

*** EXPERIMENT NO: 01 ***

Author : Atharva Paliwal
Roll No : 40 [5B]
Date : 08-August-2020

AIM: To establish a multi-relation database and execute SQL queries involving insertions, deletions and updating on it.

PROBLEM STATEMENT: Establish the **TinyStores** database and execute different SQL queries against it. The logical database schemata ,the organization of relations and their contents are as below.

EMP (EMP_CODE, EMP_LNAME, EMP_FNAME, EMP_DOB, STORE_CODE)
STORE (STORE_CODE, STORE_NAME, YTD_SALES, REGION_CODE, EMP_CODE)
REGION (REGION_CODE, REGION_DESC)

SQL> connect CS540/atharva
Connected.

QUERY 01: Write SQL code that will create the TinyStores database.

```
SQL> CREATE TABLE EMP (  
2     EMP_CODE NUMBER(2) NOT NULL,  
3     EMP_FNAME VARCHAR2(15) NOT NULL,  
4     EMP_LNAME VARCHAR2(15) NOT NULL,  
5     EMP_DOB DATE DEFAULT SYSDATE-(365*22) NOT NULL,  
6     STORE_CODE NUMBER(2) NOT NULL,  
7     SALARY NUMBER(5) NOT NULL CHECK (SALARY>=10000),  
8     CONSTRAINT EMP_PK_EMP_CODE PRIMARY KEY (EMP_CODE)  
9 );
```

Table created.

```
SQL> CREATE TABLE STORE (  
2     STORE_CODE NUMBER(2) NOT NULL,  
3     STORE_NAME VARCHAR(25) NOT NULL,  
4     YTD_SALES NUMBER(9,2) DEFAULT 0 NOT NULL,  
5     REGION_CODE NUMBER(1) NOT NULL,  
6     EMP_CODE NUMBER(2) NOT NULL,  
7     CONSTRAINT STORE_PK_STORE_CODE PRIMARY KEY (STORE_CODE),  
8     CONSTRAINT STORE_FK_EMP_EMP_CODE FOREIGN KEY(EMP_CODE) REFERENCES EMP(EMP_CODE)  
9 );
```

Table created.

```
SQL> CREATE TABLE REGION (
  2  REGION_CODE NUMBER(1) NOT NULL,
  3  REGION_DESC VARCHAR2(10) NOT NULL CHECK(REGION_DESC
    IN('EAST','WEST','NORTH','SOUTH')),
  4  CONSTRAINT REGION_PK_REGION_CODE PRIMARY KEY (REGION_CODE)
  5 );
```

Table created.

QUERY 04: Write SQL code to print the date and time of the system.
(You must ensure the system clock is correct)

```
SQL> SELECT SYSDATE,SYSTIMESTAMP FROM DUAL;
```

SYSDATE	SYSTIMESTAMP
07-AUG-20	07-AUG-20 06.43.27.650000 PM +05:30

QUERY 05: Assuming that the database is fully populated, write the SQL code that will list all employees who do not earn more than 35000.

```
SQL> SELECT * FROM EMP
  2  WHERE SALARY <=35000;
```

EMP_CODE	EMP_FNAME	EMP_LNAME	EMP_DOB	STORE_CODE	SALARY
24	VALLABH	ROY	11-DEC-74	41	32000

QUERY 06: Write SQL code to list the first names and last names of the employees who were born before 01-JAN-1972 and who are posted in the western region.

```
SQL> SELECT EMP_FNAME,EMP_LNAME FROM EMP,STORE
  2  WHERE EMP_DOB<='01-JAN-1972' AND EMP.STORE_CODE=STORE.STORE_CODE AND
STORE.REGION_CODE=2;
```

EMP_FNAME	EMP_LNAME
KASHISH	SHUKLA

QUERY 07: Write SQL code that will for each store print the name of manager alongwith the store details.

```
SQL> SELECT EMP_FNAME,EMP_LNAME,STORE_NAME,YTD_SALES,REGION_CODE,EMP.EMP_CODE FROM
EMP,STORE
2 WHERE EMP.EMP_CODE=STORE.EMP_CODE;
```

EMP_FNAME	EMP_LNAME	STORE_NAME	YTD_SALES	REGION_CODE	EMP_CODE
-----	-----	-----	-----	-----	-----
KASHISH	SHUKLA	SUCCESS JUNCTION	1000555.76	2	11
ATHARVA	PALIWAL	CURIOSITY CIRCLE	568000	4	12
GAZAL	SINGH	OPPORTUNITY SQUARE	986785.4	1	13
RISHIKESH	KALE	ATTRIBUTE ALLEY	944568.66	3	18

QUERY 08: Write SQL code to print store code, store name, region name for each store.

```
SQL> SELECT STORE_CODE,STORE_NAME,REGION_DESC FROM STORE,REGION
2 WHERE STORE.REGION_CODE=REGION.REGION_CODE;
```

STORE_CODE	STORE_NAME	REGION_DES
-----	-----	-----
11	OPPORTUNITY SQUARE	EAST
21	SUCCESS JUNCTION	WEST
31	ATTRIBUTE ALLEY	NORTH
41	CURIOSITY CIRCLE	SOUTH

```
SQL> COMMIT;
```

Commit complete.

*****VIVA-VOCE*****

1. What is SQL?

Ans: Structured Query Language or SQL is a standard Database language which is used to create, maintain and retrieve the data from relational databases like MySQL, Oracle, SQL Server, PostGres, etc.

2. Enlist functions of DBA.

Ans:

- Schema definition
- Storage structure and Access Method definition
- Software installation and maintenance
- Database backup and recovery
- Security
- Authentication
- Performance monitoring
- Database Tuning
- Troubleshooting

3. What are the advantages of a RDBMS over a DBMS?

Ans:

- Storage: RDBMS stores data in the form of tables unlike DBMS as file
- Avoid Data Redundancy: RDBMS utilizes keys and indexes in the table to avoid redundancies
- Normalization: RDBMS supports Normalization where as DBMS does not
- ACID: RDBMS are consistent and well structured. They obey ACID (Atomicity, Consistency, Isolation, Durability)
- Relationship: RDBMS maintains relationships among the tables
- Integrity constraints: RDBMS supports the integrity constraints at the schema level.

4. Differentiate between a relation and a table.

Ans: A Table is a collection of related data held in a tabular format within a database. In terms of the RDBMS, a table can be considered a convenient representation of a relation, but the two are not strictly equivalent. For instance, a SQL table can potentially contain duplicate rows, whereas a true relation cannot contain duplicate rows that we call as tuples. Similarly, representation as a table implies a particular ordering to the rows and columns, whereas a relation is explicitly unordered.

5. Differentiate between the 3GLs and the 4GLs.

Ans:

- 3GLs are procedural languages because the instructions are procedure-oriented which means code tells the computer what to do as well as how to do. On the other hand, 4GLs are non-procedural languages because the instructions only specify the computer what to do and does not tell how to do.

- In 3GLs, a large volume of assembly language and machine language instructions are generated as compared to 4GLs. 4GLs require very less statements due to its reduced complexity.
- Most of 4GLs are associated with data processing and databases whereas 3GLs with others.
- Examples:
 - o 3GLs: C, C++, C#, JAVA, PASCAL, BASIC, etc.
 - o 4GLs: SQL, RamiS, Visual FoxPro, etc.

***** INFERENCES *****

- Studied and learned about relational model
- Implemented the relational model with SQL on Oracle 11g
- Created a database with tables EMP, STORE and REGION
- Implemented various queries on different tables which are related to each other in database

***** E N D *****