***** EXPERIMENT NO: 01 *****

Aim : To study the relational model and demonstrate basic SQL commands in Oracle 11g.

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)

Author : Bhavesh Kewalramani

Roll No : 025 [5A]

Date : 18-July-2021

****

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

****

```
CREATE TABLE REGION(
  2   REGION_CODE NUMBER(1) NOT NULL,
  3   REGION_DESC VARCHAR2(10) NOT NULL,
  4   CONSTRAINT REGION_PK_REGION_CODE PRIMARY KEY (REGION_CODE),
  5   CONSTRAINT REGION_CK_REGION_DESC CHECK (REGION_DESC IN
('East','West','North','South'))
  6   );
```

Table created.

```
INSERT INTO REGION VALUES(1,'East');
```

1 row created.

```
INSERT INTO REGION VALUES(2,'West');
```

1 row created.

```
INSERT INTO REGION VALUES(3,'North');
```

1 row created.

```
INSERT INTO REGION VALUES(4,'South');
```

```
1 row created.

      SELECT *
            2  FROM REGION;

REGION_CODE  REGION_DES
-----------  ----------
          1  East
          2  West
          3  North
          4  South

4 rows selected.

      CREATE TABLE STORE(
              2    STORE_CODE NUMBER(2) NOT NULL,
              3    STORE_NAME VARCHAR2(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_REGION_REGION_CODE FOREIGN KEY
             (REGION_CODE) REFERENCES REGION(REGION_CODE);
              9    );

Table created.

      INSERT INTO STORE VALUES(21,'Success Junction',1000555.76,2,11);

1 row created.

      INSERT INTO STORE VALUES(22,'Database Corner',1420000.34,2,22);

1 row created.

      INSERT INTO STORE VALUES(11,'Opportunity Square',986785.40,1,13);

1 row created.

      INSERT INTO STORE VALUES(31,'Attribute Alley',944568.66,3,18);
```

1 row created.

```
INSERT INTO STORE VALUES(12,'Central Deluge',2930098.35,1,17);
```

1 row created.

```
INSERT INTO STORE VALUES(41,'Curiosity Circle',568000.00,4,12);
```

1 row created.

```
STORE_CODE STORE_NAME                YTD_SALES REGION_CODE   EMP_CODE
---------- ------------------------- ---------- ----------- ----------
        21 Success Junction          1000555.76           2         11
        22 Database Corner           1420000.34           2         22
        11 Opportunity Square          986785.4           1         13
        31 Attribute Alley            944568.66           3         18
        12 Central Deluge            2930098.35           1         17
        41 Curiosity Circle              568000           4         12
```

6 rows selected.

```
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 NOT NULL,
  6    STORE_CODE NUMBER(2) NOT NULL,
  7    SALARY NUMBER(5) NOT NULL,
  8    CONSTRAINT EMP_PK_EMP_CODE PRIMARY KEY (EMP_CODE),
  9    CONSTRAINT EMP_CK_SALARY CHECK (SALARY >=10000),
 10    CONSTRAINT EMP_FK_STORE_STORE_CODE FOREIGN KEY (STORE_CODE) REFERENCES
STORE(STORE_CODE)
 11    );
```

Table created.

```
INSERT INTO EMP VALUES(11,'Kashish','Shukla','25-May-1966',21,45000);
```

1 row created.

```
INSERT INTO EMP VALUES(12,'Shaswat','Puri','23-Nov-1961',11,25000);
```

1 row created.

```
INSERT INTO EMP VALUES(13,'Gazal','Singh','02-Feb-1963',11,48000);
```

1 row created.

```
INSERT INTO EMP VALUES(14,'Mohana','Seth','01-Jun-1971',22,27000);
```

1 row created.

```
INSERT INTO EMP VALUES(15,'Bhavesh','Kewalramani','11-May-2001',41,36000);
```

1 row created.

```
INSERT INTO EMP VALUES(16,'Bahar','Toshniwal','10-Sep-1968',21,30000);
```

1 row created.

```
INSERT INTO EMP VALUES(17,'Vedika','Mishra','06-Mar-1966',11,31000);
```

1 row created.

```
INSERT INTO EMP VALUES(18,'Simon','Gonsalvis','03-Sep-1961',12,25000);
```

1 row created.

```
INSERT INTO EMP VALUES(19,'Vikrant','Gokhale','31-Jul-1963',12,38000);
```

1 row created.

```
INSERT INTO EMP VALUES(20,'Harsh','Shrirame','13-Jun-2001',31,38000);
```

1 row created.

```
INSERT INTO EMP VALUES(21,'Pampa','Roy','11-Dec-1974',12,28000);
```

