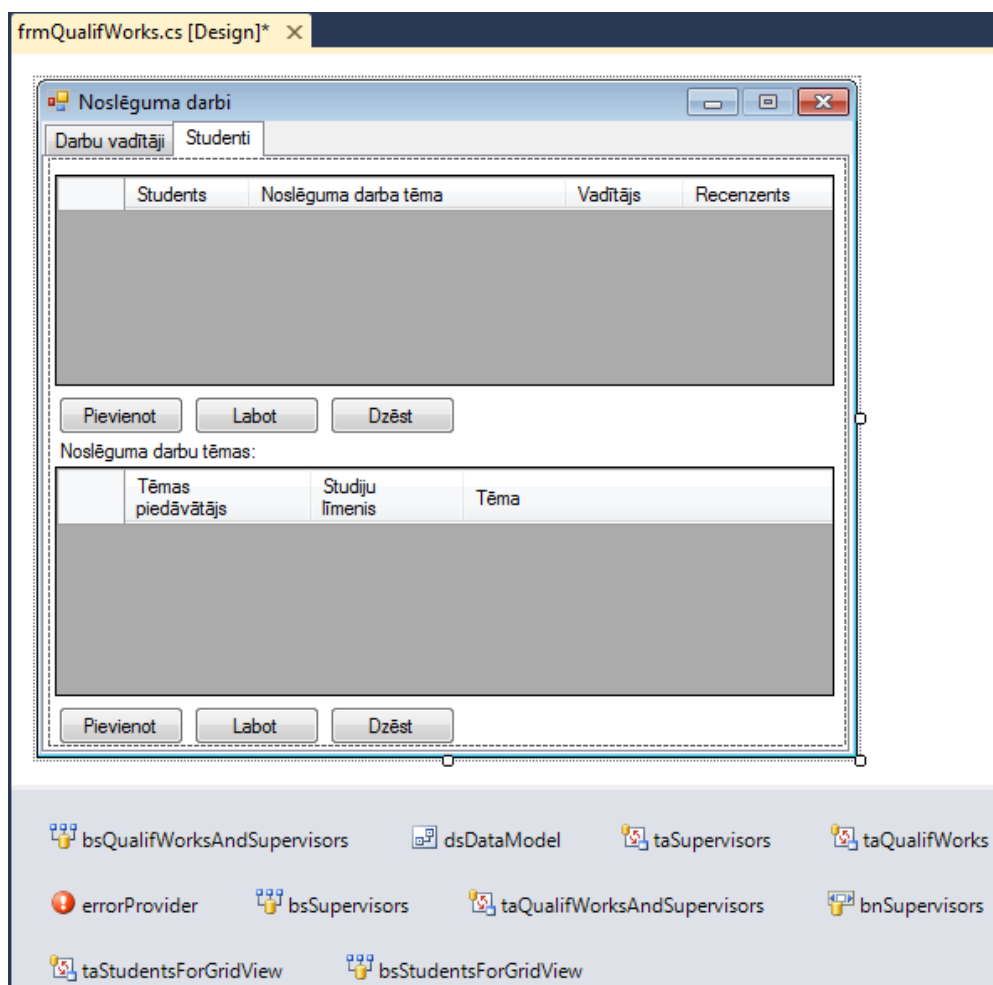


Šī laboratorijas darba mērķis ir padziļināti iepazīties ar iepriekšējā laboratorijas darbā apskatītajiem vizuālajiem rīkiem un programmēšanas paņēmieniem, kas nodrošina darbu ar modālu formu (dialogu) atspoguļošanu, datu nodošanu tiem pēc noteiktiem kritērijiem un apstrādes rezultātu saņemšanu pēc dialoga aizvēršanas.

