**Assignment 2**

**Group: DWDM19G04**

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

**Batch: B7**

**Date: 31st August, 2019**

**Title: Data Warehousing for Customer Order Processing**

**Problem Statement:**

To design and implement a data warehouse for a customer order processing system in a company using Oracle 11g.

**Theory:**

Data Warehousing: A data warehousing is defined as a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which aids the strategic use of data.

Types of Data Warehouses are:

1. Enterprise Data Warehouse:

Enterprise Data Warehouse is a centralized warehouse. It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data. It also provide the ability to classify data according to the subject and give access according to those divisions.

1. Operational Data Store:

Operational Data Store, which is also called ODS, are nothing but data store required when neither Data warehouse nor OLTP systems support organizations reporting needs. In ODS, Data warehouse is refreshed in real time. Hence, it is widely preferred for routine activities like storing records of the Employees.

1. Data Mart:

A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

**Output:**

**Headquarter Database:**

CREATE TABLE CUSTOMERDW(

CUSTOMER\_ID INTEGER PRIMARY KEY,

CUSTOMER\_NAME VARCHAR(20),

CITY\_ID INTEGER,

FIRST\_ORDER\_DATE DATE

);

INSERT INTO CUSTOMERDW VALUES(1001,'RITIK',10,'3-FEB-2014');

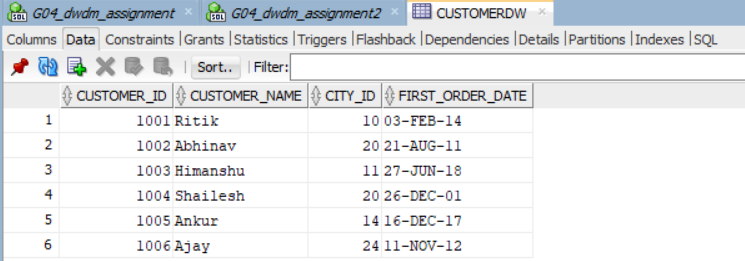
INSERT INTO CUSTOMERDW VALUES(1002,'ABHINAV',20,'21-AUG-2011');

INSERT INTO CUSTOMERDW VALUES(1003,'HIMANSHU',11,'27-JUN-2018');

INSERT INTO CUSTOMERDW VALUES(1004,'SHAILESH',20,'26-DEC-2001');

INSERT INTO CUSTOMERDW VALUES(1005,'ANKUR',14,'16-DEC-2017');

INSERT INTO CUSTOMERDW VALUES(1006,'AJAY',24,'11-NOV-2012');



CREATE TABLE WALKIN\_CUSTOMERS(

CUSTOMER\_ID INTEGER,

TOURISM\_GUIDE INTEGER,

WALKIN\_TIME TIMESTAMP,

FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMERDW(CUSTOMER\_ID)

);

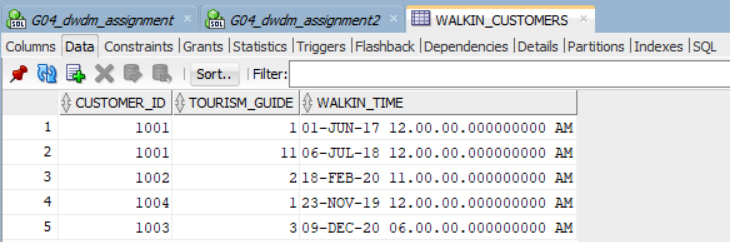
INSERT INTO WALKIN\_CUSTOMERS VALUES(1001,01,'1-JUN-2017');

INSERT INTO WALKIN\_CUSTOMERS VALUES(1001,11,'6-JUL-2018');

INSERT INTO WALKIN\_CUSTOMERS VALUES(1002,02,'18-FEB-2011');

INSERT INTO WALKIN\_CUSTOMERS VALUES(1004,01,'23-NOV-2019');

INSERT INTO WALKIN\_CUSTOMERS VALUES(1003,03,'9-DEC-2006');



