

Seminary 6

Practical Exam

~ Solution ~

Practical Exam

Create a database for a MiniFacebook system.

- ✓ The entities of interest to the problem domain are: *Users, Pages, Likes, Categories, Posts, and Comments.*
- ✓ *Each user has a name, a current city and date of birth.*
- ✓ *A user can like multiple pages.*
- ✓ *The system stores the date of each like.*
- ✓ *A page has a name and a category, e.g., sports, movies, music, etc.*
- ✓ *A category also has a category description.*
- ✓ *Users write posts and comment on existing posts.*
- ✓ *A user's post has a date, text, and number of shares.*
- ✓ *A comment is anonymous, has a text, a date, and a flag indicating whether it's a top comment for the corresponding post.*

Practical Exam

- 1) Write an SQL script that creates the corresponding relational data model. (3p)
- 2) Create a master – detail windows form that allows searching the posts for a given user. The form should have at least the following components: a grid (combo box or list box) to display the users, a grid for all the posts of the selected user, and a button for saving added / deleted / modified posts. It's up to you how to add / delete / change a post. (4p)
- 3) Create a scenario that reproduces the non-repeatable reads concurrency issue, through stored procedures or standalone queries. Find a solution to solve / workaround this concurrency issue. (2p)

(1p of)

Solution :

Practical Exam

```
create database MiniFacebook_IE
go
use MiniFacebook_IE
go
```

```
create table Users(
  Uid int primary key identity(1,1),
  UName VARCHAR(50),
  UCity VARCHAR(50),
  UDoB Date
)
```

```
create table Categories(
  Cid int primary key identity(1,1),
  CName VARCHAR(50),
  CDescription varchar(50)
)
```

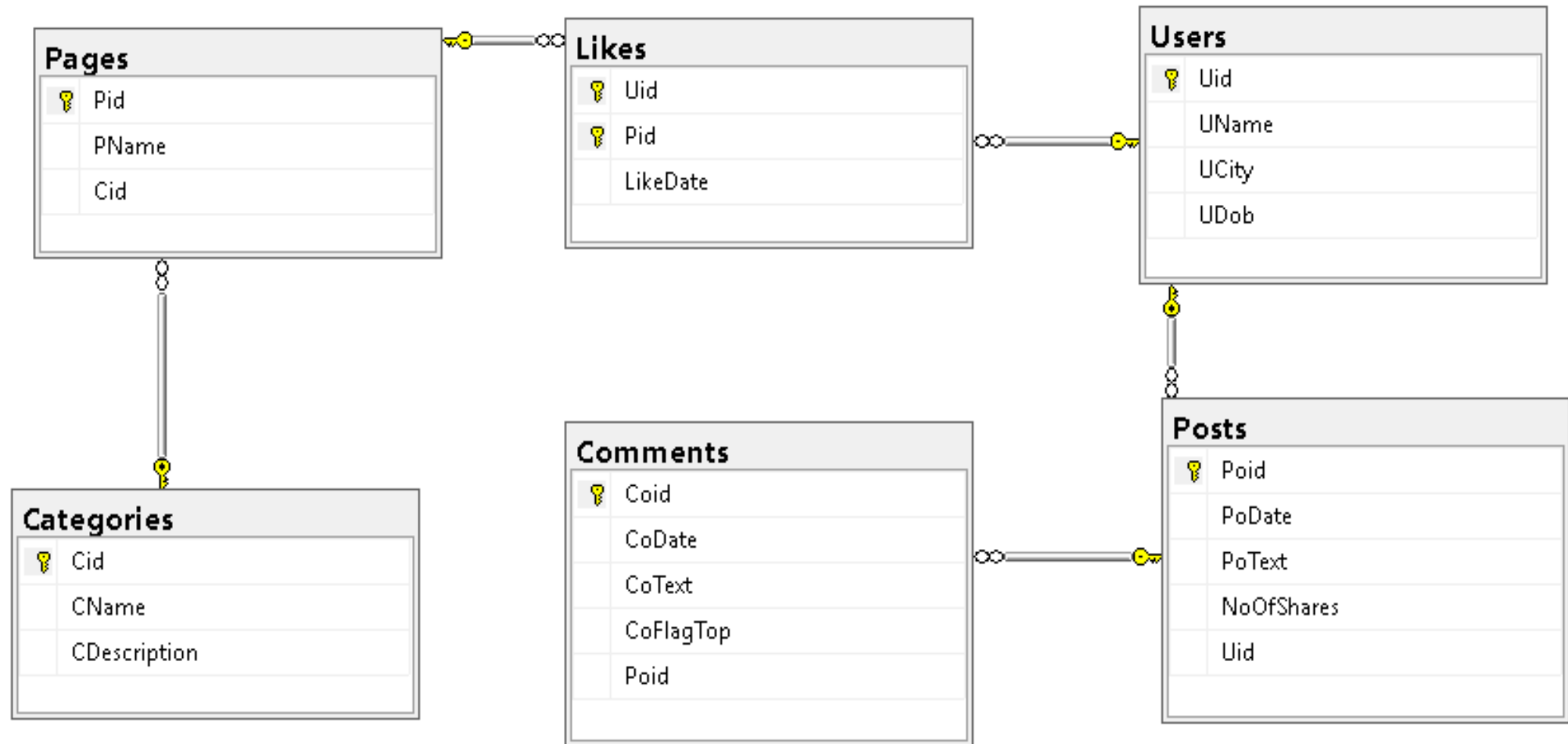
```
create table Pages(
  Pid int primary key identity(1,1),
  PName VARCHAR(50),
  Cid int FOREIGN KEY REFERENCES Categories(Cid)
)
```

```
create table Likes(
  Uid int references Users(Uid),
  Pid int references Pages(Pid),
  LikeDate date,
  CONSTRAINT pk_Like PRIMARY KEY(Uid, Pid)
)
```

```
create table Posts(
  Poid int primary key identity(1,1),
  PoDate date,
  PoText VARCHAR(50),
  NoOfShares int,
  Uid int FOREIGN KEY REFERENCES Users(Uid)
)
```

```
create table Comments(
  Coid int primary key identity(1,1),
  CoDate date,
  CoText varchar(50),
  CoFlagTop bit,
  Poid int FOREIGN KEY REFERENCES Posts(Poid)
)
```