5.1 UZDEVUMS PASNIEDZĒJA VADĪBĀ

Papildināt 4. laboratorijas darbā iesākto lietojumprogrammu ar šādu funkcionalitāti: cilnē ‘Studenti’ nodrošināt noslēguma darbu tēmu un to vadītāju ievadi un atspoguļošanu, kā arī ar studentu datu ievadi un atspoguļošanu, lai komponentu vizuālais izvietojums pēc šīs funkcionalitātes realizācijas būtu šāds:



Prasītā funkcionalitāte tiks realizēta ar diviem **DataGridView** komponentiem – augšējā atspoguļojot datus par ievadītajiem studentiem, viņu izvēlētajām tēmām, darba vadītāju un recenzentu, ar iespēju pievienot jaunu, rediģēt vai dzēst esošo informāciju. Apakšējā daļā tiks atspoguļota informācija par ievadītajām noslēguma darbu tēmām un vadītājiem, kas tās piedāvā.

Formas nevizuālo komponentu laukumā ir uzvietotas vairāku komponentu instances, kas nodrošina sasaisti ar datu avotiem un sinhronizāciju ar formas elementiem.

Tā kā ievadot studentu datus, būs jānorāda arī informācija par izvēlēto tēmu un tās vadītāju, tad sāksim realizāciju tieši ar noslēguma darbu tēmu ievades un atspoguļošanas funkcionalitātes realizāciju.

Šim nolūkam, uz cilnes instances **tbpStudents**:

- Uzvietot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	lblQualifWorksTopics
Location	4; 168
Text	Noslēguma darbu tēmas:

- Uzvietot kategorijas **Data** kontroli **DataGridView** un uzstādīt īpašības:

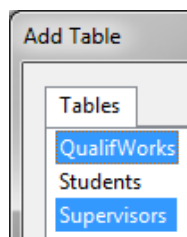
Īpašība:	Vērtība:
(Name)	dgvQualifWorksTopics
AllowUserToAddRows	False
AllowUserToDeleteRows	False
Location	4; 184
MultiSelect	False
ReadOnly	True
SelectionMode	FullRowSelect
Size	465; 137

Tā kā esošā datubāzes struktūra nenodrošina tādu atspoguļojumu, kas ļautu vienlaicīgi redzēt gan noslēguma darba vadītāja vārdu/uzvārdu, gan tēmas nosaukumu, nepieciešams izveidot virtuālu tabulu (skatu), kura realizē šāda SQL vaicājuma rezultāta atspoguļošanu:

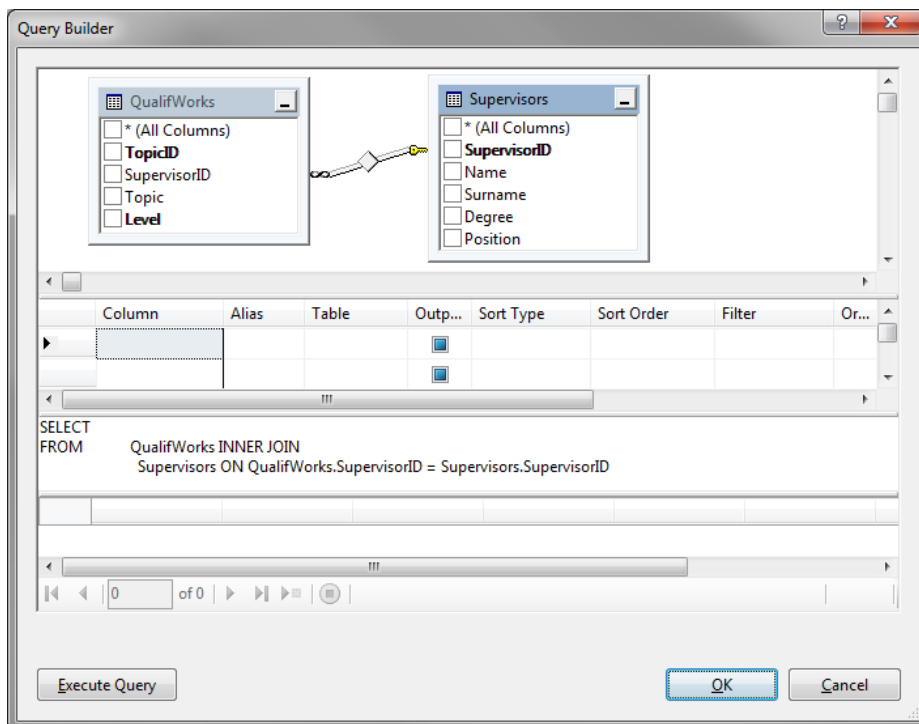
```
SELECT w.SupervisorID, s.Surname + N', ' + s.Name AS Supervisor, w.Level,
       w.TopicID, w.Topic
FROM QualifWorks AS w INNER JOIN
     Supervisors AS s ON w.SupervisorID = s.SupervisorID
ORDER BY Supervisor, w.Topic
```

Šim nolūkam, projektā **DataModel**:

- Ar peles dubultklikšķi uz artefakta **DataModelDataSet.xsd**, atvērt datu kopas redaktoru un uz brīvā laukuma ar peles labo klikšķi atvērt konteksta izvēlni un izvēlēties punktu **Add | TableAdapter....** Atvērsies tabulas adaptera konfigurācijas vednis.
- Tabulas adaptera konfigurācijas vedņa pirmajā dialogā, izvēlēties savienojuma virkni **QualificationWorksConnectionString (Settings)** un nospiegt pogu **Next>**.
- Nākošajā dialogā būs pieejama tikai viena opcija **Use SQL statements**, kas nosaka, ka tabulas adapterim datu izgūšanai no datu avota tiks izmantota SQL izteiksme. Nospiegt pogu **Next>**.
- Nākošajā dialogā ir iespēja norādīt SQL vaicājumu – to var ievadīt pilnībā manuāli, vai arī, nospiežot pogu **Query Builder...**, izveidot to vizuālā redaktorā. Lai iepazītos ar vaicājumu veidošanas vizuālo redaktoru, izveidosim šo vaicājumu vizuāli, šim nolūkam nospiegt pogu **Query Builder....**
- Atvērsies vaicājumu redaktora **Query Builder** dialogs, bet virs tā automātiski parādīsies dialogs ar izvēlētajā datu avotā pieejamām tabulām. No tabulu saraksta, turot nospiegtu klaviatūras taustiņu CTRL, izvēlēties abas vaicājumam nepieciešamās tabulas:



- Pēc tabulu iezīmēšanas, nospieš pogu **Add**, un pēc tam pogu **Close**, lai aizvērtu tabulu pievienošanas dialogu. Vaicājumu redaktorā tiek parādītas izvēlētas tabulas, kā arī to savstarpējās relācijas:



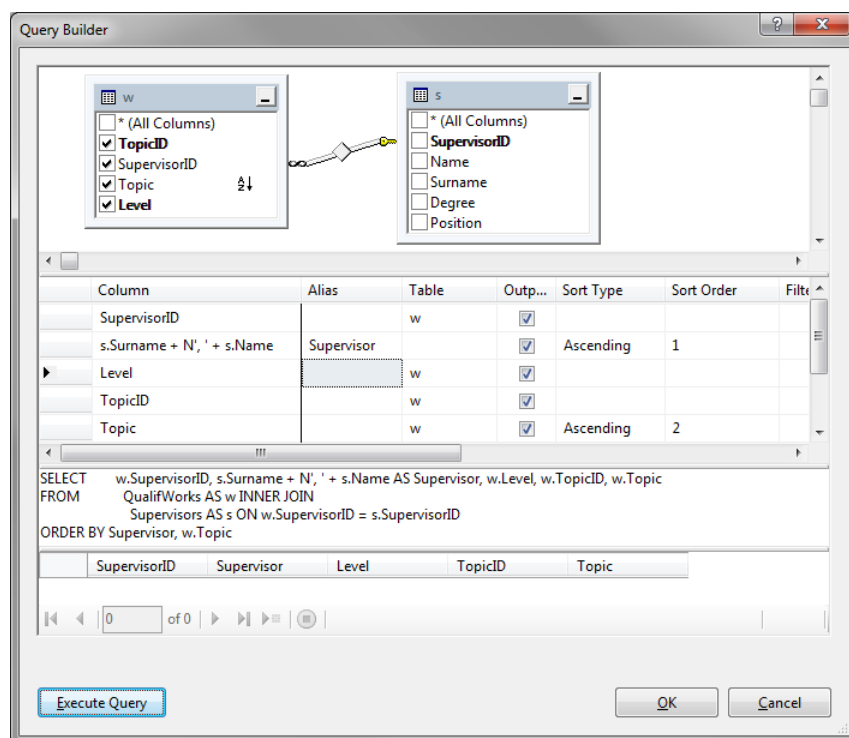
Vaicājuma vizuālā redaktora dialogs sastāv no vairākām sekcijām:

- ❖ Augšpusē tiek atspoguļotas vaicājumā izvēlētas tabulas ar to savstarpējām relācijām. Atzīmējot tabulas lauku, tas automātiski ievietosies SQL vaicājuma SELECT daļā, bet atslēdzot – tas izņemsies no SQL vaicājuma SELECT daļas.
- ❖ Zem tabulu diagrammas ir pieejams tabulveida atspoguļojums ar detalizētāku informāciju par katru izvēlēto tabulas lauku, ar iespēju norādīt papildus īpašības, kas tiks piemērotas vaicājumam. Piemēram, lauka sinonīmu (**Alias**), sakārtošanas veidu (**Sort Type**) un kārtību (**Sort Order**), filtrēšanas nosacījumus (**Filter**), kā arī filtrēšanas alternatīvas (**Or...**).
- ❖ Zem izvēlēto lauku tabulas tiek atspoguļota SQL vaicājuma tekošā struktūra – tā tiek automātiski aktualizēta, uzreiz pēc izmaiņu veikšanas iepriekšējās divās sekcijās. Veicot izmaiņas šajā sekcijā, tās automātiski atspoguļosies arī iepriekšējās divās sekcijās.
- ❖ Visbeidzot, dialoga apakšējā sekcijā tiek atspoguļots sagatavotā SQL vaicājuma izpildes rezultāts (vaicājuma izpildi aktivizē ar pogu **Execute Query**). Pirms SQL vaicājuma izpildes notiek tā pareizības pārbaude, un ja tiek konstatētas sintaktiskās vai semantiskās kļūdas, par tām tiek izvadīts paskaidrojošs ziņojums.

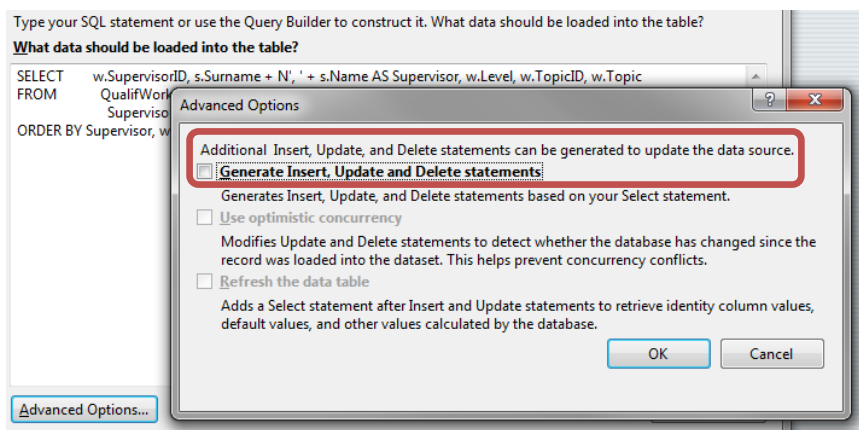
Lai pabeigtu iesākto vaicājumu, veikt šādas darbības:

- No tabulas **QualifWorks** izvēlēties kolonnas **TopicID**, **SupervisorID**, **Topic** un **Level** (ievērot, ka automātiski izmainās arī SQL vaicājuma sekcija un izvēlēto lauku tabula).
- Izvēlēto lauku tabulā, pāriet uz tukšu rindu un kolonnā **Column** ierakstīt **Surname+N**, **+Name**, bet kolonnā **Alias** – **Supervisor**. Šādi tiek definēts kombinēts lauks, kas sastāv no tabulas **Supervisors** lauku **Surname** un **Name** vērtību konkatēnācijas.

- Izvēlēto lauku tabulā uzdot rezultātu kārtības nosacījumus, vispirms lauka ar sinonīmu **Supervisor**, kolonnā **Sort Type** norādīt **Ascending**, pēc tam to pašu izdarīt laukam **Topic**.
- Lai saīsinātu rezultējošā SQL vaicājuma garumu, uzdot abām tabulām sinonīmus. To panāk, tabulu sekcijā uz tabulas nosaukuma atverot konteksta izvēlni un izvēloties punktu **Properties**. Tabulas īpašību dialoga atribūtā **Alias** norādīt: tabulai **QualifWorks – w**, **Supervisors – s**;
- Izmantojot peli, izvēlēto lauku tabulā, sakārtot izvēlētos laukus šādā secībā: **SupervisorID**, **s.Surname + N', ' + s.Name**, **Level**, **TopicID** un **Topic**.
- Pārbaudīt vaicājuma korektumu, nospiežot pogu **Execute Query**. Ja kļūdas netiek konstatētas, tad vaicājuma rezultāta sekcijā tiek izvadīts vaicājuma izpildes rezultāts. Ja tabulā **QualifWorks** vēl nav saglabāts neviens raksts, tad vaicājuma rezultātā var redzēt tikai kolonnu nosaukumus.

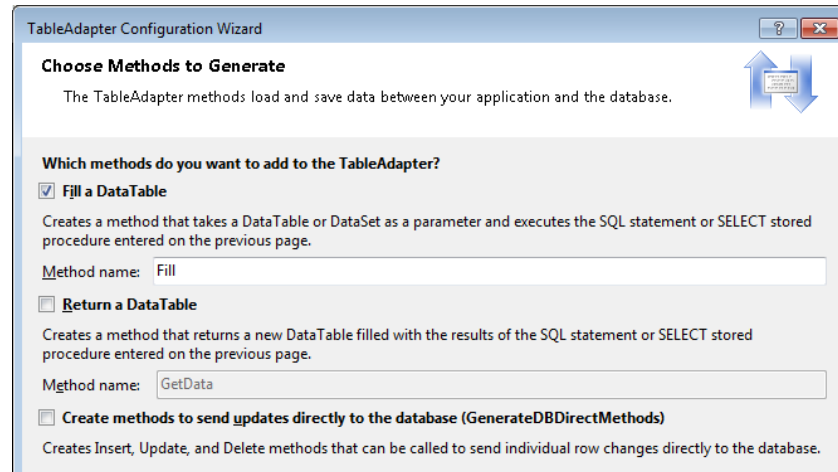


- Nospiežot **OK**, aizvērt vaicājuma redaktora dialogu.
- Lai tabulas adapteris nodrošināšanu tikai datu izgūšanu, nospieš pogu **Advanced Options...** un atslēgt opciju **Generate Insert, Update and Delete statements**:



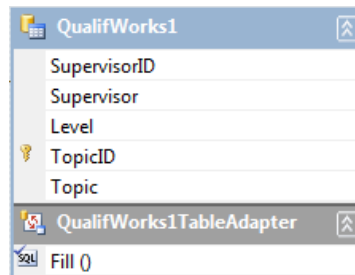
Tas nepieciešams, lai vednis nenogenerētu lieku pirmkodu metodēm, kas netiks izmantotas.

- Tabulas adaptera vednī, nospiežot pogu **Next>**, tiek izvadīts dialogs, kurā jāspecificē metodes, izmantojot kuras tiks nodrošināta datu izgūšana. Šajā dialogā atstāt tikai opciju **Fill a DataTable** un par tās metodes nosaukumu atstāt 'Fill':



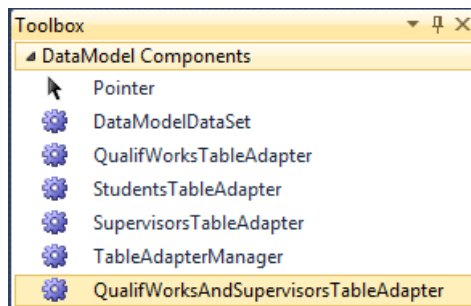
Tabulas adaptera metodi **Fill()** tiek izmantota, lai ielādētu datus jau esošā **DataTable** instancē. Metode **GetData()** izveido un atgriež jaunu **DataTable** instanci, kas mūsu lietojumprogrammas gadījumā nebūs nepieciešams, tāpēc nav jāatzīmē tās pirmkoda ģenerēšanas nepieciešamība. Pēdējā izvēles rūtiņa ļauj pielāgot datu modificēšanas metodes, kuras mūsu lietojumprogrammā nebūs nepieciešams izmantot.

