



uOttawa

Faculté de génie
Faculty of Engineering

DTI5126[EG] Fundamentals/Applied Data Science

Assignment1_SQL_Queries

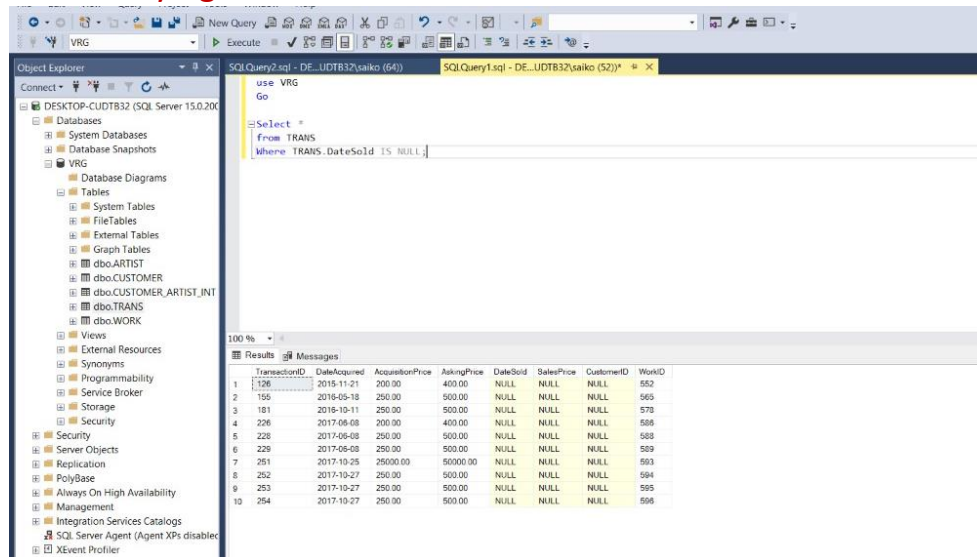
Hadeer Mohamed 300327273

Eman Mohamed 300327278

Part A: RDBMS & SQL working on VRG

a) We Identified transactions with null values on the DateSoldID and remove them from the table

- Identifying Transactions



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-CUDTB32 (SQL Server 15.0.2008.1)'. The central pane shows a query window with the following SQL code:

```
use VRG
Go
Select *
from TRANS
where TRANS.DateSold IS NULL;
```

The Results pane displays a table with 10 rows of data. The columns are TransactionID, DateAcquired, AcquisitionPrice, AskingPrice, DateSold, SalesPrice, CustomerID, and WorkID. The DateSold column contains null values for all 10 transactions.

TransactionID	DateAcquired	AcquisitionPrice	AskingPrice	DateSold	SalesPrice	CustomerID	WorkID
126	2015-11-21	200.00	400.00	NULL	NULL	NULL	552
155	2016-05-18	250.00	500.00	NULL	NULL	NULL	565
181	2016-10-11	250.00	500.00	NULL	NULL	NULL	578
226	2017-06-08	200.00	400.00	NULL	NULL	NULL	588
228	2017-06-08	250.00	500.00	NULL	NULL	NULL	588
229	2017-06-08	250.00	500.00	NULL	NULL	NULL	589
251	2017-10-25	25000.00	50000.00	NULL	NULL	NULL	593
252	2017-10-27	250.00	500.00	NULL	NULL	NULL	594
253	2017-10-27	250.00	500.00	NULL	NULL	NULL	595
254	2017-10-27	250.00	500.00	NULL	NULL	NULL	596

- After removing them



The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-CUDTB32 (SQL Server 15.0.2008.1)'. The central pane shows a query window with the following SQL code:

```
use VRG
Go
Select *
from TRANS
where TRANS.DateSold IS NULL;

DELETE FROM TRANS
where DateSold IS NULL;
```

The Messages pane at the bottom shows the following message:

(10 rows affected)
Completion time: 2022-06-09T22:10:42.5907950+02:00

- b) Listing the WorkId, Title, Medium, ArtistID, and the concatenated artist name renamed as FullName for all artwork that the title contains the word “Yellow, “Blue” or “White”, e.g., the title “On White II” would meet the criteria

```

--=====
--select CONCAT(ARTIST.FirstName ,ARTIST.LastName) As Full_Name ,Work.WorkId,Work.Title,Work.Medium,Work.ArtistID
from ARTIST,WORK
WHERE ARTIST.ArtistID=WORK.ArtistID AND WORK.Title like '%Yellow%' OR WORK.Title like '%Blue%' OR WORK.Title like '%White%';
--=====

