Information and Database Management Systems I

(CIS 4301 UF Online)

Fall 2024

Instructor: Alexander Webber

TA: Kyuseo Park

Homework 1

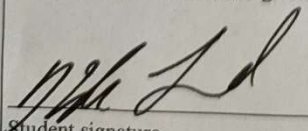
Printed Name:	Kyle Lund
UFID:	2958 1039
Email Address:	Kyle.Lund@ufl.edu

Instructions: Please provide your answers to the questions of the following pages in Word or handwritten on separate sheets of paper. Mark clearly to which question each answer belongs. Then convert or scan your work into PDF (the latter by using either a scanner or a suitable scanner app on your smartphone). Note that *only the PDF format* is allowed and that your submission must be a *single PDF file*. Finally, upload your PDF file into *Canvas* and follow the instructions there.

Note: All homework assignments are designed for a period of two, three, or even four weeks (see course deadline sheet). This means they cannot be solved in two or three hours but require a considerable amount of time and effort. Therefore, the first recommendation is to start with them as soon as they are posted. The second recommendation is to distribute the work on a homework assignment over the entire available period. The third recommendation is to submit the homework solutions *on time before the deadline*.

Pledge (Must be signed[1] according to the UF Honor Code):

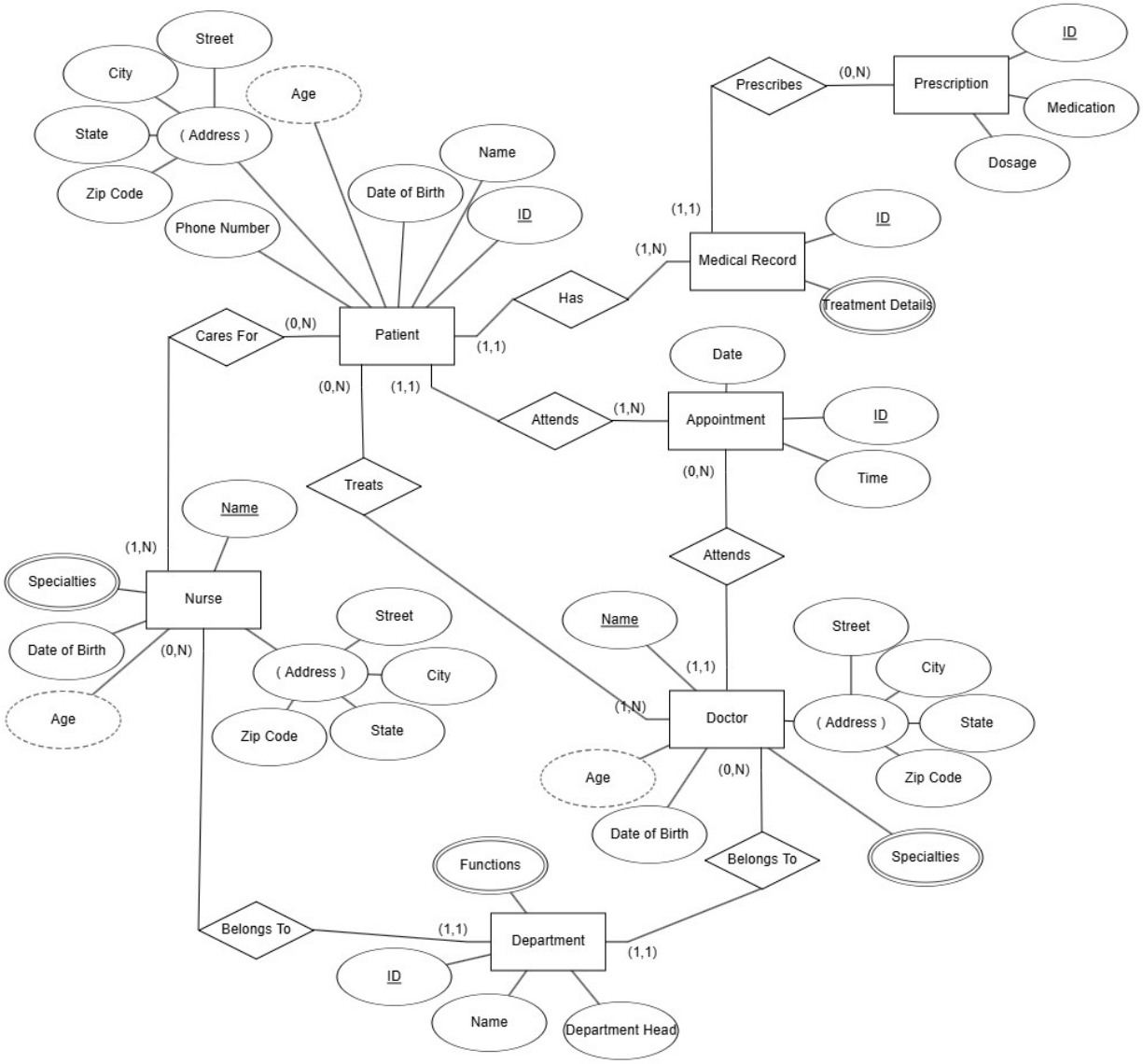
On my honor, I have neither given nor received unauthorized aid in doing this assignment.



Student signature

[1]Each student is obliged to print out this page, fill in the requested information in a handwritten and readable manner, make the *handwritten* signature, scan this page into PDF, and put this page as the first page of the PDF submission.

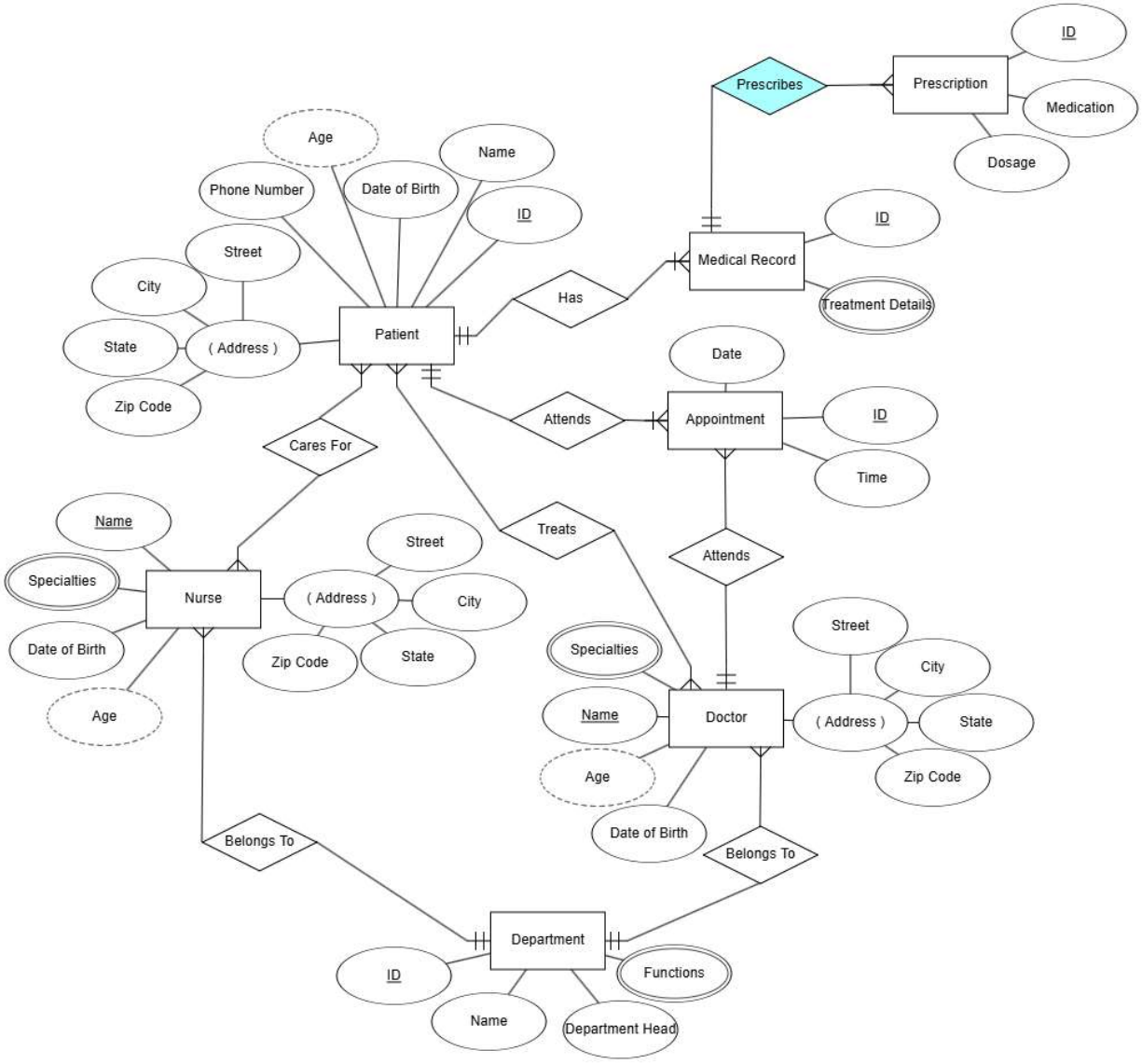
Question 1
Chen's



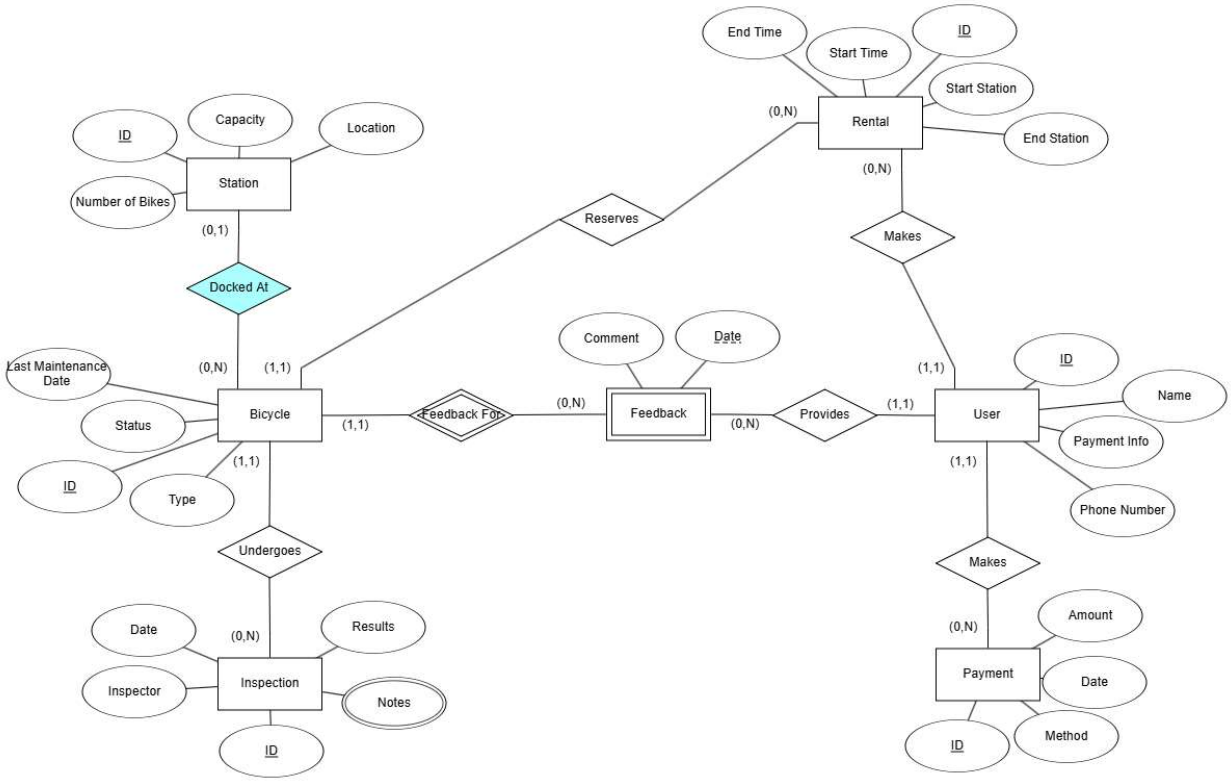
Note: I couldn't figure out how to make the double line that represents total participation. To make up for that, here's a list of the relationships which would have total participation:

- Every Appointment is attended to by both a doctor and a patient
- Every patient has a medical record, and each medical record is about a specific patient
- Every nurse and every doctor belong to a department
- Every prescription is derived from a medical record

Crow's Foot


ID
Prescribes
Prescription
Medication
Dosage
Age
Name
Phone Number
Date of Birth
ID
ID
Street
Medical Record
City
Has
Treatment Details
State
(Address)
Patient
Zip Code
Date
Attends
Appointment
ID
Cares For
Time
Name
Treats
Attends
Street
Specialties
Nurse
(Address)
City
Date of Birth
Zip Code
State
Specialties
Street
City
Name
Doctor
(Address)
State
Age
Zip Code
Date of Birth
Belongs To
Belongs To
Department
ID
Functions
Name
Department Head

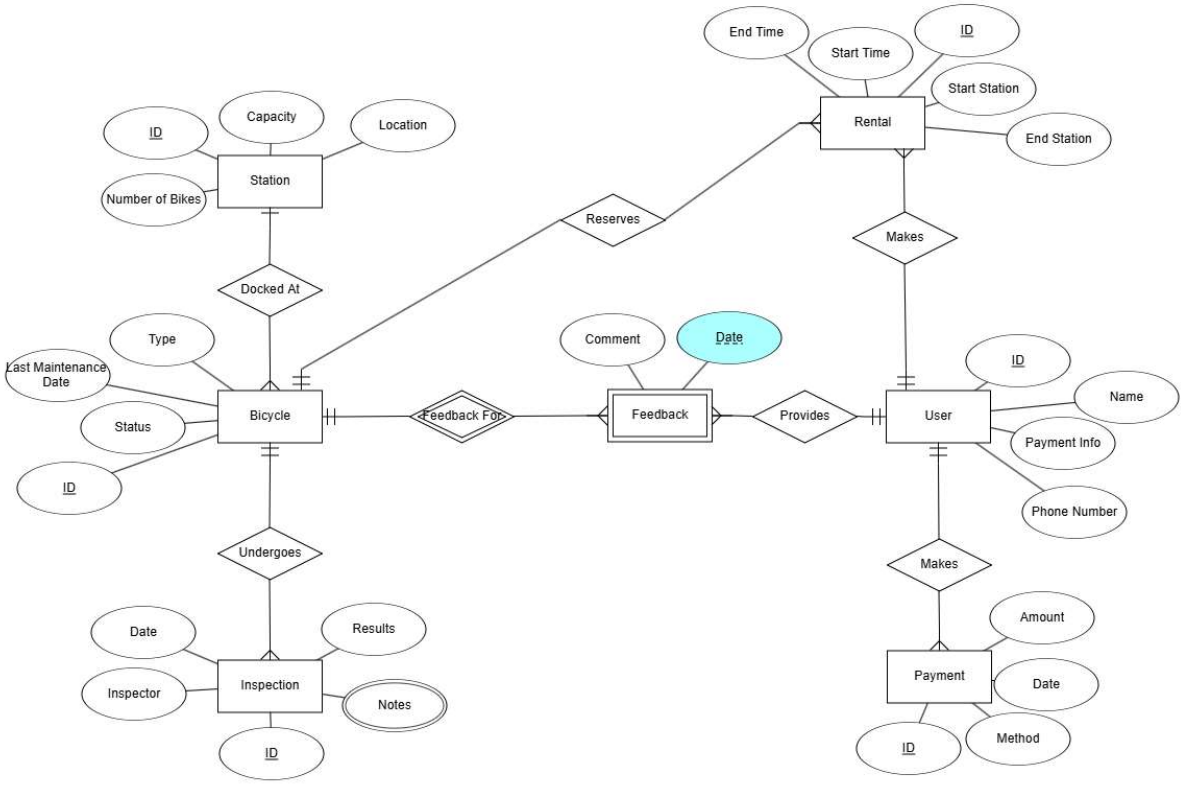
Question 2
Chen's



As before, here's the total participation lists:

- Each inspection is for one and only one bicycle
- Each rental is for one and only one bicycle (this is an assumption, would change if group rentals are involved) and is created by one and only one user
- Each payment is done by one and only one user
- Each feedback is for one and only one bicycle and is provided by one and only one user

Crow's Foot



Question 3

A) BankBranch is a weak entity set because the BranchNo attribute does not uniquely identify a member of this set. For example, Bank of America and Chase could both have branches with a BranchNo of 1. Only by the combination of the bank and the BranchNo can each branch be uniquely identified.

B) Each bank branch belongs to a bank and each bank has bank branches. The double line represents total participation and is not optional for the line from BankBranch towards Bank. The Bank to BankBranch line also represents a total participation, and this one is optional as Bank is strong entity set (also, it is theoretically possible for a bank that was just created or an online only bank to not have any branches).

C) Has (Customer – has – Account):
- Each account is associated with 1:n customers. The minimum being one is due to the total participation lines between the Account entity and the has relationship.
- Each customer has 0:m accounts, as indicated by the "m" and the single line.

