PROJECT – 2

PHASE - 2

Group Members: -

Sudharsan Rajam	: -	(#1001874246)
Dhruv Patel	: -	(#1001748952)
Nahal Maymandi	: -	(#1001785613)

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.

TASK – (1)

SQL QUERIES FOR CREATING TABLES

```
/* creating database */

 create database CarRental;
 use CarRental;

/* creating table CUSTOMER */

create table CUSTOMER(Cust_ID int NOT NULL auto_increment,
Name varchar(20), Phone varchar(15), primary key (Cust_ID));

/* creating table RATE */

create table RATE(Type int,Category boolean,Weekly decimal(5,2),
Daily decimal(5,2),primary key(Type,Category));

/* create table VEHICLE */

Create  table VEHICLE(Vehicle_ID varchar(20),
Description varchar(30), Year int , Type int,
Category boolean, primarykey(Vehicle_ID),
foreign key(Type,Category) references RATE(Type,Category));
```

```
/* create table RENTAL */


create table VEHICLE (Vehicle_ID varchar(20),
Description varchar(30), Year int , Type int,
Category boolean, primary key(Vehicle_ID),
foreign key(Type,Category) references RATE(Type,Category));
```

Comments (TASK-1)

- For the CUSTOMER table, customer id was chosen as the primary key and it was set as not null and auto increment was default value which is 1 .

- For the RATE table, Weekly and Daily rates were the given the decimal datatype and the composite primary key was defined for the table as primary keys could not have duplicate values.

- For the VEHICLE table, Category was given the Boolean data type and foreign key referential integrity constant was set to the RATE table

- For the RENTAL table, no primary key was set as both the customer ID and vehicle ID consisted of duplicates.

TASK – (2)

SQL QUERIES FOR INSERTION

For the insertion of data values into the tables, they were directly imported from the csv file into the My SQL Workbench.

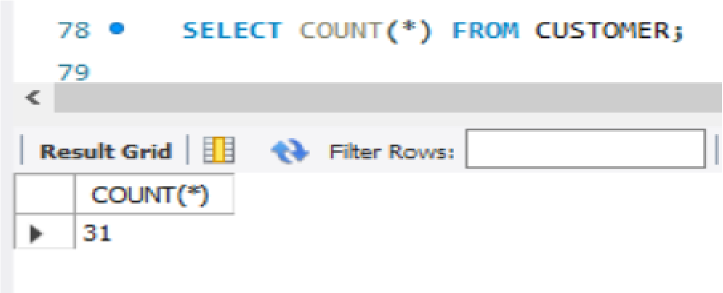
For the created tables, the values were inserted from Table Import Wizard feature from the MySQL Workbench

Just for implementation purposes, the CUSTOMER table was loaded through the SQL Syntax for insertion

Here are some of the Insertion statements used for the CUSTOMER Table :-

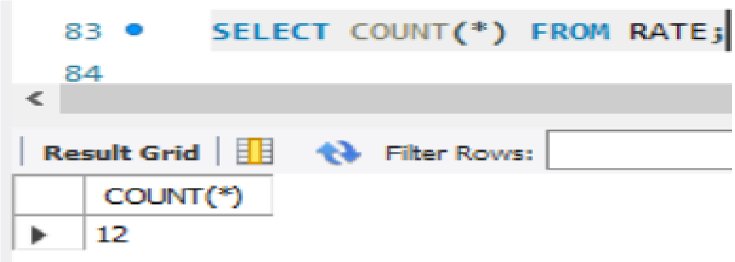
```
insert into CUSTOMER(Cust_ID, Name, Phone) values ('201','A. Parks','(214) 555-0127');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('202','S. Patel','(849) 811-6298');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('203','A. Hernandez','(355) 572-5385');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('204','G. Carver','(753) 763-8656');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('205','Sh. Byers','(912) 925-5332');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('206','L. Lutz','(931) 966-1775');

insert into CUSTOMER(Cust_ID, Name, Phone) values ('207','L. Bernal','(884) 727-0591');
```

Also for this task, the total records for every table was calculated using COUNT function



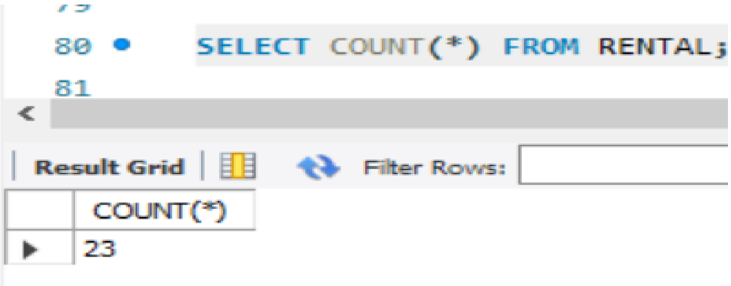
```
SELECT COUNT(*) FROM CUSTOMER;
```

```
COUNT(*)
31
```



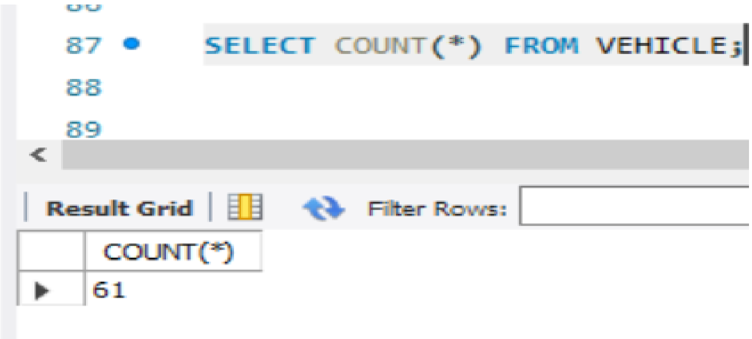
```
SELECT COUNT(*) FROM RATE;
```

```
COUNT(*)
12
```



```
SELECT COUNT(*) FROM RENTAL;
```

```
COUNT(*)
23
```



```
SELECT COUNT(*) FROM VEHICLE;
```

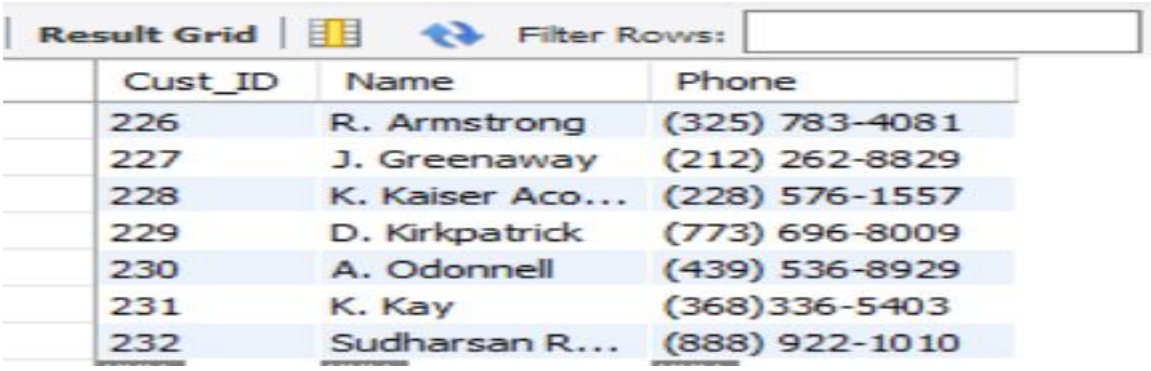
```
COUNT(*)
61
```

TASK – (3)

Execution of Queries on the database tables

Question 1: Insert yourself as a New Customer. Do not provide the CustomerID in your query.

insert into CUSTOMER(Name,Phone) values ('Sudharsan Rajam','(888) 922-1010');



Cust_ID	Name	Phone
226	R. Armstrong	(325) 783-4081
227	J. Greenaway	(212) 262-8829
228	K. Kaiser Aco...	(228) 576-1557
229	D. Kirkpatrick	(773) 696-8009
230	A. Odonnell	(439) 536-8929
231	K. Kay	(368)336-5403
232	Sudharsan R...	(888) 922-1010

The Customer ID was automatically set since it is defined as auto increment value by default in the primary key definition

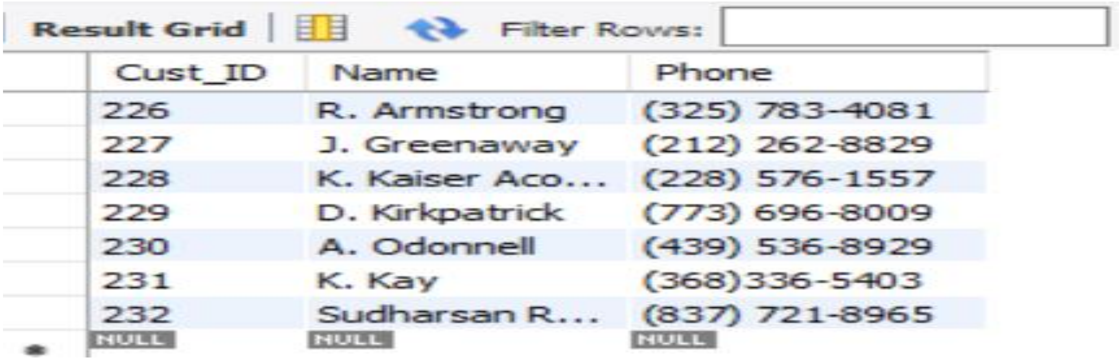
The number of records increased by 1 in this table to 32

Question 2: Update your phone number to (837) 721-8965.

UPDATE CUSTOMER

SET Phone='(837) 721-8965'

WHERE (Cust_ID ='232');



Cust_ID	Name	Phone
226	R. Armstrong	(325) 783-4081
227	J. Greenaway	(212) 262-8829
228	K. Kaiser Aco...	(228) 576-1557
229	D. Kirkpatrick	(773) 696-8009
230	A. Odonnell	(439) 536-8929
231	K. Kay	(368)336-5403
232	Sudharsan R...	(837) 721-8965
NULL	NULL	NULL

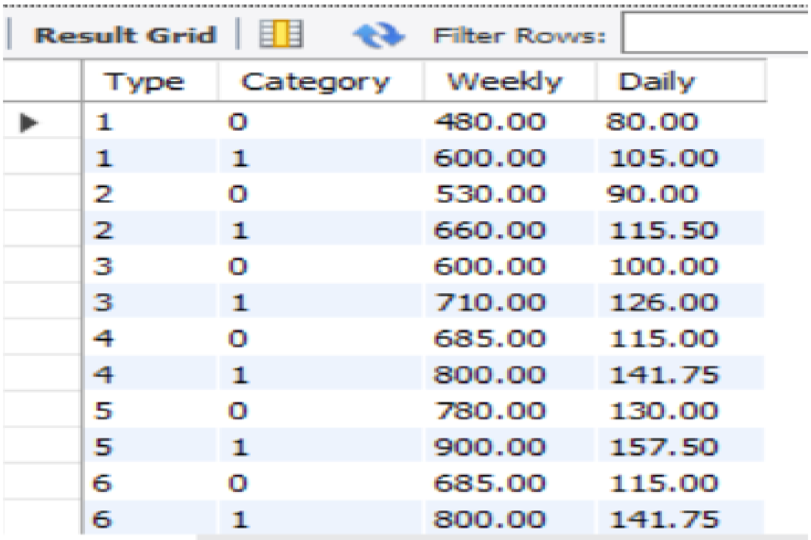
Total number of records affected was 1

Question 3: Increase only daily rates for luxury vehicles by 5%

UPDATE RATE

set Daily = Daily + (Daily * 5.0 / 100.0)

where Category = '1' AND Type >0;



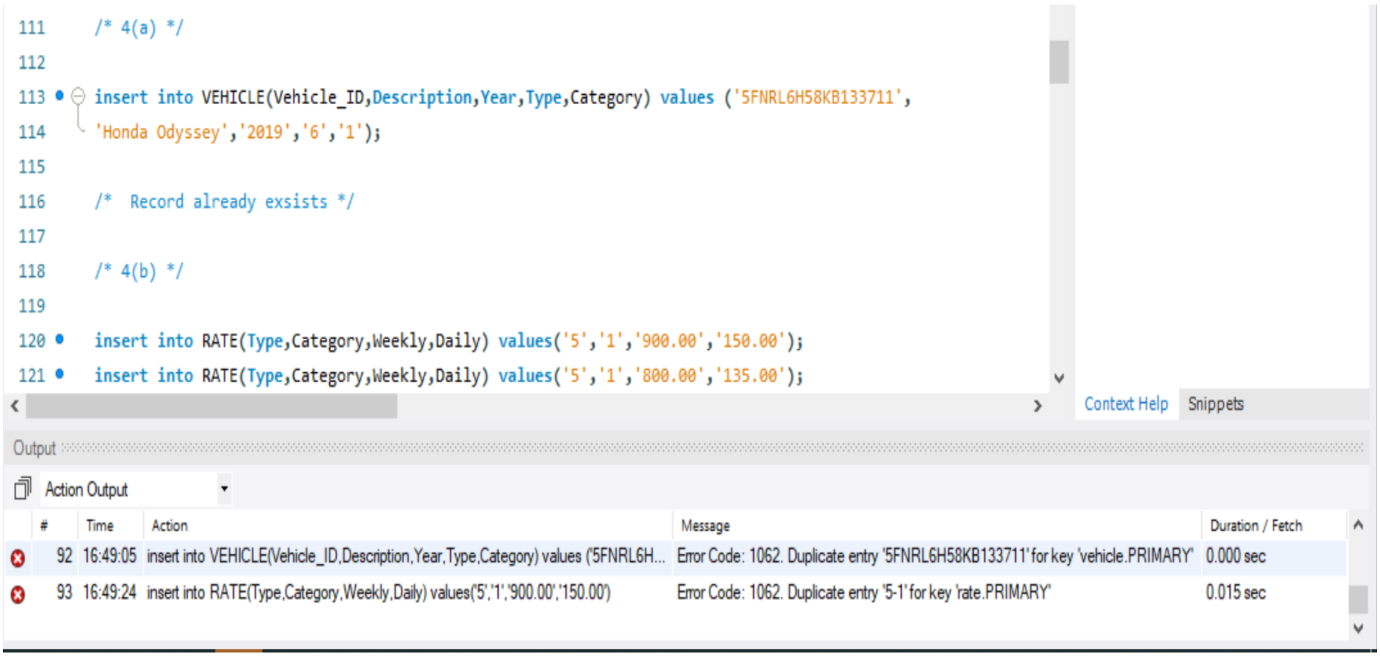
Type	Category	Weekly	Daily
1	0	480.00	80.00
1	1	600.00	105.00
2	0	530.00	90.00
2	1	660.00	115.50
3	0	600.00	100.00
3	1	710.00	126.00
4	0	685.00	115.00
4	1	800.00	141.75
5	0	780.00	130.00
5	1	900.00	157.50
6	0	685.00	115.00
6	1	800.00	141.75

