

## ASSIGNEMENT

### 1. Understand the commonly used Data Models to build DWH

#### Identify the given data model and briefly explain about it.

The given data model is in the form of snowflake schema.

##### Snowflake Schema:

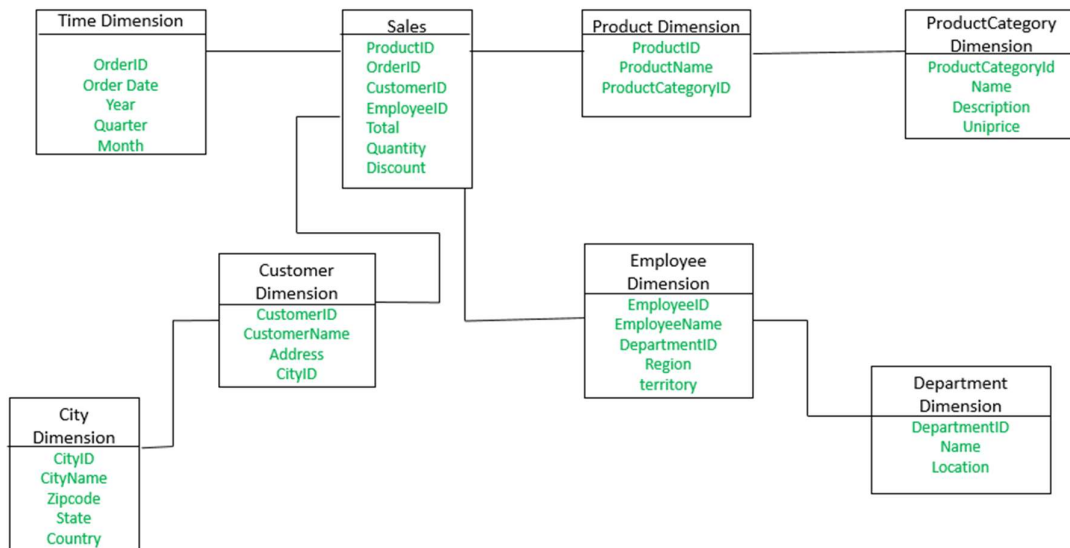
The snowflake schema is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions.

In the snowflake schema, dimensions are present in a normalized form in multiple related tables.

The snowflake structure materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship, and the child tables have multiple parent tables.

The snowflake effect affects only the dimension tables and does not affect the fact tables.

#### Example:



The **Employee** dimension table now contains the attributes: Employee ID, Employee Name, Department ID, Region and Territory. The Department ID attribute links with the **Employee** table with the **Department** dimension table.

The **Department** dimension is used to provide detail about each department, such as the Name and Location of the department. The **Customer** dimension table now contains the attributes: Customer ID, Customer Name, Address, City ID. The City ID attributes link the **Customer** dimension table with the **City** dimension table.

The **City** dimension table has details about each city such as City Name, Zip code, State, and Country.

## **2. Understand how to set the dependencies during Stage tables and Target Tables load**

1. First we have to take source data from the table.
2. We have to set primary keys.
3. Remove all redundancy data from the table and update foreign keys.
4. Then, load the data .
5. This process is called etl process.

## **3. What are common issues with this model?**

### **Advantages:**

1. Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

### **Disadvantage:**

1. Harder to design compared to a star schema.
2. The primary disadvantage of the snowflake schema is the additional maintenance efforts required due to the increasing number of lookup tables.

It is also known as a multi fact star schema.

3. More tables more join so more query execution time.

## **4. Are there any options to convert this model to START? If SO, how ?**

1. Yes the given model can be converted to snowflake model to star model.
2. The snowflake model is an extension of a star model.
3. Snow flaking is a method of normalizing the dimension tables in a STAR model.

4. When we normalize all the dimension tables entirely, the resultant structure resembles a snowflake with the fact table in the middle.

## **2. Create Stage Tables**

### **1. Provide all the CREATE statements**

#### **KPI\_STG\_CHANNEL**

```
CREATE TABLE KPI_STG_CHANNEL(  
    DATE_CREATED DATE,IS_RECORD_INACTIVE VARCHAR2(10),  
    LAST_MODIFIED_DATE DATE,LIST_ID NUMBER,  
    LIST_ITEM_NAME VARCHAR2(20)  
);  
  
SELECT * FROM KPI_STG_CHANNEL;
```

#### **KPI\_STG\_TRANSACTIONS**

```
CREATE TABLE KPI_STG_TRANSACTIONS (  
    TRANSACTION_ID NUMBER,TRANID NUMBER,  
    TRANSACTION_TYPE VARCHAR2(50), TRANDATE DATE,  
    CHANNEL_ID NUMBER  
);  
  
SELECT * FROM KPI_STG_TRANSACTIONS;
```

#### **KPI\_STG\_ITEMS**

```
CREATE TABLE KPI_STG_ITEMS (  
    ITEM_ID NUMBER, SKU VARCHAR2(100),  
    TYPE_NAME VARCHAR2(30),SALESDESCRIPTION VARCHAR2(100),  
    CLASS_ID NUMBER,  
    WS_MERCHANDISE_DEPARTMENT_ID NUMBER,  
    WS_MERCHANDISE_COLLECTION_ID NUMBER,  
    WS_MERCHANDISE_CLASS_ID NUMBER,
```

```
    WS_MERCHANDISE_SUBCLASS_ID NUMBER  
);
```

```
SELECT * FROM KPI_STG_ITEMS;
```

### **KPI\_STG\_DEPARTMENTS**

```
CREATE TABLE KPI_STG_DEPARTMENTS (  
    DATE_LAST_MODIFIED DATE, DEPARTMENT_ID NUMBER,  
    ISINACTIVE VARCHAR2(5), NAME VARCHAR2(50),  
    WS_DESCRIPTION VARCHAR2(50)  
);
```

```
SELECT * FROM KPI_STG_DEPARTMENTS;
```

### **KPI\_STG\_LOCATIONS**

```
CREATE TABLE KPI_STG_LOCATIONS (  
    LOCATION_ID NUMBER, ADDRESS VARCHAR2(120),  
    CITY VARCHAR2(50), COUNTRY VARCHAR2(50),  
    DATE_LAST_MODIFIED DATE, FULL_NAME VARCHAR2(60),  
    ISINACTIVE VARCHAR2(5), NAME VARCHAR2(50)  
);
```

```
SELECT * FROM KPI_STG_LOCATIONS;
```

### **KPI\_STG\_CLASSES**

```
CREATE TABLE KPI_STG_CLASSES (  
    CLASS_ID NUMBER, DATE_LAST_MODIFIED DATE,  
    FULL_NAME VARCHAR2(30), ISINACTIVE VARCHAR2(5),  
    NAME VARCHAR2(5)  
);
```

```
SELECT * FROM KPI_STG_CLASSES;
```

### **KPI\_STG\_TRANSACTIONS\_LINES**

```
CREATE TABLE KPI_STG_TRANSACTIONS_LINES (  
    TRANSACTION_ID NUMBER, TRANSACTION_LINE_ID NUMBER,  
    LOCATION_ID NUMBER, DEPARTMENT_ID NUMBER,  
    ITEM_ID NUMBER, AMOUNT NUMBER,
```

```

        COST NUMBER, UNITS NUMBER
    );

SELECT * FROM KPI_STG_TRANSACTIONS_LINES;

KPI_STG_ITEM_MERCHANDISE_DEPAR

CREATE TABLE KPI_STG_ITEM_MERCHANDISE_DEPAR (
    ITEM_MERCHANDISE_DEPARTMENT_ID NUMBER,
    DESCRIPTION VARCHAR2(20),
    ITEM_MERCHANDISE_DEPARTMENT_NA VARCHAR2(10)
);

SELECT * FROM KPI_STG_ITEM_MERCHANDISE_DEPAR;

KPI_STG_ITEM_MERCHANDISE_COLLE

CREATE TABLE KPI_STG_ITEM_MERCHANDISE_COLLE (
    ITEM_MERCHANDISE_COLLECTION_ID NUMBER,
    DESCRIPTION VARCHAR2(50),
    ITEM_MERCHANDISE_COLLECTION_NA VARCHAR2(50)
);

KPI_STG_ITEM_MERCHANDISE_SUBCL

CREATE TABLE KPI_STG_ITEM_MERCHANDISE_SUBCL (
    ITEM_MERCHANDISE_SUBCLASS_ID NUMBER,
    DESCRIPTION VARCHAR2(50),
    ITEM_MERCHANDISE_SUBCLASS_NAME VARCHAR2(10)
);

SELECT * FROM KPI_STG_ITEM_MERCHANDISE_SUBCL;

KPI_STG_ITEM_MERCHANDISE_CLASS

CREATE TABLE KPI_STG_ITEM_MERCHANDISE_CLASS (
    ITEM_MERCHANDISE_CLASS_ID NUMBER,
    DESCRIPTION VARCHAR2(50),
    ITEM_MERCHANDISE_CLASS_NAME VARCHAR2(5)
);

SELECT * FROM KPI_STG_ITEM_MERCHANDISE_CLASS;

