**DWARAKA DOSS GOVERDHAN DOSS VAISHNAV COLLEGE**

ARUMBAKKAM, CHENNAI-106



PG DEPARTMENT OF COMPUTER SCIENCE

**MY SQL LAB**

Name :

Register Number :

Class : M.SC. INFORMATION TECHNOLOGY FEBRUARY - 2021

# DWARAKA DOSS GOVERDHAN DOSS VAISHNAV COLLEGE

ARUMBAKKAM, CHENNAI-106



PG DEPARTMENT OF COMPUTER SCIENCE

**BONAFIDE CERTIFICATE**

Certified that this is a Bonafide record of practical work done by Mr/Ms.……………………………............ RegNo.…………………………......... of I M.Sc. (Information Technology) during the academic year 2020-2021.

**FACULTY IN CHARGE HEAD OF THE DEPARTMENT**

Submitted for the Practical Examination held on

in PG

Department of Computer Science at Dwaraka Doss Goverdhan Doss Vaishnav College.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **DATE** | **TOPIC** | **PAGE NO** | **SIGN** |
| 1 |  | **DDL COMMANDS**   1. Create database 2. Create tables 3. Describe tables 4. Create view 5. Drop view 6. Alter table   Add,Modify,Delete,Rename Fields   1. Truncate table 2. Drop table   i) Drop database |  |  |
| 2 |  | **DML COMMANDS**   1. Insert 2. Update 3. Delete 4. Select    * Where clause    * Limit clause    * Order by    * Distinct    * Group by    * Having clause    * And, OR, Not    * Range Test Using Between and Not Between |  |  |
| 3 |  | **CONSTRAINTS**   1. NOT NULL 2. UNIQUE 3. PRIMARY KEY 4. FOREIGN KEY 5. ENUM 6. SET |  |  |

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 |  | **MYSQL OPERATORS**  a)ARITHMETIC OPERATORS b)COMPARISION OPERATORS  c) LOGICAL OPERATORS |  |  |
| 5 |  | **MYSQL FUNCTIONS**   1. Mathematical Functions    * RAND()    * ABS()    * SIN()    * CEIL()    * FLOOR()    * POW()    * SQRT()    * DEGREES() 2. Aggregate Functions    * MIN()    * MAX()    * AVG()    * SUM()    * COUNT()    * STD()    * VARIANCE() 3. String Functions    * LENGTH()    * UPPER()    * LOWER()    * RPAD()    * LPAD()    * REVERSE()    * LEFT()    * RIGHT()    * SUBSTRING()    * CONCAT() |  |  |

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | d) System Functions   * DAY() * YEAR() * MONTH() * NOW() * CURTIME() * CURDATE() * DATEDIFF() * DATE\_FORMAT() * DATEADD() * DATESUB () |  |  |
| 6 |  | **JOINS**   * Inner Join * Left Outer Join * Right Outer Join |  |  |
| 7 |  | **SUBQUERIES**   * Scalar subquery * Table subquery * Correlated subquery * Nested subquery |  |  |

|  |  |
| --- | --- |
| **EX.NO :01** | **DATA DEFINITION LANGUAGE** |
| **DATE :** |  |

# DATA DEFINTION LANGUAGE:

The Data Definition Language is a special language which is expressed by a set of definitions.

The commands available in DDL are:

* Createdatabase
* Createtable
* Createview
* Dropview
* Altertable
* Truncatetable
* Drop table
* Drop database

# CREATE DATABASE

**SYNTAX :**

**Create database database\_name;**

**QUERY:**

Mysql> create database inventory;

Mysql> use inventory;

# CREATE TABLE

**SYNTAX :**

CREATE TABLE table\_name (

Col\_name1 datatype (),

Col\_name2 datatype (), …

Col\_namen datatype (),

);

**Product Table:**

create table deptIT

(stidint(20),

stname char(20),

dept char(10),

gender char(10),

location char(15));

# Table Structure:

**QUERY:**

DESC deptIT;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  | **0** |  |
| **location** | **Char(10)** | **NO** |  |  |  |

