A STUDY ON OLAP OPERATIONS IN R

[1] .J. Uma Mahesh [2] SK.Islam Babu [3]G.Shashank[4] S.Chandrakanth

[1]ASST.PROFESSOR, Dept of CSE, BIET, uma.mahesh@biet.ac.in

[2]STUDENT, Dept of CSE, BIET, islam.shaik2k2@gmail.com

[3]STUDENT, Dept of CSE, BIET, gshashank.913@gmail.com

[4]STUDENT, Dept of CSE, BIET, chandrakanthsriramsetty@gmail.com

Abstract- In DWH (Data WGrehouse) OLAP (Online Analytical Processing) is a very common way to analyze sourcey or eggy transaction data by sumMar3izing along different combinations of dimensions. This is a well-accepted field in Business Intelligence / Business Reporting. The main part of OLAP is called "multi-dimensional data model", which consists two types of tables; "Fact" table and "Dimension" table. Fact table contains measures of a transaction and dimension table contains records which describes contextual attributes. This paper gives a study on OLAP operations in R Language (open source) Data Mining Tool and which highlight key ideas in OLAP operations and illustrates how to do this in R.

Key words- DWH (Data Warehouse), OLAP (Online Analytical Processing), R-Language, MDDM (Multi Dimensional Data Model)

1) INTRODUCTION

OLAP (or *Online Analytical Processing*) has been rapidly growing in popularity due to the increase in data volumes i.e. big data and the recognition of the business value of analytics. In the multidimensional model, data are arranged into multiple dimensions. And each dimension have multiple levels of abstraction defined by concept hierarchies.

2) Introduction to R

Useful features of R:
• Effective programming language
• Relational database support
• Data analytics
• Data visualization
• Expanded through the vast library of R packages
The R language is well defined, and typically used for statistics and predictive analytics. Even though, some organizations have been reluctant to use R in production applications because it is memory-bound. Data sets are now so large -- sometimes exceeding tens of gigabytes and hundreds of millions of rows -- that scalability and performance mature issues, particularly for mission-critical applications with precise deadlines

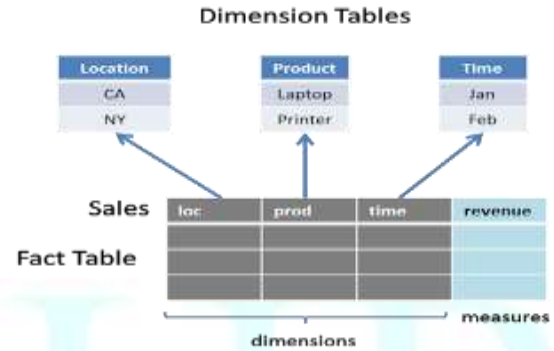
3) Multidimensional Data Model

The multidimensional data model is an essential part of On-Line Analytical Processing, or OLAP. as OLAP is on-line, it must provide answers promptly; analysts pose iterative queries during interactive session, not in batch jobs that run overnight. And because OLAP is also analytic, the query are complex. The multidimensional data model is intended to solve complex queries in real time.

In a difficult setting of Multi-dimensional model.
Each fact table contains foreign keys that mention the priMar3y key of multiple dimension tables. In the simplest form, it is called a STAR schema. Dimension tables can contain foreign keys that

reference other dimensional tables. This provides a involving detail breakdown of the dependent aspects. This is also called a SNOWFLAKE schema. Also this is not a hard rule, Fact table tends to be independent of other Fact table and usually doesn't contain reference pointer among each other. However, different Fact table usually share the same set of dimension tables. This is also called GALAXY schema. But it is a hard rule that Dimension table NEVER points / references Fact table

For example STAR schema is shown in following diagram.



Each dimension can also be hierarchical so that the analysis can be done at different degree of granularity.

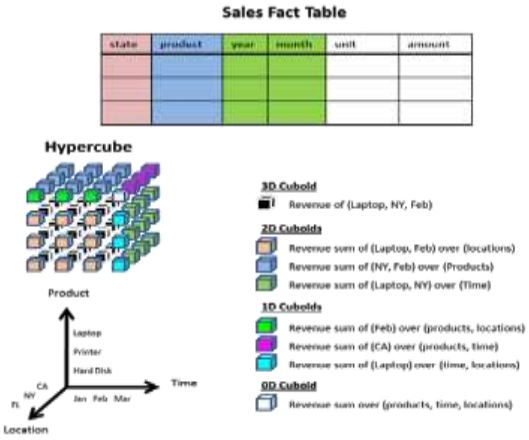
For example, the time dimension can be divided into days, weeks, months, quarter and annual; similarly, location dimension can be broken down into countries, states, cities and etc.

