**1) Understand the Data model to build DWH:**

1: Identify the given data model and briefly explain about it

The given data model is a Snowflake model.

Snowflake model is a logical arrangement of tables in a multi-dimensional database where the ER-diagram represents a shape of a Snowflake.

It is also an extension of a STAR schema where the dimension tables are normalized and splits data into another tables.

If we try to derive one dimension table from another then we manage the size of dimension table.

Snowflake schema uses less storage space as the data is normalized and hence minimum data redundancy.

It also offers protection from data integrity issues which is why maintenance is much simple.

It describes a many to many relationship.

2: How to set the dependencies during stage table and target table loads

Here, we have a Stage model and a Target model. As the ETL process says, we first need to Extract data from the source. Therefore, after creating respective tables for Stage model we need to load data in it. Irrespective of what type of data and in which format it is.

Then comes the cleansing process; after loading data we need to find the duplicates and unwanted data. We need to make sure that the data is non-redundant and contains only unique records.

Once we maintain the uniqueness, we need to create Primary and Foreign keys on them. Hence, establishing the relationship between them.

Now that the data in Stage model is cleaned up, next step is to load this data into the Target model.

Therefore, after creating respective tables we’ll load the Stage model data to Target model establishing a relationship between them.

3: Common issues with this model

Queries can be a little complex, including many level of joins.

Maintenance can be more complex due to large number of tables in database.

Because of joins on several tables the fetching cost is high which makes it a slow process.

4: Are there any options to convert this model to STAR?

The major difference between a STAR schema and Snowflake schema is that a STAR schema has only one fact table whereas a Snowflake has more than one fact table.

So yes, if we want to convert a Snowflake to Star schema then we must concatenate the fact tables from Snowflake schema into one fact table to form a Star schema.

**2) Create Stage tables:**

create table items(

item\_id number(20,0),

sku varchar(100),

type\_name varchar(30),

salesdescription varchar(100),

class\_id number(20,0),

ws\_merch\_department\_id number(20,0),

ws\_merch\_collection\_id number(20,0),

ws\_merch\_class\_id number(20,0),

ws\_merch\_subclass\_id number(20,0)

);

create table departments(

date\_last\_modified date ,

department\_id number(20,0) ,

isinactive varchar(5) ,

name varchar(50) ,

ws\_description varchar(50)

);

create table item\_merch\_collection(

item\_merch\_collection\_id number(20,0) ,

description varchar(50) ,

item\_merch\_collection\_na varchar(50)

);

create table locations(

location\_id number(20,0) ,

address varchar(120) ,

city varchar(50),

country varchar(50),

date\_last\_modified date ,

full\_name varchar(100),

isinactive varchar(5),

name varchar(50)

);

create table item\_merch\_class(

item\_merch\_class\_id number(20,0) ,

description varchar(50) ,

item\_merch\_class\_name varchar(5)

);

create table item\_merch\_subclass(

item\_merchandise\_subclass\_id number(20,0) ,

description varchar(50) ,

item\_merch\_subclass\_name varchar(10)

);

create table classes(

class\_id number(20,0),

date\_last\_modified date ,

full\_name varchar(30) ,

isinactive varchar(5) ,

name varchar(5)

);

create table departments(

date\_last\_modified date

department\_id number(20,0),

isinactive varchar(5) ,

name varchar(50) ,

ws\_description varchar(50)

);

create table transaction\_lines(

transaction\_id number(20,0) ,

transaction\_line\_id number (20,0) ,

location\_id number(20,0) ,

department\_id number(20,0) ,

item\_id number(20,0) ,

amount number(8,2) ,

cost number(8,2) ,

units number(5,0)

);

create table transactions(

transaction\_id number(20,0) ,

tranid varchar(30) ,

transaction\_type varchar(50) ,

trandate date ,

channel\_id number(20,0)

);

create table channel(

date\_created date ,

is\_record\_inactive char ,

last\_modified\_date date ,

list\_id number(20,0) ,

list\_item\_name varchar(20)

);

create table locations(

location\_id number(20,0),

address varchar(120) ,

city varchar(50),

country varchar(50) ,

date\_last\_modified date ,

full\_name varchar(100),

isinactive varchar(5) ,

name varchar(50)

);

**3) Load Data in tables:**

SQL> insert into department\_dim(date\_last\_modified, department\_id, isinactive, name, ws\_description)(select \* from transaction.departments);

105 rows created.

