

# Визуал програмчлал

Ғ.ІТ203

Лекц-8

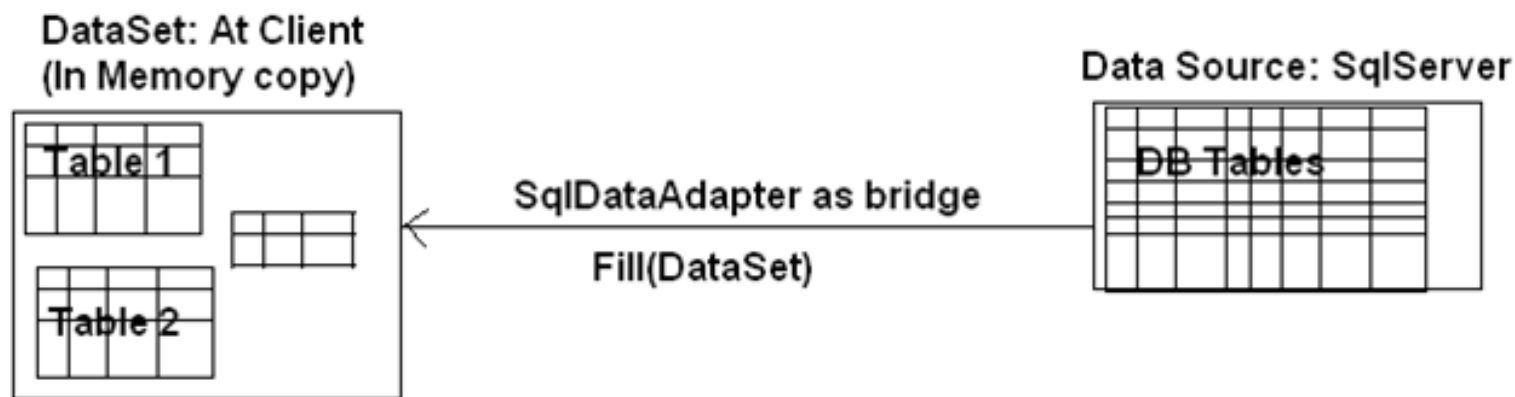
# Агуулга

- ADO.NET загварууд
  - Салангид загвар
  - Уяатай загвар
- Dataset, DataAdapter ашиглах
- Тайлан гаргах

# Салангид загвар

- SqlConnection con;
- SqlCommand cmd;
- SqlDataReader dr;

- Зөвхөн өгөгдлийн эх үүсвэрээс унших, бичих үед л холболт хийж бусад тохиолдолд салангид ажиллана.

