DELHI TECHNOLOGICAL UNIVERSITY

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



# CO 202: DATABASE MANAGEMENT SYSTEMS

PRACTICAL FILE

# Submitted To: Submitted By:

**Ms. Indu Singh Neeraj Sharma**

**Assistant Professor 2K19/CO/244 Department of CSE, DTU A4 Batch Group-1**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Title** | **Date** | **Page No.** |
| 1. | Introduction to SQL, Database, and Database Management System. | January 18, 2021 | 3 |
| 2. | Introduction to Various software of DBMS and Database Languages. | January 25, 2021 | 5 |
| 3. | Introduction to ER Diagram and Symbol table. | February 8, 2021 | 10 |
| 4. | Introduction to Different types of constraints in SQL | February 22, 2021 | 14 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## EXPEREMENT NO – 1

**AIM:** Introduction to SQL, DATABASE and DATABASE MANAGEMENT SYSTEM.

## THEORY:

1. **SQL:**
   * It is a database computer language designed for the retrieval and management of data in a relational database. **SQL** stands for **Structured Query Language.**
   * It is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.
   * It allows users to query the database in number of ways, using English-like statements.

Some of **SQL** commands are:

* + - **SELECT:** It extracts data from database.
    - **INSERT INTO**: It inserts new data into a database.
    - **UPDATE**: It updates data in a database.
    - **DELETE**: It deletes data from database.
    - **CREATE DATABASE**: It creates a new database.
    - **ALTER DATABASE**: It modifies a database.
    - **CREATE TABLE**: It create a new table.
    - **ALTER TABLE**: It modifies the table.
    - **DROP TABLE**: It deletes a Table.
    - **CREATE INDEX**: It creates an index (search key).
    - **DROP INDEX**: It deletes an index.

## DATABASE:

* + A database is a data structure that stores organized information. Most databases contain multiple tables, which may each include several different fields. For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table.
  + Nearly all e-commerce sites use the databases to store product inventory and customer information. These sites use a database management system (or DBMS), such as Microsoft Access, FileMaker Pro, or MySQL as the "back end" to the website. By storing website data in a database, the data can be easily searched, sorted, and updated. This flexibility is important for e- commerce sites and other types of dynamic websites.

## DATABASE MANAGEMENT SYSTEM:

* + A Database Management System is a software that is designed to help create and maintain databases. It allows for the efficient storage of data. Popular DBMS software’s are My SQL, Oracle, FoxPro etc. Some of its common functions are: o Data Dictionary Management: DBMS stores description about the data in the data dictionary and uses it to look up the required structures and relationships.
  + Data Storage Management: It efficiently stores all data in the database and manages it by itself. The user doesn’t need to know how the data is being stored.
    - Efficient Query Processing: DBMS allows efficient query processing as queries can be executed in simple English-like syntaxes.
    - Restrict Unauthorized Access: It uses an Access Control Matrix to manage to maintain data integrity and prevent any intrusions.

**Learning:** In this particular experiment I learned SQL, a lot of SQL commands and about DBMS.

## CONCLUSION:

This experiment introduces us to the concepts of database, DBMS and SQL. It also gives us a small outlook on SQL commands.

## EXPEREMENT NO – 2

**AIM:** Introduction to various software of Database Management System and Database Language.

## THEORY:

1. **DBMS** SOFTWARE:
   * MySQL
   * Microsoft Access
   * Oracle
   * PostgreSQL
   * dbase
   * FoxPro
   * SQLite
   * IBM DB2
   * LibreOffice Base
   * MariaDB
   * Microsoft SQL Server
   * MongoDB

## MYSQL:

MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web- based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company.

## SQLite:

SQLite is an in-process library that implements a self-contained, serverless, zero- configuration, transactional SQL database engine. It is a database, which is zero- configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

## Oracle:

It is the most widely used DBMS software in the world. It supports multiple Linux, Unix and Windows versions. The latest version, Oracle 12c supports cloud

services. It is a relational database management software. It is secured, occupies less space, supports large databases, and reduces CPU time to process data

## MongoDB:

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

## Microsoft Access:

It is a DBMS software that uses Graphical User Interface (GUI) and software development tools. It is considered ideal for beginners to learn basics of database due to its use of GUI. It is used by most e-commerce websites. It orks only on Microsoft Windows.

