount as OrderAmount

FROM RENTAL RL, VEHICLE V, CUSTOMER C, Rate R

ORDER BY RL.StartDate ASC;

SELECT \* FROM vRentalInfo;





The GUI allows user to choose to add new customer/vehicle/rental/return of rental and view results

Query 1:

String sql = "INSERT INTO CUSTOMER (Name, Phone)"

+ "VALUES (?, ?)";

preparedStatement = connection.prepareStatement(sql);

//set param values

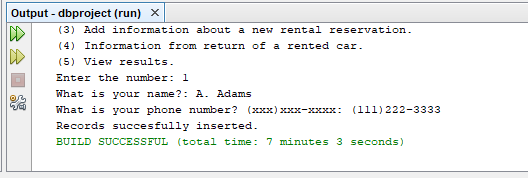
preparedStatement.setString(1, username);

preparedStatement.setString(2, userphone);

//execute query

int i = preparedStatement.executeUpdate();

This code inserts into a new customer



Query 2:

String sql = "INSERT INTO VEHICLE (VehicleID, Description, Year, Type, Category)"

+ "VALUES (?, ?, ?, ?, ?)";

preparedStatement = connection.prepareStatement(sql);

//set param values

preparedStatement.setString(1, vid);

preparedStatement.setString(2, desc);

preparedStatement.setInt(3, year);

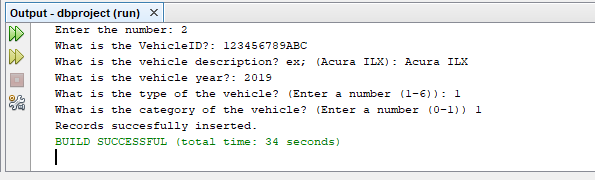
preparedStatement.setInt(4, type);

preparedStatement.setInt(5, category);

//execute query

int i = preparedStatement.executeUpdate();

This code inserts into a new vehicle



Query 3:

String sql = "INSERT INTO RENTAL (StartDate, ReturnDate, RentalType, PaymentDate)"

+ "VALUES (?, ?, ?, ?, ?)";

preparedStatement = connection.prepareStatement(sql);

//set param values

preparedStatement.setString(1, startdate);

preparedStatement.setString(2, returndate);

preparedStatement.setInt(3, rentaltype);

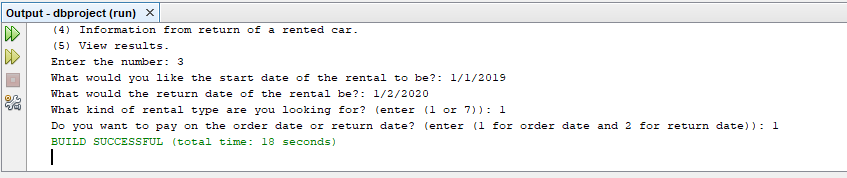
preparedStatement.setInt(4, pay);

String query = "SELECT \* FROM VEHICLE V, RENTAL R WHERE V.VehicleID = R.VehicleID";

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(query);

This code inserts into rental and selects to view the available cars to rent



Query 4:

String query = "SELECT \* FROM RENTAL R, VEHICLE V , CUSTOMER C WHERE R.VehicleID = V.VehicleID and R.CustID = C.CustID";

Statement st = connection.createStatement();

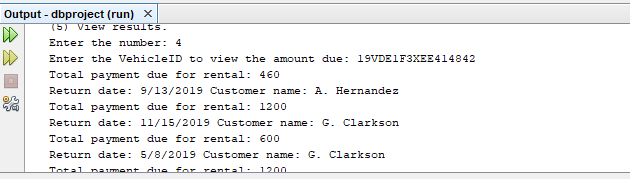
ResultSet rs = st.executeQuery(query);

String sql = "UPDATE RENTAL (TotalAmount)"

+ "VALUES (?)";

preparedStatement = connection.prepareStatement(sql);

This code selects the rental and checks the returned rental car



Query 5:

String query = "SELECT \* FROM vRentalInfo";

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(query);

String sql = "SELECT \* FROM vRentalInfo WHERE CustomerID LIKE '"+data+"' OR VIN LIKE'"+data+" OR Description LIKE'"+search+" OR CustomerName LIKE'"+search+"";

Statement sts = connection.createStatement();

ResultSet rss = sts.executeQuery(sql);

This code uses the views results and lists all the customer names/vin/description and customerid. Also, user can search partially or by the attributes.