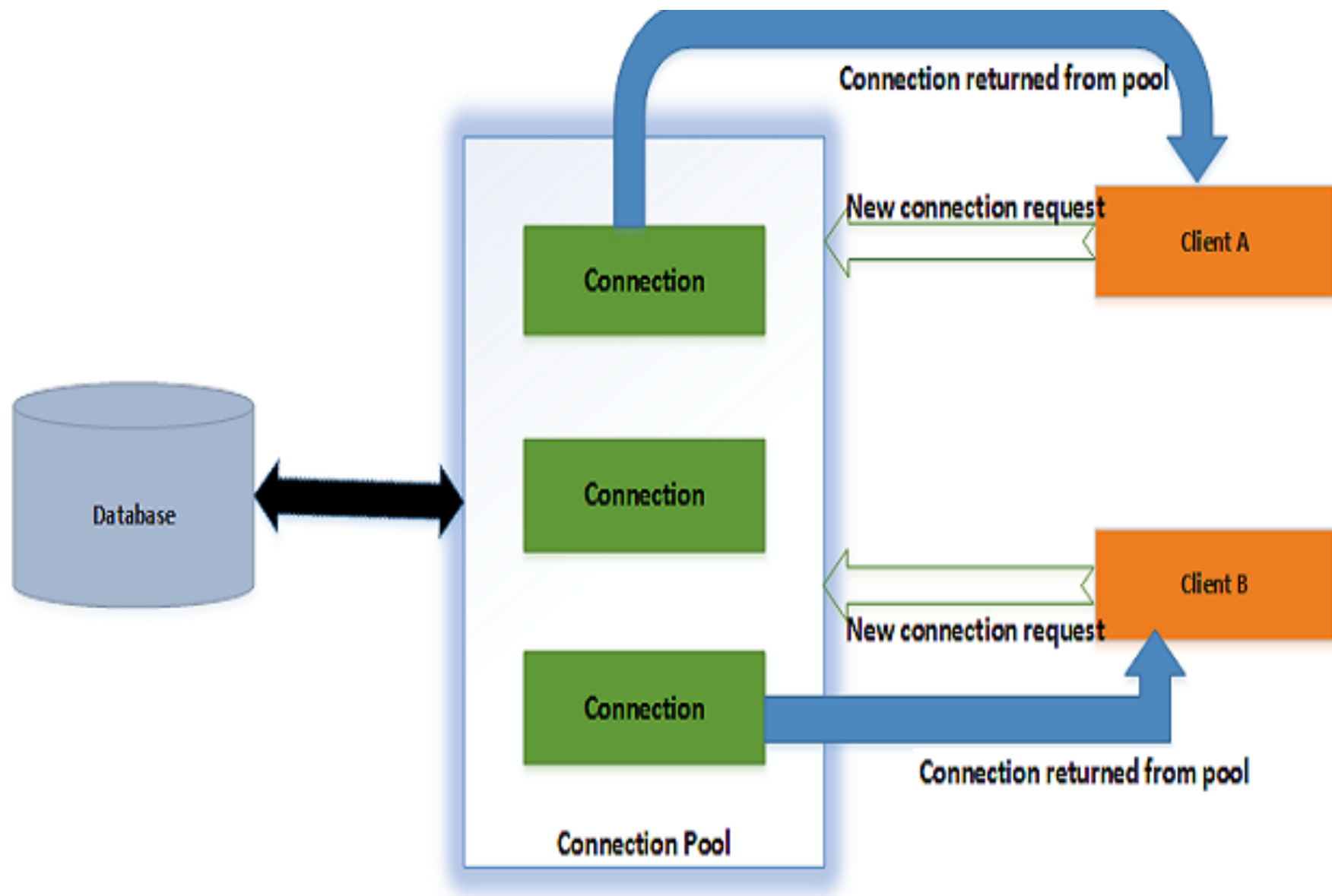


- Серверийн санах ой, холболт, боловсруулалт зэрэг нөөцүүдийг хэмнэнэ.

# Уяатай загвар

- Өгөгдлийн сан руу холболт үүсгээд түүгээр дамжуулан унших, засах үйлдлүүдийг идэвхтэй гүйцэтгэнэ. Холболт нь үйлдлүүд дуусах хүртэл нээлттэй байдаг.
- Өгөгдөл нийлүүлэгч бүрт зохицсон холбогч классууд бий болно.
  - `IDbConnectioninterface` хангасан классууд
- Холбогч мөр нь өгөгдлийн эх үүсвэр рүү хандахад шаардлагатай хэрэглэгчийн нэр, нууц үг зэрэг мэдээллийг агуулна.

```
SqlConnectionStringBuilder scb= new SqlConnectionStringBuilder();  
Scb.DataSource = "MyServer";  
Scb.Password = "qwerty1234";  
Scb.UserID = "filmadmin";  
Scb.InitialCatalog = "films";  
SqlConnection conn = new SqlConnection(scb.ConnectionString);
```



# Жишээ

```
using (SqlConnection connection = new SqlConnection(
    "Integrated Security=SSPI;Initial Catalog=Northwind"))
{
    connection.Open();
    // Pool A is created.
}

using (SqlConnection connection = new SqlConnection(
    "Integrated Security=SSPI;Initial Catalog=pubs"))
{
    connection.Open();
    // Pool B is created because the connection strings differ.
}

using (SqlConnection connection = new SqlConnection(
    "Integrated Security=SSPI;Initial Catalog=Northwind"))
{
    connection.Open();
    // The connection string matches pool A.
}
```

## Жишээ

```
cnStr= “Server=MYSERVER; Trusted_Connection=  
yes; database=films; connection  
reset=false;”+”connection lifetime=60;”+”min pool  
size = 1; max pool size = 50”;  
SqlConnectionconn = new SqlConnection(cnStr);
```

# Query дотор Параметр ашиглах

```
SqlConnection conn = new SqlConnection(connStr);
conn.Open();
String sql = "insert into movies(movie Title,movie_Year,
movie_Director) values (@title, @yr, @bestpic)";
SqlCommand cmd = new SqlCommand();
cmd.Connection = conn;
cmd.CommandText = sql;
cmd.Parameters.AddWithValue("@title", "Schindler's List");
cmd.Parameters.AddWithValue("@yr", "1993");
cmd.Parameters.AddWithValue("@bestpic", "Y");
```



# Командыг биелүүлэх

Method	Тайлбар
ExecuteNonQuery	<p>Асуулгыг биелүүлж, уг асуулгаар өөрчлөлт орсон мөрийн тоог буцаана.</p> <pre>cmd.CommandText = "DELETE movies WHERE movie_ID =220"; Int ct = cmd.ExecuteNonQuery();</pre>
ExecuteReader	<p>Асуулгыг биелүүлж үр дүнгийн олонлогыг DataReader объектоор буцаана.</p> <pre>cmd.CommandText = "SELECT * FROM movies WHERE movie_year&gt;'1945'"; SqlDataReader rdr = cmd.ExecuteReader();</pre>

# Командыг биелүүлэх

Method	Тайлбар
ExecuteScalar	<p>Асуулгыг биелүүлж, үр дүнгийн олонлогоос эхний баганы эхний мөрөнд байгаа утгыг буцаана.</p> <pre>cmd.CommandText = "SELECT COUNT(movie_title) FROM movies"; int movieCt = (int) cmd.ExecuteScalar();</pre>
ExecuteXmlReader	<p>Зөвхөн SQL Server-н хувьд хэрэглэгдэнэ. Үр дүнгийн олонлогтой ажиллах XmlReader объектыг буцаана.</p>

# Командыг биелүүлэх-ExecuteReader

```
rdr= cmd.ExecuteReader(sql,  
CommandBehavior.SingleResult);
```

SingleRow	Асуулгаар нэг мөр буцаахыг заана.
SingleResult	Асуулгаар нэг утга буцаахыг заана.
KeyInfo	Түлхүүр баганын тухай мэдээлэл
SchemaOnly	Багануудын нэрсийг буцаана. <code>dr = cmd.ExecuteReader(CommandBehavior.SchemaOnly);</code> <code>String col1 = dr.GetName(0);</code>

# Жишээ

```
string q1 = "SELECT * FROM movies WHERE movie_year<1940";
string q2 = "SELECT * FROM movies WHERE movie_year>1980";
DbDataReader rdr= cmd.ExecuteReader();
Bool readNext= true;
while(readNext){
    while (rdr.Read()){
        MessageBox.Show(rdr.GetString(1));
    }
    readNext= rdr.NextResult();
}
rdr.Close();
```

# Баганы утгыг унших

- Тухайн нэг мөрийн баганы утгуудыг GetName, GetValue, GetString, GetInt32, GetdateTime, GetDouble баганы дугаарыг зааж уншиж болно.

```
cmd.CommandText= "SELECT movieID, movieTitleFROM  
movies";
```

```
rdr= cmd.ExecuteReader();
```

```
rdr.ExecuteReader();
```

```
rdr.Read();
```

```
string title;
```

