DBMS – Database management system

We can store and manage the large database.

SQL – Structured query language

We can create, update and delete tables using sql queries.

DDL – data definition language – Create alter, drop, truncate

DML – Data manipulation language – Insert, update, delete

DQL – Data query language – Select

Create database <db\_name>

Create database 12Dec-DB

Create table candidates

(cand\_id int(10),

cand\_name varchar(100),

cand\_address varchar(200)

)

Alter table candidates change c\_id cand\_id int(10)

Create table candidates1

Insert into candidates (cand\_id, cand\_name, cand\_address, cand\_mobile)

values (1, "Sanjay Tamboli", "A-903, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", “8141234784);

insert into candidates (cand\_id, cand\_name, cand\_address, cand\_mobile)

values (2, "Deepak Chorsia", "A-904, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad",”8141234785”);

insert into candidates (cand\_id, cand\_name, cand\_address, cand\_mobile)

values (3, "Moxit Shah", "A-904, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", “8141234786”);

insert into candidates (cand\_id, cand\_name, cand\_address, cand\_mobile)

values (4, "Avinash Patel", "A-905, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", “8141234787”)

Create database school,

Create table student (student\_id, name, subject, address, age)

create DATABASE school

create table student

(student\_id int(5),

student\_name varchar(100),

student\_address varchar(300),

student\_subject varchar(300),

student\_age int(2))

alter table student add student\_mobile varchar(10)

insert into student (student\_id, student\_name, student\_address, student\_subject, student\_age, student\_mobile)

values (2, "Dipak Patel", "A-904, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", "Physics, chemistry, Biology, Maths", 18, "8141234785");

insert into student (student\_id, student\_name, student\_address, student\_subject, student\_age, student\_mobile)

values (3, "Dipak Shah", "A-901, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", "Physics, chemistry, Biology, Maths", 16, "8141234786");

insert into student (student\_id, student\_name, student\_address, student\_subject, student\_age, student\_mobile)

values (3, "Dipak Sharma", "A-902, Shivalay Height, stadium road, Motera, Sabarmati, Ahmedabad", "Physics, chemistry, Biology, Maths", 18, "8141234787")

create table employee with id, age, department, salary.

Create table employee

(emp\_id int(3), emp\_age int(2), emp\_dept varchar(30), emp\_salary int(4))

Insert into employee (emp\_id, emp\_age, emp\_dept, emp\_salary) Values (1, 18, "Account", 8000);

Insert into employee (emp\_id, emp\_age, emp\_dept, emp\_salary) Values (2, 19, "Admin", 9000);

Insert into employee (emp\_id, emp\_age, emp\_dept, emp\_salary) Values (3, 20, "HR", 7000);

Insert into employee (emp\_id, emp\_age, emp\_dept, emp\_salary) Values (4, 21, "Production", 6000)

Delete from employee where emp\_id = 4

Truncate table employee

create table category

(

cate\_id int(10) primary key AUTO\_INCREMENT,

cate\_name varchar(200)

)

Create table subcategory

(

Subcate\_id int(10) primary key auto\_increment,

Subcate\_name varchar(100),

cate\_id\_fk int(10),

FOREIGN key (cate\_id\_fk) references category (cate\_id)

)

create table product

(

prod\_id int(10) primary key AUTO\_INCREMENT,

prod\_name varchar(200),

prod\_price varchar(200),

prod\_disc\_price varchar(200),

cate\_id\_fk int(10),

subcate\_id\_fk int(10),

foreign key (cate\_id\_fk) references category (cate\_id),

foreign key (Subcate\_id\_fk) references subcat(gory (Subcate\_id)

)

Create table teacher

(

Teacher\_id int(10) primary key auto\_increment,

Teacher\_name varchar(100)

)

Create table student

(

Student\_id int(10) primary key auto\_increment,

Student\_name varchar(100),

Teacher\_id\_fk int(10),

Foreign key (Teacher\_id\_fk) references teacher (teacher\_id)

)

Select \* from product where prd\_name = “Shirts”

Select \* from product where prd\_price = 2500

Select \* from product where prd\_price > 1000

Select \* from product where prd\_price < 2000

Select \* from product where prd\_price between 1000 and 2000

Select \* from product order by prd\_name asc

Select \* from product order by prd\_price asc

Select \* from product order by prd\_price desc

Select min(prd\_price) from product

Select min(prd\_price) from product

Types of Joins

Inner

Outer

Left

Right

Full

Select category.cate\_name, subcategory.subcate\_name

From category join subcategory on category.cate\_id = subcategory.cate\_id\_fk

Select category.cate\_name, subcategory.subcate\_name, product.prd\_name, product.prd\_price,

Product.prd\_desc\_

Join subcategory on category.cate\_id = subcategory.cate\_id\_fk

Join product on category.cate\_id = product.cate\_id\_fk

INSERT INTO `student` (`Student\_id`, `Student\_name`, `Teacher\_id\_fk`) VALUES (NULL, 'Suresh Shah', '1'), (NULL, 'Mahesh Patel', '2');

Select student\_name, teacher\_name

From student join teacher on teacher\_id = teacher\_id\_fk

create table incentive (

employee\_ref\_id int(3),

incentive\_date date,

incentive\_amount int(5)

)

DELIMITER $$

create PROCEDURE PutData()

BEGIN

insert into candidates (cand\_id, cand\_mobile, cand\_address, cand\_name)

values (1, "9825434784", "Motera - Ahd", "Sanjay");

END

Call PutData;

DELIMITER $$

create PROCEDURE PutData\_1( id int(3), mobile varchar(10), address varchar(300), name varchar(100))

BEGIN

insert into candidates (cand\_id, cand\_mobile, cand\_address, cand\_name)

values (id, mobile, address, name);

END

Call PutData\_1 (2, "9825434785", "Navrangpura - Ahd", "Jayesh");

Call PutData\_1 (3, "9825434786", "Naranpura- Ahd", "Ramesh");

Alter table student add marks int(3)

DELIMITER $$

Create procedure StudentData()

Begin

Insert into student (student\_id, student\_name, teacher\_id\_fk, marks)

Values (10, "Mahesh", 1, 50);

End

create table TblProduct

(

prd\_id int(3) PRIMARY key AUTO\_INCREMENT,

prd\_name varchar(25),

Recommended\_price varchar(25),

category varchar(25)

)

create table TblCustomer

(

Customer\_id int(3) PRIMARY key AUTO\_INCREMENT,

First\_name varchar(25),

Last\_name varchar(25),

City varchar(25),

State varchar(25),

Zip varchar(25)

)

create table TblSales

(

Sales\_id int(4) PRIMARY key AUTO\_INCREMENT,

Prod\_id int(3),

Cust\_id int(3),

Salesprice varchar(25),

SalesDate Date,

FOREIGN key (Prod\_id) REFERENCES tblproduct (prd\_id),

FOREIGN key (Cust\_id) REFERENCES tblcustomer (customer\_id)

)

create table test1

(

ID int(10),

name varchar(200),

date\_time timestamp,

action\_performed varchar(200)

)

DELIMITER $$

create trigger insert\_trigger after insert on candidates for each ROW

BEGIN

insert into test1 (id, name, action\_performed)

values (new.cand\_id, new.cand\_name, "Record Inserted");

End

DELIMITER $$

create trigger update\_trigger after update on candidates for each ROW

BEGIN

insert into test1 (id, name, action\_performed)

values (new.cand\_id, new.cand\_name, "Record Updated");

End

DELIMITER $$

create trigger delete\_trigger after delete on candidates for each ROW

BEGIN

insert into test1 (id, name, action\_performed)

values (old.cand\_id, old.cand\_name, "Record Deleted");

End

DELIMITER $$

create trigger insert\_trigger1 after insert on category for each ROW

BEGIN

insert into subcategory (cate\_id\_fk, subcate\_id, subcate\_name)

values (new.cate\_id, new.cate\_id, "Sub\_"+new.cate\_name);

End

DBMS

DBMS stands for data base management system

Data + Management system

**DBMS** is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective

For example, university database organizes the data about students, faculty, and admin staff etc. Which **helps in efficient retrieval, insertion and deletion of data from it.**

Request ----🡪 DB engine ----------🡪 Data

Two most common types of DBMS are:

* Local
* Server

Popular DBMS software

* MySQL
* Microsoft access
* Oracle
* PostgreSQL
* SQLite
* Mongo DB
* IBM
* DB2

**Need of DBMS**

Storage and retrieval of large amount of data

Fast retrieval of data

**Purpose of DBMS**

Manage the data

Create database, tables and views

Insert, update, delete, select the data

**RDBMS**

It stores

Hierarchical database – Tree like structure with parent child relationship….

Network database – Multiple relations among the data

**OODBMS –** Object oriented database mgmt. system

**E-R Model** – Entity relationship Model

**Home Work** – 20-Mar-2023.

Library management system – ERD

Books

Members –ID, Name, Email ID, Phone No.

College mgmt. system – ERD – Department, classes, HoDs, Faculty, students

Study ACID concept of RDBMS

Difference between primary key and unique key.