```

### 3.Load the data in the tables

#### 1.Provide the INSERT Scripts

##### KPI\_STG\_CHANNEL

```
insert into KPI_STG_CHANNEL
values(to_date('2012/12/18','yyyy/mm/dd'),'F',to_date('2013/04/30','yyyy/mm/dd'),1,'Retail');
insert into KPI_STG_CHANNEL
values(to_date('2012/12/18','yyyy/mm/dd'),'F',to_date('2013/04/30','yyyy/mm/dd'),2,'DTC');
insert into KPI_STG_CHANNEL
values(to_date('2013/04/30','yyyy/mm/dd'),'F',to_date('2013/04/30','yyyy/mm/dd'),3, 'Care
Center');
insert into KPI_STG_CHANNEL
values(to_date('2013/05/07','yyyy/mm/dd'),'F',to_date('2013/05/07','yyyy/mm/dd'),4,'RTC');
insert into KPI_STG_CHANNEL
values(to_date('2015/08/06','yyyy/mm/dd'),'F',to_date('2015/08/14','yyyy/mm/dd'),5,'Wholesale');
```

##### KPI\_STG\_TRANSACTIONS

```
insert into KPI_STG_TRANSACTIONS values(185339066, 2186178,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185339085, 2186192,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185339701, 2186202,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185340234, 2186227,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185341664, 2186252,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185343047, 2186316,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185343053, 2186320,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185343282, 2186341,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185346146, 2186455,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
insert into KPI_STG_TRANSACTIONS values(185346454, 2186460,      'Sales Order',
to_date('2021/09/01','yyyy/mm/dd'), 2);
```

## KPI\_STG\_DEPARTMENTS

```
insert into KPI_STG_DEPARTMENTS values(to_date('2015/09/25','yyyy/mm/dd'), 1, 'No', 7001, 'Store
WS NSW, Bondi Junction, 2/13(7001)');
insert into KPI_STG_DEPARTMENTS values(to_date('2020/11/11','yyyy/mm/dd'), 2, 'No', 7002, 'Store
PB NSW, Bondi Junction, 2/13(7002)');
insert into KPI_STG_DEPARTMENTS values(to_date('2020/11/11','yyyy/mm/dd'), 3, 'No', 7003, 'Store
PK NSW, Bondi Junction, 2/13 (7003)');
insert into KPI_STG_DEPARTMENTS values(to_date('2015/09/25','yyyy/mm/dd'), 4, 'No', 7004, 'Store
WE NSW, Bondi Junction, 2/13 (7004)');
insert into KPI_STG_DEPARTMENTS values(to_date('2012/12/18','yyyy/mm/dd'), 5, 'Yes',7211,
'NULL');
insert into KPI_STG_DEPARTMENTS values(to_date('2012/12/18','yyyy/mm/dd'), 11,'Yes', 'AUS Corp
Misc', 'NULL');
insert into KPI_STG_DEPARTMENTS values(to_date('2012/12/18','yyyy/mm/dd'),
12,'Yes','2012DC/Ops- RTL','NULL');
insert into KPI_STG_DEPARTMENTS values(to_date('2012/12/18','yyyy/mm/dd'), 15,'Yes','DC/Ops-
DTC (tbd)','NULL');
insert into KPI_STG_DEPARTMENTS values(to_date('2012/12/18','yyyy/mm/dd'), 16,'Yes','Legal
Entity (tbd)','NULL');
insert into KPI_STG_DEPARTMENTS values(to_date('2013/07/31','yyyy/mm/dd'), 20,'No',
7111, 'WS Singapore LE - Global Purchases');
```

## KPI\_STG\_ITEMS

```
Insert into KPI_STG_ITEMS values(11068456, 5732022, 'Non-inventory Item','Andes UK Sectinal Set
02:RA 2.5 Str Sfa/Corner/Ottm Poly Performance Velvet Petrol DP', 1 , 47 , 408305 , 101 , 434 );
Insert into KPI_STG_ITEMS values(11086902, 6325288,'Non-inventory Item','Harlow Convertible Crib
Antique Gray DELUXE',
5 ,32, 197904,283, 52803);
Insert into KPI_STG_ITEMS values(11114043, 1458567,'Non-inventory Item','Tanner Round 44 inch
Dining Table', 1 , 20 , 1986806, 205, 52302);
Insert into KPI_STG_ITEMS values(163 , 18143,'Inventory Item','Flameless Candle4 inchesIvory' , 4,
28 , 1930706, 301, 485);
Insert into KPI_STG_ITEMS values(164, 18150,'Inventory Item','Flameless Candle6 inchesIvory',4 , 28
, 1930706, 301, 485);
Insert into KPI_STG_ITEMS values(218, 111518, 'Inventory Item','PB Essentials 300TC Fitted
SheetQueenWhite',4 , 4, 641210,4 , 2 );
Insert into KPI_STG_ITEMS values(223, 111914, 'Inventory Item','PB Essentials 300TC
ShamsEuroWhite', 4 , 4 , 123, 74 , 126 );
Insert into KPI_STG_ITEMS values(224, 111930, 'Inventory Item','PB Essentials 300TC
ShamsStandardWhite',4 , 4 , 123 ,74 , 106);
Insert into KPI_STG_ITEMS values( 226, 111989,'Inventory Item','PB Essential 300TC Pillowcase
S/2KingWhite', 4 , 4 , 4 ,4 , 2);
```

Insert into KPI\_STG\_ITEMS values(229, 115162, 'Inventory Item', 'Santino Pitcher', 4, 58, 363107, 120, 3613);

#### **KPI\_STG\_TRANSACTIONS\_LINES**

insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 1, 383, 28, 9918508, 31, 0, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 2, 383, 28, 3507200, 56, -20, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 3, 383, 28, 1406935, 31, -12, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 4, 383, 28, 9222, 56, -28, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 5, 383, 28, 2046731, 28, -16, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 6, 383, 28, 919828, 153, -73, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339085, 1, 383, 28, 962429, 22, -12, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339085, 2, 383, 28, 6066781, 9, -5, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339066, 3, 383, 28, 9222, 56, -28, 1);  
insert into KPI\_STG\_TRANSACTIONS\_LINES values(185339701, 1, 383, 28, 7965554, 125, -58, 1);

#### **KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION**

INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(4, 'PB ESSENTIALS BEDDING', 'PB1015');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(5, 'MODERN WIRE COLLECTION', 'MODERN WIRE COLLECTION');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(6, 'WE NEW LINEN COTTON GROMMET CURTAIN', 'WE7078');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(7, 'WE BULLS EYE PILLOW COVER', 'WE3386');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(8, 'PB HARRISON', 'PB159');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(9, 'PB COLTON WOVEN TRUNK', 'PB8217');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(10, 'PK CHAMOIS STRLR', 'PK133');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(11, 'PB CADEN', 'PB3680');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(12, 'PK CPC CHAMOIS', 'PK9157');  
INSERT INTO KPI\_STG\_ITEM\_MERCHANDISE\_COLLECTION VALUES(13, 'PB REBECCA', 'PB816');



### **KPI\_STG\_ITEM\_MERCHANDISE\_CLASS**

```
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(4,'SHEETS',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(5,'WILLIAMS SONOMA',69);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(6,'SOLID CURTAINS',7);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(7,'VINEGARS',2);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(8,'PATTERN + STRIPE PLW',3);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(9,'BASKETS AND STORAGE',4);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(10,'BLANKETS',6);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(11,'ACCENTS AND OTTOMANS',8);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(12,'CHANGING PADS',10);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_CLASS VALUES(13,'NURSERY WRAPS',7);
```

### **KPI\_STG\_ITEM\_MERCHANDISE\_SUBCLASS**

```
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(4,'LIGHT FILTERING',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(5,'BALSAMIC',3);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(6,'UNASSIGNED',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(7,'WOVEN',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(8,'ICON',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(9,'STOOLS',1);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(10,'SOLID COVERS',2);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(11,'DO NOT USE',4);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(12,'NURSERY WRAPS',5);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_SUBCLASS VALUES(13,'STOCKED ',1)
```

### **KPI\_STG\_CLASSES**

```
INSERT INTO KPI_STG_CLASSES VALUES(1, TO_DATE('2018-02-13','YYYY-MM-DD'), 'WE','No', 'WE');
INSERT INTO KPI_STG_CLASSES VALUES(3, TO_DATE('2013-06-13','YYYY-MM-DD'), 'PT','No', 'PT');
INSERT INTO KPI_STG_CLASSES VALUES(4, TO_DATE('2013-06-13','YYYY-MM-DD'), 'PB','No', 'PB');
INSERT INTO KPI_STG_CLASSES VALUES(5, TO_DATE('2013-06-13','YYYY-MM-DD'), 'PK','No', 'PK');
INSERT INTO KPI_STG_CLASSES VALUES(6, TO_DATE('2013-06-13','YYYY-MM-DD'), 'WS','No', 'WS');
INSERT INTO KPI_STG_CLASSES VALUES(7, TO_DATE('2014-04-18','YYYY-MM-DD'), 'DC','No', 'DC');
```

### **KPI\_STG\_ITEM\_MERCHANDISE\_DEPARTMENT**

```
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(4, 'PB BEDDING', 203);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(5, 'WS CUTLERY', 105);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(6, 'WE WINDOW', 808);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(7, 'WS SAVORY FOOD', 108);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(8, 'WE PILLOWS', 810);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(9, 'PB FUNC ACC', 221);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(10, 'PK NURSERY BEDDING',
918);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(11, 'PB OC/MEDIA FURNTURE',
201);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(12, 'PK BATH', 910);
INSERT INTO KPI_STG_ITEM_MERCHANDISE_DEPARTMENT VALUES(13, 'PK RUGS', 902);
```

## KPI\_STG\_LOCATIONS

```
INSERT INTO KPI_STG_LOCATIONS VALUES(2,'Singapore', 'NULL', 'SG', TO_DATE('2017-08-07','YYYY-MM-DD'), 'Test Location', 'YES', 'Test Location');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(3,'Singapore', 'NULL', 'SG', TO_DATE('2017-08-07','YYYY-MM-DD'), 'Test Location 2', 'YES', 'Test Location 2');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(4,'Australia', 'NULL', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'), 'Test Location 4', 'YES', 'Test Location 4');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(5,'07001 - WS NSW, Bondi Junction 472 Oxford Street Bondi Junction NSW 2022 Australia', 'Bondi Junction', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'),'D07001 - WS NSW, Bondi Junction', 'YES', 'D07001 - WS NSW, Bondi Junction');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(6,'07002 - PB NSW, Bondi Junction 470 Oxford Street Bondi Junction NSW 2022 Australia', 'Bondi Junction', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'), 'D07002 - PB NSW, Bondi Junction', 'YES', 'D07002 - PB NSW, Bondi Junction');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(7,'07003 - PK NSW, Bondi Junction 468 Oxford Street Bondi Junction NSW 2022 Australia', 'Bondi Junction', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'), 'D07003 - PK NSW, Bondi Junction', 'YES', 'D07003 - PK NSW, Bondi Junction');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(8,'07004 - WE NSW, Bondi Junction Bondi Junction NSW 2022 Australia', 'Bondi Junction', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'), 'D07004 - WE NSW, Bondi Junction', 'YES', 'D07004 - WE NSW, Bondi Junction');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(9,'RECDOCK (71-SYD) Singapore', 'NULL', 'SG', TO_DATE('2019-09-26','YYYY-MM-DD'), 'RECDOCK (71-SYD)', 'YES', 'RECDOCK (71-SYD)');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(10,'SYD DC 6 Milner Avenue Horsley Park NSW 2175 Australia', 'Horsley Park', 'AU', TO_DATE('2021-08-24','YYYY-MM-DD'), 'SYD DC', 'YES', 'SYD DC');
```

```
INSERT INTO KPI_STG_LOCATIONS VALUES(11,'07005 - WE Vic Chapel St 2013 NSW Australia', 'NULL', 'AU', TO_DATE('2017-08-07','YYYY-MM-DD'), 'D07005 - WE Vic Chapel St 2013', 'YES', 'D07005 - WE Vic Chapel St 2013');
```

## 4. Analyse the Business Keys if they meet Primary key conditions for all Stage tables

**1. Provide the SQLs to execute to ensure Primary Key conditions on business key**

**KPI\_STG\_CHANNEL**

SELECT COUNT(DISTINCT DATE\_CREATED) FROM KPI\_STG\_CHANNEL WHERE  
DATE\_CREATED IS NOT NULL;

4

SELECT COUNT(DISTINCT IS\_RECORD\_INACTIVE) FROM KPI\_STG\_CHANNEL WHERE  
IS\_RECORD\_INACTIVE IS NOT NULL;

1

SELECT COUNT(DISTINCT LAST\_MODIFIED\_DATE) FROM KPI\_STG\_CHANNEL WHERE  
LAST\_MODIFIED\_DATE IS NOT NULL;

3

SELECT COUNT(DISTINCT LIST\_ID), FROM KPI\_STG\_CHANNEL WHERE LIST\_ID IS NOT NULL;

5

SELECT COUNT(DISTINCT LIST\_ITEM\_NAME) FROM KPI\_STG\_CHANNEL WHERE  
LIST\_ITEM\_NAME IS NOT NULL;

5

**KPI\_STG\_CLASSES**

SELECT COUNT(CLASS\_ID) FROM KPI\_STG\_CLASSES;

SELECT COUNT(DISTINCT CLASS\_ID) FROM KPI\_STG\_CLASSES WHERE CLASS\_ID IS NOT NULL;

6

SELECT COUNT(DISTINCT DATE\_LAST\_MODIFIED) FROM KPI\_STG\_CLASSES WHERE  
DATE\_LAST\_MODIFIED IS NOT NULL;

3

SELECT COUNT(DISTINCT FULL\_NAME) FROM KPI\_STG\_CLASSES WHERE FULL\_NAME IS NOT  
NULL;

6

SELECT COUNT(DISTINCT ISINACTIVE) FROM KPI\_STG\_CLASSES WHERE ISINACTIVE IS NOT NULL;

1

SELECT COUNT(DISTINCT NAME) FROM KPI\_STG\_CLASSES WHERE NAME IS NOT NULL;

6

### **KPI\_STG\_DEPARTMENTS**

105

SELECT COUNT(\*) FROM KPI\_STG\_DEPARTMENTS;

SELECT COUNT(DISTINCT DATE\_LAST\_MODIFIED) FROM KPI\_STG\_DEPARTMENTS WHERE DATE\_LAST\_MODIFIED IS NOT NULL;

39

SELECT COUNT(DISTINCT DEPARTMENT\_ID) FROM KPI\_STG\_DEPARTMENTS WHERE DEPARTMENT\_ID IS NOT NULL;

105

SELECT COUNT(DISTINCT ISINACTIVE) FROM KPI\_STG\_DEPARTMENTS WHERE ISINACTIVE IS NOT NULL;

2

SELECT COUNT(DISTINCT NAME) FROM KPI\_STG\_DEPARTMENTS WHERE NAME IS NOT NULL;

105

SELECT COUNT(DISTINCT WS\_DESCRIPTION) FROM KPI\_STG\_DEPARTMENTS WHERE WS\_DESCRIPTION IS NOT NULL;

100

### **KPI\_STG\_ITEM\_MERCHANDISE\_CLASS**

83

SELECT COUNT(\*) FROM KPI\_STG\_ITEM\_MERCHANDISE\_CLASS;

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_CLASS_ID) FROM  
KPI_STG_ITEM_MERCHANDISE_CLASS WHERE ITEM_MERCHANDISE_CLASS_ID IS NOT NULL;
```

83

```
SELECT COUNT(DISTINCT DESCRIPTION) FROM KPI_STG_ITEM_MERCHANDISE_CLASS  
WHERE DESCRIPTION IS NOT NULL;
```

72

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_CLASS_NAME) FROM  
KPI_STG_ITEM_MERCHANDISE_CLASS WHERE ITEM_MERCHANDISE_CLASS_NAME IS NOT  
NULL;
```

17

#### **KPI\_STG\_ITEM\_MERCHANDISE\_COLLE**

86

```
SELECT COUNT(*) FROM KPI_STG_ITEM_MERCHANDISE_COLLE;
```

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_COLLECTION_ID) FROM  
KPI_STG_ITEM_MERCHANDISE_COLLE WHERE ITEM_MERCHANDISE_COLLECTION_ID IS  
NOT NULL;
```

86

```
SELECT COUNT(DISTINCT DESCRIPTION) FROM KPI_STG_ITEM_MERCHANDISE_COLLE  
WHERE DESCRIPTION IS NOT NULL;
```

86

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_COLLECTION_NA) FROM  
KPI_STG_ITEM_MERCHANDISE_COLLE WHERE ITEM_MERCHANDISE_COLLECTION_NA IS  
NOT NULL;
```

