

```
CREATE TABLE User(  
  User_Id INTEGER AUTO_INCREMENT,  
  First_Name VARCHAR(50) NOT NULL,  
  Last_Name VARCHAR(50) NOT NULL,  
  Email_Address VARCHAR(50) NOT NULL,  
  Password VARCHAR(50) NOT NULL,  
  Address VARCHAR(100),  
  City VARCHAR(50),  
  State VARCHAR(2),  
  Zip_Code VARCHAR(5),  
  Telephone VARCHAR(11),  
  Gender VARCHAR(1),  
  Date_Of_Birth DATE,  
  Rating INTEGER DEFAULT 0,  
  PRIMARY KEY(User_Id),  
  UNIQUE(Email_Address)  
);
```

```
CREATE TABLE Employee(  
  Employee_Id INTEGER AUTO_INCREMENT,  
  SSN INTEGER NOT NULL,  
  Password VARCHAR(50) NOT NULL,  
  First_Name VARCHAR(50) NOT NULL,  
  Last_Name VARCHAR(50) NOT NULL,  
  Hourly_Rate INTEGER NOT NULL,  
  Start_Date DATE NOT NULL,  
  Role VARCHAR(50) NOT NULL,  
  Address VARCHAR(100),  
  City VARCHAR(50),  
  State VARCHAR(2),  
  Zip_Code VARCHAR(5),  
  Telephone VARCHAR(11),  
  PRIMARY KEY (Employee_Id),  
  UNIQUE (SSN)  
);
```

```
CREATE TABLE Circle(  
  Circle_Id INT AUTO_INCREMENT,  
  Circle_NAME VARCHAR(100),  
  Owner_Of_Circle INTEGER,  
  Type VARCHAR(50),  
  PRIMARY KEY(Circle_Id),  
  FOREIGN KEY (Owner_Of_Circle) REFERENCES User(User_Id) ON DELETE CASCADE  
);
```

```
CREATE TABLE AddedTo(  
  User_Id INTEGER,  
  Circle_Id INTEGER,  
  PRIMARY KEY(User_Id,Circle_Id),  
  FOREIGN KEY (User_Id) REFERENCES User(User_Id) ON DELETE CASCADE,  
  FOREIGN KEY (Circle_Id) REFERENCES Circle(Circle_Id) ON DELETE CASCADE  
);
```

```
CREATE TABLE InviteRequest(  
  User_Id INTEGER,  
  Circle_Id INTEGER,  
  PRIMARY KEY(User_Id,Circle_Id),  
  FOREIGN KEY (User_Id) REFERENCES User(User_Id) ON DELETE CASCADE,  
  FOREIGN KEY (Circle_Id) REFERENCES Circle(Circle_Id) ON DELETE CASCADE  
);
```

```
CREATE TABLE JoinRequest (  
  User_Id INTEGER,  
  Circle_Id INTEGER,  
  PRIMARY KEY(User_Id,Circle_Id),  
  FOREIGN KEY (User_Id) REFERENCES User(User_Id) ON DELETE CASCADE,  
  FOREIGN KEY (Circle_Id) REFERENCES Circle(Circle_Id) ON DELETE CASCADE  
);
```

```
CREATE TABLE Account(  
  Account_Number INTEGER AUTO_INCREMENT,  
  User_Id INTEGER,  
  Account_Creation_Date DATETIME NOT NULL,  
  Credit_Card_Number VARCHAR(16),  
  PRIMARY KEY(Account_Number),  
  FOREIGN KEY (User_Id) REFERENCES User(User_Id) On Delete Set NULL,  
  CHECK(Account_Number>0),
```

```
UNIQUE(Credit_Card_Number)
);
```

```
CREATE TABLE User_Preferences(
  Id INTEGER,
  Preference VARCHAR(50),
  PRIMARY KEY (Id, Preference),
  FOREIGN KEY(Id) REFERENCES User(User_Id) On Delete Cascade
);
```

```
CREATE TABLE Message(
  Message_Id INTEGER AUTO_INCREMENT,
  Date DATETIME NOT NULL ,
  Subject VARCHAR(50),
  Content VARCHAR(1000),
  Sender INTEGER,
  Receiver INTEGER,
  PRIMARY KEY (Message_Id) ,
  CHECK (Message_Id>0),
  FOREIGN KEY (Sender) REFERENCES User(User_Id) On Delete Set NULL,
  FOREIGN KEY (Receiver) REFERENCES User(User_Id) On Delete Cascade
);
```

```
CREATE TABLE Has_Manager(
  Employee INTEGER,
  Manager INTEGER,
  PRIMARY KEY (Employee, Manager),
  FOREIGN KEY (Employee) REFERENCES Employee(Employee_ID) On Delete Cascade,
  FOREIGN KEY (Manager) REFERENCES Employee(Employee_ID) On Delete Cascade
);
```

```
CREATE TABLE Post(
  Post_Id INT AUTO_INCREMENT,
  Date DATETIME NOT NULL ,
  Content VARCHAR(50),
  Comment_Count INTEGER,
  Circle INTEGER,
  Author INTEGER,
  PRIMARY KEY (Post_Id),
  FOREIGN KEY (Circle) REFERENCES Circle(Circle_Id) ON DELETE CASCADE,
  FOREIGN KEY (AUTHOR) REFERENCES User(User_Id) ON DELETE CASCADE
);
```

```
CREATE TABLE Comment(  
  Comment_Id INT AUTO_INCREMENT,  
  Date DATETIME NOT NULL ,  
  Content VARCHAR(50),  
  Post INTEGER,  
  Author INTEGER,  
  PRIMARY KEY (Comment_Id),  
  FOREIGN KEY (Post) REFERENCES Post(Post_Id) ON DELETE CASCADE,  
  FOREIGN KEY (AUTHOR) REFERENCES User(User_Id) ON DELETE CASCADE  
);
```

```
CREATE TABLE User_Likes_Post(  
  User INTEGER,  
  Post INTEGER,  
  PRIMARY KEY(User,Post),  
  FOREIGN KEY (User) REFERENCES User(User_Id) On Delete Cascade,  
  FOREIGN KEY (Post) REFERENCES Post(Post_Id) On Delete Cascade  
);
```

```
CREATE TABLE User_Likes_Comment(  
  User INTEGER,  
  Comment INTEGER,  
  PRIMARY KEY(User,Comment),  
  FOREIGN KEY (User) REFERENCES User(User_Id) On Delete Cascade,  
  FOREIGN KEY (Comment) REFERENCES Comment(Comment_Id) On Delete Cascade  
);
```

```
CREATE TABLE Advertisement(  
  Advertisement_Id INTEGER AUTO_INCREMENT,  
  Employee INTEGER,  
  Type VARCHAR(50),  
  Date DATETIME NOT NULL ,  
  Company VARCHAR(50),  
  Item_Name VARCHAR(50),  
  Content VARCHAR(200),  
  Unit_Price INTEGER,  
  Available_Units INTEGER,  
  PRIMARY KEY (Advertisement_Id),  
  CHECK (Advertisement_Id>0),  
  FOREIGN KEY (Employee) REFERENCES Employee(Employee_Id) On Delete Set Null  
);
```

```
CREATE TABLE Purchase(  
Transaction_Id INTEGER AUTO_INCREMENT,  
Date DATETIME NOT NULL ,  
Advertisement INTEGER,  
Number_Of_Units INTEGER,  
Account INTEGER,  
PRIMARY KEY (Transaction_Id),  
CHECK (TRANSACTION_Id>0),  
FOREIGN KEY (Advertisement) REFERENCES Advertisement(Advertisement_Id) On Delete  
Set Null,  
FOREIGN KEY (Account) REFERENCES Account(Account_Number) On Delete Set Null  
);
```