```

	Full_Name	WorkId	Title	Medium	ArtistID
1	Joan Miro	523	On White II	High Quality Limited Print	2
2	Wassily Kandinsky	523	On White II	High Quality Limited Print	2
3	Paul Klee	523	On White II	High Quality Limited Print	2
4	Henri Matisse	523	On White II	High Quality Limited Print	2
5	Marc Chagall	523	On White II	High Quality Limited Print	2
6	John Singer Sargent	523	On White II	High Quality Limited Print	2
7	Mark Tobey	523	On White II	High Quality Limited Print	2
8	Paul Horvuchi	523	On White II	High Quality Limited Print	2
9	Morris Graves	523	On White II	High Quality Limited Print	2
10	Joan Miro	571	Yellow Covers Blue	Oil and collage	18
11	Wassily Kandinsky	571	Yellow Covers Blue	Oil and collage	18
12	Paul Klee	571	Yellow Covers Blue	Oil and collage	18
13	Henri Matisse	571	Yellow Covers Blue	Oil and collage	18
14	Marc Chagall	571	Yellow Covers Blue	Oil and collage	18
15	John Singer Sargent	571	Yellow Covers Blue	Oil and collage	18
16	Mark Tobey	571	Yellow Covers Blue	Oil and collage	18
17	Paul Horvuchi	571	Yellow Covers Blue	Oil and collage	18
18	Morris Graves	571	Yellow Covers Blue	Oil and collage	18
19	Joan Miro	590	Blue Interior	Tempera on card	17
20	Wassily Kandinsky	590	Blue Interior	Tempera on card	17
21	Paul Klee	590	Blue Interior	Tempera on card	17
22	Henri Matisse	590	Blue Interior	Tempera on card	17

Query executed successfully. DESKTOP-CUDTB32 (15.0 RTM) DESKTOP

- c) Showing the Year, ArtistID, sum of SalesPrice as SumOfSubTotal, and average of SalesPrice as AverageOfSubtotal for each year.

SQLQuery15.sql - DESKTOP-CUDTB32.VRG (DESKTOP-CUDTB32(saiko) (54)) - Microsoft SQL Server Management Studio

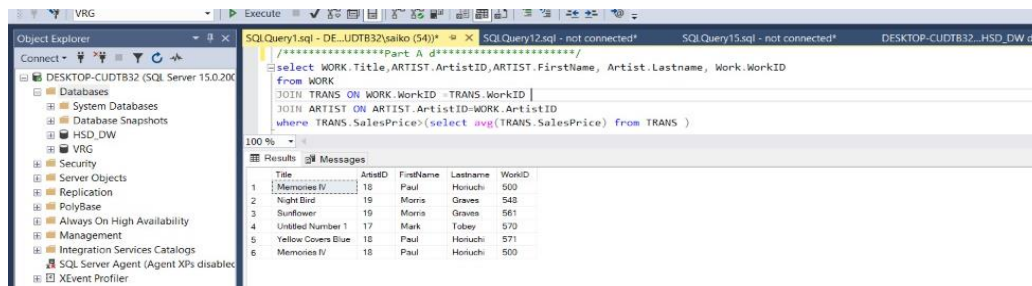
```

--=====
--select Year(TRANS.DateSold)AS YearOfSold, sum(TRANS.SalesPrice) AS SumOfSubTotal,avg(TRANS.SalesPrice) as AverageOfSubtotal
from WORK, TRANS
where WORK.WorkID=TRANS.WorkID
Group by Year(TRANS.DateSold)
--=====
--select WORK.ArtistID,DateOfBirth,sum(TRANS.SalesPrice) AS SumOfSubTotal,avg(TRANS.SalesPrice) as AverageOfSubtotal
from WORK
JOIN TRANS ON WORK.WorkID = TRANS.WorkID
JOIN ARTIST ON ARTIST.ArtistID=WORK.ArtistID
Group by WORK.ArtistID ,ARTIST.DateOfBirth
ORDER BY WORK.ArtistID
--=====

```

ArtistID	DateOfBirth	SumOfSubTotal	AverageOfSubtotal
1	1893	600.00	300.000000
2	1896	400.00	200.000000
3	1899	800.00	400.000000
4	1857	675.00	225.000000
5	1856	575.00	287.500000
6	1890	28625.00	4089.285714
7	1906	170400.00	42600.000000
8	1920	45500.00	15166.666666

d) Show the ArtistID , FirstName, Lastname, WorkID, and Title of Artists that have an artwork sold with a SalesPrice above the average SalesPrice



The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is as follows:

```

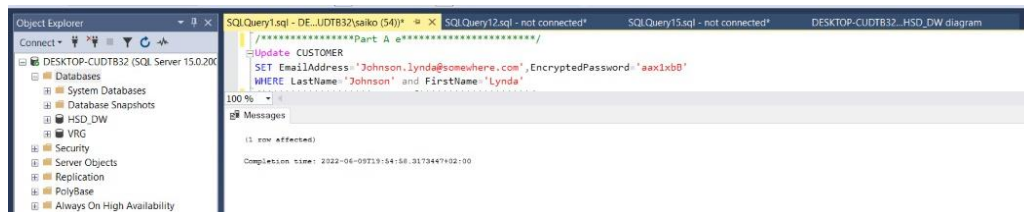
select WORK.Title,ARTIST.ArtistID,Artist.FirstName, Artist.Lastname, Work.WorkID
from WORK
JOIN TRANS ON WORK.WorkID =TRANS.WorkID
JOIN ARTIST ON ARTIST.ArtistID=WORK.ArtistID
where TRANS.SalesPrice>(select avg(TRANS.SalesPrice) from TRANS )

```

The results are displayed in a table with the following columns: Title, ArtistID, FirstName, Lastname, WorkID. The results are as follows:

Title	ArtistID	FirstName	Lastname	WorkID
Memories IV	18	Paul	Horuchi	500
Night Bird	19	Morris	Graves	548
Sunflower	19	Morris	Graves	561
Untitled Number 1	17	Mark	Tobey	570
Yellow Covers Blue	18	Paul	Horuchi	571
Memories IV	18	Paul	Horuchi	500

e) Modify the email of the customer Johnson Lynda and her encrypted password from NULL to Johnson.lynda@somewhere.com and "aax1xbB" respectively.



