**SQL**

Day 14

Complete the following assignment:

a)      Create the table PROGRAMMER with the given information using SQL CREATE TABLE command:

|  |  |
| --- | --- |
| **Attribute Name** | **Description/Data Type/Constraint** |
| EmpNo | Employee’s unique ID. Max. 5 characters should be numeric |
| ProjId | Project in which programmer participates. Max. 5 characters should be varchar |
| LastName | Surname of employee. Max. 30 characters. Required. |
| FirstName | Employee’s first name. Max. 30 characters. |
| HireDate | Date on which employee was hired. Date data type. |
| Language | Programming language used by programmer. Max. 15 characters |
| TaskNo | Number of the task associated with the project. Numeric column, max. 2 digits |
| Privilege | Type of privilege given to programmer. Max. 25 characters. |

b)     Insert the following data into the PROGRAMMER table:



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **EmpNo** | **LastName** | **FirstName** | **HireDate** | **ProjId** | **Language** | **TaskNo** | **Privilege** |
| 201 | Gupta | Saurav | 1/1/95 | NPR | VB | 52 | Secret |
| 390 | Ghosh | Pinky | 1/05/93 | KCW | Java | 11 | Top Secret |
| 789 | Agarwal | Praveen | 08/31/98 | Rnc | VB | 11 | Secret |
| 134 | Chaudhury | Supriyo | 07/15/95 | TIPPS | C++ | 52 | Secret |
| 896 | Jha | Ranjit | 06/15/97 | KCW | Java | 10 | Top Secret |
| 345 | John | Peter | 11/15/99 | TIPPS | Java | 52 |  |
| 563 | Anderson | Andy | 08/15/94 | NITTS | C++ | 89 | confidential |

c)     Create a table WEATHER with following data:

|  |  |  |  |
| --- | --- | --- | --- |
| **City** | **State** | **High** | **Low** |
| Calcutta | West Bengal | 105 | 90 |
| Trivandrum | Kerala | 101 | 92 |
| Mumbai | Maharashtra | 88 | 69 |
| Bangalore | Karnataka | 77 | 60 |
| New Delhi |  | 80 | 72 |

d)     Create a table BOOKS with the following data



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BookId** | **Title** | **TopicId** | **Publisher**  **Name** | **Placeof Publication** | **Price** | **PurchaseDate** | **ShelfNo** |
| 8293 | DBMS | DB1 | Prentice Hall | Mumbai | 255 | 1/1/95 | S11 |
| 5645 | DBMS | DB1 | Pearson Education | Mumbai | 655 | 1/05/93 | S12 |
| 6565 | C | C1 | TMH | Mumbai | 840 | 08/31/98 | S66 |
| 6567 | C++ | Cplus1 | ABC Publishers | Delhi | 300 | 07/15/95 | S77 |
| 4576 | JAVA | JAVA1 | Guru Govind Publications | Delhi | 500 | 06/15/97 | S87 |
| 3433 | OOPS | OOPS1 | Dave Publishers | Pune | 600 | 11/15/99 | S56 |
| 4655 | SAD | SAD1 | Sajan Publications | Cochin | 700 | 08/15/94 | S76 |

**Note:**

  The primary keys and other keys could be assumed if not mentioned.

  When inserting values try using all the 3 different ways to insert data into the table

e)     Write SQL queries to:

       i.         Saurav Gupta is assigned a different project with id NITTS and he would work with C++ now. Update this change in the PROGRAMMER table.

       i.         The books on DBMS are shifted to shelf with number S10. Please update this detail in BOOKS table.

      ii.         Supriyo Chaudhury has resigned his job. Incorporate this change in the table PROGRAMMER.

    iii.         A new column to state the nature of the climate with either of the value (rainy, cloudy, sunny, snow) is to be added in the WEATHER Table.

    iv.         Delete the table WEATHER from database.

Enum

    - It is type of class that mainly stores constants or fixed set of values

    - It implicitly abstract class, by default it is final so we cant inherit and it can implemented by a interface

interface A {

              void add();

}

enum Day1 implements A {

    SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY;

              @Override

              public void add() {

              }

}

/\*enum Day2 extends Day1{

}\*/

enum Day {

     SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY

}

public class Main {

              public static void main(String[] args)  {

       Day d1=Day.TUESDAY;

       System.out.println(d1);  //TUESDAY

       Day d2=d1;

       System.out.println(d2); //TUESDAY

       //Day d3=10; //error

       Day d3=Day.valueOf("MONDAY");

       System.out.println(d3); //MONDAY

       //Day d4=Day.valueOf("monday");

       //System.out.println(d4); //IllegalArgumentException

       System.out.println(d1==d2); //true

       System.out.println(d1.equals(d2)); //true

       System.out.println(Day.FRIDAY);  //FRIDAY

       for(Day d:Day.values()) {

                 System.out.println(d.name()+" "+d.ordinal());

       }

              }

}

Rules

1. enum values are declared first, then only we can define anything

enum Day {

              //int a=10;  //error

    SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY;

              int s=10; //correct

}

2. If we want to define anything after enum values then it should end with semicolon

3. A non final enum method can be overridden by any enum value

enum Day {

              SUNDAY {

                             public void printTemp() {

                                           System.out.println("55");

                             }

              },

              MONDAY,

              TUESDAY,

              WEDNESDAY,

              THURSDAY,

              FRIDAY,

              SATURDAY;

              private String temperature;

              public void printTemp() {

                             System.out.println("65");

              }

}

public class Main {

              public static void main(String[] args)  {

       Day d1=Day.TUESDAY;

       d1.printTemp(); //65

       Day.SUNDAY.printTemp(); //55

              }

}

4. If we have abstract method then it should be implemented by all enum value

enum Day {

              SUNDAY {

                             public void printTemp() {

                                           System.out.println("55");