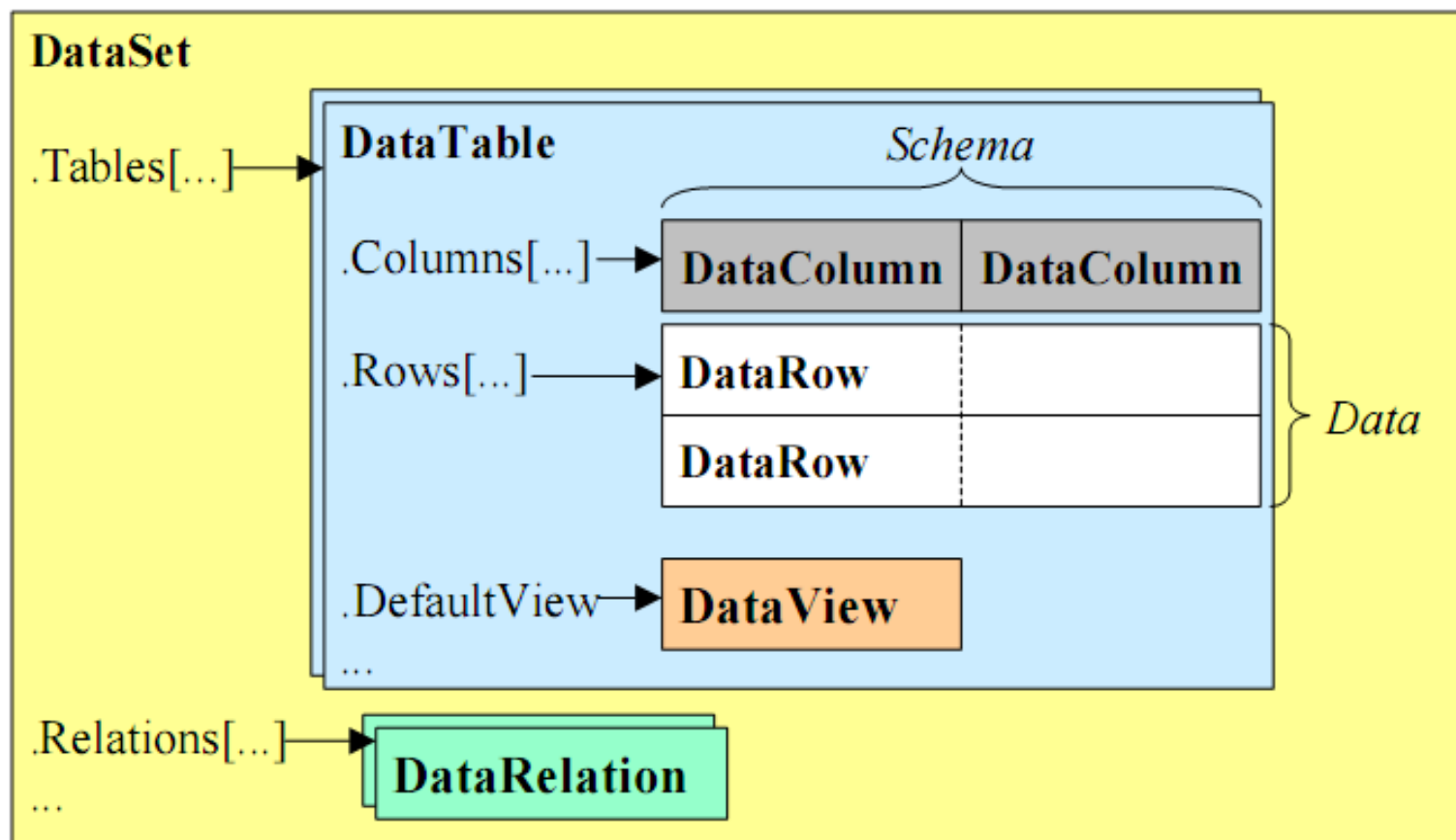
```
title =rdr.GetString(1);  
title = (string)rdr.GetSqlString(1);  
title = (string)rdr.GetValue(1);  
title = (string)rdr["movieTitle"];  
title = (string)rdr[1];
```

# Өгөгдлийн сангийн процедур

- Нэг нэрийн дор өгөгдлийн санд хадгалсан SQL кодын дараалал.
- Програмууд нэг логикуыг хамтран эзэмших, битүүмжлэх, дахин ашиглах
- ADO.NET нь OleDb, SqlClient, ODBC, OracleClient өгөгдөл нийлүүлэгчдийн процедурыг ажиллуулж чадна.

```
cmd.CommandText = "sp_AddMovie";  
cmd.CommandType = CommandType.StoredProcedure;  
cmd.ExecuteNonQuery();
```

# DataSet-ийн бүтэц



# DataTable-тэй ажиллах

```
DataTable dt = rdr.GetSchemaTable();
int ict = 0;
foreach(DataRow r in dt.Rows){
    foreach(DataColumn c in dt.Columns){
        Console.WriteLine(ict.ToString()+c.ColumnName+": " +r[c]);
        ict++;
    }
}
```



# DataSetрүү өгөгдөл ачаалах

Өгөгдлийн сангаас хүснэгтийг автоматаар дүүргэх

```
cmd.CommandText = "SELECT * FROM movies WHERE  
movie_Year < 1945";  
DBDataReader rdr =  
cmd.ExecuteReader(CommandBehavior.CloseConnection);  
dt.Load(rdr);  
Console.WriteLine(dt.Rows[0]["movie_Title"]);  
Console.WriteLine(rdr.IsClosed);
```

# DataSet ба DataAdapter

- Байгаа хүснэгт эсвэл шинэ хүснэгт үүсгэж өгөгдлөөр дүүргэхэд DataAdapter-ийг ашиглаж болно.
- Эхлээд DataAdapter объектийг үүсгэх хэрэгтэй.
- DataAdapter объектийн Fill методоор өгөгдлийг хүснэгт рүү ачааллана.