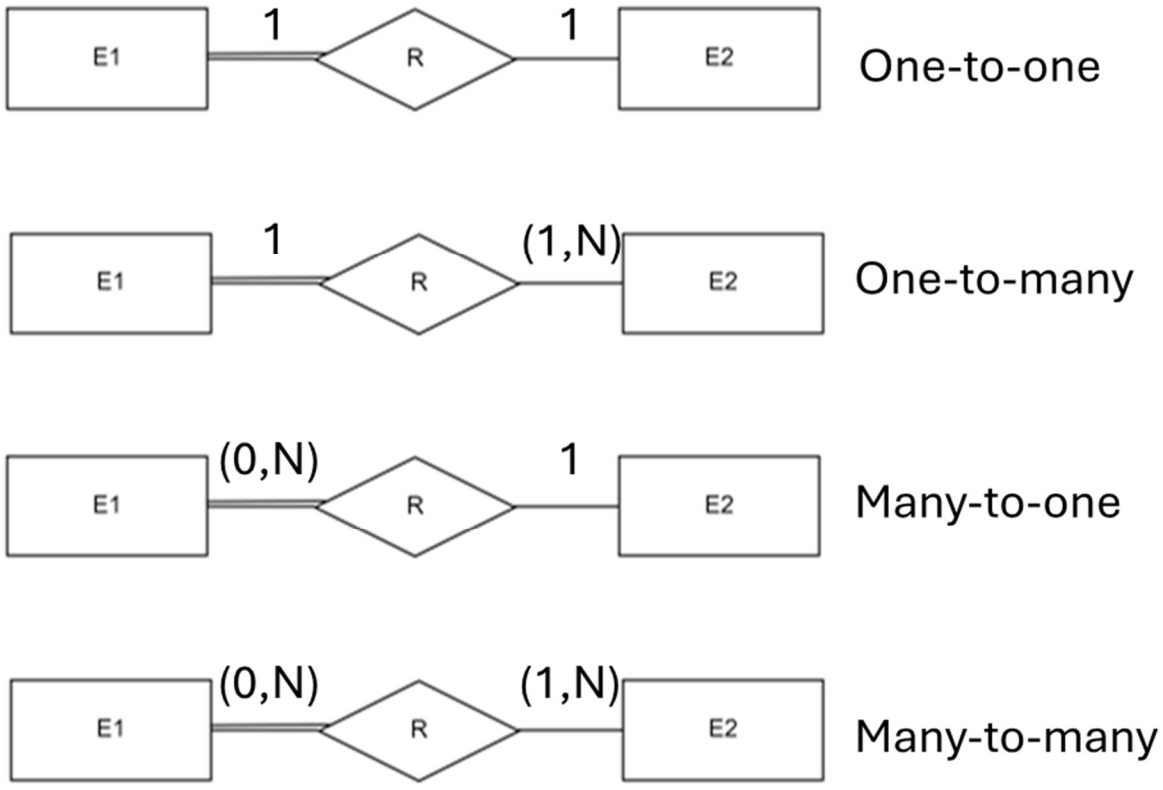
ManagesA (Account – ManagesA – BankBranch):
- Each account is managed by one and only one bank branch. This is indicated by the "1" between ManagesA and BankBranch. Total participation is also indicated by the double lines.
- Each BankBranch manages between 0:n accounts, as indicated by the "n" and the single line.

Takes (Customer – Takes – Loan):
- Each customer has between 0:m loans, as indicated by the “m” and the single line.
- Each loan is takes on by between 1:n customers, as indicated by the “n” and the double lines for total participation.

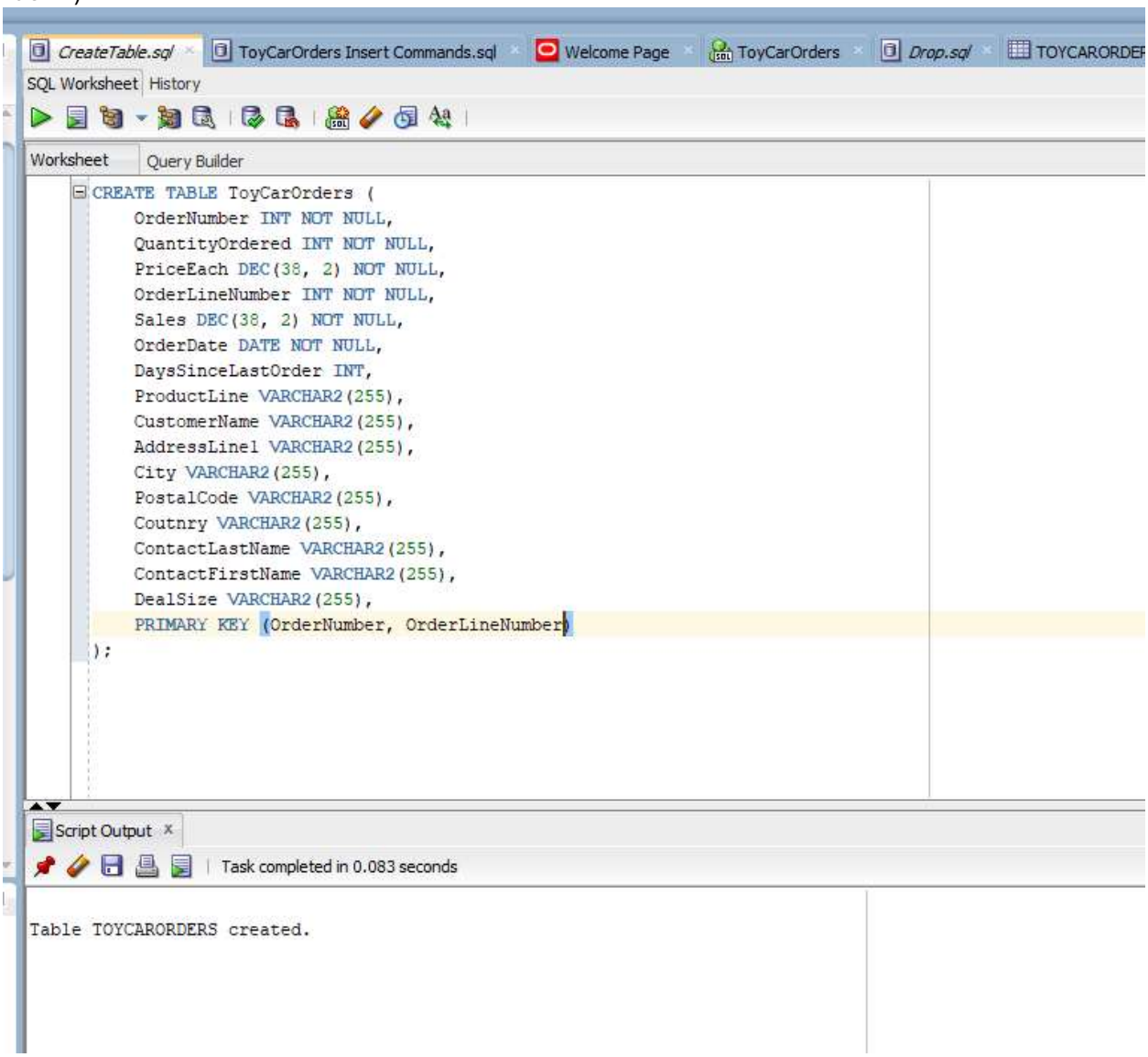
ManagesB (Loan – ManagesB – BankBranch):
- Each BankBranch manages between 0:n loans, as indicated by the “n”.
- Each Loan is managed by one and only one bank branch. This is indicated by the “1”. Total participation is also indicated by the double lines.

BelongsTo (BankBranch – BelongsTo – Bank):
- Each BankBranch belongs to one and only one bank branch. This is indicated by the “1”. Total participation is also indicated by the double lines.
- Each Bank owns between 1:n BankBranches as indicated by the n, total participation is indicated by the double lines.

D)



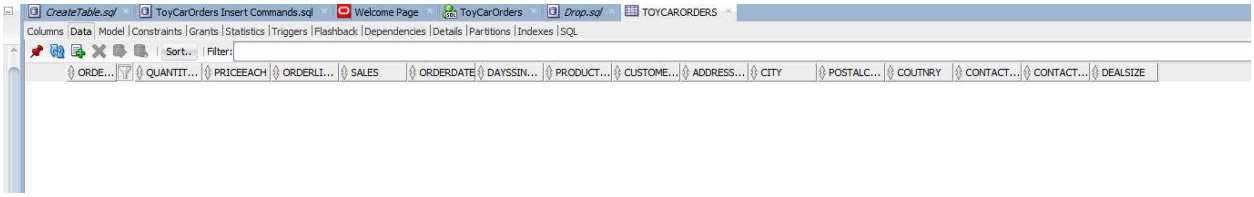
Question 4, Part A
Task 1)