                             }

              },

              MONDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("50");

                             }

              },

              TUESDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("45");

                             }

              },

              WEDNESDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("40");

                             }

              },

              THURSDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("35");

                             }

              },

              FRIDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("25");

                             }

              },

              SATURDAY {

                             @Override

                             public void printTemp() {

                                           System.out.println("36");

                             }

              };

              private String temperature;

              public abstract void printTemp();

}

public class Main {

              public static void main(String[] args)  {

       Day d1=Day.TUESDAY;

       d1.printTemp(); //45

       Day.SUNDAY.printTemp(); //55

              }

}

5. Constructor is implicitly private, called once in the beginning to create enum value

enum Day {

              SUNDAY("High") {

                             public void printTemp() {

                                           System.out.println("55");

                             }

              },

              MONDAY("Low") {

                             @Override

                             public void printTemp() {

                                           System.out.println("50");

                             }

              },

              TUESDAY("High") {

                             @Override

                             public void printTemp() {

                                           System.out.println("45");

                             }

              },

              WEDNESDAY("Moderate") {

                             @Override

                             public void printTemp() {

                                           System.out.println("40");

                             }

              },

              THURSDAY("Low") {

                             @Override

                             public void printTemp() {

                                           System.out.println("35");

                             }

              },

              FRIDAY("High") {

                             @Override

                             public void printTemp() {

                                           System.out.println("25");

                             }

              },

              SATURDAY("Low") {

                             @Override

                             public void printTemp() {

                                           System.out.println("36");

                             }

              };

              String temperature;

              /\*private\*/ Day(String temperature){

                             this.temperature=temperature;

              }

              public abstract void printTemp();

}

public class Main {

              public static void main(String[] args)  {

       Day d1=Day.TUESDAY;

       d1.printTemp(); //45

       Day.SUNDAY.printTemp(); //55

       for(Day d:Day.values()) {

                 System.out.println(d+" "+d.temperature);

       }

              }

}

enum Marks {

              ENGLISH(67),

              MATHS(100),

              SCIENCE(99),

              SOCIAL(100),

              LANGUAGE(89);

              int mark;

    Marks(int mark){

              this.mark=mark;

    }

}

public class Main1 {

              public static void main(String[] args) {

                             for(Marks m:Marks.values()) {

                                           System.out.println(m+" "+m.mark);

                             }

              }

}

6. Enum can also used in switch case

public class Main {

              public static void main(String[] args)  {

     Day1 d1=Day1.FRIDAY;

     switch(d1) {

     case SUNDAY:

               System.out.println("High");

               break;

     case MONDAY:

               System.out.println("Low");

               break;

     case TUESDAY:

               System.out.println("Moderate");

               break;

     default:

               System.out.println("No comments");

               break;

     }

              }

}

File processing system

    - we store all data inside the files

Drawbacks

1. Data redundancy - duplication of data

2. Data isolation - same data are scattered across multiple machines

3. Only particular format of data can be stored inside the file

4. No Security

5. No backup and recovery of data

6. Integrity constraints are buried under programming language

7. Only single user can access the file

Database system

     - set of logical related data - database

     - set of programs to access those data - DBMS/RDBMS

DBMS                                            RDBMS

1. No relationship between tables           1. We can relate two tables using

we have only primary key                       foreign key concepts

2. Data is stored in flat file              2. Data is stored inside the

system                                         tables

3. Only single user can access              3. Multiple user can access the

the data at a time                             data

eg: MS Access, foxpro                         eg: MySQL, Oracle, SQL Server,

                                              PostgresSQL, DB2, Sybase

Data Models

     - semantic representation of data

1. Hiearchial model

2. Network model

3. ER Model - Entity Relationship diagram

4. Relational model - represent the data in the form of tables

       Tables will contains rows(records/tuples) and columns(attributes/fields)

5. Semistructured model - represent data in xml file

DBMS/RDBMS

   - set a programs to access the data - 2 types

1. SQL - Structured Query Language

2. PLSQL - Procedural Language Structured Query Language

SQL

    - It is a english like syntax to process the data in database

Types of keys in SQL

1. Super key - any combination of columns that uniquely fetch the value from database

Consider we have Employee table - empid,ename,address,salary,dept

ename,address,dept - super keys

empid,salary - candidate key (minimal of super key)

empid - primary key (anyone of candidate key is selected as primary key)

salary - alternate key

Integrity Constraints - 5 types

1. Entity integrity constraint

       - Entity means table - constraints that applied directly on the table

       - 2 types

   1. Primary key - no null value, no duplication

   2. Unique key - no duplication but it will allow null value

2. Domain Integrity constraint

       - Domain is permitted value for the column

       - constraint that is applied to value for the column

       - 2 types

          1. NOT NULL

          2. CHECK

3. Referential Integrity Constraint

        - We can refer one table with another table using foreign key concept

        - One tables PK only will be acting as FK in another table, so first we have to create PK table then only we have to create foreign table

        - Foreign key contains duplicate values and also contain null values

        - FK column should contain only the values that are present in PK column

        - whenever we have one to many,many to one, many to many relationship

        - First we have to delete PK table then we can delete FK table

one person places many orders - one to many

Person table

pid(pk)      pname

  1            A

  2            B

  3            C

  4            D

  5            E

Orders table

ordid(pk)    ordername     pid(FK)

  100          Coffee        1

  101          Tea           2

  102          Chips         3

  103          Juice         4

  105          Biscuits      5

  106          Cakes         1

  107          Cola

4. Default constraint

        - By default if we didnt provide any value for the column, irrespective of the datatype always it will contain null value

        - But we dont want to have null value, instead we want to provide some default value, then we can use default constraint

5. NULL constraint

Composite primary key

    - If table contains more than one column as primary key where its combination should be unique

A(PK)      B(PK)      C

1          1          1     11

2          1                21

1          4          3     14

1          2          4     12

Note

1. By default if we didnt provide any value for the column, irrespective of the datatype always it will contain null value

2. SQL is a case sensitive - tablename, columnname is not case sensitivity, but the values stored inside the columns are case sensitive

   'ram'!='Ram'

SQL Datatypes

1. Numeric datatype - store any numbers +ve or -ve

      - numeric(p,q)   p=total width, q=precision after decimal point

           numeric(5,2) - 123.45

      - numeric(p)

           numeric(5) - 12345

      - int

      - integer

      - tinyint -128 to 127

      - smallint -32768 to 32767

      - mediumint -8388608 to 8388607

      - bigint -9223372036854775808 to 9223372036854775807

      - decimal(p,q)/fixed(p,q)

      - float

      - double

2. Character datatype

       - char(n) - fixed length of char

             char(10)='ram'

       - varchar(n) - variable length of char

             varchar(10)='ram'

3. Date datatype

       - date - only date(yyyy-MM-dd)

       - datetime - both date and time (yyyy-MM-dd hh:mm:ss)

       - timestamp - both date and time(yyyyMMddhhmmss)

4. LOB datatype - Large Object datatype

       - BLOB(Binary Large Object) - used to store images, audio, video

       - CLOB(Character Large Object) - used to store very long string

SQL Operators

1. +,-,\*,/,<,>,<=,>=,!=

2. ALL

3. ANY

4. AND

5. OR

6. IN, NOT IN

7. LIKE, NOT LIKE - Pattern matching operator

8. BETWEEN, NOT BETWEEN- - Range operator

9. IS NULL, IS NOT NULL

10. EXISTS, NOT EXISTS

11. UNION, UNION ALL, INTERSECT, MINUS - Set operator

mysql> show databases;  - to view all databases

mysql> create database javabatch;  - create a new db

Query OK, 1 row affected (0.03 sec)

mysql> use javabatch;  - use the db

Database changed

mysql> show tables;  - to view all tables in db

Empty set (0.02 sec)

mysql> drop database batch2;  - delete the db

Query OK, 13 rows affected (0.24 sec)

SQL Statement

1. DDL statement - Data Defination Language - 4 stmt

                 - create, alter, drop, truncate

a. create - create a new table in db

mysql> create table customer(custid integer,name varchar(20),age int);

Query OK, 0 rows affected (0.07 sec)

mysql> desc customer;   - to view structure of table

+--------+-------------+------+-----+---------+-------+

| Field  | Type        | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| custid | int         | YES  |     | NULL    |       |

| name   | varchar(20) | YES  |     | NULL    |       |

| age    | int         | YES  |     | NULL    |       |

+--------+-------------+------+-----+---------+-------+

3 rows in set (0.01 sec)

mysql> create table customer3(sno int unique,custid integer primary key AUTO\_INCREMENT,name varchar(20),age int);

Query OK, 0 rows affected (0.07 sec)

mysql> create table customer4(sno int,custid integer,name varchar(20),age int,constraint cust\_pk primary key(sno,custid));

Query OK, 0 rows affected (0.05 sec)

mysql> create table customer5(sno int unique,custid integer primary key AUTO\_INCREMENT,name varchar(20) NOT NULL,age int default 25);

Query OK, 0 rows affected (0.06 sec)

mysql> create table department(deptid int primary key,dname varchar(20));

Query OK, 0 rows affected (0.06 sec)

mysql> create table employee(empid int primary key,ename varchar(20) NOT NULL, age int, did int references department(deptid));

Query OK, 0 rows affected (0.07 sec)

- Copy both data and structure from one table and create an another table

mysql> create table customer6 as select \* from customer5;

Query OK, 0 rows affected (0.05 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> create table customer7 as select custid,name from customer5;

Query OK, 0 rows affected (0.05 sec)

Records: 0  Duplicates: 0  Warnings: 0

- Copy only structure from one table and create an another table

mysql> create table customer8 as select \* from customer5 where 1=20;

Query OK, 0 rows affected (0.05 sec)

Records: 0  Duplicates: 0  Warnings: 0

b. alter

1. To rename table name

mysql> alter table customer8 rename to cust8;

Query OK, 0 rows affected (0.04 sec)

2. To rename column name

mysql> alter table cust8 rename column age to custage;

Query OK, 0 rows affected (0.01 sec)

Records: 0  Duplicates: 0  Warnings: 0

3. To add new column to exiting table

mysql> desc cust8;

+---------+-------------+------+-----+---------+-------+

| Field   | Type        | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| sno     | int         | YES  |     | NULL    |       |

| custid  | int         | NO   |     | 0       |       |

| name    | varchar(20) | NO   |     | NULL    |       |

| custage | int         | YES  |     | 25      |       |

+---------+-------------+------+-----+---------+-------+

4 rows in set (0.00 sec)

mysql> alter table cust8 add address varchar(40);

Query OK, 0 rows affected (0.02 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table cust8 add city varchar(40),add state varchar(40);

Query OK, 0 rows affected (0.03 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cust8;

+---------+-------------+------+-----+---------+-------+

| Field   | Type        | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| sno     | int         | YES  |     | NULL    |       |

| custid  | int         | NO   |     | 0       |       |

| name    | varchar(20) | NO   |     | NULL    |       |

| custage | int         | YES  |     | 25      |       |

| address | varchar(40) | YES  |     | NULL    |       |

| city    | varchar(40) | YES  |     | NULL    |       |

| state   | varchar(40) | YES  |     | NULL    |       |

+---------+-------------+------+-----+---------+-------+

7 rows in set (0.00 sec)

mysql> alter table cust8 add email varchar(40) after custage;

