**Assignment 1**

**Group: DWDM19G04**

**Roll Numbers: 2016BTECS00063, 2016BTECS00081, 2016BTECS00103**

**Batch: B7**

**Date: 24th August, 2019**

**Title: Data Warehousing**

**Problem Statement:**

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

**Theory:**

**Data Warehouse:**A large store of data accumulated from a wide range of sources within a company and used to guide management decisions.

**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

**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

**Output:**

Activity 1. Create a Data Warehouse

CREATE TABLE DIM\_ITEM

(

ID\_ITEM INT PRIMARY KEY,

ITEMNAME CHAR(20),

COLOR CHAR(10),

PRICE REAL

);

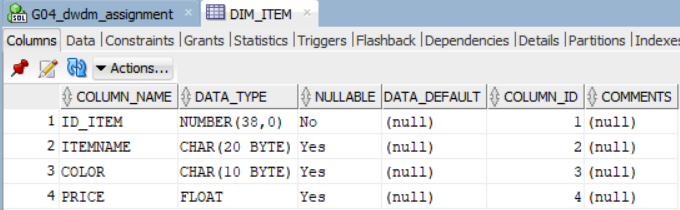
INSERT INTO DIM\_ITEM VALUES(1,'LAPTOP','RED',300);

INSERT INTO DIM\_ITEM VALUES(2,'BALL','BLUE',500);

INSERT INTO DIM\_ITEM VALUES(3,'MOBILE','ORANGE',200);

INSERT INTO DIM\_ITEM VALUES(4,'WALLET','GREY',150);

INSERT INTO DIM\_ITEM VALUES(5,'BAT','GREEN',400);



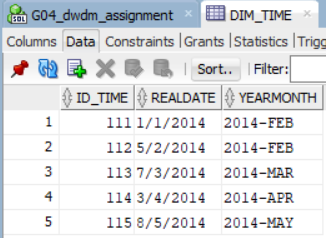
CREATE TABLE DIM\_TIME(

ID\_TIME INT PRIMARY KEY,

REALDATE CHAR(8),

YEARMONTH CHAR(8)

);



INSERT INTO DIM\_TIME VALUES(111,'1/1/2014','2014-FEB');

INSERT INTO DIM\_TIME VALUES(112,'5/2/2014','2014-FEB');

INSERT INTO DIM\_TIME VALUES(113,'7/3/2014','2014-MAR');

INSERT INTO DIM\_TIME VALUES(114,'3/4/2014','2014-APR');

INSERT INTO DIM\_TIME VALUES(115,'8/5/2014','2014-MAY');

CREATE TABLE DIM\_PLACE

(

ID\_PLACE INT PRIMARY KEY,

SHOPNAME CHAR(20),

CITY CHAR(10),

COUNTRY CHAR(3)

);

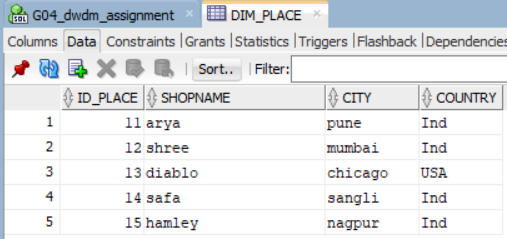
INSERT INTO DIM\_PLACE VALUES(11,'ARYA','PUNE','IND');

INSERT INTO DIM\_PLACE VALUES(12,'SHREE','MUMBAI','IND');

INSERT INTO DIM\_PLACE VALUES(13,'DIABLO','CHICAGO','USA');

INSERT INTO DIM\_PLACE VALUES(14,'SAFA','SANGLI','IND');

INSERT INTO DIM\_PLACE VALUES(15,'HAMLEY','NAGPUR','IND');



CREATE TABLE FACT\_SALE(

ID\_SALE CHAR(5) PRIMARY KEY,

ID\_ITEM INT,

ID\_PLACE INT,

ID\_TIME INT,

FOREIGN KEY (ID\_ITEM) REFERENCES DIM\_ITEM(ID\_ITEM),

FOREIGN KEY (ID\_PLACE) REFERENCES DIM\_PLACE(ID\_PLACE),

FOREIGN KEY (ID\_TIME) REFERENCES DIM\_TIME(ID\_TIME),

TOTAL REAL);

INSERT INTO FACT\_SALE VALUES('1121',1,15,111,5200);

INSERT INTO FACT\_SALE VALUES('1122',2,11,112,1500);

INSERT INTO FACT\_SALE VALUES('1123',3,14,113,5000);

INSERT INTO FACT\_SALE VALUES('1124',4,13,114,1300);

INSERT INTO FACT\_SALE VALUES('1125',5,12,115,2500);

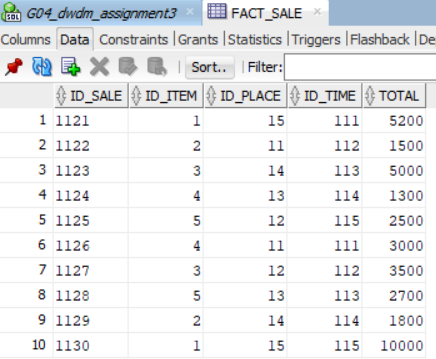
INSERT INTO FACT\_SALE VALUES('1126',4,11,111,3000);

INSERT INTO FACT\_SALE VALUES('1127',3,12,112,3500);

INSERT INTO FACT\_SALE VALUES('1128',5,13,113,2700);

INSERT INTO FACT\_SALE VALUES('1129',2,14,114,1800);

INSERT INTO FACT\_SALE VALUES('1130',1,15,115,10000);



Activity 2. Aggregation with ROLLUP

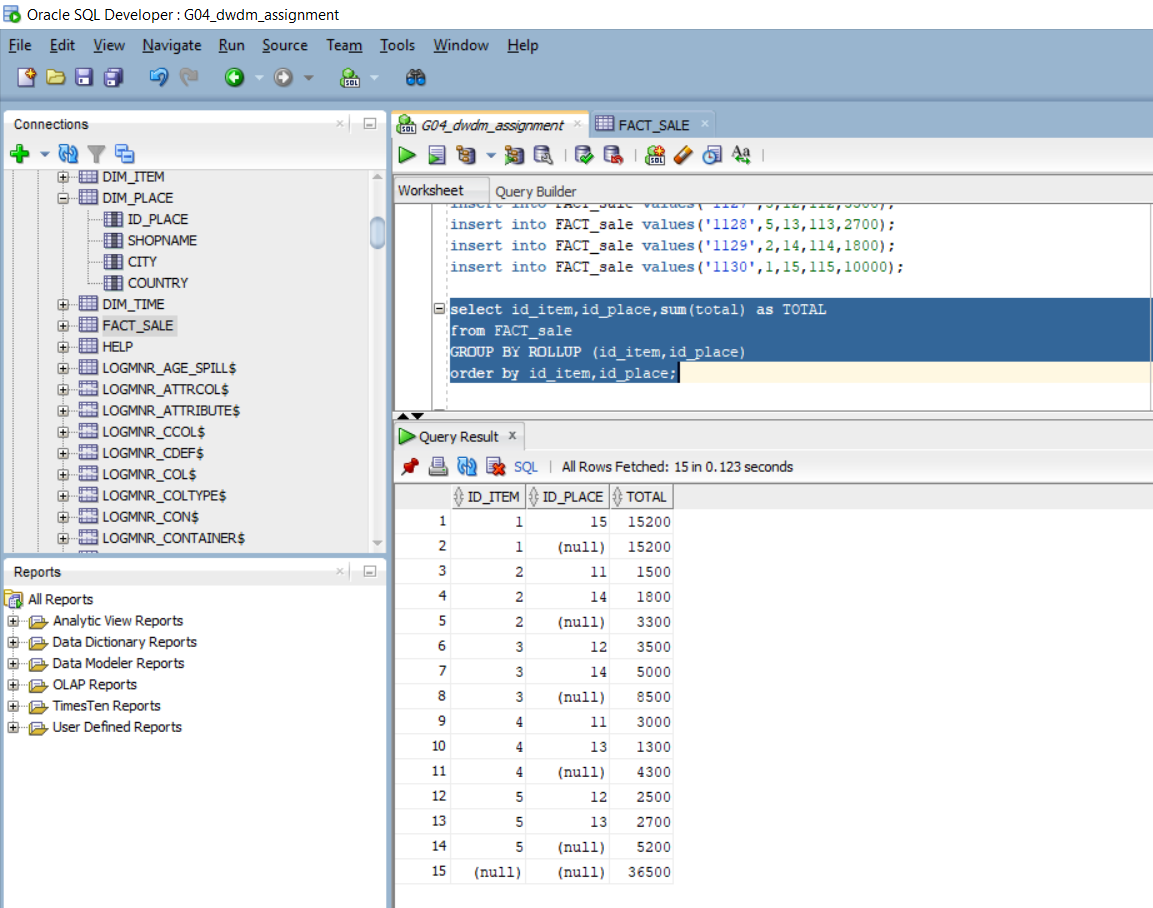
1) Using ROLLUP keyword, construct a query that calculates subtotals of the FACT\_Sale.Total field across the FACT\_Sale.ID\_Item and FACT\_Sale.ID\_Place fields:

SELECT ID\_ITEM,ID\_PLACE,SUM(TOTAL) AS TOTAL

FROM FACT\_SALE

GROUP BY ROLLUP (ID\_ITEM,ID\_PLACE)

ORDER BY ID\_ITEM,ID\_PLACE;



2) Modify the query above to calculate subtotals across the Dim\_Item.ItemName and Dim\_Place.ShopName fields.

