DTI5126[EG] Fundamentals/Applied Data Science

Assignment SQL

Students:

1-Kareem Atif Mohamed Bakli **2-** Mostafa Mahmoud Nofal

Part A: RDBMS & SQL

a-Using delete to remove values of nulls

```
/*First Question*/
Delete from TRANS where DateSold is NULL;

/*For Revice*/
Select * from trans where DateSold is NULL;
```

Results Messages

	TransactionID	DateAcquired	AcquisitionPrice	AskingPrice	DateSold	SalesPrice	CustomerID	WorkID

b-Using concat to concatenate first name and last name and trim to remove space, then make join between work table and artist table

```
/*Second Question*/
select WorkId,Title,Medium,arts.ArtistID ,CONCAT (Trim(FirstName)+' ',Trim(LastName)) as FullName
from work wrk join artist arts on  wrk.ArtistID = arts.ArtistID
where Title like '%Yellow%' or Title like '%Blue%' or Title like '%White%'
```

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-Make join between work table, trans and artist then group by Dateofbirth and ArtistID

```
/*Third Question */
Select Year(DateSold) as Year,wrk.ArtistID,sum(salesprice)  as SumOFSubTotal
,avg(salesprice) as AverageOfSubtotal
from WORK wrk
join trans trns on wrk.WorkID = trns.WorkID
Group by Year(DateSold),wrk.ArtistID
```

Results Messages

	Year	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-Make join between work table, trans and artist by ArtistID, WorkID then group by Dateofbirth and ArtistID

```
/*Fourth Question*/
Select arts.ArtistID,FirstName,LastName,wrk.WorkID,Title
from WORK wrk
join artist arts on arts.ArtistID = wrk.ArtistID
join trans trns  on trns.WorkID = wrk.WorkID
where trns.SalesPrice >( select avg(salesprice) from trans)
```

Results Messages

	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-Update all values of null where FirstName is Lynda and Johnson as LastName by update

```
/*Fifth Question */
update customer
set EmailAddress='Johnson.lynda@somewhere.com' , EncryptedPassword='aax1xbB'
where EmailAddress is Null and EncryptedPassword is Null

/*For Revice*/
select EmailAddress,EncryptedPassword from customer
where FirstName ='Lynda' and LastName = 'Johnson';
```

	EmailAddress	EncryptedPassword
1	Johnson.lynda@somewhere.com	aax1xbB

f-Display all the attributes of the customer and days between purchase as Days_Difference. using the LAG() function, from the current row, you can access data of the previous row, or the row before the previous row, and so on.

```
/*six Question */
With DateDifff(CustomerID, DateSold, NextDateSold) AS(
SELECT trn.CustomerID,DateSold, LAG(DateSold)
OVER(PARTITION BY trn.CustomerID ORDER BY DateSold) as NextDateSold
from TRANS trn inner JOIN CUSTOMER cust on cust.CustomerID = trn.CustomerID)
SELECT DateDifff.CustomerID,  DATEDIFF(DAY, NextDateSold, DateSold) AS DateDiff
FROM DateDifff where DateDifff.NextDateSold IS NOT NULL
```

Results Messages

	CustomerID	DateDiff
1	1000	655
2	1001	0
3	1001	334
4	1001	241
5	1001	0
6	1015	344
7	1015	669
8	1033	480
9	1034	0
10	1034	10
11	1036	0
12	1036	104
13	1036	173
14	1040	1
15	1040	210
16	1040	0
17	1051	0

g-Creating view display the concatenated customer name renamed as FullName Title, DateAcquired, DateSold, and difference in the AcquisitionPrice and SalesPrice as Profit with condition AskingPrice greater than $20,000

```
CREATE VIEW CustomerTransactionSummary As
Select CONCAT(Trim(LastName)+' ',Trim(FirstName)) as FullName,
Title,DateAcquired,DateSold,(SalesPrice-AcquisitionPrice) as Profit
from TRANS trn
JOIN WORK wrk  ON wrk.WorkID =trn.WorkID
JOIN CUSTOMER cust ON cust.CustomerID=trn.CustomerID
where trn.AskingPrice>20000
Order by trn.AskingPrice desc OFFSET 0 rows
```

Results Messages

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

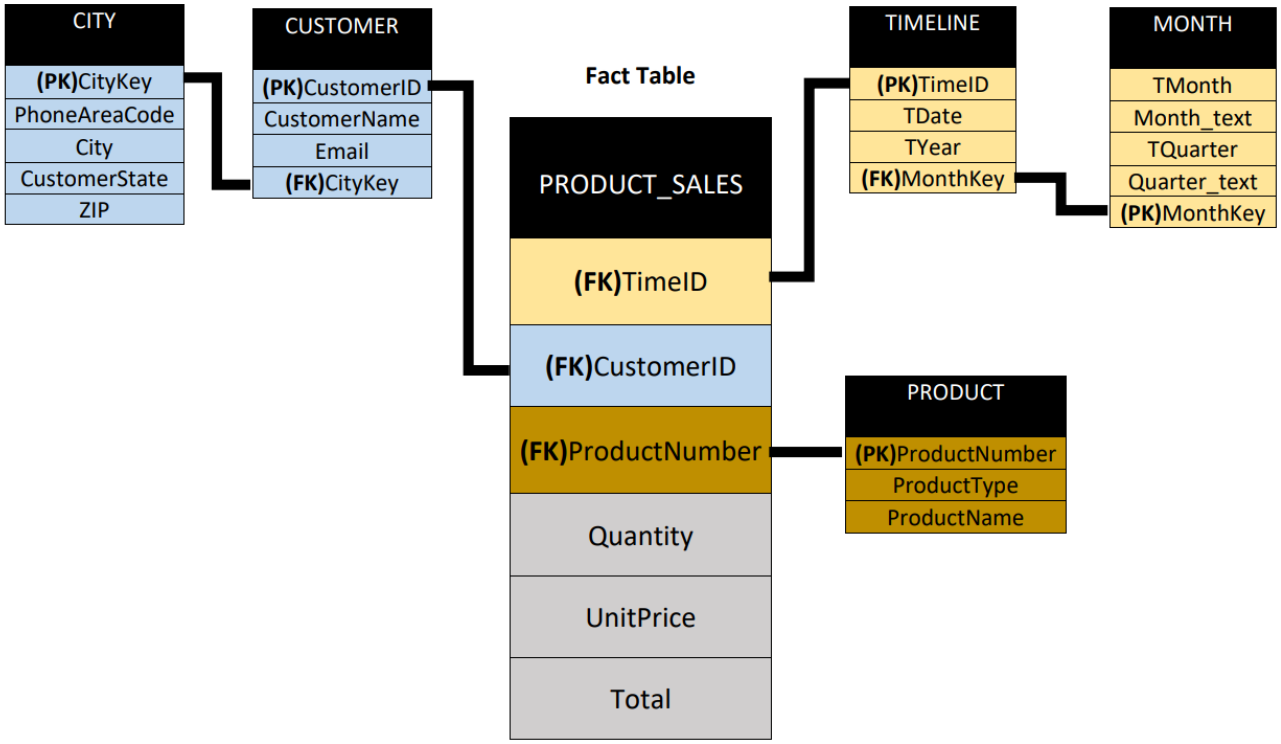
h- Build a single temporary table

```
/*Eight Question*/
With Purchase(CustomerID, MinAcquisitionDate, MaxAcquisitionDate) AS(
SELECT CustomerID, MIN(DateAcquired) as MinAcquisitionDate, MAX(DateAcquired) as MaxAcquisitionDate
FROM Trans
GROUP BY CustomerID),Purchase_var(TransactionID, DateAcquired, CustomerID, LastName, FirstName,
MinAcquisitionDate,
MaxAcquisitionDate, Medium) AS(SELECT DISTINCT trans.TransactionID, trans.DateAcquired,
trans.CustomerID,cst.LastName,cst.FirstName,
max_min.MinAcquisitionDate, max_min.MaxAcquisitionDate,
CASE
When Medium = 'High Quality Limited Print' THEN 1
When Medium = 'Color lithograph' THEN 2
When Medium = 'Watercolor and ink' THEN 3
When Medium = 'Oil and Collage' THEN 4
ELSE 5
end
FROM TRANS trans
JOIN CUSTOMER cst on trans.CustomerID = cst.CustomerID
JOIN Purchase max_min on trans.CustomerID = max_min.customerID
JOIN WORK wr on trans.WorkID = wr.WorkID
Where Year(trans.DateAcquired) BETWEEN 2015 and 2017)
SELECT* FROM Purchase_var
```

	TransactionID	DateAcquired	CustomerID	LastName	FirstName	MinAcquisitionDate	MaxAcquisitionDate	Medium
1	115	2015-03-03	1033	Smathers	Fred	2015-03-03	2016-06-28	2
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	3
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	5
16	202	2017-02-28	1040	Gray	Donald	2016-05-18	2017-02-28	5
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



2- (a) Customers who made an order containing at least five products with different product numbers

```
select C.CustomerName,C.CustomerID,COUNT(DISTINCT ProductNumber) as Quantity_ProductNumber
from PRODUCT_SALES S join CUSTOMER C
on s.CustomerID=c.CustomerID
group by C.CustomerID,C.CustomerName
having COUNT(DISTINCT productnumber )>=5
```

Results | Messages

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

(b) Customers who made the largest order those that would result in the largest bill

```
with LargestBill as(
select CustomerID,TimeID,sum(Total)as Total_bill
FROM PRODUCT_SALES
group by TimeID,CustomerID)

select c.* from LargestBill,CUSTOMER as c where LargestBill.CustomerID=c.CustomerID and LargestBill.Total_bill=
(select max(Total_bill) from LargestBill)
```

Results | Messages

	CustomerID	CustomerName	Email	PhoneAreaCode	City	CustomerState	ZIP
1	11	Tyler, Jenny	somewhere.com	972	Dallas	TX	75225

(c) SQL queries for the ”Roll-Up” operation to summarize the total sales per Year

```
Select TL.TYear ,sum(Total)
from PRODUCT_SALES as S,TIMELINE as TL
where S.TimeID=TL.TimeID
group by Rollup(TL.TYear)
```

Results | Messages

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

3- (a) If we want to check the reason for the decrease in monthly total, we will do the following queries:

- **We can determine by two ways. First, we join the Sales, Product, Timeline tables to get the ProductType and Month columns together.**
- **Then we sum the total column in respect to months and and ProductType.**
- **Then we limit the year to 2018 and the months from April to June.**
- **See the total of each product**

```
--First method
-- From april to june 2018
Select  P.ProductType, T.TMonth,sum(PS.Total) as Total
from TIMELINE as T ,PRODUCT_SALES as PS,PRODUCT as P
where T.TimeID=PS.TimeID  and T.TYear =2018 and T.TMonth in (4,5,6)and p.ProductNumber=PS.ProductNumber
Group by P.ProductType, T.TMonth order by P.ProductType
```

Results | Messages

	ProductType	TMonth	Total
1	Book	4	74.85
2	Book	6	49.90
3	Video	4	119.70
4	Video	5	59.85
5	Video	6	134.60
6	Video Companion	4	35.96
7	Video Companion	5	17.98
8	Video Companion	6	49.94

- Second method to aggregate all the months and compare the period between April and June with the other periods along the whole year to check whether its decreasing or not.

```
Select  P.ProductType, T.TYear,T.TMonth,sum(PS.Total) as Total
from TIMELINE as T ,PRODUCT_SALES as PS,PRODUCT as P
where T.TimeID=PS.TimeID  and p.ProductNumber=PS.ProductNumber
Group by T.TYear,P.ProductType, T.TMonth order by P.ProductType
```

Results | Messages

	ProductType	TYear	TMonth	Total
1	Book	2017	10	24.95
2	Book	2018	3	124.75
3	Book	2018	4	74.85
4	Book	2018	6	49.90
5	Video	2017	10	29.90
6	Video	2017	12	24.95
7	Video	2018	3	159.60
8	Video	2018	4	119.70
9	Video	2018	5	59.85
10	Video	2018	6	134.60
11	Video Companion	2017	10	15.98
12	Video Companion	2018	3	17.98
13	Video Companion	2018	4	35.96
14	Video Companion	2018	5	17.98
15	Video Companion	2018	6	49.94

- (b) The relevant “drill-down” to visualize the month beside the year to indicate more information

4- (a) Read the dimensions tables

```
# Setup the dimension tables

#create the Timeline table

Timeline_table <- data.frame(
            TimeID=c(43023, 43033, 43089, 43184, 43186, 43190, 43193, 43198, 43213, 43227, 43241, 43256),
            TDate=c("15-Oct-17", "25-Oct-17", "20-Dec-17", "25-Mar-18", "27-Mar-18", "31-Mar-18", "3-Apr-18", "8-Apr-18", "23-Apr-18", "7-May-18", "21-May-18", "5-Jun-18"),
            TMonth=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"),
            TQuarter=c(3, 3, 3, 1, 1, 1, 2, 2, 2 ,2, 2, 2),
            Quarter_text=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))

#create the Customer table

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,
            Email=c("somewhere.com", "somewhere.com", "somewhere.com", "elsewhere.com", "elsewhere.com", "somewhere.com", "somewhere.com", "elsewhere.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))

#create the Product table

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 Remo
```

- Timeline Table

```
> print(Timeline_table)
   TimeID     TDate TMonth Month_text TQuarter Quarter_text TYear
1   43023 15-Oct-17     10    October        3         Qtr3  2017
2   43033 25-Oct-17     10    October        3         Qtr3  2017
3   43089 20-Dec-17     12   December        3         Qtr3  2017
4   43184 25-Mar-18      3      March        1         Qtr1  2018
5   43186 27-Mar-18      3      March        1         Qtr1  2018
6   43190 31-Mar-18      3      March        1         Qtr1  2018
7   43193  3-Apr-18      4      April        2         Qtr2  2018
8   43198  8-Apr-18      4      April        2         Qtr2  2018
9   43213 23-Apr-18      4      April        2         Qtr2  2018
10  43227  7-May-18      5        May        2         Qtr2  2018
11  43241 21-May-18      5        May        2         Qtr2  2018
12  43256  5-Jun-18      6       June        2         Qtr2  2018
```

- Customer Table

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

- Product Table

```
> print(PRODUCT_table)
   ProductNumber     ProductType                                    ProductName
1          BK001            Book          Kitchen Remodeling Basics For Everyone
2          BK002            Book       Advanced Kitchen Remodeling For Everyone
3          BK003            Book   Kitchen Remodeling Dallas Style For Everyone
4          VB001 Video Companion                     Kitchen Remodeling Basics
5          VB002 Video Companion                  Advanced Kitchen Remodeling I
6          VB003 Video Companion                Kitchen Remodeling Dallas Style
7          VK001           Video                     Kitchen Remodeling Basics
8          VK002           Video                  Advanced Kitchen Remodeling
9          VK003           Video                Kitchen Remodeling Dallas Style
10         VK004           Video Heather Sweeney Seminar Live in Dallas on 25-OCT-16
```

- Read the Sales fact table

```
# Setup the fact table

Sales_table <- data.frame(
          TimeID=c(43023, 43023, 43033, 43033, 43033, 43089, 43184, 43184, 43184, 43186, 43186, 43186, 43186, 43186, 43186, 43186, 43186, 43186, 43190, 43190, 43190, 4
          CustomerID=c(3 ,3, 4, 4, 4, 7, 4, 4, 4, 6, 6, 6, 6, 6, 7, 7, 7, 7, 9, 9, 9, 11, 11, 11, 1, 1, 1, 1, 5, 5, 5, 3, 9, 9, 8, 8, 8, 3, 3, 3, 3, 3, 11, 11, 12, 12, 12
          ProductNumber=c("VB001", "VK001", "BK001", "VB001", "VK001", "VK004", "BK002", "VK002", "VK004", "BK002", "VB003", "VK002", "VK003", "VK004", "BK001", "BK002
          Quantity=c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 1, 1),
          UnitPrice=c(7.99, 14.95, 24.95, 7.99, 14.95, 24.95, 24.95, 14.95, 24.95, 24.95, 9.99, 14.95, 19.95, 24.95, 24.95, 24.95, 19.95, 24.95, 24.95, 7.99, 14.95, 9.
          Total=c(7.99, 14.95, 24.95, 7.99, 14.95, 24.95, 24.95, 14.95, 24.95, 24.95, 9.99, 14.95, 19.95, 24.95, 24.95, 24.95, 19.95, 24.95, 24.95, 7.99, 14.95, 19.98,
```

- Sales Table

```
> print(Sales_table)
   TimeID CustomerID ProductNumber Quantity UnitPrice Total
1   43023          3          VB001        1      7.99  7.99
2   43023          3          VK001        1     14.95 14.95
3   43033          4          BK001        1     24.95 24.95
4   43033          4          VB001        1      7.99  7.99
5   43033          4          VK001        1     14.95 14.95
6   43089          7          VK004        1     24.95 24.95
7   43184          4          BK002        1     24.95 24.95
8   43184          4          VK002        1     14.95 14.95
9   43184          4          VK004        1     24.95 24.95
10  43186          6          BK002        1     24.95 24.95
11  43186          6          VB003        1      9.99  9.99
12  43186          6          VK002        1     14.95 14.95
13  43186          6          VK003        1     19.95 19.95
14  43186          6          VK004        1     24.95 24.95
15  43186          7          BK001        1     24.95 24.95
16  43186          7          BK002        1     24.95 24.95
17  43186          7          Vk003        1     19.95 19.95
18  43186          7          VK004        1     24.95 24.95
19  43190          9          BK001        1     24.95 24.95
20  43190          9          VB001        1      7.99  7.99
21  43190          9          VK001        1     14.95 14.95
```

(b) Read the cube

```
# Build up a cube
QuantityC <-
  tapply(Sales_table$Quantity,
         Sales_table[,c("TimeID", "CustomerID", "ProductNumber")],
         FUN=function(x){return(sum(x))})

#print the cube
print(QuantityC)

#print the dimensions names of the cube
dimnames(QuantityC)
```

- The first two outputs of the cube

```
> #print the cube
> print(QuantityC)
, , 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
```

```
       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 = VB001
```

- Dimensions of the cube

```
> #print the dimensions names of the cube
> print(dimnames(QuantityC))
$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"
```