DATE : 17/10/2022

EXPERIMENT - 1

FAMILIARIZATION OF DATABASE CONCEPTS

AIM

To familianise basic concepts of database management system.

THEORY

Database

The database is a collection of inter-related data which is used to retoive, insert and delete the data efficiently. It is also used to organise the data in the form of a table, schema, views. etc

Database management system

DBMS is a software which is used to manage the database for example MYSQL, oracle etc are commonly used database. Database management system provides an interface to perform various operations like database creation, storing data in it, updating data, creating table in the database and a lot more

DDL (Data Definition Language) commands

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. It simply deals with description of the database schema and is used to create and modify the structure of database objects in database DDL is used of SQL commands

CREATE : Used to create database or its object

DROP : Used to data object from database

ALTER : Used to alter structure of database

TRUNCATE : Used to remove all record from a table including all. spaces allocated for record are removed.

COMMENT : used to add comment to data remo: dictionary
RENAME : used to rename an object existing in database.

DML (Data Manipulation Language) commands

The SQL commands that deals that with the manipulation of data Present in the database belong to DML or Data manipulation language and their includes most of SQL statements. It is congruent of SQL statement that control occur to data and to he db.

INSERT : used to insert data into a table
UPDATE : used to update existing data within a table
DELETE : Used to delete records from a database table
LOCK : Table control concurrency

RESULT

Basic concepts of Database Management system are familiarized.

Date: 17/10/2022

EXPERIMENT - 2

AIM

creation and insertion the tables and data according to the given data.

THEORY

Table Branch

- CREATE TABLE Branch (branchNO VARCHAR(30) Primary key, Street VARCHAR(30), City VARCHAR(30), Postcode VARCHAR(30));
- INSERT INTO Branch ~~VA~~
  VALUES('B005', '22 Deer Rd', 'London', 'SW14EH');
- INSERT INTO Branch
  VALUES('B007', '16 Argyll St', 'Abendeen', 'AB23SU');
- INSERT INTO Branch
  VALUES ('B003', '163 Main St', 'Glasgow', 'G119QX');
- INSERT INTO Branch
  VALUES ('B004', '32 Manse Rd', 'Bristol', 'BS991NZ');
- INSERT INTO Branch
  VALUES ('B002', '56 Cloven Dr', 'London', 'NW106FU');

Table staff

- CREATE TABLE Staff (staffNO VARCHAR (30). Primary key, fName VARCHAR(30), lName VARCHAR(30), Position VARCHAR(30), sex VARCHAR(1), DOB Date, Salary float, branchNO VARCHAR (4), Foreign key (branchNO) Reference Branch (branchNO));

- INSERT INTO PropertyForRent
  VALUES ('PG16', '5 Novar Dr', 'Glasgow', 'G12 9AX', 'Flat', 4, 450, 'CO93', 'SG14', 'B003');

Table client

- CREATE TABLE Client (clientNo VARCHAR(30) Primary Key, fName VARCHAR(30)
  lName VARCHAR(30), telNo VARCHAR(30), Preftype VARCHAR(30), max Rent
  INT, email VARCHAR(30));

- INSERT INTU Client
  VALUES('CR76', 'John', 'Kay', '0207-774-5632', 'Flat', 425, 'John.Kay@gmail.com');
- INSERT INTO Client
  VALUES('CR56', 'Aline', 'Stewart', '0141-848-1825', 'Flat', 350, 'astewart@hotmail.com');
- INSERT INTO Client
  VALUES('CR74', 'mike', 'Ritchie', '01475-392178', 'House', 750, 'mritchie01@yahoo.co.uk);
- INSERT INTO Client
  VALUES('CR62', 'mary', 'Tregear', '01224-496720', 'Flat', 600, 'maryt@hotmail.co.uk);

Table privateowner

- CREATE TABLE PrivateOwner (ownerNo VARCHAR(30) Primary Key, fName
  VARCHAR(30), lName VARCHAR(30), address VARCHAR(30), telNo VARCHAR(30),
  eMail VARCHAR(30), Password VARCHAR(30));

