DTI 5126: Fundamentals for Applied Data Science Summer 2022 Assignment 1

Part A: RDBMS & SQL:

A) Identify transactions with null values on the DateSoldID and remove them from the table.

Solution: `Sql Query (Delete from TRANS where DateSold is NULL)`

Output:

Before:

	Transactio...	DateAcquir...	AcquisitionPr...	AskingPri...	DateSold	SalesPri...	Customer...	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
10	126	2015-11-21	200.00	400.00	NULL	NULL	NULL	552
11	127	2015-11-21	125.00	500.00	2015-12-22	400.00	1034	553
12	128	2015-11-21	125.00	250.00	2016-03-16	225.00	1036	554
13	129	2015-11-21	125.00	250.00	2016-03-16	225.00	1036	555
14	151	2016-05-07	10000.00	20000.00	2016-06-28	17500.00	1036	561
15	152	2016-05-18	125.00	250.00	2016-08-15	225.00	1001	562
16	153	2016-05-18	200.00	400.00	2016-08-15	350.00	1001	563
17	154	2016-05-18	250.00	500.00	2016-09-28	400.00	1040	564
18	155	2016-05-18	250.00	500.00	NULL	NULL	NULL	565
19	156	2016-05-18	250.00	500.00	2016-09-27	400.00	1040	566
20	161	2016-06-28	7500.00	15000.00	2016-09-29	13750.00	1033	570
21	171	2016-08-23	35000.00	60000.00	2016-09-29	55000.00	1000	571
22	175	2016-09-29	40000.00	75000.00	2016-12-18	72500.00	1036	500
23	181	2016-10-11	250.00	500.00	NULL	NULL	NULL	578
24	201	2017-02-28	2000.00	3500.00	2017-04-26	3250.00	1040	580
25	202	2017-02-28	2000.00	3500.00	2017-04-26	3250.00	1040	581
26	225	2017-06-08	125.00	250.00	2017-09-27	225.00	1051	585
27	226	2017-06-08	200.00	400.00	NULL	NULL	NULL	586
28	227	2017-06-08	250.00	500.00	2017-09-27	475.00	1051	587
29	228	2017-06-08	250.00	500.00	NULL	NULL	NULL	588
30	229	2017-06-08	250.00	500.00	NULL	NULL	NULL	589
31	241	2017-08-29	2500.00	5000.00	2017-09-27	4750.00	1015	590
32	251	2017-10-25	25000.00	50000.00	NULL	NULL	NULL	593
33	252	2017-10-27	250.00	500.00	NULL	NULL	NULL	594
34	253	2017-10-27	250.00	500.00	NULL	NULL	NULL	595
35	254	2017-10-27	250.00	500.00	NULL	NULL	NULL	596

After:

	Transactio...	DateAcquir...	AcquisitionPr...	AskingPri...	DateSold	SalesPri...	Customer...	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
10	127	2015-11-21	125.00	500.00	2015-12-22	400.00	1034	553
11	128	2015-11-21	125.00	250.00	2016-03-16	225.00	1036	554
12	129	2015-11-21	125.00	250.00	2016-03-16	225.00	1036	555
13	151	2016-05-07	10000.00	20000.00	2016-06-28	17500.00	1036	561
14	152	2016-05-18	125.00	250.00	2016-08-15	225.00	1001	562
15	153	2016-05-18	200.00	400.00	2016-08-15	350.00	1001	563
16	154	2016-05-18	250.00	500.00	2016-09-28	400.00	1040	564
17	156	2016-05-18	250.00	500.00	2016-09-27	400.00	1040	566
18	161	2016-06-28	7500.00	15000.00	2016-09-29	13750.00	1033	570
19	171	2016-08-23	35000.00	60000.00	2016-09-29	55000.00	1000	571
20	175	2016-09-29	40000.00	75000.00	2016-12-18	72500.00	1036	500
21	201	2017-02-28	2000.00	3500.00	2017-04-26	3250.00	1040	580
22	202	2017-02-28	2000.00	3500.00	2017-04-26	3250.00	1040	581
23	225	2017-06-08	125.00	250.00	2017-09-27	225.00	1051	585
24	227	2017-06-08	250.00	500.00	2017-09-27	475.00	1051	587
25	241	2017-08-29	2500.00	5000.00	2017-09-27	4750.00	1015	590

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.