1 row created.

```
INSERT INTO EMP VALUES(22,'Chanchal','Bhati','08-Sep-1974',22,40000);
```

1 row created.

```
INSERT INTO EMP VALUES(23,'Sriniwas','Chintala','25-Aug-1964',31,26000);

1 row created.

INSERT INTO EMP VALUES(24,'Ranjit','Rungta','11-Dec-1974',41,32000);

1 row created.

INSERT INTO EMP VALUES(25,'Mehar','Mirpuri','09-Feb-1969',22,29000);

1 row created.

SELECT *
  2  FROM EMP;

EMP_CODE EMP_FNAME       EMP_LNAME       EMP_DOB   STORE_CODE     SALARY
---------- --------------- --------------- --------- ---------- ----------
        11 Kashish         Shukla          25-MAY-66         21      45000
        12 Shaswat         Puri            23-NOV-61         11      25000
        13 Gazal           Singh           02-FEB-63         11      48000
        14 Mohana          Seth            01-JUN-71         22      27000
        15 Bhavesh         Kewalramani     11-MAY-01         41      36000
        16 Bahar           Toshniwal       10-SEP-68         21      30000
        17 Vedika          Mishra          06-MAR-66         11      31000
        18 Simon           Gonsalvis       03-SEP-61         12      25000
        19 Vikrant         Gokhale         31-JUL-63         12      38000
        20 Harsh           Shrirame        13-JUN-01         31      38000
        21 Pampa           Roy             11-DEC-74         12      28000
        22 Chanchal        Bhati           08-SEP-74         22      40000
        23 Sriniwas        Chintala        25-AUG-64         31      26000
        24 Ranjit          Rungta          11-DEC-74         41      32000
        25 Mehar           Mirpuri         09-FEB-69         22      29000

15 rows selected.


ALTER TABLE STORE
            2  ADD
            3  CONSTRAINT STORE_FK_EMP_EMP_CODE FOREIGN KEY (EMP_CODE) REFERENCES
      EMP(EMP_CODE);

Table altered.
```

****

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

****

```
SELECT TO_CHAR (SYSDATE,'DD-MM-YYYY HH24:MI:SS') "TODAY" FROM DUAL;

TODAY
-------------------
18-07-2021 14:03:29
```

****

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.

****

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

EMP_CODE	EMP_FNAME	EMP_LNAME	EMP_DOB	STORE_CODE	SALARY
12	Shaswat	Puri	23-NOV-61	11	25000
14	Mohana	Seth	01-JUN-71	22	27000
16	Bahar	Toshniwal	10-SEP-68	21	30000
17	Vedika	Mishra	06-MAR-66	11	31000
18	Simon	Gonsalvis	03-SEP-61	12	25000
21	Pampa	Roy	11-DEC-74	12	28000
23	Sriniwas	Chintala	25-AUG-64	31	26000
24	Ranjit	Rungta	11-DEC-74	41	32000
25	Mehar	Mirpuri	09-FEB-69	22	29000

9 rows selected.

****

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.

****

```
SELECT E.EMP_FNAME,E.EMP_LNAME
  2   FROM EMP E JOIN STORE F
  3   ON E.STORE_CODE=F.STORE_CODE
  4   WHERE E.EMP_DOB < '01-JAN-1972' AND F.REGION_CODE=2;
```

```
EMP_FNAME       EMP_LNAME
--------------- ---------------
Kashish         Shukla
Mohana          Seth
Bahar           Toshniwal
Mehar           Mirpuri
```

****

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

****

```
SELECT
E.EMP_FNAME,E.EMP_LNAME,S.STORE_CODE,S.STORE_NAME,S.YTD_SALES,S.REGION_CODE,
S.EMP_CODE
  2  FROM EMP E JOIN STORE S
  3  ON E.EMP_CODE=S.EMP_CODE;
```

```
EMP_FNAME  EMP_LNAME  STORE_CODE STORE_NAME       YTD_SALES  REGION_CODE   EMP_CODE
-------------- -------------- ---------- ------------------_------ ---------- ---
Kashish    Shukla         21     Success Junction    1000555.76      2         11
Shaswat     Puri          41     Curiosity Circle    568000          4         12
Gazal       Singh         11     Opportunity Square  986785.4        1         13
Vedika      Mishra        12     Central Deluge      2930098.35      1         17
Simon       Gonsalvis     31     Attribute Alley     944568.66       3         18
Chanchal    Bhati         22     Database Corner     1420000.34      2         22

6 rows selected.
```

****

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

****

```
SELECT S.STORE_CODE,S.STORE_NAME,R.REGION_DESC
  2    FROM STORE S JOIN REGION R
  3    ON S.REGION_CODE=R.REGION_CODE;
```

```
STORE_CODE STORE_NAME                 REGION_DES
---------- ------------------------- ----------
        21 Success Junction          West
        22 Database Corner           West
        11 Opportunity Square        East
        31 Attribute Alley           North
        12 Central Deluge            East
        41 Curiosity Circle          South

6 rows selected.
```

****

QUERY-09: Write SQL code to list all employees who will retire during 2021. Consider age of retirement as 60 years.