Query OK, 0 rows affected (0.09 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cust8;

+---------+-------------+------+-----+---------+-------+

| Field   | Type        | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| sno     | int         | YES  |     | NULL    |       |

| custid  | int         | NO   |     | 0       |       |

| name    | varchar(20) | NO   |     | NULL    |       |

| custage | int         | YES  |     | 25      |       |

| email   | varchar(40) | YES  |     | NULL    |       |

| address | varchar(40) | YES  |     | NULL    |       |

| city    | varchar(40) | YES  |     | NULL    |       |

| state   | varchar(40) | YES  |     | NULL    |       |

+---------+-------------+------+-----+---------+-------+

8 rows in set (0.00 sec)

4. Modify only datatype of existing column

mysql> alter table cust8 modify address varchar(50) NOT NULL;

Query OK, 0 rows affected (0.10 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table cust8 modify city varchar(50) NOT NULL,modify state varchar(50) NOT NULL;

Query OK, 0 rows affected (0.09 sec)

Records: 0  Duplicates: 0  Warnings: 0

mysql> desc cust8;

+---------+-------------+------+-----+---------+-------+

| Field   | Type        | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| sno     | int         | YES  |     | NULL    |       |

| custid  | int         | NO   |     | 0       |       |

| name    | varchar(20) | NO   |     | NULL    |       |

| custage | int         | YES  |     | 25      |       |

| email   | varchar(40) | YES  |     | NULL    |       |

| address | varchar(50) | NO   |     | NULL    |       |

| city    | varchar(50) | NO   |     | NULL    |       |

| state   | varchar(50) | NO   |     | NULL    |       |

+---------+-------------+------+-----+---------+-------+

5. To drop existing column in table

mysql> alter table cust8 drop column email;

Query OK, 0 rows affected (0.10 sec)

Records: 0  Duplicates: 0  Warnings: 0

6. To add constraint to the table after creating

mysql> alter table customer add constraint cu\_pk primary key(custid);

Query OK, 0 rows affected (0.10 sec)

Records: 0  Duplicates: 0  Warnings: 0

7. To enable/disable primary key or unique key

mysql> alter table customer disable keys;

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> alter table customer enable keys;

Query OK, 0 rows affected, 1 warning (0.01 sec)

To enable/disable foreign key

SET FOREIGN\_KEY\_CHECKS=0;

SET FOREIGN\_KEY\_CHECKS=1;

c. drop

       - drop the data as well as the table structure

mysql> drop table customer;

Query OK, 0 rows affected (0.04 sec)

mysql> drop table customer1,customer2;

Query OK, 0 rows affected (0.05 sec)

mysql> drop table if exists cust8;

Query OK, 0 rows affected (0.03 sec)

d. truncate

       - used to drop the data but the structure will be maintained but we cannot rollback

>truncate table customer4;

2. DML - Data Manipulation Language - 3 stmt

       - insert, update, delete

a. insert - used to insert new row into the table

- insert all columns

mysql> insert into customer6 values(1,100,'Ram',24);

Query OK, 1 row affected (0.01 sec)

mysql> insert into customer6 values(2,101,'Sam',25);

Query OK, 1 row affected (0.01 sec)

- insert only into particular columns

mysql> insert into customer6(sno,custid,name) values(3,103,'Raj');

Query OK, 1 row affected (0.01 sec)

mysql> select \* from customer6;

+------+--------+------+------+

| sno  | custid | name | age  |

+------+--------+------+------+

|    1 |    100 | Ram  |   24 |

|    2 |    101 | Sam  |   25 |

|    3 |    103 | Raj  |   25 |

+------+--------+------+------+

3 rows in set (0.01 sec)

- insert data from another table, make sure both table should have same number of col and same column datatype

mysql> select \* from customer5;

Empty set (0.00 sec)

mysql> insert into customer5 select \* from customer6;

Query OK, 3 rows affected (0.00 sec)

Records: 3  Duplicates: 0  Warnings: 0

mysql> select \* from customer5;

+------+--------+------+------+

| sno  | custid | name | age  |

+------+--------+------+------+

|    1 |    100 | Ram  |   24 |

|    2 |    101 | Sam  |   25 |

|    3 |    103 | Raj  |   25 |

+------+--------+------+------+

3 rows in set (0.00 sec)

- with single insert query we want to insert multiple rows

mysql> insert into customer6(sno,custid,name,age) values

    -> (4,104,'Tam',34),(5,105,'Tim',25),(6,106,'Jim',27);

Query OK, 3 rows affected (0.01 sec)

Records: 3  Duplicates: 0  Warnings: 0

b. update - to update value of existing row

mysql> update customer6 set age=29 where custid=104;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1  Changed: 1  Warnings: 0

mysql> update customer6 set age=26,name='Jimmy' where custid=106;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1  Changed: 1  Warnings: 0

c. delete - used to delete the data from table but the structure will be maintained we can rollback

mysql> delete from customer6 where sno=6;

Query OK, 1 row affected (0.01 sec)

mysql> delete from customer6;

Query OK, 5 rows affected (0.01 sec)