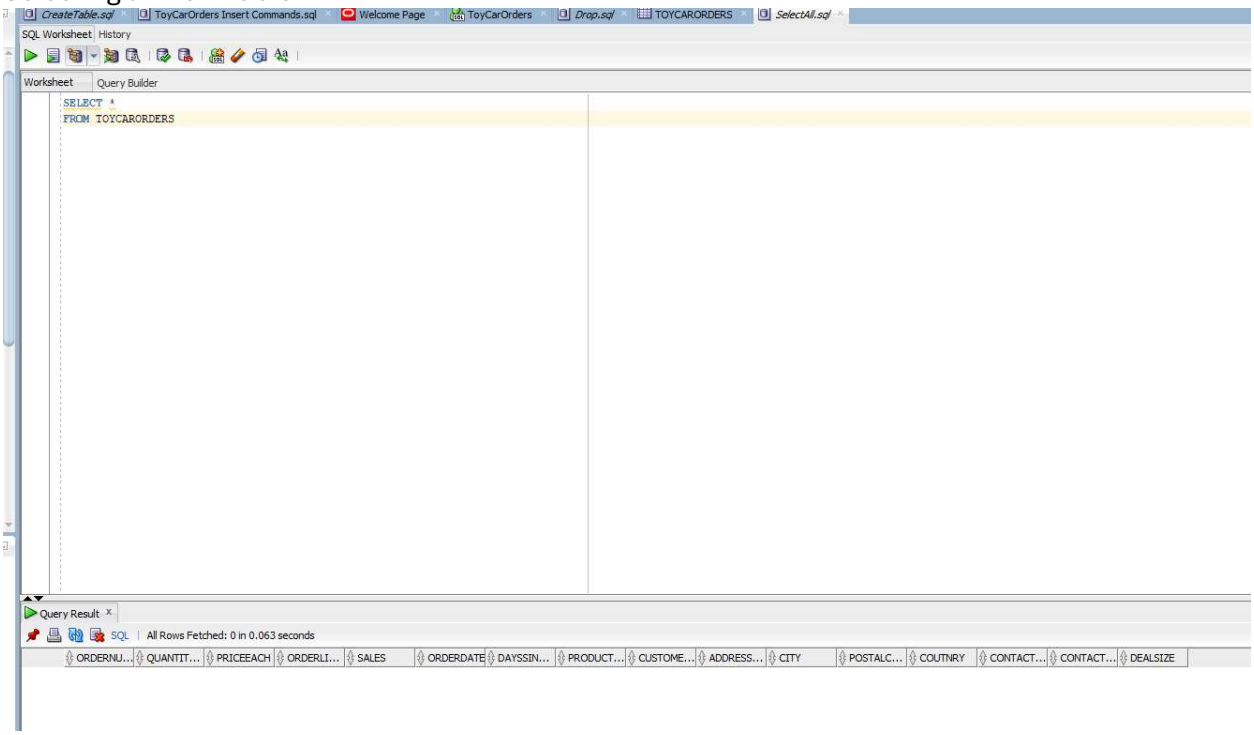
A composite key with OrderNumber and OrderLineNumber was selected as the primary key. OrderNumber was chosen as the Kaggle documentation states: “This column represents the unique identification number assigned to each order”. OrderLineNumber was combined with this because based off of the insert commands each tuple corresponds to a line number on a specific order. So both of these are needed in combination to create a unique key.

Task 2)

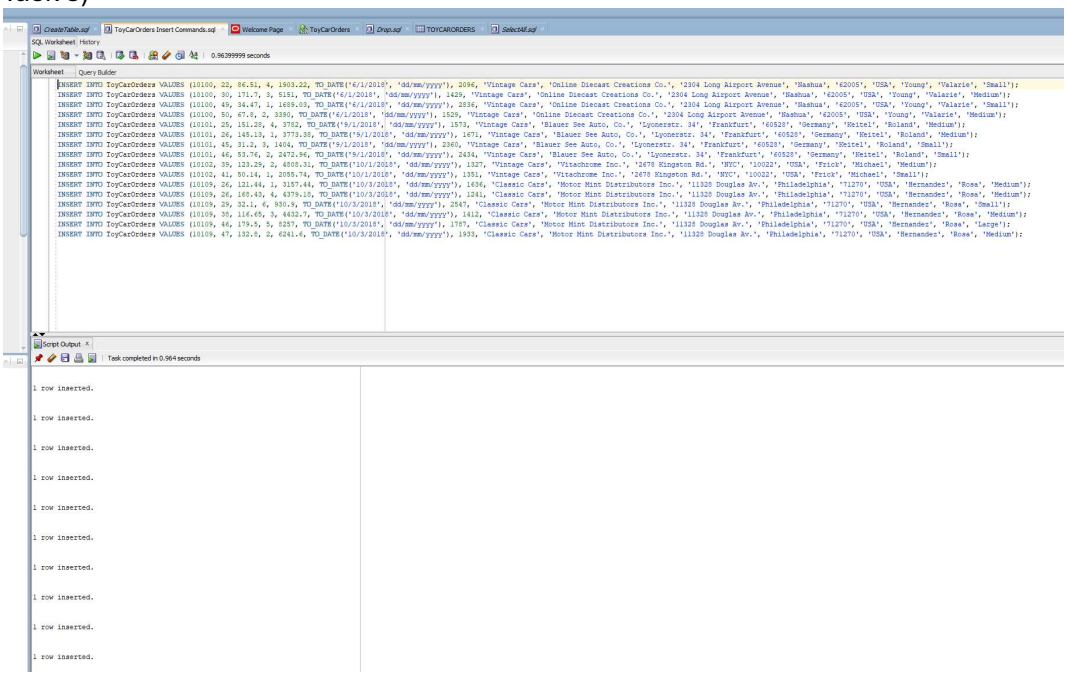
View of table



Selecting all from table



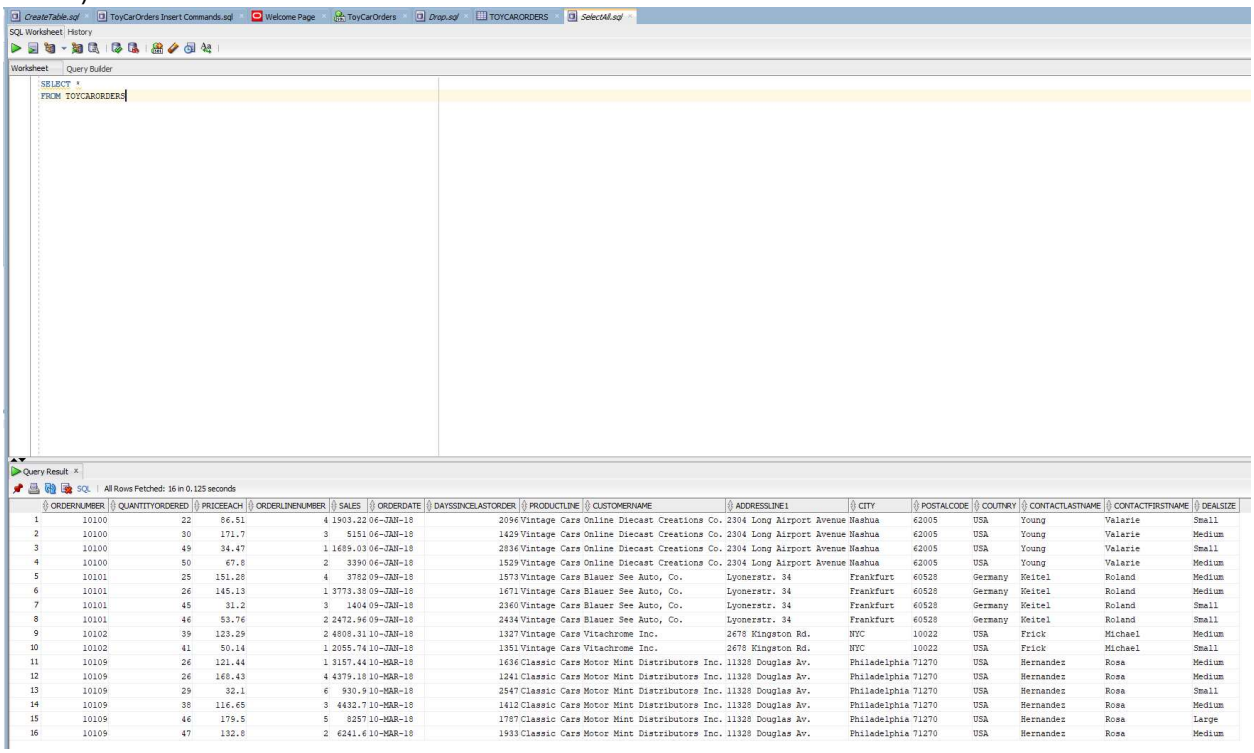
Task 3)



```
INSERT INTO ToyCarOrders VALUES (10100, 22, 86.51, 4, 1903.22, TO_DATE('6/1/2018', 'dd/mm/yyyy'), 2096, 'Vintage Cars', 'Online Diecast Creations Co.', '2304 Long Airport Avenue', 'Nashua', '62005', 'USA', 'Young', 'Valarie', 'Small');
INSERT INTO ToyCarOrders VALUES (10100, 30, 171.7, 3, 5151, TO_DATE('6/1/2018', 'dd/mm/yyyy'), 1429, 'Vintage Cars', 'Online Diecast Creations Co.', '2304 Long Airport Avenue', 'Nashua', '62005', 'USA', 'Young', 'Valarie', 'Medium');
INSERT INTO ToyCarOrders VALUES (10100, 49, 34.47, 1, 1689.03, TO_DATE('6/1/2018', 'dd/mm/yyyy'), 2836, 'Vintage Cars', 'Online Diecast Creations Co.', '2304 Long Airport Avenue', 'Nashua', '62005', 'USA', 'Young', 'Valarie', 'Small');
INSERT INTO ToyCarOrders VALUES (10100, 50, 67.8, 2, 3390, TO_DATE('6/1/2018', 'dd/mm/yyyy'), 1529, 'Vintage Cars', 'Online Diecast Creations Co.', '2304 Long Airport Avenue', 'Nashua', '62005', 'USA', 'Young', 'Valarie', 'Medium');
INSERT INTO ToyCarOrders VALUES (10101, 25, 151.28, 4, 3782, TO_DATE('9/1/2018', 'dd/mm/yyyy'), 1573, 'Vintage Cars', 'Blauer See Auto, Co.', 'Lyonerstr. 34', 'Frankfurt', '60528', 'Germany', 'Keitel', 'Roland', 'Medium');
INSERT INTO ToyCarOrders VALUES (10101, 26, 145.13, 1, 3773.38, TO_DATE('9/1/2018', 'dd/mm/yyyy'), 1671, 'Vintage Cars', 'Blauer See Auto, Co.', 'Lyonerstr. 34', 'Frankfurt', '60528', 'Germany', 'Keitel', 'Roland', 'Medium');
INSERT INTO ToyCarOrders VALUES (10101, 45, 31.2, 3, 1404, TO_DATE('9/1/2018', 'dd/mm/yyyy'), 2360, 'Vintage Cars', 'Blauer See Auto, Co.', 'Lyonerstr. 34', 'Frankfurt', '60528', 'Germany', 'Keitel', 'Roland', 'Small');
INSERT INTO ToyCarOrders VALUES (10101, 46, 53.76, 2, 2472.96, TO_DATE('9/1/2018', 'dd/mm/yyyy'), 2434, 'Vintage Cars', 'Blauer See Auto, Co.', 'Lyonerstr. 34', 'Frankfurt', '60528', 'Germany', 'Keitel', 'Roland', 'Small');
INSERT INTO ToyCarOrders VALUES (10102, 39, 123.29, 2, 4808.31, TO_DATE('10/1/2018', 'dd/mm/yyyy'), 1327, 'Vintage Cars', 'Vitachrome Inc.', '2678 Kingston Rd.', 'NYC', '10022', 'USA', 'Frick', 'Michael', 'Medium');
INSERT INTO ToyCarOrders VALUES (10102, 41, 50.14, 1, 2055.74, TO_DATE('10/1/2018', 'dd/mm/yyyy'), 1351, 'Vintage Cars', 'Vitachrome Inc.', '2678 Kingston Rd.', 'NYC', '10022', 'USA', 'Frick', 'Michael', 'Small');
INSERT INTO ToyCarOrders VALUES (10109, 26, 121.44, 1, 3157.44, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 1636, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Medium');
INSERT INTO ToyCarOrders VALUES (10109, 26, 168.43, 4, 4379.18, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 1241, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Medium');
INSERT INTO ToyCarOrders VALUES (10109, 29, 32.1, 6, 930.9, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 2547, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Small');
INSERT INTO ToyCarOrders VALUES (10109, 38, 116.65, 3, 4432.7, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 1412, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Medium');
INSERT INTO ToyCarOrders VALUES (10109, 46, 179.5, 5, 8257, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 1787, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Large');
INSERT INTO ToyCarOrders VALUES (10109, 47, 132.8, 2, 6241.6, TO_DATE('10/3/2018', 'dd/mm/yyyy'), 1933, 'Classic Cars', 'Motor Mint Distributors Inc.', '11328 Douglas Av.', 'Philadelphia', '71270', 'USA', 'Hernandez', 'Rosa', 'Medium');
```

```
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
1 row inserted.
```

