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