Fundamentals of Applied Data Science
Assignment 1

Abdelrhman Gaber Youssef Saad Rezkallah 300327290
Basma Reda Shaban Abd-Elsalam Abd-Elwahab 300327209

Part 1: RDBMS & SQL

a- Identify transactions with null values on the DateSoldID and remove them from the table:

Query

```
-- Question 1
--Identify the transaction with null value
SELECT * FROM TRANS as t
WHERE t.DateSold is null
-- return all record from the table without null value
SELECT * FROM TRANS as t
WHERE t.DateSold is not null
```

Result of Query

	TransactionID	DateAcquired	AcquisitionPrice	AskingPrice	DateSold	SalesPrice	CustomerID	WorkID
1	126	2015-11-21	200.00	400.00	NULL	NULL	NULL	552
2	155	2016-05-18	250.00	500.00	NULL	NULL	NULL	565
3	181	2016-10-11	250.00	500.00	NULL	NULL	NULL	578
4	226	2017-06-08	200.00	400.00	NULL	NULL	NULL	586

	TransactionID	DateAcquired	AcquisitionPrice	AskingPrice	DateSold	SalesPrice	CustomerID	WorkID
1	100	2014-11-04	30000.00	45000.00	2014-12-14	42500.00	1000	500
2	101	2014-11-07	250.00	500.00	2014-12-19	500.00	1015	511
3	102	2014-11-17	125.00	250.00	2015-01-18	200.00	1001	521
4	103	2014-11-17	250.00	500.00	2015-12-12	400.00	1034	522
5	104	2014-11-17	250.00	250.00	2015-01-18	200.00	1001	523
6	105	2014-11-17	200.00	500.00	2015-12-12	400.00	1034	524
7	115	2015-03-03	1500.00	3000.00	2015-06-07	2750.00	1033	537
8	121	2015-09-21	15000.00	30000.00	2015-11-28	27500.00	1015	548
9	125	2015-11-21	125.00	250.00	2015-12-18	200.00	1001	551

b- List 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:

Query

```
USE VRG

SELECT WorkID ,Title,Medium ,w.ArtistID,CONCAT (a.FirstName,' ' ,a.LastName) as FullName from WORK as w
INNER JOIN ARTIST as a on a.ArtistID = w.ArtistID
WHERE
    (w.Title like '%Yellow%' or w.Title like '%Blue%' or w.Title like '%White%')
```

Result of Query

)0 %

Results | Messages

	WorKID	Title	Medium	ArtistID	FullName	
1	523	On White II	High Quality Limited Print	2	Wassily	Kandinsky
2	571	Yellow Covers Blue	Oil and collage	18	Paul	Horiuchi
3	590	Blue Interior	Tempera on card	17	Mark	Tobey

c- For each Artist, show the Year, ArtistID, sum of SalesPrice as SumOfSubTotal, and average of SalesPrice as AverageOfSubtotal for each year:

Query

```
--Question 3
Select AVG(t.SalesPrice)as AverageOfSubTotal , SUM(t.SalesPrice) as SumOfSubTotal, DATEPART(YEAR , t.DateSold) as year_, c.ArtistID
from TRANS as t
Inner join CUSTOMER_ARTIST_INT as c on c.CustomerID = t.CustomerID
group by ArtistID ,  DATEPART(YEAR , t.DateSold)
```

Result of Query

100 %

Results | Messages

	AverageOfSubTotal	SumOfSubTotal	year_	ArtistID
1	500.000000	500.00	2014	11
2	21500.000000	43000.00	2014	17
3	21500.000000	43000.00	2014	18
4	21500.000000	43000.00	2014	19
5	300.000000	1800.00	2015	1
6	300.000000	1800.00	2015	2
7	300.000000	1800.00	2015	4
8	300.000000	1800.00	2015	5
9	7025.000000	28100.00	2015	11
10	15125.000000	30250.00	2015	17
11	15125.000000	30250.00	2015	18

Query executed successfully.

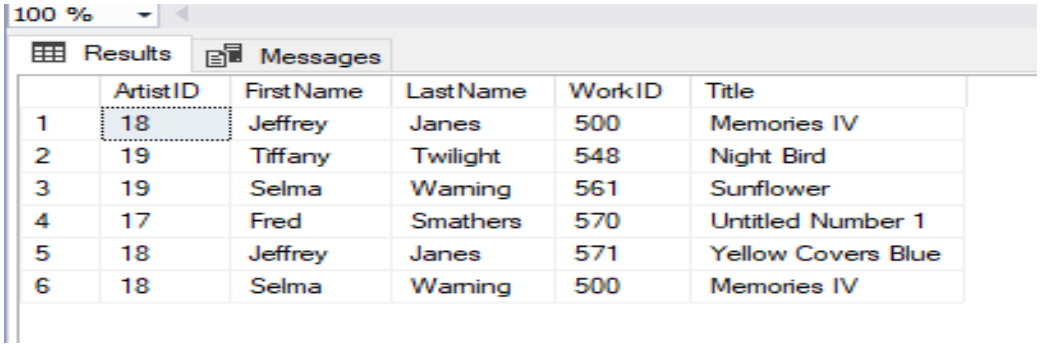
d- Show the ArtistID , FirstName, Lastname, WorkID, and Title of Artists that have an artwork sold with a SalesPrice above the average SalesPrice:

Query

```
-- Question 4
SELECT w.ArtistID , a.FirstName, a.LastName ,t.WorkID, w.Title from WORK as w
inner join TRANS as t on  t.WorkID = w.WorkID
inner join CUSTOMER as a on a.CustomerID = t.CustomerID

WHERE t.SalesPrice > (SELECT AVG(t.SalesPrice) FROM TRANS as t)
```

Result of Query



100 %

Results | Messages

	ArtistID	FirstName	LastName	WorkID	Title
1	18	Jeffrey	Janes	500	Memories IV
2	19	Tiffany	Twilight	548	Night Bird
3	19	Selma	Waming	561	Sunflower
4	17	Fred	Smathers	570	Untitled Number 1
5	18	Jeffrey	Janes	571	Yellow Covers Blue
6	18	Selma	Waming	500	Memories IV

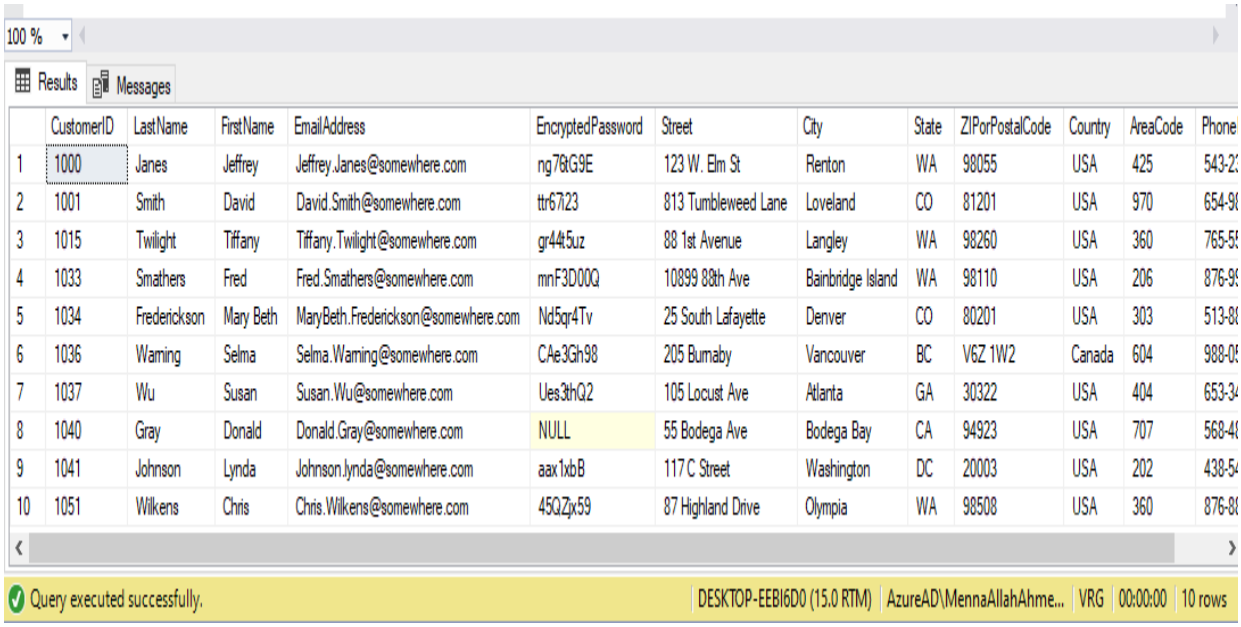
e- Modify the email of the customer Johnson Lynda and her encrypted password from NULL to Johnson.lynda@somewhere.com and “aax1xbB” respectively:

Query

```
--Question 5
UPDATE CUSTOMER
set EmailAddress = 'Johnson.lynda@somewhere.com' ,EncryptedPassword = 'aax1xbB'
WHERE LastName='Johnson' and FirstName='Lynda'

SELECT * FROM CUSTOMER
```

Result of Query



100 %

Results | Messages

	CustomerID	LastName	FirstName	EmailAddress	EncryptedPassword	Street	City	State	ZIPorPostalCode	Country	AreaCode	Phone
1	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng7GtG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2
2	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9
3	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5
4	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9
5	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8
6	1036	Waming	Selma	Selma.Waming@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0
7	1037	Wu	Susan	Susan.Wu@somewhere.com	Ues3thQ2	105 Locust Ave	Atlanta	GA	30322	USA	404	653-3
8	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4
9	1041	Johnson	Lynda	Johnson.lynda@somewhere.com	aax1xbB	117 C Street	Washington	DC	20003	USA	202	438-5
10	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8