86

#### **KPI\_STG\_ITEM\_MERCHANDISE\_DEPAR**

87

```
SELECT COUNT(*) FROM KPI_STG_ITEM_MERCHANDISE_DEPAR;
```

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_DEPARTMENT_ID) FROM  
KPI_STG_ITEM_MERCHANDISE_DEPAR WHERE ITEM_MERCHANDISE_DEPARTMENT_ID IS  
NOT NULL;
```

87

```
SELECT COUNT(DISTINCT DESCRIPTION) FROM KPI_STG_ITEM_MERCHANDISE_DEPAR  
WHERE DESCRIPTION IS NOT NULL;
```

87

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_DEPARTMENT_NA) FROM  
KPI_STG_ITEM_MERCHANDISE_DEPAR WHERE ITEM_MERCHANDISE_DEPARTMENT_NA IS  
NOT NULL;
```

87

#### **KPI\_STG\_ITEM\_MERCHANDISE\_SUBCL**

85

```
SELECT COUNT(*) FROM KPI_STG_ITEM_MERCHANDISE_SUBCL;  
  
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_SUBCLASS_ID) FROM  
KPI_STG_ITEM_MERCHANDISE_SUBCL WHERE ITEM_MERCHANDISE_SUBCLASS_ID IS NOT  
NULL;
```

85

```
SELECT COUNT(DISTINCT DESCRIPTION) FROM KPI_STG_ITEM_MERCHANDISE_SUBCL  
WHERE DESCRIPTION IS NOT NULL;
```

53

```
SELECT COUNT(DISTINCT ITEM_MERCHANDISE_SUBCLASS_NAME) FROM  
KPI_STG_ITEM_MERCHANDISE_SUBCL WHERE ITEM_MERCHANDISE_SUBCLASS_NAME IS  
NOT NULL;
```

12

#### **KPI\_STG\_ITEMS**

13101

```
SELECT COUNT(*) FROM KPI_STG_ITEMS;  
  
SELECT COUNT(DISTINCT ITEM_ID) FROM KPI_STG_ITEMS WHERE ITEM_ID IS NOT NULL;
```

13098

```
SELECT COUNT(DISTINCT SKU) FROM KPI_STG_ITEMS WHERE SKU IS NOT NULL;
```

13097

SELECT COUNT(DISTINCT TYPE\_NAME) FROM KPI\_STG\_ITEMS WHERE TYPE\_NAME IS NOT NULL;

2

SELECT COUNT(DISTINCT SALESDESCRIPTION) FROM KPI\_STG\_ITEMS WHERE SALESDESCRIPTION IS NOT NULL;

13069

SELECT COUNT(DISTINCT CLASS\_ID) FROM KPI\_STG\_ITEMS WHERE CLASS\_ID IS NOT NULL;

4

SELECT COUNT(DISTINCT WS\_MERCHANDISE\_DEPARTMENT\_ID) FROM KPI\_STG\_ITEMS WHERE WS\_MERCHANDISE\_DEPARTMENT\_ID IS NOT NULL;

87

SELECT COUNT(DISTINCT WS\_MERCHANDISE\_COLLECTION\_ID) FROM KPI\_STG\_ITEMS WHERE WS\_MERCHANDISE\_COLLECTION\_ID IS NOT NULL;

3738

SELECT COUNT(DISTINCT WS\_MERCHANDISE\_CLASS\_ID) FROM KPI\_STG\_ITEMS WHERE WS\_MERCHANDISE\_CLASS\_ID IS NOT NULL;

457

SELECT COUNT(DISTINCT WS\_MERCHANDISE\_SUBCLASS\_ID) FROM KPI\_STG\_ITEMS WHERE WS\_MERCHANDISE\_SUBCLASS\_ID IS NOT NULL;

1240

## **KPI\_STG\_LOCATIONS**

114

SELECT COUNT(\*) FROM KPI\_STG\_LOCATIONS;

SELECT COUNT(DISTINCT LOCATION\_ID) FROM KPI\_STG\_LOCATIONS WHERE LOCATION\_ID IS NOT NULL;

114

SELECT COUNT(DISTINCT ADDRESS) FROM KPI\_STG\_LOCATIONS WHERE ADDRESS IS NOT NULL;

112

SELECT COUNT(DISTINCT CITY) FROM KPI\_STG\_LOCATIONS WHERE CITY IS NOT NULL;

34

```
SELECT COUNT(DISTINCT COUNTRY) FROM KPI_STG_LOCATIONS WHERE COUNTRY IS NOT NULL;
```

5

```
SELECT COUNT(DISTINCT DATE_LAST_MODIFIED) FROM KPI_STG_LOCATIONS WHERE DATE_LAST_MODIFIED IS NOT NULL;
```

31

```
SELECT COUNT(DISTINCT FULL_NAME) FROM KPI_STG_LOCATIONS WHERE FULL_NAME IS NOT NULL;
```

114

```
SELECT COUNT(DISTINCT ISINACTIVE) FROM KPI_STG_LOCATIONS WHERE ISINACTIVE IS NOT NULL;
```

2

```
SELECT COUNT(DISTINCT NAME) FROM KPI_STG_LOCATIONS WHERE NAME IS NOT NULL;
```

114

## **KPI\_STG\_TRANSACTIONS**

```
SELECT COUNT(*) FROM KPI_STG_TRANSACTIONS;
```

43932

```
SELECT COUNT(DISTINCT TRANSACTION_ID) FROM KPI_STG_TRANSACTIONS WHERE TRANSACTION_ID IS NOT NULL;
```

43924

```
SELECT COUNT(DISTINCT TRANID) FROM KPI_STG_TRANSACTIONS WHERE TRANID IS NOT NULL;
```

43924

```
SELECT COUNT(DISTINCT TRANSACTION_TYPE) FROM KPI_STG_TRANSACTIONS WHERE TRANSACTION_TYPE IS NOT NULL;
```

2



SELECT COUNT(DISTINCT TRANDATE) FROM KPI\_STG\_TRANSACTIONS WHERE TRANDATE IS NOT NULL;

30

SELECT COUNT(DISTINCT CHANNEL\_ID) FROM KPI\_STG\_TRANSACTIONS WHERE CHANNEL\_ID IS NOT NULL;

4

### **KPI\_STG\_TRANSACTIONS\_LINES**

SELECT COUNT(\*) FROM KPI\_STG\_TRANSACTIONS\_LINES;

147616

SELECT COUNT(DISTINCT TRANSACTION\_ID) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE TRANSACTION\_ID IS NOT NULL;

43924

SELECT COUNT(DISTINCT TRANSACTION\_LINE\_ID) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE TRANSACTION\_LINE\_ID IS NOT NULL;

187

SELECT COUNT(DISTINCT LOCATION\_ID) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE LOCATION\_ID IS NOT NULL;

20

SELECT COUNT(DISTINCT DEPARTMENT\_ID) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE DEPARTMENT\_ID IS NOT NULL;

33

SELECT COUNT(DISTINCT ITEM\_ID) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE ITEM\_ID IS NOT NULL;

13097

SELECT COUNT(DISTINCT AMOUNT) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE AMOUNT IS NOT NULL;

1416

SELECT COUNT(DISTINCT COST) FROM KPI\_STG\_TRANSACTIONS\_LINES WHERE COST IS NOT NULL;

1430

```
SELECT COUNT(DISTINCT UNITS) FROM KPI_STG_TRANSACTIONS_LINES WHERE UNITS IS NOT NULL;
```

104

## **5.Delete the duplicate records if exists and maintain unique record**

### **1.Provide the DELETE scripts using Analytical function**

#### **KPI\_STG\_ITEMS**

```
DELETE FROM KPI_STG_ITEMS  
WHERE WS_MERCHANDISE_COLLECTION_ID NOT IN (SELECT ITEM_MERCHANDISE_COLLECTION_ID  
FROM KPI_STG_ITEM_MERCHANDISE_COLLE);
```

```
DELETE FROM KPI_STG_ITEMS  
WHERE WS_MERCHANDISE_CLASS_ID NOT IN (SELECT ITEM_MERCHANDISE_CLASS_ID FROM  
KPI_STG_ITEM_MERCHANDISE_CLASS);
```

```
DELETE FROM KPI_STG_ITEMS  
WHERE WS_MERCHANDISE_SUBCLASS_ID NOT IN (SELECT ITEM_MERCHANDISE_SUBCLASS_ID  
FROM KPI_STG_ITEM_MERCHANDISE_SUBCL);
```

