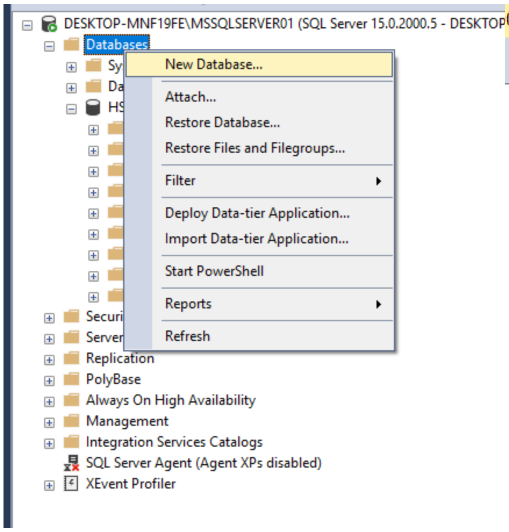
DTI5126[EG] Fundamentals/Applied Data Science

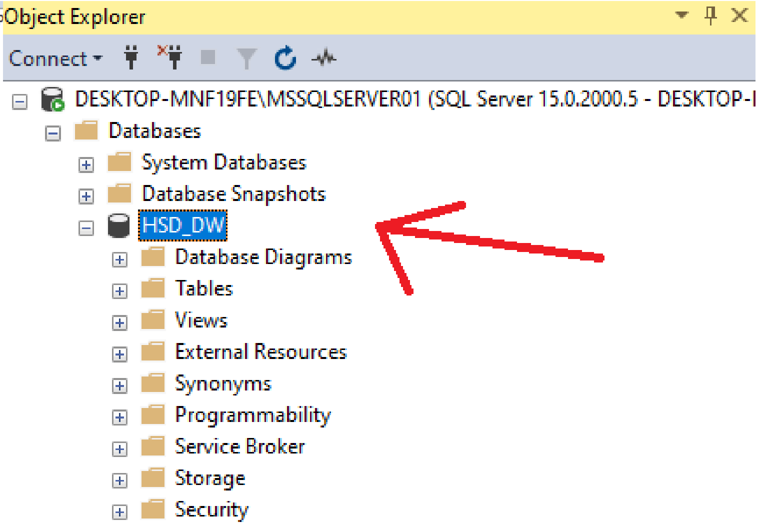
Assignment 1

1- Hosam Mahmoud Ibrahim - 300327269

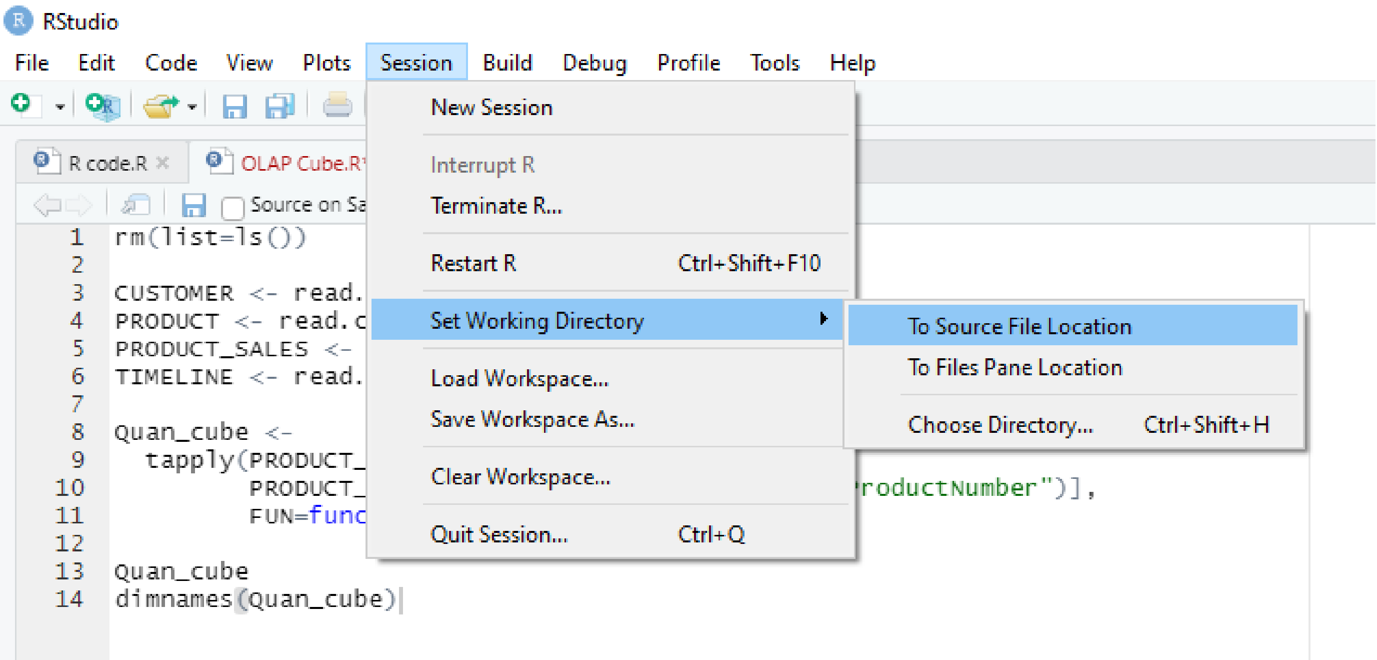
2- Abdallah Ahmed Ragab - 300327288

Please make sure that the database is created in .sql files to run as expected.

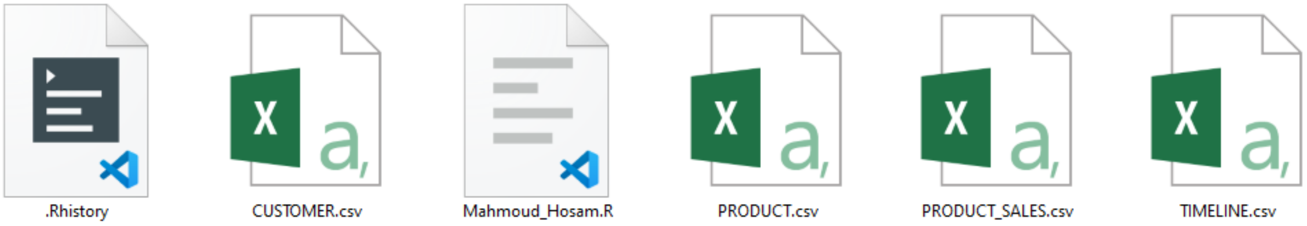

DESKTOP-MNF19FE\MSSQLSERVER01 (SQL Server 15.0.2000.5 - DESKTOP
Databases
Sy
Da
HS
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh
Securi
Server
Replication
PolyBase
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler



And please make sure that the .csv files is with .R file in the same directory.

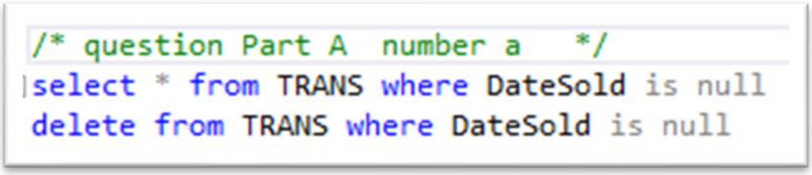


```
rm(list=ls())

CUSTOMER <- read.
PRODUCT <- read.c
PRODUCT_SALES <-
TIMELINE <- read.

Quan_cube <-
  tapply(PRODUCT_
         PRODUCT_                    roductNumber")],
         FUN=func

Quan_cube
dimnames(Quan_cube)
```



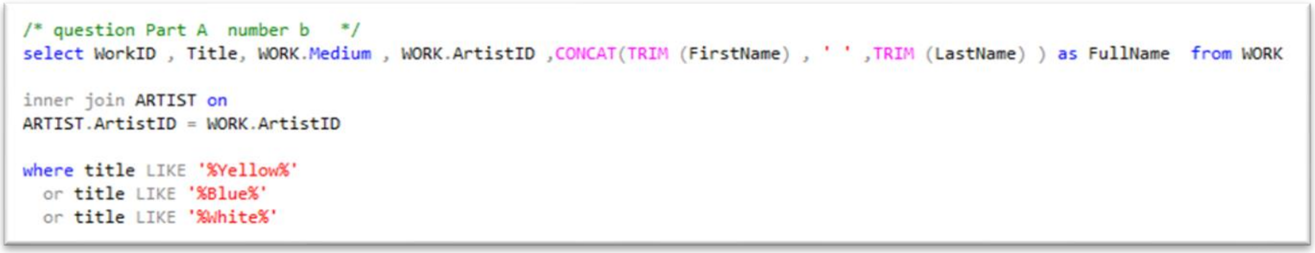
- **Part A: RDBMS & SQL**

a) Select from TRANS where DateSoldID is not null to get rows with null
then delete from TRANS where DateSoldID is not null to delete these rows

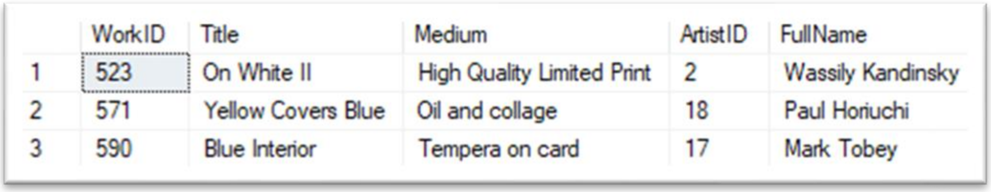


```
/* question Part A  number a   */
select * from TRANS where DateSold is null
delete from TRANS where DateSold is null
```

b) Use the CONCAT function to get FirstName and LastName and use where to get all rows that have Yellow Blue or White in the artwork title

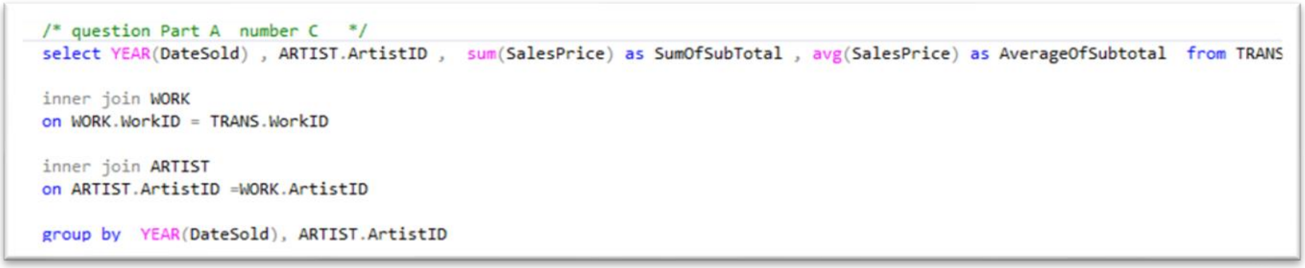


```
/* question Part A  number b   */
select WorkID , Title, WORK.Medium , WORK.ArtistID ,CONCAT(TRIM (FirstName) , ' ' ,TRIM (LastName) ) as FullName  from WORK

inner join ARTIST on
ARTIST.ArtistID = WORK.ArtistID

where title LIKE '%Yellow%'
  or title LIKE '%Blue%'
  or title LIKE '%White%'
```

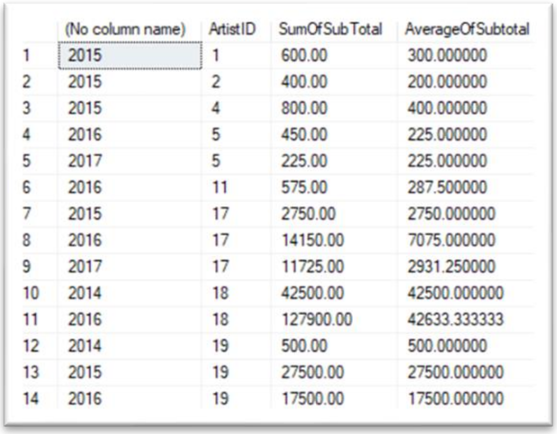


	WorkID	Title	Medium	ArtistID	FullName
1	523	On White II	High Quality Limited Print	2	Wassily Kandinsky
2	571	Yellow Covers Blue	Oil and collage	18	Paul Horiuchi
3	590	Blue Interior	Tempera on card	17	Mark Tobey

c) Uses group by `YEAR(DateSold), ARTIST.ArtistID to get the sum and average of the sales price For each artist in each year`

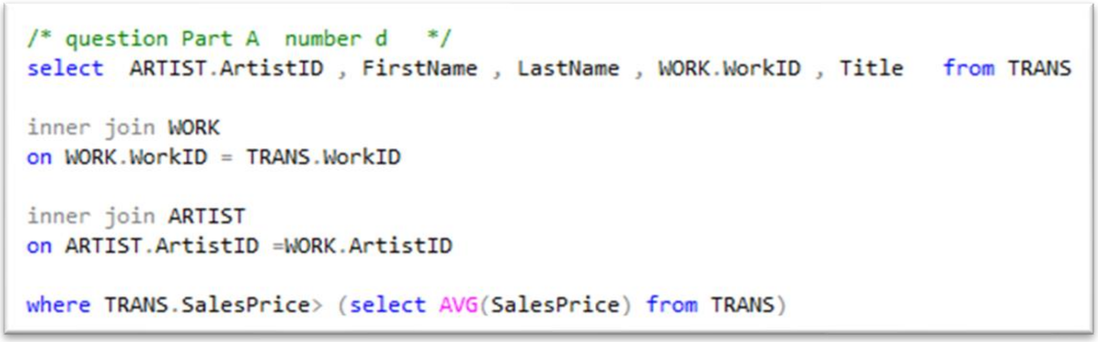


```
/* question Part A  number C   */
select YEAR(DateSold) , ARTIST.ArtistID ,  sum(SalesPrice) as SumOfSubTotal , avg(SalesPrice) as AverageOfSubtotal  from TRANS

inner join WORK
on WORK.WorkID = TRANS.WorkID

inner join ARTIST
on ARTIST.ArtistID =WORK.ArtistID

group by  YEAR(DateSold), ARTIST.ArtistID
```

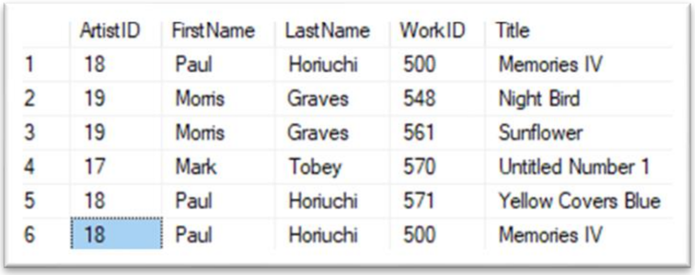


	(No column name)	ArtistID	SumOfSubTotal	AverageOfSubtotal
1	2015	1	600.00	300.000000
2	2015	2	400.00	200.000000
3	2015	4	800.00	400.000000
4	2016	5	450.00	225.000000
5	2017	5	225.00	225.000000
6	2016	11	575.00	287.500000
7	2015	17	2750.00	2750.000000
8	2016	17	14150.00	7075.000000
9	2017	17	11725.00	2931.250000
10	2014	18	42500.00	42500.000000
11	2016	18	127900.00	42633.333333
12	2014	19	500.00	500.000000
13	2015	19	27500.00	27500.000000
14	2016	19	17500.00	17500.000000

d) Calculate the average sales prices and compare the average and the sales prices for the artists and get the artist that has a sales price greater than the average

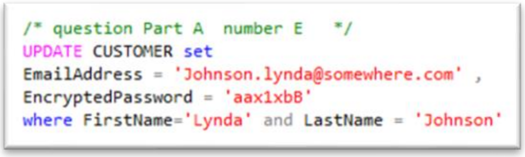


```
/* question Part A  number d   */
select  ARTIST.ArtistID , FirstName , LastName , WORK.WorkID , Title   from TRANS

inner join WORK
on WORK.WorkID = TRANS.WorkID

inner join ARTIST
on ARTIST.ArtistID =WORK.ArtistID

where TRANS.SalesPrice> (select AVG(SalesPrice) from TRANS)
```

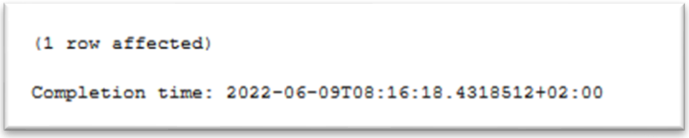


	ArtistID	FirstName	LastName	WorkID	Title
1	18	Paul	Horiuchi	500	Memories IV
2	19	Morris	Graves	548	Night Bird
3	19	Morris	Graves	561	Sunflower
4	17	Mark	Tobey	570	Untitled Number 1
5	18	Paul	Horiuchi	571	Yellow Covers Blue
6	18	Paul	Horiuchi	500	Memories IV

e) Use Update statement to update `EmailAddress and EncryptedPassword for specific Customer Lynda`