SQL> create sequence s4;

Sequence created.

SQL> update department\_dim set kpi\_dw\_skey=s4.nextval;

105 rows updated.

SQL> insert into item\_dim(item\_id, sku, type\_name, salesdescription, kpi\_class\_skey, ws\_merch\_department\_skey, ws\_merch\_collection\_skey, ws\_merch\_class\_skey, ws\_merch\_subclass\_skey)(select item\_id, sku, type\_name, salesdescription, class\_id, ws\_merch\_department\_id, ws\_merch\_collection\_id, ws\_merch\_class\_id, ws\_merch\_subclass\_id from transaction.items);

78 rows created.

SQL> create sequence s5;

Sequence created.

SQL> update item\_dim set kpi\_dw\_skey=s5.nextval;

78 rows updated.

SQL> insert into class\_dim(class\_id, date\_last\_modified, full\_name, isinactive, name)(select \* from transaction.classes);

6 rows created.

SQL> create sequence s6;

Sequence created.

SQL> update class\_dim set kpi\_dw\_skey=s6.nextval;

6 rows updated.

SQL> insert into item\_merch\_department\_dim(item\_merch\_department\_id, description, item\_merch\_department\_na)(select \* from transaction.item\_merch\_department);

87 rows created.

SQL> create sequence s7;

Sequence created.

SQL> update item\_merch\_department\_dim set kpi\_dw\_skey=s7.nextval;

87 rows updated.

SQL> insert into item\_merch\_collection\_dim(item\_merch\_collection\_id, description, item\_merch\_collection\_na)(select \* from transaction.item\_merch\_collection);

86 rows created.

SQL> create sequence s8;

Sequence created.

SQL> update item\_merch\_collection\_dim set kpi\_dw\_skey=s8.nextval;

86 rows updated.

SQL> insert into item\_merch\_class\_dim(item\_merch\_class\_id, description, item\_merch\_class\_name)(select \* from transaction.item\_merch\_class);

83 rows created.

SQL> create sequence s9;

Sequence created.

SQL> update item\_merch\_class\_dim set kpi\_dw\_skey=s9.nextval;

83 rows updated.

SQL> create sequence s10;

Sequence created.

SQL> update item\_merch\_subclass\_dim set kpi\_dw\_skey=s10.nextval;

85 rows updated.

SQL> insert into item\_merch\_subclass\_dim(item\_merch\_subclass\_id, description, item\_merch\_subclass\_name)(select \* from transaction.item\_merch\_subclass);

85 rows created.

**4) Analyze keys:**

SQL> alter table channel\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table location\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table transaction\_line\_fact modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table department\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table item\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table class\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table item\_merch\_department\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table item\_merch\_collection\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table item\_merch\_class\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table item\_merch\_subclass\_dim modify kpi\_dw\_skey primary key;

Table altered.

SQL> alter table transaction\_line\_fact modify kpi\_channel\_skey references channel\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table transaction\_line\_fact modify kpi\_location\_skey references location\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table transaction\_line\_fact modify kpi\_department\_skey references department\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table transaction\_line\_fact modify kpi\_item\_skey references item\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table item\_dim modify kpi\_class\_skey references class\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table item\_dim modify ws\_merch\_department\_skey references item\_merch\_department\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table item\_dim modify ws\_merch\_collection\_skey references item\_merch\_collection\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table item\_dim modify ws\_merch\_class\_skey references item\_merch\_class\_dim(kpi\_dw\_skey);

Table altered.

SQL> alter table item\_dim modify ws\_merch\_subclass\_skey references item\_merch\_subclass\_dim(kpi\_dw\_skey);

Table altered.