Query executed successfully. | DESKTOP-EEBI6D0 (15.0 RTM) | AzureAD\MennaAllahAhme... | VRG | 00:00:00 | 10 rows

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:

Query

```
--Question 6
select *,DATEDIFF(day,m.DateSold,m.nextpurchase) as differanceday from(

Select c.*,t.DateSold ,lead(t.DateSold,1,Null) over (PARTITION BY c.CustomerID ORDER BY t.DateSold) as nextpurchase
from CUSTOMER AS c
inner join TRANS as t on c.CustomerID = t.CustomerID) m

where m.nextpurchase is not Null
```

Result of Query

100 %

Results | Messages

	EncryptedPassword	Street	City	State	ZIPorPostalCode	Country	AreaCode	PhoneNumber	DateSold	nextpurchase	differanceday
1	somewhere.com	ng7&tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345	2014-12-14	2016-09-29	655
2	omewhere.com	ttr67?23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	2015-01-18	2015-01-18	0
3	omewhere.com	ttr67?23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	2015-01-18	2015-12-18	334
4	omewhere.com	ttr67?23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	2015-12-18	2016-08-15	241
5	omewhere.com	ttr67?23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	2016-08-15	2016-08-15	0
6	@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	2014-12-19	2015-11-28	344
7	@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	2015-11-28	2017-09-27	669
8	@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911	2015-06-07	2016-09-29	480
9	erickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	2015-12-12	2015-12-12	0
10	erickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	2015-12-12	2015-12-22	10

Query executed successfully. | DESKTOP-EEBI6D0 (15.0 RTM) | AzureAD\MennaAllahAhme... | VRG | 00:00:00 | 17 rows

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):

Query

```
CREATE VIEW CustomerTransactionSummary AS

SELECT top 100 CONCAT(c.FirstName ,' ' ,c.LastName) as FullName, w.Title ,t.DateAcquired , t.DateSold ,(t.SalesPrice - t.AcquisitionPrice) as profit
from CUSTOMER as c

inner join TRANS as t on t.CustomerID = c.CustomerID
inner join WORK as w on w.WorkID = t.WorkID
where t.AskingPrice > 20000

order by AskingPrice Desc

Select * from CustomerTransactionSummary
```

Result of Query

100 %

Results | Messages

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

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:

Query

```
--Question 8
with purchase(CustomerID,mindate,maxdate)as(
select t.CustomerID , min(t.DateAcquired) as mindate ,max(t.DateAcquired) as maxdate from TRANS as t
group by t.CustomerID

)

SELECT t.TransactionID ,t.DateAcquired , t.CustomerID , c.FirstName ,c.LastName ,p.maxdate,p.mindate,w.Medium,
CASE
    WHEN w.Medium ='High Quality Limited Print' THEN 1
    WHEN w.Medium ='Color aquatint' THEN 2
    WHEN w.Medium = 'WaterColor and Ink' THEN 3
    WHEN w.Medium = 'Oil and Collage' THEN 4
    ELSE 5
END as Medium_encode
--INTO #Purchase
From TRANS as t

inner join CUSTOMER as c on c.CustomerID = t.CustomerID
inner join WORK as w on t.WorkID = w.WorkID
inner join purchase p on p.CustomerID = c.CustomerID

Where t.DateAcquired between '2015-01-01' AND '2017-12-31'
```

Result of Query

	TransactionID	DateAcquired	CustomerID	FirstName	LastName	maxdate	mindate	Medium	Medium_encode
1	115	2015-03-03	1033	Fred	Smathers	2016-06-28	2015-03-03	Color lithograph	5
2	121	2015-09-21	1015	Tiffany	Twilight	2017-08-29	2014-11-07	Watercolor on Paper	5
3	125	2015-11-21	1001	David	Smith	2016-05-18	2014-11-17	High Quality Limited Print	1
4	127	2015-11-21	1034	Mary Beth	Frederickson	2015-11-21	2014-11-17	High Quality Limited Print	1
5	128	2015-11-21	1036	Selma	Warning	2016-09-29	2015-11-21	High Quality Limited Print	1
6	129	2015-11-21	1036	Selma	Warning	2016-09-29	2015-11-21	High Quality Limited Print	1
7	151	2016-05-07	1036	Selma	Warning	2016-09-29	2015-11-21	Watercolor and ink	3
8	152	2016-05-18	1001	David	Smith	2016-05-18	2014-11-17	High Quality Limited Print	1
9	153	2016-05-18	1001	David	Smith	2016-05-18	2014-11-17	High Quality Limited Print	1
10	154	2016-05-18	1040	Donald	Gray	2017-02-28	2016-05-18	High Quality Limited Print	1
11	156	2016-05-18	1040	Donald	Gray	2017-02-28	2016-05-18	High Quality Limited Print	1
12	161	2016-06-28	1033	Fred	Smathers	2016-06-28	2015-03-03	Monotype with tempera	5
13	171	2016-08-23	1000	Jeffrey	Janes	2016-08-23	2014-11-04	Oil and collage	4
14	175	2016-09-29	1036	Selma	Warning	2016-09-29	2015-11-21	Casein rice paper colla...	5
15	201	2017-02-28	1040	Donald	Gray	2017-02-28	2016-05-18	Color aquatint	2
16	202	2017-02-28	1040	Donald	Gray	2017-02-28	2016-05-18	Color aquatint	2
17	225	2017-06-08	1051	Chris	Wilkens	2017-06-08	2017-06-08	High Quality Limited Print	1
18	227	2017-06-08	1051	Chris	Wilkens	2017-06-08	2017-06-08	High Quality Limited Print	1
19	241	2017-08-29	1015	Tiffany	Twilight	2017-08-29	2014-11-07	Tempera on card	5

