Lab 1. Data Warehousing

OBJECTIVE. In this lab you will create a toy data warehouse and learn how to use SQL aggregation functions for OLAP purposes.

Activity 1. Create a Data Warehouse

1. Using SQL Developer create a database with the following structure:

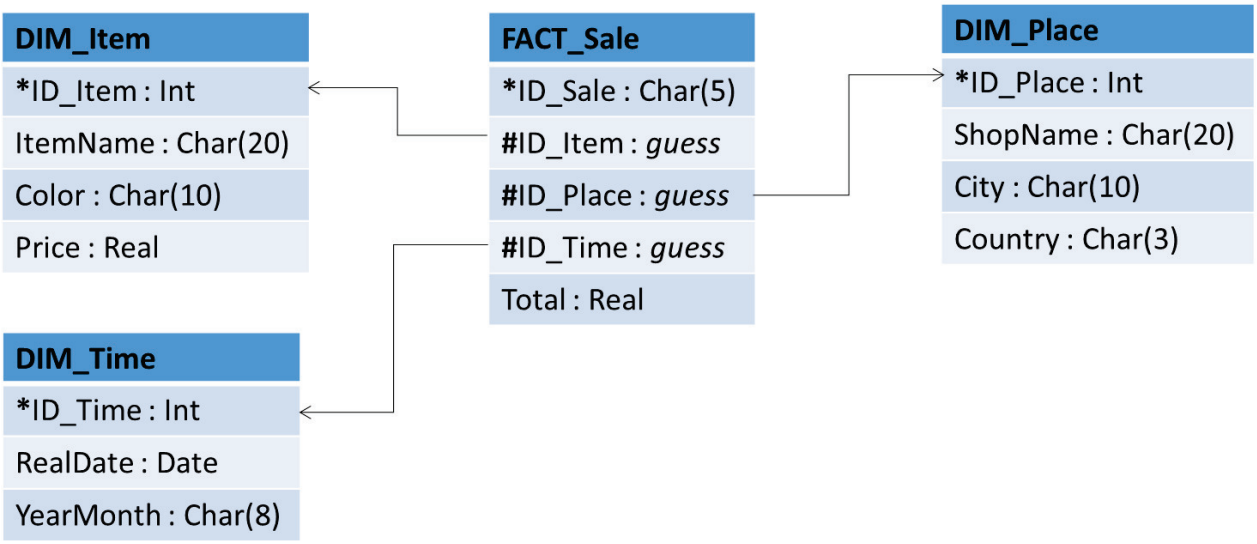

Fig. 1. Data Warehouse structure
(here sign * denotes primary key and sign # denotes foreign key)

2. Fill in the tables above with some real-like data (at least 5 records in each dimension table, at least 20 records in fact table). Fill the YearMonth field like '2014-FEB' for the respective value '01/02/2014' of the RealDate field.

ASK INSTRUCTOR to verify the results of your activity.

Activity 2. Aggregation with ROLLUP

ROLLUP enables a SELECT statement to calculate multiple levels of subtotals across a specified group of dimensions. It also calculates a grand total.

The action of ROLLUP is straightforward: it creates subtotals that roll up from the most detailed level to a grand total, following a grouping list specified in the ROLLUP clause. ROLLUP takes as its argument an ordered list of grouping columns. First, it calculates the standard aggregate values specified in the GROUP BY clause. Then, it creates progressively higher-level subtotals, moving from right to left through the list of grouping columns. Finally, it creates a grand total. ROLLUP creates subtotals at n+1 levels, where n is the number of grouping columns.

An example:

```
SELECT Dim1, Dim2, SUM(Measure)
FROM FactTab
GROUP BY ROLLUP (Dim1, Dim2)
```

```
ORDER BY Dim1, Dim2
```

1. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the FACT_Sale.ID_Item and FACT_Sale.ID_Place fields. An example of the required result:

```
ID_ITEM   ID_PLACE     TOTAL
      1          1     4363.55
      1          2     4794.76
      1          3     4718.25
      1          4     5387.45
      1          5     5027.34
      1               24291.35
      2          1     5652.84
      2          2     4583.02
      2          3     5555.77
      2          4     5936.67
      2          5     4508.74
      2               26237.04
                      50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

2. Modify the query above to calculate subtotals across the Dim_Item.ItemName and Dim_Place.ShopName fields. An example of the required result:

```
ITEMNAME    SHOPNAME    TOTAL
   Bolt     MainShop    4363.55
   Bolt     Details     4794.76
   Bolt     Repair      4718.25
   Bolt     Tools4U     5387.45
   Bolt     HomeMaster  5027.34
   Bolt                24291.35
   Screw     MainShop   5652.84
   Screw     Details    4583.02
   Screw     Repair     5555.77
   Screw     Tools4U    5936.67
   Screw     HomeMaster 4508.74
   Screw                26237.04
                        50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

3. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.ItemName, DIM_Place.City and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

4. Using ROLLUP keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.Color, DIM_Place.Country and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

Activity 3. Aggregation with CUBE

CUBE takes a specified set of grouping columns and creates subtotals for all of their possible combinations. In terms of multidimensional analysis, CUBE generates all the subtotals that could be calculated for a data cube with the specified dimensions. If n columns are specified for a CUBE, there will be 2 to the n combinations of subtotals returned.

An example:

```
SELECT Dim1, Dim2, SUM(Measure)
FROM FactTab
GROUP BY CUBE(Dim1, Dim2)
ORDER BY Dim1, Dim2
```

1. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the FACT_Sale.ID_Item and FACT_Sale.ID_Place fields. An example of the required result:

ID_ITEM	ID_PLACE	TOTAL
1	1	4363.55
1	2	4794.76
1	3	4718.25
1	4	5387.45
1	5	5027.34
1		24291.35
2	1	5652.84
2	2	4583.02
2	3	5555.77
2	4	5936.67
2	5	4508.74
2		26237.04
	1	10016.39
	2	9377.78
	3	10274.02
	4	11324.12
	5	9536.08
		50528.39

ASK INSTRUCTOR to verify the results of your activity.

2. Modify the query above to calculate subtotals across the Dim_Item.ItemName and Dim_Place.ShopName fields. An example of the required result:

ITEMNAME	SHOPNAME	TOTAL
Bolt	MainShop	4363.55
Bolt	Details	4794.76
Bolt	Repair	4718.25
Bolt	Tools4U	5387.45
Bolt	HomeMaster	5027.34
Bolt		24291.35
Screw	MainShop	5652.84
Screw	Details	4583.02
Screw	Repair	5555.77

```
Screw     Tools4U    5936.67
Screw     HomeMaster 4508.74
Screw                26237.04
          MainShop   10016.39
          Details     9377.78
          Repair     10274.02
          Tools4U    11324.12
          HomeMaster 9536.08
                     50528.39
```

ASK INSTRUCTOR to verify the results of your activity.

5. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.ItemName, DIM_Place.City and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.

6. Using CUBE keyword, construct a query that calculates subtotals of the FACT_Sale.Total field across the DIM_Item.Color, DIM_Place.Country and FACT_Time.YearMonth fields.

ASK INSTRUCTOR to verify the results of your activity.