Day 15

Create the table WORLDCITY with the following data.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **City** | **Country** | **Continent** | **Latitude** | **NorthSouth** | **Longitude** | **EastWest** |
| Athens | Greece | Europe | 37.59 | N | 23.44 | E |
| Atlanta | United States | North America | 33.45 | N | 84.23 | W |
| Dallas | United States | North America | 32.47 | N | 96.47 | W |
| Nashville | United States | North America | 36.09 | N | 86.46 | W |
| Victoria | Canada | North America | 48.25 | N | 123.21 | W |
| Peterborough | Canada | North America | 44.18 | N | 79.18 | W |
| Vancouver | Canada | North America | 49.18 | N | 123.04 | W |
| Toledo | United States | North America | 41.39 | N | 83.82 | W |
| Warsaw | Poland | Europe | 52.15 | N | 21.00 | E |
| Lima | Peru | South America | 12.03 | S | 77.03 | W |
| Rio De Janeiro | Brazil | South America | 22.43 | S | 43.13 | W |
| Santiago | Chile | South America | 33.27 | S | 70.40 | W |
| Bogota | Colombia | South America | 04.36 | N | 74.05 | W |
| Buenos Aires | Argentina | South America | 34.36 | S | 58.28 | W |
| Quito | Ecuador | South America | 00.13 | S | 78.30 | W |
| Caracas | Venezuela | South America | 10.30 | N | 66.56 | W |
| Madras | India | Asia | 28.36 | N | 77.12 | E |
| Bombay | India | Asia | 18.58 | N | 72.50 | E |
| Manchester | England | Europe | 51.30 | N | 0.0 | null |
| Moscow | Russia | Europe | 55.45 | N | 37.35 | E |
| Paris | France | Europe | 48.52 | N | 2.20 | E |
| Shenyang | China | Asia | 41.48 | N | 123.27 | E |
| Cairo | Egypt | Africa | 30.03 | N | 31.15 | E |
| Tripoli | Lybia | Africa | 32.54 | N | 13.11 | E |
| Beijing | China | Asia | 39.56 | N | 116.24 | E |
| Rome | Italy | Europe | 41.54 | N | 12.29 | E |
| Tokyo | Japan | Asia | 35.42 | N | 139.46 | E |
| Sydney | Australia | Australia | 33.52 | S | 151.13 | E |
| Sparta | Greece | Europe | 37.05 | N | 22.27 | E |
| Madrid | Spain | Europe | 40.24 | N | 3.41 | W |

Complete the following exercise:

1.              For all the different countries contained in the WORLDCITY table, display their names and the continent in which they are located. Make sure that no country name is duplicated.

1.              Write an SQL query to display the list of the city and country for all the cities that begin with letter R.

2.              Write an SQL query to display the list of the city and country for all the cities that end with letter A.

3.              Write an SQL query to display the list of the city and country for all the cities that begin with letter M and have exactly six letters in them.

4.              Write an SQL query to display the list of the city and country for all the cities that contain an A as the second letter.

Complete the following exercise:

1. Create a following tables

**CUSTOMER Table Schema:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN NAME** | **DESCRIPTION** | **DATATYPE** | **SIZE** | **CONSTRAINTS** |
| CUSTID | Customer Id | Number |  | Primary Key |
| FNAME | First Name | Character | 30 |  |
| LNAME | Last Name | Character | 30 |  |
| ADDRESS | Customer Address | Character | 50 |  |
| PHONENO | Phone | Number |  | Not Null |
| CITY | City | Character | 20 |  |
| COUNTRY | Country | Character | 20 |  |
| DATEFIRSTPURCHASED | Date of the first purchase by the customer | Date |  |  |
| SUPPLIERID | Supplier Information | Number |  | Foreign Key |

**Data for CUSTOMER table:**



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CUSTID** | **FNAME** | **LNAME** | **ADDRESS** | **PHONENO** | **CITY** | **COUNTRY** | **DATEFIRSTPURCHASED** | **SUPPLIERID** |
| 1001 | Das | Jeyaseelan | 119, park Avenue, II street, | 9841093428 | Coimbatore | India | 10-jan-2004 | 1 |
| 2001 | Gopi | Govindraj | 241, I floor, Kamaraj street, Madippakkam | 9444124590 | Chennai | India | 25-mar-2005 | 4 |
| 1201 | Dilip | Kishore | 43, II Avenue, Anna Nagar | 9997234534 | Bangalore | India | 20-aug-2004 | 2 |
| 1300 | Aanand | Chowdhury | 42/1 sector 1, II Street | 9841054348 | Bangalore | India | 15-may-2005 | 2 |
| 1220 | Chandra | Nagarajan | 83, lal bagh | 98410672356 | Bangalore | India | 12-feb-2006 | 4 |
| 1221 | Abhishek | Kumar | 13,kishori park, | 94447623901 | Chennai | India | 15-may-2004 | 1 |
| 1320 | Nikhil | Pandit | 218, alwaanya street | 94448923091 | Salem | India | 21-apr-2006 | 3 |
| 1222 | Meenu | Monica | C11, church road | 98410563421 | Trichy | India | 30-aug-2004 | 1 |
| 1225 | Pavan | Kumar | 128/A, North Mada Street | 99934782103 | maduari | India | 18-aug-2004 | 4 |

**SUPPLIER Table Schema:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN NAME** | **DESCRIPTION** | **DATATYPE** | **SIZE** | **CONSTRAINTS** |
| SUPPLIERID | Supplier Id | Number |  | Primary Key |
| SNAME | Supplier Name | Character | 30 |  |
| SCITY | Supplier City | Character | 30 |  |
| SPHONE | Supplier Phone | Number |  | Not Null |
| EMAIL | Email id of Supplier | Character | 50 | Unique |

**Data for SUPPLIER table:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SUPPLIERID** | **SNAME** | **SCITY** | **SPHONE** | **EMAIL** |
| 1 | Dilip | Chennai | 8999900000 | [dilip@abc.co.in](mailto:dilip@abc.co.in) |
| 2 | Tarun | Madurai | 8999911111 | [tarun@xyz.com](mailto:tarun@xyz.com) |
| 3 | Naresh | Coimbatore | 8999922222 | [g.naresh@xyzl.com](mailto:g.naresh@xyzl.com) |
| 4 | Ganesan | Trichy | 8999933333 | [Ganesan\_83@ijk.com](mailto:Ganesan_83@ijk.com) |

create table cust1(custid int primary key,fname varhcar(30),lname varchar(30),address varchar(30),phoneno int,city varchar(20),country varchar(20),datefirstpur date,);

**ORDERS Table Schema:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN NAME** | **DESCRIPTION** | **DATATYPE** | **SIZE** | **CONSTRAINTS** |
| ORDERID | Order Number | Number |  | Primary Key |
| ORDERDATE | Date of Order | Date |  |  |
| CUSTID | Customer Identity | Number |  |  |
| QUANTITY | Quantity Ordered | Number |  | Check (Quantity >0) |
| ITEMID | Item Code | Number |  | Foreign Key |

**Data for ORDERS Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ORDERID** | **ORDERDATE** | **CUSTID** | **QUANTITY** | **ITEMID** |
| 1 | 12-jan-2004 | 1001 | 30 | 25 |
| 2 | 6-may-2005 | 1202 | 38 | 24 |
| 3 | 16-dec-2006 | 1220 | 10 | 22 |
| 4 | 21-may-2004 | 1233 | 12 | 21 |

**ITEMS Table Schema:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN NAME** | **DESCRIPTION** | **DATATYPE** | **SIZE** | **CONSTRAINTS** |
| ITEMID | Item Code | Number |  | Primary Key |
| ITEMNAME | Item Name | Character | 35 | Not Null |
| SUPPLIERID | Supplier Code | Number |  | Foreign Key |
| MINQTY | Minimum Qty that can be ordered | Number |  | Not Null |
| MAXQTY | Maximum Qty that can be ordered | Number |  | Not Null |
| Price | Price per unit | Number(5,2) |  |  |

**Data for ITEMS Table:**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ITEMID** | **ITEMNAME** | **SUPPLIERID** | **MINQTY** | **MAXQTY** | **Price** |
| 20 | Pears Soap | 4 | 7 | 20 | 30.00 |
| 21 | V.V.D. Coconut oil 200 ml | 2 | 8 | 15 | 79.00 |
| 22 | Ponds powder 400g | 3 | 6 | 25 | 106.00 |
| 23 | Reynolds pen- blue | 1 | 10 | 30 | 15.00 |
| 24 | Reynolds pen- black | 1 | 10 | 30 | 16.00 |
| 25 | Mysore sandal soap | 4 | 7 | 25 | 25.00 |
| 26 | Fair & lovely cream- 50g | 3 | 5 | 15 | 55.00 |
| 27 | Rexono deo spary | 2 | 5 | 20 | 100.00 |
| 28 | Dove soap | 4 | 7 | 15 | 85.00 |

b)     Write SQL queries to

       i.         Display all customers from Chennai.

      ii.         Display the details of all customers who purchased from the supplier 2.

    iii.         Display CUSTID, FNAME, LNAME of all customers whose purchase date is after January 2005.

    iv.         Display the details of all suppliers who are from location Coimbatore.

      v.         Display the details of all suppliers whose name stars with G.

    vi.         Display the details of all customers, who do not have the alphabet e in their LNAME.

   vii.         Display the details of the entire customer whose first date of purchase is in 2006 and arrange it in descending order.

  viii.         Display the details of all the orders where the quantity is less than 35.

    ix.         Display the details of the items supplied by supplier 4.

      x.         Display the details of all items where SUPPLIERID is 3 and the MINQTY is greater than 7 order by ITEMID

SQL Statements

1. DDL

2. DML

3. DQL - Data Query Language

1. select - used to select the data from table

> select \* from customers;  - select all rows and columns from table

> select custid,name from customers; - select all rows but only specific column

> select \* from customers where custid>5;   - select all columns but specific rows

>select custid,name from customers where custid>5;   - select only specific column and specific rows

>select order.orderid, supplier.supname from order,supplier where order.orderid=supplier.orderid;     - select from multiple table

>select o.orderid, s.supname from order o,supplier s where o.orderid=s.orderid;   - create an alias for table to select data from                          multiple tables

>select distinct city from customers;   - To avoid duplicates values

>select \* from suppliers where city='Chennai' and type='PC manufacture';

>select \* from suppliers where city='Chennai' or type='PC manufacture';

IS NULL, IS NOT NULL - to check null values

>select \* from emp where dept is null;

>select \* from emp where dept is not null;

Pattern Matching Operator - LIKE, NOT LIKE

     - used to fetch data based on some pattern - applied only for varchar

     - 2 wildcard characters

           % - anything

           \_ - single value

>select \* from suppliers where supname like 'R%';  - select all suppliers whose name should start with R

>select \* from suppliers where supname like '%bob%';

>select \* from suppliers where supname not like 'T%';

>select \* from suppliers where supname like 'Sm\_th';

>select \* from orders where orderid like '1000\_';

supname

Hello%

HelloHo

HelloBye

Hello%

>select \* from suppliers where supname like 'Hello%';   - all 4 rows

If we want to include %,\_ at time of pattern matching then we have to use escape sequence

>select \* from suppliers where supname like 'Hello!%' ESCAPE '!';  - Return 2 rows

Gana

Girl

Giri

G%

>select \* from suppliers where supname like 'G%';

>select \* from suppliers where supname like 'G&%' ESCAPE '&';

Hello

Hello%%

Hi%

>select \* from suppliers where supname like 'H%';