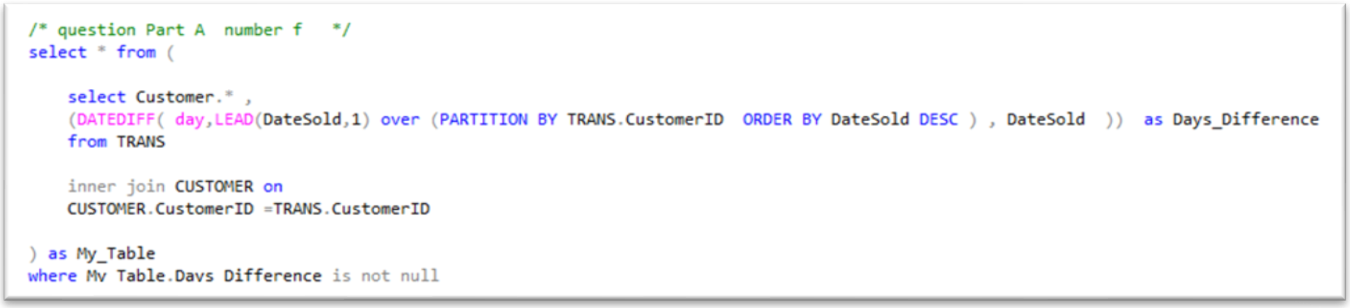


```
/* question Part A  number E   */
UPDATE CUSTOMER set
EmailAddress = 'Johnson.lynda@somewhere.com' ,
EncryptedPassword = 'aax1xbB'
where FirstName='Lynda' and LastName = 'Johnson'
```

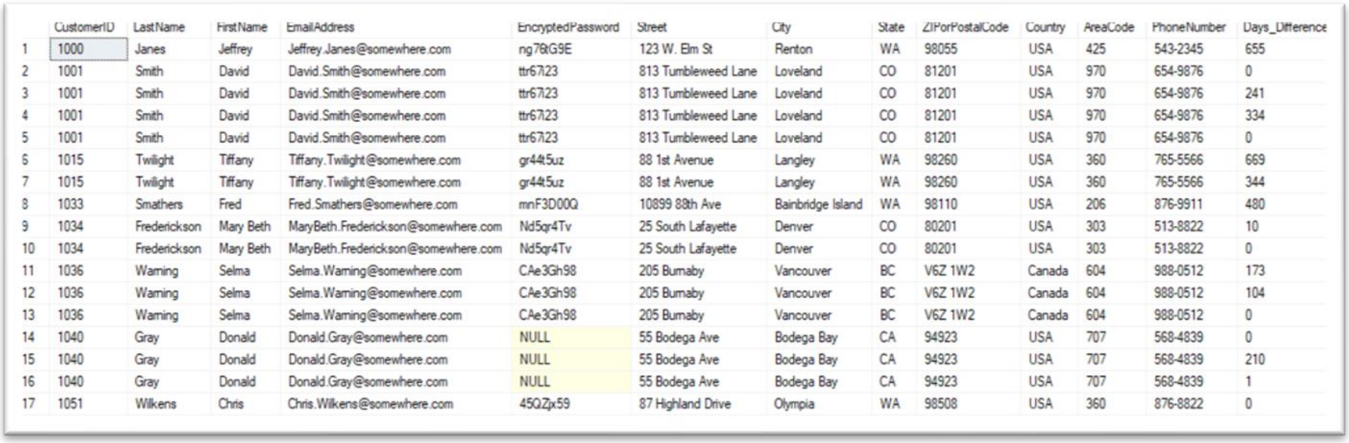


```
(1 row affected)

Completion time: 2022-06-09T08:16:18.4318512+02:00
```

f) Use the `lead` function to get the next purchase then use `DATEDIFF` to calculate days between two purchases and drop the rows that have null in the Days_Difference column

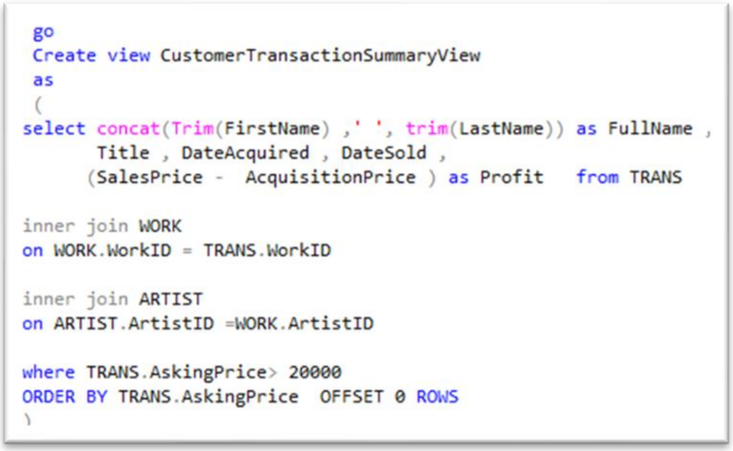


```
/* question Part A  number f   */
select * from (

    select Customer.* ,
    (DATEDIFF( day,LEAD(DateSold,1) over (PARTITION BY TRANS.CustomerID  ORDER BY DateSold DESC ) , DateSold  ))  as Days_Difference
    from TRANS

    inner join CUSTOMER on
    CUSTOMER.CustomerID =TRANS.CustomerID

) as My_Table
where My_Table.Days_Difference is not null
```

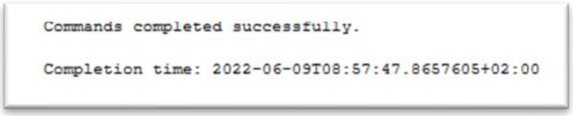


	CustomerID	LastName	FirstName	EmailAddress	EncryptedPassword	Street	City	State	ZIPorPostalCode	Country	AreaCode	PhoneNumber	Days_Difference
1	1000	Janes	Jeffrey	Jeffrey.Janes@somewhere.com	ng76tG9E	123 W. Elm St	Renton	WA	98055	USA	425	543-2345	655
2	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	0
3	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	241
4	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	334
5	1001	Smith	David	David.Smith@somewhere.com	ttr67i23	813 Tumbleweed Lane	Loveland	CO	81201	USA	970	654-9876	0
6	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	669
7	1015	Twilight	Tiffany	Tiffany.Twilight@somewhere.com	gr44t5uz	88 1st Avenue	Langley	WA	98260	USA	360	765-5566	344
8	1033	Smathers	Fred	Fred.Smathers@somewhere.com	mnF3D00Q	10899 88th Ave	Bainbridge Island	WA	98110	USA	206	876-9911	480
9	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	10
10	1034	Frederickson	Mary Beth	MaryBeth.Frederickson@somewhere.com	Nd5qr4Tv	25 South Lafayette	Denver	CO	80201	USA	303	513-8822	0
11	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0512	173
12	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0512	104
13	1036	Warning	Selma	Selma.Warning@somewhere.com	CAe3Gh98	205 Burnaby	Vancouver	BC	V6Z 1W2	Canada	604	988-0512	0
14	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	0
15	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	210
16	1040	Gray	Donald	Donald.Gray@somewhere.com	NULL	55 Bodega Ave	Bodega Bay	CA	94923	USA	707	568-4839	1
17	1051	Wilkens	Chris	Chris.Wilkens@somewhere.com	45QZjx59	87 Highland Drive	Olympia	WA	98508	USA	360	876-8822	0

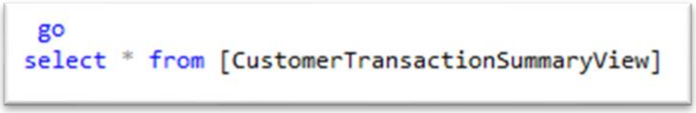
g) Create View and select the data from the view



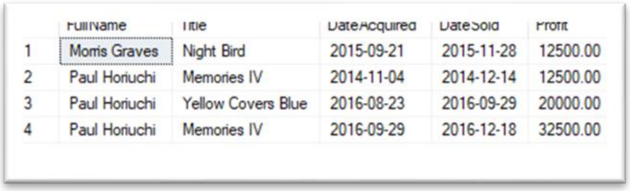
```
 go
 Create view CustomerTransactionSummaryView
 as
 (
select concat(Trim(FirstName) ,' ', trim(LastName)) as FullName ,
       Title , DateAcquired , DateSold ,
      (SalesPrice -  AcquisitionPrice ) as Profit   from TRANS

inner join WORK
on WORK.WorkID = TRANS.WorkID

inner join ARTIST
on ARTIST.ArtistID =WORK.ArtistID

where TRANS.AskingPrice> 20000
ORDER BY TRANS.AskingPrice  OFFSET 0 ROWS
)
```



```
Commands completed successfully.

Completion time: 2022-06-09T08:57:47.8657605+02:00
```