Total number of records affected were 6

Question 4-a: Insert a new luxury van with the following info: Honda Odyssey 2019, vehicle id: 5FNRL6H58KB133711

Question 4-b: You also need to insert the following rates:

For both the Parts of Question- 4, the information already existed in the database tables VEHICLE and RATE respectively. Also, there are primary keys defined for the tables which will not allow duplicate records to be inserted so error will be shown for these insertion queries



```
/* 4(a) */

insert into VEHICLE(Vehicle_ID,Description,Year,Type,Category) values ('5FNRL6H58KB133711',
'Honda Odyssey','2019','6','1');

/*  Record already exsists */

/* 4(b) */

insert into RATE(Type,Category,Weekly,Daily) values('5','1','900.00','150.00');
insert into RATE(Type,Category,Weekly,Daily) values('5','1','800.00','135.00');
```

#	Time	Action	Message	Duration / Fetch
92	16:49:05	insert into VEHICLE(Vehicle_ID,Description,Year,Type,Category) values ('5FNRL6H...	Error Code: 1062. Duplicate entry '5FNRL6H58KB133711' for key 'vehicle.PRIMARY'	0.000 sec
93	16:49:24	insert into RATE(Type,Category,Weekly,Daily) values('5','1','900.00','150.00')	Error Code: 1062. Duplicate entry '5-1' for key 'rate.PRIMARY'	0.015 sec

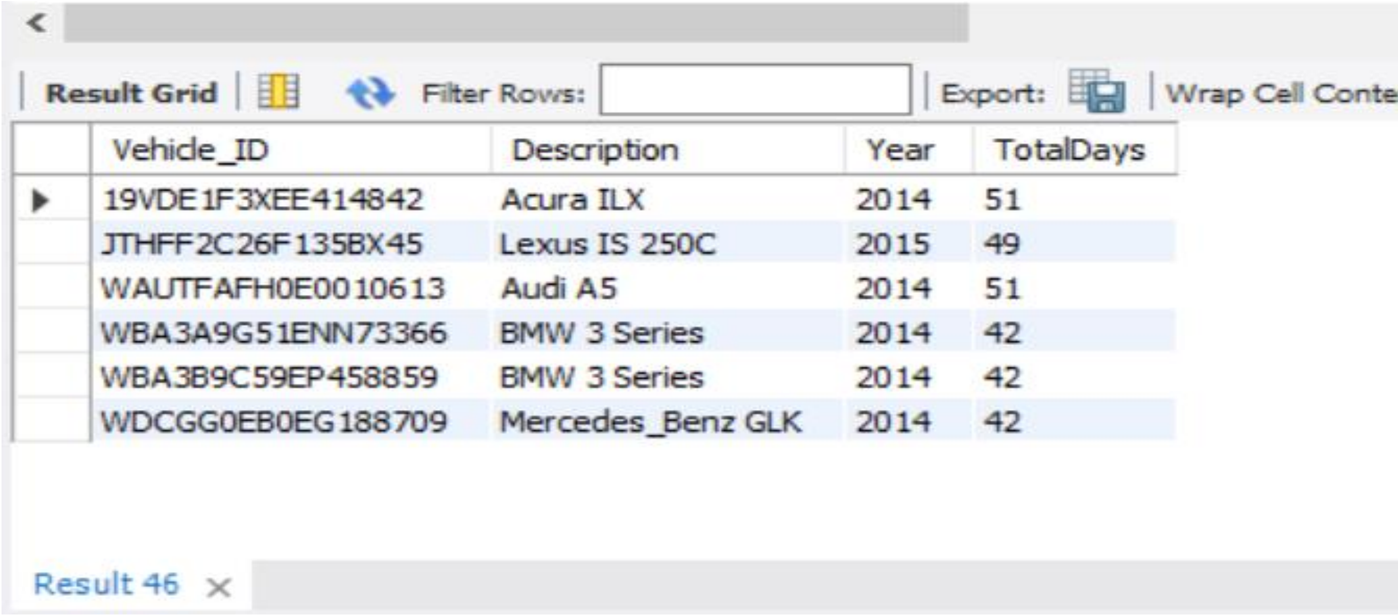
Question 5: Return all Compact(1) & Luxury(1) vehicles that were available for rent from June 01, 2019 until June 20, 2019. List VechicleID as VIN, Description, year, and how many days have been rented so far. You need to change the weeks into days.

```
SELECT
Vehicle.Vehicle_ID as VIN,Description,Year,StartDate,ReturnDate,
datediff(ReturnDate,StartDate) as TotalDays
FROM Vehicle
INNER JOIN RENTAL
```

```
on VEHICLE.Vehicle_ID = RENTAL.Vehicle_ID
WHERE Type = '1' and Category = '1'
AND StartDate NOT BETWEEN '2019-06-01' and '2019-06-20'
AND ReturnDate  NOT BETWEEN '2019-06-01' and '2019-06-20';
```

```
Create table AVAILABILITY(Vehicle_Id varchar(20)  not
null,Description varchar(30),Year int,StartDate date,ReturnDate
date,TotalDays int,foreign key (Vehicle_Id) references VEHICLE
(Vehicle_ID));
```

```
SELECT
VEHICLE.Vehicle_ID,VEHICLE.Description,VEHICLE.Year,Sum(T
otalDays) as TotalDays
FROM VEHICLE,AVAILABILITY
WHERE VEHICLE.Vehicle_ID = AVAILABILITY.Vehicle_Id
GROUP BY VEHICLE.Vehicle_ID;
```



Vehicle_ID	Description	Year	TotalDays
19VDE1F3XEE414842	Acura ILX	2014	51
JTHFF2C26F135BX45	Lexus IS 250C	2015	49
WAUTFAFH0E0010613	Audi A5	2014	51
WBA3A9G51ENN73366	BMW 3 Series	2014	42
WBA3B9C59EP458859	BMW 3 Series	2014	42
WDCGG0EB0EG188709	Mercedes_Benz GLK	2014	42

Total number of records returned were 6

For the above question, the idea implemented was to first select all the category and luxury vehicles that were available from 01/06/19 to 20/06/19 and a table was created based on the obtained results which was call as AVAILABILITY.