**5) Delete duplicate records:**

SQL> delete from items where rowid not in (select min(rowid) from items group by item\_id, class\_id);

3 rows deleted.

delete from items where ws\_merch\_collection\_id in (select distinct(ws\_merch\_collection\_id) from items where ws\_merch\_collection\_id not in (select item\_merch\_collection\_id from item\_merch\_collection));

delete from items where ws\_merch\_subclass\_id in (select distinct(ws\_merch\_subclass\_id) from items where ws\_merch\_subclass\_id not in (select item\_merch\_subclass\_id from item\_merch\_subclass));

delete from transaction\_line where department\_id in (select distinct(department\_id) from transaction\_line where department\_id not in (select department\_id from departments));

delete from transaction\_line where department\_id in (select distinct(department\_id) from departments where department\_id not in (select department\_id from transaction\_line));

**6) Create Primary key:**

SQL> alter table channel enable primary key;

Table altered.

SQL> alter table locations enable primary key;

Table altered.

SQL> alter table departments enable primary key;

Table altered.

SQL> alter table classes enable primary key;

Table altered.

SQL> alter table items enable primary key;

Table altered.

SQL> alter table item\_merch\_department enable primary key;

Table altered.

SQL> alter table item\_merch\_collection enable primary key;

Table altered.

SQL> alter table item\_merch\_class enable primary key;

Table altered.

SQL> alter table item\_merch\_subclass enable primary key;

Table altered.

**7) Identify the relationship between tables:**

SQL> alter table items modify class\_id references classes(class\_id);

Table altered.

SQL> alter table items modify ws\_merch\_department\_id references item\_merch\_department(item\_merch\_department\_id);

Table altered.

SQL> alter table transaction\_line modify location\_id references locations(location\_id);

Table altered.

SQL> alter table transaction\_line modify item\_id references items(item\_id);

Table altered.

SQL> alter table transactions modify channel\_id references channel(list\_id);

Table altered.

**8) Create Target tables:**

create table channel\_dim(date\_created date, is\_record\_inactive char, last\_modified\_date date, list\_id number(20), list\_item\_name varchar(20), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc channel\_dim

create table location\_dim(location\_id number(20), address varchar(100), city varchar(50), country varchar(50), date\_last\_modified date, full\_name varchar(50), isinactive varchar(5), name varchar(50), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc location\_dim

create table transaction\_line\_fact(transaction\_id number(20), transaction\_line\_id number(20), tranid varchar(30), transaction\_type varchar(50), trandate date, kpi\_channel\_skey number(20), kpi\_location\_skey number(20), kpi\_deapartment\_skey number(20), kpi\_item\_skey number(20), amount number(8, 2), cost number(8, 2), units number(5), kpi\_dw\_skey number(20));

desc transaction\_line\_fact

create table department\_dim(date\_last\_modified date, department\_id number(20), isinactive varchar(5), name varchar(10), ws\_description varchar(50), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc department\_dim

create table item\_dim(item\_dim number(20), sku varchar(100), tyoe\_name varchar(30), salesdescription varchar(100), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date, kpi\_class\_skey number(20), ws\_merch\_department\_skey number(20), ws\_merch\_collection\_skey number(20), ws\_merch\_class\_skey number(20), ws\_merch\_subclass\_skey number(20));

desc item\_dim

create table class\_dim(class\_id number(20), date\_last\_modified date, full\_name varchar(30), isinactive varchar(5), name varchar(5), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc class\_dim

create table item\_merch\_department\_dim(item\_merch\_department\_id number(20), description varchar(50), item\_merch\_department\_na varchar(10), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc item\_merch\_department\_dim

create table item\_merch\_collection\_dim(item\_merch\_collectiojn\_id number(20), description varchar(50), item\_merch\_collection\_na varchar(50), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc item\_merch\_collection\_dim

create table item\_merch\_class\_dim(item\_merch\_class\_id number(20), description varchar(50), item\_merch\_class\_name varchar(5), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc item\_merch\_class\_dim

create table item\_merch\_subclass\_dim(item\_merch\_subclass\_id number(20), description varchar(50), item\_merch\_subclass\_name varchar(10), kpi\_dw\_skey number(20), kpi\_dw\_insert\_date date, kpi\_dw\_update\_date date);

desc item\_merch\_subclass\_dim

**9) Load data:**

SQL> insert into channel\_dim(date\_created, is\_record\_inactive, last\_modified\_date, list\_id, list\_item\_name)(select \* from transaction.channel);

5 rows created.

SQL> update channel\_dim set kpi\_dw\_skey=s1.nextval;

5 rows updated.

insert into transaction\_line\_fact(transaction\_id, transaction\_line\_id, tranid, transaction\_type, trandate, amount, cost, units)(select tl.transaction\_id, tl.transaction\_line\_id, t.tranid, t.transaction\_type, t.trandate, tl.amount, tl.cost, tl.units from transaction.transaction\_line tl join transaction.transactions t on t.transaction\_id=tl.transaction\_id);

desc transaction\_line\_fact;

drop sequence s3;

create sequence s3;

update transaction\_line\_fact set kpi\_dw\_skey=s3.nextval;

update transaction\_line\_fact set kpi\_channel\_skey = (select c.kpi\_dw\_skey from channel\_dim c join transaction\_line\_fact t on c.kpi\_dw\_skey=t.kpi\_dw\_skey);

update transaction\_line\_fact f set f.kpi\_channel\_skey = (select c.kpi\_dw\_skey from channel\_dim c where c.kpi\_dw\_skey = f.kpi\_dw\_skey);

update transaction\_line\_fact f set f.kpi\_location\_skey = (select l.kpi\_dw\_skey from location\_dim l where l.kpi\_dw\_skey = f.kpi\_dw\_skey);

update transaction\_line\_fact f set f.kpi\_department\_skey = (select d.kpi\_dw\_skey from department\_dim d where d.kpi\_dw\_skey = f.kpi\_dw\_skey);

update transaction\_line\_fact f set f.kpi\_item\_skey = (select i.kpi\_dw\_skey from item\_dim i where i.kpi\_dw\_Skey = f.kpi\_dw\_skey);

select \* from transaction\_line\_fact;

select \* from item\_dim;

truncate table item\_dim;

alter table item\_dim disable primary key cascade;

insert into item\_dim(item\_id, sku, type\_name, salesdescription)(select item\_id, sku, type\_name, salesdescription from TRANSACTION.items);

drop SEQUENCE s5;

create sequence s5;

update item\_dim set kpi\_dw\_skey=s5.nextval;

update item\_dim i set i.kpi\_class\_skey = (select c.kpi\_dw\_skey from class\_dim c where c.kpi\_dw\_skey = i.kpi\_dw\_skey);

update item\_dim i set i.ws\_merch\_department\_skey = (select id.kpi\_dw\_skey from item\_merch\_department\_dim id where id.kpi\_dw\_skey=i.kpi\_dw\_skey);

update item\_dim i set i.ws\_merch\_collection\_skey = (select id.kpi\_dw\_skey from item\_merch\_collection\_dim id where id.kpi\_dw\_skey=i.kpi\_dw\_skey);

update item\_dim i set i.ws\_merch\_class\_skey = (select id.kpi\_dw\_skey from item\_merch\_class\_dim id where id.kpi\_dw\_skey=i.kpi\_dw\_skey);

update item\_dim i set i.ws\_merch\_subclass\_skey = (select id.kpi\_dw\_skey from item\_merch\_subclass\_dim id where id.kpi\_dw\_skey=i.kpi\_dw\_skey);

update item\_dim i set i.brand\_name = (select c.name from class\_dim c where c.kpi\_dw\_skey=i.kpi\_dw\_skey);

select \* from item\_dim;

alter table item\_dim modify kpi\_dw\_insert\_date date default sysdate;

alter table item\_dim modify kpi\_dw\_update\_date date default sysdate;

update item\_dim set kpi\_dw\_insert\_date = sysdate where item\_id is not null;

update item\_dim set kpi\_dw\_update\_date = sysdate where item\_id is not null;

update class\_dim set kpi\_dw\_insert\_date = sysdate where class\_id is not null;

update class\_dim set kpi\_dw\_update\_date = sysdate where class\_id is not null;

update department\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update department\_dim set kpi\_dw\_update\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_class\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_class\_dim set kpi\_dw\_update\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_collection\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_collection\_dim set kpi\_dw\_update\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_department\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_department\_dim set kpi\_dw\_update\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_subclass\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update item\_merch\_subclass\_dim set kpi\_dw\_update\_date = sysdate where kpi\_dw\_skey is not null;

update location\_dim set kpi\_dw\_insert\_date = sysdate where kpi\_dw\_skey is not null;

update location\_dim set kpi\_dw\_update\_date = sysdate;

commit;