Practical Exam



Practical Exam

- INSERT Users VALUES('User 1', 'Cluj-Napoca', '10/10/2000'), ('User 2', 'Bucuresti', '2/2/1998')
 - INSERT Categories VALUES('Category 1', 'very interesting'), ('Category 2', 'interesting')
 - INSERT Pages VALUES('Page 1', 1), ('Page 2', 1)
 - INSERT Likes VALUES(1,1,'2/21/2018'), (1,2,'6/7/2018'), (2,1, '2/3/2017')
 - INSERT Posts VALUES ('1/1/2018', 'Post 1', 3, 1), ('3/23/2018', 'Post 2',5, 2)
 - insert Comments values ('6/8/2019','comment 1', 1,1), ('6/9/2017', 'commmmmmmm', 0,2)
 - GO
-
- select * from Users
 - select * from Categories
 - select * from Pages
 - select * from Likes
 - select * from Posts
 - select * from Comments

Practical Exam

Results Messages

	Uid	UName	UCity	UDob
1	1	User 1	Cluj-Napoca	2000-10-10
2	2	User 2	Bucuresti	1998-02-02

	Cid	CName	CDescription
1	1	Category 1	very interesting
2	2	Category 2	interesting

	Pid	PName	Cid
1	1	Page 1	1
2	2	Page 2	1

	Uid	Pid	LikeDate
1	1	1	2018-02-21
2	1	2	2018-06-07
3	2	1	2017-02-03

	Poid	PoDate	PoText	NoOfShares	Uid
1	1	2018-01-01	Post 1	3	1
2	2	2018-03-23	Post 2	5	2

	Coid	CoDate	CoText	CoFlagTop	Poid
1	1	2019-06-08	comment 1	1	1
2	2	2017-06-09	commmmmmm	0	2

Activate Windows



Query executed successfully.