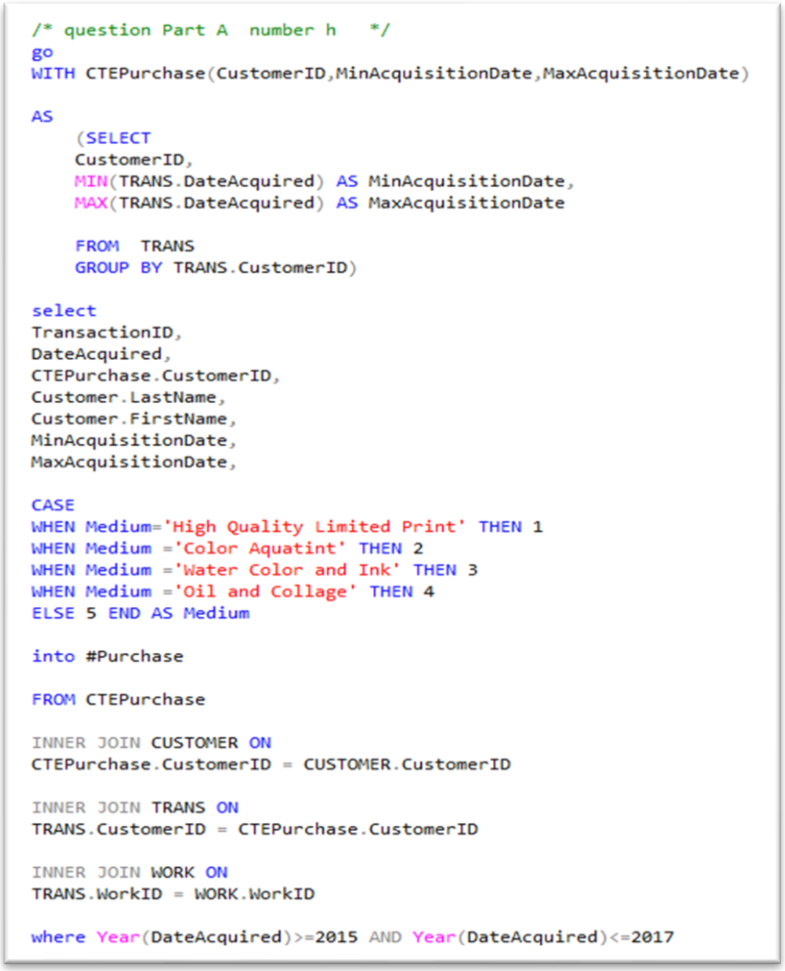


```
 go
select * from [CustomerTransactionSummaryView]
```



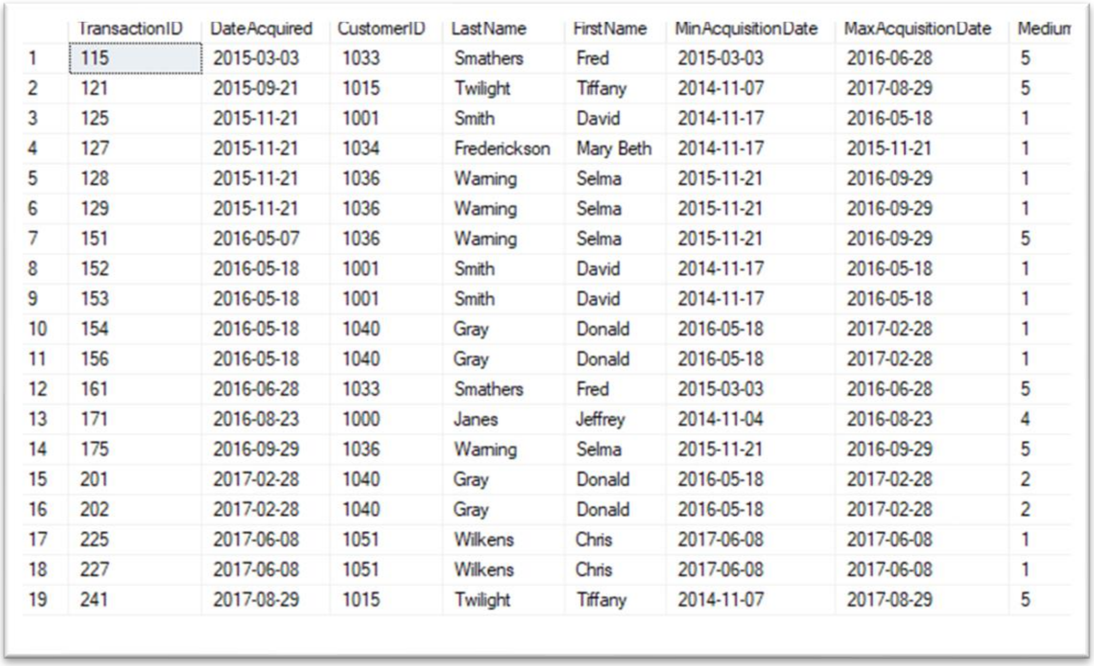
	FullName	Title	DateAcquired	DateSold	Profit
1	Morris Graves	Night Bird	2015-09-21	2015-11-28	12500.00
2	Paul Horiuchi	Memories IV	2014-11-04	2014-12-14	12500.00
3	Paul Horiuchi	Yellow Covers Blue	2016-08-23	2016-09-29	20000.00
4	Paul Horiuchi	Memories IV	2016-09-29	2016-12-18	32500.00

h) Use with statement to get min and max of the purchase date for each customer and combine the result with the rest of the requested rows then use into #TableName to create and insert into the temporary table



```
/* question Part A  number h   */
go
WITH CTEPurchase(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,
CTEPurchase.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 CTEPurchase

INNER JOIN CUSTOMER ON
CTEPurchase.CustomerID = CUSTOMER.CustomerID

INNER JOIN TRANS ON
TRANS.CustomerID = CTEPurchase.CustomerID

INNER JOIN WORK ON
TRANS.WorkID = WORK.WorkID

where Year(DateAcquired)>=2015 AND Year(DateAcquired)<=2017
```

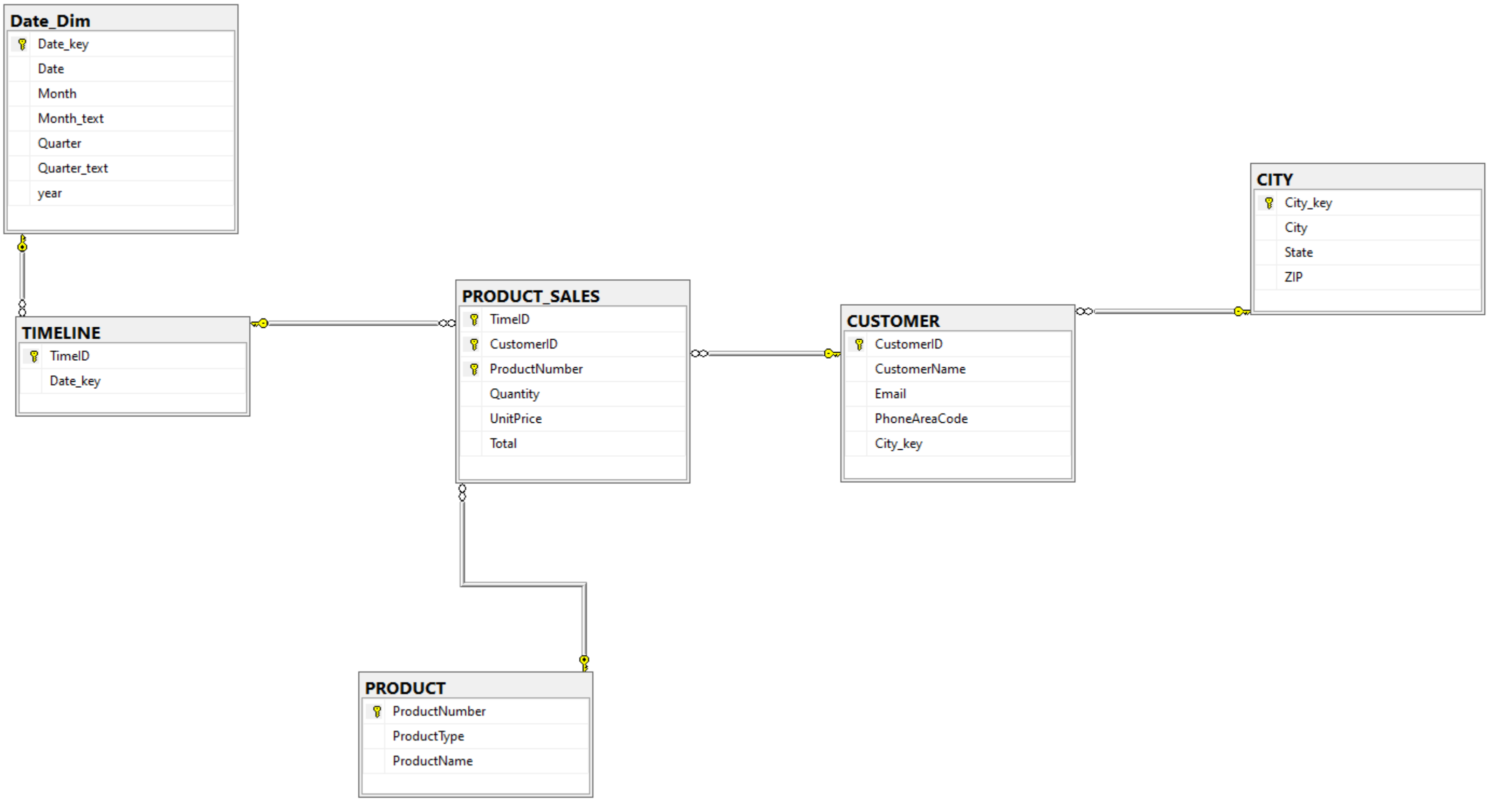
And when selecting data from the temporary table it is show as the follow