**10) Create brand\_name column in item\_dim:**

alter table item\_dim add brand\_name varchar(5);

SQL> desc item\_dim

Name Null? Type

----------------------------------------------------------------- -------- --------------------------------------------

ITEM\_ID NUMBER(20)

SKU VARCHAR2(100)

TYPE\_NAME VARCHAR2(30)

SALESDESCRIPTION VARCHAR2(100)

KPI\_DW\_SKEY NOT NULL NUMBER(20)

KPI\_DW\_INSERT\_DATE DATE

KPI\_DW\_UPDATE\_DATE DATE

KPI\_CLASS\_SKEY NUMBER(20)

WS\_MERCH\_DEPARTMENT\_SKEY NUMBER(20)

WS\_MERCH\_COLLECTION\_SKEY NUMBER(20)

WS\_MERCH\_CLASS\_SKEY NUMBER(20)

WS\_MERCH\_SUBCLASS\_SKEY NUMBER(20)

BRAND\_NAME VARCHAR2(5)

update item\_dim id set id.kpi\_class\_skey = (select i.class\_id from transaction.items i where id.item\_id=i.item\_id);

**11) Create kpi\_item\_dim\_flat:**

create table item\_dim\_flat(sku varchar(100), item\_type varchar(30), brand varchar(5), merch\_department varchar(50), merch\_dept\_name varchar(10), merch\_collection varchar(50), merch\_collection\_name varchar(50), merch\_class varchar(50), merch\_class\_name varchar(5), merch\_subclass varchar(50), merch\_subclass\_name varchar(10), item\_skey number);

SQL> desc item\_dim\_flat

Name Null? Type

----------------------------------------------------------------- -------- --------------------------------------------

SKU VARCHAR2(100)

ITEM\_TYPE VARCHAR2(30)

BRAND VARCHAR2(5)

MERCH\_DEPARTMENT VARCHAR2(50)

MERCH\_DEPT\_NAME VARCHAR2(10)

MERCH\_COLLECTION VARCHAR2(50)

MERCH\_COLLECTION\_NAME VARCHAR2(50)

MERCH\_CLASS VARCHAR2(50)

MERCH\_CLASS\_NAME VARCHAR2(5)

MERCH\_SUBCLASS VARCHAR2(50)

MERCH\_SUBCLASS\_NAME VARCHAR2(10)

ITEM\_SKEY NUMBER

SQL> insert into item\_dim\_flat select i.sku, i.type\_name, i.brand\_name,

2 id.description\_d, id.item\_merch\_department\_na,

3 icl.description\_cl, icl.item\_merch\_collection\_na,

4 ic.description\_c, ic.item\_merch\_class\_name,

5 isc.description\_s, isc.item\_merch\_subclass\_name,

6 i.kpi\_dw\_skey

7 from item\_dim i join item\_merch\_department\_dim id on i.kpi\_dw\_skey=id.kpi\_dw\_skey

8 join item\_merch\_collection\_dim icl on id.kpi\_dw\_skey=icl.kpi\_dw\_skey

9 join item\_merch\_class\_dim ic on icl.kpi\_dw\_skey=ic.kpi\_dw\_skey

10 join item\_merch\_subclass\_dim isc on ic.kpi\_dw\_skey=isc.kpi\_dw\_skey;

78 rows created.