## Microsoft SQL Server:

Developed by Microsoft in 1989, it works on both Linux and Windows Operating Systems. The language used is Assembly C, Linux, C++ for writing it. It is compatible with Oracle, provides efficient management of workload and allows multiple users to use the same database.

## Database Languages:

Database Languages are used to perform various types of operations on a database. It is used to define and manipulate a database. It is of five major types. They are:

## Data Manipulation Language:

A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

DML resembles simple English language and enhances efficient user interaction with the system. The functional capability of DML is organized in manipulation commands like SELECT, UPDATE, INSERT INTO and DELETE FROM, as described below:

* + SELECT: This command is used to retrieve rows from a table. The syntax is SELECT [column name(s)] from [table name] where [conditions]. SELECT is the most widely used DML command in SQL.
  + UPDATE: This command modifies data of one or more records. An update command syntax is UPDATE [table name] SET [column name = value] where [condition].
  + INSERT: This command adds one or more records to a database table. The insert command syntax is INSERT INTO [table name] [column(s)] VALUES [value(s)].
  + DELETE: This command removes one or more records from a table according to specified conditions. Delete command syntax is DELETE FROM [table name] where [condition].

## Data Definition Language:

* + **DDL** stands for **D**ata **D**efinition **L**anguage. It is used to define database structure or pattern.
  + It is used to create schema, tables, indexes, constraints, etc. in the database.
  + Using the DDL statements, you can create the skeleton of the database.
  + Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Here are some tasks that come under DDL:

* + **Create:** It is used to create objects in the database.
  + **Alter:** It is used to alter the structure of the database.
  + **Drop:** It is used to delete objects from the database.
  + **Truncate:** It is used to remove all records from a table.
  + **Rename:** It is used to rename an object.
  + **Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

## Data Control Language:

* **DCL** stands for **D**ata **C**ontrol **L**anguage. It is used to retrieve the stored or saved data.
* The DCL execution is transactional. It also has rollback parameters.

(But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

* **Grant:** It is used to give user access privileges to a database.
* **Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke:

CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

## Transaction Control Language:

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

* **Commit:** It is used to save the transaction on the database.
* **Rollback:** It is used to restore the database to original since the last Commit.

## Session Control Language:

* Session Control Language: It is referred to as SCL. It is used to dynamically manage the properties of a session. The SCL commands

are as follows. o Alter Session: It is used to modify conditions or parameters affecting database connections.

* o Set Role: It is used to enable or disable the roles that are currently enabled for the session.

**Learning:** In this particular experiment we learned different types of softwares for DBMS And Database Language.

## CONCLUSION:

This experiment introduced us to various DBMS software, about various Data Languages. Various SQL commands were briefly discussed.

## EXPEREMENT NO – 3

**AIM:** Introduction to Entity Relationship (ER) Diagram, Symbol Table.

## THEORY:

ER DIAGRAM:

**ER Model** stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well- designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.

ER Modeling helps you to analyze data requirements systematically to produce a well- designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

ER diagrams are a visual tool which is helpful to represent the ER model. It was proposed by Peter Chen in 1971 to create a uniform convention which can be used for relational database and network. He aimed to use an ER model as a conceptual modeling approach.

**Why use ER Diagrams?**

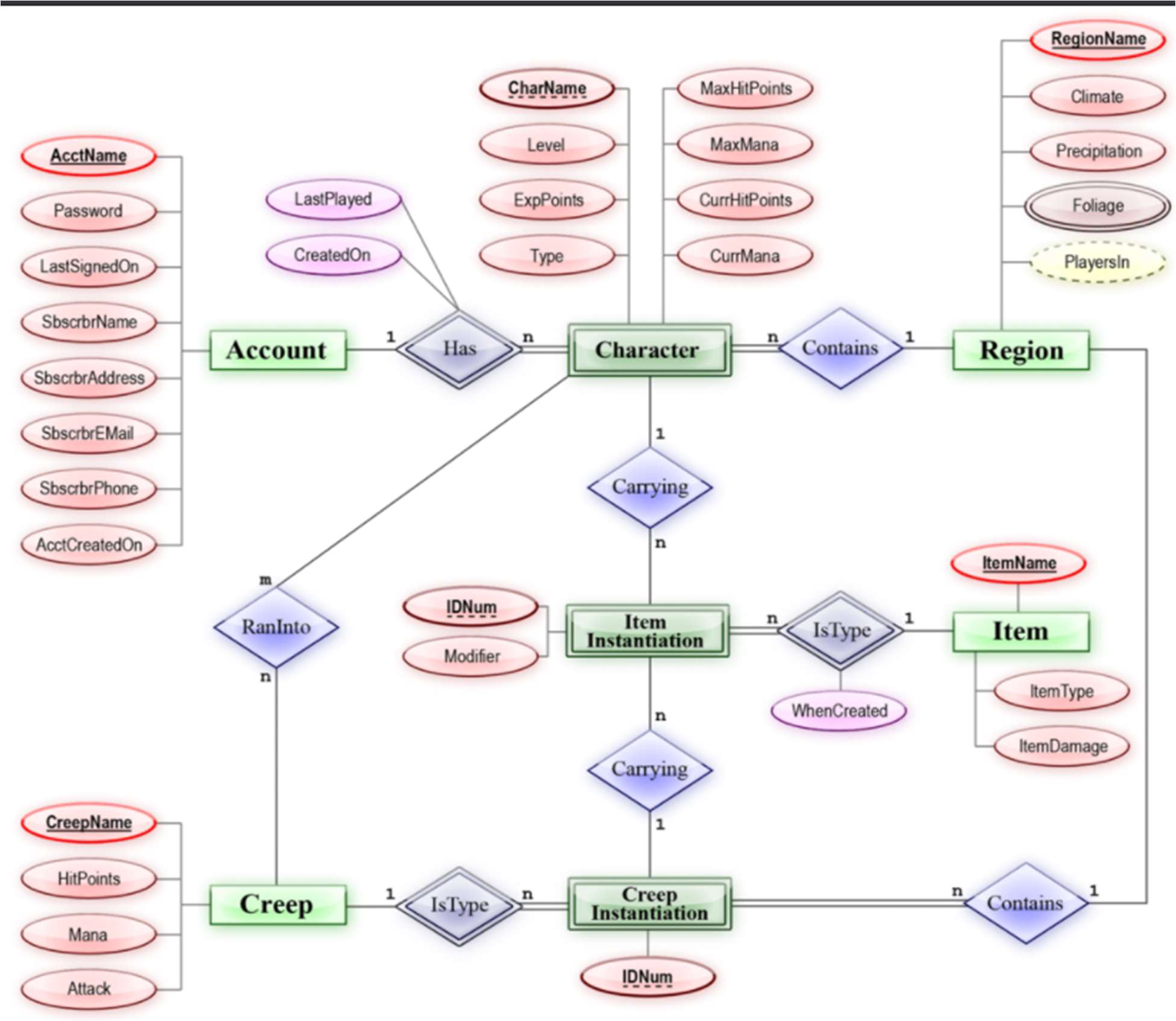
Here, are prime reasons for using the ER Diagram

* Helps you to define terms related to entity relationship modeling
* Provide a preview of how all your tables should connect, what fields are going to be on each table
* Helps to describe entities, attributes, relationships
* ER diagrams are translatable into relational tables which allows you to build databases quickly
* ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
* The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
* ERD Diagram allows you to communicate with the logical structure of the database to users

