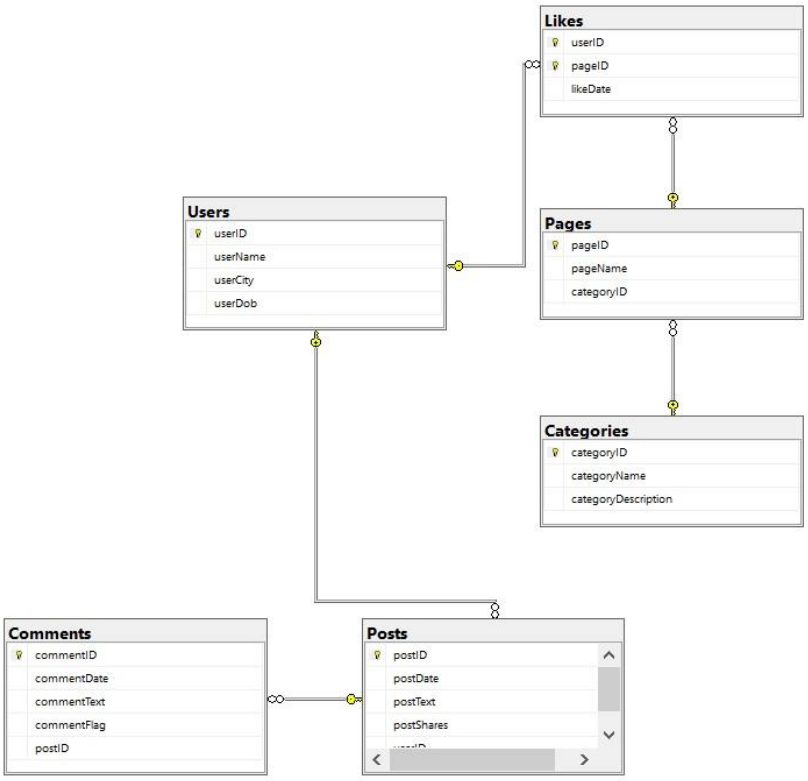
1. create database MiniFacebook

```
use MiniFacebook
go
create table Users(
        userID int primary key identity(1,1),
        userName varchar(255),
        userCity varchar(255),
        userDob varchar(255)
)
create table Categories(
        categoryID int primary key identity(1,1),
        categoryName varchar(255),
        categoryDescription varchar(255)
)


create table Pages(
        pageID int primary key identity(1,1),
        pageName varchar(255),
        categoryID int foreign key references Categories(categoryID)
)

create table Likes(
        userID int references Users(userID),
        pageID int references Pages(pageID),
        likeDate varchar(255),
        constraint likeID primary key(userID, pageID)
)
```

```
create table Posts(
	postID int primary key identity(1,1),
	postDate varchar(255),
	postText varchar(255),
	postShares int,
	userID int foreign key references Users(userID)
)


create table Comments(
	commentID int primary key identity(1,1),
	commentDate Date,
	commentText varchar(255),
	commentFlag bit,
	postID int foreign key references Posts(postID)
)

insert Users values('Cati', 'Suceava', '2010-10-10'), ('Andreea', 'Bucuresti', '1998-10-10')
insert Categories values('Category1', 'fabulous'), ('Category2', 'aw3some')
insert Pages values('Page1', 1), ('Page2', 1)
insert Likes values(1,1,'2/21/2020'), (1,2,'16/7/2020'), (2,1, '12/3/2017')
insert Posts values ('1/1/2018', 'Post1', 3, 1), ('3/23/2018', 'Post2',5, 2)
insert Comments values ('6/8/2019','damn', 1,1), ('6/9/2017', 'damn u nice af', 0,2)
```



2.

1. connection String :

```
string connectionString = @"Data Source = .\SQLEXPRESS; Initial Catalog=MiniFacebook; Integrated Security = True";
```

2. Populate child and parent tables :

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    using(SqlConnection sqlConnection = new SqlConnection(connectionString))
    {
        sqlConnection.Open();
        DataSet ds = new DataSet();

        SqlDataAdapter daChild = new SqlDataAdapter("select * from Posts", sqlConnection);
        SqlDataAdapter daParent = new SqlDataAdapter("select * from Users", sqlConnection);

        SqlCommandBuilder cb = new SqlCommandBuilder(daChild);

        daParent.Fill(ds, "Users");
        daChild.Fill(ds, "Posts");

        DataRelation dr = new DataRelation("FK_Users_Posts", ds.Tables["Users"].Columns["userID"], ds.Tables["Posts"].Columns["userID"]);

        ds.Relations.Add(dr);

        BindingSource bsParent = new BindingSource();
        BindingSource bsChild = new BindingSource();

        bsParent.DataSource = ds;
        bsParent.DataMember = "Users";

        bsChild.DataSource = bsParent;
        bsChild.DataMember = "FK_Users_Posts";

        dgvPosts.DataSource = bsChild;
        dgvUsers.DataSource = bsParent;

        daChild.Update(ds, "Posts");

    }
}
```

3. Add new Post:

```
1 reference
private void addButton_Click(object sender, EventArgs e)
{
    using (SqlConnection sqlConnection = new SqlConnection(connectionString))
    {
        SqlDataAdapter daChild = new SqlDataAdapter("select * from Posts", sqlConnection);
        daChild.InsertCommand = new SqlCommand("INSERT INTO Posts(postDate, postText, postShares, userID) VALUES(@date, @text, @nrShares, @user)", sqlConnection);
        daChild.InsertCommand.Parameters.Add("@date", SqlDbType.VarChar).Value = postDate.Text;
        daChild.InsertCommand.Parameters.Add("@text", SqlDbType.VarChar).Value = postText.Text;
        daChild.InsertCommand.Parameters.Add("@nrShares", SqlDbType.Int).Value = Int32.Parse(postShares.Text);
        daChild.InsertCommand.Parameters.Add("@user", SqlDbType.Int).Value = Int32.Parse(userID.Text);

        sqlConnection.Open();

        daChild.InsertCommand.ExecuteNonQuery();

        sqlConnection.Close();

        MessageBox.Show("Post successfully added :)");

    }

}
```

4. Delete post:

```
1 reference
private void deleteButton_Click(object sender, EventArgs e)
{
    using (SqlConnection sqlConnection = new SqlConnection(connectionString))
    {
        SqlDataAdapter daChild = new SqlDataAdapter("select * from Posts", sqlConnection);

        var selectedRowPost = Int32.Parse(dgvPosts.SelectedRows[0].Cells[0].Value.ToString());

        daChild.DeleteCommand = new SqlCommand("delete from Posts where POSTid=@PID", sqlConnection);
        daChild.DeleteCommand.Parameters.Add("@PID", SqlDbType.Int).Value = selectedRowPost;

        sqlConnection.Open();

        daChild.DeleteCommand.ExecuteNonQuery();

        sqlConnection.Close();

        MessageBox.Show("Deleted sucessfully :)");
    }

}
```

5. Update post:

```
1 reference
private void updatePostButton_Click(object sender, EventArgs e)
{
    using (SqlConnection sqlConnection = new SqlConnection(connectionString))
    {
        SqlDataAdapter daChild = new SqlDataAdapter("select * from Posts", sqlConnection);

        var selectedRowParent = Int32.Parse(dgvPosts.SelectedRows[0].Cells[0].Value.ToString());
        daChild.UpdateCommand = new SqlCommand("UPDATE Posts SET postDate=@date, postText=@text, postShares = @nrShares, userID = @userID WHERE postID = @postID", sqlConnection);

        daChild.UpdateCommand.Parameters.Add("@date", SqlDbType.VarChar).Value = postDate.Text;
        daChild.UpdateCommand.Parameters.Add("@text", SqlDbType.VarChar).Value = postText.Text;
        daChild.UpdateCommand.Parameters.Add("@nrShares", SqlDbType.Int).Value = Int32.Parse(postShares.Text);
        daChild.UpdateCommand.Parameters.Add("@userID", SqlDbType.VarChar).Value = Int32.Parse(userID.Text);

        daChild.UpdateCommand.Parameters.Add("@postID", SqlDbType.Int).Value = selectedRowParent;

        sqlConnection.Open();

        daChild.UpdateCommand.ExecuteNonQuery();

        sqlConnection.Close();

        MessageBox.Show("Updated with success");
    }
}
```

3. use MiniFacebook

go

---Non-repeatable reads

--T1

BEGIN TRAN

WAITFOR DELAY '00:00:05'

UPDATE Categories SET categoryDescription='DESCRIPTION NONREPEATABLE READS' WHERE categoryID = 2

COMMIT TRAN

--T2

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRAN

SELECT * FROM Categories

WAITFOR DELAY '00:00:05'

SELECT * FROM Categories

COMMIT TRAN

--Solution

--T1

BEGIN TRAN

WAITFOR DELAY '00:00:05'

UPDATE Categories SET

categoryDescription='DESCRIPTION NON REPEATABLEREADS' WHERE categoryID = 2

COMMIT TRAN

--T2

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRAN

SELECT * FROM Categories

WAITFOR DELAY '00:00:05'

SELECT * FROM Categories

COMMIT TRAN