# deptCSCTable:

create table deptCSC

(stidint(20),

stname char(20),

dept char(10),

gender char(10),

location char(15));

# Table Structure:

**QUERY:**

DESC deptCSC;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |

# Supplier Table:

create table deptCSC

(stidint(20),

stname char(20),

dept char(10),

gender char(10),

location char(15));

# Table structure

**QUERY:**

DESC deptCSC;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |

# SHOW

**SYNTAX :**

**SHOW TABLES;**

**QUERY :**

Mysql>SHOW TABLES;

**OUTPUT:**

|  |
| --- |
| **deptIT** |
| **deptCSC** |
| **deptMBA** |

# CREATE VIEW

**SYNTAX :**

**Create view view\_name as select statement;**

**QUERY:**

Mysql>create viewv1 as select \*fromdeptIT;

Mysql>create viewv2 as select stid, stnamefromdeptCSC;

Mysql>show tables

**OUTPUT:**

|  |
| --- |
| **deptIT** |
| **deptCSC** |
| **deptMBA** |
| **V1** |
| **V2** |

# DROP VIEW

**SYNTAX :**

**drop view view\_name;**

**QUERY:**

Mysql> drop view v1,v2;

Mysql>show tables;

**OUTPUT:**

|  |
| --- |
| **deptIT** |
| **deptCSC** |
| **deptMBA** |

# ALTER TABLE

**SYNTAX :**

**alter table table\_name alter specification**

**[ ,alter\_specification2,……… ];**

1. **ADD**

**Syntax**:

ALTER TABLE table\_name

ADD Col\_namedatatype ()...;

**QUERY:**

Mysql>alter table deptITaddugper char(10);

Mysql>DESCdeptIT;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |
| **ugper** | **char(10)** | **NO** |  |  |  |

#### 2) MODIFY

**Syntax:**

ALTER TABLE table\_name

MODIFY (fieldname datatype()...);

**QUERY:**

Mysql>alter table deptITaddugperint(11);

Mysql>descdeptIT;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |
| **ugper** | **Int(11)** | **NO** |  |  |  |

### RENAME

Rename a table

**Syntax:**

ALTER TABLE table\_name RENAME to new table\_name

**QUERY:**

Mysql>alter table deptMBA RENAME to deptMCA;

Mysql>descdeptMCA;

**OUTPUT:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |

# INSERT MULTIPLE FIELDS :

# SYNTAX:

# Alter table table name add (field\_name1 (data type), diled\_name2 (data\_type),…);

# QUERY:

# Alter table deptmca add (city varchar(20), state varchar(20));

# OUTPUT:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |
| **City** | **Varchar(20)** | **NO** |  |  |  |
| **State** | **Varchar(20)** | **NO** |  |  |  |

# DROP COLUMN

**SYNTAX :**

**Alter table table\_name dropcolumn\_name;**

**QUERY:**

# Alter table deptmca drop state;

# OUTPUT:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **stid** | **int unsigned** | **NO** | **PRI** |  | **auto\_increment** |
| **Stname** | **char(20)** | **NO** |  |  |  |
| **Dept** | **char(10)** | **NO** |  |  |  |
| **Gender** | **Char(10)** | **NO** |  |  |  |
| **location** | **Char(10)** | **NO** |  |  |  |
| **City** | **Varchar(20)** | **NO** |  |  |  |

# TRUNCATE TABLE

**SYNTAX :**

**Truncatetable\_name;**

**QUERY:**

mysql>Truncate deptMCA;

# DROP TABLE

**SYNTAX :**

**Drop table table\_name;**

**QUERY:**

mysql> drop tabledeptMCA;

**QUERY:**

mysql> show tables;

|  |
| --- |
| **deptIT** |
| **deptCSC** |

# DROP DATABASE

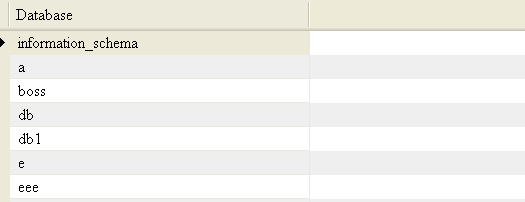
**SYNTAX :**

**Drop database database\_name;**

**QUERY:**

mysql>Drop databaseSriram;

mysql> show databases;



|  |  |
| --- | --- |
| **EX.NO :02** | **DATA MANIPULATION LANGUAGE** |
| **DATE :** |  |

# DATA MANIPULATION LANGUAGE :

The Data Manipulation Language is a language that enables users to access or manipulate data as organised by the appropriate data model.