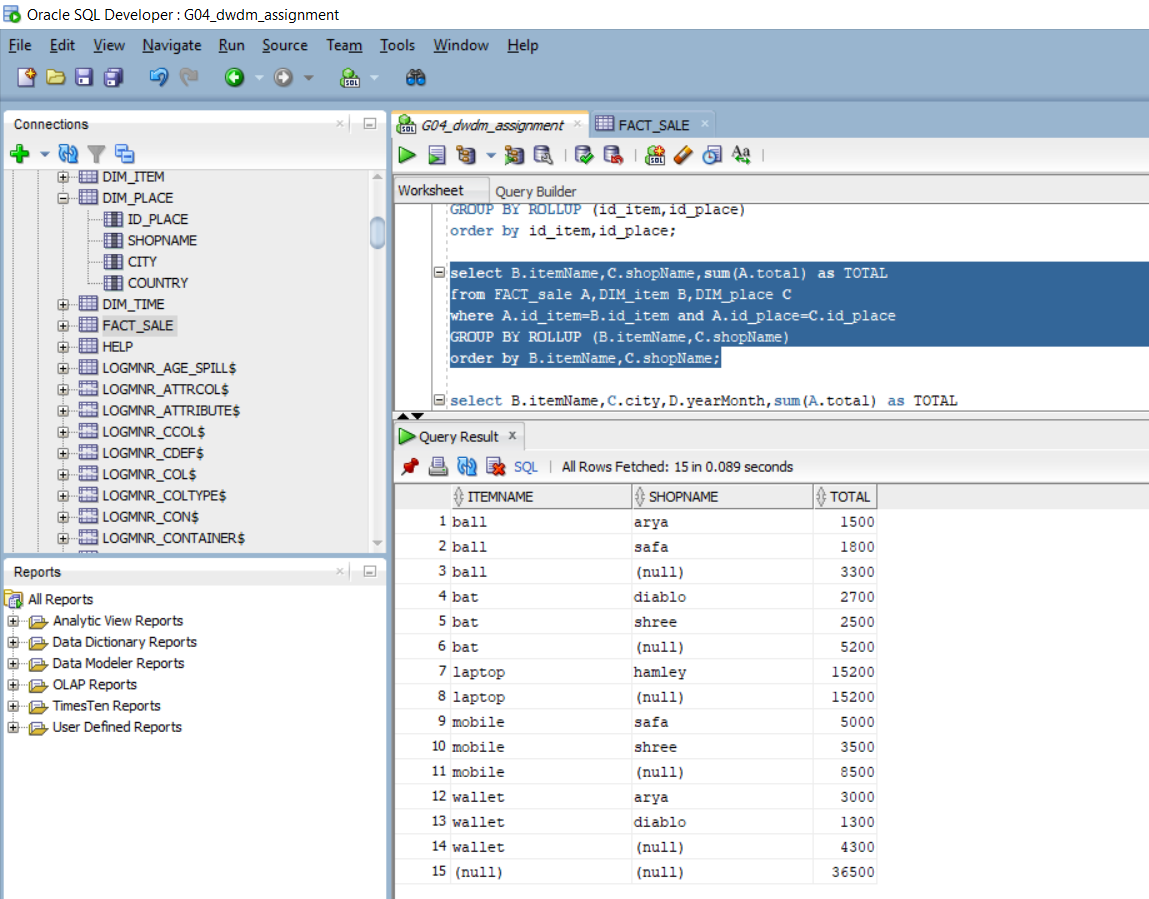
SELECT B.ITEMNAME,C.SHOPNAME,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE

GROUP BY ROLLUP (B.ITEMNAME,C.SHOPNAME)

ORDER BY B.ITEMNAME,C.SHOPNAME;



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.

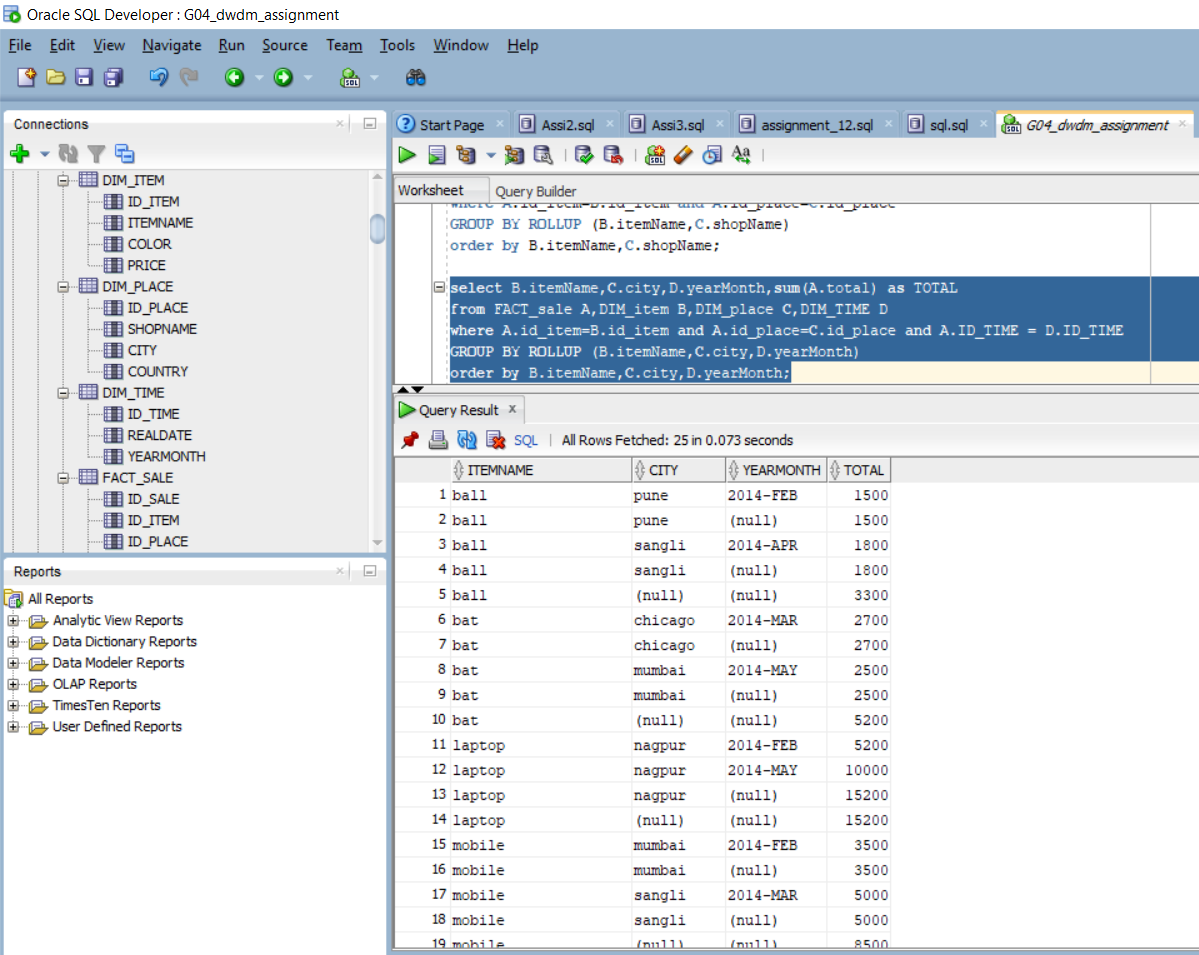
SELECT B.ITEMNAME,C.CITY,D.YEARMONTH,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C,DIM\_TIME D

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE AND A.ID\_TIME = D.ID\_TIME

GROUP BY ROLLUP (B.ITEMNAME,C.CITY,D.YEARMONTH)

ORDER BY B.ITEMNAME,C.CITY,D.YEARMONTH;



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.