Part 2: Data Warehousing and OLAP

Create tables for the DataBase:

part 2 create tabl...nnaAllahAhmed (51))

```
USE HSD_DSA
GO

CREATE TABLE TimeLine (
      TimeID                Int                 NOT NULL,
      DateText              Char(25)            NOT NULL,
      MonthNumber           int                 NOT NULL,
      MonthText             Char(30)            NULL,
      QrtNumber             Numeric(4)          NULL,
      QrtText               char(25)            NULL,
      Year_                 int                 NOT NULL,
      CONSTRAINT   TimeIDPK                     PRIMARY KEY(TimeID)

      );
CREATE TABLE Customer (
      CustomerID            Int                 NOT NULL ,
      CustomerName          Char(35)            NOT NULL,
      Email                 Char(30)            NOT NULL,
      Phone                 int                 NOT NULL,
      City                  char(50)            NULL ,
      State                 Char(25)            NOT NULL,
      Zip                   char(25)            NULL,
      CONSTRAINT   CustomerIDPK                 PRIMARY KEY(CustomerID)

      );
CREATE TABLE ProductNumberTable (
      ProductNumber         Char(25)            NOT NULL,
```

100 %

Messages

```
Commands completed successfully.
```

100 %

Query executed successfully.

Insert into product sales table:

```
INSERT INTO ProductSales VALUES (43023, 3, 'VB001', 1, 7.99, 7.99);
INSERT INTO ProductSales VALUES (43023, 3, 'VK001', 1, 14.95, 14.95);
INSERT INTO ProductSales VALUES (43033, 4, 'BK001', 1, 24.95, 24.95);
INSERT INTO ProductSales VALUES (43033, 4, 'VB001', 1, 7.99, 7.99);
INSERT INTO ProductSales VALUES (43033, 4, 'VK001', 1, 14.95, 14.95);
INSERT INTO ProductSales VALUES (43089, 7, 'VK004', 1, 24.95, 24.95);
INSERT INTO ProductSales VALUES (43184, 4, 'BK002', 1, 24.95, 24.95);
INSERT INTO ProductSales VALUES (43184, 4, 'VK002', 1, 14.95, 14.95);
INSERT INTO ProductSales VALUES (43184, 4, 'VK004', 1, 24.95, 24.95);
INSERT INTO ProductSales VALUES (43186, 6, 'BK002', 1, 24.95, 24.95);
INSERT INTO ProductSales VALUES (43186, 6, 'VB003', 1, 9.99, 9.99);
INSERT INTO ProductSales VALUES (43186, 6, 'VK002', 1, 14.95, 14.95);
INSERT INTO ProductSales VALUES (43186, 6, 'VK003', 1, 19.95, 19.95);
INSERT INTO ProductSales VALUES (43186, 6, 'VK004', 1, 24.95, 24.95);
```

Insert into Timeline table:

```
INSERT INTO TimeLine VALUES (43023, '15-OCT-2017', 10, 'October', 3, 'Qtr3', 2017);
INSERT INTO TimeLine VALUES (43033, '25-OCT-2017', 10, 'October', 3, 'Qtr3', 2017);
INSERT INTO TimeLine VALUES (43089, '20-DEC-2017', 12, 'December', 3, 'Qtr3', 2017);
INSERT INTO TimeLine VALUES (43184, '25-MAR-2018', 3, 'March', 1, 'Qtr1', 2018);
INSERT INTO TimeLine VALUES (43186, '27-MAR-2018', 3, 'March', 1, 'Qtr1', 2018);
INSERT INTO TimeLine VALUES (43190, '31-MAR-2018', 3, 'March', 1, 'Qtr1', 2018);
INSERT INTO TimeLine VALUES (43193, '03-APR-2018', 4, 'April', 2, 'Qtr2', 2018);
INSERT INTO TimeLine VALUES (43198, '08-APR-2018', 4, 'April', 2, 'Qtr2', 2018);
INSERT INTO TimeLine VALUES (43213, '23-APR-2018', 4, 'April', 2, 'Qtr2', 2018);
INSERT INTO TimeLine VALUES (43227, '07-MAY-2018', 5, 'May', 2, 'Qtr2', 2018);
INSERT INTO TimeLine VALUES (43241, '21-MAY-2018', 5, 'May', 2, 'Qtr2', 2018);
INSERT INTO TimeLine VALUES (43256, '05-JUN-2018', 6, 'June', 2, 'Qtr2', 2018);
```