# DataSet ба DataAdapter

- Data Adapter өгөгдлийг хүснэгт рүү ачаалласны дараагаар холболтыг хаадаг.
- Үүнээс хойших бүхөөрчлөлтүүд нь DataSet дээр хийгдэнэ.
- Анхны эх үүсвэр рүү өөрчлөлтийг илгээхийн тулд холболтыг дахин сэргээж өөрчлөгдсөн мөрүүдийг DataSource–рууupdate() методыг ашиглан илгээнэ.

# DataSet ба DataAdapter

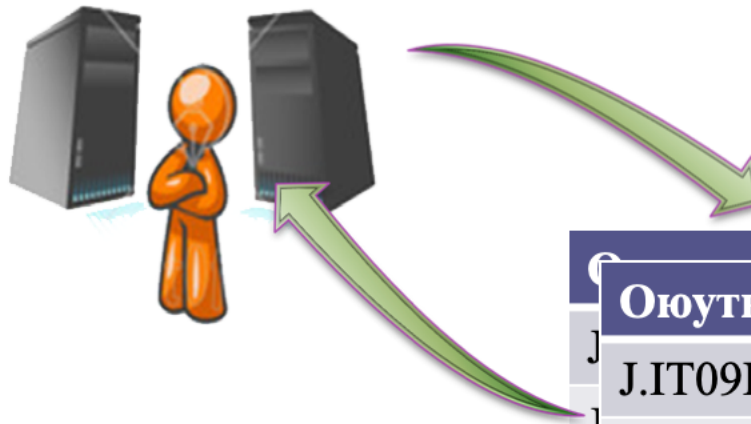
- DataAdapter ашиглах нь өгөгдлийн санд засвар хийх ажлыг автоматжуулж өгдөг.
- Гэхдээ үүний цаана олон хэрэглэгч өөрчлөлт хийх үед асуудал гардаг.
- Учир нь DataSet дахь өөрчлөлт нь санах ойд хийгдэх ба тухайн өөрчлөлтийг жинхэнэ datasource – руу илгээх үед тухайн өгөгдлийг өөр хэрэглэгч өөрчилсөн байж болно.

# DataSet ба DataAdapter

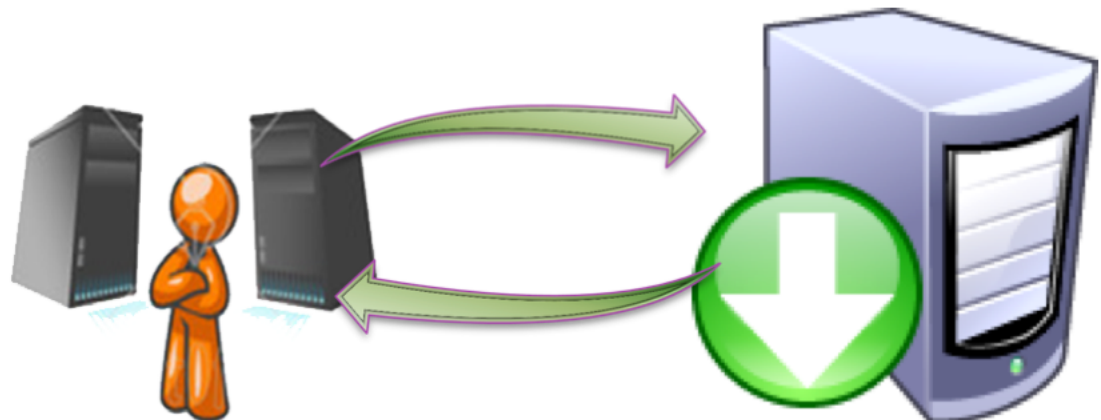
- Update метод нь өмнөх уншилт дээр үндэслэслэж өөрчлөлт хийгдсэнийг мэдэх чадвартай байдаг бөгөөд өөрчлөлт хийгдсэн мөрүүдийг шинэчлэхийг зогсоодог.
- **Энэхүү зэрэгцэн засвар хийх нөхцөлөөс үүсэх**
  1. алдааг зөрчил үүссэн бичлэгүүдийг буцаах
  2. алдаа гаргахгүй өөрчлөлтүүдийг гүйцэтгэх хоёр замаар шийдэж болно.

# DataSet ба DataAdapter

Оюутны код	GPA
J.IT09D001	3.2
J.IT09D002	3.7



Оюутны код	GPA
J.IT09D001	3.2
J.IT09D002	3.9



# DataSet ба DataAdapter

- Энэ өөрчлөлтийг хийхдээ Transaction үүсгэж үйлдлийг буцаана.
- Үйлдлийг буцаахдаа Rollback() методыг ашигладаг.
- Үгүй бол Commit() методыг дуудаж багц өөрчлөлтийг хийж болно.

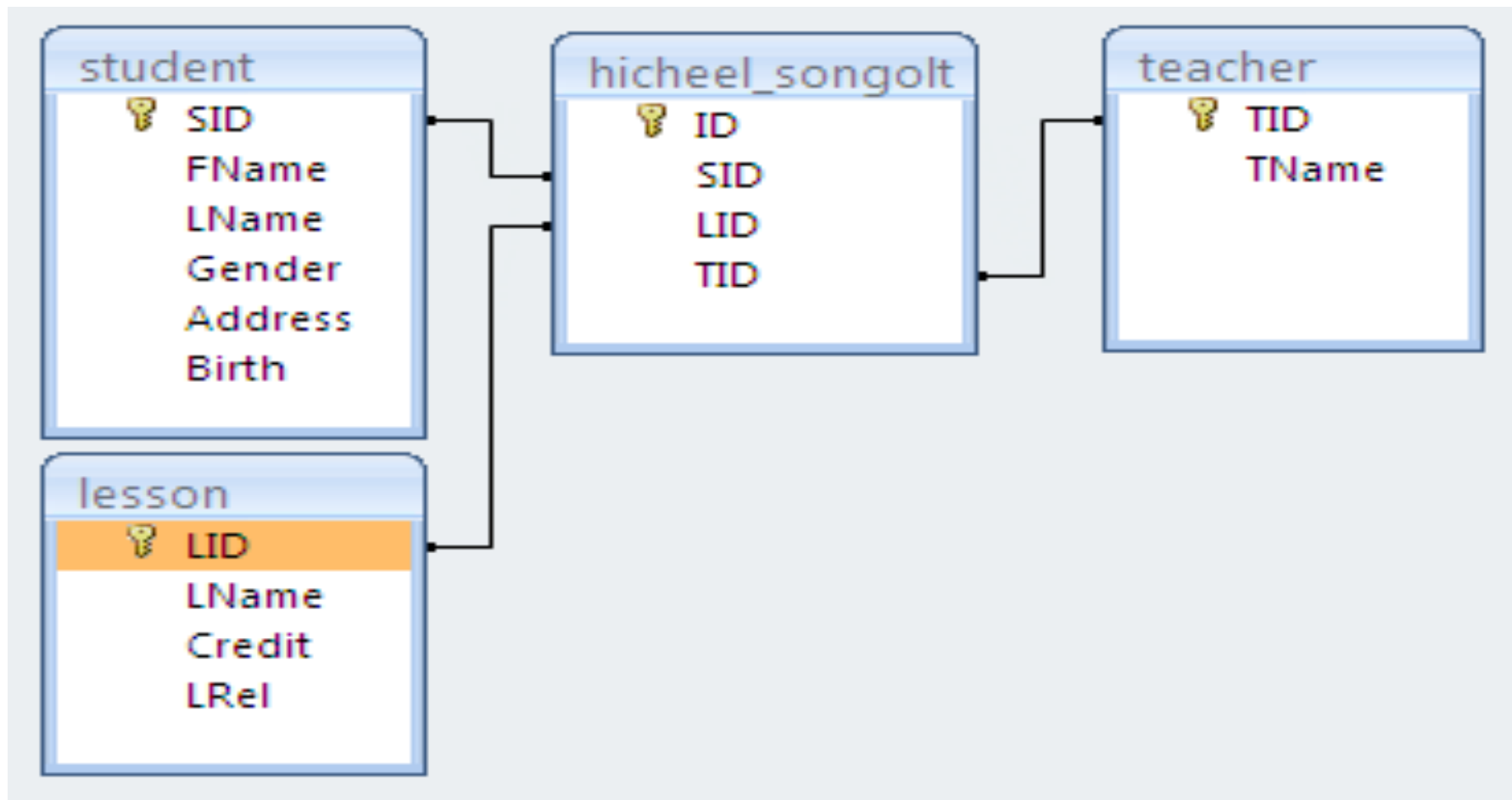
# Хүснэгтүүдийн хооронд холбоос үүсгэх

- DataRelation нь DataTable хүснэгтүүдийн хооронд холбоос үүсгэдэг. Холбоос үүсгэхдээ хоёр хүснэгтүүдийн багануудыг холбож холбоос үүсгэдэг.
- Эдгээр баганууд нь ижил болон үл давхцах утгатай байх ёстой. DataSet-д хүснэгтүүдийн хамаарлыг тодорхойлсон Relations объекттой.

```
ds.Relations.Add("show",  
    ds.Tables["student"].Columns["SID"],  
    ds.Tables["hicheel_songolt"].Columns["  
SID"]);
```



# Хүснэгтүүдийн хооронд холбоос үүсгэх



## Хүснэгтүүдийн хооронд холбоос үүсгэх

```
BindingSource masterBindingSource = new  
    BindingSource();
```

```
BindingSource detailsBindingSource =  
    new BindingSource();
```

```
dataGridView1.DataSource =  
    masterBindingSource;
```

```
dataGridView2.DataSource =  
    detailsBindingSource;
```

## Хүснэгтүүдийн хооронд холбоос үүсгэх

```
OdbcConnection conn = new  
    OdbcConnection("FIL=MS Access;  
    DSN=UNIV");  
conn.Open();  
OdbcDataAdapter da = new  
    OdbcDataAdapter();  
OdbcCommand command =  
    conn.CreateCommand();
```

## Хүснэгтүүдийн хооронд холбоос үүсгэх

```
DataSet ds = new DataSet();  
  
string str_select = "select * from  
    student";  
command.CommandText = str_select;  
da.SelectCommand = command;  
  
da.Fill(ds, "student");
```

## Хүснэгтүүдийн хооронд холбоос үүсгэх

```
str_select = "select * from  
hicheel_songolt";  
command.CommandText =  
str_select;
```

```
da.SelectCommand = command;  
da.Fill(ds, "hicheel_songolt");
```

## Хүснэгтүүдийн хооронд холбоос үүсгэх

```
masterBindingSource.DataSource = ds;  
masterBindingSource.DataMember =  
    "Student";
```

```
detailsBindingSource.DataSource =  
    masterBindingSource;  
detailsBindingSource.DataMember =  
    "show";
```

# Тайлан



# Бүртгэл

Сургалтын алба

Оюутан

Багш

Хичээл

Хичээл сонголт

Бүртгэл

Оюутны код

Овог

Нэр

Хүйс

☐ эр

☐ эм

Гэрийн хаяг

Төрсөн өдөр

80.01.01

Хайх

Бүртгэх

Засах

Устгах

Оюутны мэдээлэл

Эхний бичлэг

Өмнөх бичлэг

Дараагийн бичлэг

Төгсгөлийн бичлэг

Харах



# Тайлан

DataSet классын relation-ийг ашиглан хүснэгтүүдийг хооронд нь холбож тухайн холбосон хүснэгтэн дээр шүүлтүүр хийж ихэвчлэн тайланг гаргадаг. Жишээ нь

- Хичээл сонголт хийсэн оюутны нэрсийг ангиар шүүж харах
- Голч оноо нь тухайн өгсөн утгаас дээш доош оюутнуудын нэрсийг мэргэжилээр шүүж харуулах
- Мөн тухайн өгөгдлийн эзлэх хувийг графикаар харуулна.

# Тайлан

Main Report

PRODUCT ID	PRODUCT NAME	PRODUCT PRIC
1	Product1	10.00
2	Product2	
3	Product3	
4	Product4	
5	Product5	

## World Sales Report

5/11/2004 1:06:00PM



## Top 5 Countries' Sales