CREATE TABLE MAIL\_ORDER\_CUSTOMERS(

CUSTOMER\_ID INTEGER,

POST\_ADDRESS VARCHAR(20),

MAIL\_ORDER\_TIME TIMESTAMP,

FOREIGN KEY(CUSTOMER\_ID) REFERENCES CUSTOMERDW(CUSTOMER\_ID)

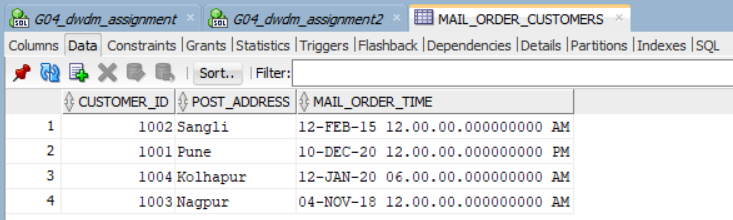
);

INSERT INTO MAIL\_ORDER\_CUSTOMERS VALUES(1002,'SANGLI','12-FEB-2015');

INSERT INTO MAIL\_ORDER\_CUSTOMERS VALUES(1001,'PUNE','10-DEC-2012');

INSERT INTO MAIL\_ORDER\_CUSTOMERS VALUES(1004,'KOLHAPUR','12-JAN-2006');

INSERT INTO MAIL\_ORDER\_CUSTOMERS VALUES(1003,'NAGPUR','4-NOV-2018');



SALES DATABASE

CREATE TABLE HEADQARTERS(

CITY\_ID INTEGER PRIMARY KEY,

CITY\_NAME VARCHAR(20),

HEADQUARTER\_ADDR VARCHAR(20),

STATE VARCHAR(20),

HEADQARTERS\_TIME TIMESTAMP

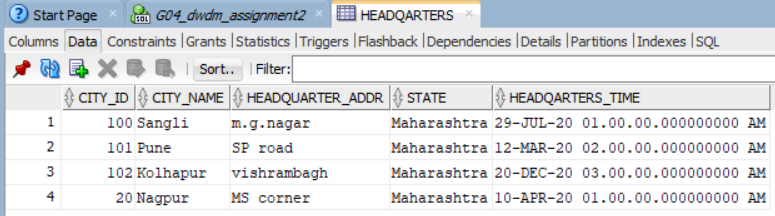
);

INSERT INTO HEADQARTERS VALUES(100,'SANGLI','M.G.NAGAR','MAHARASHTRA','29-JUL-2001');

INSERT INTO HEADQARTERS VALUES(101,'PUNE','SP ROAD','MAHARASHTRA','12-MAR-2002');

INSERT INTO HEADQARTERS VALUES(102,'KOLHAPUR','VISHRAMBAGH','MAHARASHTRA','20-DEC-2003');

INSERT INTO HEADQARTERS VALUES(20,'NAGPUR','MS CORNER','MAHARASHTRA','10-APR-2001');



CREATE TABLE STORES(

STORE\_ID INTEGER PRIMARY KEY,

CITY\_ID INTEGER,

PHONE INTEGER,

STORES\_TIME TIMESTAMP,

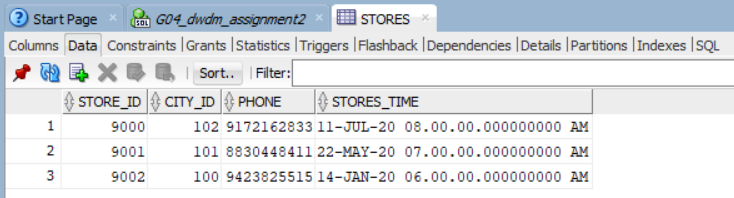
FOREIGN KEY (CITY\_ID) REFERENCES HEADQARTERS(CITY\_ID)

);

INSERT INTO STORES VALUES(9000,102,9172162833,'11-JUL-2008');

