WEEK-1: HANDS-ON SQL

NAME : INIYA J

EMP_ID : 2211477

COHORT CODE: CDB22DW022

SOFTWARE USED: Postgresql and Pg Admin4

EXERCISE 1:

Objective:

To create a table.

Query:

```
CREATE TABLE Trainer_info(
   Trainer_Id VARCHAR(20) PRIMARY KEY NOT NULL,
   Salution VARCHAR(7) NOT NULL,
   Trainer_Name VARCHAR(30) NOT NULL,
   Trainer_Location VARCHAR(30) NOT NULL,
   Trainer_Track VARCHAR(15) NOT NULL,
   Trainer_Qualification VARCHAR(100) NOT NULL,
   Trainer_Experience INTEGER NOT NULL,
   Trainer_Email VARCHAR(100) NOT NULL,
   Trainer_Password VARCHAR(20) NOT NULL
)

CREATE TABLE Batch_info(
   Batch_Id VARCHAR(20) PRIMARY KEY NOT NULL,
   Batch_Owner VARCHAR(30) NOT NULL,
   Batch_Name VARCHAR(30) NOT NULL
)

CREATE TABLE  Module_info (
   Module_Id VARCHAR(20) PRIMARY KEY NOT NULL,
   Module_Name VARCHAR(40) NOT NULL,
   Module_Duration INTEGER NOT NULL
)

CREATE TABLE Associate_info (
   Associate_Id VARCHAR(20) PRIMARY KEY NOT NULL,
   Salutation VARCHAR(7) NOT NULL,
   Associate_name VARCHAR(30) NOT NULL,
   Associate_Location VARCHAR(30) NOT NULL,
   Associate_Track VARCHAR(15) NOT NULL,
   Associate_Qualification VARCHAR(200) NOT NULL,
   Associate_Email VARCHAR(100) NOT NULL,
   Asssociate_Password VARCHAR(20) NOT NULL
)

CREATE TABLE Questions(
   question_id VARCHAR(20) PRIMARY KEY NOT NULL,
   module_id VARCHAR(20) REFERENCES module_info(module_id),
   question_text VARCHAR(900) NOT NULL
)

CREATE TABLE Associate_Status(
   Associate_Id VARCHAR(20) REFERENCES Associate_Info(Associate_id),
   Module_Id VARCHAR(20) REFERENCES module_info(module_id),
   Batch_Id VARCHAR(20) REFERENCES batch_info(batch_id),
   Trainer_id VARCHAR(20) REFERENCES trainer_info(trainer_id),
   Start_Date VARCHAR(20) NOT NULL,
   End_Date VARCHAR(20) NOT NULL
)

CREATE TABLE Trainer_Feedback(
   Trainer_Id VARCHAR(20) REFERENCES trainer_info(trainer_id),
   Question_Id VARCHAR(20) REFERENCES Questions(question_id),
   Batch_Id VARCHAR(20) REFERENCES Batch_Info(batch_id),
   Module_Id VARCHAR(20) REFERENCES Module_Info(module_id),
   Trainer_Rating INTEGER NOT NULL
)

CREATE TABLE Associate_Feedback(
   Associate_Id VARCHAR(20) REFERENCES Associate_Info(associate_id),
   Question_Id VARCHAR(20) REFERENCES Questions(question_id),
   Module_Id VARCHAR(20) REFERENCES Module_Info(module_id),
   Associate_Rating INTEGER NOT NULL
)

CREATE TABLE Login_Details(
   User_Id VARCHAR(20) PRIMARY KEY NOT NULL,
   User_Password VARCHAR(20) NOT NULL
)
```

EXERCISE 2

Objective:

To insert the given details into table.

Query:

```
INSERT INTO
trainer_info(trainer_id,salution,trainer_name,trainer_location,trainer_track,trainer_qualification,trainer
_experience,trainer_email,trainer_password)

VALUES

('F001','Mr.','PANKAJ GHOSH','Pune', 'Java','Bachelor of
Technology',12,'Pankaj.Ghosh@alliance.com','fac1@123'),

('F002','Mr.','SANJAY RADHAKRISHNAN','Bangalore','DotNet','Bachelor of
Technology',12,'Sanjay.Radhakrishnan@alliance.com','fac2@123'),

('F003','Mr.','VIJAY MATHUR','Chennai','Mainframe','Bachelor of
Technology',10,'Vijay.Mathur@alliance.com','fac3@123'),

('F004','Mrs.','NANDINI NAIR','Kolkata','Java','Master of Computer
Application',9,'Nandini.Nair@alliance.com','fac4@123'),

('F005','Miss.','ANITHA PAREKH','Hyderabad','Testing','Master of Computer
Application',6,'Anitha.Parekh@alliance.com','fac5@123'),

('F006','Mr.','MANOJ AGRAWAL','Mumbai','Mainframe','Bachelor of
Technology',9,'Manoj.Agrawal@alliance.com','fac6@123'),

('F007','Ms.','MEENA KULKARANI','Coimbatore','Testing','Bachelor of
Technology',5,'Meena.Kulkarni@alliance.com','fac7@123'),

('F009','Mr.','SAGAR MENON','Mumbai','Java','Master of Science In Information
Technology',12,'Sagar.Menon@alliance.com','fac8@123')


INSERT INTO batch_info(batch_id,batch_owner,batch_name)

VALUES

('B001','MRS.SWATI ROY','MSP'),

('B002','MRS.ARURNA K', 'HEALTHCARE'),

('B003','MR.RAJESH KRISHNAN', 'LIFE SCIENCE'),
('B004','MR.SACHIN SHETTY','BFS'),

('B005','MR.RAMESH PATEL','COMMUNICATION'),

('B006','MRS.SUSAN CHERIAN','RETAIL & HOSPITALITY'),

('B007','MRS.SAMPADA JAIN','MSP'),

('B008','MRS.KAVITA REGRE','BPO'),

('B009','MR.RAVI SEJPAL','MSP')


INSERT INTO module_info(module_id,module_name,module_duration)

VALUES

('O10SQL','Oracle 10g SQL',16),

('O10PLSQL','Oracle 10g PL/SQL',16),

('J2SE','Core Java SE 1.6',288),

('J2EE','Advanced Java EE 1.6',80),

('JAVAFX','JavaFX 2.1',80),

('DOTNT4','.Net Framework 4.0',50),

('SQL2008', 'MS SQL Server 2008',158),

('MSBI08','MS BI Studio 2008',158),

('SHRPNT', 'MS Share Point',80),

('ANDRD4','Android 4.0',200),

('EM001','Instruction',0),

('EM002','Course Material',0),

('EM003','Learning Effectiveness',0),

('EM004','Environment',0),

('EM005','Job Impact',0),

('TM001','Attendees',0),

('TM002','Course Material',0),

('TM003','Environment',0)
```

```
INSERT INTO associate_info(associate_id, salutation, associate_name,associate_location, associate_track, associate_qualification, associate_email,asssociate_password)
VALUES
('A001','Miss.','GAYATHRI NARAYANAN','Gurgaon','Java','Bachelor of Technology','Gayathri.Narayanan@hp.com','tne1@123'),
('A002','Mrs.','RRADGIKA MOHAN','Kerala','Java','Bachelor of Engineering In Information Technology','Radhika.Mohan@cognizant.com','tne2@123'),
('A003','Mr.','KISHORE SRINNIVAS','Chennai','Java','Bachelor of Engineering in Computers','Kishore.Srinivas@ibm.com','tne3@123'),
('A004','Mr.','ANAND RANGANATHAN','Mumbai','DotNet','Master of Computer Applications','Anand.Ranganathan@finolex.com','tne4@123'),
('A005','Miss.','LEELA MENON','Kerala','Mainframe','Bachelor of Engineering In Information Technology','Leela.Menon@microsoft.com','tne5@123'),
('A006','Mrs.','ARTI KRISHNAN','Pune','Testing','Master of COmputer Application','Arti.Krishnan@cognizant.com','tne6@123'),
('A007','Mr.','PRABHAKAR SHANMUGHAM','Mumbai','Java','Bachelor of Technology','Prabhakar.Shanmugham@honda.com','tne7@123')

INSERT INTO questions(question_id,module_id,question_text)
VALUES
('Q001','EM001','Instructor knowledgeable and able to handle all your queries.'),
('Q002','EM001','All the topics in a particular course handled by thr trainer without any gaps or slippages.'),
('Q003','EM002','The course materials presentation, handson, etc. refered during the training are relevant and useful.'),
('Q004','EM002','The handson session adequate enough to grasp the understanding of  the topic.'),
('Q005','EM002','The referencwe materials suggested for each module are adequate.'),
('Q006','EM003','Knowledge and skills presentedd in this training are applicable at your work.'),
('Q007','EM003','This training increases my proficiency level.'),
('Q008','EM004','The physical  environment e.g. classroom space, air-conditioniing was conductive to learning.'),
('Q009','EM004','The software/hardware environment provided was sufficient for the purpose of the training.'),
('Q010','EM005','This training will improve your job performance.'),
('Q011','EM005','This training align with the business priorities and goals.'),
('Q012','TM001','Participants were receptive and had attitude towards learning.'),
('Q013','TM001','All participant gained the knowledgge and the practical skills after this training.'),
('Q014','TM002','The course materials presentation, handson, etc. available for the session covers the entire objectives of the course.'),
('Q015','TM001','Participants were receptive and had attitude towards learning.'),
('Q016','TM002','Case study and practical demos helpful in understanding of the topic.'),
('Q017','TM003','The phyical environment eg.classroom space, air-conditioning was conductive to learning.'),
('Q018','TM003','The software/hardware environment provided was adequate for the purpose of the training.')

INSERT INTO associate_status(associate_id,module_id,batch_id,trainer_id,start_date,end_date)
VALUES
('A001','O10SQL','B001','F001','2000-12-15','2000-12-25'),
('A002','O10SQL','B001','F001','2000-12-15','2000-12-25'),
('A003','O10SQL','B001','F001','2000-12-15','2000-12-25'),
('A001','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),
('A002','O10PLSQL','B002','F002','2001-2-1','2001-2-12'),
('A001','J2SE','B003','F003','2002-8-20','2002-10-25'),
('A002','J2SE','B003','F003','2002-8-20','2002-10-25'),
('A001','J2EE','B004','F004','2005-12-1','2005-12-25'),
('A002','J2EE','B004','F004','2005-12-1','2005-12-25'),
('A003','J2EE','B004','F004','2005-12-1','2005-12-25'),
('A004','J2EE','B004','F004','2005-12-1','2005-12-25'),
('A005','JAVAFX','B005','F006','2005-12-4','2005-12-20'),
('A006','JAVAFX','B005','F006','2005-12-4','2005-12-20'),
('A006','SQL2008','B006','F007','2007-6-21','2007-6-28'),
('A007','SQL2008','B006','F007','2007-6-21','2007-6-28'),
('A002','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A003','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A004','MSBI08','B007','F006','2009-6-26','2009-6-29'),

('A002','ANDRD4','B008','F005','2010-6-5','20010-6-28'),

('A003','ANDRD4','B008','F005','2010-6-5','20010-6-28'),

('A003','ANDRD4','B009','F005','2011-8-1','20011-8-20'),

('A006','ANDRD4','B009','F005','2011-8-1','20011-8-20')
```

EXERCISE:3

Objective:

To change data in table

Query:

```
UPDATE trainer_info

SET trainer_password = 'nn4@123'

WHERE trainer_id = 'F004'

RETURNING trainer_password;
```

EXERCISE:4

Objective:

To delete data from table.

Query:

```
DELETE FROM associate_status

WHERE associate_id = 'A003' AND module_id = 'J2EE' AND batch_id = 'B004' AND trainer_id = 'F004'

AND start_date = '2005-12-1' AND end_date = '2005-12-25'

RETURNING associate_id, module_id, batch_id, trainer_id, start_date, end_date;
```

EXERCISE:5

Objective:

To perform top N records

Query:

```
SELECT * FROM trainer_info

ORDER BY trainer_experience

LIMIT 5;
```

EXERCISE:6

Objective:

Understanding TCL

Query:

```
BEGIN TRANSACTION;

INSERT INTO login_details(user_id, user_password)

VALUES

('U001','Admin@123'),

('U002','Admin@123');


SELECT * FROM login_details;


ROLLBACK TRANSACTION;


SELECT * FROM login_details;
```

EXERCISE:7

Objective:

To understand DCL.

Query:

```
CREATE USER dummy_user PASSWORD 'dummy';

GRANT SELECT ON login_details TO dummy_user;

GRANT INSERT ON login_details TO dummy_user;


SELECT * FROM login_details;

BEGIN TRANSACTION;

INSERT INTO login_details(user_id, user_password)

VALUES

('U001','Admin@123'),

('U002','Admin@123');

SELECT * FROM login_details;

ROLLBACK TRANSACTION;

SELECT * FROM login_details;


REVOKE SELECT,INSERT ON login_details FROM dummy_user;
```

EXERCISE:8

Objective:

To understand how to drop table.

Query:

```
DROP TABLE login_details;
```

EXERCISE:9

Objective:

To understand the Constraints.

Query:

```
CREATE TABLE suppliers(
    supplier_id INTEGER NOT NULL,
    supplier_name VARCHAR(50) NOT NULL,
    address VARCHAR(50),
    city VARCHAR(50),
    state VARCHAR(25),
    zip_code VARCHAR(10)
)
```

EXERCISE:10

Objective:

To understand distinct and Union.

Query:

```
CREATE TABLE course_fee_history(
    course_code VARCHAR(20),
    base_fees INTEGER,
    special_fees INTEGER,
    created_by VARCHAR(20),
    updated_by VARCHAR(20)
)

INSERT INTO course_fee_history(course_code,base_fees,special_fees,created_by,updated_by)
VALUES
('1',120,123,'Ram','Ramesh'),
('2',150,110,'Bala','Ram'),
('3',160,170,'Bala','Vinu'),
('4',170,235,'Ram','Ram'),
('6',190,100,'Vinod','Vinod');

CREATE TABLE course_fees(
    course_code VARCHAR(10),
    base_fees INTEGER,
    special_fees INTEGER,
    discount INTEGER
);

INSERT INTO course_fees(course_code,base_fees,special_fees,discount)
VALUES
('1',180, 100, 10),
('2',150,110,10),
('3',160,170,5),
('4',150,100,10),
('6',190,100,40);

SELECT DISTINCT course_code FROM course_fee_history
UNION
SELECT DISTINCT course_code FROM course_fees;
```

EXERCISE:11

Objective:

To understand uniqueness of columns.

Query:

```sql
SELECT COUNT(DISTINCT course_code), COUNT(course_code) FROM course_fees;
-- since the unique number of course code is equal to its total, it is unique.
SELECT COUNT(DISTINCT base_fees), COUNT(base_fees) FROM course_fees;
-- since the unique number of base fees is not equal to its total, it is not unique.
SELECT COUNT(DISTINCT special_fees), COUNT(special_fees) FROM course_fees;
-- since the unique number of special fees is not equal to its total, it is not unique.

SELECT COUNT(DISTINCT course_code), COUNT(course_code) FROM course_fee_history;
-- since the unique number of course code is equal to its total, it is unique.
SELECT COUNT(DISTINCT base_fees), COUNT(base_fees) FROM course_fee_history;
-- since the unique number of base fees is equal to its total, it is unique.
SELECT COUNT(DISTINCT special_fees), COUNT(special_fees) FROM course_fee_history;
-- since the unique number of special fees is equal to its total, it is unique.
```

EXERCISE:12

Objective:

To understand minimum and maximum functions.

Query:

```sql
INSERT INTO course_fees(course_code, special_fees,discount)
```

```
VALUES
('7',120,15),
('8',140,20);

INSERT INTO course_fees(course_code,base_fees,special_fees,discount)
VALUES
('7',140,120,15),
('8',160,140,20);

SELECT MIN(base_fees) FROM course_fees;
SELECT MAX(base_fees) FROM course_fees;
```

EXERCISE:13

Objective:

To understand the average function.

Query:

```
ALTER TABLE course_fees
ADD COLUMN infra_fees DECIMAL(5,3);
UPDATE course_fees
SET infra_fees='45.751'
WHERE course_code = '1';
UPDATE course_fees
SET infra_fees='43.453'
WHERE course_code = '2';
UPDATE course_fees
SET infra_fees='34.543'
WHERE course_code = '3';
```

```
UPDATE course_fees
SET infra_fees='41.345'
WHERE course_code = '4';
UPDATE course_fees
SET infra_fees='52.721'
WHERE course_code = '5';
UPDATE course_fees
SET infra_fees='39.211'
WHERE course_code = '6';
UPDATE course_fees
SET infra_fees='53.700'
WHERE course_code = '7';
UPDATE course_fees
SET infra_fees='35.553'
WHERE course_code = '8';

SELECT AVG(infra_fees) FROM course_fees;
```

EXERCISE:14

Objective:

To obtain number of days between given date and current date.

Query:

```
CREATE TABLE course_info(
   course_code VARCHAR(10),
   course_name VARCHAR(20),
   course_description VARCHAR(25),
   course_start_date DATE,
   course_duration INT,
   no_of_participants INT,
   course_type char(3)
);
```

```
SELECT course_name, AGE(course_start_date) FROM course_info;
```

EXERCISE:15

Objective:

To understand the concatenation function.

Query:

```
SELECT course_name || ' ' || course_code FROM course_info;
```

EXERCISE:16

Objective:

To understand the average function.

Query:

```
SELECT COALESCE(AVG(base_fees),0) FROM course_fees;
```

EXERCISE:17

Objective:

To type the appropriate message as mentioned.

Query:

```
ALTER TABLE course_info

ADD COLUMN messages VARCHAR(20);


INSERT INTO course_info(course_type,messages)

VALUES

('CLR','Class Room'),

('EL','ELearning'),

('OF','Offline Reading');


SELECT course_type,messages FROM course_info

WHERE course_type IN ('CLR','EL','OF');
```

EXERCISE:18

Objective:

To understand Group By Clause.

Query:

```
SELECT course_start_date, SUM(no_of_participants) AS total_no_of_participants FROM course_info

GROUP BY  course_start_date;
```

EXERCISE:19

Objective:

To understand the use of Where and Group by Clauses.

Query:

```
SELECT course_start_date, SUM(no_of_participants) AS total_no_of_participants FROM course_info

WHERE course_type = 'CLR'

GROUP BY  course_start_date;
```

EXERCISE:20

Objective:

To understand Having Clause.

Query:

```
SELECT course_start_date, SUM(no_of_participants) AS total_no_of_participants FROM course_info

WHERE course_type = 'CLR'

GROUP BY  course_start_date

HAVING SUM(no_of_participants)>10;
```

EXERCISE:21

Objective:

To understand Order By function.

Query:

```
SELECT course_name FROM course_info

ORDER BY course_duration ASC;
```

EXERCISE:22

Objective:

To understand Inner Join.

Query:

```
CREATE TABLE student_info(
   student_id VARCHAR(10),
   first_name VARCHAR(20),
   last_name VARCHAR(25),
   course_code VARCHAR(10),
   address VARCHAR(150)
);

SELECT first_name,last_name,student_info.course_code FROM student_info
INNER JOIN course_info
ON student_info.course_code = course_info.course_code
WHERE student_info.course_code ='167';
```

EXERCISE:23

Objective:

To understand Inner Join.

Query:

```
SELECT course_fees.discount, course_info.course_description FROM course_fees
INNER JOIN course_info ON course_info.course_code = course_info.course_code;
```

EXERCISE:24

Objective:

To understand Left Join and Right Join.

Query:

```
INSERT INTO student_info
VALUES
('007','John','Snow','D004','Kings Landing');

SELECT first_name,course_info.course_code FROM student_info
LEFT JOIN course_info
ON student_info.course_code = course_info.course_code;

SELECT first_name,course_info.course_code FROM course_info
RIGHT JOIN student_info
ON course_info.course_code = student_info.course_code;
```

EXERCISE:25

Objective:

To understand Subqueries.

Query:

```
INSERT INTO Course_Info(Course_code,Course_Name,No_Of_Participants,Course_Type)
VALUES
('A13','ML',5,'AB'),
('A14','NLP',3,'BC');
INSERT INTO Course_Fees(Course_Code,Base_Fees,Special_Fees,Discount,Infra_Fees)
VALUES
```

```
('A12',150,140,18,43.643),
('B11',135,155,14,47.980);


INSERT INTO Student_Info(student_id,first_name,last_name,course_code)
VALUES
('C004','Anita','Suresh','A13'),
('C005','Sundar','Pichai','A14');
SELECT Student_Info.Student_Id From Student_Info
INNER JOIN Course_Fees
ON Student_Info.Course_code=Course_Fees.Course_Code
WHERE ((Base_Fees + Special_Fees + Infra_Fees)*(1-Discount/100))<1500
```

EXERCISE:26

Objective:

To understand Subqueries.

Query:

```
SELECT Student_Info.Student_Id,Student_Info.First_Name,Student_Info.Last_Name From
Student_Info
INNER JOIN Course_Fees
ON Student_Info.Course_code=Course_Fees.Course_Code
WHERE ((Base_Fees + Special_Fees + Infra_Fees)*(1-Discount/100))<1500;
```