Solution: Sql Query

```
(select WorkID, Title, Medium, ARTIST.ArtistID, CONCAT(FirstName, ' ',LastName) as
FullName from WORK join ARTIST on WORK.ArtistID = ARTIST.ArtistID where Title like
'%Yellow%' or Title like '%Blue%' or Title like '%White%')
```

Output:

	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.

Solution: Sql query

```
(SELECT YEAR(DateSold) as Year, ArtistID, sum(SalesPrice) as SumOfSubTotal,
avg(SalesPrice) as AverageOfSubtotal from TRANSjoin CUSTOMER_ARTIST_INT on
TRANS.CustomerID = CUSTOMER_ARTIST_INT.CustomerID group by ArtistID,
YEAR(DateSold) order by ArtistID)
```

Output:

	Year	ArtistID	SumOfSubT...	AverageOfSubt...
1	2015	1	1800.00	300.000000
2	2016	1	575.00	287.500000
3	2015	2	1800.00	300.000000
4	2016	2	575.00	287.500000
5	2015	4	1800.00	300.000000
6	2016	4	575.00	287.500000
7	2015	5	1800.00	300.000000
8	2016	5	91025.00	15170.833333
9	2014	11	500.00	500.000000
10	2015	11	28100.00	7025.000000
11	2016	11	91025.00	15170.833333
12	2017	11	4750.00	4750.000000
13	2014	17	43000.00	21500.000000
14	2015	17	30250.00	15125.000000
15	2016	17	69550.00	17387.500000
16	2017	17	11950.00	2390.000000
17	2014	18	43000.00	21500.000000
18	2015	18	30250.00	15125.000000
19	2016	18	69550.00	17387.500000
20	2017	18	11950.00	2390.000000
21	2014	19	43000.00	21500.000000
22	2015	19	30250.00	15125.000000
23	2016	19	160000.00	20000.000000
24	2017	19	11950.00	2390.000000

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

Solution: Sql query

```
(select ARTIST.ArtistID, FirstName, LastName, WORK.WorkID, Title from ARTIST
inner join WORK on Artist.ArtistID = WORK.ArtistID
inner join TRANS on TRANS.WorkID = WORK.WorkID
where SalesPrice > (select avg(SalesPrice) from TRANS))
```

Output:

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

(6 row(s) affected)
```

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

Solution: Sql query

```
(update CUSTOMER set EmailAddress = 'Johnson.lynda@somewhere.com',
EncryptedPassword = 'aax1xbB'
where FirstName = 'Lynda' and LastName = 'Johnson')
```

Output:

Before:

	CustomerID	LastName	FirstName	EmailAddress	EncryptedPassw...	Street	City	St...	ZIPorPostalCo...	Country	AreaCo...	PhoneNum...
1	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng76tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345
2	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876
3	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566
4	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911
5	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822
6	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0512
7	1037	Wu	Susan	Susan.Wu@somewhere.com	Ues3thQ2	105 Locust Ave	Atlanta	GA	30322	USA	404	653-3465
8	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839
9	1041	Johnson	Lynda	NULL	NULL	117 C Street	Washington	DC	20003	USA	202	438-5498
10	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8822

After:

	CustomerID	LastName	FirstName	EmailAddress	EncryptedPassw...	Street	City	St...	ZIPorPostalCo...	Country	AreaCo...	PhoneNum...
1	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng76tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345
2	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876
3	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566
4	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911
5	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822
6	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0512
7	1037	Wu	Susan	Susan.Wu@somewhere.com	Ues3thQ2	105 Locust Ave	Atlanta	GA	30322	USA	404	653-3465
8	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839
9	1041	Johnson	Lynda	Johnson.lynda@somewhere.com	aax1xbB	117 C Street	Washington	DC	20003	USA	202	438-5498
10	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8822

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.

Solution: Sql query

```
(select Customer.CustomerID, LastName, FirstName, EmailAddress, EncryptedPassword,
Street,City, State, ZIPorPostalCode, Country, AreaCode, PhoneNumber,
DATEDIFF(DAY, DateSold, LEAD(DateSold) OVER(ORDER BY DateSold)) as Days_Difference
from TRANS join CUSTOMER on TRANS.CustomerID = CUSTOMER.CustomerID)
```

Output:

	CustomerID	LastName	FirstName	EmailAddress	EncryptedPassw...	Street	City	St...	ZIPorPostalCo...	Coun...	AreaCo...	PhoneNum...	Days_Differe...
1	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng76tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345	5
2	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	30
3	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	0
4	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	140
5	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911	174
6	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	14
7	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	0
8	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	6
9	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	4
10	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	85
11	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Cana...	604	988-0512	0
12	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Cana...	604	988-0512	104
13	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Cana...	604	988-0512	48
14	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	0
15	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	43
16	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	1
17	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	1
18	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911	0
19	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng76tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345	80
20	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Cana...	604	988-0512	129
21	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	0
22	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	154
23	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8822	0
24	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8822	0
25	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	NULL

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).

Solution: Sql query

```
(CREATE VIEW CustomerTransactionSummaryView
  as
      as select CONCAT(LastName, ' ', FirstName) as Full_Name, Title,
      DateAcquired, DateSold, SalesPrice - AcquisitionPrice as Profit,AskingPrice
      from CUSTOMER
      join TRANS on CUSTOMER.CustomerID = TRANS.CustomerID
      join WORK on TRANS.WorkID = WORK.WorkID
      where AskingPrice > 20000)

      (select * from CustomerTransactionSummaryView
      order by CustomerTransactionSummaryView.AskingPrice desc)
```

Output:

Full_Name		Title	DateAcquired	DateSold	Profit	AskingPrice
Warning	Selma	Memories IV	2016-09-29	2016-12-18	32500.00	75000.00
Janes	Jeffrey	Yellow Covers Blue	2016-08-23	2016-09-29	20000.00	60000.00
Janes	Jeffrey	Memories IV	2014-11-04	2014-12-14	12500.00	45000.00
Twilight	Tiffany	Night Bird	2015-09-21	2015-11-28	12500.00	30000.00

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.

Sql queries:

```
(select TransactionID, CUSTOMER.CustomerID,
 DateAcquired, LastName, FirstName, Medium
 into Purchase from CUSTOMER join TRANS
 on CUSTOMER.CustomerID = TRANS.CustomerID
 join WORK on TRANS.WorkID = WORK.WorkID
 where YEAR(DateSold) BETWEEN 2015 AND 2017)

(update Purchase set Medium = CASE
When Medium = 'High Quality Limited Print' then 1
When Medium = 'Color Aquatint' then 2
When Medium = 'Water Color and Ink' then 1
When Medium = 'Oil and Collage' then 1
else 5
end)
```

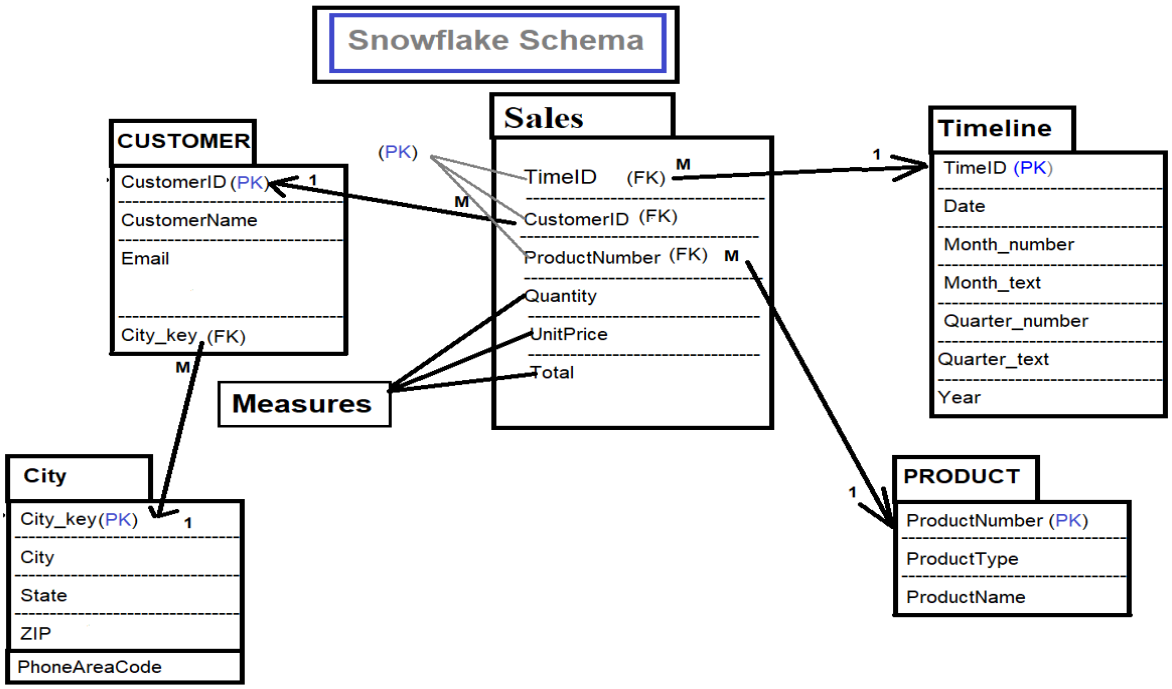
Output:

	Transactio...	Customer...	DateAcquir...	LastName	FirstName	Medium
1	102	1001	2014-11-17	Smith	David	1
2	103	1034	2014-11-17	Frederickson	Mary Beth	1
3	104	1001	2014-11-17	Smith	David	1
4	105	1034	2014-11-17	Frederickson	Mary Beth	1
5	115	1033	2015-03-03	Smathers	Fred	5
6	121	1015	2015-09-21	Twilight	Tiffany	5
7	125	1001	2015-11-21	Smith	David	1
8	127	1034	2015-11-21	Frederickson	Mary Beth	1
9	128	1036	2015-11-21	Warning	Selma	1
10	129	1036	2015-11-21	Warning	Selma	1
11	151	1036	2016-05-07	Warning	Selma	5
12	152	1001	2016-05-18	Smith	David	1
13	153	1001	2016-05-18	Smith	David	1
14	154	1040	2016-05-18	Gray	Donald	1
15	156	1040	2016-05-18	Gray	Donald	1
16	161	1033	2016-06-28	Smathers	Fred	5
17	171	1000	2016-08-23	Janes	Jeffrey	1
18	175	1036	2016-09-29	Warning	Selma	5
19	201	1040	2017-02-28	Gray	Donald	2
20	202	1040	2017-02-28	Gray	Donald	2
21	225	1051	2017-06-08	Wilkens	Chris	1
22	227	1051	2017-06-08	Wilkens	Chris	1
23	241	1015	2017-08-29	Twilight	Tiffany	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).

   Solution:



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.

Solution: Sql query

```
(select CustomerName, CustomerID from CUSTOMER
 where CustomerID in(select CustomerID from PRODUCT_SALES
                     group by TimeID, CustomerID
                     having count(DISTINCT ProductNumber) >= 5))
```

Output:

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

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?”

Solution: Sql query

