Expt No 1

ER diagram and Relational Schema

AIM:Design a database schema for an application with ER diagram from a problem description

Problem Description:

The COMPANY database keeps track of a company's employees, departments, and projects. Suppose that after the requirements collection and analysis phase, the database designers provide the following description of the miniworld—the part of the company that will be represented in the database.

- The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.
- A department controls a number of projects, each of which has a unique name, a unique number, and a single location.
- We store each employee's name, Social Security number,address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).
- We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

1. Draw ER Diagram with cardinality ratio and participation constraints.
2. List Entities
3. List Relationship Types
4. Synthesize into Relational Schema.

Expt No 2

APPLICATION OF DDL COMMANDS USING UI AND SQL

AIM : Creation, modification, configuration, and deletion of databases using UI and SQL Commands

QUERY

Create database students and execute various commands on it.

create database students;
show databases;
use students;

Create a table students with the fields student id, name, email and phone number.

create table Student(stud_id INT AUTO_INCREMENT PRIMARY KEY,stud_fname VARCHAR(20),stud_lname VARCHAR(20),stud_email VARCHAR(20),stud_ph VARCHAR(10));

Create a table subject to store the list of subjects.

create table Subject(sub_id INT AUTO_INCREMENT PRIMARY KEY,sub_name VARCHAR(20));

Create a table marks to store marks of students for various subjects.

create table Marks(sub_id INT,stud_id INT,marks INT, PRIMARY KEY(sub_id,stud_id));
show tables;

Display the details of the tables created.

desc Student;
desc Subject;
desc Marks;

Alter the tables to include foreign keys.

alter table marks ADD FOREIGN KEY (stud_id) REFERENCES student(stud_id);
alter table marks ADD FOREIGN KEY (sub_id) REFERENCES subject(sub_id);

Alter the tables to initialize auto increment value.

alter table Student AUTO_INCREMENT=100;
alter table Subject AUTO_INCREMENT=200;

Alter the tables to add new column.

alter table Student add address char(50);
describe Student;

Alter the tables to modify datatype of a column.

alter table Student modify column address char(50);
describe Student;

Alter the tables to rename column.

alter table Student rename column address to stud_address;
describe Student;

Alter the tables to delete column.

alter table Student drop column stud_address;
describe Student;

Truncate command

To test truncate command, we have to populate table.

insert into Student (stud_fname, stud_lname, stud_email, stud_ph) values ('Alice', 'Tom', ' alice@alice.com', '9889975555');

insert into Student (stud_fname, stud_lname, stud_email, stud_ph) values ('Bob', 'John', ' bob@bob.com', '9889975556');

*select * from Student;*

Test Truncate command

SET FOREIGN_KEY_CHECKS=0; # Used to disable referential integrity
truncate table Student;
*select * from Student; # Observe entire data is cleared from table.*

Drop the tables created.

drop table Student;
drop table Subject;
drop table Marks;
drop database students;

RESULT: Successfully executed the queries using MySQL Workbench.