Now, from the created table the total days was calculated and all the compact, luxurious vehicles which were available from 01/06/19 to 20/06/19 were returned.

Total number of record returned are 6

Question 6: Return a list with the remaining balance for the customer with the id ‘221’. List customer name, and the balance.

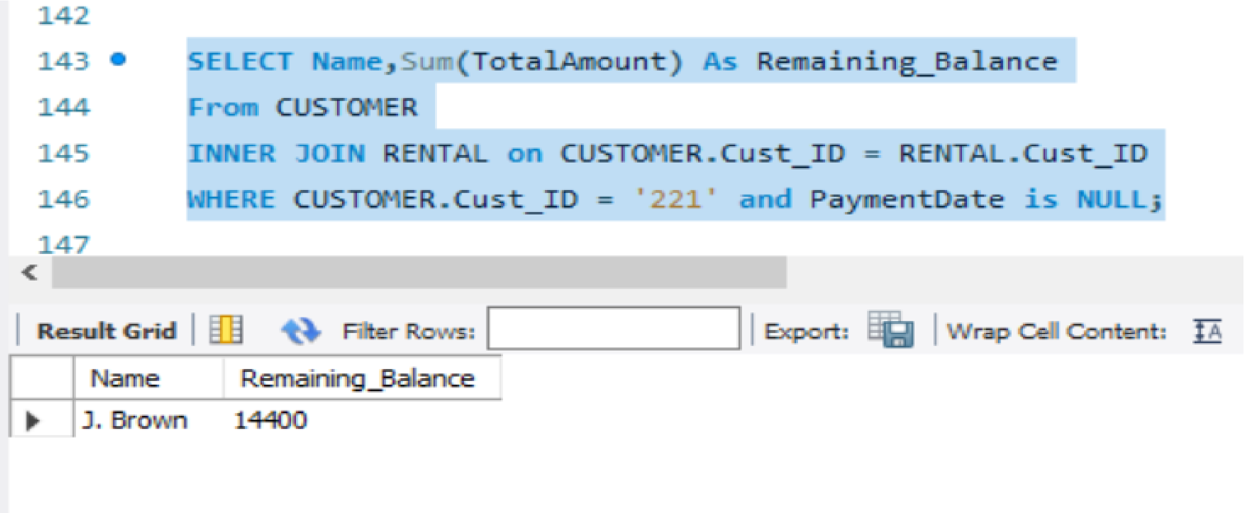
SELECT Name,Sum(TotalAmount) As Remaining_Balance

From CUSTOMER

INNER JOIN RENTAL

on CUSTOMER.Cust_ID = RENTAL.Cust_ID

WHERE CUSTOMER.Cust_ID = '221' and PaymentDate is NULL;



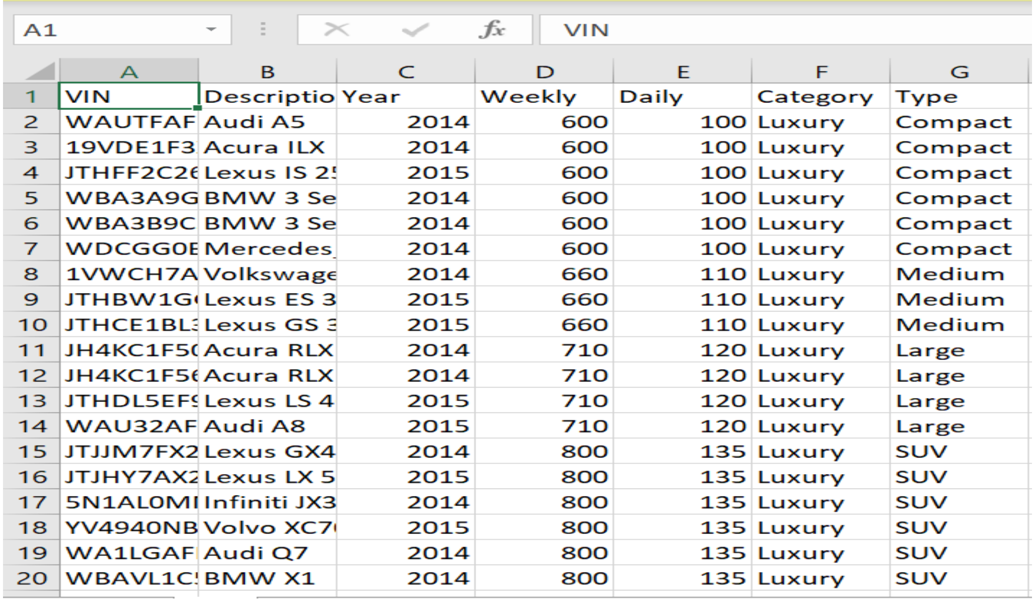
```
SELECT Name,Sum(TotalAmount) As Remaining_Balance
From CUSTOMER
INNER JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID
WHERE CUSTOMER.Cust_ID = '221' and PaymentDate is NULL;
```

Name	Remaining_Balance
J. Brown	14400

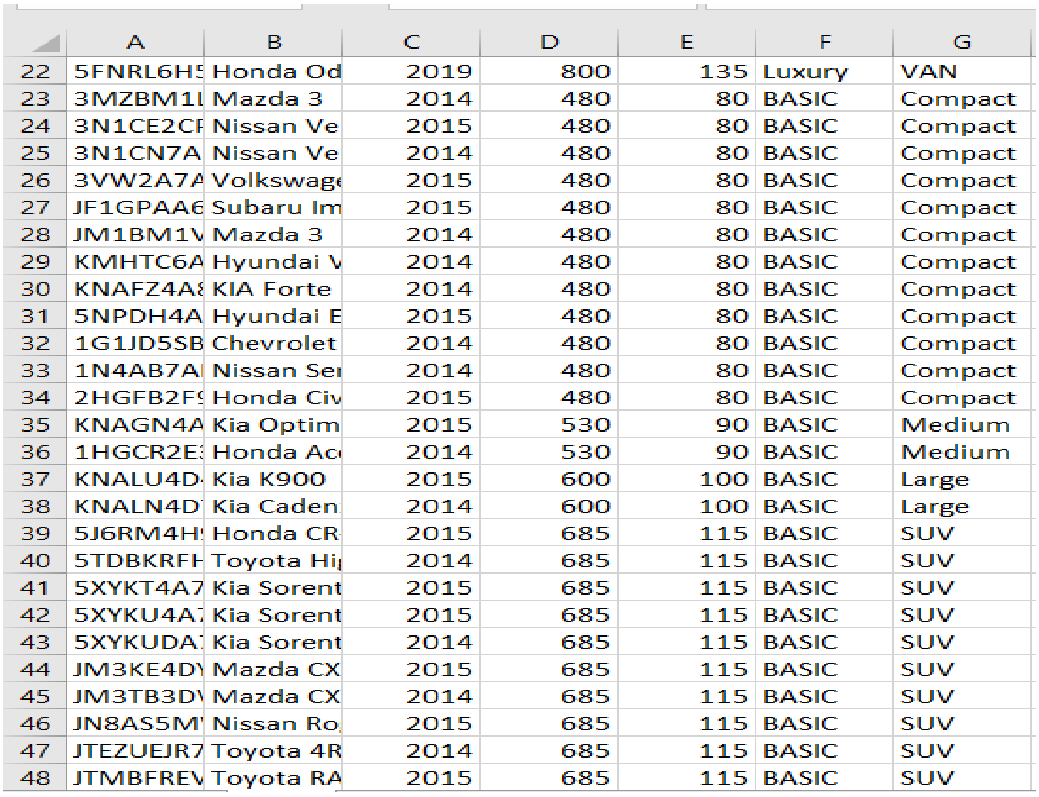
Total number of records returned was 1

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.

```
SELECT VEHICLE.Vehicle_ID as VIN, Description,Year, Weekly,Daily,

CASE

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

ELSE 'Luxury'

END AS 'Category',

CASE

WHEN RATE.Type = '1' THEN 'Compact'

WHEN RATE.Type = '2' THEN 'Medium'

WHEN RATE.Type = '3' THEN 'Large'

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

WHEN RATE.Type = '5' THEN 'Truck'

ELSE 'VAN'

END AS Type

FROM VEHICLE

inner join RATE on VEHICLE.Type = RATE.Type and Vehicle.CATEGORY = RATE.Category

ORDER BY RATE.Category = 'BASIC', Vehicle.Type ;
```



	A	B	C	D	E	F	G
1	VIN	Descriptio	Year	Weekly	Daily	Category	Type
2	WAUTFAF	Audi A5	2014	600	100	Luxury	Compact
3	19VDE1F3	Acura ILX	2014	600	100	Luxury	Compact
4	JTHFF2C2(	Lexus IS 2!	2015	600	100	Luxury	Compact
5	WBA3A9G	BMW 3 Se	2014	600	100	Luxury	Compact
6	WBA3B9C	BMW 3 Se	2014	600	100	Luxury	Compact
7	WDCGG0E	Mercedes	2014	600	100	Luxury	Compact
8	1VWCH7A	Volkswage	2014	660	110	Luxury	Medium
9	JTHBW1G(	Lexus ES 3	2015	660	110	Luxury	Medium
10	JTHCE1BL3	Lexus GS 3	2015	660	110	Luxury	Medium
11	JH4KC1F5(	Acura RLX	2014	710	120	Luxury	Large
12	JH4KC1F5(	Acura RLX	2014	710	120	Luxury	Large
13	JTHDL5EF!	Lexus LS 4	2015	710	120	Luxury	Large
14	WAU32AF	Audi A8	2015	710	120	Luxury	Large
15	JTJJM7FX2	Lexus GX4	2014	800	135	Luxury	SUV
16	JTJHY7AX2	Lexus LX 5	2015	800	135	Luxury	SUV
17	5N1AL0MI	Infiniti JX3	2014	800	135	Luxury	SUV
18	YV4940NB	Volvo XC7(	2015	800	135	Luxury	SUV
19	WA1LGAF	Audi Q7	2014	800	135	Luxury	SUV
20	WBAVL1C!	BMW X1	2014	800	135	Luxury	SUV