****

```
SELECT *
  2  FROM EMP
  3  WHERE ROUND((SYSDATE-EMP_DOB)/365)=60;
```

```
EMP_CODE EMP_FNAME       EMP_LNAME       EMP_DOB   STORE_CODE     SALARY
---------- --------------- --------------- --------- ---------- ----------
        12 Shaswat         Puri            23-NOV-61         11      25000
        18 Simon           Gonsalvis       03-SEP-61         12      25000
```

****

QUERY-10: Write SQL code to list employee id, first name and salary for employees whose first name ends in –”T”.

****

```
SELECT EMP_CODE,EMP_FNAME,SALARY
  2  FROM EMP
  3  WHERE EMP_FNAME LIKE '%T';

no rows selected
```

```
SELECT EMP_CODE,EMP_FNAME,SALARY
  2  FROM EMP
  3  WHERE EMP_FNAME LIKE '%t';

EMP_CODE      EMP_FNAME      SALARY
-------------------------------------
        12    Shaswat         25000
        19    Vikrant         38000
        24    Ranjit          32000
```

Inference:

SQL also known as Structured Query Language is a programming Language which deals with the handling of large amount of data. Every data has different type of requirements and so we require constraints to handle such special requirements. To join the tables and perform special operation we requires keys. There are two types of keys mainly Primary Key and Foreign key. Some different types of constraints we use are NOT NULL, DEFAULT, Unique, CHECK, etc., Each of these constraints has a special function. Now there are different types of SQL Command like SELECT, CREATE, UPDATE, DELETE, etc., that can be used to perform different operations.

Conclusion:

The SQL commands' breadth and reach allow for the creation and manipulation of a wide range of database objects using the CREATE, ALTER, and DROP commands. These database objects may then be filled with data using commands like INSERT. A broad range of operations, including as SELECT, DELETE, and TRUNCATE, may be used to alter the data. Other commands, like as GRANT and REVOKE, regulate a user's access to database resources.

Viva Questions:

1. **What is SQL?**

   SQL is an abbreviation for Structured Query Language. SQL is a domain-specific programming language. It is used in programming to handle data stored in a relational database system. It is concerned with the modification, deletion, manipulation, and retrieval of data contained in relational database systems. SQL manages a huge quantity of data, especially when a significant amount of data is being written concurrently and there are too many data transactions.

2. **Discuss different key constraints in SQL.**

   Constraints are used to define the rules that govern the data in a table.

   Different key constraints are:

- NOT NULL Constraint prevents a column from having a NULL value.
- DEFAULT Constraint Sets a column's default value when none is given.
- UNIQUE Constraint Ensures that all values in a column are unique.
- PRIMARY KEY Identifies each row/record in a database table in a unique way.
- FOREIGN KEY Identifies a row/record in any of the supplied database tables in a unique way.
- The CHECK Constraint guarantees that all of the values in a column meet specified criteria.
- INDEX Used to easily build and retrieve data from a database.
- Drop Constraints
- Integrity constraints are used in relational databases to assure the correctness and consistency of the data. The idea of referential integrity governs data integrity in a relational database.

3. **Differentiate between a relation and a table.**

   A relation has no specified ordering and cannot include duplicates. A table is an unsatisfactory representation of a relation unless it has a unique key requirement.

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

   3GLs are classified as procedural languages since the programme instructions are procedure-oriented and include a sequence of stages that not only tell the computer what to do, but also how to accomplish it. 4GLs, on the other

hand, are non-procedural languages; they let users and developers to build programmes that simply define what the machine is expected to do and not how it is supposed to do it.

Languages of the third generation are also known as languages of high level, whereas languages of the fourth generation occasionally call them languages of extremely high levels. A huge amount of assembly language and machine speech instructions are created by generating 3GL statements. A fourth-generation language uses just around 10% of the statements that a third-generation language requires to complete a same job due to its reduced complexity.

The lowered complexity of fourth generation languages has also expanded the number of professionals capable of participating in software development. The majority of 4GLs are linked with data processing and databases; they represent the language used by domain specialists to create business processing sequences and rules, allowing for the effective creation of business-oriented systems. SQL (Structured Query Language) is the foundation of the majority of data-oriented fourth-generation languages. Third-generation languages include Java, C, and C++, to name a few. High-level code is typically inserted into 4GLs to implement specialised system logic.