**The ICFAI Foundation of Higher Education**

(Declared as Deemed to be University u/s 3 of UGC Act 1956)

**Faculty of Science and Technology, Hyderabad**

CS 314 - DATABASE MANAGEMENT SYSTEMS LAB MANUAL

for

III Year CSE

Department of Computer Science and Engineering

List of Experiments

|  |  |
| --- | --- |
| S. No. | Name of the Experiment |
| 1 | Introduction to SQL & MySQL and Database Design |
| 2 | Implementation of Data Definition Language Commands.   * CREATE * ALTER * DROP * TRUNCATE * RENAME |
| 3 | Implementation of Data Manipulation Language commands   * INSERT * UPDATE * DELETE |
| 4 | Implementation of Constraints   * PRIMARY KEY * FOREIGN KEY * UNIQUE * NULL * NOT NULL * DEFAULT * CHECK |
| 5 | Implementation of different types of function   * NUMBER FUNCTION * AGGREGATE FUNCTION * STRING FUNCTION * DATE FUNCTION |
| 6 | Implementation of different types of operators .   * ARITHMETIC OPERATORS * LOGICAL OPERATORS * COMPARISON OPERATORS * SPECIAL OPERATORS * SET OPERATORS |
| 7 | Implementation of different types of Joins.   * INNER JOIN * OUTER JOIN * NATURAL JOIN * SELF JOIN |
| 8 | Implementation of GROUP BY, HAVING, ORDER BY clause(s). |
| 9 | Implementation of Views and Indexes. |
| 10 | Implementation of Database Connectivity. |

**Exp : 1**

**Experiment Name ;** Introduction to SQL & MySQL and Database Design

**SQL (Structured Query Language):**

Structured Query Language is a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon Relational Algebra. Its scope includes data query and update, schema creation and modification, and data access control.

**SQL language is sub-divided into several language elements, including:**

* Clauses, which are in some cases optional, constituent components of statements and queries.
* Expressions, which can produce either scalar values or tables consisting of columns and rows of data.
* Queries which retrieve data based on specific criteria.
* Statements which may have a persistent effect on schemas and data, or which may control transactions, program flow, connections, sessions, or diagnostics.
* SQL statements also include the semicolon (";") statement terminator. Though not required on every platform, it is defined as a standard part of the SQL grammar.
* Insignificant white space is generally ignored in SQL statements and queries, making it
* easier to format SQL code for readability.

There are five types of SQL statements. They are:

1. DATA DEFINITION LANGUAGE (DDL)

2. DATA MANIPULATION LANGUAGE (DML)

3. DATA RETRIEVAL LANGUAGE (DRL)

4. TRANSATIONAL CONTROL LANGUAGE (TCL)

5. DATA CONTROL LANGUAGE (DCL)

**MySQL**

MySQL is the widely used open source database. MySQL is the backend database of most of the websites. As a Free Software(Free as in freedom), MySQL can be downloaded and used by the developer for free.MySQL is robust and it provides excellent performance due to usage of MyISAM. MySQL occupies very less disk space.MySQL can be easily installed in all major operating systems like Microsoft Windows, Linux, UNIX. MySQL is best suited for small and medium applications.

Before creating any tables, MySQL requires you to create a database by executing the CREATE DATABASE command. To create a database called SaleCo you would type the following:

**mysql> CREATE DATABASE SaleCo;**

Notice that you need a semi-colon to end the command.

To view the different databases created in that schema use the show database command.

**mysql> SHOW databases;**

A specific database has to be chosen to work upon. Databases objects created under one database will not be available in the other. To choose a particular database use the below command.

**mysql> USE SaleCo;**

In order to view the tables created under a specifc databases use the show table command.

**mysql> SHOW TABLES;**

**Database Design :**

A database model is a type of data model that determines the logical structure of adatabase and fundamentally determines in which manner data can be stored, organized and manipulated. The most popular example of a database model is the relational model, which uses a table-based format.

Logical database design is the process of deciding how to arrange the attributes of the entities in a given business environment into database structures, such as the tables of a relational database. The goal of logical database design is to create well structured tables that properly reflect the company's business environment. The tables will be able to store data about the company's entities in a non-redundant manner and foreign keys will be placed in the tables so that all the relationships among the entities will be supported.

An entity-relationship model (ERM) is a theoretical and conceptual way of showing data relationships in software development. ERM is a database modeling technique that generates an abstract diagram or visual representation of a system's data that can be helpful in designing a relational database.

**Example**

Consider the Insurance Database given below. The primary keys are underlined and the datatypes are specified.

**PERSON** (driver\_id: string, name : string, address : string)

**CAR** ( regno: string, model : string, year: int)

**ACCIDENT** ( report-number :int, acc\_date :date, locations: string)

**OWNS**(driver\_id :string, reg\_no: string)

**PARTICIPATED**(driver\_id : string, regno :string, report\_number: int , damage\_amt :int)

*Sample ER diagram*

CAR

OWNS

PERSON

makes

ACCIDENT

**Experiment:1 ER DIAGRAM Date: 12.8.2021**

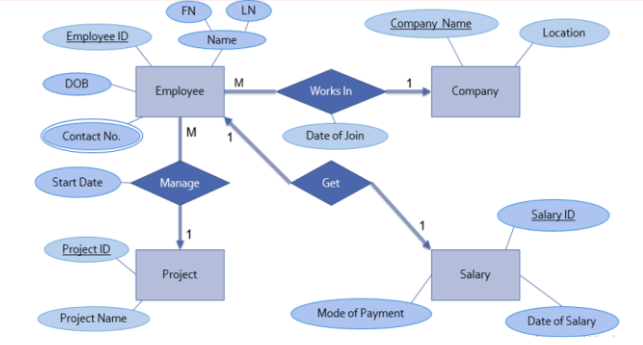
The Prescriptions-R-X chain of pharmacies has offered to give you a Free lifetime supply of medicines if you design its database. Given the rising cost of health care, you agree.

Here's the information that you gather:

* Patients are identified by an SSN, and their names, addresses, and ages must be recorded.
* Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.
* Each pharmaceutical company is identified by name and has a phone number. For each drug, the trade name and formula must be recorded.
* Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
* Each pharmacy has a name, address, and phone number.
* Every patient has a primary physician. Every doctor has at least one patient.
* Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
* Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors.
* Each prescription has a date and a quantity associated with it. You can assume that if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
* Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.
* Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.
* Draw an ER diagram that captures the above information. Identify any constraints that are not captured by the ER diagram.

**Experiment:2 ER Model to Relational Model Date: 17.8.2021**

a. Translate the ER model given below to relational model.



b. Translate the ER model you have drawn in Experiment1 to a corresponding relational model. Keep all your assumptions and make the translation.

**Experiment : 3 DDL COMMANDS**

**The Data Definition Language (DDL)** is used to create and destroy databases and database objects. It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in database..

Let's take a look at the structure and usage of four basic DDL commands:

1. CREATE

2. ALTER

3. DROP

4. RENAME

5. TRUNCATE

**1. CREATE:**

**(a)CREATE TABLE:** This is used to create a new relation (table)

***Syntax:***

CREATE TABLE <relation\_name/table\_name > (field\_1 data\_type(size),field\_2 data\_type(size), .. . );

***Example:***

SQL> CREATE TABLE Student (sno int, sname CHAR (10), class CHAR (5));

**2. ALTER:**

**(a)ALTER TABLE ...ADD...:**

This is used to add some extra fields into existing relation.

***Syntax:***

ALTER TABLE relation\_name ADD (new field\_1 data\_type(size), new field\_2

data\_type(size),..);

***Example:***

SQL>ALTER TABLE std ADD (Address CHAR(10));

**(b)ALTER TABLE...MODIFY...:**

This is used to change the width as well as data type of fields of existing relations.

***Syntax:***

ALTER TABLE relation\_name MODIFY (field\_1 newdata\_type(Size), field\_2

newdata\_type(Size),....field\_newdata\_type(Size));

***Example:***

SQL>ALTER TABLE student MODIFY(sname VARCHAR(10),class

VARCHAR(5));

**c) ALTER TABLE..DROP...:**

This is used to remove any field of existing relations.

***Syntax:***

ALTER TABLE relation\_name DROP COLUMN (field\_name);

***Example:***

SQL>ALTER TABLE student DROP column (sname);

**d)ALTER TABLE..RENAME...:**

This is used to change the name of fields in existing relations.

***Syntax:***

ALTER TABLE relation\_name RENAME COLUMN (OLD field\_name) to (NEW field\_name);

***Example:***

SQL>ALTER TABLE student RENAME COLUMN sname to stu\_name;

**3. DROP TABLE:**

This is used to delete the structure of a relation. It permanently deletes the records in the table.

***Syntax:***

DROP TABLE relation\_name;

***Example:***

SQL>DROP TABLE std;

**4. RENAME:**

It is used to modify the name of the existing database object.

***Syntax:***

RENAME TABLE old\_relation\_name TO new\_relation\_name;

***Example:***

SQL>RENAME TABLE std TO std1;

**5.TRUNCATE**:

It is used to remove all the records from the relations.

***Syntax :***

TRUNCATE TABLE relation\_name;

***Example:***

mysql> TRUNCATE TABLE student;

***LAB Exercise:***

**1.** Create a table EMPLOYEE with following schema:

***(S.No, Emp\_no, E\_name, E\_address, E\_ph\_no, Dept\_no, Dept\_name,Job\_id , Salary)***

Create EMPLOYEE {

    Emp\_no varchar(50);

    E\_name varchar(50);

    E\_address varchar(50);

    E\_ph\_no varchar(50);

    Dept\_no varchar(50);

    Dept\_name varchar(50);

    job\_id char(50);

    Salary int(7);

}

2. Add a new column; HIREDATE to the existing relation.

ALTER TABLE EMPLOYEE

ADD COLUMN HIREDATE VARCHAR(50);

3. Change the datatype of JOB\_ID from char to varchar2.

ALTER TABLE EMPLOYEE

ADD job\_id varchar(50);

4. Change the name of column/field Emp\_no to E\_no.

ALTER TABLE EMPLOYEE

RENAME Emp\_no TO E\_no;

5. Modify the column width of the job field of emp table.

ALTER TABLE EMPLOYEE ALTER COLUMN job\_id VARCHAR(20);

6. Remove the column S.No from the relation.

ALTER TABLE EMPLOYEE

DROP S.No;

7. Change the table name EMPLOYEE to EMP\_INFO.

ALTER TABLE EMPLOYEE RENAME TO EMP\_INFO;

8. Create a dummy table EMP like EMP\_INFO table.

Create TABLE Emp

AS SELECT \* FROM EMP\_INFO;

9. Test the truncate and drop command on EMP table.

TRUNCATE TABLE Emp;

ALTER TABLE Emp DROP E\_address;