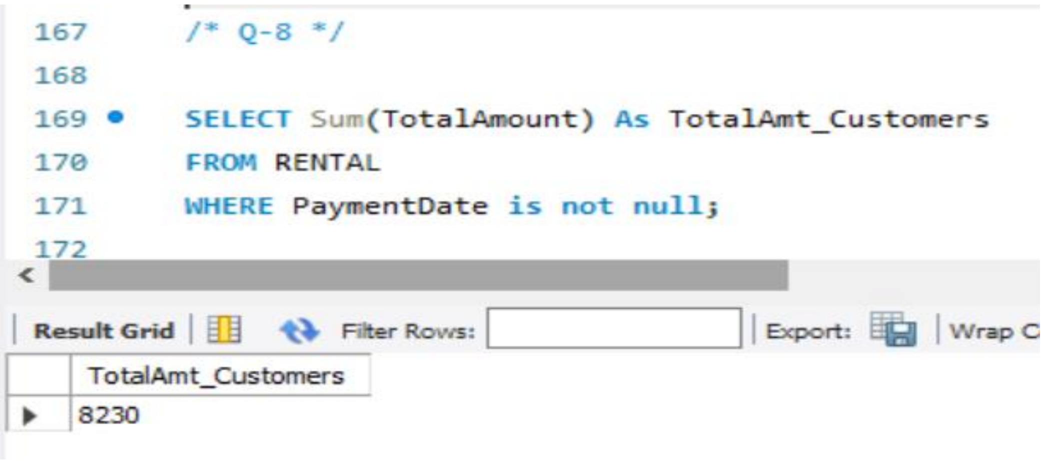


	A	B	C	D	E	F	G
22	5FNRL6H5	Honda Od	2019	800	135	Luxury	VAN
23	3MZBM1L	Mazda 3	2014	480	80	BASIC	Compact
24	3N1CE2CF	Nissan Ve	2015	480	80	BASIC	Compact
25	3N1CN7A	Nissan Ve	2014	480	80	BASIC	Compact
26	3VW2A7A	Volkswage	2015	480	80	BASIC	Compact
27	JF1GPAA6	Subaru Im	2015	480	80	BASIC	Compact
28	JM1BM1V	Mazda 3	2014	480	80	BASIC	Compact
29	KMHTC6A	Hyundai V	2014	480	80	BASIC	Compact
30	KNAFZ4A8	KIA Forte	2014	480	80	BASIC	Compact
31	5NPDH4A	Hyundai E	2015	480	80	BASIC	Compact
32	1G1JD5SB	Chevrolet	2014	480	80	BASIC	Compact
33	1N4AB7AI	Nissan Se	2014	480	80	BASIC	Compact
34	2HGFB2F9	Honda Civ	2015	480	80	BASIC	Compact
35	KNAGN4A	Kia Optim	2015	530	90	BASIC	Medium
36	1HGCR2E3	Honda Ac	2014	530	90	BASIC	Medium
37	KNALU4D	Kia K900	2015	600	100	BASIC	Large
38	KNALN4D	Kia Caden	2014	600	100	BASIC	Large
39	5J6RM4H!	Honda CR	2015	685	115	BASIC	SUV
40	5TDBKRFH	Toyota Hi	2014	685	115	BASIC	SUV
41	5XYKT4A7	Kia Sorent	2015	685	115	BASIC	SUV
42	5XYKU4A7	Kia Sorent	2015	685	115	BASIC	SUV
43	5XYKUDA	Kia Sorent	2014	685	115	BASIC	SUV
44	JM3KE4DY	Mazda CX	2015	685	115	BASIC	SUV
45	JM3TB3DV	Mazda CX	2014	685	115	BASIC	SUV
46	JN8AS5M	Nissan Ro	2015	685	115	BASIC	SUV
47	JTEZUEJR7	Toyota 4R	2014	685	115	BASIC	SUV
48	JTMBFREV	Toyota RA	2015	685	115	BASIC	SUV

Total Records returned were 61 based on Category(Luxury First) and Type based on the number

Question 8: What is the total of money that customers paid to us until today?

```
SELECT Sum(TotalAmount) As TotalAmt_Customers
FROM RENTAL
WHERE PaymentDate is not null;
```



```
/* Q-8 */

SELECT Sum(TotalAmount) As TotalAmt_Customers
FROM RENTAL
WHERE PaymentDate is not null;
```

```
TotalAmt_Customers
8230
```

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.

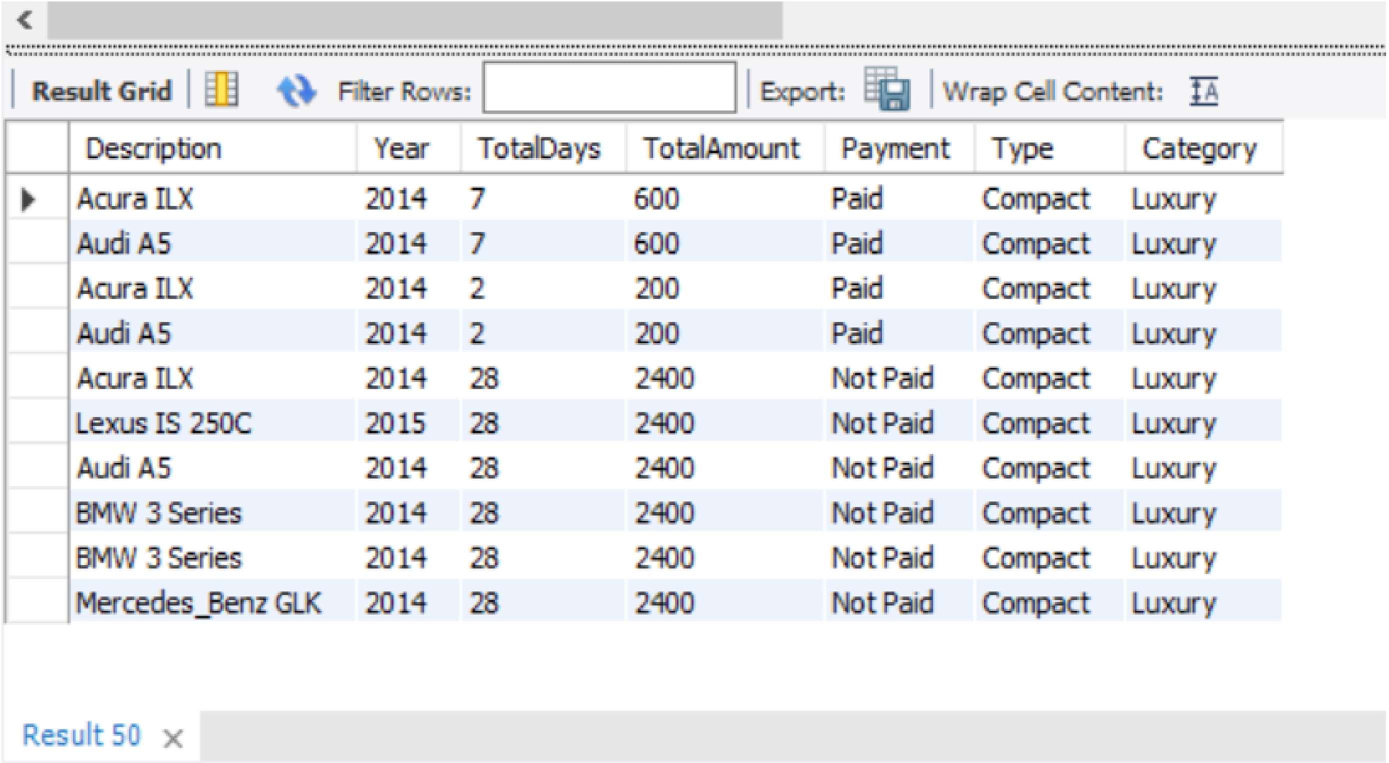
```
SELECT Description,Year,datediff(ReturnDate,StartDate) as TotalDays,TotalAmount,
CASE
WHEN PaymentDate is not null then 'Paid'
```

```
ELSE 'Not Paid'
END AS 'Payment',
CASE
WHEN VEHICLE.Type = '1' THEN 'Compact'
ELSE 'null'
END AS 'Type',
CASE
WHEN VEHICLE.Category = '1' THEN 'Luxury'
END AS Category
FROM RENTAl
INNER JOIN VEHICLE on RENTAL.Vehicle_ID = VEHICLE.Vehicle_ID
INNER JOIN CUSTOMER on RENTAL.Cust_ID = CUSTOMER.Cust_ID
WHERE CUSTOMER.Name = 'J. Brown'
ORDER BY StartDate;
```