>select \* from suppliers where supname like 'H%!%' ESCAPE '!';

>select \* from suppliers where supname like 'H%\%';

IN, NOT IN  - used of using multiple or operator

>select \* from suppliers where supname='HCL' or supname='CTS' or supname='TCS';

>select \* from suppliers where supname in('HCL','CTS',TCS');

>select \* from orders where orderid not in(1000,2000,3000);

BETWEEN, NOT BETWEEN - Range operator - includes the  ranges

>select \* from suppliers where supid between 1000 and 2000;

>select \* from suppliers where supid>=1000 and supid<=2000;

Set operator - UNION, UNION ALL, INTERSECT, MINUS

>select supid from supplier

      union

select supid from order;

        - select both supid from supplier and order table, if there is any duplication it will be considered only once

>select supid from supplier

      union all

select supid from order;

       - select both supid from supplier and order table, and print everything with duplication

>select supid from supplier

      intersect

select supid from order;

       - select only common supid from supplier and order table

>select supid from supplier

      minus

select supid from order;

       - remove supid from supplier that is present in order table

SQL Functions

1. Numeric function - apply these function to numbers

1. abs(n) - return absolute value

>select abs(9);  9

>select abs(-9);  9

2. ceil(n) - largest number >= n

>select ceil(9.5);  10

>select ceil(-9.5); -9

3. floor(n) - nearest number <= n

>select floor(9.5);  9

>select floor(-9.5); -10

4. mod(m,n) - mod(7,2) = 1

5. powe(m,n) - power(3,2) = 9

6. sqrt(n)

7. exp(n)

8. log(n)

9. sin(n)

10. cos(n)

11. tan(n)

12. round(n) - round to nearest decimal value

>select round(125.07);  125

>select round(125.78);  126

>select round(125.67,1); 125.7

>select round(125.11,1); 125.1

>select round(125.67,2); 125.67

>select round(125.678,2);  125.68

>select round(125.67,-1);  130         120   125   130

   -1 represents 10's, if the value is greater than mid value it returns higher value otherwise it returns lower value

>select round(222.12,-1);  220     220  225   230

>select round(222.12,-2);  200       200   250    300

   -2 represents 100's, if the value is greater than mid value it returns higher value otherwise it returns lower value

>select round(673.12,-2);  700      600   650   700

>select round(3567.12,-3); 4000       3000   3500  4000

   -3 represents 1000's, if the value is greater than mid value it returns higher value otherwise it returns lower value

13. truncate(n)

>select truncate(125.13);   error

>select truncate(125.13,1);  125.1

>select truncate(125.67,1);  125.6

>select truncate(125.678,2); 125.67

>select truncate(125.67,-1); 120

    -1 represents 10's, if the value is greater or lesser than mid value it always returns only lower value

>select truncate(435.12,-2); 400

2. Character function - apply only on varchar

1. ascii(n) - return ascii value of left most char

>select ascii('a');  97

>select ascii('bad'); 98

2. char\_length(n)

   character\_length(n)

   length(n)

     - used to find length of string

>select char\_length('hello');  5

>select character\_length('hello');  5

>select length('hello');  5

3. concat()/space operator - concatenate two or more string

>select concat('The answer is',24);   The answer is 24

>select concat('The answer is', 10+10);  The answer is 20

>select concat('The answer is', '10+10');  The answer is 10+10

>select concat('The answer is', null);  null

>select 'The answer is' 24;  The answer is 24

4. concat\_ws() - concatenate two or more expr and adds a separator between each concatenated expr

>select concat\_ws('+','a','b','c');  a+b+c

>select concat\_ws('xyz','A','B','C');  AxyzBxyzC

>select concat\_ws(null,'1','2');  null

5. field() - return the posiiton of a value in a list of values

>select field('b','a','b','c','d','e','f');  2

>select field(15,10,20,15,34);  3

>select field('c','a','b'); 0

>select field(null,'a','b');  null

6. find\_in\_set() - return the posiiton of a value in a comma delimited string

>select find\_in\_set('b','a,b,c,d,e,f'); 2

>select find\_in\_set(2,'3,4,1,5,2');  5

7. format() - used to format number with comma and rounding to nearest decimal places

>select format(123453.6789,2); 1,23,453.68

8. instr() - return location of substring in a string

>select instr('hello','h');  1

>select instr('hello','e');  2

>select instr('hello','i');  0

9. lcase()

   lower()

      - convert string to lowercase

10. left() - extract leftmost char from a string

>select left('hello',1);  h

>select left('hello',4);  hell

>select left('hello',20);  hello

11. right() - extract rightmost char from a string

>select right('hello',1);  o

>select right('hello',4);  ello

>select right('hello',20);  hello

12. locate() - return the location of first occurence of substring in a string

>select locate('H','Hello');  1

>select locate('l','Hello');  3

>select locate('l','Hello',3); 4  - from 3rd positin, first occurence of l

>select locate('l','heelo',3); 4

>select locate('e','Technology internet'); 2

>select locate('e','technology internet',3); 15

>select locate('z','technology internet',3); 0

13. lpad(char1,n,char2) - char1 is left padded with char2 of length n

>select lpad('world',10,'hello');  helloworld

>select lpad('world',11,'hello');  hellohworld

>select lpad('world',12,'hello');  helloheworld

>select lpad('world',13,'hello');  hellohelworld

>select lpad('world',9,'hello');  hellworld

>select lpad('world',4,'hello');  worl

14. rpad(char1,n,char2) - char1 is right padded with char2 of length n

>select rpad('world',10,'hello');  worldhello

>select rpad('world',11,'hello');  worldhelloh

>select rpad('world',12,'hello');  worldhellohe

>select rpad('world',13,'hello');  worldhellohel

>select rpad('world',9,'hello');   worldhell

>select rpad('world',4,'hello');   worl

15. ltrim() - removes all spaces from left side

>select ltrim(' Hello '); Hello\_

>select ltrim(' Hello world ');  Hello world\_

16. rtrim() - removes all spaces from right side

>select rtrim(' Hello '); \_Hello

>select rtrim(' Hello world ');  \_Hello world

17. trim() - trims both leading and trailing space

>select trim(leading from ' hello '); hello\_

>select trim(trailing from ' hello '); \_hello

>select trim(both from ' hello '); hello

>select trim(leading '0' from '00012340'); 12340

>select trim(trailing '1' from '111hello111'); 111hello

18. ucase()

    upper()