#### **KPI\_STG\_TRANSACTIONS\_LINES**

```
DELETE FROM KPI_STG_TRANSACTIONS_LINES  
WHERE ITEM_ID NOT IN (SELECT ITEM_ID FROM KPI_STG_ITEMS);
```

```
DELETE FROM KPI_STG_TRANSACTIONS_LINES  
WHERE DEPARTMENT_ID NOT IN (SELECT DEPARTMENT_ID FROM KPI_STG_DEPARTMENTS);
```

```
DELETE FROM KPI_STG_TRANSACTIONS_LINES  
WHERE LOCATION_ID NOT IN (SELECT LOCATION_ID FROM KPI_STG_LOCATIONS);
```

#### **KPI\_STG\_TRANSACTIONS**

```
DELETE FROM KPI_STG_TRANSACTIONS  
WHERE CHANNEL_ID NOT IN (SELECT LIST_ID FROM KPI_STG_CHANNEL);
```

```
COMMIT
```

## 6. Create Primary Key on Stage tables

**Provide the scripts used to create Primary Key**

### PRIMARY KEY

1. ALTER TABLE KPI\_STG\_CHANNEL ADD PRIMARY KEY(LIST\_ID);
2. ALTER TABLE KPI\_STG\_CLASSES ADD PRIMARY KEY(CLASS\_ID);
3. ALTER TABLE KPI\_STG\_DEPARTMENTS ADD PRIMARY KEY(DEPARTMENT\_ID);
4. ALTER TABLE KPI\_STG\_ITEM\_MERCHANDISE\_CLASS ADD PRIMARY KEY(ITEM\_MERCHANDISE\_CLASS\_ID);
5. ALTER TABLE KPI\_STG\_ITEM\_MERCHANDISE\_COLLE ADD PRIMARY KEY(ITEM\_MERCHANDISE\_COLLECTION\_ID);
6. ALTER TABLE KPI\_STG\_ITEM\_MERCHANDISE\_DEPAR ADD PRIMARY KEY(ITEM\_MERCHANDISE\_DEPARTMENT\_ID);
7. ALTER TABLE KPI\_STG\_ITEM\_MERCHANDISE\_SUBCL ADD PRIMARY KEY(ITEM\_MERCHANDISE\_SUBCLASS\_ID);
8. ALTER TABLE KPI\_STG\_ITEMS ADD PRIMARY KEY(ITEM\_ID);
9. ALTER TABLE KPI\_STG\_LOCATIONS ADD PRIMARY KEY(LOCATION\_ID);
10. ALTER TABLE KPI\_STG\_TRANSACTIONS ADD PRIMARY KEY(TRANSACTION\_ID);
11. ALTER TABLE KPI\_STG\_TRANSACTIONS\_LINES ADD PRIMARY KEY(TRANSACTION\_ID,TRANSACTION\_LINE\_ID);

## 7. Identify the relationships between each table

**Provide the SELECT SQLs executed to identify the relationships**

### FOREIGN KEYS

## **KPI\_STG\_TRANSACTIONS**

```
ALTER TABLE KPI_STG_TRANSACTIONS ADD CONSTRAINT FK_KPI_STG_TRANSACTIONS  
FOREIGN KEY(CHANNEL_ID) REFERENCES KPI_STG_CHANNEL(LIST_ID);
```

## **KPI\_STG\_ITEMS**

1. ALTER TABLE KPI\_STG\_ITEMS ADD CONSTRAINT FK\_KPI\_STG\_ITEMS  
FOREIGN KEY(CLASS\_ID) REFERENCES KPI\_STG\_CLASSES(CLASS\_ID);
2. ALTER TABLE KPI\_STG\_ITEMS ADD CONSTRAINT FK\_KP\_STG\_ITEMS  
FOREIGN KEY(WS\_MERCHANDISE\_DEPARTMENT\_ID) REFERENCES  
KPI\_STG\_ITEM\_MERCHANDISE\_DEPAR(ITEM\_MERCHANDISE\_DEPARTMENT\_ID);
3. ALTER TABLE KPI\_STG\_ITEMS ADD CONSTRAINT FK\_KPI\_STG\_ITEMS  
FOREIGN KEY(WS\_MERCHANDISE\_COLLECTION\_ID)  
REFERENKPI\_STG\_ITEM\_MERCHANDISE\_COLLE(ITEM\_MERCHANDISE\_COLLECTION\_ID);
4. ALTER TABLE KPI\_STG\_ITEMS ADD CONSTRAINT FK\_KPI\_ST\_ITEMS  
FOREIGN KEY(WS\_MERCHANDISE\_CLASS\_ID) REFERENCES  
KPI\_STG\_ITEM\_MERCHANDISE\_CLASS(ITEM\_MERCHANDISE\_CLASS\_ID);
5. ALTER TABLE KPI\_STG\_ITEMS ADD CONSTRAINT FK\_KPI\_STG\_ITEMS  
FOREIGN KEY(WS\_MERCHANDISE\_SUBCLASS\_ID) REFERENCES  
KPI\_STG\_ITEM\_MERCHANDISE\_SUBCL(ITEM\_MERCHANDISE\_SUBCLASS\_ID);

## **KPI\_STG\_TRANSACTION\_LINES**

1. ALTER TABLE KPI\_STG\_TRANSACTIONS\_LINES ADD CONSTRAINT  
FK\_KPI\_STG\_TRANSACTIONS\_LINES  
FOREIGN KEY (LOCATION\_ID) REFERENCES KPI\_STG\_LOCATIONS(LOCATION\_ID);
- 2.ALTER TABLE KPI\_STG\_TRANSACTIONS\_LINES ADD CONSTRAINT FK\_KPI\_TRANSACTIONS\_LINES  
FOREIGN KEY(DEPARTMENT\_ID) REFERENCES KPI\_STG\_DEPARTMENTS(DEPARTMENT\_ID);
3. ALTER TABLE KPI\_STG\_TRANSACTIONS\_LINES ADD CONSTRAINT FK\_STG\_TRANSACTIONS\_LINES  
FOREIGN KEY(ITEM\_ID) REFERENCES KPI\_STG\_ITEMS(ITEM\_ID);

## 8. Create Target Tables

### 1. Create all the target tables

#### KPI\_LOCATION\_DIM

```
create table KPI_LOCATION_DIM(  
  LOCATION_ID NUMBER(20,0), 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(20,0), KPI_DW_INSERT_DATE DATE,  
  KPI_DW_UPDATE_DATE DATE  
);
```

#### KPI\_TRANSACTION\_LINE\_FACT

```
CREATE TABLE KPI_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)  
);
```

#### KPI\_CHANNEL\_DIM

```
create table KPI_CHANNEL_DIM (  
  DATE_CREATED date, IS_RECORD_INACTIVE varchar2(100),  
  LAST_MODIFIED_DATE date, LIST_ID number(20,0),  
  LIST_ITEM_NAME varchar2(20), KPI_DW_SKEY number(20,0),  
  KPI_DW_INSERT_DATE date, KPI_DW_UPDATE_DATE date  
);
```

#### KPI\_CLASS\_DIM

```
create table KPI_CLASS_DIM (  
  CLASS_ID number(20,0), DATE_LAST_MODIFIED date,  
  FULL_NAME varchar2(30), ISINACTIVE varchar2(5),  
  NAME varchar2(5), KPI_DW_SKEY number(20,0),  
  KPI_DW_INSERT_DATE date, KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM**

```
create table KPI_ITEM_MERCHANDISE_DEPAR_DIM (  
    ITEM_MERCHANDISE_DEPARTMENT_ID number(20,0),  
    DESCRIPTION varchar2(50),  
    ITEM_MERCHANDISE_DEPARTMENT_NA varchar2(10),  
    KPI_DW_SKEY number(20,0),  
    KPI_DW_INSERT_DATE date,  
    KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_ITEM\_MERCHANDISE\_COL\_DIM**

```
create table KPI_ITEM_MERCHANDISE_COL_DIM (  
    ITEM_MERCHANDISE_COLLECTION_ID number(20,0),  
    DESCRIPTION varchar2(100),  
    ITEM_MERCHANDISE_COLLECTION_NA varchar2(100),  
    KPI_DW_SKEY number(20,0),  
    KPI_DW_INSERT_DATE date,  
    KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_ITEM\_MERCHANDISE\_CLASS\_DIM**