INSERT INTO STORES VALUES(9001,101,8830448411,'22-MAY-2007');

INSERT INTO STORES VALUES(9002,100,9423825515,'14-JAN-2006');



CREATE TABLE ITEMS\_2(

ITEM\_ID INTEGER PRIMARY KEY,

DESCRIPTION VARCHAR(20),

ITEM\_SIZE INTEGER,

WEIGHT REAL,

UNIT\_PRICE REAL,

ITEMS\_TIME TIMESTAMP

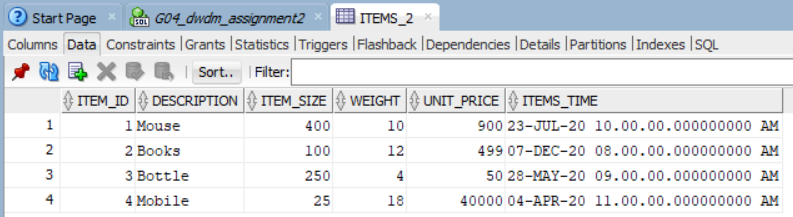
);

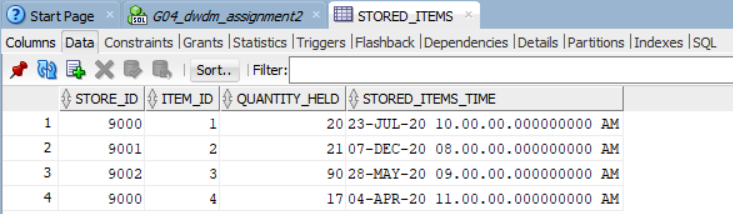
INSERT INTO ITEMS\_2 VALUES(1,'MOUSE',10\*40,10,900,'23-JUL-2010');

INSERT INTO ITEMS\_2 VALUES(2,'BOOKS',10\*10,12,499,'7-DEC-2008');

INSERT INTO ITEMS\_2 VALUES(3,'BOTTLE',5\*50,4,50,'28-MAY-2009');

INSERT INTO ITEMS\_2 VALUES(4,'MOBILE',5\*5,18,40000,'4-APR-2011');





CREATE TABLE STORED\_ITEMS(

STORE\_ID INTEGER,

ITEM\_ID INTEGER,

QUANTITY\_HELD INTEGER,

STORED\_ITEMS\_TIME TIMESTAMP,

PRIMARY KEY(STORE\_ID,ITEM\_ID),

FOREIGN KEY (STORE\_ID) REFERENCES STORES(STORE\_ID),

FOREIGN KEY (ITEM\_ID) REFERENCES ITEMS\_2(ITEM\_ID)

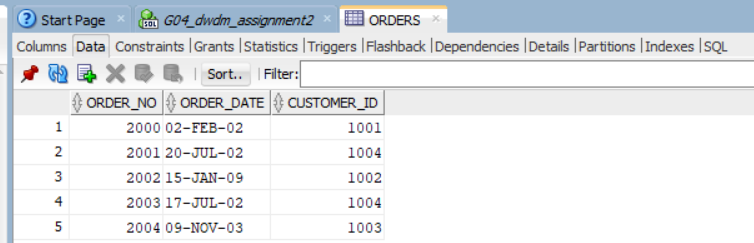
);

INSERT INTO STORED\_ITEMS VALUES(9000,1,20,'23-JUL-2010');

INSERT INTO STORED\_ITEMS VALUES(9001,2,21,'7-DEC-2008');

INSERT INTO STORED\_ITEMS VALUES(9002,3,90,'28-MAY-2009');

INSERT INTO STORED\_ITEMS VALUES(9000,4,17,'4-APR-2011');



CREATE TABLE ORDERS(

ORDER\_NO INTEGER PRIMARY KEY,

ORDER\_DATE DATE,

CUSTOMER\_ID INTEGER

);