Insert into customer table:

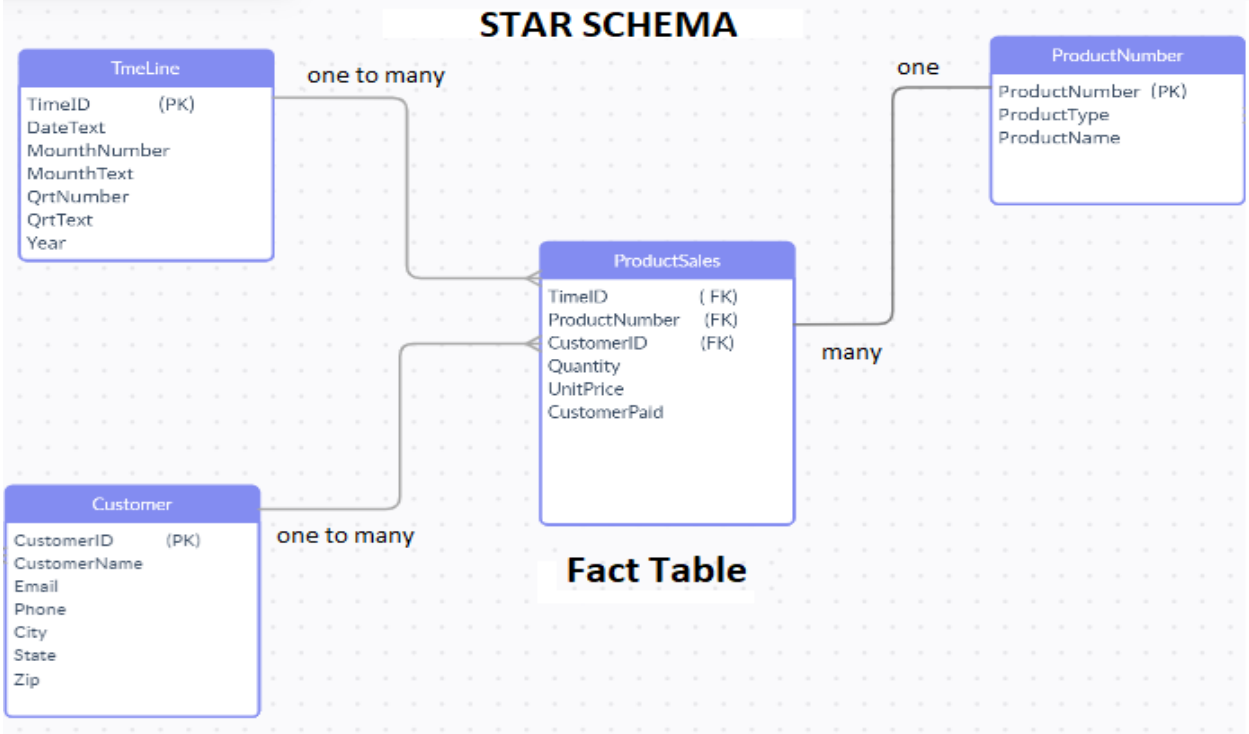
```
INSERT INTO Customer VALUES (1, 'Jacobs, Nancy', 'somewhere.com', '817', 'Fort Worth', 'TX', '76110');
INSERT INTO Customer VALUES (2, 'Jacobs, Chantel', 'somewhere.com', '817', 'Fort Worth', 'TX', '76112');
INSERT INTO Customer VALUES (3, 'Able, Ralph', 'somewhere.com', '210', 'San Antonio', 'TX', '78214');
INSERT INTO Customer VALUES (4, 'Baker, Susan', 'elsewhere.com', '210', 'San Antonio', 'TX', '78216');
INSERT INTO Customer VALUES (5, 'Eagleton, Sam', 'elsewhere.com', '210', 'San Antonio', 'TX', '78218');
INSERT INTO Customer VALUES (6, 'Foxtrot, Kathy', 'somewhere.com', '972', 'Dallas', 'TX', '75220');
INSERT INTO Customer VALUES (7, 'George, Sally', 'somewhere.com', '972', 'Dallas', 'TX', '75223');
INSERT INTO Customer VALUES (8, 'Hullett, Shawn', 'elsewhere.com', '972', 'Dallas', 'TX', '75224');
INSERT INTO Customer VALUES (9, 'Pearson, Bobbi', 'elsewhere.com', '512', 'Austin', 'TX', '78710');
INSERT INTO Customer VALUES (10, 'Ranger, Terry', 'somewhere.com', '512', 'Austin', 'TX', '78712');
INSERT INTO Customer VALUES (11, 'Tyler, Jenny', 'somewhere.com', '972', 'Dallas', 'TX', '75225');
INSERT INTO Customer VALUES (12, 'Wayne, Joan', 'elsewhere.com', '817', 'Fort Worth', 'TX', '76115');
```

