Activity No. 5	
Advanced Database Programming	
Course Code: CPE011	**Program:BSCPE**
Course Title: Database Management System	**Date Performed: 16/09/2022**
Section: S3	**Date Submitted: 16/09/2022**
Name: **CARO, AARON MARTIN** **EFA, CHRISTIAN ED**	**Instructor: Engr. Jonathan Taylar**

1. Objective(s):

This activity aims to introduce the Structured Query Language (SQL) using the Data Definition Language (DDL) commands in a MySQL Database

2. Intended Learning Outcomes (ILOs):

The students should be able to:
2.1 Use SQL Built-in functions in an SQL Script
2.2 Use JOIN SQL Commands to perform complex queries in databases
2.2 Create subqueries and correlated queries
.

3. Discussion:

As your databases becomes more complex and more relations between entities are created, we would need to implement queries that can retrieve records from multiple tables and display them in one single query. Some queries may require outputs from nested queries also called subqueries in order to get more specific results. Some tasks such as date generation, and summing quantities of products can also be automated through the use of SQL built-in functions similar to a programming language built-in function.

SQL Function

SQL Functions

An SQL function is similar to a function in programming language both in syntax and in its general use. A function in SQL can be defined with the following syntax function_name(). The opening and closing parentheses is the main indicator that it is a function. There are many built-in SQL Functions available. Some of the possible SQL functions are given below:

SQL Function	Description
COUNT()	Returns the number of the rows in a table.
SUM()	Returns the sum of a set of values. The SUM function ignores NULL values. If no matching row found, the SUM function returns a NULL value.
AVG()	Calculates the average value of a set of values. It ignores NULL values in the calculation.
MAX()	Returns the maximum value in a set of values.
MIN()	Returns the minimum value in a set of values.
ROUND(number, decimals)	Rounds a number to a specified number of decimal places.
ABS()	Return the absolute value of a number
CURRENT_DATE()	Returns the current date
NOW()	Returns the current date and time

More functions can be found through the official documentation:

https://dev.mysql.com/doc/refman/5.6/en/functions.html Sample use:

ID	NAME	AGE	AMOUNT
3	Kaushik	23	3000
3	Kaushik	23	1500
2	Khilan	25	1560
4	Chaitali	25	2060

Here, it is noticeable that the join is performed in the WHERE clause. Several operators can be used to join tables, such as =, <, >,
<>, <=, >=, !=, BETWEEN, LIKE, and NOT; they can all be used to join tables. However, the most common operator is the equal symbol.

4. Materials and Equipment:

- Desktop Computer
- Windows Operating System
- XAMPP Application

5. Procedure

1. CREATE DATABASE myDatabase5_<LASTNAME>.

```
Query OK, 0 rows affected (0.010 sec)

MariaDB [mydatabase5_efa_caro]>
```

2. Create three tables named Item_List, Nanays,Kaloys and Supplier.

Item List		
PID	**Product**	**supID**
p01	Beef Steak	mla2
p02	Beef Cutlet	mla1
p03	Pork Steak	mla1
p04	Chicken Wings	mla4
p05	Ground Pork	mla6
p06	Chicken Liver	mla7
p07	Beef Liver	mla1
p08	Chicken Feet	mla6
p09	Chicken Skin	mla5
p10	Whole Pork	mla5
p11	Whole Chicken	mla1
p12	Whole Cow	mla2
p13	Young Pork	mla6
p14	One Day Old	mla4
p15	Beef Franks	mla2

Kaloys Diner	
PID	**QTY**
p01	1
p04	5
p06	4
p10	4
p12	1
p14	6

Nanay's Eatery	
PID	**QTY**
p04	5
p09	4
p13	3

SupplierList	
supID	**Location**
mla1	Navotas
mla2	Navotas
mla3	Marikina
mla4	Marikina
mla5	Paranaque
mla6	Paranaque
mla7	Quezon City
mla8	Quezon City
mla9	Zapote
mla10	Zapote

```
MariaDB [mydatabase5_efa_caro]> INSERT INTO ITEM_LIST(PID,PRODUCT,SUPID)
    -> VALUES ('P01','BEEF STEAK','MLA2'),
    -> ('P02','BEEF CUTLET','MLA1'),
    -> ('P03','PORK STEAK','MLA1'),
    -> ('P04','CHICKEN WINGS','MLA4'),
    -> ('P05','GROUND PORK','MLA6'),
    -> ('P06','CHICKEN LIVER','MLA7'),
    -> ('P07','BEEF LIVER','MLA1'),
    -> ('P08','CHICKEN FEET','MLA6'),
    -> ('P09','CHICKEN SKIN','MLA5'),
    -> ('P10','WHOLE PORK','MLA5'),
    -> ('P11','WHOLE CHICKEN','MLA1'),
    -> ('P12','WHOLE COW','MLA2'),
    -> ('P13','YOUNG PORK','MLA6'),
    -> ('P14','ONE DAY OLD','MLA4'),
    -> ('P15','BEEF FRANKS','MLA2');
Query OK, 15 rows affected (0.002 sec)
Records: 15  Duplicates: 0  Warnings: 0

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> SELECT * FROM ITEM_LIST;
+-----+---------------+-------+
| PID | PRODUCT       | SUPID |
+-----+---------------+-------+
| P01 | BEEF STEAK    | MLA2  |
| P02 | BEEF CUTLET   | MLA1  |
| P03 | PORK STEAK    | MLA1  |
| P04 | CHICKEN WINGS | MLA4  |
| P05 | GROUND PORK   | MLA6  |
| P06 | CHICKEN LIVER | MLA7  |
| P07 | BEEF LIVER    | MLA1  |
| P08 | CHICKEN FEET  | MLA6  |
| P09 | CHICKEN SKIN  | MLA5  |
| P10 | WHOLE PORK    | MLA5  |
| P11 | WHOLE CHICKEN | MLA1  |
| P12 | WHOLE COW     | MLA2  |
| P13 | YOUNG PORK    | MLA6  |
| P14 | ONE DAY OLD   | MLA4  |
| P15 | BEEF FRANKS   | MLA2  |
+-----+---------------+-------+
15 rows in set (0.001 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> CREATE TABLE KALOYS_DINER(
    -> PID VARCHAR(100) PRIMARY KEY, QTY INT);
Query OK, 0 rows affected (0.018 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> INSERT INTO KALOYS_DINER(PID,QTY)
    -> VALUES('P01',1),
    -> ('P04',5),
    -> ('P06',4),
    -> ('P10',4),
    -> ('P12',1),
    -> ('P14',6);
Query OK, 6 rows affected (0.006 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

```
MariaDB [mydatabase5_efa_caro]> SELECT * FROM KALOYS_DINER;
+-----+------+
| PID | QTY  |
+-----+------+
| P01 |    1 |
| P04 |    5 |
| P06 |    4 |
| P10 |    4 |
| P12 |    1 |
| P14 |    6 |
+-----+------+
6 rows in set (0.000 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> CREATE TABLE NANAYS_EATERY(
    -> PID VARCHAR(100) PRIMARY KEY, QTY INT);
Query OK, 0 rows affected (0.018 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> INSERT INTO NANAYS_EATERY(PID,QTY)
    -> VALUES('P04',5),
    -> ('P09',4),
    -> ('P13',3);
Query OK, 3 rows affected (0.003 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> SELECT * FROM NANAYS_EATERY;
+-----+------+
| PID | QTY  |
+-----+------+
| P04 |    5 |
| P09 |    4 |
| P13 |    3 |
+-----+------+
3 rows in set (0.000 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> CREATE TABLE SUPPLIERLIST(
    -> SUPID VARCHAR(100),
    -> LOCATION VARCHAR(100));
Query OK, 0 rows affected (0.018 sec)

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> INSERT INTO SUPPLIERLIST(SUPID,LOCATION)
    -> VALUES
    -> ('MLA1','NAVOTAS'),
    -> ('MLA2','NAVOTAS'),
    -> ('MLA3','MARIKINA'),
    -> ('MLA4','MARIKINA'),
    -> ('MLA5','PARANAQUE'),
    -> ('MLA6','PARANAQUE'),
    -> ('MLA7','QUEZON CITY'),
    -> ('MLA8','QUEZON CITY'),
    -> ('MLA9','ZAPOTE'),
    -> ('MLA10','ZAPOTE');
Query OK, 10 rows affected (0.003 sec)
Records: 10  Duplicates: 0  Warnings: 0

MariaDB [mydatabase5_efa_caro]>
```

```
MariaDB [mydatabase5_efa_caro]> SELECT * FROM SUPPLIERLIST;
+-------+-------------+
| SUPID | LOCATION    |
+-------+-------------+
| MLA1  | NAVOTAS     |
| MLA2  | NAVOTAS     |
| MLA3  | MARIKINA    |
| MLA4  | MARIKINA    |
| MLA5  | PARANAQUE   |
| MLA6  | PARANAQUE   |
| MLA7  | QUEZON CITY |
| MLA8  | QUEZON CITY |
| MLA9  | ZAPOTE      |
| MLA10 | ZAPOTE      |
+-------+-------------+
10 rows in set (0.000 sec)

MariaDB [mydatabase5_efa_caro]>
```

Perform the following tasks using JOIN commands.

- List of Product ID’s, Product Names and Quantities That Kaloys dinner need

```
MariaDB [mydatabase5_efa_caro]> SELECT I.PID, I.PRODUCT, K.QTY
    -> FROM ITEM_LIST I
    -> INNER JOIN KALOYS_DINER K USING(PID);
+-----+---------------+------+
| PID | PRODUCT       | QTY  |
+-----+---------------+------+
| P01 | BEEF STEAK    |    1 |
| P04 | CHICKEN WINGS |    5 |
| P06 | CHICKEN LIVER |    4 |
| P10 | WHOLE PORK    |    4 |
| P12 | WHOLE COW     |    1 |
| P14 | ONE DAY OLD   |    6 |
+-----+---------------+------+
6 rows in set (0.003 sec)

MariaDB [mydatabase5_efa_caro]>
```

- Combined List of products that Both Customers need, including quantites

```
MariaDB [mydatabase5_efa_caro]> SELECT I.PID, I.PRODUCT, SUM(K.QTY+N.QTY)TOTALQTY
    -> FROM ITEM_LIST I
    -> INNER JOIN KALOYS_DINER K USING (PID)
    -> INNER JOIN NANAYS_EATERY N USING(PID);
+-----+---------------+----------+
| PID | PRODUCT       | TOTALQTY |
+-----+---------------+----------+
| P04 | CHICKEN WINGS |       10 |
+-----+---------------+----------+
1 row in set (0.003 sec)

MariaDB [mydatabase5_efa_caro]>
```

- List of suppliers that Kaloy's need and their location

```
MariaDB [mydatabase5_efa_caro]> SELECT S.SUPID, S.LOCATION
    -> FROM ITEM_LIST I
    -> INNER JOIN KALOYS_DINER USING(PID)
    -> INNER JOIN SUPPLIERLIST S USING(SUPID);
+-------+-------------+
| SUPID | LOCATION    |
+-------+-------------+
| MLA2  | NAVOTAS     |
| MLA2  | NAVOTAS     |
| MLA4  | MARIKINA    |
| MLA4  | MARIKINA    |
| MLA5  | PARANAQUE   |
| MLA7  | QUEZON CITY |
+-------+-------------+
6 rows in set (0.001 sec)

MariaDB [mydatabase5_efa_caro]>
```

- List of Suppliers that Nanay's need to contact and their location

```
MariaDB [mydatabase5_efa_caro]> SELECT S.SUPID, S.LOCATION
    -> FROM ITEM_LIST I
    -> INNER JOIN NANAYS_EATERY USING(PID)
    -> INNER JOIN SUPPLIERLIST S USING(SUPID);
+-------+-----------+
| SUPID | LOCATION  |
+-------+-----------+
| MLA4  | MARIKINA  |
| MLA5  | PARANAQUE |
| MLA6  | PARANAQUE |
+-------+-----------+
3 rows in set (0.000 sec)

MariaDB [mydatabase5_efa_caro]>
```

- Determine which Products are not being used by Nanays.

```
MariaDB [mydatabase5_efa_caro]> SELECT I.PID, I.PRODUCT
    -> FROM ITEM_LIST I
    -> LEFT JOIN NANAYS_EATERY N USING (PID)
    -> WHERE N.PID IS NULL;
+-----+---------------+
| PID | PRODUCT       |
+-----+---------------+
| P01 | BEEF STEAK    |
| P02 | BEEF CUTLET   |
| P03 | PORK STEAK    |
| P05 | GROUND PORK   |
| P06 | CHICKEN LIVER |
| P07 | BEEF LIVER    |
| P08 | CHICKEN FEET  |
| P10 | WHOLE PORK    |
| P11 | WHOLE CHICKEN |
| P12 | WHOLE COW     |
| P14 | ONE DAY OLD   |
| P15 | BEEF FRANKS   |
+-----+---------------+
12 rows in set (0.001 sec)

MariaDB [mydatabase5_efa_caro]>
```

- Determine which Products are not being used by Kaloy

```
MariaDB [mydatabase5_efa_caro]> SELECT I.PID, I.PRODUCT
    -> FROM ITEM_LIST I
    -> LEFT JOIN KALOYS_DINER K USING (PID)
    -> WHERE K.PID IS NULL;
+-----+---------------+
| PID | PRODUCT       |
+-----+---------------+
| P02 | BEEF CUTLET   |
| P03 | PORK STEAK    |
| P05 | GROUND PORK   |
| P07 | BEEF LIVER    |
| P08 | CHICKEN FEET  |
| P09 | CHICKEN SKIN  |
| P11 | WHOLE CHICKEN |
| P13 | YOUNG PORK    |
| P15 | BEEF FRANKS   |
+-----+---------------+
9 rows in set (0.000 sec)

MariaDB [mydatabase5_efa_caro]>
```

Conclusion:

In this activity we are able to use complex commands to make queries and subqueries. Through this activity we learned the different usage of join clause, where there are 4 main command being use inner, left, right and full join. All of which can be used depending on the requirements of the program being made. Join clause will be useful in sorting data and joining separate tables together.

Proof of collaboration: