Database and Information Systems

Michal Krátký

Department of Computer Science
Faculty of Electrical Engineering and Computer Science
VŠB – Technical University of Ostrava

2020/2021

Content

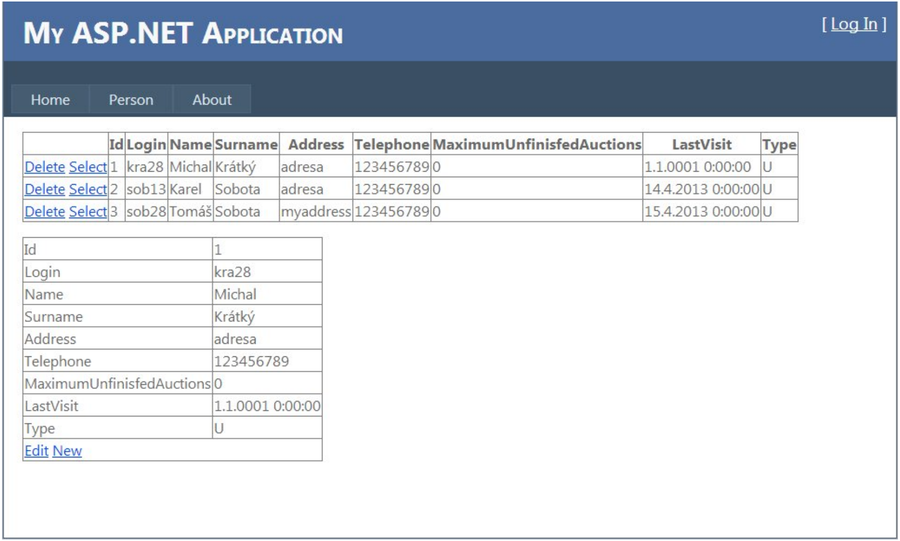
1 Pattern 1: Disabling the Update of Primary Key

2 Pattern 2: 1:1 Relationship - Reading, Updating

3 Pattern 3: Many-Attributes Table

4 Pattern 4: Form Parameters

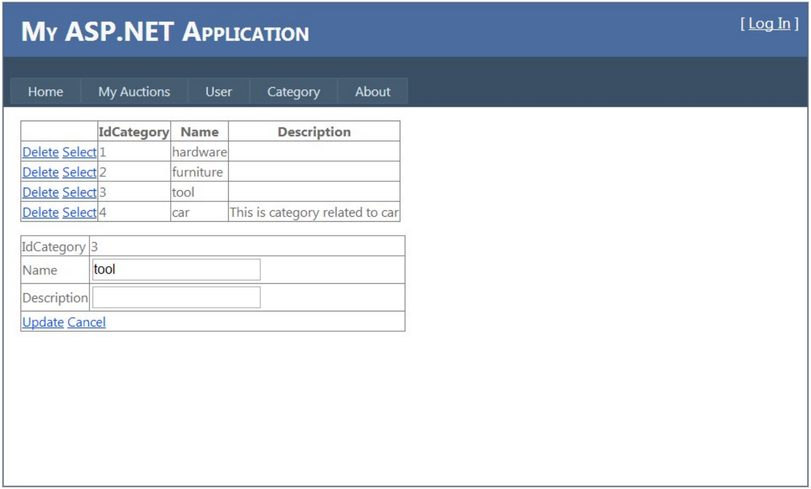
Example, ORM



	Id	Login	Name	Surname	Address	Telephone	MaximumUnfinisfedAuctions	LastVisit	Type
Delete Select	1	kra28	Michal	Krátký	adresa	123456789	0	1.1.0001 0:00:00	U
Delete Select	2	sob13	Karel	Sobota	adresa	123456789	0	14.4.2013 0:00:00	U
Delete Select	3	sob28	Tomáš	Sobota	myaddress	123456789	0	15.4.2013 0:00:00	U

Id	1
Login	kra28
Name	Michal
Surname	Krátký
Address	adresa
Telephone	123456789
MaximumUnfinisfedAuctions	0
LastVisit	1.1.0001 0:00:00
Type	U
Edit New	

Disabling the Update of Primary Key – Update form



	IdCategory	Name	Description
Delete Select	1	hardware	
Delete Select	2	furniture	
Delete Select	3	tool	
Delete Select	4	car	This is category related to car

IdCategory	3
Name	tool
Description	
Update Cancel	

Data Layer Issues

Example: Form/Category.aspx

- This is a primary key with IO IDENTITY in MS SQL or Oracle:

```
CREATE TABLE Category (
  idCategory       INTEGER PRIMARY KEY IDENTITY,
  name                         VARCHAR(30) NOT NULL,
  description      VARCHAR(100));
```

- In the case of ORM, the primary key is not set in the insert statement:

```
public String SQL_INSERT =
 "INSERT␣INTO␣Category␣VALUES(@name,␣@description)";
```

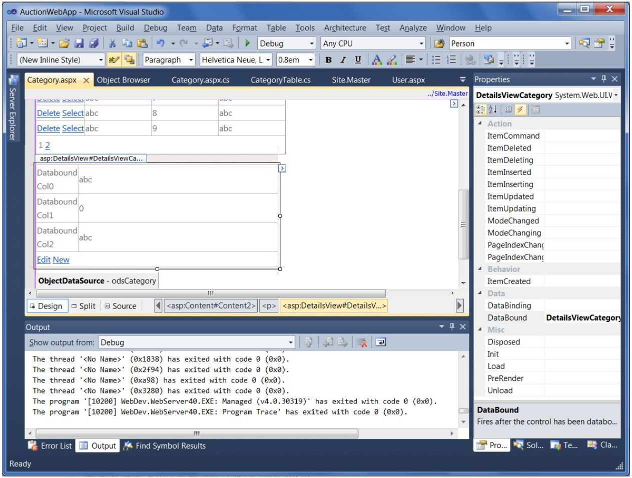
Presentation Layer Issues 1/3

- Since we use AutoGenerateRows="true" in the GridView, all attributes of the domain object Category are shown.
- In the case of the Category.aspx file, add the name of the method invoked after the DetailsView component is bound with its data source:

```
<asp:DetailsView ID="DetailsViewCategory" ...
    OnDataBound="DetailsViewCategory_OnDataBound">
```

Presentation Layer Issues 2/3

- Or, you can use the `Properties` window, the sheet `Events`:



DetailsViewCategory System.Web.UI.V

DataBound: DetailsViewCategory

DataBound
Fires after the control has been databo...

Show output from: Debug

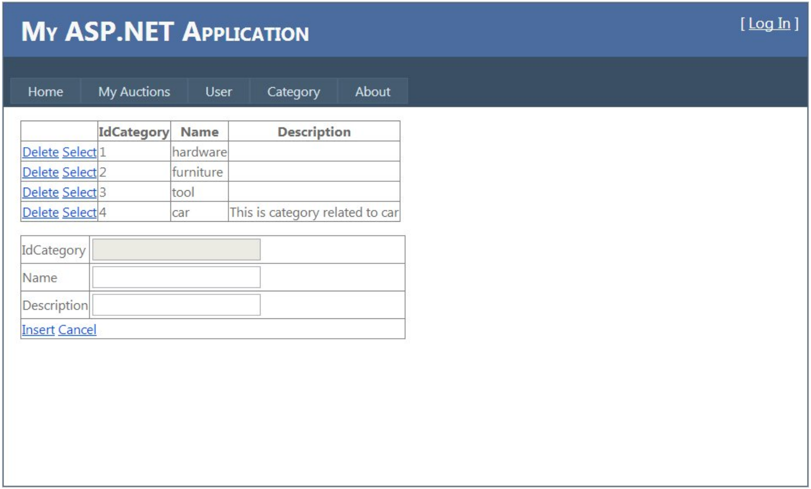
```
The thread '<No Name>' (0x1838) has exited with code 0 (0x0).
The thread '<No Name>' (0x2f94) has exited with code 0 (0x0).
The thread '<No Name>' (0xa98) has exited with code 0 (0x0).
The thread '<No Name>' (0x3280) has exited with code 0 (0x0).
The program '[10200] WebDev.WebServer40.EXE: Managed (v4.0.30319)' has exited with code 0 (0x0).
The program '[10200] WebDev.WebServer40.EXE: Program Trace' has exited with code 0 (0x0).
```

Presentation Layer Issues 3/3

- In the case of the Category.aspx.cs file, add the definition of DetailsViewCategory_OnDataBound:

```
protected void DetailsViewCategory_OnDataBound(
  object sender, EventArgs e)
{
  // The row of the table is hidden
  // DetailsViewCategory.Rows[0].Visible = false;
  // The row of the table is read only
  DetailsViewCategory.Rows[0].Enabled = false;
}
```

Disabling the Update of Primary Key - Insert Form

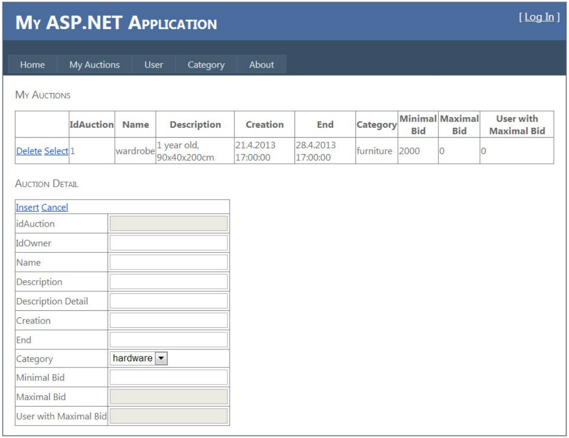


	IdCategory	Name	Description
Delete Select	1	hardware	
Delete Select	2	furniture	
Delete Select	3	tool	
Delete Select	4	car	This is category related to car

IdCategory	
Name	
Description	
Insert Cancel	

My Auctions Form 1/3

Pattern 2, 1:1 Relationship, Reading: We want to show **the category name** instead of its id in the list of my auctions (the 1:1 relationship).



My Auctions

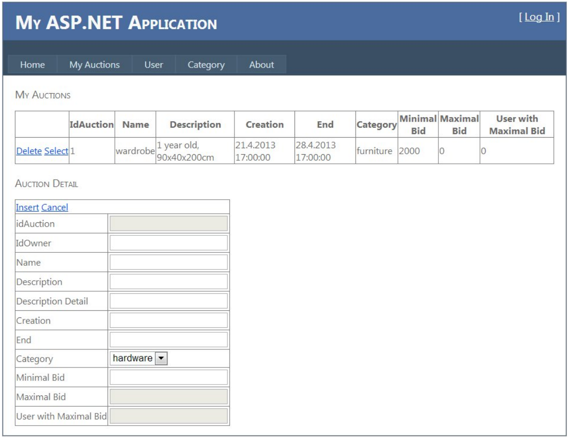
	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

Auction Detail

Insert Cancel	
idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

My Auctions Form 2/3

Pattern 2, 1:1 Relationship, Updating: We want to show a list of all categories when a new auction is created (the 1:1 relationship).



My Auctions

	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

Auction Detail

Insert Cancel

idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

Data Layer Issues 1/6

Example: Form/Auction.aspx

- The category attribute includes the foreign key:

```
CREATE TABLE Auction (
  idAuction             INTEGER PRIMARY KEY IDENTITY,
  owner                 INTEGER NOT NULL REFERENCES "User",
  name                  VARCHAR(20) NOT NULL,
  description           VARCHAR(100) NOT NULL,
  description_detail    VARCHAR(2000),
  creation              DATETIME NOT NULL,
  "end"                 DATETIME NOT NULL,
  category              INTEGER NOT NULL REFERENCES Category,
  min_bid_value         INTEGER,
  max_bid_value         INTEGER,
  max_bid_user          INTEGER REFERENCES "User");
```

Data Layer Issues 2/6 – Domain Object

Since we want to show the category name instead of category id, the domain object includes both attributes: `int mIdCategory` as well as `Category mCategory`.

```
public class Auction {
  private int mIdAuction;
  private int mIdOwner;
  private String mName;
  private String mDescription;
  private String mDescriptionDetail;
  private DateTime mCreation;
  private DateTime mEnd;
  private int mIdCategory;      // id of the category
  private Category mCategory;   // the complete record
  private int mMaxBidValue;
  private int mMinBidValue;
  private int mIdMaxBidUser;
        ...
```

Data Layer Issues 3/6 – DAO Variant I – Negative

When a `Select` method of `AuctionTable` reads values from the result of a query we must read the record of the related category.

```
private Collection<Auction> Read(SqlDataReader reader) {
  Collection<Auction> auctions = new Collection<Auction>();
  while (reader.Read()) {
    Auction auction = new Auction();
    auction.IdAuction = reader.GetInt32(0);
    ...
    auction.IdCategory = reader.GetInt32(6);
    // read the record with the 1:1 relationship
    auction.Category =
      new CategoryTable().Select(auction.IdCategory);
    ...
    auctions.Add(auction);
  }
  return auctions;
}
```