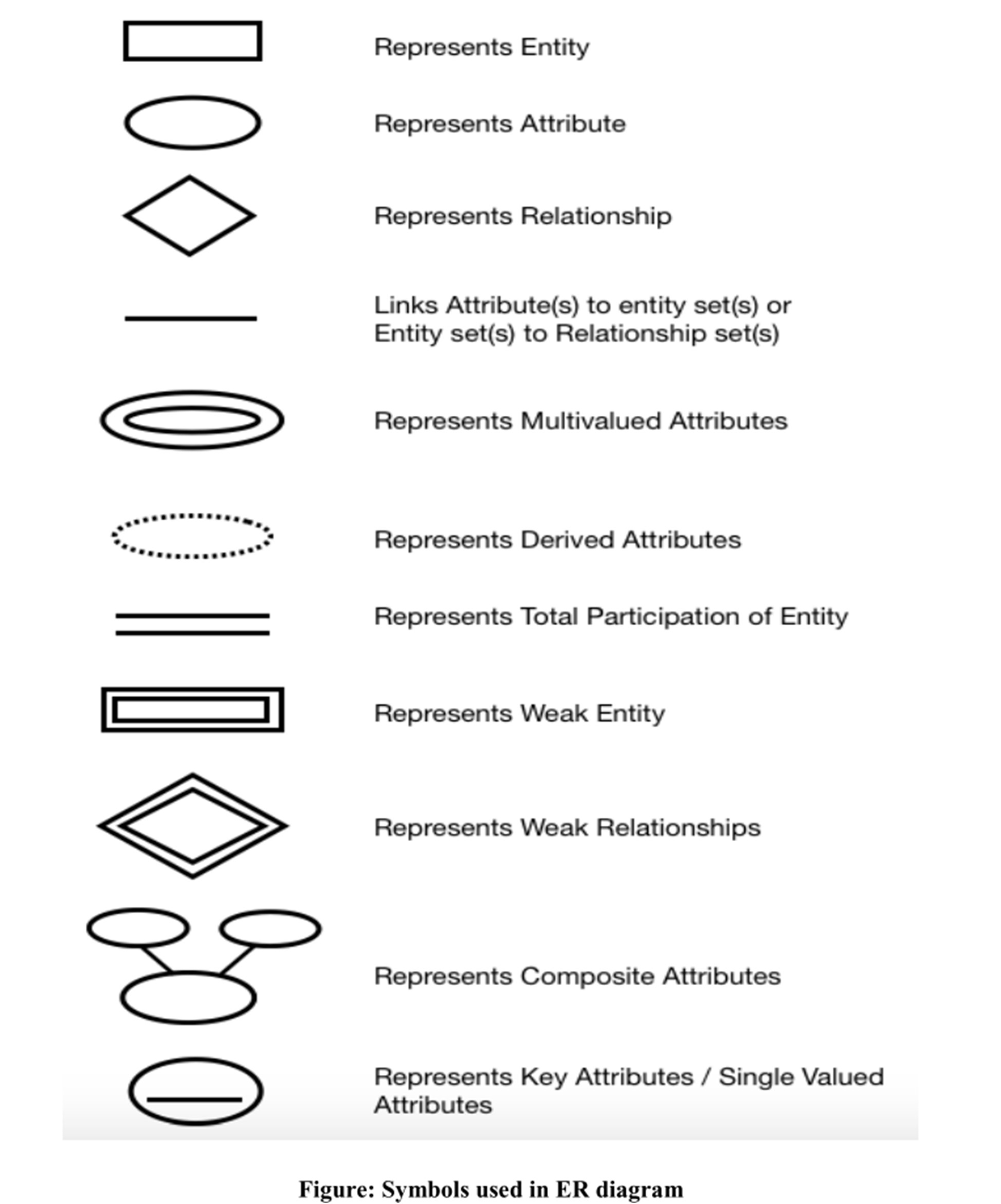
Disadvantages of using an ER diagram:

 Loss of information content: Some information be lost or hidden in ER model  Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.

 No representation of data manipulation: It is difficult to show data manipulation in ER model.



**Symbol Tables:** Symbol table is an important data structure created and maintained by compilers in order to store information about the occurrence of various entities such as variable names, function names, objects, classes, interfaces, etc.