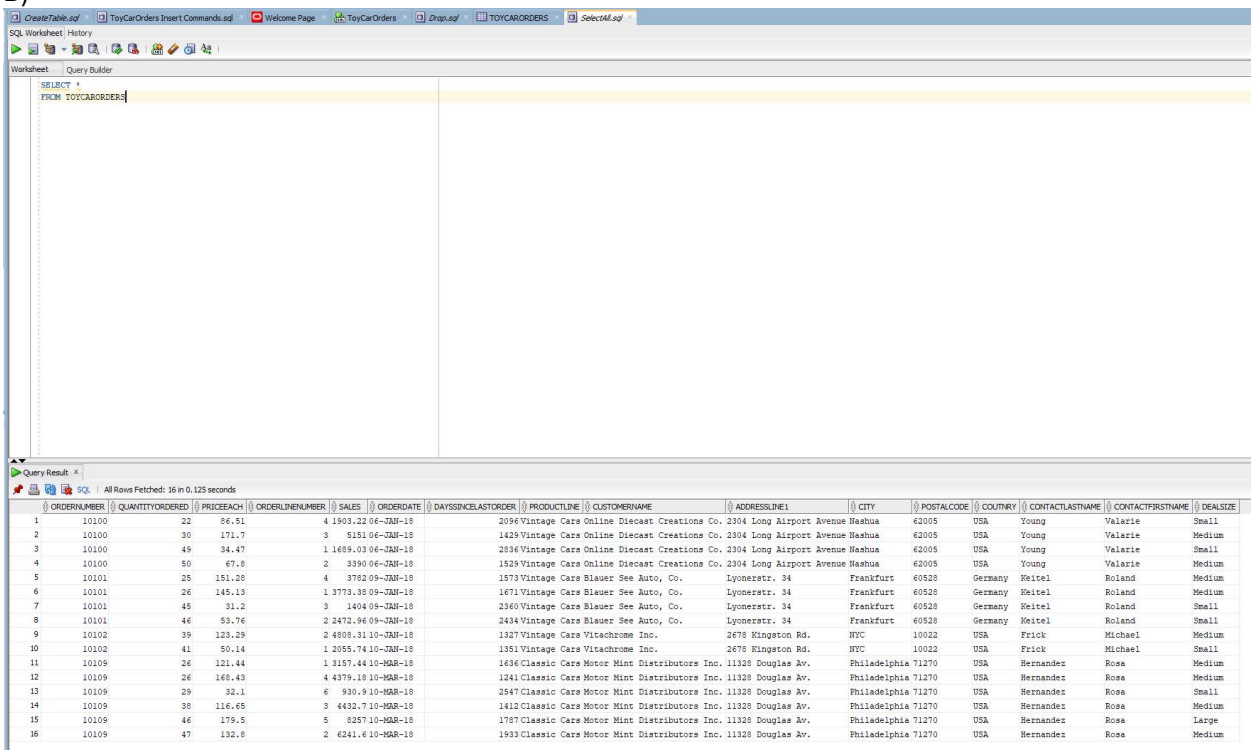
Task 4)



```
SELECT *
FROM TOYCARORDERS
```

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	DAYSSINCELASTORDER	PRODUCTLINE	CUSTOMERNAME	ADDRESSLINE1	CITY	POSTALCODE	COUNTRY	CONTACTLASTNAME	CONTACTFIRSTNAME	DEALSIZE
1	10100	22	86.51	4	1903.22	06-JAN-18	2096	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Small
2	10100	30	171.7	3	5151	06-JAN-18	1429	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Medium
3	10100	49	34.47	1	1689.03	06-JAN-18	2836	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Small
4	10100	50	67.8	2	3390	06-JAN-18	1529	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Medium
5	10101	25	151.28	4	3782	09-JAN-18	1573	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Medium
6	10101	26	145.13	1	3773.38	09-JAN-18	1671	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Medium
7	10101	45	31.2	3	1404	09-JAN-18	2360	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Small
8	10101	46	53.76	2	2472.96	09-JAN-18	2434	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Small
9	10102	39	123.29	2	4808.31	10-JAN-18	1327	Vintage Cars	Vitachrome Inc.	2678 Kingston Rd.	NYC	10022	USA	Frick	Michael	Medium
10	10102	41	50.14	1	2055.74	10-JAN-18	1351	Vintage Cars	Vitachrome Inc.	2678 Kingston Rd.	NYC	10022	USA	Frick	Michael	Small
11	10109	26	121.44	1	3157.44	10-MAR-18	1636	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
12	10109	26	168.43	4	4379.18	10-MAR-18	1241	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
13	10109	29	32.1	6	930.9	10-MAR-18	2547	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Small
14	10109	38	116.65	3	4432.7	10-MAR-18	1412	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
15	10109	46	179.5	5	8257	10-MAR-18	1787	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Large
16	10109	47	132.8	2	6241.6	10-MAR-18	1933	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium

B)