```
create table KPI_ITEM_MERCHANDISE_CLASS_DIM (  
    ITEM_MERCHANDISE_CLASS_ID number(20,0),  
    DESCRIPTION varchar2(100),  
    ITEM_MERCHANDISE_CLASS_NAME varchar2(100),  
    KPI_DW_SKEY number(20,0),  
    KPI_DW_INSERT_DATE date,  
    KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_ITEM\_MERCHANDISE\_SUBCL\_DIM**

```
create table KPI_ITEM_MERCHANDISE_SUBCL_DIM (  
    ITEM_MERCHANDISE_SUBCLASS_ID number(20,0),  
    DESCRIPTION varchar2(100),  
    ITEM_MERCHANDISE_SUBCLASS_NAME varchar2(100),  
    KPI_DW_SKEY number(20,0),  
    KPI_DW_INSERT_DATE date,  
    KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_DEPARTMENT\_DIM**

```
create table KPI_DEPARTMENT_DIM    (  
    DATE_LAST_MODIFIED date,  
    DEPARTMENT_ID number(20,0),  
    ISINACTIVE varchar2(100),  
    NAME varchar2(10),  
    WS_DESCRIPTION varchar2(100),  
    KPI_DW_SKEY number(20,0),  
    KPI_DW_INSERT_DATE date,  
    KPI_DW_UPDATE_DATE date  
);
```

## **KPI\_ITEM\_DIM**

```
create table KPI_ITEM_DIM (  
    ITEM_ID number(20,0),  
    SKU varchar2(100),  
    TYPE_NAME varchar2(100),  
    SALESDESCRIPTION varchar2(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)  
);
```

## **2. CREATE SEQUENCE to populate KPI\_DW\_SKEY field in all Target tables.**

### **Provide all the scripts**

1. CREATE SEQUENCE M1;

UPDATE KPI\_CHANNEL\_DIM SET KPI\_DW\_SKEY=M1.NEXTVAL;

ALTER TABLE KPI\_CHANNEL\_DIM MODIFY KPI\_DW\_INSERT\_DATE DEFAULT SYSDATE;

ALTER TABLE KPI\_CHANNEL\_DIM MODIFY KPI\_DW\_UPDATE\_DATE DEFAULT SYSDATE;

UPDATE KPI\_CHANNEL\_DIM SET KPI\_DW\_INSERT\_DATE=SYSDATE,KPI\_DW\_UPDATE\_DATE=SYSDATE  
WHERE KPI\_DW\_SKEY IS NOT NULL;

```
SELECT * FROM KPI_CHANNEL_DIM;
```

## 2. CREATE SEQUENCE M2;

```
UPDATE KPI_CLASS_DIM SET KPI_DW_SKEY=M2.NEXTVAL;
```

```
ALTER TABLE KPI_CLASS_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT SYSDATE;
```

```
ALTER TABLE KPI_CLASS_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT SYSDATE;
```

```
UPDATE KPI_CLASS_DIM SET KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE  
WHERE KPI_DW_SKEY IS NOT NULL;
```

## 3. CREATE SEQUENCE M3;

```
UPDATE KPI_DEPARTMENT_DIM SET KPI_DW_SKEY=M3.NEXTVAL;
```

```
ALTER TABLE KPI_DEPARTMENT_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT SYSDATE;
```

```
ALTER TABLE KPI_DEPARTMENT_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT SYSDATE;
```

```
UPDATE KPI_DEPARTMENT_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

## 4. CREATE SEQUENCE M4;

```
UPDATE KPI_ITEM_DIM SET KPI_DW_SKEY=M4.NEXTVAL;
```

```
ALTER TABLE KPI_ITEM_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT SYSDATE;
```

```
ALTER TABLE KPI_ITEM_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT SYSDATE;
```

```
UPDATE KPI_ITEM_DIM SET KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE  
WHERE KPI_DW_SKEY IS NOT NULL;
```

## 5. CREATE SEQUENCE M5;

```
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET KPI_DW_SKEY=M5.NEXTVAL;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_CLASS_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT  
SYSDATE;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_CLASS_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT  
SYSDATE;
```

```
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```



6. CREATE SEQUENCE M6;

```
UPDATE KPI_ITEM_MERCHANDISE_COL_DIM SET KPI_DW_SKEY=M6.NEXTVAL;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_COL_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT  
SYSDATE;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_COL_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT  
SYSDATE;
```

```
UPDATE KPI_ITEM_MERCHANDISE_COL_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

7. CREATE SEQUENCE M7;

```
UPDATE KPI_ITEM_MERCHANDISE_DEPAR_DIM SET KPI_DW_SKEY=M7.NEXTVAL;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_DEPAR_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT  
SYSDATE;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_DEPAR_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT  
SYSDATE;
```

```
UPDATE KPI_ITEM_MERCHANDISE_DEPAR_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

8. CREATE SEQUENCE M8;

```
UPDATE KPI_ITEM_MERCHANDISE_SUBCL_DIM SET KPI_DW_SKEY=M7.NEXTVAL;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_SUBCL_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT  
SYSDATE;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_SUBCL_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT  
SYSDATE;
```

```
UPDATE KPI_ITEM_MERCHANDISE_SUBCL_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

9. CREATE SEQUENCE M9;

```
UPDATE KPI_LOCATION_DIM SET KPI_DW_SKEY=M9.NEXTVAL;
```

```
ALTER TABLE KPI_LOACTION_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT SYSDATE;
```

```
ALTER TABLE KPI_LOCATION_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT SYSDATE;
```

```
UPDATE KPI_LOCATION_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

```
SELECT * FROM KPI_LOCATION_DIM;
```

**10.**CREATE SEQUENCE M10;

```
UPDATE KPI_TRANSACTION SET KPI_DW_SKEY=M10.NEXTVAL;  
ALTER TABLE KPI_ITEM_MERCHANDISE_CLASS_DIM MODIFY KPI_DW_INSERT_DATE DEFAULT  
SYSDATE;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_CLASS_DIM MODIFY KPI_DW_UPDATE_DATE DEFAULT  
SYSDATE;
```

```
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET  
KPI_DW_INSERT_DATE=SYSDATE,KPI_DW_UPDATE_DATE=SYSDATE WHERE KPI_DW_SKEY IS NOT  
NULL;
```

### **3. Create PRIMARY KEY on KPI\_DW\_SKEY**

1. ALTER TABLE KPI\_LOCATION\_DIM ADD PRIMARY KEY(KPI\_DW\_SKEY);  
DESC KPI\_LOCATION\_DIM;
2. ALTER TABLE KPI\_TRANSACTION\_LINE\_FACT ADD PRIMARY  
KEY(KPI\_DW\_SKEY);  
DESC KPI\_TRANSACTION\_LINE\_FACT;
3. ALTER TABLE KPI\_CHANNEL\_DIM ADD PRIMARY KEY(KPI\_DW\_SKEY);  
DESC KPI\_CHANNEL\_DIM;
4. ALTER TABLE KPI\_CLASS\_DIM ADD PRIMARY KEY(KPI\_DW\_SKEY);  
DESC KPI\_CLASS\_DIM;
5. ALTER TABLE KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM ADD PRIMARY  
KEY(KPI\_DW\_SKEY);  
DESC KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM;
6. ALTER TABLE KPI\_ITEM\_MERCHANDISE\_COL\_DIM ADD PRIMARY  
KEY(KPI\_DW\_SKEY);  
DESC KPI\_ITEM\_MERCHANDISE\_COL\_DIM;
7. ALTER TABLE KPI\_ITEM\_MERCHANDISE\_CLASS\_DIM ADD PRIMARY

```

KEY(KPI_DW_SKEY);

DESC KPI_ITEM_MERCHANDISE_CLASS_DIM;

8. ALTER TABLE KPI_ITEM_MERCHANDISE_SUBCL_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_ITEM_MERCHANDISE_SUBCL_DIM;

9. ALTER TABLE KPI_DEPARTMENT_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_DEPARTMENT_DIM;

10 . ALTER TABLE KPI_ITEM_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_ITEM_DIM;

```

## 9. Target Tables load

### Load the Target Tables using Stage Tables.

#### 1. Identify the sequence in which the Target Tables has to be loaded.

##### Provide the reasons

```

CONNECT MOUNISHA
ENTER PASSWORD:
CONNECTED.

```

```

1.GRANT SELECT KPI_STG_CHANNEL TO AMMU;

GRANT SUCCEEDED.

2.GRANT SELECT ON KPI_STG_CLASSES TO AMMU;

GRANT SUCCEEDED.

3.GRANT SELECT ON KPI_STG_DEPARTMENTS TO AMMU;

GRANT SUCCEEDED.

4. GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_CLASS TO AMMU;

GRANT SUCCEEDED.

5. GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_COLLE TO AMMU;

GRANT SUCCEEDED.

6. GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_DEPAR TO AMMU;

GRANT SUCCEEDED.

7.GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_SUBCL TO AMMU;

GRANT SUCCEEDED.