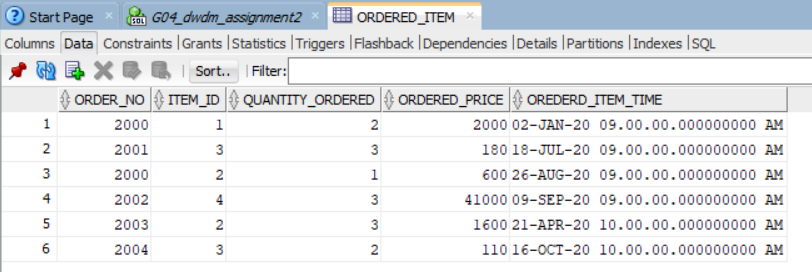
INSERT INTO ORDERS VALUES(2000,'2-FEB-2002',1001);

INSERT INTO ORDERS VALUES(2001,'20-JUL-2002',1004);

INSERT INTO ORDERS VALUES(2002,'15-JAN-2009',1002);

INSERT INTO ORDERS VALUES(2003,'17-JUL-2002',1004);

INSERT INTO ORDERS VALUES(2004,'9-NOV-2003',1003);



CREATE TABLE ORDERED\_ITEM(

ORDER\_NO INTEGER,

ITEM\_ID INTEGER,

QUANTITY\_ORDERED INTEGER,

ORDERED\_PRICE REAL,

OREDERD\_ITEM\_TIME TIMESTAMP,

PRIMARY KEY(ORDER\_NO,ITEM\_ID),

FOREIGN KEY (ORDER\_NO) REFERENCES ORDERS(ORDER\_NO),

FOREIGN KEY (ITEM\_ID) REFERENCES ITEMS\_2(ITEM\_ID)

);

INSERT INTO ORDERED\_ITEM VALUES(2000,1,2,2000,'2-JAN-2009');

INSERT INTO ORDERED\_ITEM VALUES(2001,3,3,180,'18-JUL-2009');

INSERT INTO ORDERED\_ITEM VALUES(2000,2,1,600,'26-AUG-2009');

INSERT INTO ORDERED\_ITEM VALUES(2002,4,3,41000,'9-SEP-2009');

INSERT INTO ORDERED\_ITEM VALUES(2003,2,3,1600,'21-APR-2010');

INSERT INTO ORDERED\_ITEM VALUES(2004,3,2,110,'16-OCT-2010');

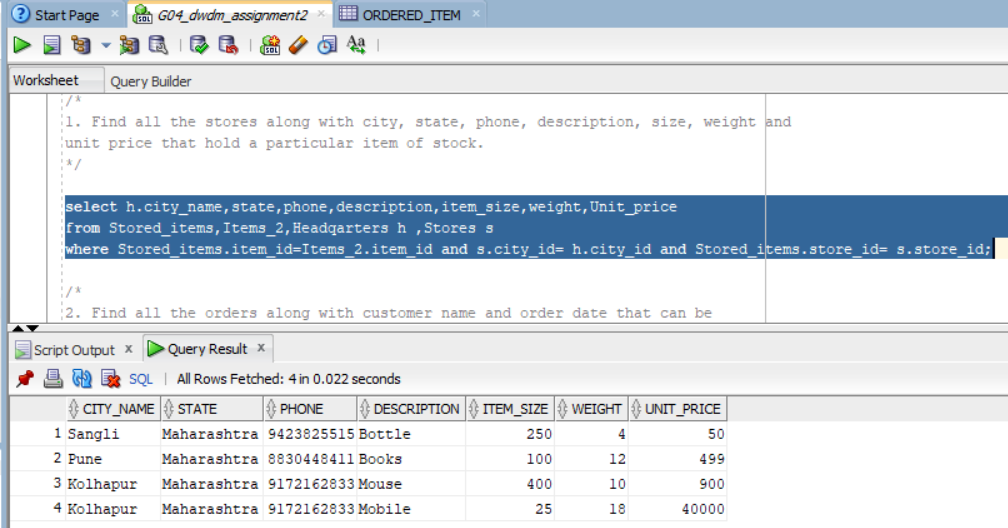
1. Find all the stores along with city, state, phone, description, size, weight and

unit price that hold a particular item of stock.

SELECT H.CITY\_NAME,STATE,PHONE,DESCRIPTION,ITEM\_SIZE,WEIGHT,UNIT\_PRICE

FROM STORED\_ITEMS,ITEMS\_2,HEADQARTERS H ,STORES S

WHERE STORED\_ITEMS.ITEM\_ID=ITEMS\_2.ITEM\_ID AND S.CITY\_ID= H.CITY\_ID AND STORED\_ITEMS.STORE\_ID= S.STORE\_ID;



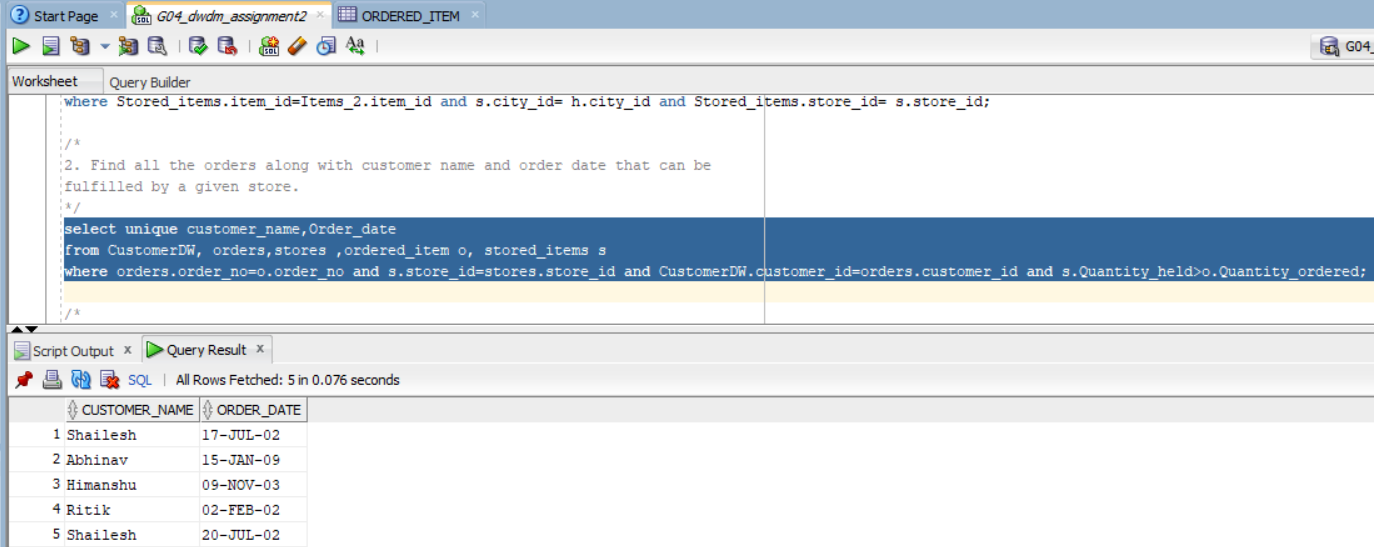
2. Find all the orders along with customer name and order date that can be

fulfilled by a given store.

SELECT UNIQUE CUSTOMER\_NAME,ORDER\_DATE

FROM CUSTOMERDW, ORDERS,STORES ,ORDERED\_ITEM O, STORED\_ITEMS S

WHERE ORDERS.ORDER\_NO=O.ORDER\_NO AND S.STORE\_ID=STORES.STORE\_ID AND CUSTOMERDW.CUSTOMER\_ID=ORDERS.CUSTOMER\_ID AND S.QUANTITY\_HELD>O.QUANTITY\_ORDERED;