DESKTOP-ATJN5FL\SQLEXPRESS ... | DESKTOP-ATJN5FL\Emi (54) | MiniFacebook_IE | 00:00:00 | 13 rows

Practical Exam

2. Aplicatie Windows Forms Application

BindingSource_DataRelation (Running) - Microsoft Visual Studio

Form1

CONNECT Parent Table - Users UPDATE DB

Post Date

Post Text

Number of Shares

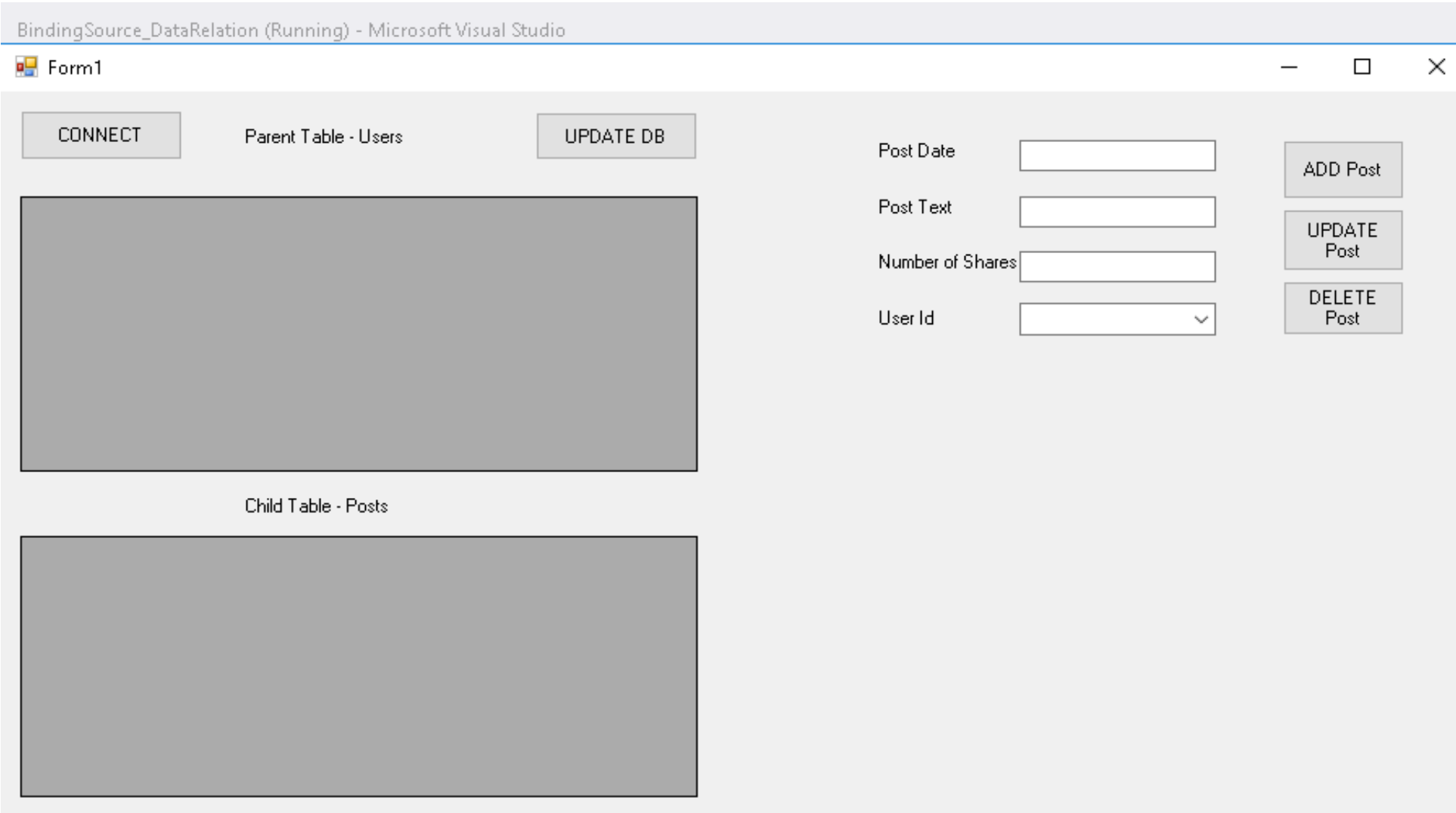
User Id

ADD Post

UPDATE Post

DELETE Post

Child Table - Posts



Practical Exam

Connect to Database – Parent-Child relationship (the tables Users-Posts)

```
connection = new SqlConnection(@"Data Source = DESKTOP-ATJN5FL\SQLEXPRESS; Initial Catalog=MiniFacebook_IE;  
Integrated Security = True");
```

```
ds = new DataSet();
```

```
daChild = new SqlDataAdapter("select * from Posts", connection);
```

```
daParent = new SqlDataAdapter("select * from Users", connection);
```

```
cb = new SqlCommandBuilder(daChild);
```

```
daParent.Fill(ds, "Users");
```

```
daChild.Fill(ds, "Posts");
```

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

```
ds.Relations.Add(dr);
```

Practical Exam

```
bsParent = new BindingSource();
```

```
bsChild = new BindingSource();
```

```
bsParent.DataSource = ds;
```

```
bsParent.DataMember = "Users";
```

```
bsChild.DataSource = bsParent;
```

```
bsChild.DataMember = "FK_Users_Posts";
```

```
GridChild.DataSource = bsChild;
```

```
GridParent.DataSource = bsParent;
```

```
Update Button: daChild.Update(ds, "Posts");
```

Practical Exam

Form1

CONNECT Parent Table - Users UPDATE DB

	Uid	UName	UCity	UDob
▶	1	User 1	City Transaction 1	10/10/2000
	2	User 2	Bucuresti	2/2/1998
*				

Post Date

Post Text

Number of Shares

User Id

ADD Post

UPDATE Post

DELETE Post

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
▶	1	1/1/2018	Post 1	3
*				

Form1

CONNECT Parent Table - Users UPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date

Post Text

Number of Shares

User Id

ADD Post

UPDATE Post

DELETE Post

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
▶	2	3/23/2018	Post 2	5
*				

Practical Exam

INSERT – in the child table (Posts)

```
daChild.InsertCommand = new SqlCommand("INSERT INTO Posts(PoDate, PoText, NoOfShares, Uid) VALUES  
(@d, @t, @n, @c)", connection);
```

```
daChild.InsertCommand.Parameters.Add("@d", SqlDbType.Date).Value = DateTime.Parse(textBox1.Text);
```

```
daChild.InsertCommand.Parameters.Add("@t", SqlDbType.VarChar).Value = textBox2.Text;
```

```
daChild.InsertCommand.Parameters.Add("@n", SqlDbType.Int).Value = Int32.Parse(textBox3.Text);
```

```
daChild.InsertCommand.Parameters.Add("@c", SqlDbType.Int).Value = Int32.Parse(comboBox1.Text);
```

```
connection.Open();
```

```
daChild.InsertCommand.ExecuteNonQuery();
```

```
connection.Close();
```

```
MessageBox.Show("Added successfully!!");
```

Practical Exam

Form1

CONNECTParent Table - UsersUPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date02/03/2019ADD Post

Post TextThis is my first postUPDATE Post

Number of Shares2DELETE Post

User Id2

Form1

CONNECTParent Table - UsersUPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date02/03/2019ADD Post

Post TextThis is my first postUPDATE Post

Number of Shares2DELETE Post

User Id2

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
▶	2	3/23/2018	Post 2	5
*				

Added succesfully!!

OK

Form1

CONNECTParent Table - UsersUPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date02/03/2019ADD Post

Post TextThis is my first postUPDATE Post

Number of Shares2DELETE Post

User Id1

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
▶	2	3/23/2018	Post 2	5
	3	2/3/2019	This is my first post	2
*				

Practical Exam

UPDATE – for a selected child (a row from table Posts)

```
var selectedRowP = Int32.Parse(GridChild.SelectedRows[0].Cells[0].Value.ToString());
daChild.UpdateCommand = new SqlCommand("UPDATE Posts SET PoDate=@d, PoText=@t,
NoOfShares=@n, Uid=@c WHERE Poid=@pid", connection);
daChild.UpdateCommand.Parameters.Add("@d", SqlDbType.Date).Value = DateTime.Parse(textBox1.Text);
daChild.UpdateCommand.Parameters.Add("@t", SqlDbType.VarChar).Value = textBox2.Text;
daChild.UpdateCommand.Parameters.Add("@n", SqlDbType.Int).Value = Int32.Parse(textBox3.Text);
daChild.UpdateCommand.Parameters.Add("@c", SqlDbType.Int).Value = Int32.Parse(comboBox1.Text);
daChild.UpdateCommand.Parameters.Add("@pid", SqlDbType.Int).Value = selectedRowP;

connection.Open();
daChild.UpdateCommand.ExecuteNonQuery();
connection.Close();

MessageBox.Show("Updated succesfully!");
```

Practical Exam

Form1

CONNECT Parent Table - Users UPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date 05/05.2015 ADD Post

Post Text Update UPDATE Post

Number of Shares 5

User Id 1 DELETE Post

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
	2	3/23/2018	Post 2	5
▶	3	2/3/2019	This is my first post	2
*				

Form1

CONNECT Parent Table - Users UPDATE DB

	Uid	UName	UCity	UDob
	1	User 1	City Transaction 1	10/10/2000
▶	2	User 2	Bucuresti	2/2/1998
*				

Post Date 05/05.2015 ADD Post

Post Text Update UPDATE Post

Number of Shares 5

User Id 1 DELETE Post

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
	2	3/23/2018	Post 2	5
▶	3	2/3/2019	This is my first post	2
*				

Updated successfully!

OK

Form1

CONNECT Parent Table - Users UPDATE DB

	Uid	UName	UCity	UDob
▶	1	User 1	City Transaction 1	10/10/2000
	2	User 2	Bucuresti	2/2/1998
*				

Post Date 05/05.2015 ADD Post

Post Text Update UPDATE Post

Number of Shares 5

User Id 1 DELETE Post

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
	1	1/1/2018	Post 1	3
▶	3	5/5/2015	Update	5
*				

Practical Exam

DELETE – for a selected child (from the table Posts)

```
var selectedRowP = Int32.Parse(GridChild.SelectedRows[0].Cells[0].Value.ToString());  
daChild.DeleteCommand = new SqlCommand("DELETE FROM Posts WHERE Poid=@pid", connection);  
daChild.DeleteCommand.Parameters.Add("@pid", SqlDbType.Int).Value = selectedRowP;  
  
connection.Open();  
daChild.DeleteCommand.ExecuteNonQuery();  
connection.Close();  
  
MessageBox.Show("Deleted succesfully!");
```


Practical Exam

Form1

CONNECT

Parent Table - Users

UPDATE DB

	Uid	UName	UCity	UDob
▶	1	User 1	City Transaction 1	10/10/2000
	2	User 2	Bucuresti	2/2/1998
*				

Post Date

05/05.2015

ADD Post

Post Text

Update

UPDATE Post

Number of Shares

5

DELETE Post

User Id

1

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
	1	1/1/2018	Post 1	3
▶	3	5/5/2015	Update	5
*				

Form1

CONNECT

Parent Table - Users

UPDATE DB

	Uid	UName	UCity	UDob
▶	1	User 1	City Transaction 1	10/10/2000
	2	User 2	Bucuresti	2/2/1998
*				

Post Date

05/05.2015

ADD Post

Post Text

Update

UPDATE Post

Number of Shares

5

DELETE Post

User Id

1

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
	1	1/1/2018	Post 1	3
▶	3	5/5/2015	Update	5
*				

Deleted successfully!

OK

Form1

CONNECT

Parent Table - Users

UPDATE DB

	Uid	UName	UCity	UDob
▶	1	User 1	City Transaction 1	10/10/2000
	2	User 2	Bucuresti	2/2/1998
*				

Post Date

05/05.2015

ADD Post

Post Text

Update

UPDATE Post

Number of Shares

5

DELETE Post

User Id

1

Child Table - Posts

	Poid	PoDate	PoText	NoOfShares
▶	1	1/1/2018	Post 1	3
*				

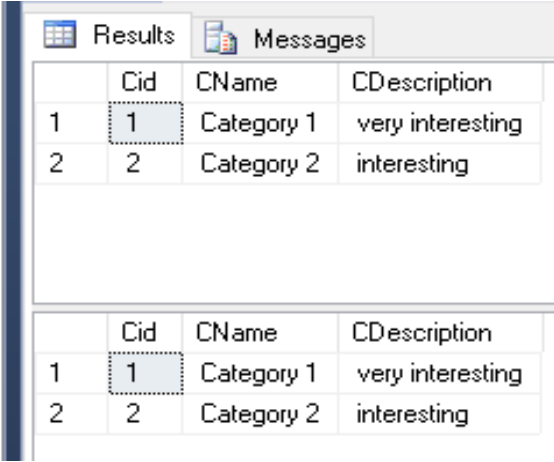
Practical Exam

3. Dirty reads

Transaction 1	Transaction 2																								
use MiniFacebook_IE Go BEGIN TRANSACTION UPDATE Categories SET CDescription='MY DESCRIPTION' WHERE Cid = 2 WAITFOR DELAY '00:00:10' ROLLBACK TRANSACTION	use MiniFacebook_IE Go SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED BEGIN TRAN select * from Categories WAITFOR DELAY '00:00:15' select * from Categories COMMIT TRAN																								
(1 row(s) affected)	<div><div>ResultsMessages</div><table><thead><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>MY DESCRIPTION</td></tr></tbody></table><div></div><table><thead><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>interesting</td></tr></tbody></table></div>		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	MY DESCRIPTION		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	interesting
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	MY DESCRIPTION																						
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	interesting																						

Practical Exam

Dirty reads - Solution

Transaction 1	Transaction 2
<pre>use MiniFacebook_IE Go BEGIN TRANSACTION UPDATE Categories SET CDescription='MY DESCRIPTION'WHERE Cid = 2 WAITFOR DELAY '00:00:10' ROLLBACK TRANSACTION</pre>	<pre>use MiniFacebook_IE Go --Solution: SET TRANSACTION ISOLATION LEVEL TO READ COMMITTED SET TRANSACTION ISOLATION LEVEL READ COMMITTED BEGIN TRAN select * from Categories WAITFOR DELAY '00:00:15' select * from Categories COMMIT TRAN</pre>
(1 row(s) affected)	

Practical Exam

Non-repeatable reads

Transaction 1	Transaction 2																								
use MiniFacebook_IE Go BEGIN TRAN WAITFOR DELAY '00:00:05' UPDATE Categories SET CDescription='DESCRIPTION NON REPEATABLE READS' WHERE Cid = 2 COMMIT TRAN	use MiniFacebook_IE Go SET TRANSACTION ISOLATION LEVEL READ COMMITTED BEGIN TRAN SELECT * FROM Categories WAITFOR DELAY '00:00:05' SELECT * FROM Categories COMMIT TRAN																								
(1 row(s) affected)	<div><div>ResultsMessages</div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>interesting</td></tr></table><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>DESCRIPTION NON REPEATABLE READS</td></tr></table></div>		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	interesting		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	DESCRIPTION NON REPEATABLE READS
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	interesting																						
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	DESCRIPTION NON REPEATABLE READS																						

Practical Exam

Non-repeatable reads - Solution

Transaction 1	Transaction 2																								
use MiniFacebook_IE Go BEGIN TRAN WAITFOR DELAY '00:00:05' UPDATE Categories SET CDescription='DESCRIPTION NON REPEATABLE READS' WHERE Cid = 2 COMMIT TRAN	use MiniFacebook_IE Go --Solution: SET TRANSACTION ISOLATION LEVEL TO REPEATABLE READ SET TRANSACTION ISOLATION LEVEL READ COMMITTED BEGIN TRAN SELECT * FROM Categories WAITFOR DELAY '00:00:05' SELECT * FROM Categories COMMIT TRAN																								
(1 row(s) affected)	<div><div>ResultsMessages</div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>interesting</td></tr></table><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>interesting</td></tr></table></div>		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	interesting		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	interesting
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	interesting																						
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	interesting																						

Practical Exam

Phantom reads

Transaction 1	Transaction 2																												
use MiniFacebook_IE Go BEGIN TRAN WAITFOR DELAY '00:00:05' INSERT INTO Categories(CName,CDescription) VALUES ('Phantom','Phantom description') COMMIT TRAN	use MiniFacebook_IE go SET TRANSACTION ISOLATION LEVEL REPEATABLE READ BEGIN TRAN SELECT * FROM Categories WAITFOR DELAY '00:00:05' SELECT * FROM Categories COMMIT TRAN																												
(1 row(s) affected)	<div><div>ResultsMessages</div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>DESCRIPTION NON REPEATABLE READS</td></tr></table><div></div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>DESCRIPTION NON REPEATABLE READS</td></tr><tr><td>3</td><td>3</td><td>Phantom</td><td>Phantom description</td></tr></table></div>		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	DESCRIPTION NON REPEATABLE READS		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	DESCRIPTION NON REPEATABLE READS	3	3	Phantom	Phantom description
	Cid	CName	CDescription																										
1	1	Category 1	very interesting																										
2	2	Category 2	DESCRIPTION NON REPEATABLE READS																										
	Cid	CName	CDescription																										
1	1	Category 1	very interesting																										
2	2	Category 2	DESCRIPTION NON REPEATABLE READS																										
3	3	Phantom	Phantom description																										

Practical Exam

Phantom reads - Solution

Transaction 1	Transaction 2																								
use MiniFacebook_IE Go BEGIN TRAN WAITFOR DELAY '00:00:05' INSERT INTO Categories(CName,CDescription) VALUES ('Phantom','Phantom description') COMMIT TRAN	use MiniFacebook_IE Go --Solution: Set transaction isolation level to SERIALIZABLE SET TRANSACTION ISOLATION LEVEL SERIALIZABLE BEGIN TRAN SELECT * FROM Categories WAITFOR DELAY '00:00:05' SELECT * FROM Categories COMMIT TRAN																								
(1 row(s) affected)	<div><div>ResultsMessages</div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>DESCRIPTION NON REPEATABLE READS</td></tr></table><div></div><table><tr><th></th><th>Cid</th><th>CName</th><th>CDescription</th></tr><tr><td>1</td><td>1</td><td>Category 1</td><td>very interesting</td></tr><tr><td>2</td><td>2</td><td>Category 2</td><td>DESCRIPTION NON REPEATABLE READS</td></tr></table></div>		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	DESCRIPTION NON REPEATABLE READS		Cid	CName	CDescription	1	1	Category 1	very interesting	2	2	Category 2	DESCRIPTION NON REPEATABLE READS
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	DESCRIPTION NON REPEATABLE READS																						
	Cid	CName	CDescription																						
1	1	Category 1	very interesting																						
2	2	Category 2	DESCRIPTION NON REPEATABLE READS																						

Practical Exam

Deadlock

<i>Transaction 1</i>	<i>Transaction 2</i>
use MiniFacebook_IE Go begin tran update Users set UCity='City Transaction 1' where Uid=1 -- this transaction has exclusively lock on table Users waitfor delay '00:00:10' update Categories set CName='Category Transaction 1' where Cid=1 commit tran	use MiniFacebook_IE Go begin tran update Categories set CName='Category Transaction 2' where Cid=1 -- this transaction has exclusively lock on table Categories waitfor delay '00:00:10' update Users set UCity='City Transaction 2' where Uid=1 commit tran
(1 row(s) affected)	(1 row(s) affected)
(1 row(s) affected)	Msg 1205, Level 13, State 51, Line 12 Transaction (Process ID 56) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.

Practical Exam

Deadlock

- select * from Categories
- select * from Users

Results		Messages		
	Cid	CName	CDescription	
1	1	Category Transaction 1	very interesting	
2	3	Phantom	Phantom description	

	Uid	UName	UCity	UDob
1	1	User 1	City Transaction 1	2000-10-10
2	2	User 2	Bucuresti	1998-02-02

Practical Exam

Deadlock - Solution

Transaction 1	Transaction 2
<pre>use MiniFacebook_IE Go begin tran update Users set UCity='City Transaction 1' where Uid=1 -- this transaction has exclusively lock on table Users waitfor delay '00:00:10' update Categories set CName='Category Transaction 1' where Cid=1 commit tran</pre>	<pre>use MiniFacebook_IE Go SET DEADLOCK_PRIORITY HIGH begin tran update Categories set CName='Category Transaction 2' where Cid=1 -- this transaction has exclusively lock on table Categories waitfor delay '00:00:10' update Users set UCity='City Transaction 2' where Uid=1 commit tran</pre>
<p>(1 row(s) affected)</p> <p>Msg 1205, Level 13, State 51, Line 12</p> <p>Transaction (Process ID 56) was deadlocked on lock resources with another process and has been chosen as the deadlock victim. Rerun the transaction.</p>	<p>(1 row(s) affected)</p> <p>(1 row(s) affected)</p>

Practical Exam

Deadlock – Solution

- select * from Categories
- select * from Users

Results		Messages		
	Cid	CName	CDescription	
1	1	Category Transaction 2	very interesting	
2	3	Phantom	Phantom description	

	Uid	UName	UCity	UDob
1	1	User 1	City Transaction 2	2000-10-10
2	2	User 2	Bucuresti	1998-02-02