**DBMS - MINI PROJECT**

TEAM MEMBERS :

PRINITHA V(RA1911003010902)

JYOTHISWAR CHALLA(RA1911003010914)

**DATABASE DESIGN**

**INTRODUCTION :**

The database is a collection of information and is systematically stored in tables in the form of rows and columns. The table in the database has unique name that identifies its contents. The database in turn is further described in detail giving all the fields used with the data types, constraints available, primary key and foreign key.

Database design is used to manage large of information. In this database we describe the entire 4 table available in the software, which are used to store all the records.

**ABSTRACT:**

Student Grade Calculator (SGC) can be used to calculate a percentage based on the marks of students. (SGC) is a fairly reliable indicator of student results.

It allows obtaining student marks for all students course wise. Manipulate the result for a department only where there may be several programs. The instructor can record students’ marks for a particular course for which he/she teaches. Data is used to generate result report.

The students can see the report for all courses in a particular semester whereas an instructor can enter the marks for a course he teaches.Once an instructor submitted the marks sheet he/she cannot change furthermore.

**Data types and its description:**

Fields in database table have a data type used in database table are explained below.

**Integer**: one optional sign character (+ or -) followed by at least one digit (0-9). Leading and trailing blanks are ignored. No other character is allowed.

**Varchar**: It is used to store alpha numeric characters. In this data type we can set the maximum number of characters up to 8000 ranges by defaults SQL server will set the size to 50 characters range.

**Data/time**: Data/time data type is used for representing date or time.

**Database Name : GRADES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Relation** |
| Student\_id | int | 50 | Primary key |
| Student\_name | Varchar | 50 | Not null |
| semester | int | 50 | Not null |
| marks | int | 50 | Not null |
| Reg\_no | Varchar | 50 | Not null |
| Rank | int | 50 | Not null |
| average | Varchar | 50 | Not null |
| dob | date | 50 | Not null |

**Database Name: student**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Relation** |
| Student\_Name | Varchar | 50 | Not null |
| Regno | Varchar | 50 | Primary key |
| Addresss | Varchar | 50 | Not null |
| Mail\_id | varchar | 50 | Not null |
| Course | Varchar | 50 | Not null |
| Student\_id | Varchar | 50 | Foreign key |
| Gender |  | 50 |  |

**Database Name: results**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Relation** |
| Regno | Varchar | 50 | Foreign key |
| studentname | Varchar | 50 | Not Null |
| Result\_type | Varchar | 50 | Not Null |
| Result\_desp | Varchar | 50 | Not Null |

**Entity Relationship Diagram:**

Entity Relationship Diagram is used in modern database software engineering to illustrate logical structure of database. It is a relational schema database modeling method used to

Model a system and approach. This approach commonly used in database design. The diagram created using this method is called ER-diagram.

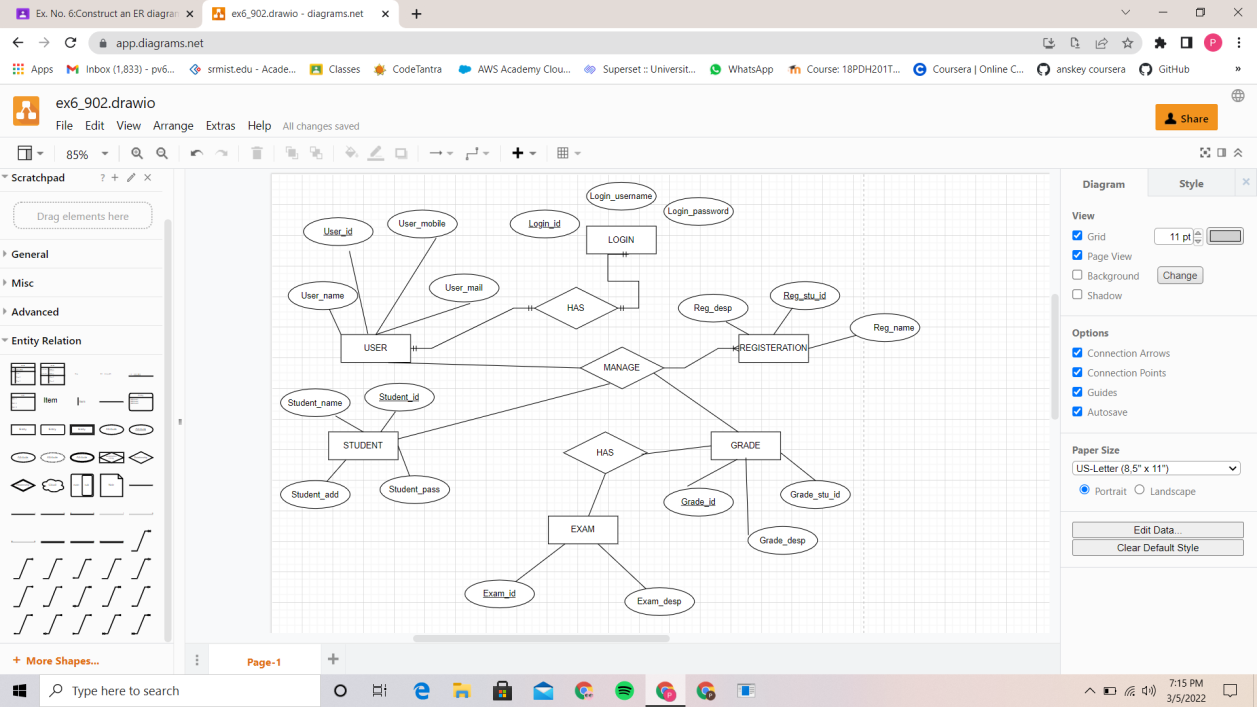
The ER-diagram depicts the various relationships among entities, considering each object as entity. Entity is represented as rectangle shape and relationship represented as diamond shape. It depicts the relationship between data object. The ER-diagram is the notation that is used to conduct the data modeling activity.

**Entity:** Entity is the things which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object defined within the information system about which you want to store information. Entities are distinct things in the enterprise.

**Relationship:** A relationship is normal connection or association between entities used to relate two or more entities with some common attributes or meaningful interaction between the object.

**Attributes:** Attributes are the properties of the entities and relationship descriptor of the entity. Attributes are elementary pieces of information attached to an entity.

**ER DIAGRAM :**

****