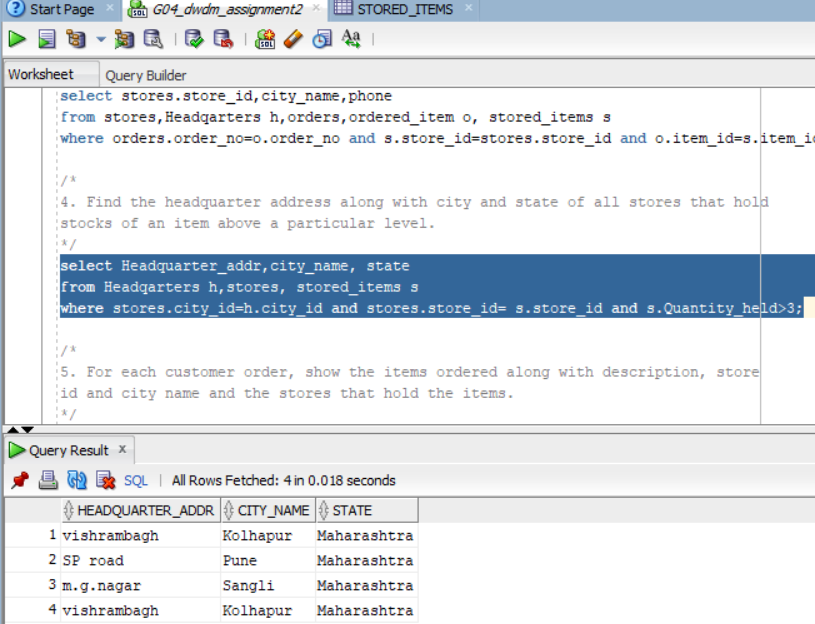
3. Find all stores along with city name and phone that hold items ordered by

given customer.

SELECT STORES.STORE\_ID,CITY\_NAME,PHONE

FROM STORES,HEADQARTERS H,ORDERS,ORDERED\_ITEM O, STORED\_ITEMS S

WHERE ORDERS.ORDER\_NO=O.ORDER\_NO AND S.STORE\_ID=STORES.STORE\_ID AND O.ITEM\_ID=S.ITEM\_ID AND H.CITY\_ID= STORES.CITY\_ID;



4. Find the headquarter address along with city and state of all stores that hold

stocks of an item above a particular level.

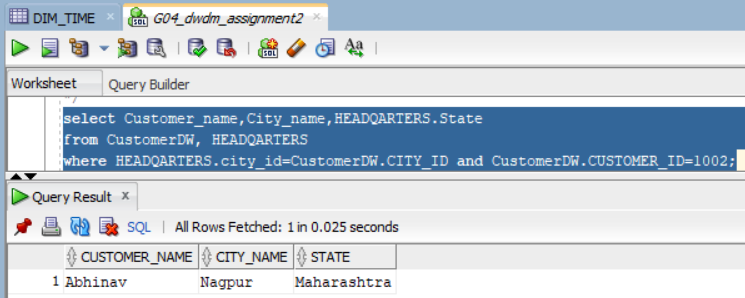
SELECT HEADQUARTER\_ADDR,CITY\_NAME, STATE

FROM HEADQARTERS H,STORES, STORED\_ITEMS S

WHERE STORES.CITY\_ID=H.CITY\_ID AND STORES.STORE\_ID= S.STORE\_ID AND S.QUANTITY\_HELD>3;

5. For each customer order, show the items ordered along with description, store

id and city name and the stores that hold the items.