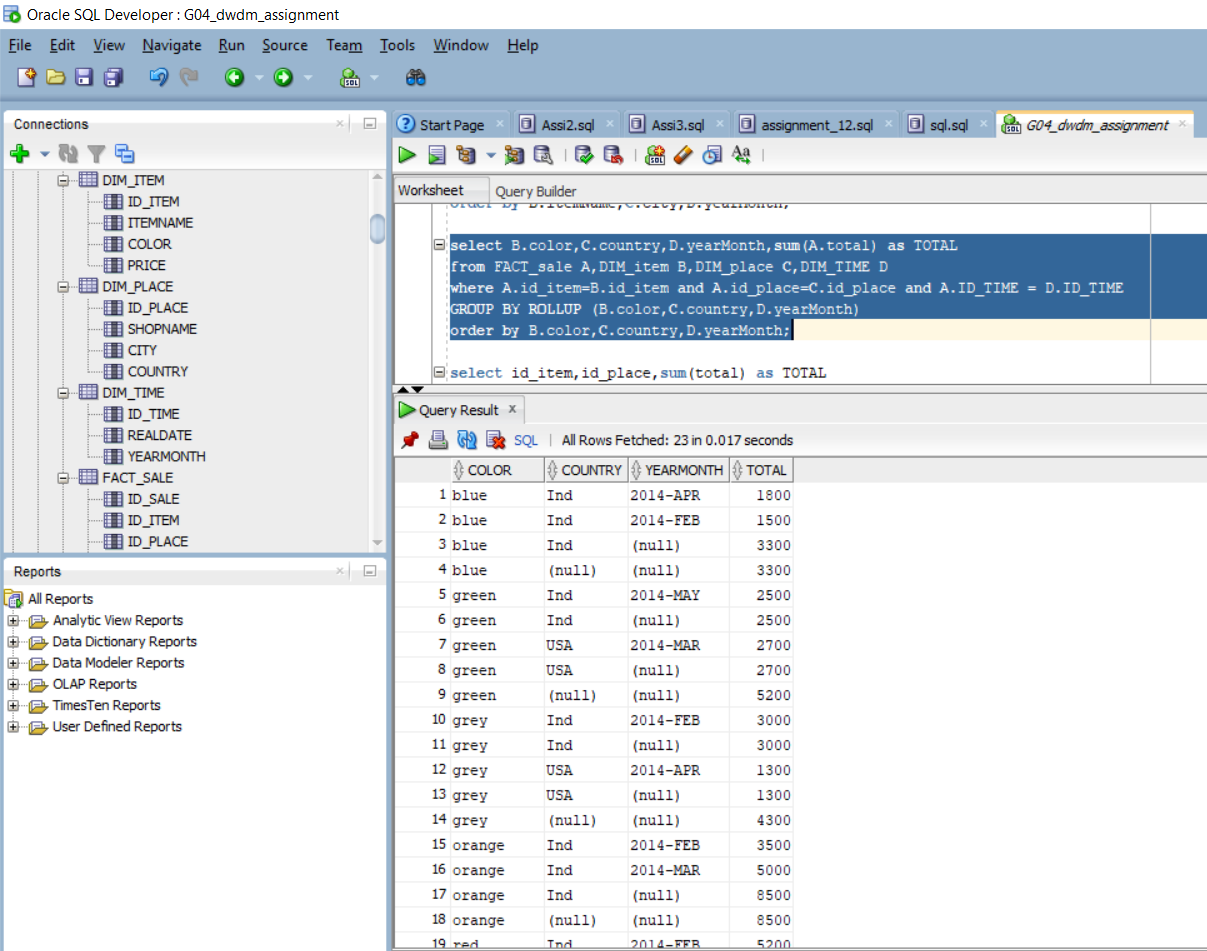
SELECT B.COLOR,C.COUNTRY,D.YEARMONTH,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C,DIM\_TIME D

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE AND A.ID\_TIME = D.ID\_TIME

GROUP BY ROLLUP (B.COLOR,C.COUNTRY,D.YEARMONTH)

ORDER BY B.COLOR,C.COUNTRY,D.YEARMONTH;



Activity 3. Aggregation with CUBE

1) Using CUBE keyword, construct a query that calculates subtotals of the

FACT\_Sale.Total field across the FACT\_Sale.ID\_Item and

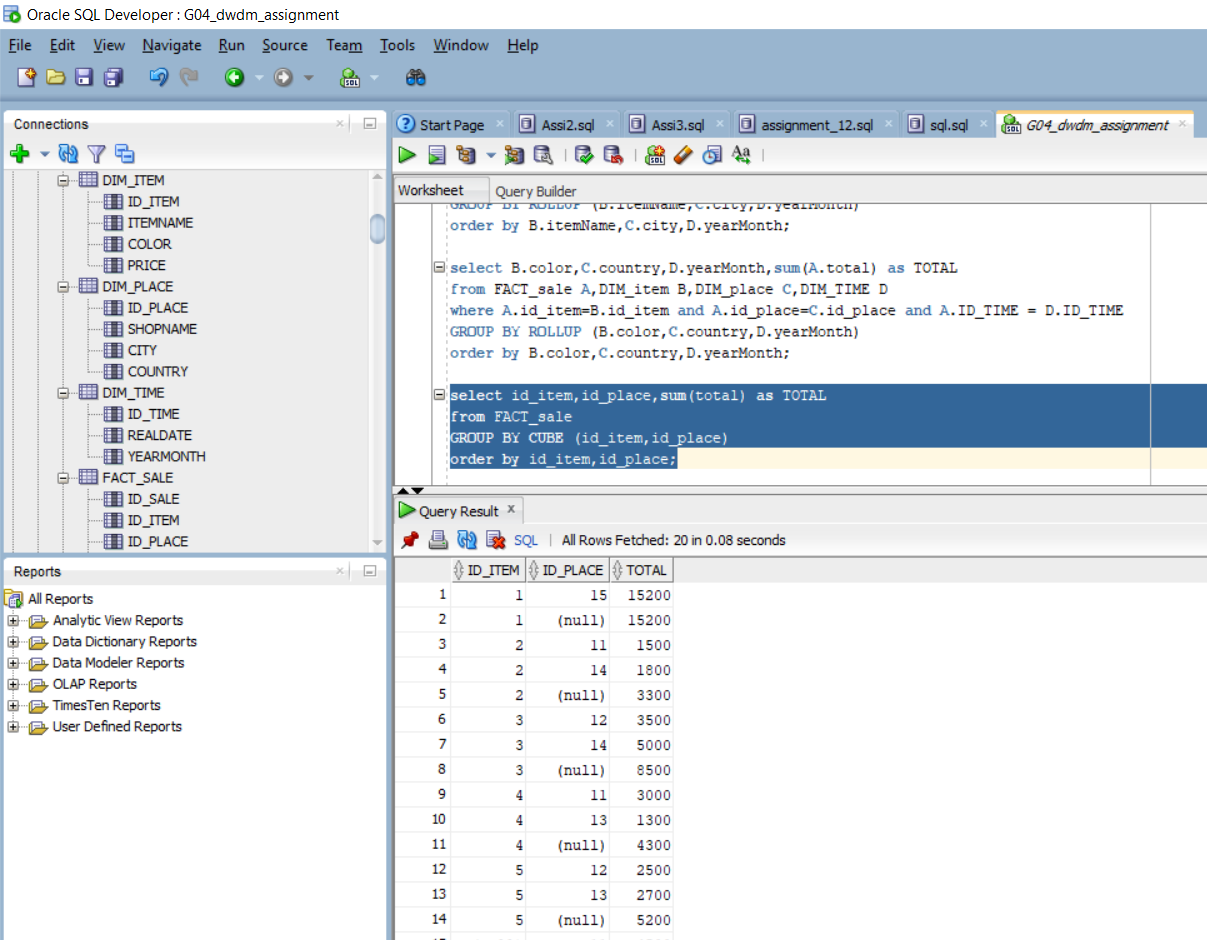
FACT\_Sale.ID\_Place fields.