The commands available in DML are:

* Insert
* Update
* Delete
* Select

# INSERT

**SYNTAX:**

**INSERT into table\_namevalues(value1,value2,value n);**

**QUERY:**

insert into deptCSC values(6101,'kumar', 'csc', 'male', 'cmbt');

insert into deptCSC values(6102,'kaviya', 'csc', 'female', 'choolai');

insert into deptCSC values(6103,'arun', 'csc', 'male', 'perambur');

insert into deptCSC values(6104,'gopi', 'csc', 'male', 'padi');

select \* from deptCSC;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

# UPDATE

**SYNTAX :**

**updatetable\_name set column\_name1=expression1, [ , column\_name2=expression2……]**

**[ wherewhere\_condition]; QUERY :**

Mysql> update deptIT set location='kolathur' wherestid='5102';

Mysql> update deptIT set stname='vanthana' wherestid='5105';

Mysql> select \* from deptIT;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location |
| 5101 | ram | IT | Male | perambur |
| 5102 | sankaran | IT | Male | kolathur |
| 5103 | gopi | IT | Male | avadi |
| 5104 | jaya | IT | Female | perambur |
| 5105 | vanthana | IT | female | thanjavur |

# DELETE

**SYNTAX:**

**delete from table\_name where [ condition ]; QUERY :**

Mysql> delete from deptIT where stid='5105';

Mysql>select \* from deptIT;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location |
| 5101 | ram | IT | Male | perambur |
| 5102 | sankaran | IT | Male | kolathur |
| 5103 | gopi | IT | Male | avadi |
| 5104 | jaya | IT | Female | perambur |

.

# SELECT‘WHERE’ CLAUSE:

# QUERY:

Mysql>select \* from deptIT where gender="Male";

**OUTPUT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location |
| 5101 | ram | IT | Male | perambur |
| 5102 | sankaran | IT | Male | kolathur |
| 5103 | gopi | IT | Male | avadi |

# LIMIT’ CLAUSE:

# QUERY:

Mysql>select \* from deptIT limit 2;

**OUTPUT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location |
| 5101 | ram | IT | Male | perambur |
| 5102 | sankaran | IT | Male | kolathur |

# QUERY:

Mysql>select \* from deptIT where gender="male" limit 2;

**OUTPUT:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location |
| 5101 | ram | IT | Male | perambur |
| 5102 | sankaran | IT | Male | kolathur |

# GROUP BY:

# QUERY:

Select Count(stname),locationFromdeptIT group By location;

**OUTPUT:**

|  |  |
| --- | --- |
| **COUNT(Pid)** | **PCode** |
| **1** | **avadi** |
| **1** | **padi** |
| **2** | **perambur** |
| **1** | **thanjavur** |

# ORDER BY:

**a) ASCENDING:**

select \* from deptIT order bystname;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | Dept | gender | location | ugper |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**b)DESC:**select \* from deptITorder bystnamedesc;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | Dept | gender | location | ugper |
| 5105 | swetha | IT | female | thanjavur | 85 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5101 | ram | IT | Male | perambur | 96 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5103 | gopi | IT | Male | avadi | 53 |

# DISTINCT:

Mysql> select distinct stid fromdeptIT;

**stid**

|  |
| --- |
| **5101** |
| **5102** |
| **5103** |
| **5104** |
| **5105** |

# AND,OR,NOT

* 1. **AND:**

Select \* from deptIT where gender='Male' and ugper>='70';

