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

Data Warehousing (DW): A Data Warehousing is process for collecting and managing data from varied sources to provide meaningful business insights. A Data warehouse is typically used to connect and analyse business data from heterogeneous sources. The data warehouse is the core of the BI system which is built for data analysis and reporting.

1.1.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. 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.

1.2. Understand how to set the dependencies during Stage tables and Target Tables load.

By using ETL process we can set the dependencies during stage tables and target tables.

- In stage tables first we need the load the source data and remove the duplicate records. After removing the duplicate records from the tables we need to set the dependencies(like primary keys and foreign keys).

- In target tables to set the dependencies the table must have doesn't have duplicate records. Then only we can set the dependencies.

1.3. What are common issues with this model.

- Snowflake works on very few cloud tools like AWS, Azure and Google cloud.

- Snowflaking reduces space consumed by dimension tables but compared with the entire data warehouse the saving is usually insignificant.

- Snowflake does not support the parallelism functionality. As a result, parallelism does not work when importing data from the Snowflake data store either by using the command composer on the Analyse page or by using the DB Import command.

- Cancelling of PySpark paragraphs for Snowflake query in Notebooks does not cancel the corresponding Snowflake query in the Snowflake UI.

1.4. Are there any options to convert this model to START? If SO, how ?

Yes, we can convert snowflake model to star model. In the snowflake model we have only one fact table and multiple dimensional tables in a normalized form. In order to convert snowflake model to star model we need to join the dimensional tables which are normalized using joins(this is also known as denormalization).

2. Create Stage Tables. Provide all the CREATE statements.

```
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;


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


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;

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;

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;

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;
```

```
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;

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;

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

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;

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. 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_TRANSACTION:

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.

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

ANALYSING THE BUSINESS KEYS/DISTINCT RECORDS:

KPI_STG_CHANNEL:

SELECT COUNT (*) FROM 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 (*) FROM KPI_STG_CLASSES;

6

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:

SELECT COUNT (*) FROM KPI_STG_DEPARTMENTS;

105

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:

SELECT COUNT (*) FROM KPI_STG_ITEM_MERCHANDISE_CLASS;

83

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:

SELECT COUNT (*) FROM KPI_STG_ITEM_MERCHANDISE_COLLE;

86

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:

SELECT COUNT (*) FROM KPI_STG_ITEM_MERCHANDISE_DEPAR;

87

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:

SELECT COUNT (*) FROM KPI_STG_ITEM_MERCHANDISE_SUBCL;

85

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:

SELECT COUNT (*) FROM KPI_STG_ITEMS;

13101

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

FROM KPI_STG_LOCATIONS:

SELECT COUNT (*) FROM KPI_STG_LOCATIONS;

114

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.

Provide the DELETE scripts using Analytical function.

REMOVING DUPLICATE RECORDS:

```
DELETE FROM KPI_STG_CHANNEL
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM KPI_STG_CHANNEL
GROUP BY LIST_ID) ;

DELETE FROM KPI_STG_CLASSES
WHERE ROWID NOT IN
(SELECT MIN(ROWID) FROM KPI_STG_CLASSES GROUP BY CLASS_ID);

DELETE FROM KPI_STG_DEPARTMENTS
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM KPI_STG_DEPARTMENTS
GROUP BY DEPARTMENT_ID) ;

DELETE FROM KPI_STG_ITEM_MERCHANDISE_CLASS
WHERE ROWID NOT IN
(SELECT MIN(ROWID) FROM KPI_STG_ITEM_MERCHANDISE_CLASS
GROUP BY ITEM_MERCHANDISE_CLASS_ID);

DELETE FROM KPI_STG_ITEM_MERCHANDISE_COLLE
WHERE ROWID NOT IN (SELECT MIN(ROWID)
FROM KPI_STG_ITEM_MERCHANDISE_COLLE GROUP BY ITEM_MERCHANDISE_COLLECTION_ID);
```

```
DELETE FROM KPI_STG_ITEM_MERCHANDISE_DEPAR
WHERE ROWID NOT IN (SELECT MIN(ROWID)
FROM KPI_STG_ITEM_MERCHANDISE_DEPAR GROUP BY ITEM_MERCHANDISE_DEPARTMENT_ID);