SELECT ID\_ITEM,ID\_PLACE,SUM(TOTAL) AS TOTAL

FROM FACT\_SALE

GROUP BY CUBE (ID\_ITEM,ID\_PLACE)

ORDER BY ID\_ITEM,ID\_PLACE;



2) Modify the query above to calculate subtotals across the Dim\_Item.ItemName

and Dim\_Place.ShopName fields.

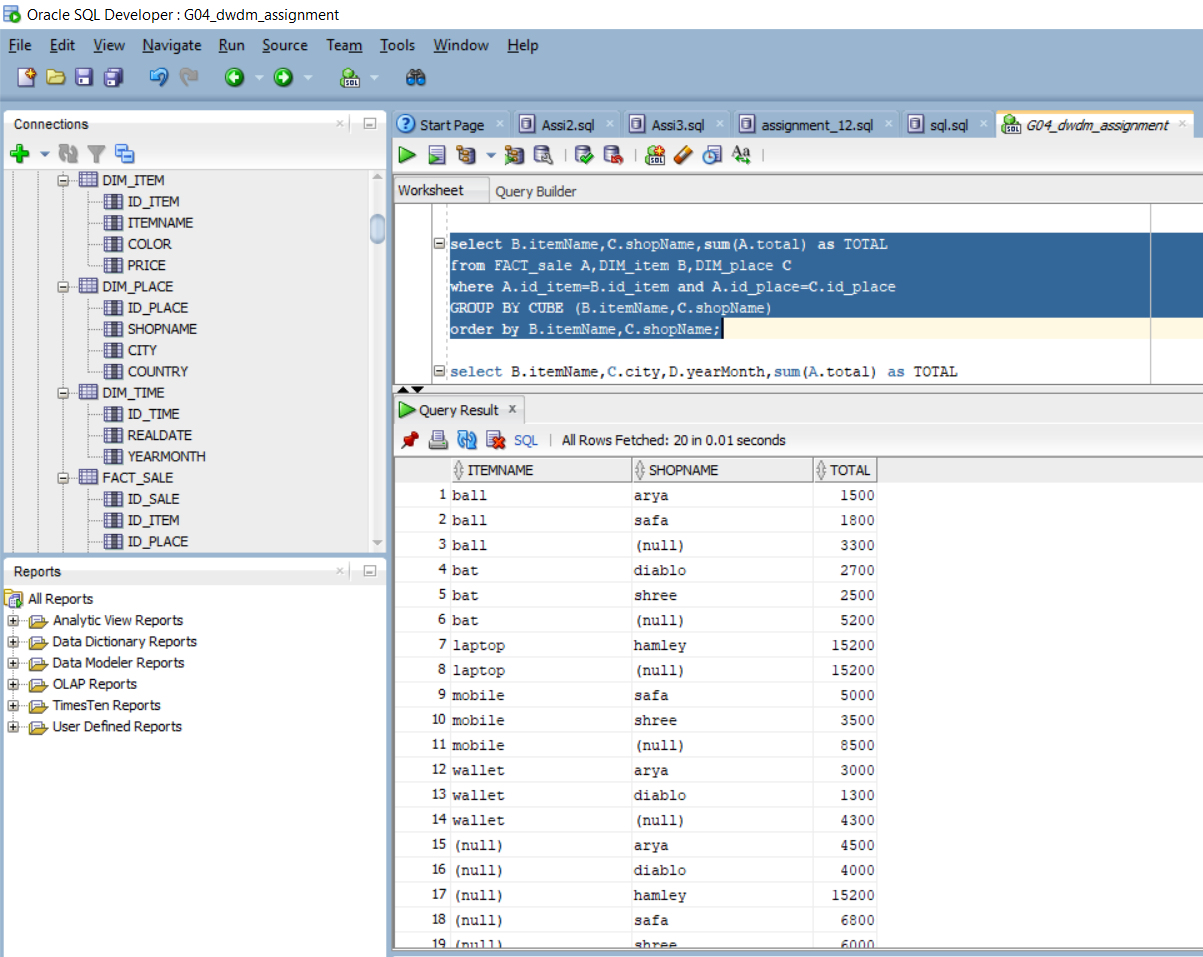
SELECT B.ITEMNAME,C.SHOPNAME,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE

GROUP BY CUBE (B.ITEMNAME,C.SHOPNAME)

ORDER BY B.ITEMNAME,C.SHOPNAME;



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.

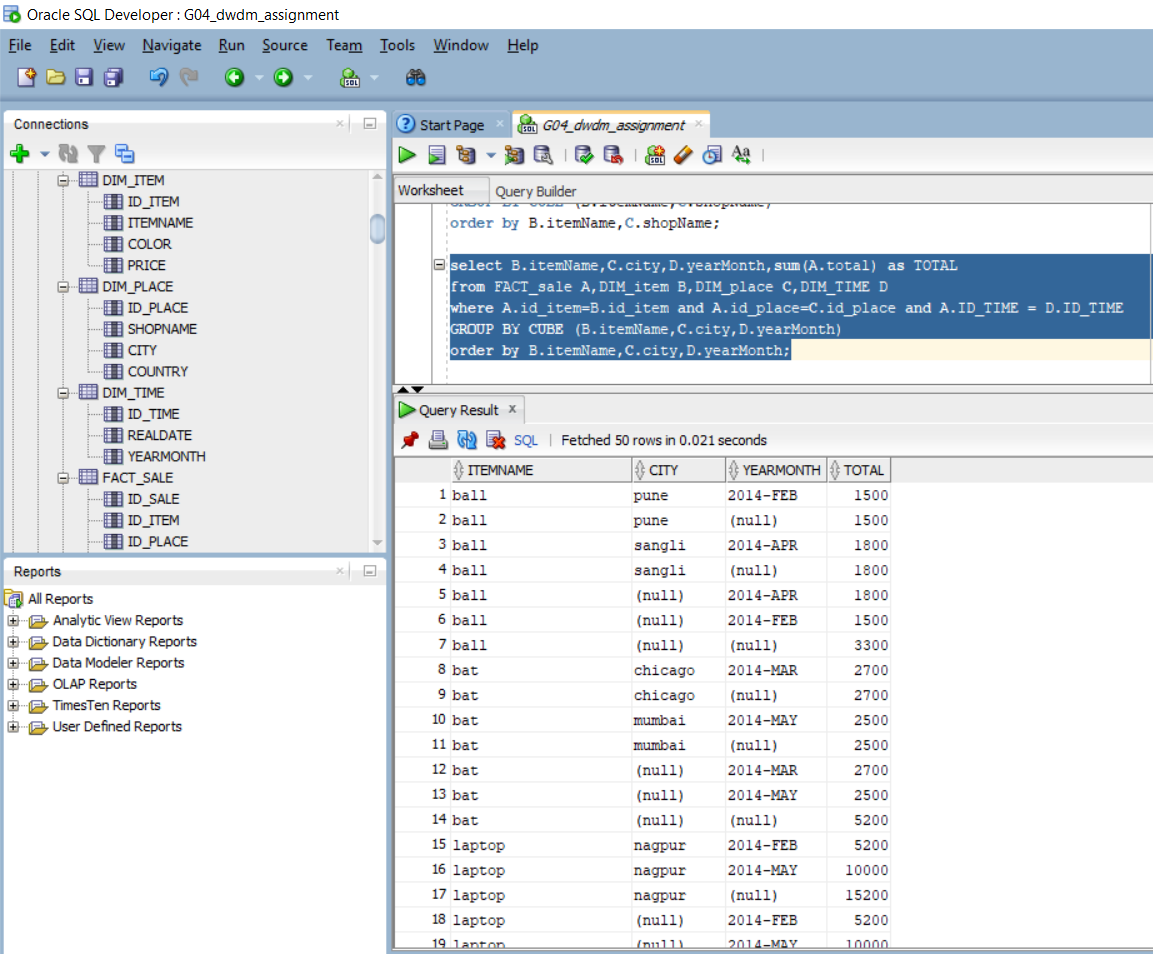
SELECT B.ITEMNAME,C.CITY,D.YEARMONTH,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C,DIM\_TIME D

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE AND A.ID\_TIME = D.ID\_TIME

GROUP BY CUBE (B.ITEMNAME,C.CITY,D.YEARMONTH)

ORDER BY B.ITEMNAME,C.CITY,D.YEARMONTH;



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.