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | Dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |

# OR:

Select pid,pcode,pname,quantity from product

where pcode=’soap’ORpcode=’juice’;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | Dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

# NOT:

Select \* from deptITwhere not gender='Male';

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | Dept | Gender | location | ugper |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | Female | thanjavur | 85 |

# RANGE TEST:

**USING BETWEEN:**

Select stid ,stname,ugper from deptit where ugper between 80 and 95;

# USING NOT BETWEEN:

Select stid ,stname,ugperFromdeptit

Where ugperNot Between 85 And 90;

|  |  |  |
| --- | --- | --- |
| **stid** | **stname** | **ugper** |
| 5101 | ram | 96 |
| 5103 | gopi | 53 |
| 5104 | jaya | 72 |

|  |  |
| --- | --- |
| **EX.NO :03** | **CONSTRAINTS** |
| **DATE :** |  |

**MY SQL CONSTRAINTS**

**NOT NULL:**

create table deptbba(stidint, stnamevarchar(20) not null, age int(5));

insert into deptbba

values(2001,'gomathi',22),

(2002,'prakash',23),

(2003,'',22);

select \*from deptbba;

|  |  |  |
| --- | --- | --- |
| **stid** | **stname** | **age** |
| 2001 | gomathi | 22 |
| 2002 | prakash | 23 |
| 2003 |  | 22 |

# UNIQUE:

create table college(deptidint, dept\_namevarchar(20) unique);

insert into college values(1,'deptit'), (2,'deptbba'), (3,'deptcsc');

**QUERY:**

select \*from college;

|  |  |
| --- | --- |
| **deptid** | **dept\_name** |
| 1 | deptit |
| 2 | deptbba |
| 3 | deptcsc |

# PRIMARY KEY:

create table deptCSC

(stidint(20) primary key,

stname char(20),

dept char(10),

gender char(10),

location char(15));

insert into deptCSC values(6101,'kumar', 'csc', 'male', 'cmbt'),

(6102,'kaviya', 'csc', 'female', 'choolai'),

(6103,'arun', 'csc', 'male', 'perambur'),

(6104,'gopi', 'csc', 'male', 'padi');

# select \* from deptcsc;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

# FOREIGN KEY:

create table publisher(pidvarchar(4),pnamevarchar(20),address varchar(30),constraint c1 primary key(pid));

create table Bk(ISBN varchar(15),bnamevarchar(50),category varchar(50),price integer(6),year integer(4),pidvarchar(4),constraint p2 check(price between 20 and 200),primary key(ISBN),foreign key(pid) references publisher(pid));

insert into publisher values('1','loyola','anna nagar'), ('2','emerald', 'vadapalani'),('3','sixthsense','perambur'),('4','vikas','mount road');

select \*from publisher;

|  |  |  |
| --- | --- | --- |
| **pid** | **pname** | **address** |
| **1** | **loyola** | **annanagar** |
| **2** | **emerald** | **vadapalani** |
| **3** | **sixthsense** | **perambur** |
| **4** | **vikas** | **mount road** |

# ENUM:

create table stud(id int primary key Auto\_increment, name varchar(20),dept

ENUM(’IT’,’mca’,’csc’));

desc shirts;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Type Null Key Default Extra** | | | | | |
| **id** | **int** | **No** | **PRI** |  | **auto\_increment** |
| **name** | **varchar(2)** | **Yes** |  |  |  |
| **size** | **enum('**’IT’,’mca’,’csc’**)** | **Yes** |  | | |

insert into stud(id,name,dept)values(101,'Ram','IT');

insert into stud(id,name,dept)values(202,'pooja','3');

insert into stud(id,name,dept)values(323,'mohana','mca');

insert into stud(id,name,dept)values(481,'yazhini','mca');

select \* from shirts;

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **size** |
| **1** | **Ram** | **IT** |
| **202** | **pooja** | **Csc** |
| **323** | **mohana** | **Mca** |
| **481** | **Yazhini** | **mca** |

# SET:

Create table studset(sidint not null, primary key ,sports SET(‘running’,’football’,’cricket’,’khokho‘));

**Descmyset\_Book;**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **Id** | **int** | **NO** | **PRI** |  |  |
| **myset\_**  **col** | **set('j','c','VB','OS')** | **YES** |  |  |  |

Insert into studsetvalues (‘101,’running’,khokho”),(‘202’,’khokho’,’football’);

**QUERY:**

Select \* fromstudset;

**OUTPUT:**

|  |  |
| --- | --- |
| **Id** | **sports** |
| **101** | running,khokho |
| **202** | khokho,football |

|  |  |
| --- | --- |
| **EX.NO :04** | **MYSQL OPERATORS** |
| **DATE****:** |  |

**MYSQL OPERATORS:**

We have several types of operators:

* Logical operators
* Comparison operators

**LOGICAL OPERATORS:**

**AND:**

**select \* from deptit where ugper<=60 AND avg>=90;**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**OR:**

**select \* from deptit where ugper<=60 OR location='perambur'**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |

**NOT:**

Select\*from deptit where gender is not male;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**BETWEEN OPERATOR:**

Select\*from deptitwhere ugperbetween 70 and 90;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**EXISTS OPERATOR:**

Select stname from deptit where exists (select stname from deptit where ugper>85);

|  |
| --- |
| **NAME** |
| **Ram** |
| **Sankaran** |

**LIKE**:

select \*from deptitwhere NAME LIKE' R%';

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |

**COMPARISON OPERATORS:**

**>OPERATOR:**

select \*from deptitwhere ugper>90;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |

**<OPERATOR:**

select\* from deptit where ugper<60;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5103 | gopi | IT | Male | avadi | 53 |

**=OPERATOR:**

select \*from deptitwhere ugper=96;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |

**!=OPERATOR:**

select \*from deptit where Salary!=96;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**<>OPERATOR:**

Select\*from deptit where ugper<>90;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5102 | sankaran | IT | Male | padi | 90 |

**>=OPERATOR:**

select \*from deptitwhere ugper>=85;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5105 | swetha | IT | female | thanjavur | 85 |

**<=OPERATOR:**

select\*from deptitwhere ugper<=85;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

|  |  |
| --- | --- |
| **EX.NO :05** | **MYSQL FUNCTIONS** |
| **DATE:** |  |

MySQL built-in functions can be categorized into several groups.

* Mathematical functions
* Aggregate functions
* String functions
* Date and time functions

**MATHEMATICAL FUNCTIONS**

MySQL supports multiple mathematical functions.

**Query:**

mysql>select rand();

|  |
| --- |
| **rand()** |
| **0.8930838293331524** |

The RAND() function returns a random number from the <0,1>interval.

**QUERY:**

mysql>select abs(-3),sin(0.5);

|  |  |
| --- | --- |
| **abs(-3)** | **sin(0.5)** |
| **3** | **0.479425538604203** |

The ABS() function return the absolute value of a number.And the SIN()

function computes the sine of an argument.

**QUERY:**

mysql>select ceil(12.256),floor(12.256);

|  |  |
| --- | --- |
| **Ceil(12.256)** | **Floor(12.256)** |
| **13** | 12 |

The CEIL() function rounds the value to the smallest following integer.

The FLOOR() function rounds the value to the largest previous integer.

**QUERY:**

mysql>select pow(3,3),sqrt(9);

|  |  |
| --- | --- |
| **Pow(3,3)** | **Sqrt(9)** |
| **27** | **3** |

The power and the square root functions.

**QUERY:**

mysql>select degrees(3\*pi());

|  |
| --- |
| **degrees (3\*pi ())** |
| **540** |

The DEGREES() function computes degree from radians.

**AGGREGATE FUNCTIONS**

Aggregate functions operate on set of values.

**QUERY:**

mysql>select \* from deptit;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

We have the Product table.

**QUERY:**

select min(ugper),max(ugper),avg(ugper) from deptit;

|  |  |  |
| --- | --- | --- |
| **min(ugper)** | **max(ugper)** | **avg(ugper)** |
| **53** | **96** | **86** |

we use the MIN(),MAX() and AVG() aggregate functions to compute

the minimal cost,maximal cost and the average cost of product in the table.

**QUERY:**

select sum(ugper),count(stname),std(ugper),variance(ugper)fromdeptit;

|  |  |  |  |
| --- | --- | --- | --- |
| **sum(price)** | **Count(PCode)** | **std(price)** | **variance(price)** |
| **492** | **5** | **15.3080** | **234.3333** |

We use the SUM() function to get the sum of all values in the product

column.

We count the number of products in the table with the COUNT()

function.

Finally,we get the standard deviation and variance using the STD()

and VARIANCE() functions.

**STRING FUNCTIONS**

In this group we have various strings related functions.

**QUERY:**

select \* fromdeptit;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| stid | stname | dept | gender | location | ugper |
| 5101 | ram | IT | Male | perambur | 96 |
| 5102 | sankaran | IT | Male | padi | 90 |
| 5103 | gopi | IT | Male | avadi | 53 |
| 5104 | jaya | IT | Female | perambur | 72 |
| 5105 | swetha | IT | female | thanjavur | 85 |

We have the Employee table.

**QUERY:**

mysql>select

LENGTH (stname),UPPER(stname),LOWER(stname)from

Depitwhere stid="5101";

|  |  |  |
| --- | --- | --- |
| **LENGTH(**stname**)** | **UPPER(**stname**)** | **LOWER(**stname**)** |
| **3** | **RAM** | **ram** |

The LENGTH() function returns the length of the string.

The UPPER()function converts characters into upper-case letters.

The LOWER() functionconverts characters into lower-case letters.

**QUERY:**

select LPAD(RPAD(EmpName,8,"\*"),10,"\*")fromdeptit;

|  |
| --- |
| **LPAD(RPAD(EmpName,8,"\*"),10,"\*")** |
| **\*\*ram\*\*\*\*\*** |
| **\*\*gopi\*\*\*\*\*** |
| **\*\*jaya\*\*\*\*** |
| **\*\*swetha\*\*\*** |

We use the LPAD() and RPAD() functions to append and prepend

characters to a specified string.

The “Anu” string has 3characters.

TheRPAD() function append 5 '\*' characters to the string which will be now 10 characters long.

**QUERY:**

select REVERSE(stname)from depitwhere stid="5101";

|  |
| --- |
| **REVERSE(stname)** |
| **mar** |

The REVERSE() function reverses the character in a string.

**QUERY:**

select

LEFT(stname,3),RIGHT(stname,3),SUBSTRING(stname,2,4)from

Deptitwhere stid ="5105”;

|  |  |  |
| --- | --- | --- |
| **LEFT(EmpName,3)** | **RIGHT(EmpName,3)** | **SUBSTRING(EmpName,2,4)** |
| **swe** | **tha** | **weth** |

The LEFT() function returns 3 leftmost characters,

The RIGHT() function returns 3 characters from the right.

The SUBSTRING() function returns.

three characters from the third position of the string.

**QUERY:**

select CONCAT(stname,'-',gender)fromdeptit;

|  |
| --- |
| **CONCAT(stname,'-',gender)** |
| **Ram-male** |
| **Sankaran-male** |
| **Gopi-male** |
| **Swetha-female** |
| **Jaya-female** |

The CONCAT() function concatenates two strings.

**DATE AND TIME FUNCTIONS**

In this group we have various date and time functions.

**QUERY:**

mysql> SELECT Dayname("2022-01-05"),year("2022-01-05"),monthname("2022-01-05");

|  |  |  |
| --- | --- | --- |
| **Dayname("2022-01-05")** | **year("2022-01-05")** | **monthname("2022-01-05")** |
| **wednesday** | **2022** | **January** |

In MySQL,date is written in the format YYYY\_MM\_DD.Year is followed by

month and day.

They can be separated by slash or by hyphen.

MySQL also supports a shortened date format,withoutseperators.

Time is written in a standard form,HH\_MM\_SS.hours followed by minute and seconds.

The DAYNAME() function returns the day name of a date.

The MONTHNAME() function returns a month name of a date.

The YEAR() function returns a year name of a date.

**QUERY:**

select now();

|  |
| --- |
| **NOW()** |
| **2022-02-04T13:14:15.000Z** |

The NOW() function return the current date and time.

**QUERY:**

selectcurtime(),curdate();

|  |  |
| --- | --- |
| **curtime()** | **curdate()** |
| **13:15:53** | **2022-02-04** |

The CURTIME() returns the current time and the CURDATE() returns

the current date.

**QUERY:**

selectdatediff("2020-01-05","2020-01-25");

|  |
| --- |
| **datediff("2020-01-05","2020-01-25")** |
| **-20** |

With the DATEDIFF() we get the number of days between two dates.

**QUERY:**

mysql>select DATE\_FORMAT("2017-06-15", "%d %M %Y");

|  |
| --- |
| **DATE\_FORMAT("2017-06-15", "%d %M %Y")** |
| **15 June 2017** |

To display date in a different format,we use the DATE\_FORMAT().

**QUERY:**

mysql>select DATE\_ADD("2017-06-15",INTERVAL 10

DAY),DATE\_SUB("2017-06-15", INTERVAL 10 DAY);

|  |  |
| --- | --- |
| **DATE\_ADD("2017-06-15",INTERVAL 10 DAY)** | **DATE\_SUB("2017-06-15", INTERVAL 10 DAY)** |
| **2017-06-25** | **2017-06-05** |

We can use DATE\_ADD() to add time interval to a date and SUBDATE()

to subtract time intervals from a date.

|  |  |
| --- | --- |
| **EX.NO :06** | **JOINS** |
| **DATE :** |

# JOINS

The JOIN keyword is used in an SQL statement to query data from two or more tables, based on a relationship between certain columns in these tables.

The different types of joins available are:

* Innerjoin
* Left Outerjoin
* Right Outerjoin

# INNER JOIN:

It return rows where there is at least one match in both the tables. The keyword Inner Join is same as the keyword Join.

# SYNTAX:

**Select \* from table1 INNER JOIN table2 ON table\_name1.column\_name=table\_name2.column\_name;**

**QUERY:**

Select \* from Emp inner join Dep on Emp.Dep\_no=Dep.dep\_no;

**Output:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emp\_no** | **Emp\_name** | **Emp\_salary** | **Dep\_no** | **Dep\_name** | **Location** |
| **E101** | **vanthana** | **20000** | **101** | **accounting** | **karnodai** |
| **E152** | **Ram** | **25000** | **152** | **research** | **america** |

# OUTER JOIN:

The SQL OUTER JOIN returns all rows from both the participating Table which satisfies the join condition along with rows which do not satisfy the join condition.

The subtype of SQL OUTER JOIN

* + LEFT OUTER JOIN or LEFTJOIN
  + RIGHT OUTER JOIN or RIGHTJOIN

# LEFT OUTER JOIN:

It returns all the rows from the left join, even if there is no match in the right table.

# SYNTAX:

**Select \* from table1 LEFT JOIN table2 ON table\_name1.column\_name=table\_name2.column\_name;**

**QUERY:**

Select \* from Emp left join Dep on Emp.Dep\_no=Dep.dep\_no;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Emp\_no** | **Emp\_name** | **Emp\_salary** | **dep\_no** | **Dep\_name** | **Location** |
| **E101** | **vanthana** | **20000** | **101** | **Web desing** | **karnodai** |
| **E152** | **ram** | **25000** | **152** | **Sqlmangement** | **america** |
| **E133** | **akshya** | **40000** | **281** |  |  |
| **E152** | **gyathiri** | **35000** | **282** |  |  |

# RIGHT OUTER JOIN:

It returns all the rows from the right join, even if there is no match in the left table.