DELETE FROM KPI_STG_ITEM_MERCHANDISE_SUBCL
WHERE ROWID NOT IN (SELECT MIN(ROWID)
FROM KPI_STG_ITEM_MERCHANDISE_SUBCL GROUP BY ITEM_MERCHANDISE_SUBCLASS_ID);

DELETE FROM KPI_STG_ITEMS
WHERE ROWID NOT IN
(SELECT MIN(ROWID)
FROM KPI_STG_ITEMS
GROUP BY ITEM_ID);

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);

DELETE FROM KPI_STG_LOCATIONS
WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM KPI_STG_LOCATIONS GROUP BY LOCATION_ID);

DELETE FROM KPI_STG_TRANSACTIONS
WHERE ROWID NOT IN (SELECT MIN(ROWID) FROM KPI_STG_TRANSACTIONS GROUP BY TRANSACTION_ID);
```

```
DELETE FROM KPI_STG_TRANSACTIONS_LINES
WHERE ROWID NOT IN (SELECT MIN(ROWID)
FROM KPI_STG_TRANSACTIONS_LINES
GROUP BY TRANSACTION_ID,TRANSACTION_LINE_ID);
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);
```

6. Create Primary Key on Stage tables. Provide the scripts used to create Primary Key.

CREATING PRIMARY_KEYS:

```
ALTER TABLE KPI_STG_CHANNEL ADD PRIMARY KEY (LIST_ID);
ALTER TABLE KPI_STG_CLASSES ADD PRIMARY KEY (CLASS_ID);
ALTER TABLE KPI_STG_DEPARTMENTS ADD PRIMARY KEY (DEPARTMENT_ID);
ALTER TABLE KPI_STG_ITEM_MERCHANDISE_CLASS ADD PRIMARY KEY (ITEM_MERCHANDISE_CLASS_ID);
ALTER TABLE KPI_STG_ITEM_MERCHANDISE_COLLE ADD PRIMARY KEY (ITEM_MERCHANDISE_COLLECTION_ID);
ALTER TABLE KPI_STG_ITEM_MERCHANDISE_DEPAR ADD PRIMARY KEY (ITEM_MERCHANDISE_DEPARTMENT_ID);
ALTER TABLE KPI_STG_ITEM_MERCHANDISE_SUBCL ADD PRIMARY KEY (ITEM_MERCHANDISE_SUBCLASS_ID);
ALTER TABLE KPI_STG_ITEMS ADD PRIMARY KEY (ITEM_ID);
ALTER TABLE KPI_STG_LOCATIONS ADD PRIMARY KEY (LOCATION_ID);
ALTER TABLE KPI_STG_TRANSACTIONS ADD PRIMARY KEY (TRANSACTION_ID);
ALTER TABLE KPI_STG_TRANSACTIONS_LINES ADD PRIMARY KEY (TRANSACTION_ID, RANSACTION_LINE_ID);
```

7. Identify the relationships between each table. Provide the SELECT SQLs executed to identify the relationships.

CREATING FOREIGN_KEYS:

```
ALTER TABLE KPI_STG_TRANSACTIONS ADD CONSTRAINT FK_KPI_STG_TRANSACTIONS

FOREIGN KEY(CHANNEL_ID) REFERENCES KPI_STG_CHANNEL(LIST_ID);


ALTER TABLE KPI_STG_TRANSACTIONS_LINES ADD CONSTRAINT FK_KPI_STG_TRANSACTIONS_LINES

FOREIGN KEY(LOCATION_ID) REFERENCES KPI_STG_LOCATIONS(LOCATION_ID);


ALTER TABLE KPI_STG_TRANSACTIONS_LINES ADD CONSTRAINT FK_KPI_TRANSACTIONS_LINES

FOREIGN KEY(DEPARTMENT_ID) REFERENCES KPI_STG_DEPARTMENTS(DEPARTMENT_ID);


ALTER TABLE KPI_STG_TRANSACTIONS_LINES ADD CONSTRAINT FK_STG_TRANSACTIONS_LINES

FOREIGN KEY(ITEM_ID) REFERENCES KPI_STG_ITEMS(ITEM_ID);