Data Layer Issues 4/6 – DAO - Analysis

- Let us go analyse the efficiency of this technique.
- If a table includes 10,000 records, we execute the following number of queries to obtain the category name:
    - 1 query: `SELECT * FROM Auction`
    - 10,000 queries: `SELECT * FROM Category WHERE id=?`
- Since the capacity of a network is up to 3,000 queries/s, **this technique is inefficient**.
- **ORM Rule 1: Minimize the number of operations sent to a DBMS**.

Data Layer Issues 5/6 – DAO - Variant II 1/2

Therefore we use only 1 query including a join in DAO (instead, e.g. 10,001 queries):

```
public String SQL_SELECT = "SELECT␣a.idAuction,a.name,␣" +
 "a.description,a.creation,a.\"end\",a.category," +
 "a.min_bid_value,a.max_bid_value,␣a.max_bid_user,␣" +
 "c.name␣FROM␣Auction␣a,␣Category␣c␣" +
 "WHERE␣a.category=c.idCategory";

public String SQL_SELECT_ID = "SELECT␣a.*,c.name␣" +
 "FROM␣Auction␣a,␣Category␣c␣WHERE␣a.idAuction=@idAuction␣" +
 "and␣a.category=c.idCategory";
```

Data Layer Issues 6/6 – DAO - Variant II 2/2

```
private Collection<Auction> Read(SqlDataReader reader,
 bool complete) {
 Collection<Auction> auctions = new Collection<Auction>();
 while (reader.Read()) {
  Auction auction = new Auction();
  int i = 0;
  auction.IdAuction = reader.GetInt32(i++);
  ...
  auction.IdCategory = reader.GetInt32(i++);
  auction.Category = new Category();
  auction.Category.IdCategory = auction.IdCategory;
  ...
  auction.Category.Name = reader.GetString(i++);
  auctions.Add(auction);
 }
 return auctions;}
```

Presentation Layer Issues – GridView

```
 1 <asp:GridView ID="GridViewAuction" runat="server"
 2     DataKeyNames="IdAuction"
 3     AutoGenerateColumns="False"
 4     AllowPaging="True" DataSourceID="odsAuction">
 5  <Columns>
 6   <asp:CommandField ShowSelectButton="True" ... />
 7   <asp:TemplateField HeaderText="Category">
 8    <ItemTemplate>
 9     <asp:Label ID="LabelCategory" runat="server"
10      Text='<%# Eval("Category.Name") %>'></asp:Label>
11    </ItemTemplate>
12   </asp:TemplateField>
13   ...
```

Lines 7–12: Instead of showing the category id, we define a column showing the value of the attribute `Name` of the attribute `Category` of the domain object `Auction` (`AutoGenerateColumns='False'` is used).

Presentation Layer Issues – DetailsView 1/2

```
 1 <asp:DetailsView ID="DetailsViewAuction" runat="server"
 2   AutoGenerateRows="false" DataSourceID="odsAuctionDetail"
 3   DataKeyNames="idAuction" ... >
 4   <Fields>
 5    <asp:CommandField ShowInsertButton="True"/>
 6            ...
 7    <asp:TemplateField HeaderText="Category">
 8    <ItemTemplate>
 9     <asp:Label ID="LabelCategory" runat="server"
10              Text='<%# Eval("Category.Name") %>'></asp:Label>
11    </ItemTemplate>
```

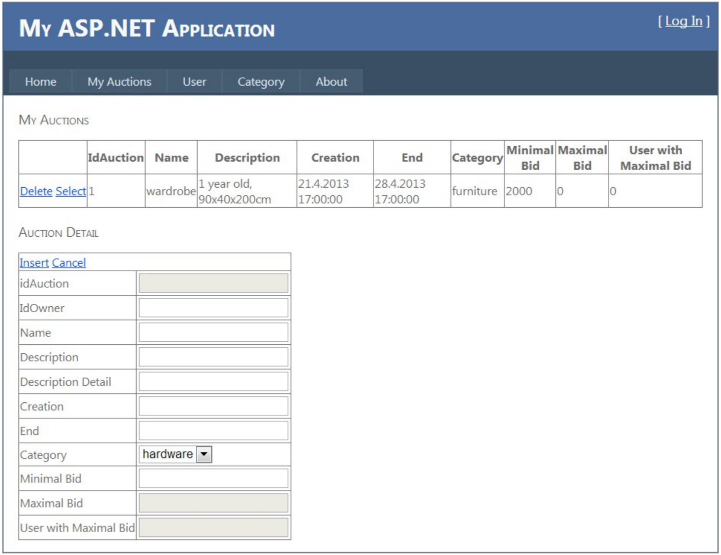
Lines 8–11: In the read only view of `DetailsView`, show the value of the attribute `Name` of the attribute `Category` of the domain object `Auction` (again `AutoGenerateColumns='false'` is used).

Presentation Layer Issues – DetailsView 2/2

```
1   <EditItemTemplate>
2    <asp:DropDownList ID="ListCategory" runat="server"
3           DataSourceID="odsCategory"
4     DataTextField="Name" DataValueField="IdCategory"
5     SelectedValue='<%# Bind("IdCategory") %>'>
6    </asp:DropDownList>
7   </EditItemTemplate>
8 </asp:TemplateField>
9  ...
```

Lines 2–7: In the edit mode of `DetailsView` (attribute `EditItemTemplate`), we show a list of all records from the `Category` table, `Name` is shown, however `IdCategory` is selected. This value is automatically inserted in the instance of the domain object passed to methods of DAO, e.g. Insert, Update, ...

My Auctions Form



MY AUCTIONS

	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

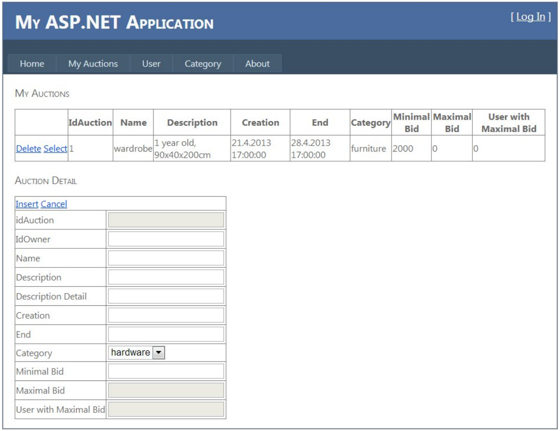
AUCTION DETAIL

Insert Cancel

idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

My Auctions Form

Pattern 3: We want to show only values of some attributes, e.g. `description_detail VARCHAR(2000)` is not shown in the list of my auctions (a many-attributes table).



My Auctions

	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

Auction Detail

Insert Cancel	
idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware ▾
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

Why?

- Let go analyse the efficiency of this technique.
- We consider a table with 10,000 records, the average size of `description_details` is 1,000B.
- If we read all complete records, it is 9,5MB for this attribute although it is not shown to a user.
- **ORM Rule 2: Minimize the volume of data retrieved from a DBMS**.
  - A similar rule should be defined for the update operation.
- **ORM Rule 3: Retrieve only data to be shown to a user**.

Data Layer Issues 1/4

Example: Altough the `Auction` table includes many attributes, in the list of all auctions, we do not want to show some of them, e.g. `description_detail VARCHAR(2000)`

```
CREATE TABLE Auction (
 idAuction            INTEGER PRIMARY KEY IDENTITY,
 owner                INTEGER NOT NULL REFERENCES "User",
 name                 VARCHAR(20) NOT NULL,
 description          VARCHAR(100) NOT NULL,
 description_detail   VARCHAR(2000),
 creation             DATETIME NOT NULL,
 "end"                DATETIME NOT NULL,
 category             INTEGER NOT NULL REFERENCES Category,
 min_bid_value        INTEGER,
 max_bid_value        INTEGER,
 max_bid_user         INTEGER REFERENCES "User");
```

Data Layer Issues 2/4 – Domain Object

This attribute (`mDescriptionDetail`) is otherwise included in the domain object, but its value is often `null`.

```
public class Auction {
  private int mIdAuction;
  private int mIdOwner;
  private String mName;
  private String mDescription;
  private String mDescriptionDetail;
  private DateTime mCreation;
  private DateTime mEnd;
  private int mIdCategory;      // id of the category
  private Category mCategory;   // the complete record
  private int mMaxBidValue;
  private int mMinBidValue;
  private int mIdMaxBidUser;
        ...
```

Data Layer Issues 3/4 – DAO 1/2

- A select method has to explicitly list the attributes' values read from the database (`description_detail` is not included):

```
public String SQL_SELECT = "SELECT␣idAuction,name,"+
    "description,creation,\"end\",category," +
    "min_bid_value,max_bid_value," +
    "max_bid_user␣FROM␣Auction";
```

Data Layer Issues 4/4 – DAO 2/2

- A read method must include a mechanism enabling to read all or selected attributes:

```
private Collection<Auction> Read(
  SqlDataReader reader, bool complete) {
...
while (reader.Read()) {
 Auction auction = new Auction();
 int i = 0;
 auction.IdAuction = reader.GetInt32(i++);
 ...
 if (complete) {
  if (!reader.IsDBNull(i++)) {
    // desc detail is not always read
    auction.DescriptionDetail = reader.GetString(i);
  }
 }
 ...
```

Presentation Layer Issues – GridView

- We must use the attribute `AutoGenerateColumns="False"` and element `asp:BoundField` for each column of the table, where:
  - `DataField` is an attribute of the domain object
  - `HeaderText` is a header of the UI table

```
<asp:GridView ID="GridViewAuction" runat="server"
 DataKeyNames="IdAuction"
 AutoGenerateColumns="False"
 AllowPaging="True" DataSourceID="odsAuction">
 <Columns>
  <asp:CommandField ShowSelectButton="True" ShowDeleteButton="True
  <asp:BoundField DataField="IdAuction" HeaderText="IdAuction"/>
  <asp:BoundField DataField="Name" HeaderText="Name"/>
  <asp:BoundField DataField="Description" HeaderText="Description"
  <asp:BoundField DataField="Creation" HeaderText="Creation"/>
  <asp:BoundField DataField="End" HeaderText="End"/>
   ...
```

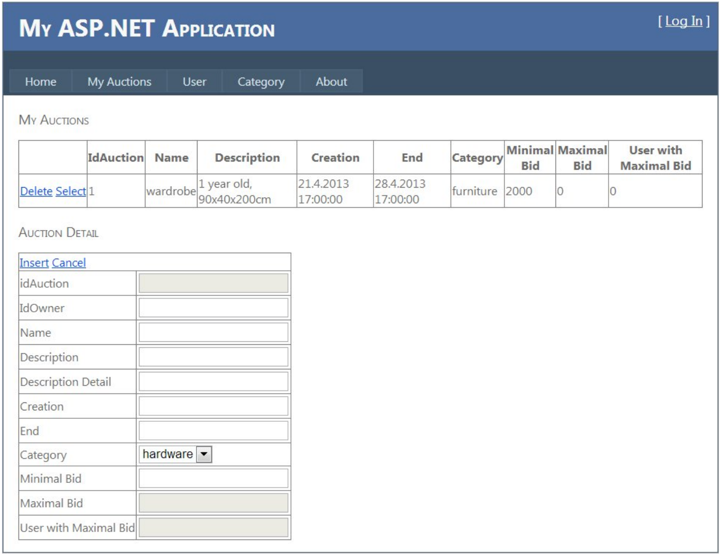
Presentation Layer Issues – DetailsView

- We must also use the attribute `AutoGenerateRows="false"` and element `asp:BoundField` for each column of the table where:
  - `DataField` is an attribute of the domain object
  - `HeaderText` is a header of the UI table

```
<asp:DetailsView ID="DetailsViewAuction" runat="server"
 AutoGenerateRows="false" DataSourceID="odsAuctionDetail"
 DataKeyNames="idAuction" ... >
 <Fields>
 <asp:CommandField ShowInsertButton="True"/>
 <asp:BoundField DataField="IdAuction" HeaderText="idAuction"/>
 <asp:BoundField DataField="IdOwner" HeaderText="IdOwner"/>
 <asp:BoundField DataField="Name" HeaderText="Name"/>
 <asp:BoundField DataField="Description" HeaderText="Description"/
 <asp:BoundField DataField="DescriptionDetail" ... />
 <asp:BoundField DataField="Creation" HeaderText="Creation"/>
  ...
```

Notice: In this case, DetailsView includes all attribute values.

My Auctions Form



My Auctions

	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

Auction Detail

Insert Cancel

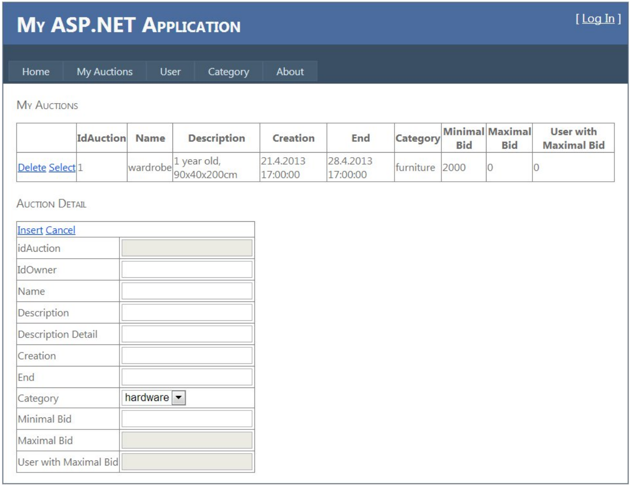
idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware ▾
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

My Auctions Form

- In some cases, we do not want to list all records of a table, but we want to list records for a parameter.
- For example, in the 'My Auctions' Form we want to list all auctions where the owner is the current user.
- In general, there are two ways how to pass parameters to a form:
  - HTTP parameters
  - Session variables

My Auctions Form

- In this case, the list includes only auctions of the current user.



My Auctions

	IdAuction	Name	Description	Creation	End	Category	Minimal Bid	Maximal Bid	User with Maximal Bid
Delete Select	1	wardrobe	1 year old, 90x40x200cm	21.4.2013 17:00:00	28.4.2013 17:00:00	furniture	2000	0	0

Auction Detail

Insert Cancel	
idAuction	
IdOwner	
Name	
Description	
Description Detail	
Creation	
End	
Category	hardware
Minimal Bid	
Maximal Bid	
User with Maximal Bid	

Setting Session Variable

```
protected void Page_Load(object sender, EventArgs e) {
 if (Session["IDUSER"] == null) {
  Session.Add("IDUSER", 1);
 }
}
```

Notice: In a real system, the session variable `IDUSER` should include id the user log on the system, in this case we only set a user with id = 1.

Passing Session Variable - GridView

We must set the session variable as a parameter of DAO's select.

```
<asp:ObjectDataSource ID="odsAuction" runat="server"
 TypeName="AuctionWebApp.Database.AuctionTable"
 SelectMethod="Select"  DeleteMethod="Delete">
  ...
 <SelectParameters>
  <asp:SessionParameter Name="IDUSER" SessionField="IDUSER"
   DefaultValue="1"/>
  </SelectParameters>
</asp:ObjectDataSource>
```

Passing Session Variable - DAO

In a DAO, we must set the parameter – id of the auction owner in this case.

```
public String SQL_SELECT = "SELECT␣a.idAuction,a.name,a.descriptio
 "a.creation,a.\"end\",a.category,a.min_bid_value,␣" +
 "a.max_bid_value,␣a.max_bid_user,c.name␣FROM␣Auction␣a," +
 "Category␣c␣WHERE␣a.owner=@owner␣and␣a.category=c.idCategory";
...
public Collection<Auction> Select(int iduser) {
 Database db = new Database();
 db.Connect();
 SqlCommand command = db.CreateCommand(SQL_SELECT);
 command.Parameters.Add(new SqlParameter("@owner", SqlDbType.Int))
 command.Parameters["@owner"].Value = iduser;
 SqlDataReader reader = db.Select(command);

 Collection<Auction> auctions = Read(reader, false);
 reader.Close();
 db.Close();
 return auctions;
}
```

ORM Rules 1/2

- Developers often ask whether he/she should add some attributes in a domain object.
- For example, all bids of a user in the User domain object.
- Or how to implement a Many-to-Many relationship.

ORM Rules 2/2

We must keep in mind these rules ORM design and implementation:

1. **Minimize the number of queries sent to a DBMS**.
2. **Minimize the volume of data retrieved/sent from/in a DBMS**.
3. **Retrieve only data to be shown to a user**.

Answer: If bids are not shown after data of a user are loaded, then all bids should not be included in the User domain object.

In general: Follow UI forms and these rules to make the data layer of your information system as efficient as possible.

Enhanced Issues

- Loading only data shown in a UI table (data paging).
- Update only changed attributes values.
- and many others.

Notice: The performance of the data layer should be tested after the implementation.