Primary key one per table.

Unique key multiple per table.

Primary key does not allow null value.

Unique key allows null value.

use 12dec\_db;

create table user1

(

user\_id int primary key AUTO\_INCREMENT,

user\_name varchar(30),

user\_email\_id varchar(30),

user\_mobile\_no varchar(10)

)

Create following table.

Category

Sub category

Product

* Composite primary key

create table sanjay

(

order\_id int,

prd\_id varchar(20),

prd\_name varchar(20),

primary key (order\_id, prd\_id)

)

update customer

set working\_area = 'New Area', Phone\_no = '9825098250'

where cust\_city = ‘City3’

delete from customer where phone\_no = '23423442'

Alter table customer drop phone\_no

Select cust\_country group by cust\_country

Select cust\_country as total\_countries from customer

Select sum(rollnumber) from student

Select min(rollnumber) from student

Select max(rollnumber) from student

Select avg(rollnumber) from student

Select name from student where rollno = (select max(rollno) from student)

Select name from student order by name ASC (or desc)

Between, like, %%, \_,

create table student

(

student varchar(30),

percentage int(3)

)

INSERT INTO `student`(`student`, `percentage`) VALUES ('Isha Patel','98');

INSERT INTO `student`(`student`, `percentage`) VALUES ('Harsh Das','94');

INSERT INTO `student`(`student`, `percentage`) VALUES ('Rachit Shah','93');

INSERT INTO `student`(`student`, `percentage`) VALUES ('Sumedha','98');

INSERT INTO `student`(`student`, `percentage`) VALUES ('Rahat Ali','98')

Create table employee

(

EmployeeId int(3),

Name varchar(30),

Gender Char(1),

Salary int(5),

Department varchar(20),

Experience varchar(20)

)

INSERT INTO `employee`(`EmployeeId`, `Name`, `Gender`, `Salary`, `Department`, `Experience`)

VALUES (1, 'Rachit', 'M', 50000, 'Engineering', '6 Years' ) ;

INSERT INTO `employee`(`EmployeeId`, `Name`, `Gender`, `Salary`, `Department`, `Experience`)

VALUES (2, 'Shobhit', 'M', 37000, ‘HR’, '3 Years' ) ;

INSERT INTO `employee`(`EmployeeId`, `Name`, `Gender`, `Salary`, `Department`, `Experience`)

VALUES (3, ‘Isha’, 'F', 58000, 'Engineering', '6 Years' )

Alter table employee add PRIMARY KEY (EmployeeId), change EmployeeId EmployeeId int AUTO\_INCREMENT

Alter table student add EmployeeID int(3),

add FOREIGN key (EmployeeID) REFERENCES employee(EmployeeID)

Select Department, sum(salary) as Salary group by Department

Select Department, sum(salary) group by Department having sum(Salary) > 50000

Select \* from product limit 3 /\* Top 3\*/

Select \* from product limit 3 offset 2 /\* skip first 2 records \*/

Select \* from product order by product\_id limit 3 /\* Last 3 record \*/

Select user\_id from users union select union all product\_id from products

Use 12dec\_db;

create table triggerTbl

(

username varchar(20),

action\_performed varchar(20),

date\_time timestamp

)

DELIMITER $$

create trigger insert\_trigger2 after insert on user1 for each ROW

BEGIN

insert into triggerTbl ( username, action\_performed ) values (new.user\_name, 'Record Inserted');

END

DELIMITER $$

Create PROCEDURE Add\_user1(name varchar(30), email\_id varchar(30), mobile\_no varchar(10))

BEGIN

insert into user1 (user\_name, user\_email\_id, user\_mobile\_no) values (name, email\_id, mobile\_no);

END

Call add\_user1 (‘Moxit’, ‘moxit@gmail.com’, ‘1234567890’);

Call add\_user1 (‘Raj’, ‘Raj@gmail.com’, ‘1234567891’);

Call add\_user1 (‘Harshad’, ‘Harshad @gmail.com’, ‘1234567891’)

DELIMITER $$

Create procedure TestCursor()

BEGIN

declare arjun int;

declare name varchar(30);

declare email\_id varchar(30);

declare mobile\_no varchar(10);

declare t cursor for select \* from user1;

declare CONTINUE HANDLER set arjun = 1;

arjun = 0;

open t;

label:loop

fetch t into name, email\_id, mobile\_no;

insert into user(user\_name, user\_email\_id, user\_mobile\_no) values (name, email\_id, mobile\_no);

if arjun = 1

leave label;

end if;

end loop;

close t;

End

Write trigger on update and delete.