       - convert string to uppercase

19. mid() - used to extract substring from a string

>select mid('technology',5,2);   no

>select mid('technology',1,4);   tech

>select mid('technology',-7,4);  hnol

>select mid('technology',-3,3);  ogy

20. substr() - used to extract substring from a string

>select substr('technology.com',5);  nology.com

>select substr('technology.com',1,4);  tech

>select substr('technology.com',-3,3);  com

>select substr('technology.com' from 1 for 4); tech

21. substring() - used to extract substring from a string

>select substring('technology.com',5);  nology.com

>select substring('technology.com',1,4);  tech

>select substring('technology.com',-3,3);  com

>select substring('technology.com' from 1 for 4); tech

22. substring\_index() - return substring from string before number of occurences of delimiter

>select substring\_index('www.technology.com','.',1); www

>select substring\_index('www.technology.com','.',2); [www.technology](https://apc01.safelinks.protection.outlook.com/?url=http%3A%2F%2Fwww.technology%2F&data=05%7C01%7Ckajavevaibhav.mahdev%40hcl.com%7C2f0a0b58444141493f8d08dbc8ade704%7C189de737c93a4f5a8b686f4ca9941912%7C0%7C0%7C638324421725996332%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=BcnsRXEvtKddqvMtwPEr3rc%2BIp32bfZq59vg%2FCdoE7I%3D&reserved=0)

>select substring\_index('www.technology.com','.',-1); com

23. strcmp() - compare 2 string

>select strcmp('hello','hello');  0

>select strcmp('cat','hello');  -1

>select strcmp('hello','cat');  1

24. space() - return a string with specified number of space

>select space(3);  '   '

25. reverse() - reverse the string

>select reverse('abcd');  dcba

26. replace()

>select replace('abc abc','a','B');  Bbc Bbc

3. Aggregate function/Group function

       - used to group any value

        - It will ignore the null value at a time of grouping

        - sum(),avg(), max(), min(), count(), stddev(), median()

        - For date datatype, we can apply only  max(), min(), count()

Employee table

empid(int)  name(varchar)  gender(varchar) salary(double) dob(date) - consider we have totally 30 rows

What we provide in select list that only will be coming as column name in output

>select count(\*) from employee;

count(\*)

-------

30

>select count(gender) from employee;

count(gender)

-------------

30

>select count(distinct gender) from employee;

count(distinct gender)

-------------

2

Create an Alias for the column

>select count(\*) as "Number of Employees" from employee;

Number of Employees

-------------------

30

>select count(\*) "Number of Employees" from employee;

Number of Employees

-------------------

30

>select count(\*) NumberofEmployees from employee;

NumberofEmployees

-------------------

30

>select sum(salary) as "Total Salary" from employee;

>select max(dob) as DOB from employee;

>select min(dob) DOB from employee;

>select stddev(salary) from employee;

>select median(salary) from employee;

order by clause

     - used for sorting purpose only at time of displaying

     - It should be always present in the last of select stmt

select - from - where - group by - having - order by

     - By default sorted in ascending order

     - column index represent the column from select list not from table

     - If we sort on two or more columns, then by default it will be sorted only on first column, in the first column if there is any repition for those state column alone, the city will be sorted in desc order

>select supplier\_city,supplier\_state from supplier where supname='IBM' order by supplier\_city;

>select supplier\_city,supplier\_state from supplier where supname='IBM' order by supplier\_city asc;

>select supplier\_city,supplier\_state from supplier where supname='IBM' order by supplier\_city desc;

Supplier table

sup\_id supname supplier\_city supplier\_state address

>select supplier\_city,supplier\_state from supplier where supname='IBM' order by 2 desc;  //column index represent the column from select list not from table

Supplier

supplier\_city    supplier\_state

Chennai            Tamilnadu

Allepey            Kerela

Coimbatore         Tamilnadu

Mumbai             Maharastra

Pune               Maharastra

>select supplier\_city,supplier\_state from supplier where supname='IBM' order by supplier\_state asc, supplier\_city desc;

Supplier

supplier\_state    supplier\_city

Kerela              Allepey

Maharastra          Pune

Maharastra          Mumbai

Tamilnadu           Coimbatore

Tamilnadu           Chennai

group by clause

     - used to group the values in a table based on some aggregate function

Employee table

empid       ename       dept        salary

1             A          HR          4000

2             B          IT          6000

3             C          HR          6000

4             D          Admin       5000

5             E          IT          10000

6             F          Admin       6000

7             G          HR          5000

We want to calculate the total sum of salary

>select sum(salary) from employee;

We want to calculate the total sum of salary in each dept

>select dept,sum(salary) from employee group by dept;

dept    sum(salary)

HR       15000

IT       16000

Admin    11000

having clause

    - used to exclude the result from groupby clause

>select dept,sum(salary) from employee group by dept having sum(salary)>14000

order by dept asc;

dept    sum(salary)

HR       15000

IT       16000