ALTER TABLE KPI_STG_ITEMS ADD CONSTRAINT FK_KPI_STG_ITEMS

FOREIGN KEY (CLASS_ID) REFERENCES KPI_STG_CLASSES CLASS_ID);


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);


ALTER TABLE KPI_STG_ITEMS ADD CONSTRAINT FK_K_STG_ITEMS

FOREIGN KEY(WS_MERCHANDISE_COLLECTION_ID) REFERENCES
KPI_STG_ITEM_MERCHANDISE_COLLE(ITEM_MERCHANDISE_COLLECTION_ID);


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);


ALTER TABLE KPI_STG_ITEMS ADD CONSTRAINT FK_KPI_S_ITEMS

FOREIGN KEY(WS_MERCHANDISE_SUBCLASS_ID) REFERENCES
KPI_STG_ITEM_MERCHANDISE_SUBCL(ITEM_MERCHANDISE_SUBCLASS_ID);
```

8. Create Target Tables.

8.1. 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_DEPTARMEN_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));
```

8.2. CREATE SEQUENCE to populate KPI_DW_SKEY field in all Target tables. Provide all the scripts.

CREATING SEQUENCE:

```
CREATE SEQUENCE T1;
UPDATE KPI_CHANNEL_DIM SET KPI_DW_SKEY=T1.NEXTVAL;
UPDATE KPI_CHANNEL_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_CHANNEL_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_CHANNEL_DIM;

CREATE SEQUENCE T2;
UPDATE KPI_CLASS_DIM SET KPI_DW_SKEY=T2.NEXTVAL;
UPDATE KPI_CLASS_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_CLASS_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_CLASS_DIM;
```

```
CREATE SEQUENCE T3;
UPDATE KPI_DEPARTMENT_DIM SET KPI_DW_SKEY=T3.NEXTVAL;
UPDATE KPI_DEPARTMENT_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_DEPARTMENT_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_DEPARTMENT_DIM;

CREATE SEQUENCE T4;
UPDATE KPI_ITEM_DIM SET KPI_DW_SKEY=T4.NEXTVAL;
UPDATE KPI_ITEM_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_ITEM_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_ITEM_DIM;


CREATE SEQUENCE T5;
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET KPI_DW_SKEY=T5.NEXTVAL;
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_ITEM_MERCHANDISE_CLASS_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_ITEM_MERCHANDISE_CLASS_DIM;

CREATE SEQUENCE T6;
UPDATE KPI_ITEM_MERCHANDISE_COL_DIM SET KPI_DW_SKEY=T6.NEXTVAL;
UPDATE KPI_ITEM_MERCHANDISE_COL_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_ITEM_MERCHANDISE_COL_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_ITEM_MERCHANDISE_COL_DIM;

CREATE SEQUENCE T7;
UPDATE KPI_ITEM_MERCHANDISE_DEPAR_DIM SET KPI_DW_SKEY=T7.NEXTVAL;
UPDATE KPI_ITEM_MERCHANDISE_DEPAR_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_ITEM_MERCHANDISE_DEPAR_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_ITEM_MERCHANDISE_DEPAR_DIM;
```

```
CREATE SEQUENCE T8;
UPDATE KPI_ITEM_MERCHANDISE_SUBCL_DIM SET KPI_DW_SKEY=T8.NEXTVAL;
UPDATE KPI_ITEM_MERCHANDISE_SUBCL_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_ITEM_MERCHANDISE_SUBCL_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_ITEM_MERCHANDISE_SUBCL_DIM;

CREATE SEQUENCE T9;
UPDATE KPI_LOCATION_DIM SET KPI_DW_SKEY=T9.NEXTVAL;
UPDATE KPI_LOCATION_DIM SET KPI_DW_INSERT_DATE = SYSDATE;
UPDATE KPI_LOCATION_DIM SET KPI_DW_UPDATE_DATE = SYSDATE;
SELECT * FROM KPI_LOCATION_DIM;

CREATE SEQUENCE T10;
UPDATE KPI_TRANSACTION_LINE_FACT SET KPI_DW_SKEY=T10.NEXTVAL;
SELECT * FROM KPI_TRANSACTION_LINE_FACT;
```