- INSERT INTO PrivateOwner
  VALUES('CO46', 'Joe', 'Keogh', '2 Fergus Dr, Aberdeen AB2 7SX', '01224-861212',
  'jkeogh@lhh.com, '********');
- INSERT INTO PrivateOwner
  VALUES('CO87', 'Carol', 'Farrel', '6 Achray St, Glasgow G32 9DX', '0141-357-7419',
  'carol@gmail.com', '*********');

- INSERT INTO PrivateOwner
  VALUES ('CR ~~76~~ 40', ~~'Mike'~~ 'Tina', 'murphy', '63 well st, Glasgow G42', '0141-943-1728',
  'tinam@hotmail.com', '********');

```
• INSERT INTO Staff
  VALUES('SL21', 'John', 'White', 'Manager', 'M', '1-Oct-45', '30000', 'B005');
• INSERT INTO Staff
  VALUES('SG37', 'Ann', 'Beech', 'Assistant', 'F', '10-NOV-60', '12000', 'B003');
• INSERT INTO Staff
  VALUES('SG14', 'David', 'Ford', 'Supervisor', 'M', '24-Mar-58', '18000, 'B003');
• INSERT INTO Staff
  VALUES ('SA9', 'Mary', 'Howe', 'Assistant', 'F', '19-Feb-70', '9000', 'B007');
• INSERT INTO Staff
  VALUES('SG5', 'Susan', 'Brand', 'Manager', 'F', '3-Jun-40', '24000', 'B003');
• INSERT INTO Staff
  VALUES( 'SL41', 'Julie', 'Lee', 'Assistant', 'F', '13-Jun-65', '9000', 'B005');
```

Table PropertyForRent

```
• CREATE Table PropertyForRent (PropertyNO VARCHAR(30) Primary key, stret
  VARCHAR(30), city VARCHAR(30), Postcode VARCHAR(30), type VARCHAR(30),
  rooms INT, rent INT, ownerNO VARCHAR(30), staffNO VARCHAR(30),
  branchNO VARCHAR(30), foreign key (branchNO) references Branch(branch
  no), foreign key (staffNO) reference Staff (staffNO));

• INSERT INTO PropertyForRent
  VALUES ('PA14', '16 Holhead', 'Aberdeen', 'AB75SU', 'House', '6', 650, 'CO46',
          'SA9', 'B007');

• INSERT INTO PropertyForRent
  VALUES ('PL94', '6 Argyll St', 'London', 'NW2', 'Flat', 4, 400, 'CO87', 'SL41',
          'B005');

• INSERT INTO PropertyForRent
  VALUES ('PG4', '6 Lawrence St', 'Glasgow', 'G119QX', 'Flat', 3, 350, 'CO40', ' ',
          'B003');

• INSERT INTO PropertyForRent
  VALUES ('PG36', '2 manor Rd', 'Glasgow', 'G324QX', 'Flat', 3, 375, 'CO93', 'SG37',
          'B003');

• INSERT INTO PropertyForRent
  VALUES('PG21', '18 Dale Rd', 'Glasgow', 'G12', 'House', 5, 600, 'CO87', 'SG37', 'B003');
```

```
• INSERT INTO PrivateOwner
  VALUES ('Co93','Tony','Shaw','12 Park Pl, Glagow G40QR','0141-225-7025',
  'tony.shaw@ark.com','* * * * * * * * *');
```

Table viewing

```
• CREATE TABLE viewing (clientNO VARCHAR(30), PropertyNO
  VARCHAR(30), viewDate Date, coment VARCHAR (30) ~~);~~, foreign key
  (clientNO) reference client(clientNO));
• INSERT INTO Viewing
  VALUES('CR56', 'PA14', '24-may-13', 'too small');
• INSERT INTO Viewing
  VALUES('CR76', 'PG4', '06-Apr-13', 'too remote');
• INSERT INTO Viewing
  VALUES('CR56', 'PG4', '26-may-13 ," ');
• INSERT INTO Viewing
  VALUES('CR62', 'PA14', '14-may-13', 'no dining room');
• INSERT INTO Viewing
  VALUES('CR56', 'PG36', '28-Apr-13', '  ');
```

Table Registration

```
• CREATE TABLE Registration (clientNO VARCHAR(30), branchNO
  VARCHAR(30), staffNO VARCHAR(30), datejoined Date ~~);~~, Primary
  key(clientNO, branchNO]);
• INSERT INTO Registration
  VALUES('CR76','B005', 'SL41', 2-jan-13);
• INSERT INTO Registration
  VALUES('CR56','B003', 'SG37', '11-Apr-12');
• INSERT INTO Registration
  VALUES('CR74','B003','SG37','16-Nov-11');
```

```
- INSERT INTO Registration
  VALUES ('CR62', 'B007', 'SA9', '7-may-12');
```

RESULT

Creation and of tables and insertion of data successfully completed.

EXPERIMENT - 3

Date : 21/10/2022

QUERY USING SELECT COMMAND

AIM

To retrieve data from the ~~given~~ created table

THEORY

1. Retrive details of Branch situated in London

```
SELECT * FROM Branch
WHERE City = 'London';
```

2. Retrive details of Staff where name is David Ford

```
SELECT * FROM Staff
WHERE fname = 'David' AND lname = 'Ford';
```

3. Retrieve staff and branch details of staff who work as assistant

```
SELECT * FROM ~~BRANC~~ Branch, Staff
WHERE Position = 'Assistant' AND ~~BranchNo~~ Branch.branchNo = Staff.branchNo;
```

4. Retrieve distinct Salary from Staff

```
SELECT DISTINCT Salary FROM Staff
```

5. Find names of all staff having 'u' as the 2nd letter in their name

```
SELECT fname, lname FROM Staff
WHERE fname LIKE '_u%';
```

RESULT

~~SELECT~~
Retrieving of data successfully completed by using select command and output verified.

POCO
SHOT ON POCO M2

Date 21/10/2022

EXPERIMENT - 4

QUERIES USING UPDATE COMMAND

AIM

To update rows in table using UPDATE command

THEORY

1. Change the Salary of Manager to 20,000

```
UPDATE Staff
SET Salary = 20,000
WHERE position ='Manager';
```

2. change the DOB of David to 24-Mar-59

~~SE~~

```
UPDATE Staff
  SET DOB ='24-Mar-59'
  WHERE fName ='David';
```

3. Change the position of all staff where name starts with J as Clerk

```
UPDATE Staff
SET position ='Clerk'
WHERE fName LIKE 'J%';
```

4. Increment 10% rent for those who are in a flat having rooms greater than or equal to 4

```
UPDATE Staff
SET rent = rent * 1.1
WHERE rooms >= 4 AND types ='Flat';
```

RESULT

Update commands are performed successfully to update rows. and output verified

POCO
SHOT ON POCO M2

EXPERIMENT -5

Date: 21/10/2022

QUERIES USING AGGREGATE FUNCTION

AIM

select and update rows by using aggregate function

THEORY

1. Find the Average of Staff's salary

```
SELECT AVG(salary)
FROM Staff;
```

2. Retrieve the details of staff who is having salary >= Avg Salary

```
SELECT * FROM Staff
WHERE Salary >= (SELECT AVG(Salary) FROM Staff);
```

3. find the maximum and minimum Salary

```
SELECT MAX(salary), MIN(Salary)
From Staff;
```

4. Increase the Salary of staff who is having minimum salary by 1000

```
UPDATE Staff
SET Salary = Salary + 10000
WHERE Salary = (SELECT MIN(salary) FROM Staff);
```

5. Find total number of types in propertyForRent

```
SELECT COUNT (DISTINCT Types) FROM PropertyForRent;
```

6. Find total number of flat

```
SELECT COUNT(Types) FROM PropertyForRent
WHERE types = 'Flat';
```

POCO
SHOT ON POCO M2