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.A Snowflake model is a logical arrangement of tables in a multidimensional 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:

```
--CHANNEL

CREATE TABLE CHANNEL(

DATE_CREATED DATE,

IS_RECORD_INACTIVE CHAR,

LAST_MODIFIED_DATE DATE,

LIST_ID NUMBER(20),
```

```
LIST_ITEM_NAME VARCHAR(20)
);
--TRANSACTIONS
 CREATE TABLE TRANSACTIONS(
TRANSACTION_ID NUMBER(20,0),
TRANID VARCHAR(30),
TRANSACTION_TYPE VARCHAR(50),
TRANDATE DATE,
CHANNEL_ID NUMBER(20,0)
);
--LOCATIONS
CREATE TABLE LOCATIONS(
LOCATION_ID NUMBER(20,0),
ADDRESS    VARCHAR(150),
CITY VARCHAR(50),
COUNTRY VARCHAR(50),
DATE_LAST_MODIFIED DATE,
FULL_NAME VARCHAR(60),
ISINACTIVE VARCHAR(5),
NAME VARCHAR(50)
);
--TRANSACTION_LINES
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)
);
-- DEPARTMENTS
CREATE TABLE  DEPARTMENTS(
```

```
DATE_LAST_MODIFIED DATE,
DEPARTMENT_ID NUMBER(20,0),
ISINACTIVE VARCHAR(5),
NAME VARCHAR(20),
WS_DESCRIPTION VARCHAR(50)
);
--ITEMS
CREATE TABLE ITEMS(
ITEM_ID NUMBER(20,0),
SKU VARCHAR(100),
TYPE_NAME VARCHAR(30),
SALESDESCRIPTION VARCHAR(100),
CLASS_ID NUMBER(20,0),
WS_MERCHANDISE_DEPARTMENT_ID NUMBER(20,0),
WS_MERCHANDISE_COLLECTION_ID NUMBER(20,0),
WS_MERCHANDISE_CLASS_ID      NUMBER(20,0),
WS_MERCHANDISE_SUBCLASS_ID   NUMBER(20,0)
);
--CLASSES
CREATE TABLE CLASSES(
CLASS_ID NUMBER(20,0),
DATE_LAST_MODIFIED DATE,
FULL_NAME VARCHAR(30),
ISINACTIVE VARCHAR(5),
NAME       VARCHAR(5)
);

--ITEM_MERCHANDISE_DEPTARMENT
CREATE TABLE ITEM_MERCHANDISE_DEPTARMENT(
ITEM_MERCHANDISE_DEPARTMENT_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
ITEM_MERCHANDISE_DEPARTMENT_NA VARCHAR(10)
);
--ITEM_MERCHANDISE_COLLECTION
```

```
CREATE TABLE ITEM_MERCHANDISE_COLLECTION(
ITEM_MERCHANDISE_COLLECTION_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
ITEM_MERCHANDISE_COLLECTION_NA VARCHAR(50)
);
--ITEM_MERCHANDISE_CLASS
CREATE TABLE ITEM_MERCHANDISE_CLASS(
ITEM_MERCHANDISE_CLASS_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
ITEM_MERCHANDISE_CLASS_NAME VARCHAR(5)
);
--ITEM_MERCHANDISE_SUBCLASS
CREATE TABLE ITEM_MERCHANDISE_SUBCLASS(
ITEM_MERCHANDISE_SUBCLASS_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
ITEM_MERCHANDISE_SUBCLASS_NAME VARCHAR(10)
);
```

4) Analyze keys:

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:

--CREATING CHANNEL TABLE

CREATE TABLE CHANNEL_DIM(

DATE_CREATED DATE,

```
IS_RECORD_INACTIVE CHAR,
LAST_MODIFIED_DATE DATE,
LIST_ID NUMBER(20,0),
LIST_ITEM_NAME VARCHAR(20),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);

--LOCATION_DIM
CREATE TABLE LOCATION_DIM(
LOCATION_ID NUMBER(20,0),
ADDRESS    VARCHAR(150),
CITY VARCHAR(50),
COUNTRY VARCHAR(50),
DATE_LAST_MODIFIED DATE,
FULL_NAME VARCHAR(60),
ISINACTIVE VARCHAR(5),
NAME VARCHAR(50),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);


--TRANSACTION_LINE_FACT
CREATE TABLE TRANSACTION_LINE_FACT(
TRANSACTION_ID NUMBER(20,0),
TRANSACTION_LINE_ID NUMBER(20,0),
```

```
TRANID VARCHAR(30),
TRANSACTION_TYPE VARCHAR(50),
TRANDATE DATE ,
KPI_CHANNEL_SKEY NUMBER(20,0),
KPI_LOCATION_SKEY NUMBER(20,0),
KPI_DEPARTMENT_SKEY NUMBER(20,0),
KPI_ITEM_SKEY        NUMBER(20,0),
AMOUNT NUMBER(8,2),
COST NUMBER(8,2),
UNITS NUMBER(5,0),
KPI_DW_SKEY NUMBER(20,0)
);

--DEPARTMENT_DIM
CREATE TABLE DEPARTMENT_DIM(
DATE_LAST_MODIFIED DATE,
DEPARTMENT_ID NUMBER(20,0),
ISINACTIVE VARCHAR(5),
NAME VARCHAR(10),
WS_DESCRIPTION VARCHAR(50),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);

--ITEM_DIM
CREATE TABLE ITEM_DIM (
ITEM_ID NUMBER(20,0),
SKU VARCHAR(100),
```

```
TYPE_NAME VARCHAR(30),
SALESDESCRIPTION VARCHAR(100),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE,
KPI_CLASS_SKEY NUMBER(20,0),
WS_MERCHANDISE_DEPARTMENT_SKEY NUMBER(20,0),
WS_MERCHANDISE_COLLECTION_SKEY NUMBER(20,0),
WS_MERCHANDISE_CLASS_SKEY      NUMBER(20,0),
WS_MERCHANDISE_SUBCLASS_SKEY   NUMBER(20,0)
);

--CLASS_DIM
CREATE TABLE CLASS_DIM(
CLASS_ID NUMBER(20,0),
DATE_LAST_MODIFIED DATE,
FULL_NAME VARCHAR(30),
ISINACTIVE VARCHAR(5),
NAME VARCHAR(5),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);

--KPI_ITEM_MERCHANDISE_DEPTARMENT_DIM

CREATE TABLE MERCHANDISE_DEPTARMENT_DIM (
ITEM_MERCHANDISE_DEPARTMENT_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
```

```
ITEM_MERCHANDISE_DEPARTMENT_NA VARCHAR(10),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);

--ITEM_MERCHANDISE_COLLECTION_DIM
CREATE TABLE MERCHANDISE_COLLECTION_DIM (
ITEM_MERCHANDISE_COLLECTION_ID NUMBER(20,0),
DESCRIPTION  VARCHAR(50),
ITEM_MERCHANDISE_COLLECTION_NA VARCHAR(50),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);
--KPI_ITEM_MERCHANDISE_CLASS_DIM
CREATE TABLE MERCHANDISE_CLASS_DIM(
ITEM_MERCHANDISE_CLASS_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
ITEM_MERCHANDISE_CLASS_NAME VARCHAR(5),
KPI_DW_SKEY NUMBER(20,0),
KPI_DW_INSERT_DATE DATE,
KPI_DW_UPDATE_DATE DATE
);

--KPI_TEM_MERCHANDISE_SUBCLASS_DIM
CREATE TABLE MERCHANDISE_SUBCLASS_DIM (
ITEM_MERCHANDISE_SUBCLASS_ID NUMBER(20,0),
DESCRIPTION VARCHAR(50),
```

```
ITEM_MERCHANDISE_SUBCLASS_NAME VARCHAR(10),

KPI_DW_SKEY  NUMBER(20,0),

KPI_DW_INSERT_DATE DATE,

KPI_DW_UPDATE_DATE DATE

);
```

9) Load data:

```
--CHANNEL_DIM

INSERT INTO
CHANNEL_DIM(DATE_CREATED,IS_RECORD_INACTIVE,LAST_MODIFIED_DATE,LIST_ID,
LIST_ITEM_NAME)(SELECT * FROM HII.CHANNEL);



--LOCATION_DIM

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:

```
--CREATE BRAND_NAME field in KPI_ITEM_DIM and populate values from NAME field present in KPI_CLASS_DIM

--1. Provide the script to add the new column

--2. Provide the UPDATE script to populate BRAND_NAME field


ALTER TABLE item_dim

ADD BRAND_NAME VARCHAR2(5) ;


UPDATE item_dim

SET BRAND_NAME =(SELECT NAME FROM class_dim WHERE class_dim.KPI_DW_SKEY=item_dim.KPI_DW_SKEY);
```

11) Create kpi_item_dim_flat:

```
CREATE TABLE KPI_ITEM_DIM_FLAT(

SKU VARCHAR(100),

ITEM_TYPE VARCHAR(30),

BRAND   VARCHAR2(5),

MERCHANDISE_DEPARTMENT VARCHAR(50),

MERCHANDISE_DEPT_NAME  VARCHAR(10),

MERCHANDISE_COLLECTION VARCHAR(50),

MERCHANDISE_COLLECTION_NAME VARCHAR(50),

MERCHANDISE_CLASS VARCHAR(50),

MERCHANDISE_CLASS_NAME VARCHAR(5),

MERCHANDISE_SUBCLASS VARCHAR(50),

MERCHANDISE_SUBCLASS_NAME VARCHAR(10),

KPI_ITEM_SKEY  NUMBER(20,0)

);


INSERT INTO KPI_ITEM_DIM_FLAT(SELECT
ID.SKU,ID.TYPE_NAME,ID.BRAND_NAME,IMDD.DESCRIPTION,IMDD.ITEM_MERCHANDIS
E_DEPARTMENT_NA,

IMCD.DESCRIPTION,IMCD.ITEM_MERCHANDISE_COLLECTION_NA,IMC.DESCRIPTION,

IMC.ITEM_MERCHANDISE_CLASS_NAME,IMSD.DESCRIPTION,IMSD.ITEM_MERCHANDIS
E_SUBCLASS_NAME,
```

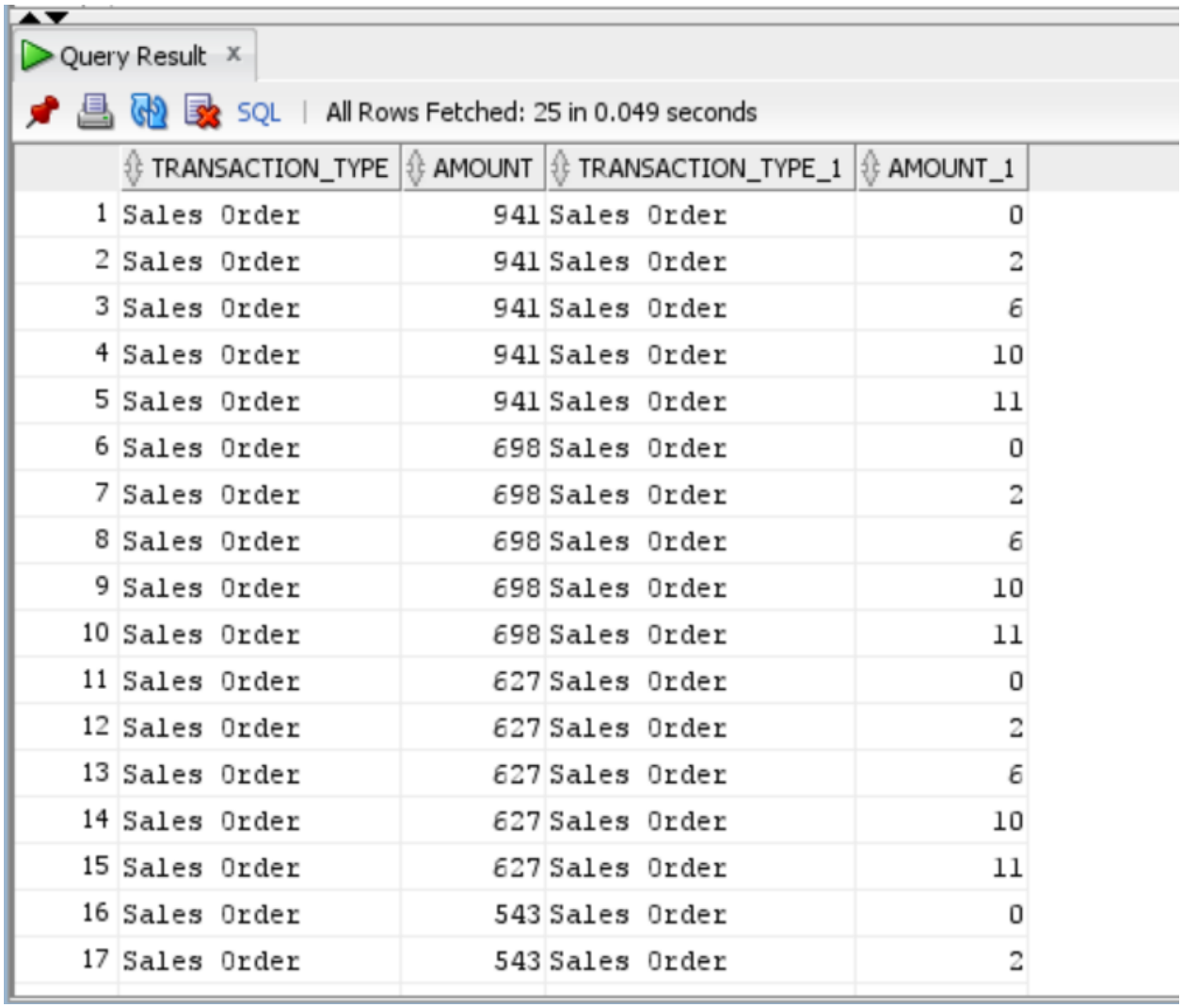
```
ID.KPI_DW_SKEY

FROM ITEM_DIM ID

JOIN CLASS_DIM  CD ON ID.KPI_CLASS_SKEY=CD.KPI_DW_SKEY

JOIN ITEM_MERCHANDISE_DEPTARMENT_DIM IMDD ON
IMDD.KPI_DW_SKEY=ID.WS_MERCHANDISE_DEPARTMENT_SKEY

JOIN ITEM_MERCHANDISE_COLLECTION_DIM  IMCD ON
IMCD.KPI_DW_SKEY=ID.WS_MERCHANDISE_COLLECTION_SKEY

JOIN ITEM_MERCHANDISE_CLASS_DIM IMC ON
IMC.KPI_DW_SKEY=ID.WS_MERCHANDISE_CLASS_SKEY

JOIN ITEM_MERCHANDISE_SUBCLASS_DIM IMSD ON
IMSD.KPI_DW_SKEY=ID.WS_MERCHANDISE_SUBCLASS_SKEY

);
```

12) <u>Questions on transaction_type</u>:

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

```
select * from (select * from(select distinct(t.transaction_type), s.amount

from transactions t join transaction_lines s on t.transaction_id=s.transaction_id where
t.transaction_type='Sales Order'

 group by t.transaction_type, s.amount

 order by s.amount desc)

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

 (select distinct(t.transaction_type), s.amount

 from transactions t

 join transaction_lines s on t.transaction_id=s.transaction_id

 where t.transaction_type='Sales Order'