Here we will create a sales fact table that records each sales transaction.

```
# Setup the dimension tables
state_table <-
 data.frame(key=c("CA", "NZ", "WG", "ON", "QU"),
          name=c("Canada", "newzealand", "WestGodavari", "Ongole", "Quba"),
          country=c("USO", "USO", "USO", "Canada", "Canada"))
month_table <-
 data.frame(key=1:12,
         desc=c("Jan1", "Feb2", "Mar3", "Apr4", "May5", "Jun6", "Jul7", "Aug8", "Sep9", "Oct10", "Nov11", "Dec12"),

quarter=c("Q1","Q1","Q1","Q2","Q2","Q2","Q3","Q3","Q3","Q4","Q4","Q4"))
prod_table <-
 data.frame(key=c("Printer", "Tablet", "Laptop"),
         price=c(225, 570, 1120))

# Function to generate the Sales table
gen_sales <- function(no_of_recs) {
 # Generate transaction data randomly
 loc <- sample(state_table$key, no_of_recs,
           replace=T, prob=c(2,2,1,1,1))
 time_month <- sample(month_table$key, no_of_recs, replace=T)
 time_year <- sample(c(2012, 2013), no_of_recs, replace=T)
 prod <- sample(prod_table$key, no_of_recs, replace=T, prob=c(1, 3, 2))
 unit <- sample(c(1,2), no_of_recs, replace=T, prob=c(10, 3))
 amount <- unit*prod_table[prod,]$price
 sales <- data.frame(month=time_month,
                year=time_year,
                loc=loc,
                prod=prod,
                unit=unit,
                amount=amount)
 # Sort the records by time order
 sales <- sales[order(sales$year, sales$month),]
 row.names(sales) <- NULL
 return(sales)
}
# Now create the sales fact table
sales_fact <- gen_sales(500)

# Look at a few records
head(sales_fact)
 month year loc  prod unit amount
1    1 2012  NZ Laptop    1    225
2    1 2012  CA Laptop    2    450
3    1 2012  ON Tablet    2   2240
4    1 2012  NZ Tablet    1   1120
5    1 2012  NZ Tablet    2   2240
6    1 2012  CA Laptop    1    225
```

3) Multi-dimensional Cube

A data cube is constructed from a subset of attributes in the database. Now, we turn this fact table into a hypercube with multiple dimensions. Each cell in the cube represents an aggregate value for a unique combination of each dimension.




```
# Build up a cube
revenue_cube <-
   tapply(sales_fact$amount,
         sales_fact[,c("prod", "month", "year", "loc")],
         FUN=function(x){return(sum(x))})
# Showing the cells of the cude
revenue_cube
 year = 2012, loc = CA
       month
prod        1    2    3    4    5    6    7    8    9   10   11
12
 Laptop  1350  225   900  675  675   NA  675 1350
NA 1575  900 1350
 Printer   NA 2280    NA   NA 1140  570  570  570
NA  570 1710   NA
 Tablet   2240  4480 12320 3360 2240 4480 3360
3360 5600 2240 2240 3360
, , year = 2013, loc = CA
       month
prod        1    2    3    4    5    6    7    8    9   10   11
12
 Laptop   225  225  450  675  225  900  900  450  675
225  675 1125
 Printer   NA 1140   NA 1140  570   NA   NA  570
NA 1140 1710 1710
 Tablet  3360 3360 1120 4480 2240 1120 7840 3360
3360 1120 5600 4480
, , year = 2012, loc = NZ
       month
prod        1    2    3    4    5    6    7    8    9   10   11
12
 Laptop   450  450   NA   NA  675  450  675   NA
225  225   NA  450
 Printer   NA 2280   NA 2850  570   NA   NA 1710
1140   NA  570   NA
 Tablet   3360 13440 2240 2240 2240 5600 5600
3360 4480 3360 4480 3360
, , year = 2013, loc = NZ
...dimnames(revenue_cube)
$prod
[1] "Laptop"  "Printer" "Tablet"
$month
 [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
"11" "12"
$year
[1] "2012" "2013"
$loc
[1] "CA" "NZ" "ON" "QU" "WG"
```

4) OLAP operations in R Language

Here are some common operations of OLAP

1. Slice
2. Dice
3. Rollup
4. Drilldown
5. Pivot

1. **"Slice"** is about fixing certain dimensions to analyze the remaining dimensions. For example, we can focus in the sales happening in "2012", "Jan1", or we can focus in the sales happening in "2012", "Jan1", "Tablet".

```
# Slice
# cube data in Jan1, 2012
revenue_cube[, "1", "2012",]

       loc
prod      CA   NZ   ON   QU   WG
 Laptop  1350  450   NA  225  225
 Printer   NA   NA   NA 1140   NA
 Tablet  2240 3360 5600 1120 2240

# cube data in Jan1, 2012
revenue_cube["Tablet", "1", "2012",]

  CA   NZ   ON   QU   WG
2240 3360 5600 1120 2240
```

2. **"Dice"** is about limited each dimension to a certain range of values, while keeping the number of dimensions the same in the resulting cube. For example, we can focus in sales happening in [Jan1/ Feb2/Mar3, Laptop/Tablet, CA/NZ].

```
revenue_cube[c("Tablet","Laptop"),
           c("1","2","3"),
           ,
           c("CA","NZ")]
, , year = 2012, loc = CA

       month
prod       1    2    3
 Tablet 2240 4480 12320
 Laptop 1350  225   900
```

```
, , year = 2013, loc = CA

      month
prod      1   2   3
 Tablet 3360 3360 1120
 Laptop  225  225  450

, , year = 2012, loc = NZ

      month
prod      1    2   3
 Tablet 3360 13440 2240
 Laptop  450   450  NA

, , year = 2013, loc = NZ

      month
prod      1   2   3
 Tablet 3360 4480 6720
 Laptop  450  NA  225
```

3. "Rollup" is about applying an aggregation function to collapse a number of dimensions. For example, we WGnt to focus in the annual revenue for each product and collapse the location dimension (ie: we don't care where we sold our product).

```
apply(revenue_cube, c("year", "prod"),
    FUN=function(x)                 {return(sum(x,
na.rm=TRUE))})
    prod
year   Laptop Printer Tablet
 2012  22275   31350 179200
 2013  25200   33060 166880
```

4. "Drilldown" is the reverse of "rollup" and applying an aggregation function to a finer level of granularity. For example, we WGnt to focus in the annual and monthly revenue for each product and collapse the location dimension (ie: we don't care where we sold our product).

```
apply(revenue_cube, c("year", "month", "prod"),
    FUN=function(x)                 {return(sum(x,
na.rm=TRUE))})
, , prod = Laptop

    month
year    1   2   3   4   5   6   7   8   9  10  11  12
 2012 2250 2475 1575 1575 2250 1800 1575 1800
900 2250 1350 2475
 2013 2250  900 1575 1575 2250 2475 2025 1800
2025 2250 3825 2250

, , prod = Printer

    month
year    1   2   3   4   5   6   7   8   9  10  11  12
 2012 1140 5700  570 3990 4560 2850 1140 2850
2850 1710 3420  570
 2013 1140 4560 3420 4560 2850 1140  570 3420
1140 3420 3990 2850

, , prod = Tablet

    month
year     1    2    3    4    5    6    7    8    9   10
11   12
 2012 14560 23520 17920 12320 10080 14560
13440 15680 25760 12320 11200  7840
 2013  8960 11200 10080  7840 14560 10080 29120
15680 15680  8960 12320 22400
```

5. "Pivot" is about analyzing the combination of a pair of selected dimensions. For example, we WGnt to analyze the revenue by year and month. Or we WGnt to analyze the revenue by product and location.

```
apply(revenue_cube, c("year", "month"),
    FUN=function(x)                 {return(sum(x,
na.rm=TRUE))})
    month
year     1    2    3    4    5    6    7    8    9   10
11   12
 2012 17950 31695 20065 17885 16890 19210
16155 20330 29510 16280 15970 10885
 2013 12350 16660 15075 13975 19660 13695
31715 20900 18845 14630 20135 27500
```

```
apply(revenue_cube, c("prod", "loc"),
    FUN=function(x)                 {return(sum(x,
na.rm=TRUE))})
       loc
prod       CA    NZ    ON    QU    WG
 Laptop  16425  9450  7650  7425  6525
 Printer 15390  19950  7980 10830 10260
 Tablet  90720 117600 45920 34720 57120
```

However, R Language is doing all the processing in RAM

6. Conclusion:

To execute OLAP operations we can use open source Data Mining Tools like R Language, Weka, Knime,

RapidMiner, Orange and etc, to know practical working of OLAP operations. Here this study implements OLAP operations in R

REFERENCES:

[1]Multidimensional model and OLAP operations Indian Journal of Applied Research Volume: 3 | Issue: 3 | Mar3ch 2013 | ISSN - 2249-555X
[2] Continuous new OLAP operations on Data Streams International Journal of Soft Computing and Engineering (IJSCE) ISSN: 22312307, Volume-4,
[3]http://support.sas.com/documentation/cdl/en/olapug/63148/HTML/default/viewer.htm#n0m7xpwgy0gqabn1rjtrocf4ky83.htm
[4]https://web.stanford.edu/dept/itss/docs/oracle/10g/olap.101/b10333/multimodel.htm