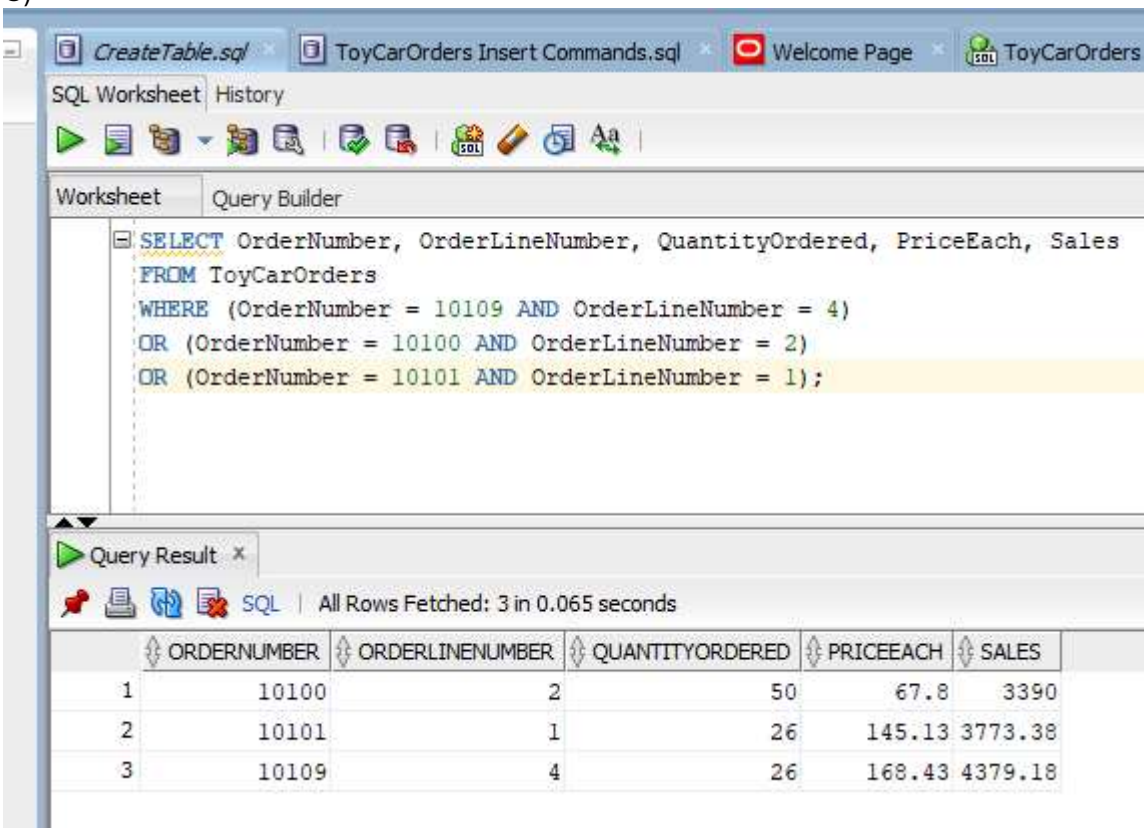
```
SELECT *
FROM TOYCARORDERS
```

Query Result — All Rows Fetched: 16 in 0.125 seconds

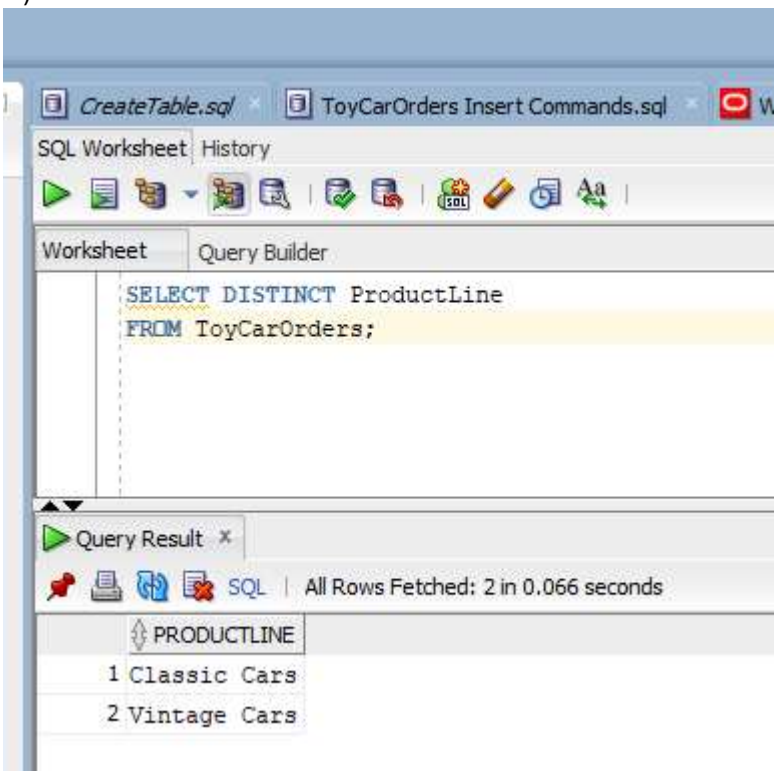
	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	DAYSSINCELASTORDER	PRODUCTLINE	CUSTOMERNAME	ADDRESSLINE1	CITY	POSTALCODE	COUTNRY	CONTACTLASTNAME	CONTACTFIRSTNAME	DEALSIZE
1	10100	22	86.51	4	1903.22	06-JAN-18	2096	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Small
2	10100	30	171.7	3	5151	06-JAN-18	1429	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Medium
3	10100	49	34.47	1	1689.03	06-JAN-18	2836	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Small
4	10100	50	67.8	2	3390	06-JAN-18	1529	Vintage Cars	Online Diecast Creations Co.	2304 Long Airport Avenue	Nashua	62005	USA	Young	Valarie	Medium
5	10101	25	151.28	4	3782	09-JAN-18	1573	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Medium
6	10101	26	145.13	1	3773.38	09-JAN-18	1671	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Medium
7	10101	45	31.2	3	1404	09-JAN-18	2360	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Small
8	10101	46	53.76	2	2472.96	09-JAN-18	2434	Vintage Cars	Blauer See Auto, Co.	Lyonerstr. 34	Frankfurt	60528	Germany	Keitel	Roland	Small
9	10102	39	123.29	2	4808.31	10-JAN-18	1327	Vintage Cars	Vitachrome Inc.	2678 Kingston Rd.	NYC	10022	USA	Frick	Michael	Medium
10	10102	41	50.14	1	2055.74	10-JAN-18	1351	Vintage Cars	Vitachrome Inc.	2678 Kingston Rd.	NYC	10022	USA	Frick	Michael	Small
11	10109	26	121.44	1	3157.44	10-MAR-18	1636	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
12	10109	26	168.43	4	4379.18	10-MAR-18	1241	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
13	10109	29	32.1	6	930.9	10-MAR-18	2547	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Small
14	10109	38	116.65	3	4432.7	10-MAR-18	1412	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium
15	10109	46	179.5	5	8257	10-MAR-18	1767	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Large
16	10109	47	132.8	2	6241.6	10-MAR-18	1933	Classic Cars	Motor Mint Distributors Inc.	11328 Douglas Av.	Philadelphia	71270	USA	Hernandez	Rosa	Medium

With the small amount of data (both in terms of tuples and attributes) present in the table, the best overview of the data is gained simply by using a SELECT * statement and viewing the entirety of the table. If there were more attributes present, limiting the attribute selection to only the essential information would have been beneficial. If there were more tuples present, it would have been beneficial to group the data in some way. A later question deals with summary statistics such as average, max, and min, so a different option would be to use GROUP BY on the OrderNo and consolidate the data for each order.

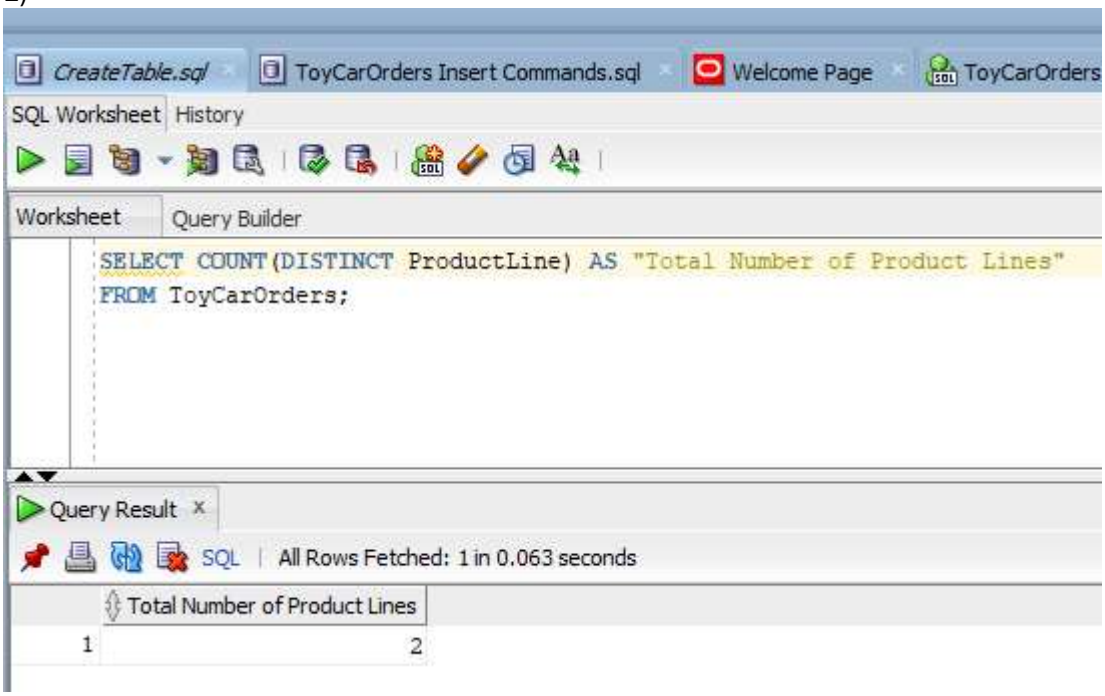
C)