SELECT UNIQUE O.ITEM\_ID,DESCRIPTION,S.STORE\_ID,CITY\_NAME

FROM ITEMS\_2,STORES,STORED\_ITEMS S,ORDERS,ORDERED\_ITEM O,HEADQARTERS H,CUSTOMERDW

WHERE H.CITY\_ID=STORES.CITY\_ID

AND ITEMS\_2.ITEM\_ID=O.ITEM\_ID

AND O.ITEM\_ID=S.ITEM\_ID AND STORES.STORE\_ID= S.STORE\_ID;

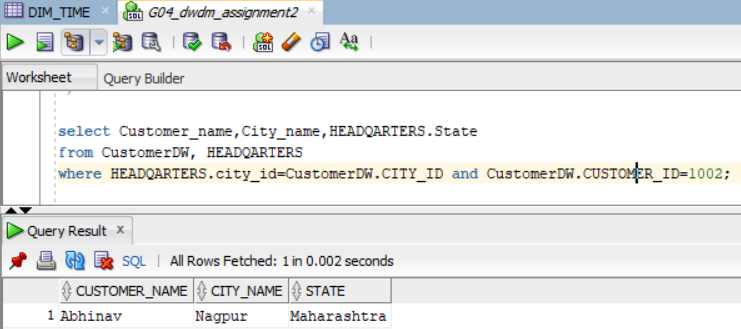
6. Find the city and the state in which a given customer lives.

SELECT CUSTOMER\_NAME,CITY\_NAME,HEADQARTERS.STATE

FROM CUSTOMERDW, HEADQARTERS

WHERE HEADQARTERS.CITY\_ID=CUSTOMERDW.CITY\_ID AND CUSTOMERDW.CUSTOMER\_ID=1002;

7. Find the stock level of a particular item in all stores in a particular city.



SELECT CUSTOMER\_NAME,CITY\_NAME,HEADQARTERS.STATE

FROM CUSTOMERDW, HEADQARTERS

WHERE HEADQARTERS.CITY\_ID=CUSTOMERDW.CITY\_ID AND CUSTOMERDW.CUSTOMER\_ID=1002;

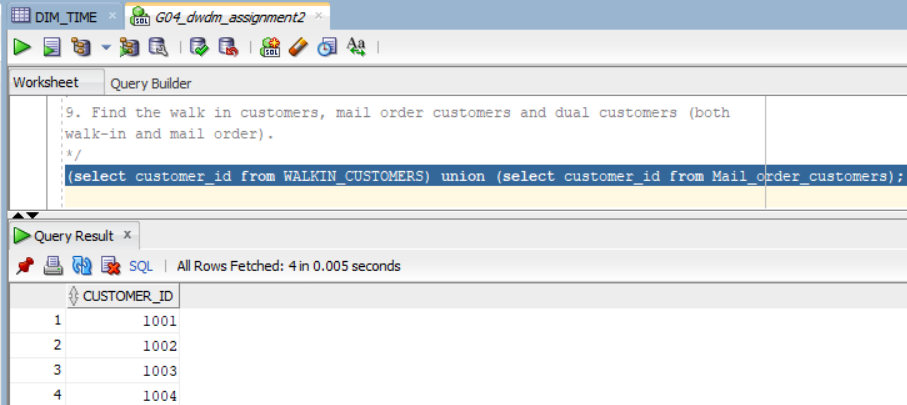
8. Find the items, quantity ordered, customer, store and city of an order.

SELECT UNIQUE CUSTOMER\_NAME,CITY\_ID,QUANTITY\_ORDERED,DESCRIPTION,S.STORE\_ID

FROM ORDERS,CUSTOMERDW,ORDERED\_ITEM,ITEMS\_2, STORED\_ITEMS S

WHERE ORDERS.ORDER\_NO = ORDERED\_ITEM.ORDER\_NO AND

CUSTOMERDW.CUSTOMER\_ID=ORDERS.CUSTOMER\_ID AND ORDERED\_ITEM.ITEM\_ID= ITEMS\_2.ITEM\_ID AND S.ITEM\_ID= ITEMS\_2.ITEM\_ID;



9. Find the walk in customers, mail order customers and dual customers (both

walk-in and mail order).

(SELECT CUSTOMER\_ID FROM WALKIN\_CUSTOMERS) UNION (SELECT CUSTOMER\_ID FROM MAIL\_ORDER\_CUSTOMERS);

**Conclusion**:

We implemented multiple databases for an OLTP scenario and ran various queries over the entire system.