**Following are the main components and its symbols in ER Diagrams:**

* **Rectangles:** This Entity Relationship Diagram symbol represents entity types
* **Ellipses:** Symbol represent attributes
* **Diamonds:** This symbol represents relationship types
* **Lines:** It links attributes to entity types and entity types with other relationship types
* **Primary key:** attributes are underlined
* **Double Ellipses:** Represent multi-valued attributes

**Learning:** In this particular experiment we learned about the ER diagram and symbol table.

## CONCLUSION:

In this experiment, the concept of ER Diagrams and Symbol Table was tried to explained. The various symbols used to represent various components of an ER diagram.

## EXPEREMENT NO – 4

**AIM:** Introduction to different types of Constraints in SQL.

## THEORY:

Constraints are the rules that we can apply on the type or data in a table. That is, we can specify the limit on type of data can be stored in a particular column in a table using constraints.

The available constraints in SQL are:

* **NOT NULL:** This constraint tells that we cannot store a NULL value in a
* column. That is, if a column is specified as NOT NULL then we will not be able
* to store null in this particular column anymore.
* **UNIQUE:** This constraint when specified with a column, tells that all the values
* in the column must be unique. That is, the value in any row of a column must not
* be repeated.
* **PRIMARY KEY:** A primary key is a field which can uniquely identify each row
* in a table. And this constraint is used to specify a field in a table as primary key.
* **FOREIGN KEY:** A Foreign key is a field which can uniquely identify each row
* in another table. And this constraint is used to specify a field as Foreign key.
* **CHECK:** This constraint helps to validate the values of a column to meet a
* particular condition. That is, it helps to ensure that he value stored in a column
* meets a specific condition.
* **DEFAULT:** This constraint specifies a default value for the column when no
* value in specified by user.

## NOT NULL CONSTRAINT:

By default, a column can hold NULL values. If you do not want a column to have a NULL value, then you need to define such a constraint on this column specifying that NULL is now not allowed for that column.

**Example**

For example, the following SQL query creates a new table called CUSTOMERS and adds five columns, three of which, are ID NAME and AGE, In this we specify not to accept NULLs −

CREATE TABLE CUSTOMERS ( ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT

NOT NULL,

ADDRESS CHAR (25),

SALARY DECIMAL (18, 2), PRIMARY KEY (ID)

);

## UNIQUE CONSTRAINT:

he UNIQUE constraint uniquely identifies each record in a database table.

The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns.

For eg.

CREATE TABLE Persons (

P\_Id int NOT NULL UNIQUE,

LastName varchar(255) NOT NULL, FirstName varchar(255), Address varchar(255),

City varchar(255)

)

DEFAULT **CONSTRAINT**: It is used to provide a default value to a column when none is specified. For example, country of students could be given a default value.

CREATE TABLE Students ( Roll\_No. int NOT NULL UNIQUE, Name varchar(255) NOT NULL, Address varchar(255), Age int, Country varchar(255) DEFAULT 'India'

);

## PRIMARY KEY CONSTRAINT:

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

CREATE TABLE Persons ( ID int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255), Age int,

PRIMARY KEY (ID)

);

## CHECK CONSTRAINT:

It is used to ensure that a certain condition is met when a record is entered into a relation. For example: students studying in a school should not be more than 25 years old.

CREATE TABLE Students ( Roll\_No. int NOT NULL UNIQUE PRIMARY KEY, Name varchar(255) NOT NULL, Address varchar(255), Age int NOT NULL CHECK (age<=25), Country varchar(255) DEFAULT 'India'

);

**FOREIGN KEY CONSTRAINT:** A It is used to indicate a row that uniquely identifies a record in another table. For

example: department id acts as foreign key in employees table which acts as primary key

in departments table. CREATE TABLE Employees (

emp\_id int NOT NULL UNIQUE PRIMARY KEY,

Name varchar(255) NOT NULL, salary int,

dept\_id int,

FOREIGN KEY(dept\_id) REFERENCES departments(dept\_id)

);

## CONCLUSION:

We learned SQL contains multiple constraints and we also learned how to use them.