

2218-CSE-5330-002-DATABASE SYSTEMS

KORIKANA SAI TEJA SREENIVAS - 1001964006

NAVYASHREE BUDHIHAL MUTT - 1001965572

Project 2

Entities:

- **User_accounts table** : Contains attributes such as UserID , UserName, PhoneNo ,RoleID
- **User_Roles table** : Contains attributes such as RoleName , Description, RoleID
- **Tables table** : Contains attributes such as TableName , TableID , Description , UserID
- **Privileges table** : Contains attributes such as PrivilegeName , PrivilegeID , PrivilegeType , Description
 - **Account_Privileges**
 - **Relation_Privileges**

Relations between entities:

- **Owner_of**: it is the relation between User_accounts and Tables, where one User_account is the owner_of one table it is sharing one to one relation
- **Has**: it is the relation between User_accounts and User_roles , where one User_roles have many User_accounts here it shares many to one relationship
- **Related**: it is the relation between Tables and Relational_Privileges where many Tables is related to many Relation_Privileges

- **Owns**: relation between User_Roles and Account-Privileges here it shares many to many relationship
- **Related** :it is a relation between User_Roles and Tables here it shares many to many relationship

Attributes of each entity:

User_Account:

- UserID: It consists the ID's of the users (Primary key)
- UserName: It consists names of the users
- PhoneNo: Phone numbers of the user
- RoleID: Role ID of the user (Foreign Key)

User_Roles:

- RoleName:Role of the user
- Description: Role description
- RoleID:it consists role id's (Primary key)

Tables:

- TableName: Name of the table where the user_account are stored
- TableID: ID's of the Tables(Primary Key)
- Description:Tables description
- UserID:contains ID's from User_account(Foreign Key)

Privileges:

- PrivilegeName: Name of the Privilege
- PrivilegeID: Id's of the Privilege(Primary Key)
- PrivilegeType: type of the Privilege
- Description:Privilege description

Assumptions made while designing EER diagram:

- Privileges is divided into 2 different types
 - 1)Account_Privileges
 - 2)Relation_Privileges
- By using a Disjoint relation super-set can be related in multiple sub-class sets
- A relation “Related” between Tables and Relation_Privileges , has N:N cardinality relation
- A relation “Has” between User_Roles and Account_Privileges N:N cardinality relation
- A relation “Related” between User_Roles and Tables N:N cardinality relation

EER to Relational Schema:**1.User_Account Table:**

(UserID(Primary Key), UserName, PhoneNo, RoleID(Foreign Key))

2.User_Roles Table:

(RoleID (Primary Key) , Description , RoleName)

3.Tables Table:

(TableID (Primary Key) , UserID(Foreign Key) , TableName , Description)

4.Privileges Table:

(PrivilegeName, PrivilegeID (Primary Key) , PrivilegeName , Description)

Create table Queries

```
CREATE TABLE USER_ACCOUNTS
( UserName          VARCHAR(10)      ,
  PhoneNo           INT               ,
  UserID            INT               NOT NULL,
  RoleID            INT               ,
  PRIMARY KEY       (UserID)
  FOREIGN KEY (RoleID) references USER_ROLES (RoleID));
```

```
CREATE TABLE TABLES
( TableID           VARCHAR(10)       NOT NULL,
  TableName         VARCHAR(10)      ,
  Description       VARCHAR(20)      ,
  UserID            INT               ,
  PRIMARY KEY       (TableID)         ,
  foreign key (UserID) references USER_ACCOUNTS (UserID));
```

```
CREATE TABLE USER_ROLES
( RoleID            INT               NOT NULL,
  RoleName          VARCHAR(20)      ,
  Description       VARCHAR(20)      ,
```

```
PRIMARY KEY (RoleID) );
```

```
CREATE TABLE PRIVILEGES
```

```
(PrivilegeID      VARCHAR(10)      NOT NULL,  
  PrivilegeName    VARCHAR(10)      ,  
  Description      VARCHAR(20)      ,  
  PrivilegeType    VARCHAR(20)      ,  
  PRIMARY KEY (PrivilegeID) );
```

```
CREATE TABLE OWNS
```

```
( RoleID          INT          ,  
  PrivilegeID     VARCHAR(20)   ,  
  PRIMARY KEY (RoleID,PrivilegeID),  
  foreign key (RoleID) references USER_ROLES(RoleID));  
  foreign key (PrivilegeID) references PRIVILEGE(PrivilegeID));
```

```
CREATE TABLE RELATED
```

```
( RoleID          INT          ,  
  PrivilegeID     VARCHAR(20)   ,  
  TableID         VARCHAR(10)   ,  
  PRIMARY KEY (RoleID,PrivilegeID,TableID) ,  
  foreign key (RoleID) references USER_ROLES(RoleID) ,  
  foreign key (TableID) references TABLES (TableID) ,  
  foreign key (PrivilegeID) references PRIVILEGE(PrivilegeID));
```