8. GRANT SELECT ON KPI_STG_ITEMS TO AMMU;

GRANT SUCCEEDED.

```

9. GRANT SELECT ON KPI\_STG\_TRANSACTIONS TO AMMU;

GRANT SUCCEEDED.

10. GRANT SELECT ON KPI\_STG\_TRANSACTIONS\_LINES TO AMMU;

GRANT SUCCEEDED.

11. GRANT SELECT ON KPI\_STG\_LOCATION TO AMMU;

GRANT SUCCEEDED.

WE HAVE TWO DATABASES THAT IS 'STAGE TABLE' AND 'TARGET TABLE'. WE DON'T HAVE DATA SCRIPT FOR TARGET TABLE. SO, BY GIVING COMMAND (CONNECT SOURCE USER NAME). TO STAGE TABLE WE ARE INSERTING THE SCRIPT BY (CONNECT TARGET USER NAME).

## **2.PROVIDE THE INSERT SCRIPTS USED TO PERFORM THE DATA LOAD**

### **KPI\_CHANNEL\_DIM**

1. INSERT INTO  
KPI\_CHANNEL\_DIM (DATE\_CREATED, IS\_RECORD\_INACTIVE, LAST\_MODIFIED\_DATE, LIST\_ID, LIST\_ITEM\_NAME)  
(SELECT \* FROM MOUNISHA.KPI\_STG\_CHANNEL);  
  
SELECT \* FROM KPI\_CHANNEL\_DIM;

### **KPI\_CLASS\_DIM**

2. INSERT INTO  
KPI\_CLASS\_DIM (CLASS\_ID, DATE\_LAST\_MODIFIED, FULL\_NAME, IS\_INACTIVE, NAME) (SELECT \*  
FROM MOUNISHA.KPI\_STG\_CLASSES);  
SELECT \* FROM KPI\_CLASS\_DIM;

### **KPI\_DEPARTMENT\_DIM**

ALTER TABLE KPI\_CLASS\_DIM MODIFY NAME VARCHAR2(50);

3. INSERT INTO  
KPI\_DEPARTMENT\_DIM (DATE\_LAST\_MODIFIED, DEPARTMENT\_ID, IS\_INACTIVE, NAME, WS\_DESCRIPTION)  
(SELECT \* FROM MOUNISHA.KPI\_STG\_DEPARTMENTS);  
  
SELECT \* FROM KPI\_DEPARTMENT\_DIM;

### **KPI\_ITEM\_MERCHANDISE\_CLASS\_DIM**

4. INSERT INTO  
KPI\_ITEM\_MERCHANDISE\_CLASS\_DIM (ITEM\_MERCHANDISE\_CLASS\_ID, DESCRIPTION, ITEM\_MERCHANDISE\_CLASS\_NAME)  
(SELECT \* FROM MOUNISHA.KPI\_STG\_ITEM\_MERCHANDISE\_CLASS);  
  
SELECT \* FROM KPI\_ITEM\_MERCHANDISE\_CLASS\_DIM;

### **KPI\_ITEM\_MERCHANDISE\_COL\_DIM**

5. INSERT INTO  
 KPI\_ITEM\_MERCHANDISE\_COL\_DIM(ITEM\_MERCHANDISE\_COLLECTION\_ID,DESCRIPTION,ITEM\_MERCHANDISE\_COLLECTION\_NAME)  
 (SELECT \* FROM MOUNISHA.KPI\_STG\_ITEM\_MERCHANDISE\_COLLE);

SELECT \* FROM KPI\_ITEM\_MERCHANDISE\_COL\_DIM;

#### **KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM**

6. INSERT INTO  
 KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM(ITEM\_MERCHANDISE\_DEPARTMENT\_ID,DESCRIPTION,ITEM\_MERCHANDISE\_DEPARTMENT\_NAME)  
 (SELECT \* FROM MOUNISHA.KPI\_STG\_ITEM\_MERCHANDISE\_DEPAR);

SELECT \* FROM KPI\_ITEM\_MERCHANDISE\_DEPAR\_DIM;

#### **KPI\_ITEM\_MERCHANDISE\_SUBCL\_DIM**

7. INSERT INTO  
 KPI\_ITEM\_MERCHANDISE\_SUBCL\_DIM(ITEM\_MERCHANDISE\_SUBCLASS\_ID,DESCRIPTION,ITEM\_MERCHANDISE\_SUBCLASS\_NAME)  
 (SELECT \* FROM MOUNISHA.KPI\_STG\_ITEM\_MERCHANDISE\_SUBCL);

SELECT \* FROM KPI\_ITEM\_MERCHANDISE\_SUBCL\_DIM;

#### **KPI\_LOCATION\_DIM**

8. INSERT INTO  
 KPI\_LOCATION\_DIM(LOCATION\_ID,ADDRESS,CITY,COUNTRY,DATE\_LAST\_MODIFIED,FULL\_NAME,ISINACTIVE,NAME)  
 (SELECT \* FROM MOUNISHA.KPI\_STG\_LOCATIONS);

SELECT \* FROM KPI\_LOCATION\_DIM;

#### **KPI\_TRANSACTION\_LINE\_FACT**

9. INSERT INTO  
 KPI\_TRANSACTION\_LINE\_FACT(TRANSACTION\_ID,TRANSACTION\_LINE\_ID,TRANID,TRANSACTION\_TYPE,TRANDATE,KPI\_CHANNEL\_SKEY,  
 KPI\_LOCATION\_SKEY,KPI\_DEPARTMENT\_SKEY,KPI\_ITEM\_SKEY,AMOUNT,COST,UNITS)  
 (SELECT  
 A.TRANSACTION\_ID,B.TRANSACTION\_LINE\_ID,A.TRANID,A.TRANSACTION\_TYPE,A.TRANDATE,  
 A.CHANNEL\_ID,  
 B.LOCATION\_ID,B.DEPARTMENT\_ID,B.ITEM\_ID,B.AMOUNT,B.COST,B.UNITS  
 FROM MOUNISHA.KPI\_STG\_TRANSACTIONS A,MOUNISHA.KPI\_STG\_TRANSACTIONS\_LINES B WHERE  
 B.TRANSACTION\_ID = A.TRANSACTION\_ID);  
 DROP TABLE KPI\_TRANSACTION\_LINE\_FACT;

SELECT \* FROM KPI\_TRANSACTION\_LINE\_FACT;

#### **KPI\_ITEM\_DIM**

10. INSERT INTO  
 KPI\_ITEM\_DIM(ITEM\_ID,SKU,TYPE\_NAME,SALESDESCRIPTION,KPI\_CLASS\_SKEY,WS\_MERCHANDISE\_DEPARTMENT\_SKEY,  
 WS\_MERCHANDISE\_COLLECTION\_SKEY,WS\_MERCHANDISE\_CLASS\_SKEY,WS\_MERCHANDISE\_SUBCLASS\_SKEY)  
 (SELECT \* FROM MOUNISHA.KPI\_STG\_ITEMS);

SELECT \* FROM KPI\_ITEM\_DIM;

**10.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**

```
ALTER TABLE KPI_ITEM_DIM ADD BRAND_NAME VARCHAR2(100);
```

**2.Provide the UPDATE script to populate BRAND\_NAME field**

```
UPDATE KPI_ITEM_DIM M SET M.BRAND_NAME=(SELECT R.NAME
FROM KPI_CLASS_DIM R WHERE R.CLASS_ID=M.KPI_CLASS_SKEY);
SELECT * FROM KPI_ITEM_DIM;
```

**11. CREATE KPI\_ITEM\_DIM\_FLAT table STRUCTURE ONLY with following fields using SELECT statement joining the required Target tables**

**1. Provide the CREATE script**

```
CREATE TABLE ITEM_DIM_FLAT(SKU VARCHAR(100), ITEM_TYPE VARCHAR(50), BRAND
VARCHAR(50), MERCHANDISE_DEPARTMENT VARCHAR(50),
MERCHANDISE_DEPT_NAME VARCHAR(50), 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(50), KPI_ITEM_SKEY NUMBER);
ITEM_MERCHANDISE_CLASS.DESCRPTION,ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLAS
S_NAME,
ITEM_MERCHANDISE_SUBCLASS.DESCRPTION,ITEM_MERCHANDISE_SUBCLASS.ITEM_MERCHANDIS
E_SUBCLASS_NAME,ITEMS.KPI_DW_SKEY
from
ITEMS,ITEM_MERCHANDISE_DEPARTMENT,ITEM_MERCHANDISE_COLLECTION,ITEM_MERCHANDIS
E_CLASS,
ITEM_MERCHANDISE_SUBCLASS);
```

**2.Provide the BULK INSERT script to load this table**

