Kalai Chelvan kalai.chelvan@accolitedigital.com

1. Assuming you are ready with ER Model (from Morning session Assignment), transform it into a Database schema. Create tables keeping up good practices and send me the create scripts you've written.
2. Write a query to retrieve the most sold product per day in a specific location (take any location) in last week.
3. Write a query to list all the sales persons details along with the count of products sold by them (if any) till current date.

Note: Along with the queries you've written, attach screenshots of the output for Q's 2 & 3.

1. Create scripts attached in this folder itself
    a. *AUTables.sql* - holds all the create and alter foreign key statements
    b. *AUInserts.sql* – holds all the tuple insert statements
    c. *AUQueries*.sql – holds all the queries to be executed

Note: Since int and Bigint holds more memory, I have used varchar(10) to store the mobile numbers

```
-- Create tables with primary key and some check constraints
--      check contraint checks whether the gender is in either of the 3 mentioned type
CREATE TABLE PRODUCT(
    PROD_CODE VARCHAR(5) PRIMARY KEY,
    PROD_NAME VARCHAR(15),
    CAT_CODE VARCHAR(5)
);

CREATE TABLE CATEGORY(
    CAT_CODE VARCHAR(5) PRIMARY KEY,
    CAT_NAME VARCHAR(15)
);

CREATE TABLE CUSTOMER(
    CUST_ID VARCHAR(5) PRIMARY KEY,
    CUST_NAME VARCHAR(20),
    CUST_DOB DATE,
    CUST_GENDER CHAR(1),
    CUST_MOBILE VARCHAR(10),
    LOC_CODE VARCHAR(5),
    CONSTRAINT CHECK_CUSTOMER_GENDER CHECK(CUST_GENDER IN ('M','F','O'))
);

CREATE TABLE SALES_EXE(
    SE_ID VARCHAR(5) PRIMARY KEY,
    SE_NAME VARCHAR(20),
    SE_DOB DATE,
    SE_GENDER CHAR(1),
    SE_MOBILE VARCHAR(10),
```

```
    CONSTRAINT CHECK_SE_GENDER CHECK(SE_GENDER IN('M','F','O'))
);

CREATE TABLE LOCATION(
    LOC_CODE VARCHAR(5) PRIMARY KEY,
    LOC_NAME VARCHAR(15)
);

CREATE TABLE SALES(
    SE_ID VARCHAR(5),
    CUST_ID VARCHAR(5),
    PROD_CODE VARCHAR(5),
    DOP DATE,
    NOU INT,
    PRIMARY KEY(SE_ID, CUST_ID, PROD_CODE, DOP)
);

-- Updating Foreign keys to the tables
ALTER TABLE PRODUCT ADD FOREIGN KEY (CAT_CODE) REFERENCES CATEGORY(CAT_CODE);

ALTER TABLE CUSTOMER ADD FOREIGN KEY (LOC_CODE) REFERENCES LOCATION(LOC_CODE);

ALTER TABLE SALES ADD FOREIGN KEY (SE_ID), REFERENCES SALES_EXE(SE_ID);
ALTER TABLE SALES ADD FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER(CUST_ID);
ALTER TABLE SALES ADD FOREIGN KEY (PROD_CODE) REFERENCES PRODUCT(PROD_CODE);
```

2 & 3 Queries

```
-- View to find all the sales done in the past 7 days
CREATE VIEW LAST_WEEK_SALES AS SELECT * FROM SALES
WHERE DOP BETWEEN date_sub(current_date(),interval 7 DAY) and current_date();

-- Write a query to retrieve the most sold product per day in a specific location
-- (take any location) in last week.
select P.PROD_CODE, P.PROD_NAME, T.DOP AS DATE_OF_SALE, max(TOT_UNITS) AS MAX_UNITS_SOLD_P
ERDAY from PRODUCT P
INNER JOIN
(SELECT sum(NOU) as TOT_UNITS, PROD_CODE, DOP
    FROM LAST_WEEK_SALES S,CUSTOMER C
    WHERE C.LOC_CODE="L001"
    AND C.CUST_ID=S.CUST_ID
    GROUP BY DOP,PROD_CODE) T ON T.PROD_CODE=P.PROD_CODE;

-- Write a query to list all the sales persons details along with the
-- count of products sold by them (if any) till current date.
SELECT SE.*, IFNULL(T.PRODUCTS_SOLD, 0) AS NO_OF_PRODUCTS_SOLD, IFNULL(T.NO_OF_UNITS_SOLD,
 0) AS NO_OF_UNITS_SOLD FROM SALES_EXE SE
LEFT JOIN
(SELECT SE_ID, COUNT(DISTINCT PROD_CODE) PRODUCTS_SOLD, SUM(NOU) AS NO_OF_UNITS_SOLD FROM
SALES GROUP BY SE_ID) T
ON T.SE_ID=SE.SE_ID;
```

## Screenshots: