**CSE 3330: Project 2 - Part 2**

Rabindra Yadav

Saugat Karki

Shiska Raut

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.

Rabindra Yadav

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.

Saugat Karki

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.

Shiska Raut

TASK 1:

To create tables and define constraints such as Primary keys, Foreign keys and Null values we used the following CREATE TABLE Commands:

CREATE TABLE CUSTOMER

(CustID INT NOT NULL UNIQUE AUTO\_INCREMENT,

Name VARCHAR(60) NOT NULL,

Phone VARCHAR(14) NOT NULL,

PRIMARY KEY(CustID));

CREATE TABLE VEHICLE

(VehicleID VARCHAR(18) NOT NULL UNIQUE,

Description VARCHAR(100) NOT NULL,

Year INT NOT NULL,

Type TINYINT NOT NULL DEFAULT 1,

Category TINYINT NOT NULL DEFAULT 0,

CHECK (Category IN (0,1)),

PRIMARY KEY (VehicleID));

CREATE TABLE RENTAL

(RentalID INT NOT NULL UNIQUE AUTO\_INCREMENT,

CustID INT NOT NULL,

VehicleID VARCHAR(18) NOT NULL,

StartDate DATE NOT NULL,

OrderDate DATE NOT NULL,

RentalType TINYINT NOT NULL DEFAULT 1

CHECK (RentalType IN (1,7)),

Qty TINYINT NOT NULL DEFAULT 1,

ReturnDate DATE NOT NULL,

TotalAmount FLOAT NOT NULL,

PaymentDate DATE,

PRIMARY KEY(RentalID),

FOREIGN KEY(VehicleID) REFERENCES VEHICLE(VehicleID),

FOREIGN KEY(CustID) REFERENCES CUSTOMER(CustID));

CREATE TABLE RATE

(Type TINYINT NOT NULL DEFAULT 1

CHECK (Type IN (1,2,3,4,5,6)),

Category TINYINT NOT NULL DEFAULT 0

CHECK (Category IN (0,1)),

Weekly FLOAT NOT NULL,

Daily FLOAT NOT NULL);

Schema Datatype and Rationale:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table | Attribute | Datatype | Rationale | Key Type |
| CUSTOMER | CustID | INT | 32-bit value to sore large number of customers | PRIMARY, AUTO INCREMENT |
|  | Name | VARCHAR | Store name as characters of variable length | - |
|  | Phone | VARCHAR | Store number as characters of variable length containing ‘(‘,’)’,’-‘ | - |
|  | | | | |
| VEHICLE | VahicleID | VARCHAR | Vehicle ID has both letters and numbers | PRIMARY |
|  | Description | VARCHAR | Description has both letters and numbers | - |
|  | Year | INT | 32-bit value to store year | - |
|  | Type\*\* | TINYINT | 8-bit int to store type as 1-6 | - |
|  | Category\* \*\* | TINYINT | 8-bit int to store category as 0 or 1 | - |
|  | | | | |
| RENTAL | RentalID | INT | 32-bit value to sore large number of rentals | PRIMARY,  AUTO INCREMENT |
|  | CustID | INT | - | FOREIGN |
|  | VehicleID | VARCHAR | - | FOREIGN |
|  | StartDate | DATE | To store date info in YYYY-MMM-DD format for ease of use | - |
|  | OrderDate | DATE | To store date info in YYYY-MMM-DD format for ease of use | - |
|  | RentalType\* \*\* | TINYINT | 8-bit int to store rental type as 1 or 7 | - |
|  | Qty\*\* | TINYINT | 8-bit int to store quantity 255 o unsigned range | - |
|  | ReturnDate | DATE | To store date info in YYYY-MMM-DD format for ease of use | - |
|  | TotalAmount | FLOAT | Monetary data is best stored as float or double to save decimal values | - |
|  | PaymentDate | DATE | To store date info in YYYY-MMM-DD format for ease of use | - |
|  | | | | |
| RATE | Type\* \*\* | TINYINT | - | - |
|  | Category\* \*\* | TINYINT | - | - |
|  | Weekly | FLOAT | Monetary data is best stored as float or double to save decimal values | - |
|  | Daily | FLOAT | Monetary data is best stored as float or double to save decimal values | - |

\*Schematic constraints placed to restrict data scope using the CHECK command:

VEHICLE : Category IN (0,1)

RENTAL : RentalType IN (1,7)

RATE : Type IN (1,2,3,4,5,6)

: Category IN (0,1)

\*\*Default values included using DEFAULT command.

VEHICLE : Type defaulted to 1(Compact)

: Category defaulted to 0(Basic)

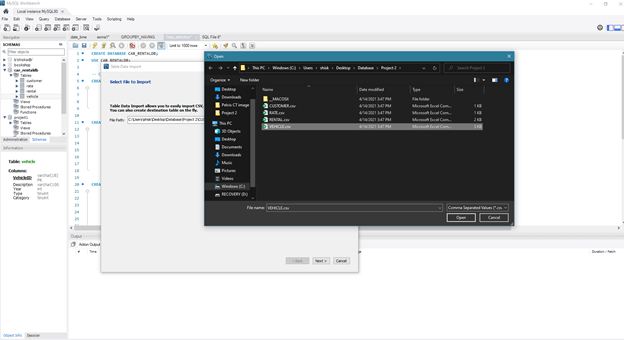
RENTAL : Rental type defaulted to 1(Daily)

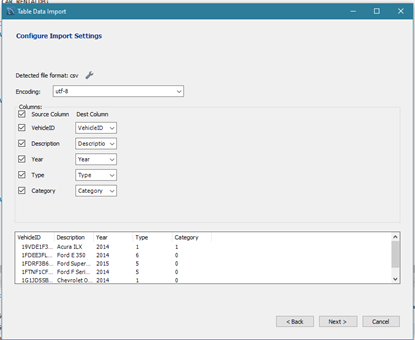
RATE : Type defaulted to 1(Compact)

: Category defaulted to 0(Basic)

TASK 2:

To insert values into the database provided as excel.csv files, ‘Table data Import Wizard’ was used in MySQL Workbench. Screenshots are provided below:



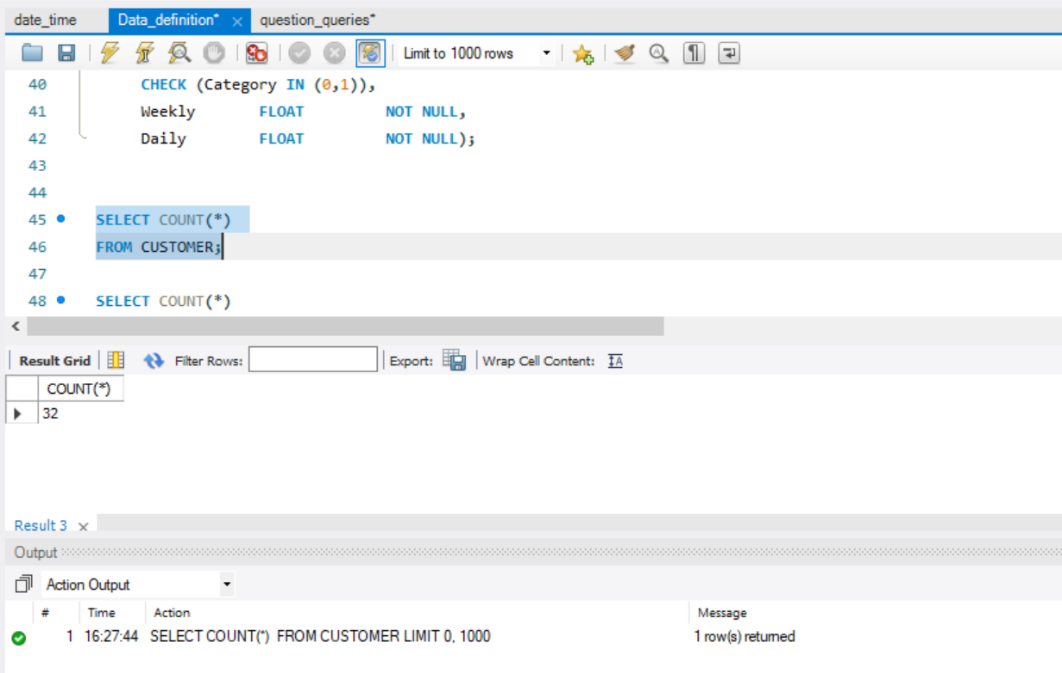


A COUNT command was used to count rows in each table:

Results for table CUSTOMER:

Query: SELECT COUNT(\*)

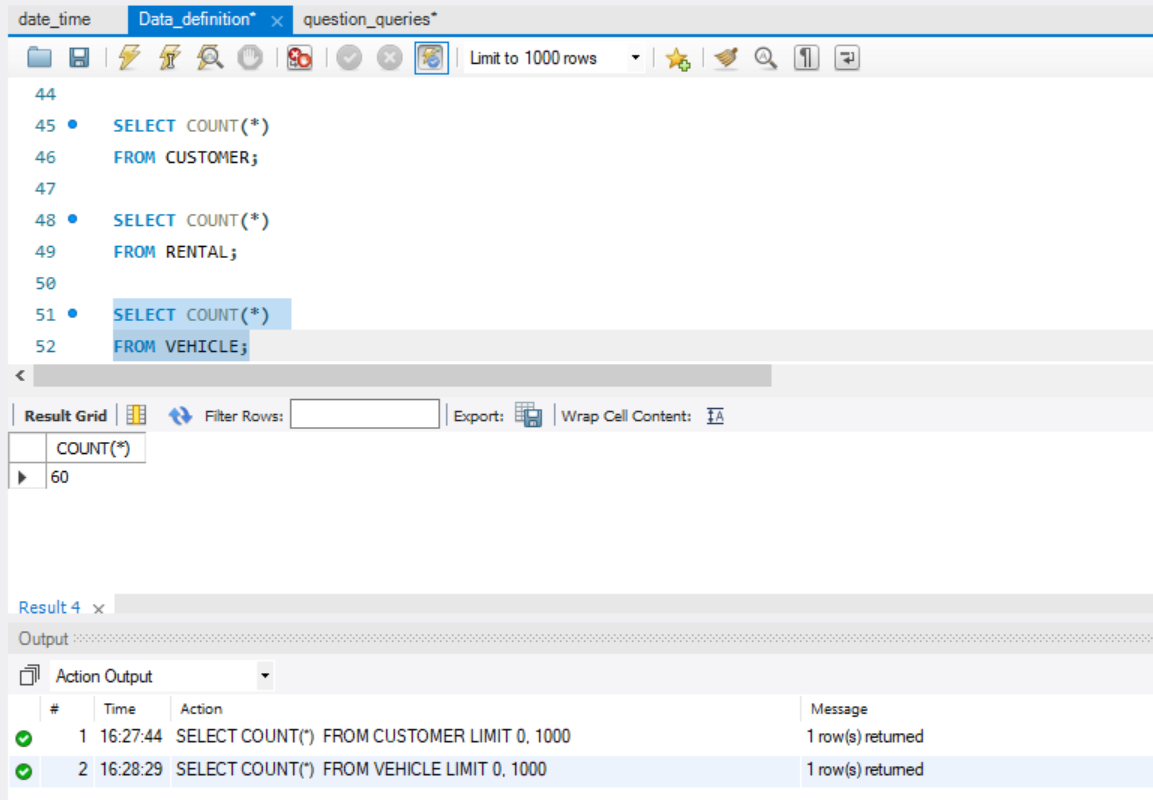
FROM CUSTOMER;



Results for table VEHICLE:

Query: SELECT COUNT(\*)

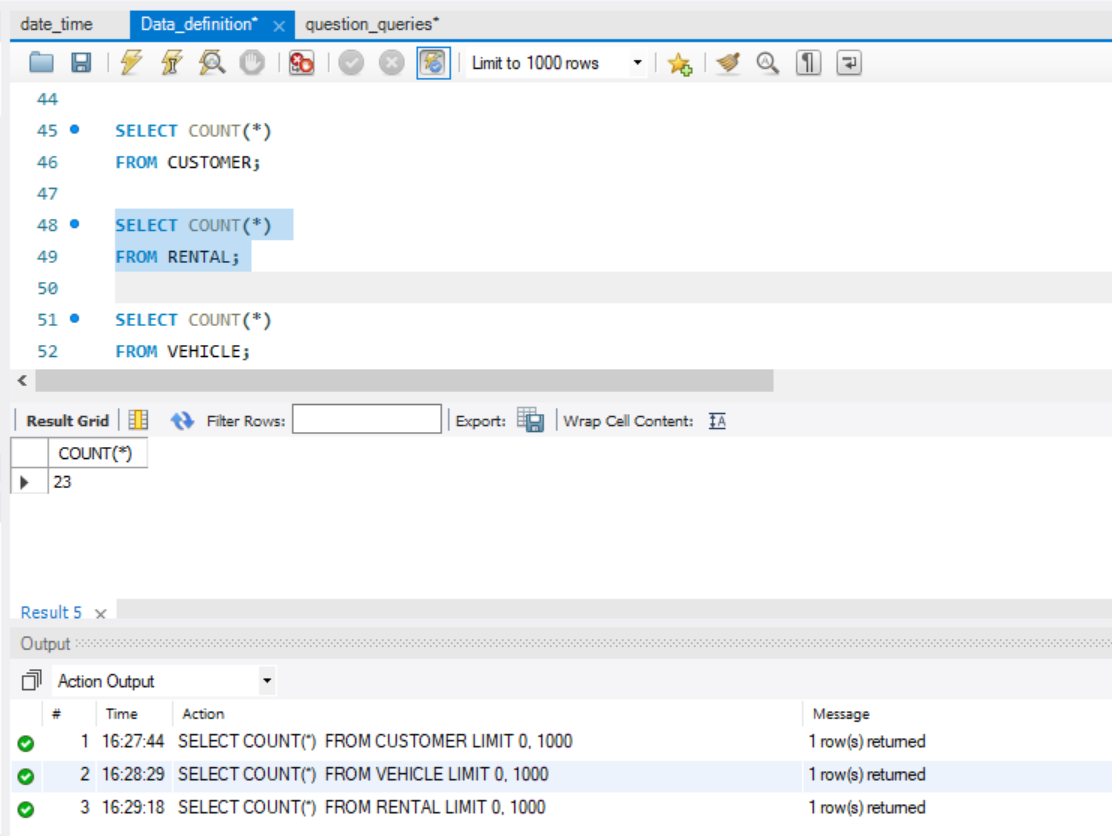
FROM VEHICLE;



Results for table RENTAL:

Query: SELECT COUNT(\*)

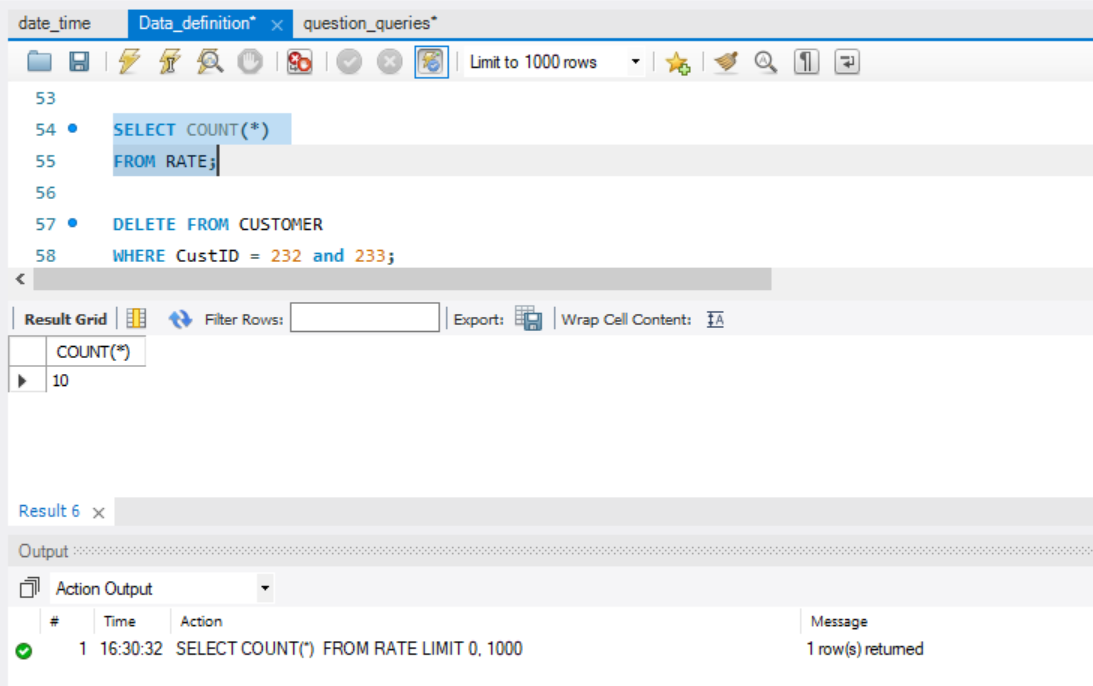
FROM RENTAL;



Results for table RATE:

Quert: SELECT COUNT(\*)

FROM RATE;



TASK 3:

**Question 1:**

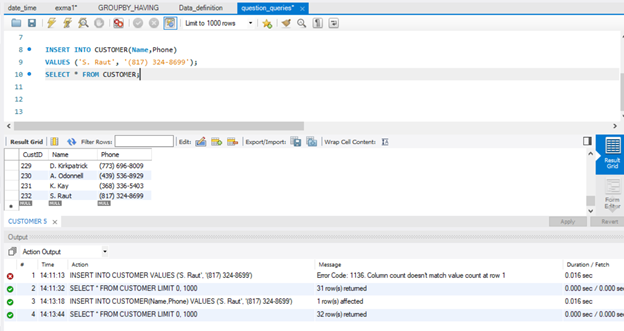
Insert yourself as a New Customer. Do not provide the CustomerID in your query.

Query:

INSERT INTO CUSTOMER(Name,Phone)

VALUES ('S. Raut', '(817) 324-8699');

SELECT \* FROM CUSTOMER;



**Question 2:**

Update your phone number to (837) 721-8965

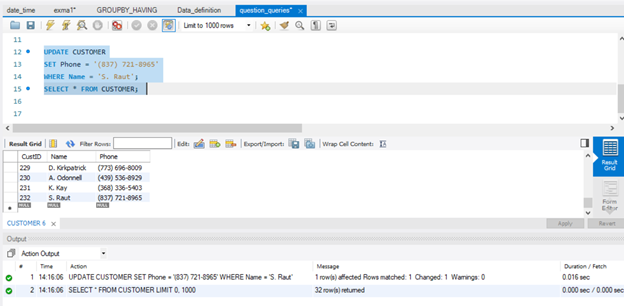
Query:

UPDATE CUSTOMER

SET Phone = '(837) 721-8965'

WHERE Name = 'S. Raut';

SELECT \* FROM CUSTOMER;



**Question 3:**

Increase only daily rates for luxury vehicles by 5%

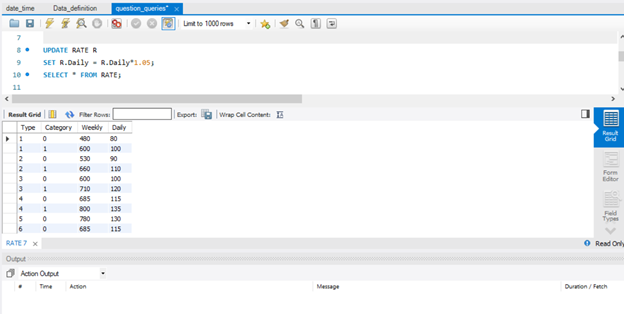
Query:

UPDATE RATE R

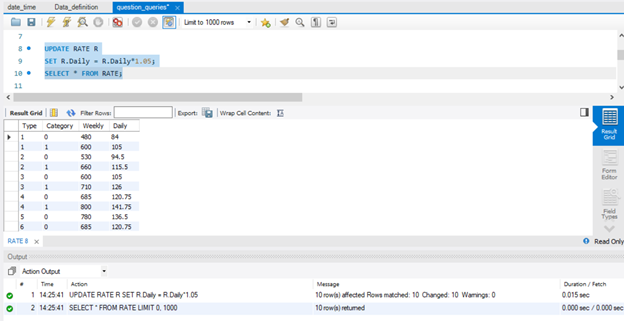
SET R.Daily = R.Daily\*1.05;

SELECT \* FROM RATE;

Before update:



After update:



**Question 4a:**

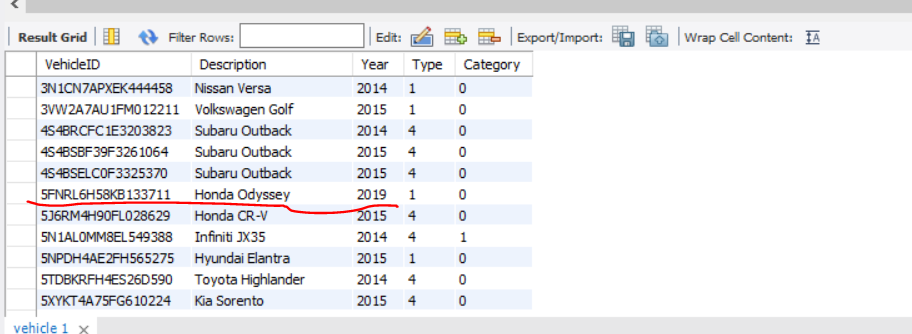
Insert a new luxury van with the following info: Honda Odyssey 2019, vehicle id: 5FNRL6H58KB133711

Query:

INSERT INTO VEHICLE

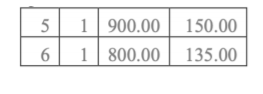
(VehicleID,Description,Year)

VALUES ('5FNRL6H58KB133711','Honda Odyssey',2019) ;



**Questions 4b:**

You also need to insert the following rates:



Query:

INSERT INTO RATE

(Type,Category,Weekly,Daily)

VALUES

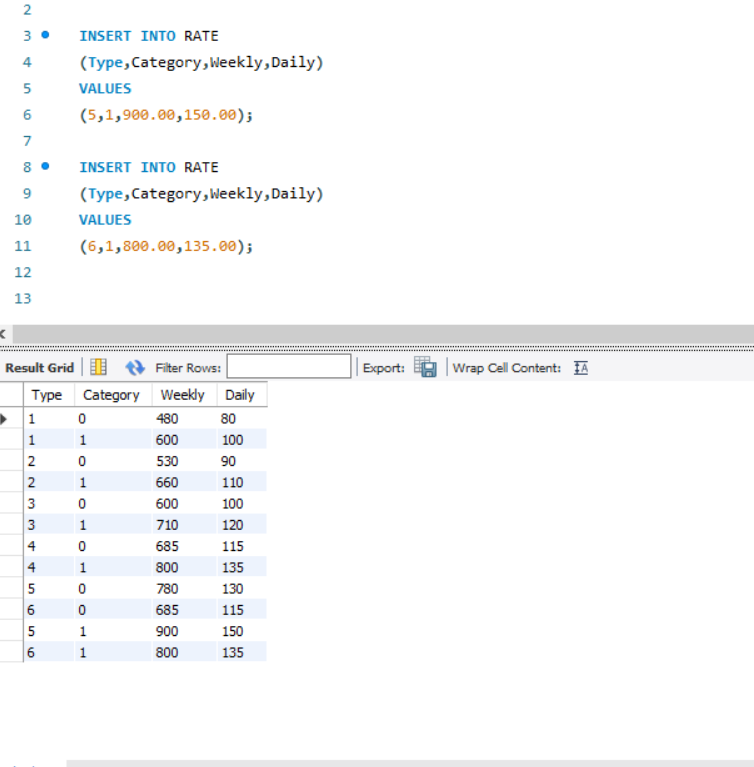
(5,1,900.00,150.00);

INSERT INTO RATE

(Type,Category,Weekly,Daily)

VALUES

(6,1,800.00,135.00);



**Question 5:**

Return all Compact(1) & Luxury(1) vehicles that were available for rent from June 01, 2019 until June 20, 2019. List VechicleID as VIN, Description, year, and how many days have been rented so far. You need to change the weeks into days.

Query:

CREATE VIEW AVAILABILITY1

AS SELECT V.VehicleID, R.RentalID, R.StartDate, R.ReturnDate,

CASE

WHEN R.RentalID IS NOT NULL THEN DATEDIFF(R.ReturnDate, R.StartDate)

WHEN R.RentalID IS NULL THEN DATEDIFF('2021-04-16', '2021-04-16') -- zero

END AS Days\_Rented

FROM VEHICLE V

LEFT JOIN RENTAL R

ON V.VehicleID = R.VehicleID;

SELECT \* FROM AVAILABILITY1;

SELECT DISTINCT V.VehicleID AS VIN, V.Description, V.Year, SUM(A.Days\_Rented) AS Total\_days\_rented

FROM VEHICLE V

LEFT JOIN AVAILABILITY1 A

ON V.VehicleID = A.VehicleID

WHERE V.VehicleID IN (SELECT DISTINCT A1.VehicleID

FROM AVAILABILITY1 A1

WHERE ((A1.StartDate < '2019-06-01' AND A1.ReturnDate < '2019-06-01') -- avaiable bc already returned before 1

OR (A1.StartDate > '2019-06-20' AND A1.ReturnDate > '2019-06-20')) -- avaiable bc will only ne rented after 20

AND A1.VehicleID NOT IN (SELECT DISTINCT A3.VehicleID

FROM AVAILABILITY1 A2, AVAILABILITY1 A3

WHERE A2.VehicleID = A3.VehicleID

AND ((A3.ReturnDate > '2019-06-00' AND A3.ReturnDate < '2019-06-21') -- vehicle returned during forbidden period

OR (A3.StartDate > '2019-06-00' AND A3.StartDate < '2019-06-21')))) -- rented during forbidden period