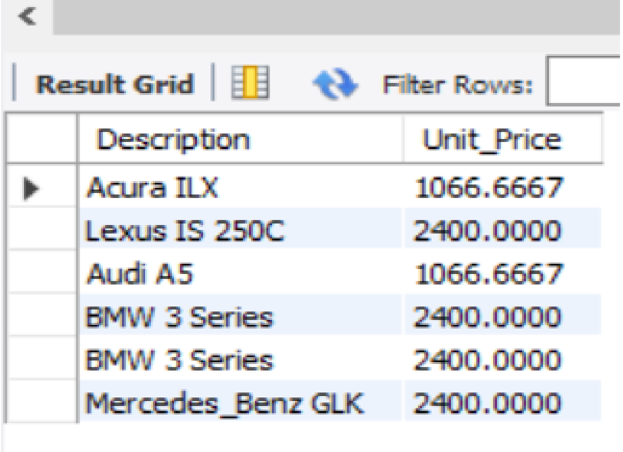
For Calculating the unit price for every rental, separate select statement query was written since the column length was not similar

The Query is written below:-

```
SELECT Description, AVG(TotalAmount) as Unit_Price
FROM RENTAL INNER JOIN VEHICLE
on RENTAL.Vehicle_ID = VEHICLE.Vehicle_ID
WHERE Cust_ID = '221'
GROUP BY RENTAL.Vehicle_ID;
```



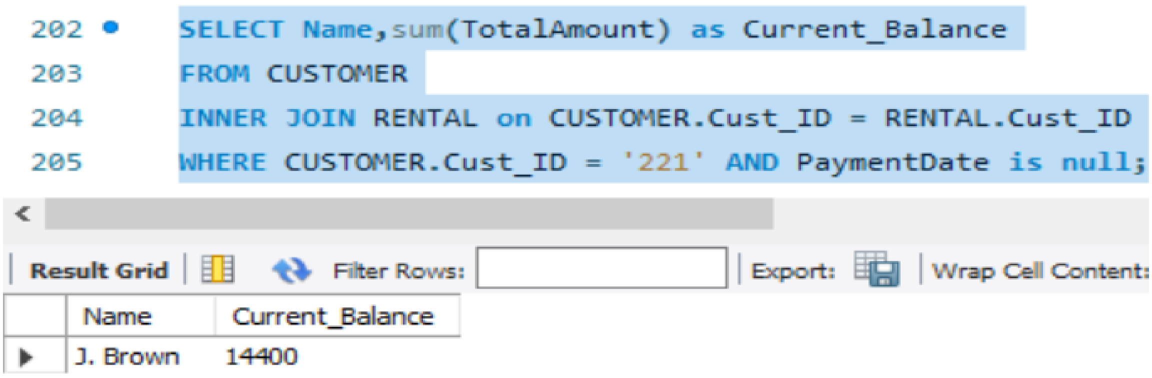
Description	Year	TotalDays	TotalAmount	Payment	Type	Category
Acura ILX	2014	7	600	Paid	Compact	Luxury
Audi A5	2014	7	600	Paid	Compact	Luxury
Acura ILX	2014	2	200	Paid	Compact	Luxury
Audi A5	2014	2	200	Paid	Compact	Luxury
Acura ILX	2014	28	2400	Not Paid	Compact	Luxury
Lexus IS 250C	2015	28	2400	Not Paid	Compact	Luxury
Audi A5	2014	28	2400	Not Paid	Compact	Luxury
BMW 3 Series	2014	28	2400	Not Paid	Compact	Luxury
BMW 3 Series	2014	28	2400	Not Paid	Compact	Luxury
Mercedes_Benz GLK	2014	28	2400	Not Paid	Compact	Luxury



Description	Unit_Price
Acura ILX	1066.6667
Lexus IS 250C	2400.0000
Audi A5	1066.6667
BMW 3 Series	2400.0000
BMW 3 Series	2400.0000
Mercedes_Benz GLK	2400.0000

Question 9-b: For the same customer return the current balance.

```
SELECT Name,sum(TotalAmount) as Current_Balance
FROM CUSTOMER
INNER JOIN RENTAL on CUSTOMER.Cust_ID =
RENTAL.Cust_ID
WHERE CUSTOMER.Cust_ID = '221' AND PaymentDate is null;
```

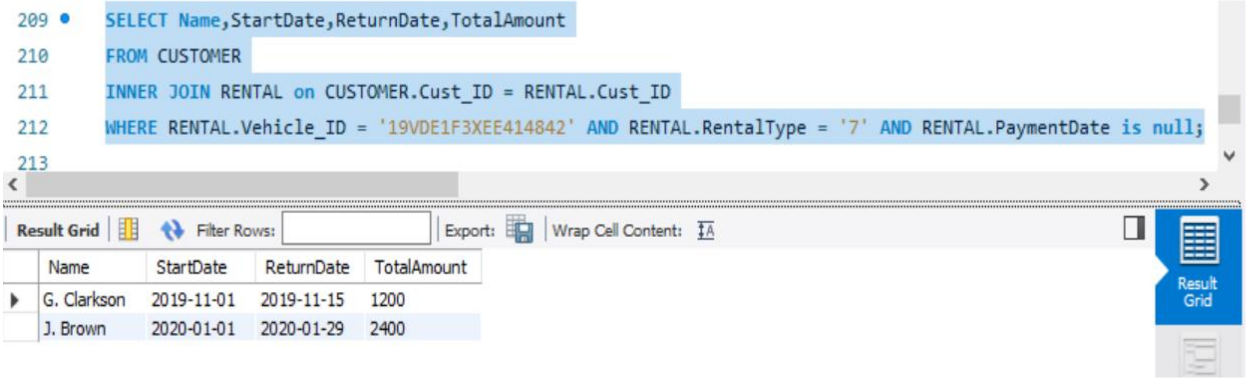


```
SELECT Name,sum(TotalAmount) as Current_Balance
FROM CUSTOMER
INNER JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID
WHERE CUSTOMER.Cust_ID = '221' AND PaymentDate is null;
```

Name	Current_Balance
J. Brown	14400

Question 10: Retrieve all weekly rentals for the vechicleID ‘19VDE1F3XEE414842’ that are not paid yet. List the Customer Name, the start and return date, and the amount.

```
SELECT Name,StartDate,ReturnDate,TotalAmount
FROM CUSTOMER
INNER JOIN RENTAL
ON CUSTOMER.Cust_ID = RENTAL.Cust_ID
WHERE RENTAL.Vehicle_ID = '19VDE1F3XEE414842' AND
RENTAL.RentalType = '7' AND RENTAL.PaymentDate is null;
```



```
SELECT Name,StartDate,ReturnDate,TotalAmount
FROM CUSTOMER
INNER JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID
WHERE RENTAL.Vehicle_ID = '19VDE1F3XEE414842' AND RENTAL.RentalType = '7' AND RENTAL.PaymentDate is null;
```