**12) Questions on transaction\_type:**

1) Find the Top 5 and Bottom 5 Items based on the Demand Amount values in a single query

SQL> select \* from (select \* from

2 (select distinct(t.transaction\_type), s.amount

3 from transactions t

4 join transaction\_line s on t.transaction\_id=s.transaction\_id

5 where t.transaction\_type='Sales Order'

6 group by t.transaction\_type, s.amount

7 order by s.amount desc)

8 where rownum<=5) top5, (select \* from

9 (select distinct(t.transaction\_type), s.amount

10 from transactions t

11 join transaction\_line s on t.transaction\_id=s.transaction\_id

12 where t.transaction\_type='Sales Order'

13 group by t.transaction\_type, s.amount

14 order by s.amount)

15 where rownum<=5) bottom5;

TRANSACTION\_TYPE AMOUNT TRANSACTION\_TYPE AMOUNT

-------------------- ---------- -------------------- ----------

Sales Order 1707 Sales Order 0

Sales Order 1707 Sales Order 2

Sales Order 1707 Sales Order 6

Sales Order 1707 Sales Order 12

Sales Order 1707 Sales Order 24

Sales Order 1443 Sales Order 0

Sales Order 1443 Sales Order 2

Sales Order 1443 Sales Order 6

Sales Order 1443 Sales Order 12

Sales Order 1443 Sales Order 24

Sales Order 1199 Sales Order 0

Sales Order 1199 Sales Order 2

Sales Order 1199 Sales Order 6

Sales Order 1199 Sales Order 12

Sales Order 1199 Sales Order 24

Sales Order 1003 Sales Order 0

Sales Order 1003 Sales Order 2

Sales Order 1003 Sales Order 6

Sales Order 1003 Sales Order 12

Sales Order 1003 Sales Order 24

Sales Order 947 Sales Order 0

Sales Order 947 Sales Order 2

Sales Order 947 Sales Order 6

Sales Order 947 Sales Order 12

Sales Order 947 Sales Order 24

25 rows selected.

2) Which Department has the highest Demand and Sales Amount

SQL> select distinct(d.name) from departments d join transaction\_line t on d.department\_id=t.department\_id join transactions s on s.transaction\_id=t.transaction\_id group by s.transaction\_type, d.name having max(t.amount) in (select max(t.amount) from transaction\_line t);

NAME

--------------------------------------------------

7355

4) Populate top 10 LOCATIONS based on number of Demand Transactions using Analytical functions

SQL> select \* from(select distinct(l.city) from location\_dim l join transaction\_line\_fact t on l.kpi\_dw\_skey=t.kpi\_dw\_skey where transaction\_type='Sales Order') city where rownum<=10;

CITY

--------------------------------------------------

BROSSARD

Bondi Junction

Brisbane

CALGARY

Chadstone

Chatswood

Corby

EDMONTON

ETOBICOKE

Essendon Fields

10 rows selected.

5) Find Demand Amount, Demand Units, Sales Amount and Sales Units for each Channel

SQL> select transaction\_type, sum(amount) as amount, sum(units) as units from transaction\_line\_fact group by transaction\_type;

TRANSACTION\_TYPE AMOUNT UNITS

-------------------------------------------- ---------- ----------

Invoice 14967 487

Sales Order 56942 1463

6) Write a VIEW using target tables

create force view target\_view as select t.transaction\_id, t.transaction\_line\_id, t.trandate, t.transaction\_type,

i.type\_name,

l.city,

d.name,

cd.list\_item\_name,

id.item\_merch\_department\_na,

id.description,

ic.item\_merch\_collection\_na,

ic.description,

c.item\_merch\_class\_name,

c.description,

s.item\_merch\_subclass\_name,

s.description,

t.amount,

t.units

from transaction\_line\_fact t join item\_dim i on t.kpi\_dw\_skey = i.kpi\_dw\_skey

join location\_dim l on i.kpi\_dw\_skey = l.kpi\_dw\_skey

join department\_dim d on l.kpi\_dw\_skey = d.kpi\_dw\_skey

join channel\_dim cd on d.kpi\_dw\_skey = cd.kpi\_dw\_skey

join item\_merch\_department\_dim id on cd.kpi\_dw\_skey = id.kpi\_dw\_skey

join item\_merch\_collection\_dim ic on id.kpi\_dw\_skey = ic.kpi\_dw\_skey

join item\_merch\_class\_dim c on ic.kpi\_dw\_skey = c.kpi\_dw\_skey

join item\_merch\_subclass\_dim s on c.kpi\_dw\_skey = s.kpi\_dw\_skey;