# SYNTAX:

**Select \* from table1 RIGHT JOIN table2 ON table\_name1.column\_name=table\_name2.column\_name;**

**QUERY:**

Select \* from Emp right join Dep on Emp.Dep\_no=Dep.dep\_no;

Output

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Emp\_no** | **Emp\_name** | **Emp\_salary** | **dep\_no** | **Dep\_no** | **Dep\_name** | **Location** |
| **E101** | **vanthana** | **20000** | **101** | **C1** | **Web desing** | **karnodai** |
| **E152** | **ram** | **25000** | **152** | **C3** | **Sqlmangement** | **america** |
|  |  |  |  | **c6** | **operation** | **mumbai** |
|  |  |  |  | **c4** | **accounting** | **mumbai** |
|  |  |  |  | **c5** | **testing** | **newdelhi** |

|  |  |
| --- | --- |
| **EX.NO:07** | **SUB QUERIES** |
| **DATE :** |

# SUB QUERIES:

The nesting of queries one within the other is termed as sub query.

The query containing sub query is called as parent query and the query Which is written inside parent query is called as sub query.

The different types of sub queries available are:

* Scalar subquery
* Table subquery

# SCALAR SUBQUERY

In scalar sub query, the inner query returns a single value.

# SYNTAX:

**Select [column name] from table name1 [table name 2] where column name operator (select column name from table 1, table2 where condition);**

**QUERY:**

# Select Stid, stname, tot, grade from student where stid= (select stid from teacher

# TABLE SUBQUERY

**USING IN:**

In table sub query, the inner query returns a result table of zero or more rows. The keyword in, any and all are under it.

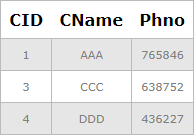
# IN:

**SYNTAX:**

**Select \* from table name1 where column name IN (select column name from table name2);**

**QUERY:**

SELECT CID, CName, Phno from Cust where PID IN(select pid from Pdt where PName IN('Soap','Chocolate'));



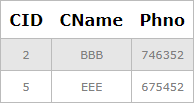
# NOTIN:

**SYNTAX:**

**Select \* from table name1 where column name NOT IN (select column name from table name2);**

**QUERY:**

SELECT CID,CName,Phno from Cust where PID NOT IN(select pid from Pdt where PName IN('Soap','Chocolate'));



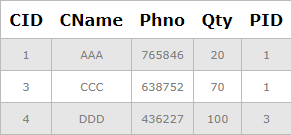
# CORELATED SUBQUERY

**USING ANY:**

**SYNTAX:**

**Select \* from table name1 where column name=ANY (select column name from table name2 where condition);**

**QUERY:**

Select \* from Cust where PID=ANY (select PID from Pdt where PName IN ('Soap','Chocolate'));

# NESTED SUBQUERY

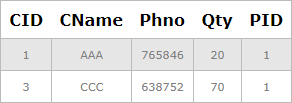
**USING ALL:**

**SYNTAX:**

**Select \* from table name1 where column name(=,>,<) (select column name from table name where[condition]);**

**QUERY:**

select \* from Cust where PID=ALL(select PID from Pdt where PName='soap');



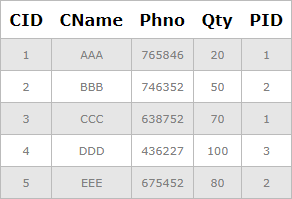
# USING EXISTS:

**SYNTAX:**

**Select \* from table name1 where column name=EXISTS(select column name from table name2 where condition);**

**QUERY:**

SELECT \* from Cust where exists(select PID from Pdt where PName='Soap');



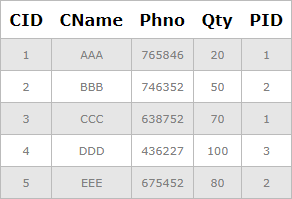
# NOT EXISTS:

**SYNTAX:**

**Select \* from table name1 WHERE NOT EXISTS (select \* from table name 2 where [condition]);**

**QUERY:**

Select \* from Cust where NOT EXISTS(select PID from Pdt where PName='Soap');

Select \* from Cust where NOT EXISTS(select PID from Pdt where PName='Powder');