```
(select * from CUSTOMER
 where CustomerID in(select top(1) CustomerID
                     from PRODUCT_SALES
                     group by CustomerID
                     order by Sum(Total) desc))
```

Output:

Results | Messages

	CustomerID	CustomerNa...	Email	PhoneAreaCo...	City	St...	ZIP
1	11	Tyler, Jenny	somewhere.com	972	Dallas	TX	75225

C) Write SQL queries for the ”Roll-Up” operation to summarise the total sales per Year.

Solution: Sql query

```
select  T.TYear, sum(total) 'the
total sales per Year'
from PRODUCT_SALES PS join TIMELINE
T
on T.TimeID = PS.TimeID
group by rollup(T.TYear)
```

Output:

Results | Messages

	TYear	the total sales per Year
1	2017	95.78
2	2018	845.11
3	NULL	940.89

```
(select T.TYear, T.N_Month, sum(total) 'the
total sales per Year'
from PRODUCT_SALES PS join TIMELINE T
on T.TimeID = PS.TimeID
group by rollup(T.TYear, T.N_Month))
```

Output:

Results | Messages

	TYear	N_Mo...	the total sales per Y...
1	2017	10	70.83
2	2017	12	24.95
3	2017	NULL	95.78
4	2018	3	302.33
5	2018	4	230.51
6	2018	5	77.83
7	2018	6	234.44
8	2018	NULL	845.11
9	NULL	NULL	940.89

3. Suppose an analyst finds that monthly total have decreased from April 2018 to June 2018, instead of growing. The analyst wishes to check if there are specific product type or customer city that are responsible for the decrease.
   Solution:
   A) Solution:
      if the analyst wants to know a specific product type that is responsible for the decrease, he should use Group by on the PRODUCT_SALES table to get the total quantity and cost by using Sum (Quantity) and Sum (Total) for each product so he can know each month in the year the number of products by using group by rollup. And also, he should take this action with the cities' customers too.
   B) Solution : Sql queries

```
(select TYear ,N_Month, P.ProductType, sum(Quantity) 'Quantity',
sum(Total) 'the cost'
from PRODUCT_SALES sales join PRODUCT P
on sales.ProductNumber = P.ProductNumber
join TIMELINE T
on sales.TimeID = T.TimeID
group by rollup(TYear, N_Month, P.ProductType)
having T.TYear = 2018 and T.N_Month in (3,4,5,6))

(select TYear ,N_Month, c.City, sum(Quantity) 'Quantity', sum(Total)
'the cost'
from PRODUCT_SALES sales join CUSTOMER c
on sales.CustomerID = c.CustomerID
join TIMELINE T
on sales.TimeID = T.TimeID
group by rollup (TYear, N_Month, c.City)
having T.TYear = 2018 and T.N_Month in (3,4,5,6))
```

Output:

Product Type

	TYear	N_Mo...	ProductType	Quantity	the cost
1	2018	3	Book	5	124.75
2	2018	3	Video	8	159.6
3	2018	3	Video Companion	2	17.98
4	2018	3	NULL	15	302.33
5	2018	4	Book	3	74.85
6	2018	4	Video	6	119.7
7	2018	4	Video Companion	4	35.96
8	2018	4	NULL	13	230.51
9	2018	5	Video	3	59.85
10	2018	5	Video Companion	2	17.98
11	2018	5	NULL	5	77.83
12	2018	6	Book	2	49.9
13	2018	6	Video	8	134.6
14	2018	6	Video Companion	6	49.94
15	2018	6	NULL	16	234.44

Customer City

	TYear	N_Month	City	Quant...	the cost
1	2018	3	Austin	3	47.89
2	2018	3	Dallas	9	189.59
3	2018	3	San Antonio	3	64.85
4	2018	3	NULL	15	302.33
5	2018	4	Dallas	6	109.78
6	2018	4	Fort Worth	3	47.89
7	2018	4	San Antonio	4	72.84
8	2018	4	NULL	13	230.51
9	2018	5	Austin	2	22.94
10	2018	5	Dallas	3	54.89
11	2018	5	NULL	5	77.83
12	2018	6	Dallas	4	45.88
13	2018	6	Fort Worth	5	94.79
14	2018	6	San Antonio	7	93.77
15	2018	6	NULL	16	234.44

The tables show how many different product kinds were sold in each month (3, 4, 5, and 6). Five books, eight Videos, and two video companions were sold for $303 on Mars. Three books, six Videos, and four video companions were sold for $230.51 in April. Three Videos and two video companions were sold for $77.83 in May. 234.44 sold two books, eight Videos, and six video companions in June. So, for Mars, April, May, and June, the total cost of decline is 302.33, 230.51, 77.83, and 234.44, respectively.

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

Solution:

OLAPcube.R

Setup the dimension tables