The screenshot shows an SQL update query in SQL Server Enterprise Manager. The query is as follows:

```

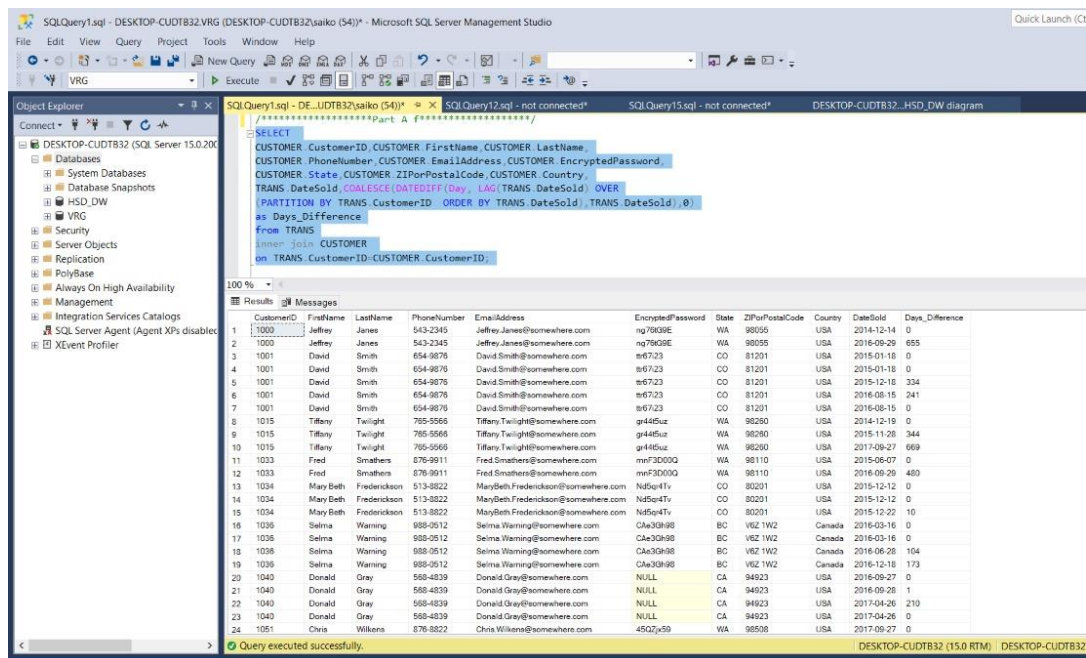
Update CUSTOMER
SET EmailAddress='Johnson.Lynda@somewhere.com',EncryptedPassword='aax1xbB'
WHERE LastName='Johnson' and FirstName='Lynda'

```

The results are displayed in a table with the following columns: Title, ArtistID, FirstName, Lastname, WorkID. The results are as follows:

Title	ArtistID	FirstName	Lastname	WorkID
Memories IV	18	Paul	Horuchi	500
Night Bird	19	Morris	Graves	548
Sunflower	19	Morris	Graves	561
Untitled Number 1	17	Mark	Tobey	570
Yellow Covers Blue	18	Paul	Horuchi	571
Memories IV	18	Paul	Horuchi	500

f) For each customer, find the time (in days) between a purchase and the next for the DateSoldID. Display all the attributes of the customer and days between purchase as Days_Difference. Consider using the Lead or Lag function.



The screenshot shows a SQL query in SQL Server Enterprise Manager. The query is as follows:

```

SELECT
CUSTOMER.CustomerID,CUSTOMER.FirstName,CUSTOMER.LastName,
CUSTOMER.PhoneNumber,CUSTOMER.EmailAddress,CUSTOMER.EncryptedPassword,
CUSTOMER.State,CUSTOMER.ZIPorPostalCode,CUSTOMER.Country,
TRANS.DateSold,COALESCE(DATEDIFF(Day,LAG(TRANS.DateSold)OVER
(PARTITION BY TRANS.CustomerID ORDER BY TRANS.DateSold),TRANS.DateSold),0)
as Days_Difference
FROM TRANS
Join CUSTOMER
on TRANS.CustomerID=CUSTOMER.CustomerID

```

The results are displayed in a table with the following columns: CustomerID, FirstName, LastName, PhoneNumber, EmailAddress, EncryptedPassword, State, ZIPorPostalCode, Country, DateSold, Days_Difference. The results are as follows:

CustomerID	FirstName	LastName	PhoneNumber	EmailAddress	EncryptedPassword	State	ZIPorPostalCode	Country	DateSold	Days_Difference
1000	Jeffrey	Janes	543-2345	Jeffrey.Janes@somewhere.com	ng70G09E	WA	98055	USA	2014-12-14	0
1000	Jeffrey	Janes	543-2345	Jeffrey.Janes@somewhere.com	ng70G09E	WA	98055	USA	2016-09-29	655
1001	David	Smith	654-9876	David.Smith@somewhere.com	te6723	CO	81201	USA	2015-01-18	0
1001	David	Smith	654-9876	David.Smith@somewhere.com	te6723	CO	81201	USA	2015-01-18	0
1001	David	Smith	654-9876	David.Smith@somewhere.com	te6723	CO	81201	USA	2015-12-16	334
1001	David	Smith	654-9876	David.Smith@somewhere.com	te6723	CO	81201	USA	2016-08-15	241
1001	David	Smith	654-9876	David.Smith@somewhere.com	te6723	CO	81201	USA	2016-08-15	0
1015	Tiffany	Twilight	785-5566	Tiffany.Twilight@somewhere.com	g44d5u	WA	98280	USA	2014-12-19	0
1015	Tiffany	Twilight	785-5566	Tiffany.Twilight@somewhere.com	g44d5u	WA	98280	USA	2015-11-28	344
1015	Tiffany	Twilight	785-5566	Tiffany.Twilight@somewhere.com	g44d5u	WA	98280	USA	2017-09-27	669
1033	Fred	Smithers	876-9911	Fred.Smithers@somewhere.com	mrf3D009G	WA	98110	USA	2015-06-07	0
1033	Fred	Smithers	876-9911	Fred.Smithers@somewhere.com	mrf3D009G	WA	98110	USA	2016-09-29	480
1034	Mary Beth	Frederickson	513-8822	MaryBeth.Frederickson@somewhere.com	Nd5q4TfV	CO	80201	USA	2015-12-12	0
1034	Mary Beth	Frederickson	513-8822	MaryBeth.Frederickson@somewhere.com	Nd5q4TfV	CO	80201	USA	2015-12-12	0
1034	Mary Beth	Frederickson	513-8822	MaryBeth.Frederickson@somewhere.com	Nd5q4TfV	CO	80201	USA	2015-12-22	10
1036	Selma	Warring	988-0512	Selma.Warring@somewhere.com	Cae3Qh98	BC	V6Z 1W2	Canada	2016-03-16	0
1036	Selma	Warring	988-0512	Selma.Warring@somewhere.com	Cae3Qh98	BC	V6Z 1W2	Canada	2016-03-16	0
1036	Selma	Warring	988-0512	Selma.Warring@somewhere.com	Cae3Qh98	BC	V6Z 1W2	Canada	2016-09-28	104
1036	Selma	Warring	988-0512	Selma.Warring@somewhere.com	Cae3Qh98	BC	V6Z 1W2	Canada	2016-12-18	173
1040	Donald	Gray	588-4839	Donald.Gray@somewhere.com	NULL	CA	94923	USA	2016-09-27	0
1040	Donald	Gray	588-4839	Donald.Gray@somewhere.com	NULL	CA	94923	USA	2016-09-28	1
1040	Donald	Gray	588-4839	Donald.Gray@somewhere.com	NULL	CA	94923	USA	2017-04-26	210
1040	Donald	Gray	588-4839	Donald.Gray@somewhere.com	NULL	CA	94923	USA	2017-04-26	0
1051	Chris	Wilkins	678-8822	Chris.Wilkins@somewhere.com	45Q7y59	WA	98058	USA	2017-09-27	0

- g) Create a view called CustomerTransactionSummaryView to display the concatenated customer name renamed as FullName using the LastName and FirstName, Title, DateAcquired, DateSold, and difference in the AcquisitionPrice and SalesPrice as Profit for art works with an AskingPrice greater than \$20,000. Use the JOIN ON syntax and order by the AskingPrice in descending order (Ensure to add space between the full name if required)

```

/*****Part A g*****/
CREATE VIEW CustomerTransactionSummary
AS
Select Title,DateAcquired,DateSold,(SalesPrice-AcquisitionPrice) as Profit,CONCAT(CUSTOMER.FirstName,CUSTOMER.LastName) as FullName
from TRANS
JOIN WORK ON WORK.WorkID =TRANS.WorkID
JOIN CUSTOMER ON CUSTOMER.CustomerID=TRANS.CustomerID
where WORK.WorkID=TRANS.WorkID AND TRANS.AskingPrice>20000
Order by TRANS.AskingPrice desc OFFSET 0 rows

select * from CustomerTransactionSummary

```

	Title	DateAcquired	DateSold	Profit	FullName	
1	Memories IV	2016-09-29	2016-12-18	32500.00	Selma	Warning
2	Yellow Covers Blue	2016-08-23	2016-09-29	20000.00	Jeffrey	Janes
3	Memories IV	2014-11-04	2014-12-14	12500.00	Jeffrey	Janes
4	Night Bird	2015-09-21	2015-11-28	12500.00	Tiffany	Twilight

- h) Build a single temporary table called Purchase that captures customers' purchases from 2015 to 2017. The table should contain the TransactionID, DateAcquired, CustomerID, LastName, FirstName, first AcquisitionDate as MinAcquisitionDate, last AcquisitionDate as MaxAcquisitionDate, and Medium used for the artwork. Also, the Medium values should be represented as numeric values using High Quality Limited Print – 1, Color Aquatint – 2, Water Color and Ink – 3, Oil and Collage – 4, Others - 5.

Note: consider using CTEs and CASE statement in your query if required.

Queries:

```

WITH Purchase_new (CustomerID,MinAcquisitionDate,MaxAcquisitionDate)
AS
(
    SELECT
        CustomerID,
        MIN(TRANS.DateAcquired) AS MinAcquisitionDate,
        MAX(TRANS.DateAcquired) AS MaxAcquisitionDate
    FROM TRANS
    GROUP BY TRANS.CustomerID
)

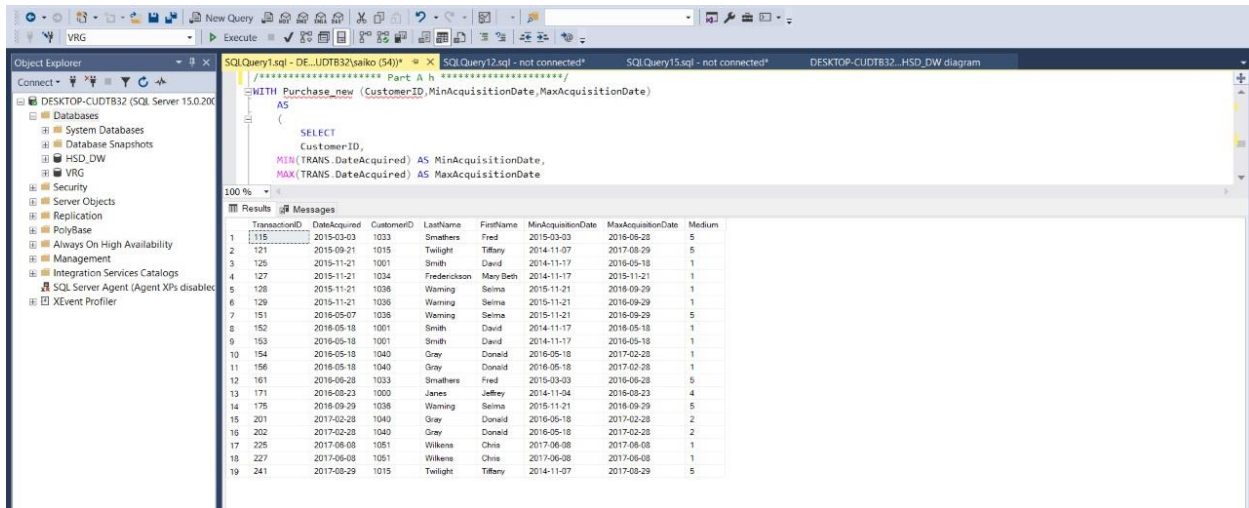
select
    TransactionID,DateAcquired,Purchase_new.CustomerID,Customer.LastName,
    Customer.FirstName,MinAcquisitionDate,MaxAcquisitionDate,
    CASE
        WHEN Medium='High Quality Limited Print' THEN 1
        WHEN Medium='Color Aquatint' THEN 2
        WHEN Medium='Water Color and Ink' THEN 3
        WHEN Medium='Oil and Collage' THEN 4
        ELSE 5 END AS Medium
    into #Purchase
FROM Purchase_new

INNER JOIN CUSTOMER ON
Purchase_new.CustomerID = CUSTOMER.CustomerID
INNER JOIN TRANS ON
Purchase_new.CustomerID = TRANS.CustomerID
INNER JOIN WORK ON
TRANS.WorkID = WORK.WorkID

where Year(DateAcquired)>=2015 AND Year(DateAcquired)<=2017
select * from #Purchase

```

Output:



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'DESKTOP-CUDTB32 (SQL Server 15.0.200)' instance expanded, showing 'Databases' and 'HSD_DW'. The right pane shows a query window with the following SQL code:

```
/****** Part A h *****/
WITH Purchase_new (CustomerID, MinAcquisitionDate, MaxAcquisitionDate)
AS
(
    SELECT
        CustomerID,
        MIN(TRANS.DateAcquired) AS MinAcquisitionDate,
        MAX(TRANS.DateAcquired) AS MaxAcquisitionDate

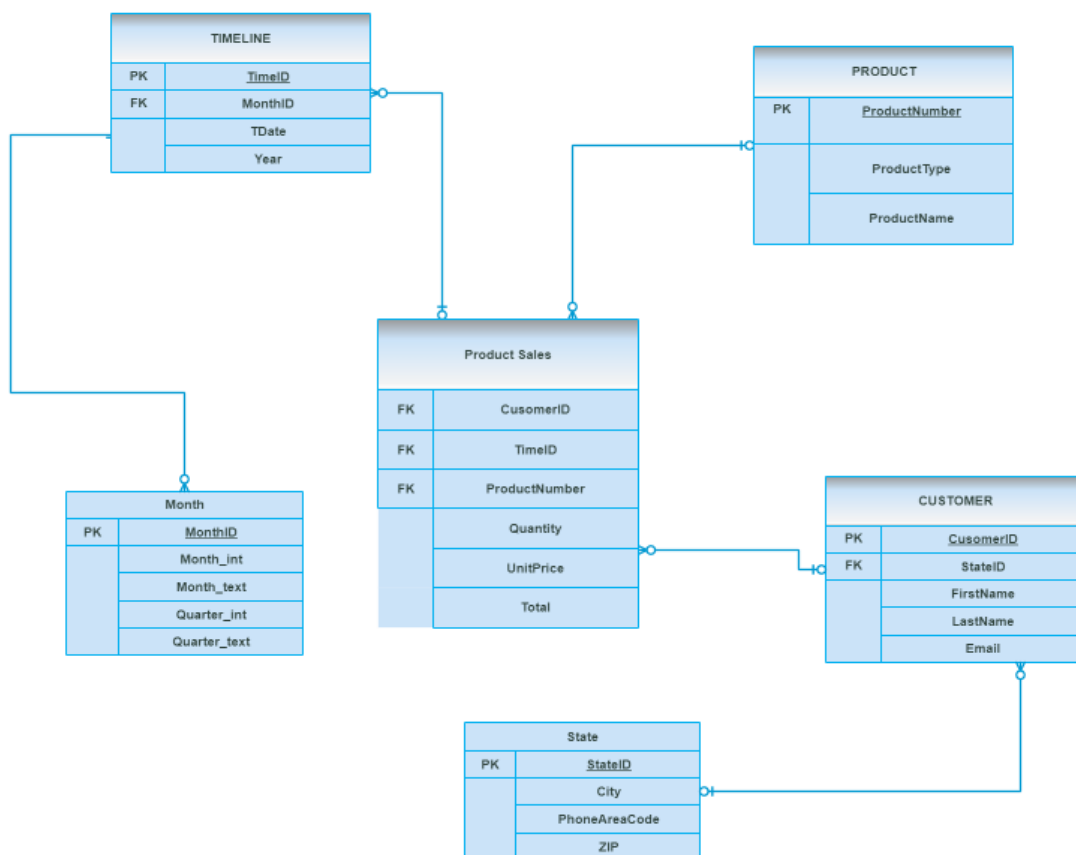
```

Below the query window, the 'Results' pane displays the output of the query. The results are shown in a table with the following columns: TransactionID, DateAcquired, CustomerID, LastName, FirstName, MinAcquisitionDate, MaxAcquisitionDate, and Medium. The results are sorted by TransactionID in ascending order.

TransactionID	DateAcquired	CustomerID	LastName	FirstName	MinAcquisitionDate	MaxAcquisitionDate	Medium
1	2015-03-03	1033	Smathers	Fred	2015-03-03	2016-06-28	5
2	2015-09-21	1015	Twilight	Tiffany	2014-11-07	2017-08-29	5
3	2015-11-21	1001	Smith	David	2014-11-17	2016-05-18	1
4	2015-11-21	1034	Frederickson	Mary Beth	2014-11-17	2015-11-21	1
5	2015-11-21	1036	Warning	Seima	2015-11-21	2016-09-29	1
6	2015-11-21	1036	Warning	Seima	2015-11-21	2016-09-29	1
7	2016-05-07	1036	Warning	Seima	2015-11-21	2016-09-29	5
8	2016-05-18	1001	Smith	David	2014-11-17	2016-05-18	1
9	2016-05-18	1001	Smith	David	2014-11-17	2016-05-18	1
10	2016-05-18	1040	Gray	Donald	2016-05-18	2017-02-28	1
11	2016-05-18	1040	Gray	Donald	2016-05-18	2017-02-28	1
12	2016-06-28	1033	Smathers	Fred	2015-03-03	2016-06-28	5
13	2016-08-23	1000	Janes	Jeffrey	2014-11-04	2016-08-23	4
14	2016-09-29	1036	Warning	Seima	2015-11-21	2016-09-29	5
15	2017-02-28	1040	Gray	Donald	2016-05-18	2017-02-28	2
16	2017-02-28	1040	Gray	Donald	2016-05-18	2017-02-28	2
17	2017-06-08	1051	Wilkens	Chris	2017-06-08	2017-06-08	1
18	2017-06-08	1051	Wilkens	Chris	2017-06-08	2017-06-08	1
19	2017-08-29	1015	Twilight	Tiffany	2014-11-07	2017-08-29	5

Part B: Data Warehousing & OLAP

- 1) Sketch a representative snowflake schema for the data warehouse (specifying the relations, the attributes, the primary keys, and the foreign keys).