 group by t.transaction_type, s.amount

 order by s.amount) where rownum<=5) bottom5;
```

OUTPUT:-

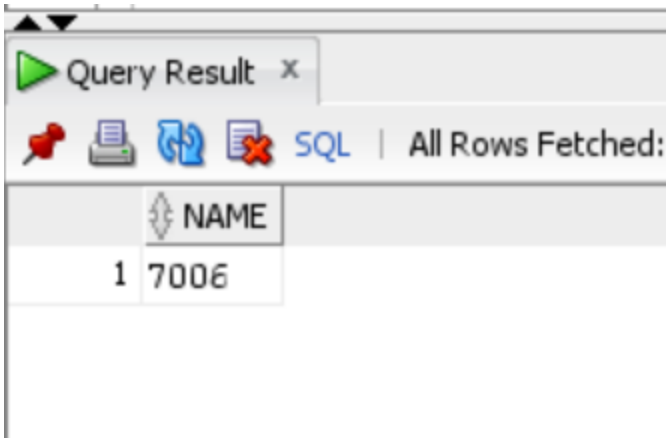

Query Result

SQL | All Rows Fetched: 25 in 0.049 seconds

	TRANSACTION_TYPE	AMOUNT	TRANSACTION_TYPE_1	AMOUNT_1
1	Sales Order	941	Sales Order	0
2	Sales Order	941	Sales Order	2
3	Sales Order	941	Sales Order	6
4	Sales Order	941	Sales Order	10
5	Sales Order	941	Sales Order	11
6	Sales Order	698	Sales Order	0
7	Sales Order	698	Sales Order	2
8	Sales Order	698	Sales Order	6
9	Sales Order	698	Sales Order	10
10	Sales Order	698	Sales Order	11
11	Sales Order	627	Sales Order	0
12	Sales Order	627	Sales Order	2
13	Sales Order	627	Sales Order	6
14	Sales Order	627	Sales Order	10
15	Sales Order	627	Sales Order	11
16	Sales Order	543	Sales Order	0
17	Sales Order	543	Sales Order	2

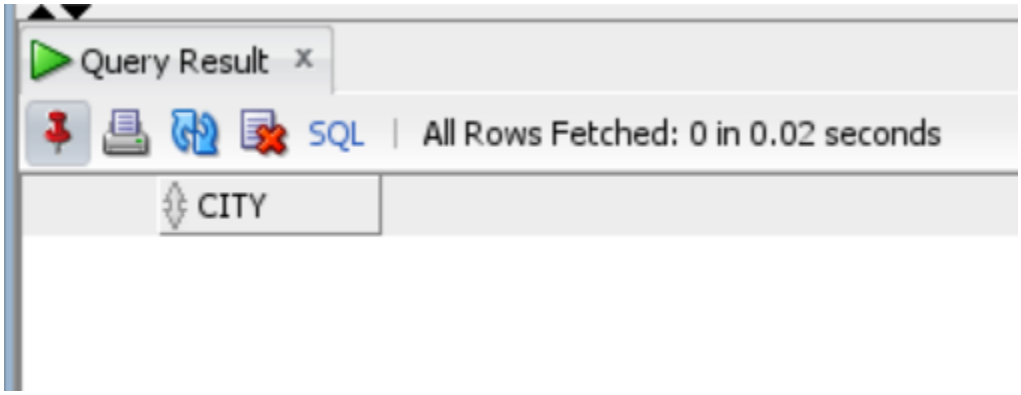
--2) Which Department has the highest Demand and Sales Amount

```
select distinct(d.name) from departments d

join transaction_lines 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_lines t);
```



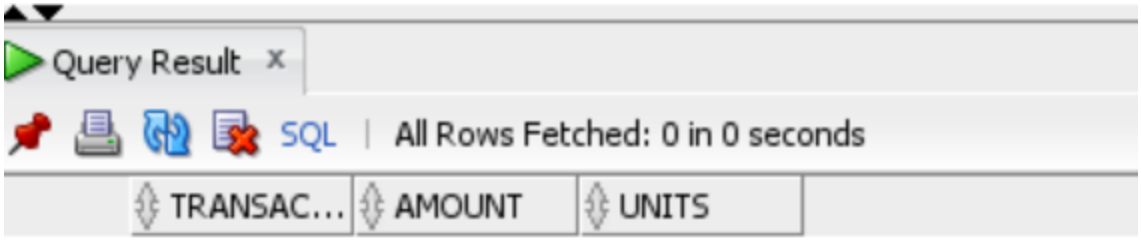
--4) Populate top 10 LOCATIONS based on number of Demand Transactions using Analytical functions

```
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;
```



--5.Find Demand Amount, Demand Units, Sales Amount and Sales Units for each Channel

```
select transaction_type, sum(amount) as amount, sum(units) as units

from transaction_line_fact

group by transaction_type;
```



--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;
```