```
INSERT INTO KPI_ITEM_DIM_FLAT (SKU VARCHAR2(100),ITEM_TYPE VARCHAR(100),BRAND
VARCHAR2(100),MERCHANDISE_DEPARTMENT VARCHAR2(120),
MERCHANDISE_DEPT_NAME VARCHAR2(100),MERCHANDISE_COLLECTION
VARCHAR2(100),ERCHANDISE_COLLECTION_NAME VARCHAR2(100),
```

```

MERCHANDISE_CLASS VARCHAR2(100),MERCHANDISE_CLASS_NAME
VARCHAR2(100),MERCHANDISE_SUBCLASS VARCHAR2(100),
MERCHANDISE_SUBCLASS_NAME VARCHAR2(100),KPI_ITEM_SKEY NUMBER)

SELECT
ITEMS.NAME,ITEMS.TYPE_NAME,ITEMS.BRAND_NAME,ITEM_MERCHANDISE_DEPARTMENT.DESCRIP
TION,ITEM_MERCHANDISE_DEPARTMENT.ITEM_MERCHANDISE_DEPARTMENT_NA,
ITEM_MERCHANDISE_COLLECTION.DESCRPTION,ITEM_MERCHANDISE_COLLECTION.ITEM_MERCHA
NDISE_COLLECTION_NA,
ITEM_MERCHANDISE_CLASS.DESCRPTION,ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLAS
S_NAME,
ITEM_MERCHANDISE_SUBCLASS.DESCRPTION,ITEM_MERCHANDISE_SUBCLASS.ITEM_MERCHANDIS
E_SUBCLASS_NAME,ITEMS.KPI_DW_SKEY
FROM
ITEMS,ITEM_MERCHANDISE_DEPARTMENT,ITEM_MERCHANDISE_COLLECTION,ITEM_MERCHANDIS
E_CLASS,
ITEM_MERCHANDISE_SUBCLASS);

```

### **3.Create a CURSOR to perform ROW by ROW inserts into this table**

```

CREATE TABLE ITEM_DIM_FLAT(SKU VARCHAR(100), ITEM_TYPE VARCHAR(50), BRAND
VARCHAR(50), MERCHANDISE_DEPARTMENT VARCHAR(50),
MERCHANDISE_DEPT_NAME VARCHAR(50), 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(50), KPI_ITEM_SKEY NUMBER);

DECLARE

CURSOR C1 IS SELECT I.SKU, I.TYPE_NAME, I.BRAND_NAME, I.KPI_DW_SKEY, D.DESCRPTION,
D.ITEM_MERCHANDISE_DEPARTMENT_NA,
CL.DESCRPTION, CL.ITEM_MERCHANDISE_COLLECTION_NA, C.DESCRPTION,
C.ITEM_MERCHANDISE_CLASS_NAME,
S.DESCRPTION, S.ITEM_MERCHANDISE_SUBCLASS_NAME FROM KPI_ITEM_DIM I JOIN
KPI_ITEM_MERCHANDISE_DEPAR_DIM
D ON I.KPI_DW_SKEY=D.KPI_DW_SKEY JOIN KPI_ITEM_MERCHANDISE_COL_DIM CL ON
D.KPI_DW_SKEY=CL.KPI_DW_SKEY JOIN KPI_ITEM_MERCHANDISE_CLASS_DIM
C ON CL.KPI_DW_SKEY=C.KPI_DW_SKEY JOIN KPI_ITEM_MERCHANDISE_SUBCL_DIM S ON
C.KPI_DW_SKEY=S.KPI_DW_SKEY;

```

```

BEGIN

FOR CUR IN C1 LOOP

INSERT INTO ITEM_DIM_FLAT VALUES(C1.SKU, C1.ITEM_TYPE,
C1.BRAND,C1.MERCHANDISE_DEPARTMENT,C1.MERCHANDISE_DEPT_NAME,C1.MERCHANDISE_COLLECTION,
C1.MERCHANDISE_COLLECTION_NAME,C1.MERCHANDISE_CLASS,C1.MERCHANDISE_CLASS_NAME,
C1.MERCHANDISE_SUBCLASS,C1.MERCHANDISE_SUBCLASS_NAME,C1.KPI_ITEM_KEY NUMBER)

(SELECT I.SKU,I.TYPE_NAME,
I.BRAND_NAME,I.KPI_DW_KEY,D.DESCRPTION,D.ITEM_MERCHANDISE_DEPARTMENT_NA,CL.DESCRPTION,CL.ITEM_MERCHANDISE_COLLECTION_NA,
C.DESCRPTION,C.ITEM_MERCHANDISE_CLASS_NAME,S.DESCRPTION,S.ITEM_MERCHANDISE_SUBCLASS_NAME FROM KPI_ITEM_DIM I,KPI_ITEM_MERCHANDISE_DEPAR_DIM
D,KPI_ITEM_MERCHANDISE_COL_DIM CL,KPI_ITEM_MERCHANDISE_CLASS_DIM
C,KPI_ITEM_MERCHANDISE_SUBCL_DIM S);

END LOOP;

CLOSE C1;

END;

```

**12.If TRANSACTION\_TYPE is " Sales Order " then its Demand, if TRANSACTION\_TYPE is " Invoice" then its Sales**

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

```

SELECT TRANSACTION_TYPE, AMOUNT FROM (SELECT TRANSACTION_TYPE, AMOUNT,
ROW_NUMBER() OVER (PARTITION BY TRANSACTION_TYPE ORDER BY AMOUNT DESC) TOP_VAL,
ROW_NUMBER() OVER (PARTITION BY TRANSACTION_TYPE ORDER BY AMOUNT) BOTTOM_VAL)
WHERE TOP_VAL<=5 OR BOTTOM_VAL<=5;

```

**2.Which Department has the highest Demand and Sales Amount**

```

SELECT D.NAME, MAX(T.AMOUNT) FROM DEPARTMENT_DIM D JOIN TRANSACTION_LINE_FACT T
ON D.KPI_DW_KEY=T.KPI_DW_KEY GROUP BY T.TRANSACTION_TYPE, D.NAME HAVING
TRANSACTION_TYPE='SALES ORDER' OR TRANSACTION_TYPE='INVOICES';

```

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



```

SELECT L.CITY FROM LOCATION_DIM L JOIN TRANSACTION_LINE_FACT F ON
F.KPI_DW_SKEY=L.KPI_DW_SKEY WHERE TRANSACTION_TYPE='SALES ORDER' ORDER BY
TRANSACTION_TYPE;

```

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

```

SELECT TRANSACTION_TYPE, AMOUNT, UNITS FROM TRANSACTION_LINE_FACT GROUP BY
TRANSACTION_TYPE, AMOUNT, UNITS ORDER BY 1;

```

### 6.Write a VIEW using target tables with following fields

```

CREATE FORCE VIEW TARGET_VIEW AS SELECT T.TRANSACTION_ID, T.TRANSACTION_LINE_ID,
T.TRANSDATE, T.TRANSACTION_TYPE,
        I.TYPE_NAME,
        L.CITY,
        D.NAME,
        CD.LIST_ITEM_NAME,
        ID.ITEM_MERCH_DEPARTMENT_NA,
        ID.DESCRPTION,
        IC.ITEM_MERCH_COLLECTION_NA,
        IC.DESCRPTION,
        C.ITEM_MERCH_CLASS_NAME,
        C.DESCRPTION,
        S.ITEM_MERCH_SUBCLASS_NAME,
        S.DESCRPTION,
        T.AMOUNT,
        T.UNITS
FROM KPI_TRANSACTION_LINE_FACT T JOIN KPI_ITEM_DIM I ON T.KPI_DW_SKEY =
I.KPI_DW_SKEY JOIN KPI_LOCATION_DIM L ON I.KPI_DW_SKEY = L.KPI_DW_SKEY
JOIN KPI_DEPARTMENT_DIM D ON L.KPI_DW_SKEY = D.KPI_DW_SKEY
JOIN KPI_CHANNEL_DIM CD ON D.KPI_DW_SKEY = CD.KPI_DW_SKEY
JOIN KPI_ITEM_MERCH_DEPARTMENT_DIM ID ON CD.KPI_DW_SKEY = ID.KPI_DW_SKEY
JOIN KPI_ITEM_MERCH_COLLECTION_DIM IC ON ID.KPI_DW_SKEY = IC.KPI_DW_SKEY
JOIN KPI_ITEM_MERCH_CLASS_DIM C ON IC.KPI_DW_SKEY = C.KPI_DW_SKEY
JOIN KPI_ITEM_MERCH_SUBCLASS_DIM S ON C.KPI_DW_SKEY = S.KPI_DW_SKEY;

```