CreateTable.sql
ToyCarOrders Insert Commands.sql
Welcome Page
ToyCarOrders
SQL Worksheet
History
Worksheet
Query Builder
SELECT OrderNumber, OrderLineNumber, QuantityOrdered, PriceEach, Sales
FROM ToyCarOrders
WHERE (OrderNumber = 10109 AND OrderLineNumber = 4)
OR (OrderNumber = 10100 AND OrderLineNumber = 2)
OR (OrderNumber = 10101 AND OrderLineNumber = 1);
Query Result
SQL
All Rows Fetched: 3 in 0.065 seconds
ORDERNUMBER ORDERLINENUMBER QUANTITYORDERED PRICEEACH SALES
1 10100 2 50 67.8 3390
2 10101 1 26 145.13 3773.38
3 10109 4 26 168.43 4379.18

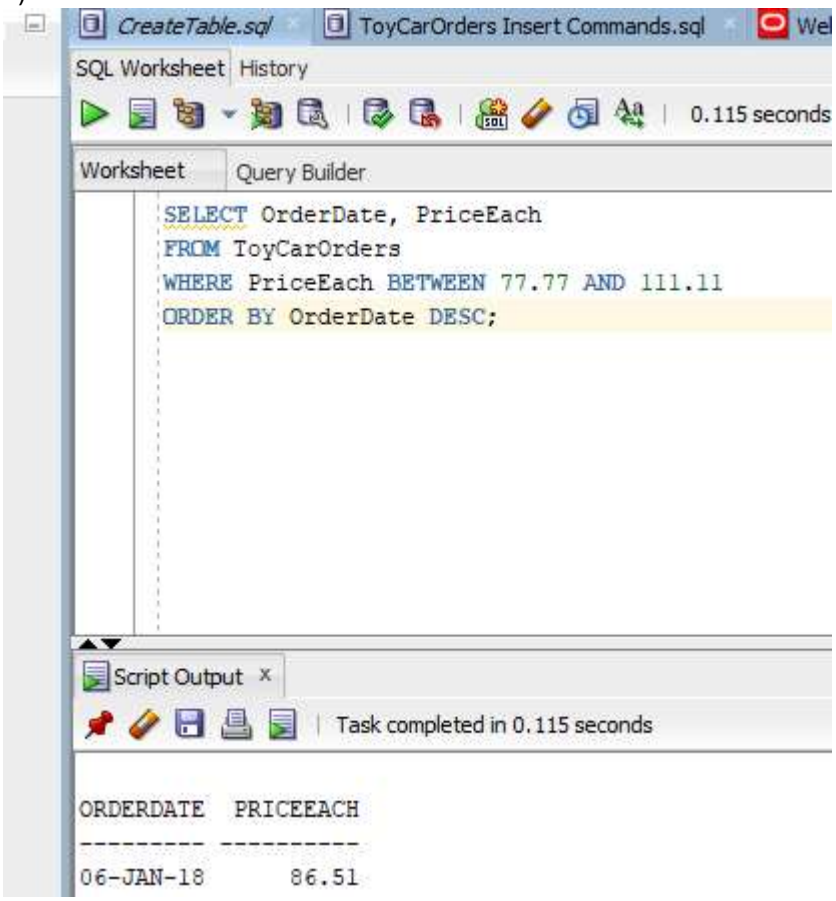
D)


CreateTable.sql
ToyCarOrders Insert Commands.sql
SQL Worksheet
History
Worksheet
Query Builder
SELECT DISTINCT ProductLine
FROM ToyCarOrders;
Query Result
SQL
All Rows Fetched: 2 in 0.066 seconds
PRODUCTLINE
1 Classic Cars
2 Vintage Cars

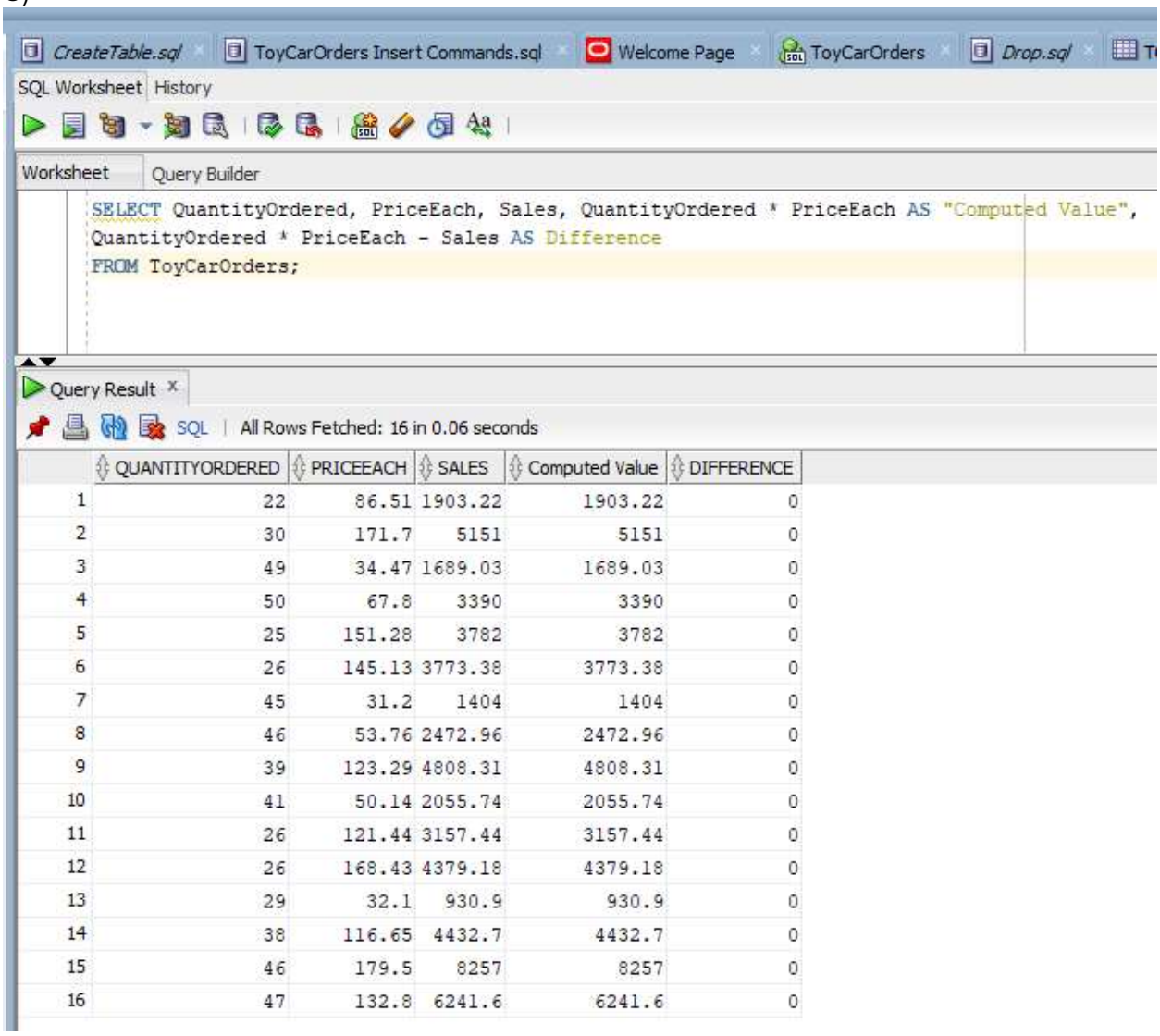
E)


CreateTable.sql
ToyCarOrders Insert Commands.sql
Welcome Page
ToyCarOrders
SQL Worksheet
History
Worksheet
Query Builder
SELECT COUNT(DISTINCT ProductLine) AS "Total Number of Product Lines"
FROM ToyCarOrders;
Query Result
SQL | All Rows Fetched: 1 in 0.063 seconds
Total Number of Product Lines
1
2

F)