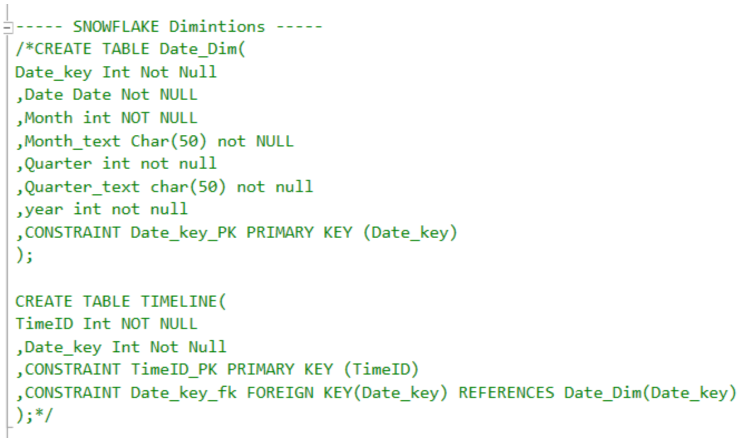


	TransactionID	DateAcquired	CustomerID	LastName	FirstName	MinAcquisitionDate	MaxAcquisitionDate	Mediun
1	115	2015-03-03	1033	Smathers	Fred	2015-03-03	2016-06-28	5
2	121	2015-09-21	1015	Twilight	Tiffany	2014-11-07	2017-08-29	5
3	125	2015-11-21	1001	Smith	David	2014-11-17	2016-05-18	1
4	127	2015-11-21	1034	Frederickson	Mary Beth	2014-11-17	2015-11-21	1
5	128	2015-11-21	1036	Warning	Selma	2015-11-21	2016-09-29	1
6	129	2015-11-21	1036	Warning	Selma	2015-11-21	2016-09-29	1
7	151	2016-05-07	1036	Warning	Selma	2015-11-21	2016-09-29	5
8	152	2016-05-18	1001	Smith	David	2014-11-17	2016-05-18	1
9	153	2016-05-18	1001	Smith	David	2014-11-17	2016-05-18	1
10	154	2016-05-18	1040	Gray	Donald	2016-05-18	2017-02-28	1
11	156	2016-05-18	1040	Gray	Donald	2016-05-18	2017-02-28	1
12	161	2016-06-28	1033	Smathers	Fred	2015-03-03	2016-06-28	5
13	171	2016-08-23	1000	Janes	Jeffrey	2014-11-04	2016-08-23	4
14	175	2016-09-29	1036	Warning	Selma	2015-11-21	2016-09-29	5
15	201	2017-02-28	1040	Gray	Donald	2016-05-18	2017-02-28	2
16	202	2017-02-28	1040	Gray	Donald	2016-05-18	2017-02-28	2
17	225	2017-06-08	1051	Wilkens	Chris	2017-06-08	2017-06-08	1
18	227	2017-06-08	1051	Wilkens	Chris	2017-06-08	2017-06-08	1
19	241	2017-08-29	1015	Twilight	Tiffany	2014-11-07	2017-08-29	5

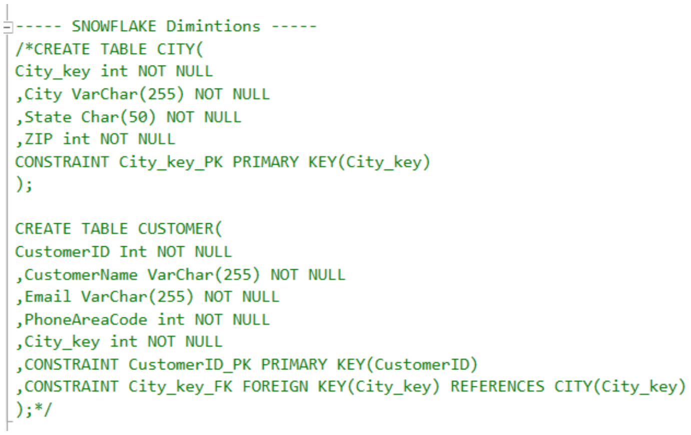
- **Part B: Data Warehousing & OLAP**

1- snowflake schema for the data warehouse



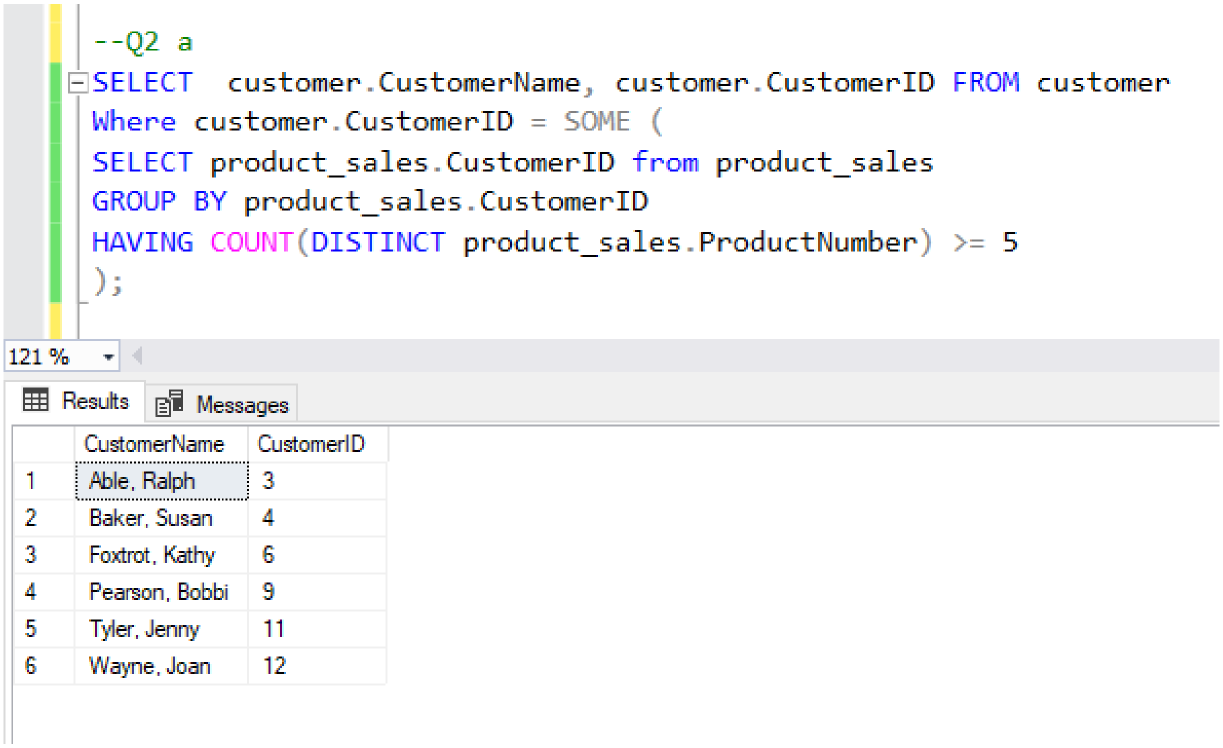


```
----- SNOWFLAKE Dimintions -----
/*CREATE TABLE Date_Dim(
Date_key Int Not Null
,Date Date Not NULL
,Month int NOT NULL
,Month_text Char(50) not NULL
,Quarter int not null
,Quarter_text char(50) not null
,year int not null
,CONSTRAINT Date_key_PK PRIMARY KEY (Date_key)
);

CREATE TABLE TIMELINE(
TimeID Int NOT NULL
,Date_key Int Not Null
,CONSTRAINT TimeID_PK PRIMARY KEY (TimeID)
,CONSTRAINT Date_key_fk FOREIGN KEY(Date_key) REFERENCES Date_Dim(Date_key)
);*/
```



```
----- SNOWFLAKE Dimintions -----
/*CREATE TABLE CITY(
City_key int NOT NULL
,City VarChar(255) NOT NULL
,State Char(50) NOT NULL
,ZIP int NOT NULL
CONSTRAINT City_key_PK PRIMARY KEY(City_key)
);

CREATE TABLE CUSTOMER(
CustomerID Int NOT NULL
,CustomerName VarChar(255) NOT NULL
,Email VarChar(255) NOT NULL
,PhoneAreaCode int NOT NULL
,City_key int NOT NULL
,CONSTRAINT CustomerID_PK PRIMARY KEY(CustomerID)
,CONSTRAINT City_key_FK FOREIGN KEY(City_key) REFERENCES CITY(City_key)
);*/
```

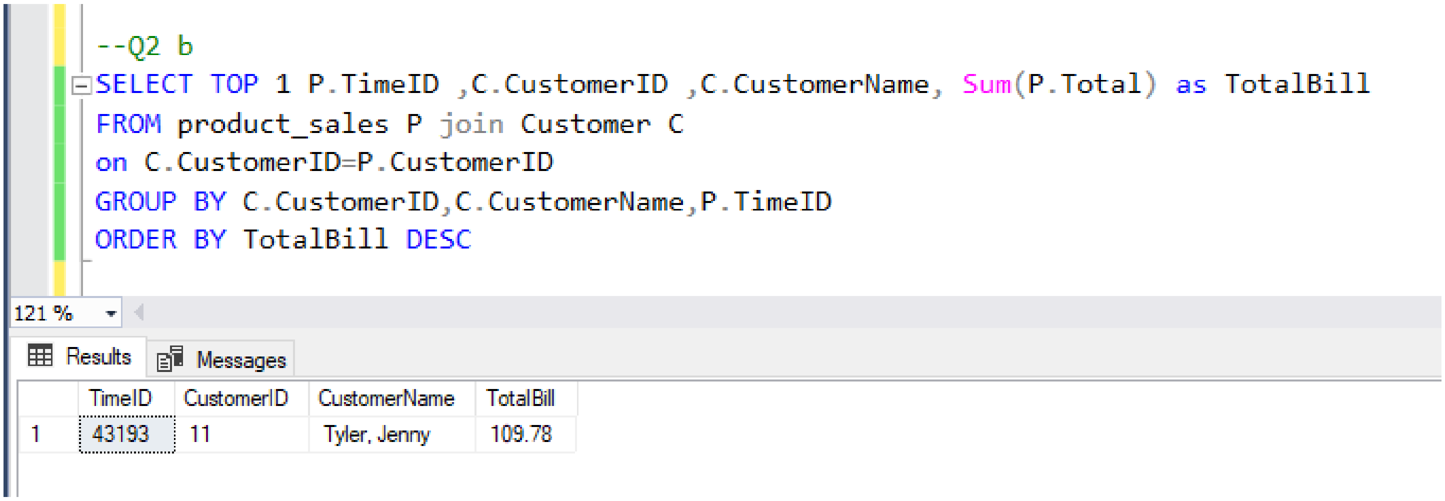
2-(a): Customers that at least made five products with different product numbers.



```
--Q2 a
SELECT  customer.CustomerName, customer.CustomerID FROM customer
Where customer.CustomerID = SOME (
SELECT product_sales.CustomerID from product_sales
GROUP BY product_sales.CustomerID
HAVING COUNT(DISTINCT product_sales.ProductNumber) >= 5
);
```

	CustomerName	CustomerID
1	Able, Ralph	3
2	Baker, Susan	4
3	Foxtrot, Kathy	6
4	Pearson, Bobbi	9
5	Tyler, Jenny	11
6	Wayne, Joan	12

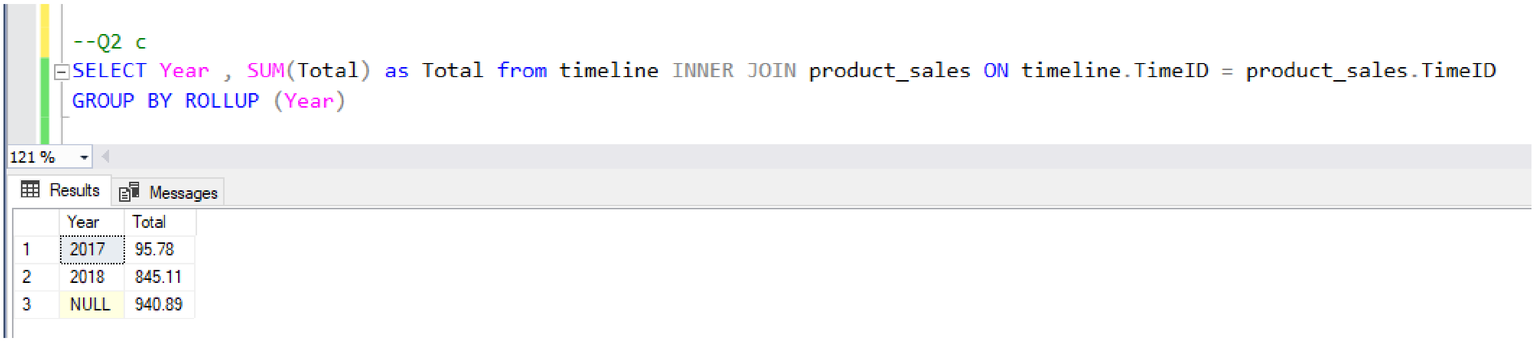
2-(b): customers that made the largest orders.



```
--Q2 b
SELECT TOP 1 P.TimeID ,C.CustomerID ,C.CustomerName, Sum(P.Total) as TotalBill
FROM product_sales P join Customer C
on C.CustomerID=P.CustomerID
GROUP BY C.CustomerID,C.CustomerName,P.TimeID
ORDER BY TotalBill DESC
```

	TimeID	CustomerID	CustomerName	TotalBill
1	43193	11	Tyler, Jenny	109.78

2-(c): Roll-Up operation to summaries the total sales per Year.



```
--Q2 c
SELECT Year , SUM(Total) as Total from timeline INNER JOIN product_sales ON timeline.TimeID = product_sales.TimeID
GROUP BY ROLLUP (Year)
```

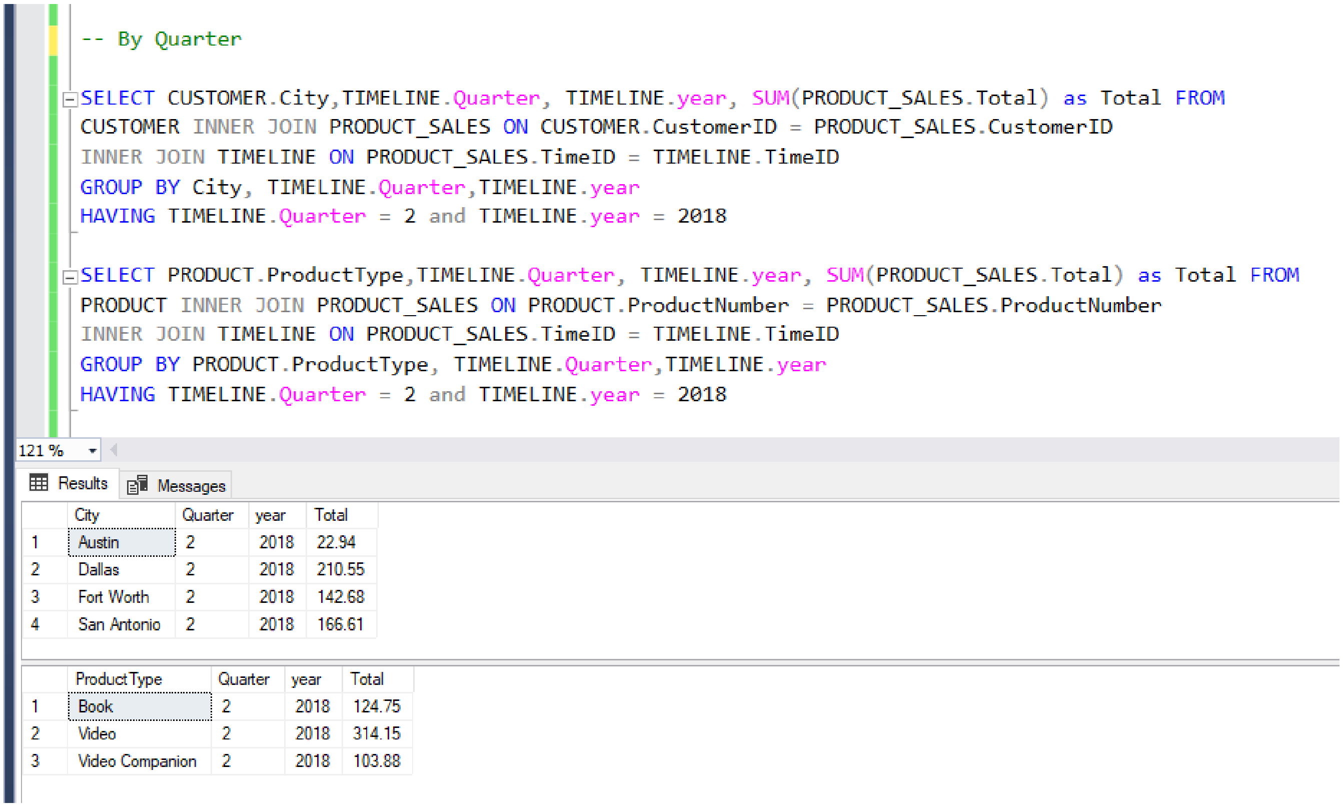
	Year	Total
1	2017	95.78
2	2018	845.11
3	NULL	940.89

3-(a): customers that made the largest orders.

Here we aggregated the next for columns together (**City** and **Quarter** and **Year** and **Total**) because we want to know which (city or product type) is responsible for decreasing.

And we observed that the city **Austin** has the minimum value and that would indiacted that this city has a relation with the decrease that happened.

And the same for product type…



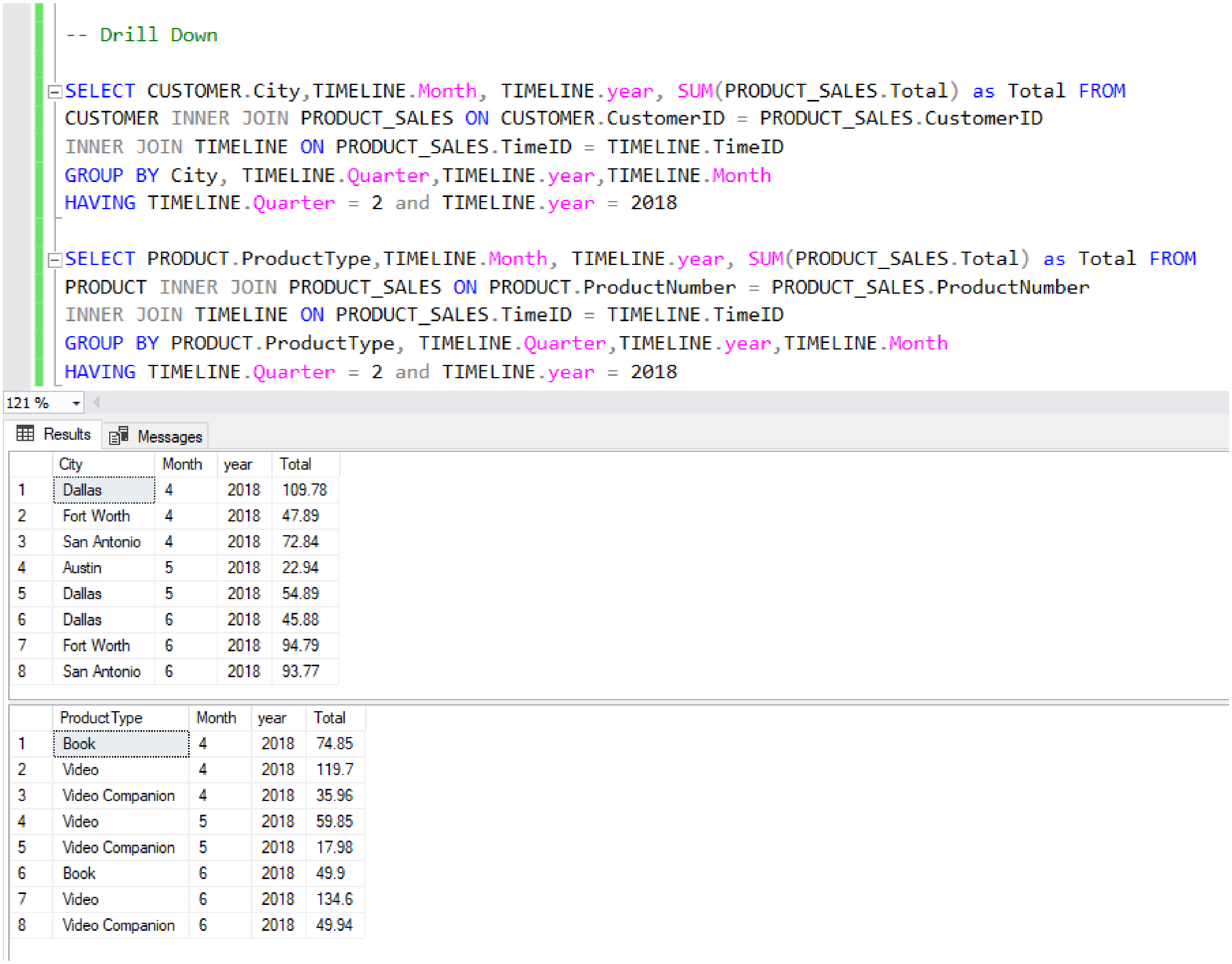
```
-- By Quarter

SELECT CUSTOMER.City,TIMELINE.Quarter, 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,TIMELINE.year
HAVING TIMELINE.Quarter = 2 and TIMELINE.year = 2018

SELECT PRODUCT.ProductType,TIMELINE.Quarter, 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,TIMELINE.year
HAVING TIMELINE.Quarter = 2 and TIMELINE.year = 2018
```

	City	Quarter	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	year	Total
1	Book	2	2018	124.75
2	Video	2	2018	314.15
3	Video Companion	2	2018	103.88

And as we will see in **3-(b)** when we apply (Drill Down Operation) we will know when (city or product type) has decreased in each month from **April 2018 to June 2018.**

3-(b): here we used Drill Down Operation to stepping down a concept hierarchy for a dimension (change quarter column to months column).



```
-- Drill Down

SELECT CUSTOMER.City,TIMELINE.Month, 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,TIMELINE.year,TIMELINE.Month
HAVING TIMELINE.Quarter = 2 and TIMELINE.year = 2018

SELECT PRODUCT.ProductType,TIMELINE.Month, 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,TIMELINE.year,TIMELINE.Month
HAVING TIMELINE.Quarter = 2 and TIMELINE.year = 2018
```

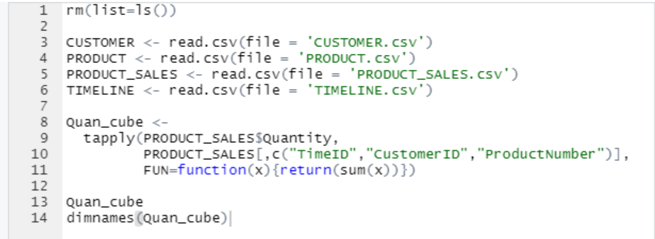
	City	Month	year	Total
1	Dallas	4	2018	109.78
2	Fort Worth	4	2018	47.89
3	San Antonio	4	2018	72.84
4	Austin	5	2018	22.94
5	Dallas	5	2018	54.89
6	Dallas	6	2018	45.88
7	Fort Worth	6	2018	94.79
8	San Antonio	6	2018	93.77

	ProductType	Month	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

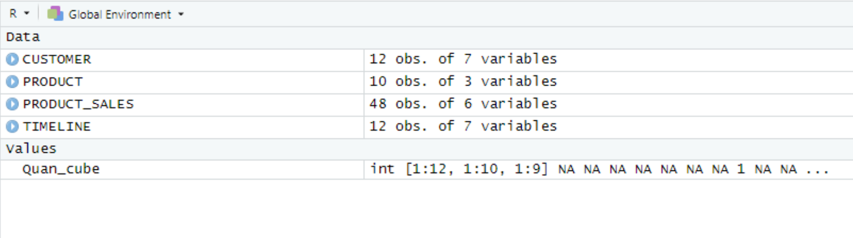
In the above figure we have observed that the value of city **Dallas** has decreased from April (was 109.78) to (54.89 in month 5) to (45.88 in month 6), so it's decreasing constantly and of course it have an effect in some way on the overall decreasing.