```
Timeline_table <-
data.frame (
   TimeID = c (43023, 43033, 43089, 43184, 43186, 43190, 43193, 43198,
               43213, 43227, 43241, 43256),
   TDate = c ('15-OCT-2017', '25-OCT-2017', '20-DEC-2017', '25-MAR-2018', '27-MAR-2018', '31-MAR-2018', '03-APR-2018',
              '08-APR 2018',  '23-APR-2018', '07-MAY-2018', '21-MAY-2018', '05-JUN-2018'),
   N_Month = c (10, 10, 12, 3, 3, 3, 4, 4, 4, 5, 5, 6),
   Month_text = c('October', 'October', 'December', 'March', 'March', 'March', 'April', 'April', 'April', 'May', 'May', 'June'),
   N_Qtr = c (3, 3, 3, 1, 1, 1, 2, 2, 2, 2, 2, 2),
   Qtr = c('Qtr3', 'Qtr3', 'Qtr3', 'Qtr1', 'Qtr1', 'Qtr1', 'Qtr2', 'Qtr2', 'Qtr2', 'Qtr2', 'Qtr2', 'Qtr2'),
   TYear = c (2017, 2017, 2017, 2018, 2018, 2018, 2018, 2018, 2018, 2018,  2018, 2018))
```

```
CUSTOMER_table <-
  data.frame(
    CustomerID = c(1:12),
    CustomerName = c ('Jacobs, Nancy', 'Jacobs, Chantel', 'Able, Ralph', 'Baker, Susan', 'Eagleton, Sam', 'Foxtrot, Kathy',
                      'George, Sally', 'Hullett, Shawn', 'Pearson, Bobbi',   'Ranger, Terry', 'Tyler, Jenny', 'Wayne, Joan'),
    Email = c ('somewhere.com', 'somewhere.com', 'somewhere.com',  'elsewhere.com', 'elsewhere.com', 'somewhere.com',
               'somewhere.com', 'elsewhere.com', 'elsewhere.com',  'somewhere.com', 'somewhere.com', 'elsewhere.com'),
    PhoneAreaCode = c ('817', '817', '210', '210', '210', '972', '972', '972', '512',   '512', '972', '817'),
    City = c ('Fort Worth', 'Fort Worth', 'San Antonio', 'San Antonio', 'San Antonio', 'Dallas', 'Dallas', 'Dallas', 'Austin', 'Austin', 'Dallas', 'Fort
              Worth'),
    State = c('TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX', 'TX'),
    ZIP = c('76110', '76112', '78214', '78216', '78218', '75220', '75223', '75224', '78710', '78712', '75225', '76115'))
```

```
PRODUCT_table <-
  data.frame(
    ProductNumber = c ('BK001', 'BK002', 'BK003', 'VB001', 'VB002', 'VB003',  'VK001', 'VK002', 'VK003', 'VK004'),
    ProductType = c ('Book', 'Book', 'Book', 'Video Companion', 'Video Companion', 'Video Companion', 'Video', 'Video', 'Video',  'Video'),
    ProductName = c ('Kitchen Remodeling Basics For Everyone', 'Advanced Kitchen Remodeling For Everyone', 'Kitchen Remodeling
                      Dallas Style For Everyone', 'Kitchen Remodeling Basics', 'Advanced Kitchen Remodeling I', 'Kitchen Remodeling
                     Dallas Style', 'Kitchen Remodeling Basics', 'Advanced  Kitchen Remodeling', 'Kitchen Remodeling Dallas Style',
                    'Heather Sweeney Seminar Live in Dallas on 25-OCT-16') )
```