OR (A.RentalID IS NULL) -- avaialble bc no rentals scheduled yet

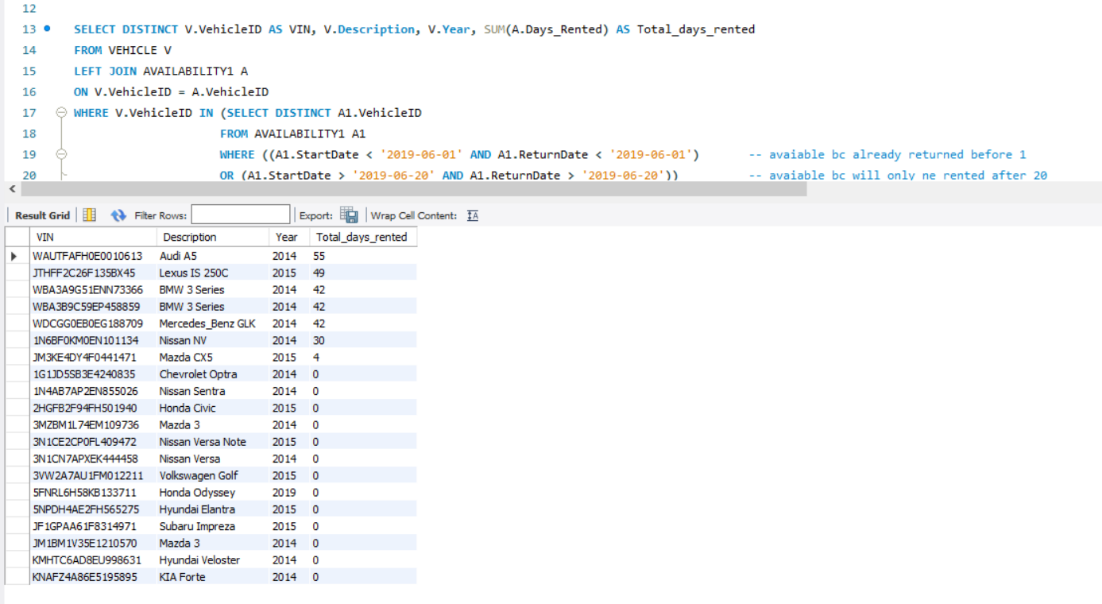
AND V.Type = 1 AND V.Category = 0

GROUP BY V.VehicleID

ORDER BY Total\_days\_rented desc

;

NOTE: A view named Availability1 was created to calculate the number of days each car was rented.



**Question 6:**

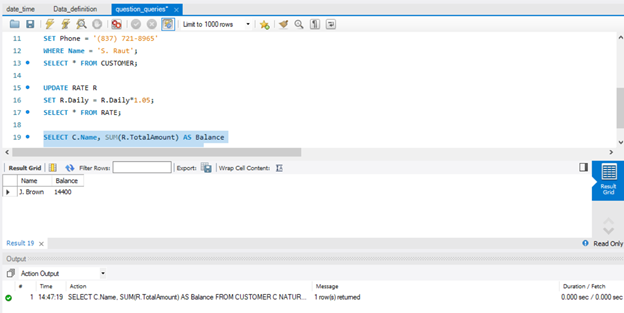
Return a list with the remaining balance for the customer with the id ‘221’. List customer name, and the balance.

Query:

ELECT C.Name, SUM(R.TotalAmount) AS Balance

FROM CUSTOMER C NATURAL JOIN RENTAL R

WHERE CustID = 221 AND R.PaymentDate IS NULL;



**Question 7:**

Create a report that will return all vehicles. List the VehicleID as VIN, Description, Year, Type, Category, and Weekly and Daily rates. For the vehicle Type and Category, you need to use the SQL Case statement to substitute the numbers with text. Order your results based on Category (first Luxury and then Basic) and Type based on the Type number, not the text.

Query:

SELECT V.VehicleID AS VIN, V.Description, V.Year,

CASE

WHEN V.Category = 1 THEN 'Luxury'

WHEN V.Category = 0 THEN 'Basic'

END AS Category,

CASE

WHEN V.Type = 1 THEN 'Compact'

WHEN V.Type = 2 THEN 'Medium'

WHEN V.Type = 3 THEN 'Large'

WHEN V.Type = 4 THEN 'SUV'

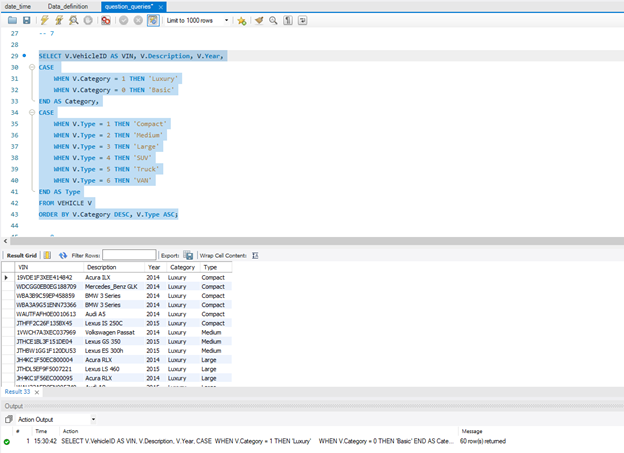
WHEN V.Type = 5 THEN 'Truck'

WHEN V.Type = 6 THEN 'VAN'

END AS Type

FROM VEHICLE V

ORDER BY V.Category DESC, V.Type ASC;



**Question 8:**

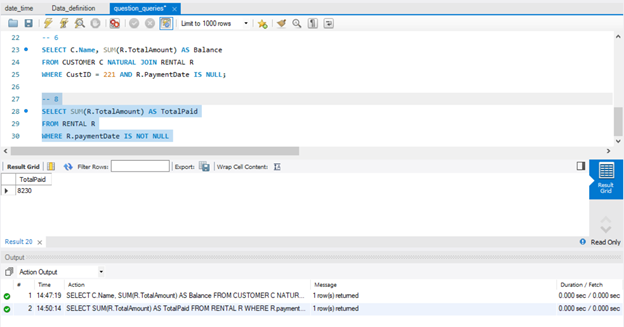
What is the total of money that customers paid to us until today?

