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:

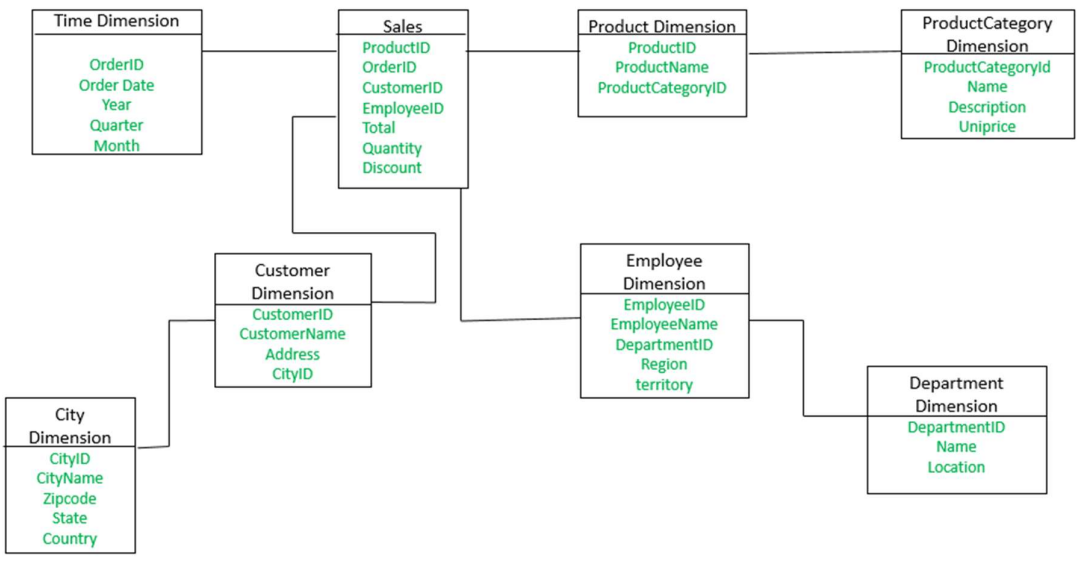
The snowflake schema is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions.

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 â€“ 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_SKEYnumber(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_DEPARTENT_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,ISINACTIVE,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,ISINACTIVE,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_NA)
   (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_NA)

(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,IT
EM_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_N
AME,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,TRANSAC
TION_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_MERCH
ANDISE_DEPARTMENT_SKEY,
WS_MERCHANDISE_COLLECTION_SKEY,WS_MERCHANDISE_CLASS_SKEY,WS_MERCHANDISE_SUBCL
ASS_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.DESCRIPTION,ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLAS S_NAME,

ITEM_MERCHANDISE_SUBCLASS.DESCRIPTION,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.DESCRI PTION,ITEM_MERCHANDISE_DEPARTMENT.ITEM_MERCHANDISE_DEPARTMENT_NA,

ITEM_MERCHANDISE_COLLECTION.DESCRIPTION,ITEM_MERCHANDISE_COLLECTION.ITEM_MERCHA NDISE_COLLECTION_NA,

ITEM_MERCHANDISE_CLASS.DESCRIPTION,ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLAS S_NAME,

ITEM_MERCHANDISE_SUBCLASS.DESCRIPTION,ITEM_MERCHANDISE_SUBCLASS.ITEM_MERCHANDISE_SUBCLASS_NAME,ITEMS.KPI_DW_SKEY

FROM
ITEMS,ITEM_MERCHANDISE_DEPARTMENT,ITEM_MERCHANDISE_COLLECTION,ITEM_MERCHANDISE_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.DESCRIPTION, D.ITEM_MERCHANDISE_DEPARTMENT_NA,

CL.DESCRIPTION, CL.ITEM_MERCHANDISE_COLLECTION_NA, C.DESCRIPTION, C.ITEM_MERCHANDISE_CLASS_NAME,

S.DESCRIPTION, 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_SKEY NUMBER)

(SELECT I.SKU,I.TYPE_NAME, I.BRAND_NAME,I.KPI_DW_SKEY,D.DESCRIPTION,D.ITEM_MERCHANDISE_DEPARTMENT_NA,CL.DESCRIPTION,CL.ITEM_MERCHANDISE_COLLECTION_NA,

C.DESCRIPTION,C.ITEM_MERCHANDISE_CLASS_NAME,S.DESCRIPTION,S.ITEM_MERCHANDISE_SUBC
LASS_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_SKEY=T.KPI_DW_SKEY 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.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 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;
```