2- a) Customer(s) made an order containing at least five products with different product numbers” Providing the CustomerName and CustomerID

```

/*****Part B 2a*****/
select distinct PRODUCT_SALES.CustomerID ,CUSTOMER.FullName from PRODUCT_SALES
join CUSTOMER on CUSTOMER.CustomerID=PRODUCT_SALES.CustomerID
join Product on PRODUCT_SALES.ProductNumber=Product.ProductNumber
group by TimeID ,PRODUCT_SALES.CustomerID,CUSTOMER.FullName
having COUNT(PRODUCT_SALES.ProductNumber)>=5

```

100 %

Results Messages

	CustomerID	FullName
1	3	Able, Ralph
2	6	Foxtrot, Kathy
3	12	Wayne, Joan

b. customer(s) made the largest order, those that would result in the largest bill

```

with largest as(
select sum(PRODUCT_SALES.Total) as summition,CUSTOMER.CustomerID from PRODUCT_SALES
join CUSTOMER on CUSTOMER.CustomerID=PRODUCT_SALES.CustomerID
group by CUSTOMER.CustomerID
)
select c.* ,largest.CustomerID from largest ,CUSTOMER as c
where largest.CustomerID=c.CustomerID AND largest.summition IN(select max(largest.summition)from largest)

```

%

Results Messages

CustomerID	FullName	Email	PhoneAreaCode	City	State	ZIP	CustomerID
11	Tyler, Jenny	somewhere.com	972	Dallas	TX	75225	11

c. Write SQL queries for the” Roll-Up” operation to summaries the total sales per Year.

```

/***** part B 2c*****/
Select y.Year ,sum(Total) from PRODUCT_SALES as S,TIMELINE as Y
where S.TimeID=Y.TimeID
group by Rollup(y.Year)

```

100 %

Results Messages

	Year	(No column name)
1	2017	95.78
2	2018	845.11
3	NULL	940.89

3 A.The aggregates that the analyst would start with:

In problem 3 Part B, we want to find the reason for the decreased total from April 2018 to June 2018 according to either product type or customer city

So according to the city attribute, we sum the total product sales of each city in the year 2018, Quarter 2

The result was not enough to get insightful meaning from it and you can't understand the reason behind the decrease in the total sales price

We also analyze the decreased rate of total sales price according to product type we sum up the total sales price for each product type in quarter 2 of the year 2018

However, the result wasn't useful and didn't give a full understanding of our case

So, we use the drill-down operation to go deep into our data to give more information and more useful insights

We apply drill down on both city and product type analysis:

Drill down operation on city attribute firstly join three tables which are customer table, product sales, and timeline then we each city, Quarter, year and month apply the sum of the product sales in quarter 2 and year 2018.

The results show clear and insightful meaning for the decreased total sales price for each city

Drill down operation on product type attribute firstly join three tables which are product table, product sales, and timeline then for each product type, Quarter, year and month apply the sum of the product sales in quarter 2 and year 2018.

The results show clear and insightful meaning for the decreased total sales price for each product type

B.The Relevant “drill-down” operations that the analyst would need to execute

```
-- Quarter --
SELECT CUSTOMER.City,TIMELINE.Quarter_int, TIMELINE.year, SUM(PRODUCT_SALES.Total) as Total FROM
CUSTOMER INNER JOIN PRODUCT_SALES ON CUSTOMER.CustomerID = PRODUCT_SALES.CustomerID
INNER JOIN TIMELINE ON PRODUCT_SALES.TimeID = TIMELINE.TimeID
GROUP BY City, TIMELINE.Quarter_int,TIMELINE.year
HAVING TIMELINE.Quarter_int = 2 and TIMELINE.year = 2018

SELECT PRODUCT.ProductType,TIMELINE.Quarter_int, TIMELINE.year, SUM(PRODUCT_SALES.Total) as Total FROM
PRODUCT INNER JOIN PRODUCT_SALES ON PRODUCT.ProductNumber = PRODUCT_SALES.ProductNumber
INNER JOIN TIMELINE ON PRODUCT_SALES.TimeID = TIMELINE.TimeID
GROUP BY PRODUCT.ProductType, TIMELINE.Quarter_int,TIMELINE.year
HAVING TIMELINE.Quarter_int = 2 and TIMELINE.year = 2018

-- Drill Down

SELECT CUSTOMER.City,TIMELINE.Month_int, TIMELINE.year, SUM(PRODUCT_SALES.Total) as Total FROM
CUSTOMER INNER JOIN PRODUCT_SALES ON CUSTOMER.CustomerID = PRODUCT_SALES.CustomerID
INNER JOIN TIMELINE ON PRODUCT_SALES.TimeID = TIMELINE.TimeID
GROUP BY City, TIMELINE.Quarter_int,TIMELINE.year,TIMELINE.Month_int
HAVING TIMELINE.Quarter_int = 2 and TIMELINE.year = 2018

SELECT PRODUCT.ProductType,TIMELINE.Month_int, TIMELINE.year, SUM(PRODUCT_SALES.Total) as Total FROM
PRODUCT INNER JOIN PRODUCT_SALES ON PRODUCT.ProductNumber = PRODUCT_SALES.ProductNumber
INNER JOIN TIMELINE ON PRODUCT_SALES.TimeID = TIMELINE.TimeID
GROUP BY PRODUCT.ProductType, TIMELINE.Quarter_int,TIMELINE.year,TIMELINE.Month_int
HAVING TIMELINE.Quarter_int = 2 and TIMELINE.year = 2018
```