Query:

SELECT SUM(R.TotalAmount) AS TotalPaid

FROM RENTAL R

WHERE R.paymentDate IS NOT NULL



**Question 9 -a)**

Create a report for the J. Brown customer with all vehicles he rented. List the description, year, type, and category. Also calculate the unit price for every rental, the total duration mention if it is on weeks or days, the total amount, and if there is any payment. Similarly, as in Question 7, you need to change the numeric values to the corresponding text. Order the results by the StartDate.

Ans:

Query:

-- 9-a) Report for the J. Brown Customer

SELECT Description, Year,

CASE

WHEN Type = 1 THEN 'Compact'

WHEN Type = 2 THEN 'Medium'

WHEN Type = 3 THEN 'Large'

WHEN Type = 4 THEN 'SUV'

WHEN Type = 5 THEN 'Truck'

WHEN Type = 6 THEN 'VAN'

END AS Type,

CASE

WHEN Category = 0 THEN 'Basic'

WHEN Category = 1 THEN 'Luxury'

END AS Category,

(TotalAmount / Qty) AS UnitPrice,

CASE

WHEN RentalType = 1 THEN CONCAT((ReturnDate-StartDate), ' Day/s')

WHEN RentalType =7 THEN CONCAT(FLOOR(((ReturnDate-StartDate)/7)), ' Week/s')

END AS TotalDuration,

TotalAmount,

CASE

WHEN PaymentDate is NOT NULL THEN 'Paid'

ELSE 'Not paid'

END AS PaymentStatus

FROM CUSTOMER

INNER JOIN RENTAL

ON CUSTOMER.CustID = RENTAL.CustID

INNER JOIN VEHICLE

ON RENTAL.VehicleID = VEHICLE.VehicleID

WHERE Name = 'J. Brown'

ORDER BY StartDate;

Screenshot:

A picture containing text, monitor, screenshot, indoor

Description automatically generated

**Question 9-b)**

For the same customer return the current balance.

Query

-- 9-b) Return current balance of J. Brown

SELECT SUM(TotalAmount) AS CurrentBalance

FROM CUSTOMER

INNER JOIN RENTAL

ON CUSTOMER.CustID = RENTAL.CustID

WHERE Name = "J. Brown"

AND PaymentDate IS NULL ;

Screenshot:

A picture containing text, screenshot, monitor, indoor

Description automatically generated

**Question 10)**

Retrieve all weekly rentals for the VehicleID ‘19VDE1F3XEE414842’ that are not paid yet. List the customer’s name, the start and return date, and the amount.

Query:

SELECT Name, StartDate, ReturnDate, TotalAmount

FROM CUSTOMER

INNER JOIN RENTAL

ON CUSTOMER.CustID = RENTAL.CustID

WHERE

(

VehicleID = '19VDE1F3XEE414842'

AND RentalType = 7

AND PaymentDate IS NULL

);

Screenshot:

A picture containing text, screenshot, monitor, black

Description automatically generated

**Question 11)**

Return all customers that they never rent a vehicle.

Query:

SELECT DISTINCT CUSTOMER.CustID, Name, Phone

FROM CUSTOMER

LEFT JOIN RENTAL

ON CUSTOMER.CustID = RENTAL.CustID

WHERE CUSTOMER.CustID = RENTAL.CustID;

Screenshot:

A picture containing text, screenshot, monitor, black

Description automatically generated

**Question 12)**

Return all rentals that the customer paid on the StartDate. List Customer Name, Vehicle Description, StartDate, ReturnDate, and TotalAmount. Order by customer Name.

Query:

SELECT Name, Description, StartDate, ReturnDate,TotalAmount

FROM CUSTOMER

INNER JOIN RENTAL

ON CUSTOMER.CustID = RENTAL.CustID

INNER JOIN VEHICLE

ON RENTAL.VehicleID = VEHICLE.VehicleID

WHERE StartDate = PaymentDate

ORDER BY Name;

Screenshot:

A picture containing text, screenshot, monitor, black

Description automatically generated