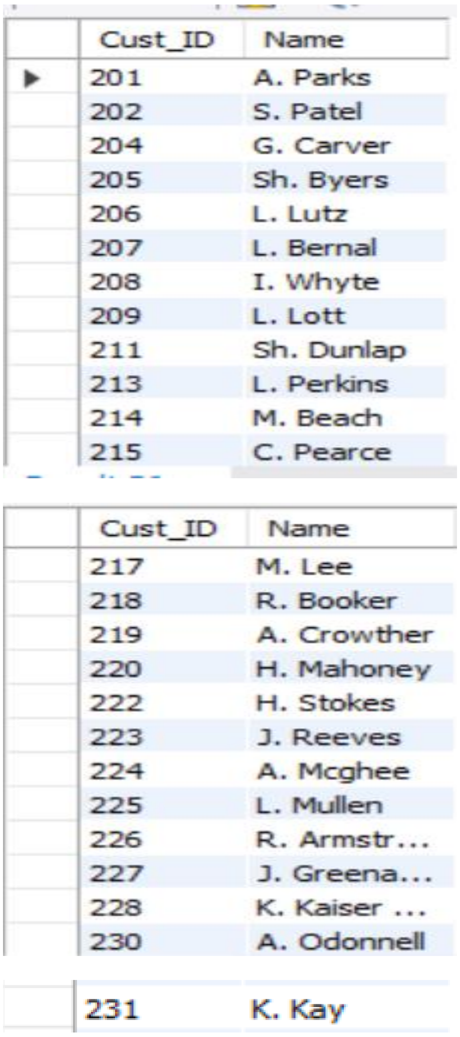
Name	StartDate	ReturnDate	TotalAmount
G. Clarkson	2019-11-01	2019-11-15	1200
J. Brown	2020-01-01	2020-01-29	2400

Question 11: Return all customers that they never rent a vehicle.

```
SELECT CUSTOMER.Cust_ID,Name
FROM CUSTOMER
```

LEFT JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID

WHERE RENTAL.Cust_ID is null;



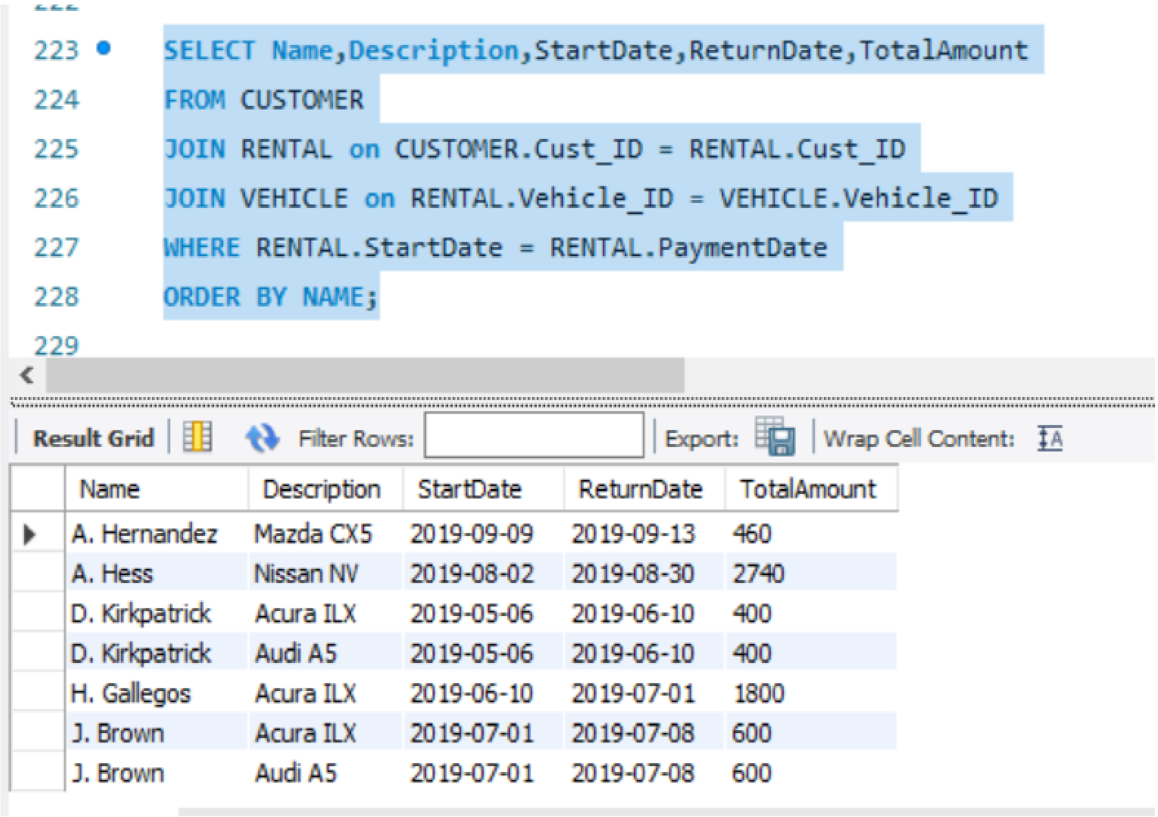
Cust_ID	Name
201	A. Parks
202	S. Patel
204	G. Carver
205	Sh. Byers
206	L. Lutz
207	L. Bernal
208	I. Whyte
209	L. Lott
211	Sh. Dunlap
213	L. Perkins
214	M. Beach
215	C. Pearce
217	M. Lee
218	R. Booker
219	A. Crowther
220	H. Mahoney
222	H. Stokes
223	J. Reeves
224	A. Mcghee
225	L. Mullen
226	R. Armstr...
227	J. Greena...
228	K. Kaiser ...
230	A. Odonnell
231	K. Kay

Total number of records returned were 25

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.

SELECT Name,Description,StartDate,ReturnDate,TotalAmount

```
FROM CUSTOMER
JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID
JOIN VEHICLE on RENTAL.Vehicle_ID = VEHICLE.Vehicle_ID
WHERE RENTAL.StartDate = RENTAL.PaymentDate
ORDER BY NAME;
```



```
SELECT Name,Description,StartDate,ReturnDate,TotalAmount
FROM CUSTOMER
JOIN RENTAL on CUSTOMER.Cust_ID = RENTAL.Cust_ID
JOIN VEHICLE on RENTAL.Vehicle_ID = VEHICLE.Vehicle_ID
WHERE RENTAL.StartDate = RENTAL.PaymentDate
ORDER BY NAME;
```

Name	Description	StartDate	ReturnDate	TotalAmount
A. Hernandez	Mazda CX5	2019-09-09	2019-09-13	460
A. Hess	Nissan NV	2019-08-02	2019-08-30	2740
D. Kirkpatrick	Acura ILX	2019-05-06	2019-06-10	400
D. Kirkpatrick	Audi A5	2019-05-06	2019-06-10	400
H. Gallegos	Acura ILX	2019-06-10	2019-07-01	1800
J. Brown	Acura ILX	2019-07-01	2019-07-08	600
J. Brown	Audi A5	2019-07-01	2019-07-08	600

Total Number of record returned were 7