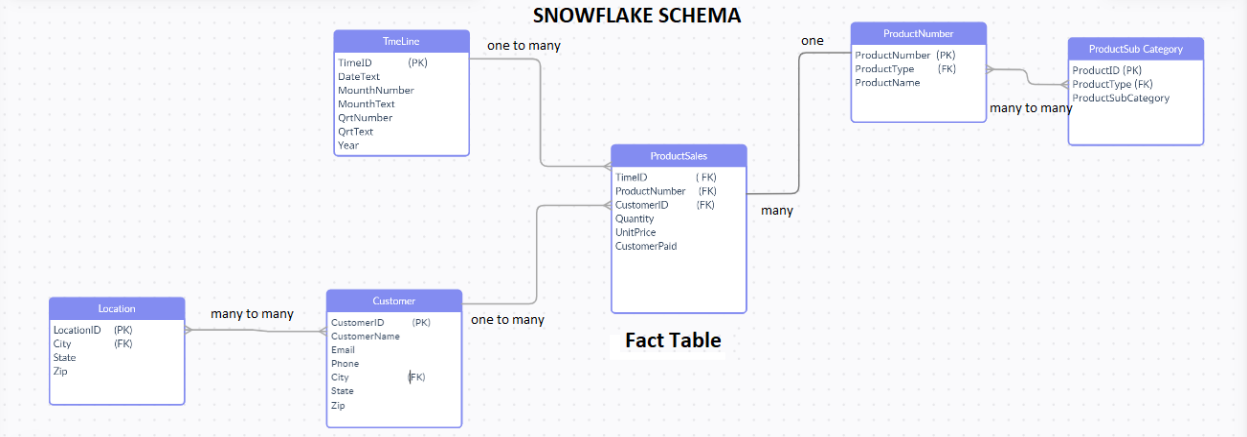
Insert into product number table:

insert in to produ...nnaAllahAhmed (55)) | insert product nu...naAllahAhmed (51))

```
INSERT INTO ProductNumberTable VALUES ('BK001', 'Book', 'Kitchen Remodeling Basics For Everyone');
INSERT INTO ProductNumberTable VALUES ('BK002', 'Book', 'Advanced Kitchen Remodeling For Everyone');
INSERT INTO ProductNumberTable VALUES ('BK003', 'Book', 'Kitchen Remodeling Dallas Style For Everyone');
INSERT INTO ProductNumberTable VALUES ('VB001', 'Video Companion', 'Kitchen Remodeling Basics');
INSERT INTO ProductNumberTable VALUES ('VB002', 'Video Companion', 'Advanced Kitchen Remodeling I');
INSERT INTO ProductNumberTable VALUES ('VB003', 'Video Companion', 'Kitchen Remodeling Dallas Style');
INSERT INTO ProductNumberTable VALUES ('VK001', 'Video', 'Kitchen Remodeling Basics');
INSERT INTO ProductNumberTable VALUES ('VK002', 'Video', 'Advanced Kitchen Remodeling');
INSERT INTO ProductNumberTable VALUES ('VK003', 'Video', 'Kitchen Remodeling Dallas Style');
INSERT INTO ProductNumberTable VALUES ('VK004', 'Video', 'Heather Sweeney Seminar Live in Dallas on 25-OCT-16');
```

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





2: a: Write an SQL query to answer the following question: ”Which customer(s) made an order containing at least five products with different product numbers?” Provide the CustomerName and CustomerID

Query

```
part 2 Q2.a - DESK...naAllahAhmed (53))  |  part 2 Q 2.c - DES...nnaAllahAhmed (51))
--Question 2 . a
Select c.CustomerName ,c.CustomerID ,sum(p.Quantity) as Quantity  From Customer as c
inner join ProductSales as p on c.CustomerID = p.CustomerID
where Quantity = 1

GROUP BY c.CustomerID,c.CustomerName

HAVING SUM(Quantity) >= 5
```

Result of Query

100 %

Results | Messages

	CustomerName	CustomerID	Quantity
1	Able, Ralph	3	6
2	Baker, Susan	4	6
3	Foxtrot, Kathy	6	5
4	George, Sally	7	5
5	Pearson, Bobbi	9	5
6	Wayne, Joan	12	5

2:b: Write an SQL query for the following report: ”Which customer(s) made the largest order, i.e., those that would result in the largest bill?”