8.3. Create PRIMARY KEY on KPI_DW_SKEY.

ADDING PRIMARY KEYS:

```
ALTER TABLE KPI_CHANNEL_DIM ADD PRIMARY KEY(KPI_DW_SKEY);
DESC KPI_CHANNEL_DIM;

ALTER TABLE KPI_CLASS_DIM ADD PRIMARY KEY(KPI_DW_SKEY);
DESC KPI_CLASS_DIM;

ALTER TABLE KPI_DEPARTMENT_DIM ADD PRIMARY KEY(KPI_DW_SKEY);
DESC KPI_DEPARTMENT_DIM;

ALTER TABLE KPI_ITEM_DIM ADD PRIMARY KEY(KPI_DW_SKEY);
DESC KPI_ITEM_DIM;

ALTER TABLE KPI_ITEM_MERCHANDISE_CLASS_DIM ADD PRIMARY KEY(KPI_DW_SKEY);
DESC KPI_ITEM_MERCHANDISE_CLASS_DIM;
```

```
ALTER TABLE KPI_ITEM_MERCHANDISE_COL_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_ITEM_MERCHANDISE_COL_DIM;


ALTER TABLE KPI_ITEM_MERCHANDISE_DEPAR_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_ITEM_MERCHANDISE_DEPAR_DIM;


ALTER TABLE KPI_ITEM_MERCHANDISE_SUBCL_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_ITEM_MERCHANDISE_SUBCL_DIM;


ALTER TABLE KPI_LOCATION_DIM ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_LOCATION_DIM;


ALTER TABLE KPI_TRANSACTION_LINE_FACT ADD PRIMARY KEY(KPI_DW_SKEY);

DESC KPI_TRANSACTION_LINE_FACT;
```

9. Target tables load. Load the Target Tables using Stage Tables.

9. 1. Identify the sequence in which the Target Tables has to be loaded. Provide the reasons.

We have 2 databases here SOURCEDB(stage tables) AND TARGETDB(target tables). In the sourcedb(stage tables) we have data in order to load the data from sourcedb to targetdb. We need to use the ‘connect user’ command to connect to the sourcedb and need to give the permission to grant connection to targetdb and connect to the targetdb to load the data from the sourcedb.

```
SHOW USER;

GRANT CONNECT,RESOURCE TO SOURCEDB;

GRANT SELECT ON KPI_STG_CHANNEL TO TARGETDB;

GRANT SELECT ON KPI_STG_CLASSES TO TARGETDB;

GRANT SELECT ON KPI_STG_DEPARTMENTS TO TARGETDB;

GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_CLASS TO TARGETDB;

GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_COLLE TO TARGETDB;

GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_DEPAR TO TARGETDB;

GRANT SELECT ON KPI_STG_ITEM_MERCHANDISE_SUBCL TO TARGETDB;

GRANT SELECT ON KPI_STG_ITEMS TO TARGETDB;

GRANT SELECT ON KPI_STG_LOCATIONS TO TARGETDB;

GRANT SELECT ON KPI_STG_TRANSACTIONS TO TARGETDB;
```

```
GRANT SELECT ON KPI_STG_TRANSACTIONS_LINES TO TARGETDB;

GRANT CONNECT,RESOURCE TO TARGETDB;
```

9. 2. Provide the INSERT scripts used to perform the data load.