```
# Loading
install.packages("readxl")
library(readxl)
#PRODUCT_SALES.xlsx
```

PRODUCT_SALES.xlsx

```
exceldata_PRODUCT_SALES  = read_excel("C:/Users/AhmedPro/Desktop/the_dimension_tables/PRODUCT_SALES.xlsx")
PRODUCT_SALES_table = data.frame(exceldata_PRODUCT_SALES)
```

#Build an OLAP cube for the Sum of Total Quantity.

```
revenue_cube <-
        tapply (PRODUCT_SALES_table$Quantity,
                PRODUCT_SALES_table [, c ("TimeID", "CustomerID", "ProductNumber")],
                FUN = function(x) {return (sum(x))})
```

Replace values of null by 0

```
revenue_cube [is.na (revenue_cube)] <- 0
```

Print Cube

Print (revenue_cube)

Output:

```
, , ProductNumber = BK001

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 1 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 0 1 0 0 0  0
  43190 0 0 0 0 0 0 0 1 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 1 0 0 1 0 0 0 0 0  0
  43213 0 1 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 0 0 0 0 0 0 0 0  0
```

```
, , ProductNumber = BK002

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 1 0 0 0 0 0 0  0
  43186 0 0 0 0 1 1 0 0 0  0
  43190 0 0 0 0 0 0 0 0 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 0 0 0 0 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 1 0 0 0 0 0 0 0  1
```

```
, , ProductNumber = VB001

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 1 0 0 0 0 0 0 0  0
  43033 0 0 1 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 0 0 0 0 0  0
  43190 0 0 0 0 0 0 0 1 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 1 0 0 1 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 1 0 0 0 0 0 0 0  0
```

```
, , ProductNumber = VB002

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 0 0 0 0 0  0
  43190 0 0 0 0 0 0 0 0 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 0 0 0 0 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 1 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 2 0 0 0 0 0 0 2  0
```

```
, , ProductNumber = VB003

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 1 0 0 0 0  0
  43190 0 0 0 0 0 0 0 0 0  0
  43193 0 0 0 0 0 0 0 0 2  0
  43198 0 0 0 0 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 1 0  0
  43256 0 0 0 0 0 0 0 0 0  1
```

```
, , ProductNumber = VK001

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 1 0 0 0 0 0 0 0  0
  43033 0 0 1 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 0 0 0 0 0  0
  43190 0 0 0 0 0 0 0 1 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 1 0 0 1 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 1 0 0 0 0 0 0 0  0
```

```
, , ProductNumber = VK002

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 1 0 0 0 0 0 0  0
  43186 0 0 0 0 1 0 0 0 0  0
  43190 0 0 0 0 0 0 0 0 0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 0 0 0 0 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 1 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 2 0 0 0 0 0 0 2  1
```

```
, , ProductNumber = Vk003

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
  43186 0 0 0 0 0 1 0 0 0  0
  43190 0 0 0 0 0 0 0 0.0  0
  43193 0 0 0 0 0 0 0 0 0  0
  43198 0 0 0 0 0 0 0 0 0  0
  43213 0 0 0 0 0 0 0 0 0  0
  43227 0 0 0 0 0 0 0 0 0  0
  43241 0 0 0 0 0 0 0 0 0  0
  43256 0 0 0 0 0 0 0 0 0  0
```

```
, , ProductNumber = VK003

       CustomerID
TimeID  1 3 4 5 6 7 8 9 11 12
  43023 0 0 0 0 0 0 0 0 0  0
  43033 0 0 0 0 0 0 0 0 0  0
  43089 0 0 0 0 0 0 0 0 0  0
  43184 0 0 0 0 0 0 0 0 0  0
```