And the same happened in the product type table…

4- Build an OLAP cube for the Sum of Total Quantity.

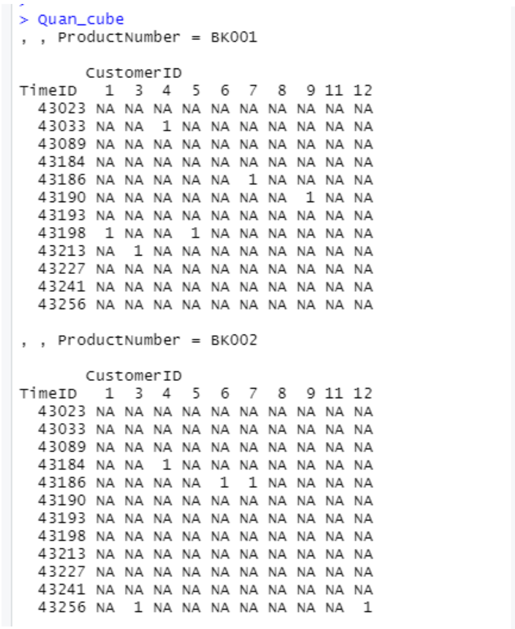


```
rm(list=ls())

CUSTOMER <- read.csv(file = 'CUSTOMER.csv')
PRODUCT <- read.csv(file = 'PRODUCT.csv')
PRODUCT_SALES <- read.csv(file = 'PRODUCT_SALES.csv')
TIMELINE <- read.csv(file = 'TIMELINE.csv')

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

Quan_cube
dimnames(Quan_cube)
```

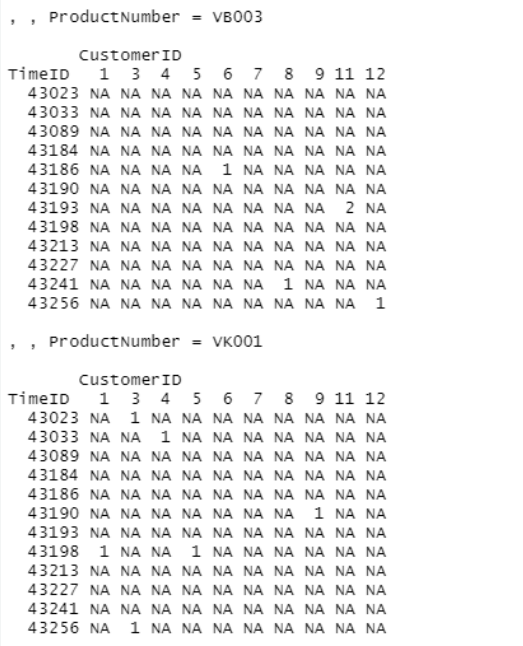


```
Data
CUSTOMER        12 obs. of 7 variables
PRODUCT         10 obs. of 3 variables
PRODUCT_SALES   48 obs. of 6 variables
TIMELINE        12 obs. of 7 variables
Values
  Quan_cube     int [1:12, 1:10, 1:9] NA NA NA NA NA NA NA NA 1 NA NA ...
```

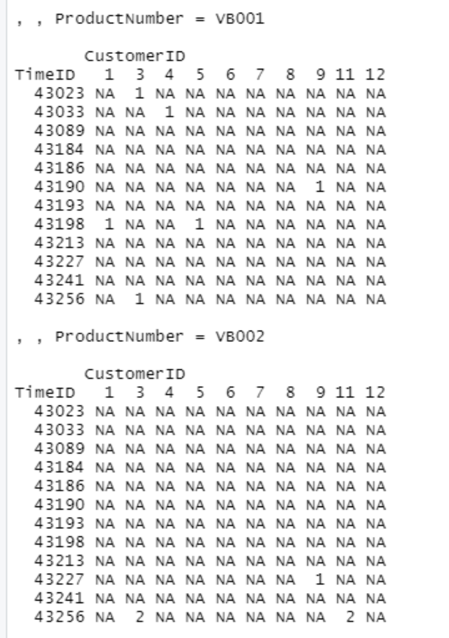
Cube layers:



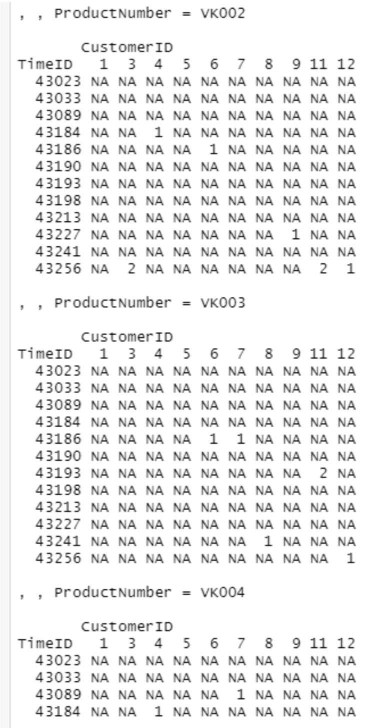
```
> Quan_cube
, , ProductNumber = BK001

       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  1 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  1 NA NA
  43193 NA NA NA NA NA NA NA NA NA NA
  43198  1 NA NA  1 NA NA NA NA NA NA
  43213 NA  1 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 = BK002

       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  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  1 NA NA NA NA NA NA NA  1
```



```
, , ProductNumber = VB003

       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  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  2 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  1 NA NA NA
  43256 NA NA NA NA NA NA NA NA NA  1

, , ProductNumber = VK001

       CustomerID
TimeID   1  3  4  5  6  7  8  9 11 12
  43023 NA  1 NA NA NA NA NA NA NA NA
  43033 NA NA  1 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 NA NA NA NA NA
  43190 NA NA NA NA NA NA NA  1 NA NA
  43193 NA NA NA NA NA NA NA NA NA NA
  43198  1 NA NA  1 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  1 NA NA NA NA NA NA NA NA
```

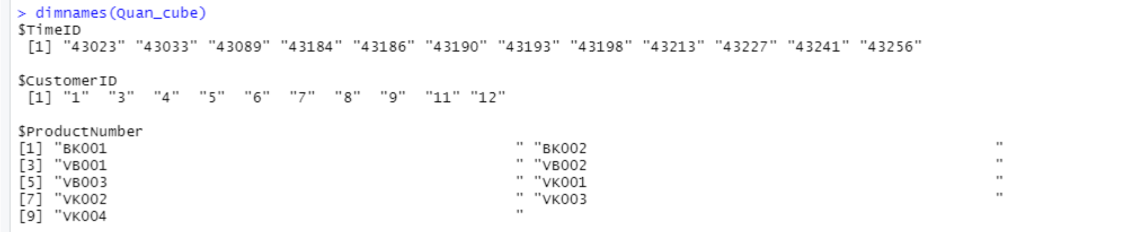


```
, , ProductNumber = VB001

       CustomerID
TimeID   1  3  4  5  6  7  8  9 11 12
  43023 NA  1 NA NA NA NA NA NA NA NA
  43033 NA NA  1 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 NA NA NA NA NA
  43190 NA NA NA NA NA NA NA  1 NA NA
  43193 NA NA NA NA NA NA NA NA NA NA
  43198  1 NA NA  1 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  1 NA NA NA NA NA NA NA NA

, , ProductNumber = VB002

       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 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 NA  2 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 NA  2  1

, , 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  1  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  2 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  1 NA NA NA
  43256 NA NA NA NA NA NA NA NA NA  1

, , ProductNumber = VK004

       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  1 NA NA NA NA
  43184 NA NA  1 NA NA NA NA NA NA NA
```

Dimentions



```
> dimnames(Quan_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                                        "
[3] "VB001                                        " "VB002                                        "
[5] "VB003                                        " "VK001                                        "
[7] "VK002                                        " "VK003                                        "
[9] "VK004                                        "
```