INSERT scripts used to perform the data load:

```
INSERT INTO KPI_CHANNEL_DIM(DATE_CREATED,

IS_RECORD_INACTIVE,

LAST_MODIFIED_DATE,

LIST_ID,

LIST_ITEM_NAME)(SELECT * FROM SOURCEDB.KPI_STG_CHANNEL);

--5 ROWS INSERTED.


INSERT INTO KPI_CLASS_DIM(CLASS_ID,DATE_LAST_MODIFIED,

FULL_NAME,ISINACTIVE,NAME)

(SELECT * FROM SOURCEDB.KPI_STG_CLASSES);

--6 ROWS INSERTED.


INSERT INTO KPI_DEPARTMENT_DIM(DATE_LAST_MODIFIED,

DEPARTMENT_ID,ISINACTIVE,NAME,WS_DESCRIPTION)

(SELECT * FROM SOURCEDB.KPI_STG_DEPARTMENTS);

--105 ROWS INSERTED.


INSERT INTO KPI_ITEM_MERCHANDISE_CLASS_DIM(ITEM_MERCHANDISE_CLASS_ID,

DESCRIPTION,ITEM_MERCHANDISE_CLASS_NAME)

(SELECT * FROM SOURCEDB.KPI_STG_ITEM_MERCHANDISE_CLASS);

--83 ROWS INSERTED.


INSERT INTO KPI_ITEM_MERCHANDISE_COL_DIM(ITEM_MERCHANDISE_COLLECTION_ID,

DESCRIPTION,ITEM_MERCHANDISE_COLLECTION_NA)

(SELECT * FROM SOURCEDB.KPI_STG_ITEM_MERCHANDISE_COLLE);

--86 ROWS INSERTED.
```

```
INSERT INTO KPI_ITEM_MERCHANDISE_DEPAR_DIM(ITEM_MERCHANDISE_DEPARTMENT_ID,
DESCRIPTION,ITEM_MERCHANDISE_DEPARTMENT_NA)
(SELECT * FROM SOURCEDB.KPI_STG_ITEM_MERCHANDISE_DEPAR);
--87 ROWS INSERTED.

INSERT INTO KPI_ITEM_MERCHANDISE_SUBCL_DIM(ITEM_MERCHANDISE_SUBCLASS_ID,
DESCRIPTION,ITEM_MERCHANDISE_SUBCLASS_NAME)
(SELECT * FROM SOURCEDB.KPI_STG_ITEM_MERCHANDISE_SUBCL);
--85 ROWS INSERTED.

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 SOURCEDB.KPI_STG_ITEMS);
--88 ROWS INSERTED.

INSERT INTO KPI_LOCATION_DIM(LOCATION_ID,
ADDRESS,CITY,COUNTRY,DATE_LAST_MODIFIED,
FULL_NAME,ISINACTIVE,NAME)
(SELECT * FROM SOURCEDB.KPI_STG_LOCATIONS);
--114 ROWS INSERTED.

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 SOURCEDB.KPI_STG_TRANSACTIONS A,

SOURCEDB.KPI_STG_TRANSACTIONS_LINES B WHERE B.TRANSACTION_ID = A.TRANSACTION_ID);

--2,758 ROWS INSERTED.
```

10. CREATE BRAND_NAME field in KPI_ITEM_DIM and populate values from NAME field present in KPI_CLASS_DIM.

10.1. Provide the script to add the new column.

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

10.2. Provide the UPDATE script to populate BRAND_NAME field.

```
UPDATE KPI_ITEM_DIM SET BRAND_NAME=(SELECT NAME

FROM KPI_CLASS_DIM WHERE KPI_ITEM_DIM.KPI_DW_SKEY=KPI_CLASS_DIM.KPI_DW_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. ITEMS.NAME AS SKU
2. ITEMS.TYPE_NAME AS ITEM_TYPE
3. ITEMS.BRAND_NAME AS BRAND
4. ITEM_MERCHANDISE_DEPARTMENT.DESCRIPTION AS MERCHANDISE_DEPARTMENT
5. ITEM_MERCHANDISE_DEPARTMENT.ITEM_MERCHANDISE_DEPARTMENT_NA AS MERCHANDISE_DEPT_NAME
6. ITEM_MERCHANDISE_COLLECTION.DESCRIPTION AS MERCHANDISE_COLLECTION
7.ITEM_MERCHANDISE_COLLECTION.ITEM_MERCHANDISE_COLLECTION_NA MERCHANDISE_COLLECTION_NAME
8. ITEM_MERCHANDISE_CLASS.DESCRIPTION AS MERCHANDISE_CLASS
9. ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLASS_NAME AS MERCHANDISE_CLASS_NAME
10. ITEM_MERCHANDISE_SUBCLASS.DESCRIPTION AS MERCHANDISE_SUBCLASS
11. ITEM_MERCHANDISE_SUBCLASS.ITEM_MERCHANDISE_SUBCLASS_NAME AS MERCHANDISE_SUBCLASS_NAME
12. ITEMS.KPI_DW_SKEY as KPI_ITEM_SKEY

11. 1. Provide the CREATE script.

```
CREATE TABLE KPI_ITEM_DIM_FLAT(SKU VARCHAR2(100),

ITEM_TYPE VARCHAR2(100),

BRAND VARCHAR2(100),

MERCHANDISE_DEPARTMENT VARCHAR2(200),

MERCHANDISE_DEPT_NAME VARCHAR2(100),

MERCHANDISE_COLLECTION VARCHAR2(200),
```

MERCHANDISE_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);

11.2. Provide the BULK INSERT script to load this table.

BULK INSERT script to load into INSERT INTO KPI_ITEM_DIM_FLAT 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.DESCRIPTION,ITEM_MERCHANDISE_DEPARTMENT.ITEM_MERCHANDISE_DEPARTMENT_NA,

ITEM_MERCHANDISE_COLLECTION.DESCRIPTION,ITEM_MERCHANDISE_COLLECTION.ITEM_MERCHANDISE_COLLECTION_NA,

ITEM_MERCHANDISE_CLASS.DESCRIPTION,ITEM_MERCHANDISE_CLASS.ITEM_MERCHANDISE_CLASS_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);

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

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_SUBCLA
SS,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_MERCHA
NDISE_COLLECTION_NA,

C.DESCRIPTION,C.ITEM_MERCHANDISE_CLASS_NAME,S.DESCRIPTION,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. Answer the requested questions.

12.1. 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;
```

12.2. 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';
```

12.4. 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;
```

12.5. 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;
```

12.6. Write a VIEW using target tables with following fields

- TRANSACTION_ID
- TRANSACTION_LINE_ID
- TRANDATE

- TRANSACTION_TYPE
- ITEM_NAME
- ITEM_TYPE_NAME
- LOCATION_NAME
- DEPARTMENT_NAME
- CHANNEL_NAME
- MERCHANDISE_DEPARTMENT_NAME
- MERCHANDISE_DEPARTMENT_DESCRIPTION
- MERCHANDISE_COLLECTION_NAME
- MERCHANDISE_COLLECTION_DESCRIPTION
- MERCHANDISE_CLASS_NAME
- MERCHANDISE_CLASS_DESCRIPTION
- MERCHANDISE_SUBCLASS_NAME
- MERCHANDISE_SUBCLASS_DESCRIPTION
- Demand_Amount
- Demand_Units
- Demand_Profit
- Demand_Profit%
- Sales_Amount
- Sales_Units
- Sales_Profit
- Sales_Profit%

```
CREATE FORCE VIEW TARGET_VIEW AS SELECT T.TRANSACTION_ID, T.TRANSACTION_LINE_ID, T.TRANDATE,
T.TRANSACTION_TYPE,I.TYPE_NAME, L.CITY, D.NAME, CD.LIST_ITEM_NAME, ID.ITEM_MERCH_DEPARTMENT_NA,

ID.DESCRIPTION,IC.ITEM_MERCH_COLLECTION_NA,IC.DESCRIPTION,C.ITEM_MERCH_CLASS_NAME,

C.DESCRIPTION,S.ITEM_MERCH_SUBCLASS_NAME,S.DESCRIPTION,T.AMOUNT,T.UNITS

FROM TRANSACTION_LINE_FACT T JOIN ITEM_DIM I ON T.KPI_DW_SKEY = I.KPI_DW_SKEY

JOIN LOCATION_DIM L ON I.KPI_DW_SKEY = L.KPI_DW_SKEY

JOIN DEPARTMENT_DIM D ON L.KPI_DW_SKEY = D.KPI_DW_SKEY

JOIN CHANNEL_DIM CD ON D.KPI_DW_SKEY = CD.KPI_DW_SKEY

JOIN ITEM_MERCH_DEPARTMENT_DIM ID ON CD.KPI_DW_SKEY = ID.KPI_DW_SKEY

JOIN ITEM_MERCH_COLLECTION_DIM IC ON ID.KPI_DW_SKEY = IC.KPI_DW_SKEY

JOIN ITEM_MERCH_CLASS_DIM C ON IC.KPI_DW_SKEY = C.KPI_DW_SKEY

JOIN ITEM_MERCH_SUBCLASS_DIM S ON C.KPI_DW_SKEY = S.KPI_DW_SKEY;
```