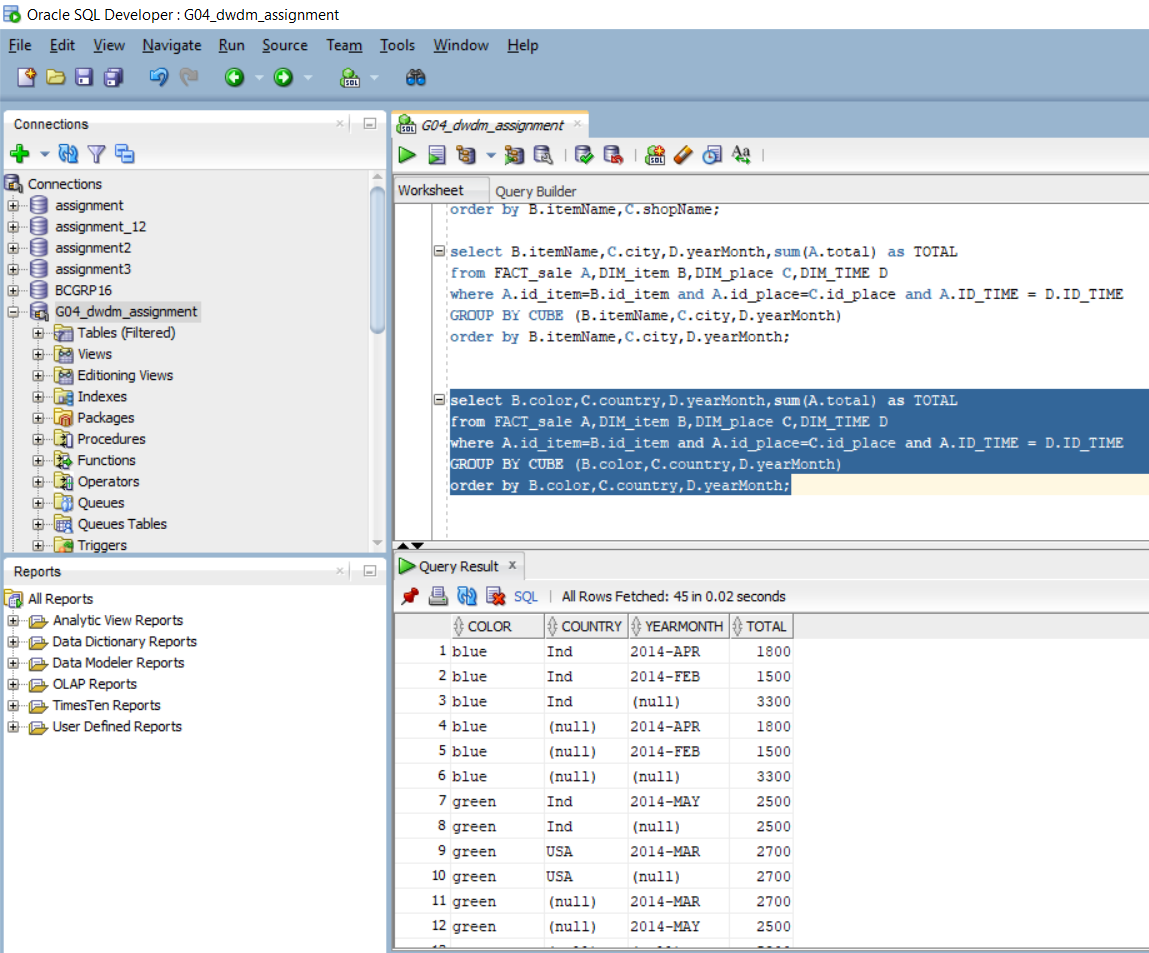
SELECT B.COLOR,C.COUNTRY,D.YEARMONTH,SUM(A.TOTAL) AS TOTAL

FROM FACT\_SALE A,DIM\_ITEM B,DIM\_PLACE C,DIM\_TIME D

WHERE A.ID\_ITEM=B.ID\_ITEM AND A.ID\_PLACE=C.ID\_PLACE AND A.ID\_TIME = D.ID\_TIME

GROUP BY CUBE (B.COLOR,C.COUNTRY,D.YEARMONTH)

ORDER BY B.COLOR,C.COUNTRY,D.YEARMONTH;



**Conclusion**:

We learnt how to design a star schema database with ROLL-UP and CUBE operations.