SQL Concepts **Spring AU’21**

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.

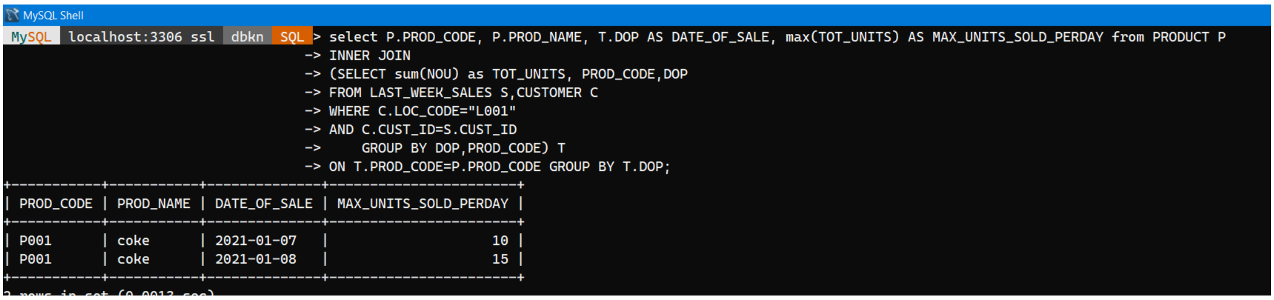
1. Create scripts attached in this folder itself
   a. *AUTables.sql* - holds all the create and alter foreign key statements
   b. *AUInserts.sql* – holds all the tuple insert statements
   c. *AUQueries*.sql – holds all the queries to be executed

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

```sql
-- 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),
    UNIT_PRICE FLOAT(5,3)
);

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),
    LOC_CODE VARCHAR(5),
    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 SALE(
    SALE_ID VARCHAR(5) PRIMARY KEY,
    SE_ID VARCHAR(5),
    CUST_ID VARCHAR(5),
    DOP DATE,
    LOC_CODE VARCHAR(5)
);

CREATE TABLE SALE_PRODUCT(
    SALE_ID VARCHAR(5),
    PROD_CODE VARCHAR(5),
    NOP INT,
    PRIMARY KEY(SALE_ID, PROD_CODE)
);

-- 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_EXE ADD FOREIGN KEY (LOC_CODE) REFERENCES LOCATION(LOC_CODE);
ALTER TABLE SALE ADD FOREIGN KEY (SE_ID) REFERENCES SALES_EXE(SE_ID);
ALTER TABLE SALE ADD FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER(CUST_ID);
ALTER TABLE SALE_PRODUCT ADD FOREIGN KEY (SE_ID) REFERENCES SALES_EXE(SE_ID);
ALTER TABLE SALE_PRODUCT ADD FOREIGN KEY (PROD_CODE) REFERENCES PRODUCT(PROD_CODE);
```

Query 1:

```
-- 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 GROUP BY T.DOP;
```



```
 MySQL  localhost:3306 ssl  dbkn  SQL > select P.PROD_CODE, P.PROD_NAME, T.DOP AS DATE_OF_SALE, max(TOT_UNITS) AS MAX_UNITS_SOLD_PERDAY 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 GROUP BY T.DOP;
+-----------+-----------+--------------+-----------------------+
| PROD_CODE | PROD_NAME | DATE_OF_SALE | MAX_UNITS_SOLD_PERDAY |
+-----------+-----------+--------------+-----------------------+
| P001      | coke      | 2021-01-07   |                    10 |
| P001      | coke      | 2021-01-08   |                    15 |
+-----------+-----------+--------------+-----------------------+
2 rows in set (0.0013 sec)
```

In the above query the max units retrieved correctly but the respective product is not, it returns the first product in each group and also retrieves only one record per day, even if there are two products have the same max value. In-order to get that I have altered query and some schema changes.

Q1.

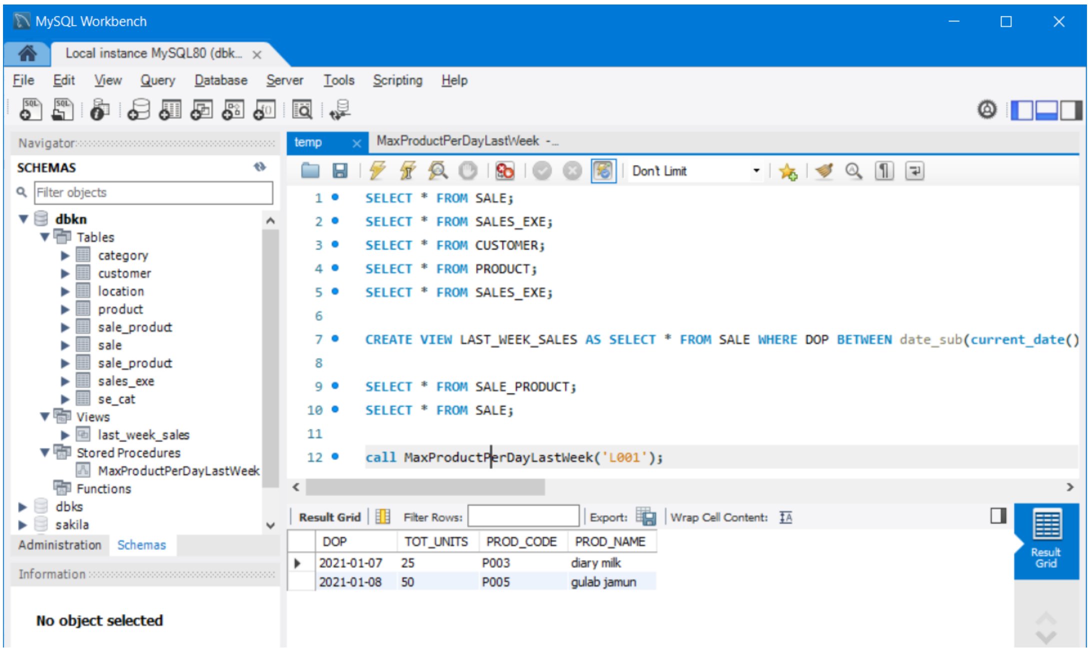
```
SELECT RESF.DOP, RESF.TOT_UNITS, RESF.PROD_CODE, P.PROD_NAME FROM
(SELECT  DOP, PROD_CODE, SALE_ID, sum(NOU) as TOT_UNITS
        FROM( SELECT S.DOP,S.SALE_ID,SP.PROD_CODE,SP.NOU FROM LAST_WEEK_SALES S, SALE_PRODUCT SP WHERE S.SALE_I
D=SP.SALE_ID AND LOC_CODE="L001") AS RES1
        GROUP BY DOP, PROD_CODE ORDER BY DOP) AS RESF
INNER JOIN
    (SELECT RES2.DOP AS DOP12, MAX(TOT_UNITS) AS TOT_UNITS12 FROM
        (SELECT  DOP, PROD_CODE, SALE_ID, sum(NOU) as TOT_UNITS
        FROM( SELECT S.DOP,S.SALE_ID,SP.PROD_CODE,SP.NOU FROM LAST_WEEK_SALES S, SALE_PRODUCT SP WHERE S.SALE_I
D=SP.SALE_ID AND LOC_CODE="L001") AS RES1
        GROUP BY DOP, PROD_CODE ORDER BY DOP) AS RES2
    GROUP BY RES2.DOP) AS RES3
ON RES3.DOP12=RESF.DOP AND RES3.TOT_UNITS12=RESF.TOT_UNITS
,
PRODUCT P
WHERE P.PROD_CODE=RESF.PROD_CODE;
```

To avoid repeated subqueries and change location, implemented a procedure as below

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `MaxProductPerDayLastWeek`(IN locCode VARCHAR(5))
BEGIN
    -- get all max prodcut sold per day for the last week
    CREATE TEMPORARY TABLE IF NOT EXISTS  MaxProductPerDayLastWeek2(
        DOP DATE, PROD_CODE VARCHAR(5), SALE_ID VARCHAR(5), TOT_UNITS INT
    );
    CREATE TEMPORARY TABLE IF NOT EXISTS  MaxProductPerDayLastWeek3(
        DOP12 DATE, TOT_UNITS12 INT
    );
    TRUNCATE TABLE MaxProductPerDayLastWeek2;
    TRUNCATE TABLE MaxProductPerDayLastWeek3;

    INSERT INTO MaxProductPerDayLastWeek2 SELECT DOP, PROD_CODE, SALE_ID,sum(NOU) AS TOT_UNITS
    FROM (SELECT S.DOP,S.SALE_ID,SP.PROD_CODE,SP.NOU FROM LAST_WEEK_SALES S, SALE_PRODUCT SP WHERE S.SALE_ID=SP
.SALE_ID AND LOC_CODE=locCode) AS RES1
    GROUP BY DOP, PROD_CODE ORDER BY DOP;

    INSERT INTO MaxProductPerDayLastWeek3 SELECT RES2.DOP AS DOP12, MAX(RES2.TOT_UNITS) AS TOT_UNITS12
    FROM MaxProductPerDayLastWeek2 AS RES2
    GROUP BY RES2.DOP;

    SELECT DOP, TOT_UNITS, P.PROD_CODE, P.PROD_NAME FROM MaxProductPerDayLastWeek2 AS RES2
    INNER JOIN MaxProductPerDayLastWeek3 AS RES3
    ON RES3.DOP12=DOP AND RES3.TOT_UNITS12=TOT_UNITS
    ,
    PRODUCT P
    WHERE P.PROD_CODE=RES2.PROD_CODE;
END
```

```
Call MaxProductPerDayLastWeek(‘L001’);
```

Screenshots:



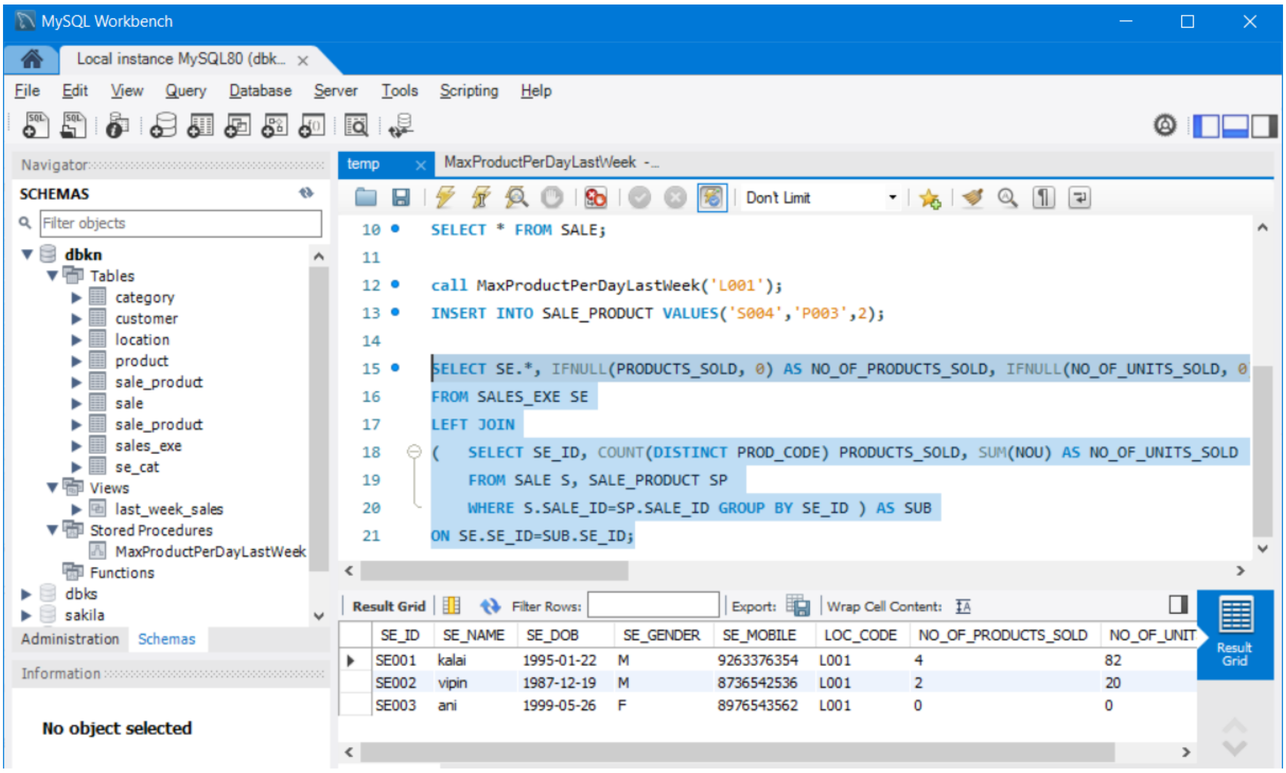
```
SELECT * FROM SALE;
SELECT * FROM SALES_EXE;
SELECT * FROM CUSTOMER;
SELECT * FROM PRODUCT;
SELECT * FROM SALES_EXE;

CREATE VIEW LAST_WEEK_SALES AS SELECT * FROM SALE WHERE DOP BETWEEN date_sub(current_date()

SELECT * FROM SALE_PRODUCT;
SELECT * FROM SALE;

call MaxProductPerDayLastWeek('L001');
```

DOP	TOT_UNITS	PROD_CODE	PROD_NAME
2021-01-07	25	P003	diary milk
2021-01-08	50	P005	gulab jamun

Query 2:

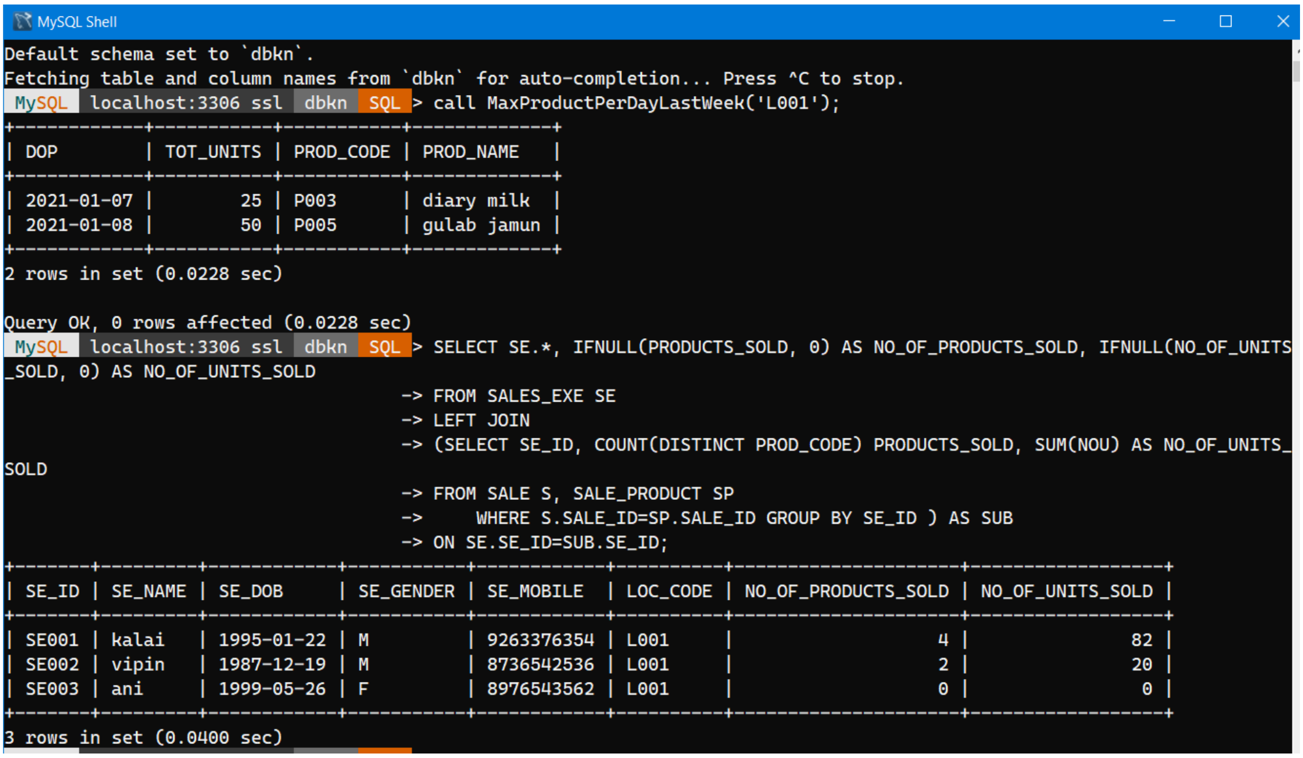
```
-- Write a query to list all the sales persons details along with the count of products
-- sold by them (if any) till current date.
SELECT SE.*, IFNULL(PRODUCTS_SOLD, 0) AS NO_OF_PRODUCTS_SOLD, IFNULL(NO_OF_UNITS_SOLD, 0) AS NO_OF_UNITS_SOLD
FROM SALES_EXE SE
LEFT JOIN
(   SELECT SE_ID, COUNT(DISTINCT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

Screenshots:



```sql
SELECT * FROM SALE;

call MaxProductPerDayLastWeek('L001');
INSERT INTO SALE_PRODUCT VALUES('S004','P003',2);

SELECT SE.*, IFNULL(PRODUCTS_SOLD, 0) AS NO_OF_PRODUCTS_SOLD, IFNULL(NO_OF_UNITS_SOLD, 0
FROM SALES_EXE SE
LEFT JOIN
(   SELECT SE_ID, COUNT(DISTINCT PROD_CODE) PRODUCTS_SOLD, SUM(NOU) AS NO_OF_UNITS_SOLD
    FROM SALE S, SALE_PRODUCT SP
    WHERE S.SALE_ID=SP.SALE_ID GROUP BY SE_ID ) AS SUB
ON SE.SE_ID=SUB.SE_ID;
```

SE_ID	SE_NAME	SE_DOB	SE_GENDER	SE_MOBILE	LOC_CODE	NO_OF_PRODUCTS_SOLD	NO_OF_UNIT
SE001	kalai	1995-01-22	M	9263376354	L001	4	82
SE002	vipin	1987-12-19	M	8736542536	L001	2	20
SE003	ani	1999-05-26	F	8976543562	L001	0	0



```
Default schema set to `dbkn`.
Fetching table and column names from `dbkn` for auto-completion... Press ^C to stop.
 MySQL  localhost:3306 ssl  dbkn  SQL > call MaxProductPerDayLastWeek('L001');
+------------+-----------+-----------+-------------+
| DOP        | TOT_UNITS | PROD_CODE | PROD_NAME   |
+------------+-----------+-----------+-------------+
| 2021-01-07 |        25 | P003      | diary milk  |
| 2021-01-08 |        50 | P005      | gulab jamun |
+------------+-----------+-----------+-------------+
2 rows in set (0.0228 sec)

Query OK, 0 rows affected (0.0228 sec)
 MySQL  localhost:3306 ssl  dbkn  SQL > SELECT SE.*, IFNULL(PRODUCTS_SOLD, 0) AS NO_OF_PRODUCTS_SOLD, IFNULL(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 SALE S, SALE_PRODUCT SP
                                     ->     WHERE S.SALE_ID=SP.SALE_ID GROUP BY SE_ID ) AS SUB
                                     -> ON SE.SE_ID=SUB.SE_ID;
+-------+---------+------------+-----------+------------+----------+---------------------+------------------+
| SE_ID | SE_NAME | SE_DOB     | SE_GENDER | SE_MOBILE  | LOC_CODE | NO_OF_PRODUCTS_SOLD | NO_OF_UNITS_SOLD |
+-------+---------+------------+-----------+------------+----------+---------------------+------------------+
| SE001 | kalai   | 1995-01-22 | M         | 9263376354 | L001     |                   4 |               82 |
| SE002 | vipin   | 1987-12-19 | M         | 8736542536 | L001     |                   2 |               20 |
| SE003 | ani     | 1999-05-26 | F         | 8976543562 | L001     |                   0 |                0 |
+-------+---------+------------+-----------+------------+----------+---------------------+------------------+
3 rows in set (0.0400 sec)
```