Query Results:

100 %				
Results Messages				
	City	Quarter_int	year	Total
1	Austin	2	2018	22.94
2	Dallas	2	2018	210.55
3	Fort Worth	2	2018	142.68
4	San Antonio	2	2018	166.61
	ProductType	Quarter_int	year	Total
1	Book	2	2018	124.75
2	Video	2	2018	314.15
3	Video Companion	2	2018	103.88
	City	Month_int	year	Total
1	Dallas	4	2018	109.78
2	Fort Worth	4	2018	47.89
3	San Antonio	4	2018	72.84
4	Austin	5	2018	22.94
5	Dallas	5	2018	54.89
6	Dallas	6	2018	45.88
7	Fort Worth	6	2018	94.79
8	San Antonio	6	2018	93.77
	ProductType	Month_int	year	Total
1	Book	4	2018	74.85
2	Video	4	2018	119.7
3	Video Companion	4	2018	35.96
4	Video	5	2018	59.85
5	Video Companion	5	2018	17.98
6	Book	6	2018	49.9
7	Video	6	2018	134.6
8	Video Companion	6	2018	49.94

Code:

10

11

CustomerID

43023	NA	NA	NA	NA	NA	NA	NA	NA	NA
43033	NA	NA	NA	NA	NA	NA	NA	NA	NA
43089	NA	NA	NA	NA	NA	NA	NA	NA	NA
43184	NA	NA	NA	NA	NA	NA	NA	NA	NA
43186	NA	NA	NA	NA	NA	NA	NA	NA	NA
43190	NA	NA	NA	NA	NA	NA	NA	NA	NA
43193	NA	NA	NA	NA	NA	NA	NA	NA	NA
43198	NA	NA	NA	NA	NA	NA	NA	NA	NA
43213	NA	NA	NA	NA	NA	NA	NA	NA	NA
43227	NA	NA	NA	NA	NA	NA	1	NA	NA
43241	NA	NA	NA	NA	NA	NA	NA	NA	NA
43256	NA	2	NA	NA	NA	NA	NA	2	NA

, , ProductNumber = BK002

[illegible]

```
ProductNumber = VB001
```

[illegible]

```

, , ProductNumber = VB002

```

TimeID	CustomerID											
	1	3	4	5	6	7	8	9	11	12		
43023	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA		
43033	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA		
43089	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA		
43184	NA	NA	1	NA	NA	NA	NA	NA	NA	NA		
43186	NA	NA	NA	NA	1	NA	NA	NA	NA	NA		

```

, , ProductNumber = vk002
      CustomerID
TimeID  1  3  4  5  6  7  8  9 11 12
43023 NA NA NA NA NA NA NA NA NA NA
43033 NA NA NA NA NA NA NA NA NA NA
43089 NA NA NA NA NA NA NA NA NA NA
43184 NA NA 1 NA NA NA NA NA NA NA
43186 NA NA NA NA 1 NA NA NA NA NA
43190 NA NA NA NA NA NA NA NA NA NA
43193 NA NA NA NA NA NA NA NA NA NA
43198 NA NA NA NA NA NA NA NA NA NA
43213 NA NA NA NA NA NA NA NA NA NA
43227 NA NA NA NA NA NA NA 1 NA NA
43241 NA NA NA NA NA NA NA NA NA NA
43256 NA 2 NA NA NA NA NA 2 1 NA

, , ProductNumber = vk003
      CustomerID
TimeID  1  3  4  5  6  7  8  9 11 12
43023 NA NA NA NA NA NA NA NA NA NA
43033 NA NA NA NA NA NA NA NA NA NA
43089 NA NA NA NA NA NA NA NA NA NA
43184 NA NA NA NA NA NA NA NA NA NA
43186 NA NA NA NA NA 1 NA NA NA NA
43190 NA NA NA NA NA NA NA NA NA NA
43193 NA NA NA NA NA NA NA NA NA NA
43198 NA NA NA NA NA NA NA NA NA NA
43213 NA NA NA NA NA NA NA NA NA NA
43227 NA NA NA NA NA NA NA NA NA NA
43241 NA NA NA NA NA NA NA NA NA NA
43256 NA NA NA NA NA NA NA NA NA NA

, , ProductNumber = vk003
      CustomerID
TimeID  1  3  4  5  6  7  8  9 11 12
43023 NA NA NA NA NA NA NA NA NA NA
43033 NA NA NA NA NA NA NA NA NA NA
43089 NA NA NA NA NA NA NA NA NA NA
43184 NA NA NA NA NA NA NA NA NA NA

[ reached getoption("max.print") -- omitted 8 row(s) and 1 matrix slice(s) ]
>
> dimnames(Quantity_cube)
$TimeID
[1] "43023" "43033" "43089" "43184" "43186" "43190" "43193" "43198" "43213" "43227" "43241" "43256"

$CustomerID
[1] "1" "3" "4" "5" "6" "7" "8" "9" "11" "12"

$ProductNumber
[1] "BK001" "BK002" "VB001" "VB002" "VB003" "VK001" "VK002" "VK003" "VK003" "VK004"

```