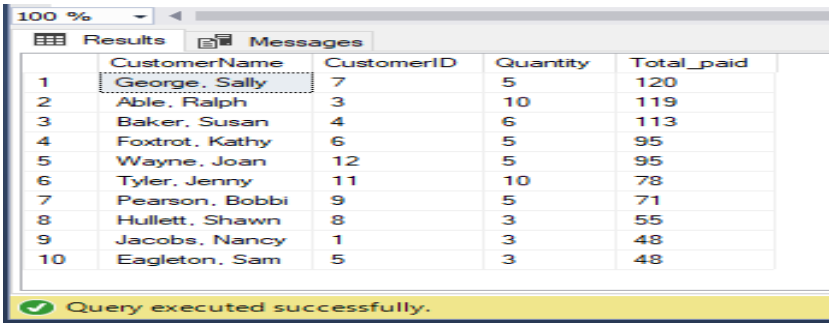
Query

```
part 2Q 2c.sql - D...nnaAllahAhmed (51))
-- Question 2.b
Select c.CustomerName ,c.CustomerID ,sum(p.Quantity)as Quantity ,SUM(p.UnitPrice) as Total_paid  From Customer as c
inner join ProductSales as p on c.CustomerID = p.CustomerID

GROUP BY c.CustomerID,c.CustomerName

ORDER BY Total_paid DESC
```

Result of Query

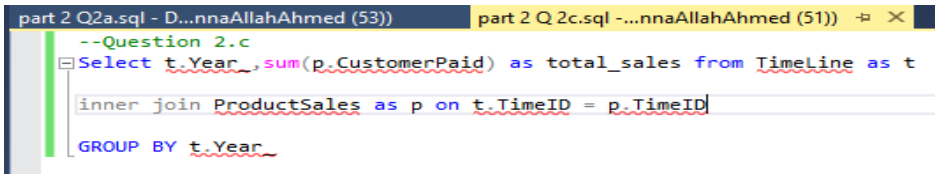


	CustomerName	CustomerID	Quantity	Total_paid
1	George, Sally	7	5	120
2	Able, Ralph	3	10	119
3	Baker, Susan	4	6	113
4	Foxtrot, Kathy	6	5	95
5	Wayne, Joan	12	5	95
6	Tyler, Jenny	11	10	78
7	Pearson, Bobbi	9	5	71
8	Hullett, Shawn	8	3	55
9	Jacobs, Nancy	1	3	48
10	Eagleton, Sam	5	3	48

Query executed successfully.

2:c: SQL queries for the ”Roll-Up” operation to summarise the total sales per Year.

Query

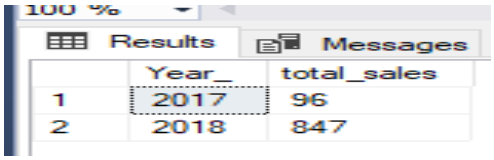


```
--Question 2.c
Select t.Year_,sum(p.CustomerPaid) as total_sales from TimeLine as t

inner join ProductSales as p on t.TimeID = p.TimeID

GROUP BY t.Year_
```

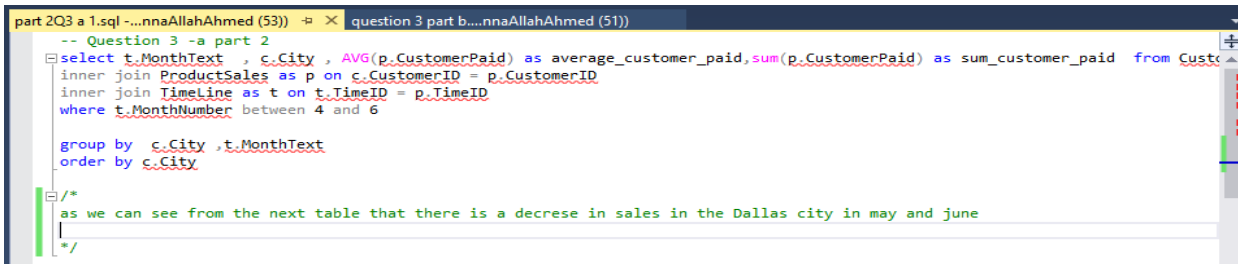
Result of Query



	Year_	total_sales
1	2017	96
2	2018	847

3:a: What are the aggregates that the analyst would start with?

Query



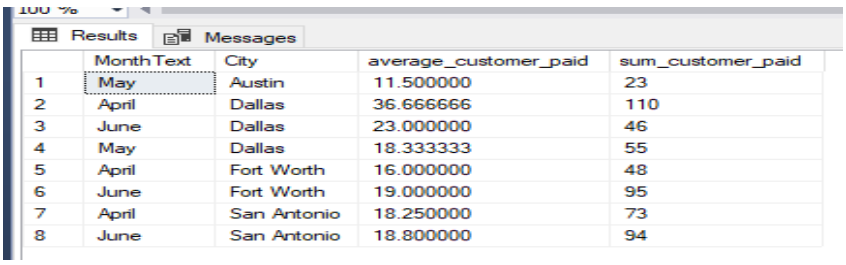
```
-- Question 3 -a part 2
select t.MonthText  , c.City , AVG(p.CustomerPaid) as average_customer_paid,sum(p.CustomerPaid) as sum_customer_paid  from Custo
inner join ProductSales as p on c.CustomerID = p.CustomerID
inner join TimeLine as t on t.TimeID = p.TimeID
where t.MonthNumber between 4 and 6

group by  c.City ,t.MonthText
order by c.City

/*
as we can see from the next table that there is a decrese in sales in the Dallas city in may and june

*/
```