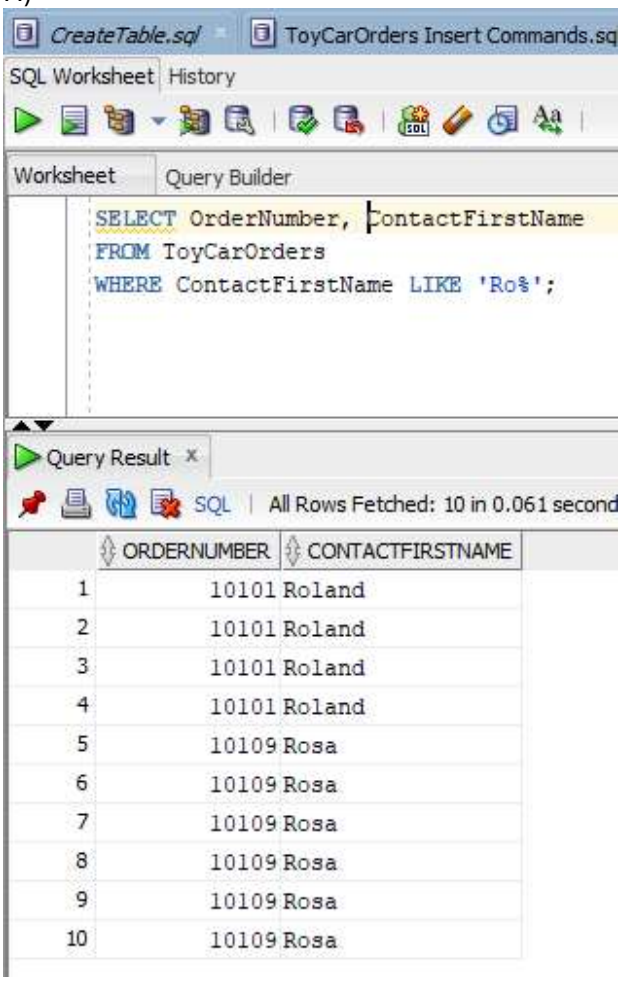

CreateTable.sql
ToyCarOrders Insert Commands.sql
SQL Worksheet
History
0.115 seconds
Worksheet
Query Builder
SELECT OrderDate, PriceEach
FROM ToyCarOrders
WHERE PriceEach BETWEEN 77.77 AND 111.11
ORDER BY OrderDate DESC;
Script Output
Task completed in 0.115 seconds
ORDERDATE PRICEEACH
06-JAN-18 86.51

G)

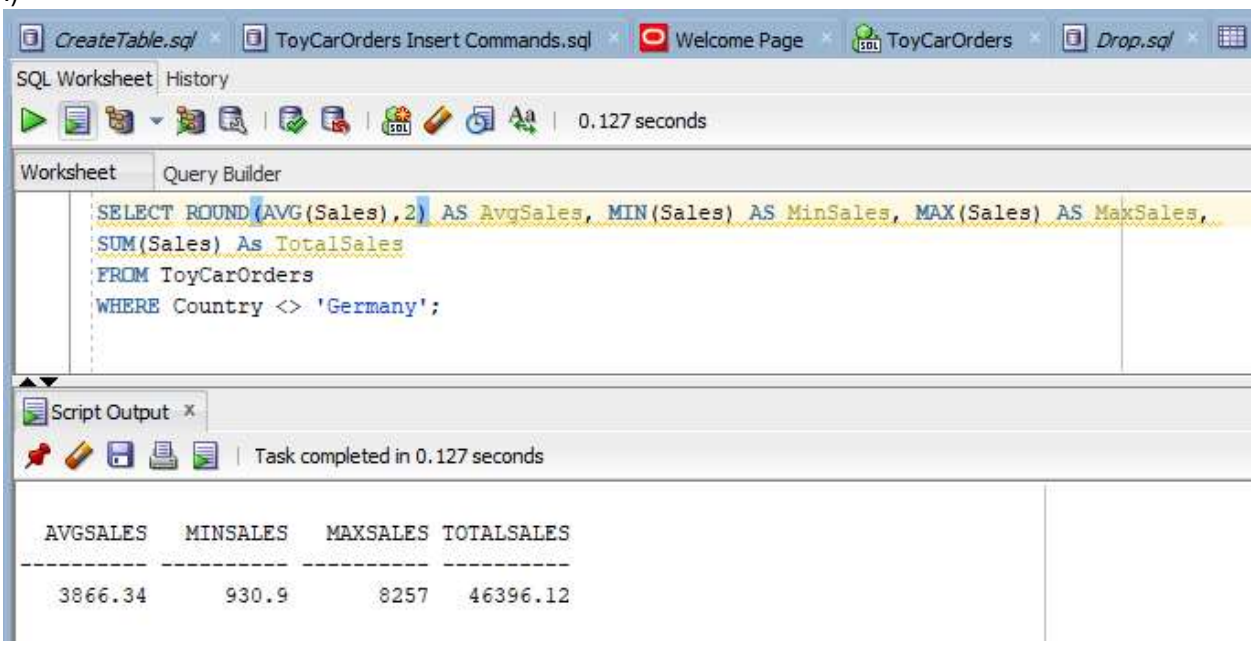


	QUANTITYORDERED	PRICEEACH	SALES	Computed Value	DIFFERENCE
1	22	86.51	1903.22	1903.22	0
2	30	171.7	5151	5151	0
3	49	34.47	1689.03	1689.03	0
4	50	67.8	3390	3390	0
5	25	151.28	3782	3782	0
6	26	145.13	3773.38	3773.38	0
7	45	31.2	1404	1404	0
8	46	53.76	2472.96	2472.96	0
9	39	123.29	4808.31	4808.31	0
10	41	50.14	2055.74	2055.74	0
11	26	121.44	3157.44	3157.44	0
12	26	168.43	4379.18	4379.18	0
13	29	32.1	930.9	930.9	0
14	38	116.65	4432.7	4432.7	0
15	46	179.5	8257	8257	0
16	47	132.8	6241.6	6241.6	0

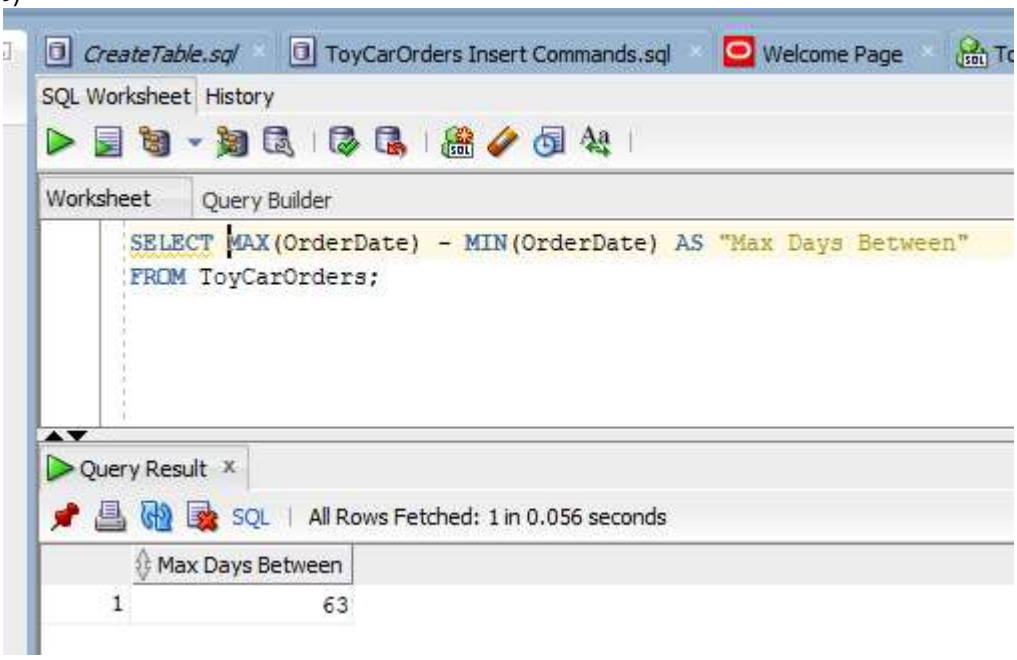
H)


CreateTable.sql
ToyCarOrders Insert Commands.sq
SQL Worksheet
History
Worksheet
Query Builder
SELECT OrderNumber, ContactFirstName
FROM ToyCarOrders
WHERE ContactFirstName LIKE 'Ro%';
Query Result
SQL
All Rows Fetched: 10 in 0.061 second
ORDERNUMBER
CONTACTFIRSTNAME
1 10101 Roland
2 10101 Roland
3 10101 Roland
4 10101 Roland
5 10109 Rosa
6 10109 Rosa
7 10109 Rosa
8 10109 Rosa
9 10109 Rosa
10 10109 Rosa

I)


CreateTable.sql
ToyCarOrders Insert Commands.sql
Welcome Page
ToyCarOrders
Drop.sql
SQL Worksheet
History
0.127 seconds
Worksheet
Query Builder
SELECT ROUND(AVG(Sales),2) AS AvgSales, MIN(Sales) AS MinSales, MAX(Sales) AS MaxSales,
SUM(Sales) As TotalSales
FROM ToyCarOrders
WHERE Country <> 'Germany';
Script Output
Task completed in 0.127 seconds
AVGSALES MINSALES MAXSALES TOTALSALES
3866.34 930.9 8257 46396.12

J)


CreateTable.sql
ToyCarOrders Insert Commands.sql
Welcome Page
SQL Worksheet
History
Worksheet
Query Builder
SELECT MAX(OrderDate) - MIN(OrderDate) AS "Max Days Between"
FROM ToyCarOrders;
Query Result
SQL
All Rows Fetched: 1 in 0.056 seconds
Max Days Between
1
63