Result of Query



	MonthText	City	average_customer_paid	sum_customer_paid
1	May	Austin	11.500000	23
2	April	Dallas	36.666666	110
3	June	Dallas	23.000000	46
4	May	Dallas	18.333333	55
5	April	Fort Worth	16.000000	48
6	June	Fort Worth	19.000000	95
7	April	San Antonio	18.250000	73
8	June	San Antonio	18.800000	94

Query

```
/*
here the analyst want to aggregate by the total sales for each product to see what happed during this three month
*/
-- Question 3 -b part 2
select n.ProductType,t.MonthText , AVG(p.CustomerPaid) as average_sales ,sum(p.CustomerPaid) as sum_of_sales from ProductSales
inner join ProductNumberTable as n on n.ProductNumber = p.ProductNumber
inner join TimeLine as t on t.TimeID = p.TimeID
where t.MonthNumber between 4 and 6
GROUP BY n.ProductType,t.MonthText
order by n.ProductType

/*
in the book product there is a decrease in sales from April to june
in the video product it achive a good sales in april , decrease in may and achive the best sales in june
in the video campanion product it achive a good sales in april , decrease in may and achive the best sales in june
*/
```

Result of Query

Results | Messages

	ProductType	MonthText	average_sales	sum_of_sales
1	Book	April	25.000000	75
2	Book	June	25.000000	50
3	Video	April	30.000000	120
4	Video	June	22.500000	135
5	Video	May	20.000000	60
6	Video Companion	April	12.000000	36
7	Video Companion	June	12.500000	50
8	Video Companion	May	9.000000	18

3:b: What are the relevant “drill-down” operations that the analyst would need to execute?

Query

```
--Question 3 part 2 b
/*
here in the nex query it will contain a detail report about total sales fro each state categorized
by the product type so we can see why dallas achive the high sales in april due to the video prouct
*/

Select c.City ,p.ProductType,t.MonthText,sum(s.CustomerPaid) as totalsales from Customer as c
inner join ProductSales s on c.CustomerID = s.CustomerID
inner join TimeLine t on s.TimeID = t.TimeID
inner join ProductNumberTable p on p.ProductNumber = s.ProductNumber
where t.MonthNumber between 4 and 6
group by c.City ,p.ProductType,t.MonthText
```

Result of Query

100 %

Results | Messages

	City	ProductType	MonthText	totalsales
1	Austin	Video	May	15
2	Austin	Video Companion	May	8
3	Dallas	Video	April	90
4	Dallas	Video	June	30
5	Dallas	Video	May	45
6	Dallas	Video Companion	April	20
7	Dallas	Video Companion	June	16
8	Dallas	Video Companion	May	10
9	Fort Worth	Book	April	25
10	Fort Worth	Book	June	25
11	Fort Worth	Video	April	15

Query executed successfully.

4. Using R, read the dimensions files and the Product_Sales fact table. Build an OLAP cube for the Sum of Total Quantity:

Merging the 3 tables:

```
#Merging Tables

Time_sales_df <- merge(x=TimeLine_table,y=ProductSales_table,by = "TimeID")

Customers_Sales <- merge(x=Customer_table,y=Time_sales_df,by="CustomerID")

Fact_Quantity <- merge(x=Product_table, y=Customers_Sales, by="ProductNumber")
```

Building the cube:

```
#Building Cube

Quantity_cube <- tapply(Fact_Quantity$Quantity,
                        Fact_Quantity[,c("ProductType", "Year", "City")],
                        FUN=function(x){return(sum(x))})

Quantity_cube[is.na(Quantity_cube)] <- 0
Quantity_cube


# Showing the cells of the cube
Quantity_cube
head(Quantity_cube)

# Showing the cells of a subset of the cells
dimnames(Quantity_cube)
```

```
R 4.2.0 · C:/Users/mm/Downloads/data science/Tut4-OLAP/Tut4-OLAP/
+                         Fact_Quantity[,c("ProductType", "Year", "City")],
+                         FUN=function(x){return(sum(x))})
>
> Quantity_cube[is.na(Quantity_cube)] <- 0
> Quantity_cube
, , City = Austin

                Year
ProductType      2017 2018
  Book              0    1
  Video             0    2
  Video Companion   0    2

, , City = Dallas

                Year
ProductType      2017 2018
  Book              0    3
  Video             1   13
  Video Companion   0    6

, , City = Fort Worth

                Year
ProductType      2017 2018
  Book              0    2
  Video             0    4
  Video Companion   0    2

, , City = San Antonio

                Year
ProductType      2017 2018
  Book              1    4
  Video             2    6
  Video Companion   2    4

>
```

Showing the cells of a subset of the cells:

```
> dimnames(Quantity_cube)
$ProductType
[1] "Book"             "Video"            "Video Companion"

$Year
[1] "2017" "2018"

$City
[1] "Austin"      "Dallas"      "Fort Worth"  "San Antonio"

>
```