- Tabulas adaptera vednī, nospiežot pogu **Next>**, tiek atspoguļots kopsavilkums par veiktajām darbībām, un, nospiežot pogu **Finish**, tabulu artefaktu redaktorā tiek ievietots jaunā tabulas adaptera klase, ar atbilstošām relācijām uz jau esošajām tabulām:



- Iezīmējot tabulu **QualifWorks1**, īpašību logā pārsaukt tās nosaukumu (īpašība **Name**) uz **QualifWorksAndSupervisors**.
- Lai veiktās izmaiņas stātos spēkā (t.i. tiku nogenerēts klašu pirmkods), nokompilēt risinājumu.

- Lai pārlicinātos, ka pēc risinājuma nokompilēšanas ir izveidots jauns tabulas adaptera komponents, kā arī aktualizēta datu kopas klase **DataModelDataSet**, atvērt formas **frmQualifWorks** redaktoru un komponentu loga kategorijā **DataModel Components** atrast jaunizveidotā tabulas adaptera komponentu:



Pabeigt kontroles instances **dgvQualifWorksTopics** sasaisti ar jaunizveidoto tabulas adapteri. Šim nolūkam:

- Uz formas **frmQualifWorks** nevizuālo komponentu laukuma, uzvietot komponentu **QualifWorksAndSupervisors** un pārsaukt par **taQualifWorksAndSupervisors**.
- Uz formas **frmQualifWorks** nevizuālo komponentu laukuma, uzvietot kategorijas **Data** komponentu **BindingSource** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	bsQualifWorksAndSupervisors
AllowNew	False
DataSource	dsDataModel
DataMember	QualifWorksAndSupervisors

- Kontroli **dgvQualifWorksTopics** sasaistīt ar tabulas adapteri **QualifWorksAndSupervisors**, tās īpašībā **DataSource** norādot **bsQualifWorksAndSupervisors**.
- Kontrolē **dgvQualifWorksTopics** automātiski ielādēsies tabulas adaptera **QualifWorksAndSupervisors** kolonnas. Atvērt kontroles kolonnu redaktoru un kolonnām **SupervisorID** un **TopicID** uzstādīt īpašību **Visible** uz **False**.
- Atlikušajām kolonnām nomainīt īpašības:

Īpašība:	Vērtība:
Kolonna: supervisor DataGridViewTextBoxColumn	
HeaderText	Tēmas piedāvātājs
AutoSizeMode	DisplayedCells

Īpašība:	Vērtība:
Kolonna: level DataGridViewTextBoxColumn	
AutoSizeMode	DisplayedCells
ColumnType	DataGridViewComboBoxColumn
DisplayStyle	Nothing
HeaderText	Studiju līmenis

Īpašība:	Vērtība:
Kolonna: topic DataGridViewTextBoxColumn	
HeaderText	Tēma
AutoSizeMode	Fill

- Nodrošināt, ka pie formas **frmQualifWorks** parādīšanas, tiek aizpildīta kontrole **dgvQualifWorksTopics** ar datiem no tabulas adaptera

QualifWorksAndSupervisors. Lai to nodrošinātu, formas **frmQualifWorks** notikuma **Load** apstrādātāju papildināt ar šādu pirmkoda rindu:

```
taQualifWorksAndSupervisors.Fill(dsDataModel.QualifWorksAndSupervisors);
```

- Redzams, ka kontroles **dgvQualifWorksTopics** kolonna **levelDataGridViewTextBoxColumn** ir kombinētais saraksts - tas nepieciešams, lai studiju līmeņu skaitliskās vērtības tiktu aizveidotas ar atbilstošu aprakstošu tekstu. Tā kā aprakstošie teksti, netiek glabāti datu bāzē, tad izveidosim atbilstoša formāta klasi, lai informācija tiktu ņemta no tās definīcijas.
- Patstāvīgi pievienot projektam **DataModel** jaunu klases artefaktu, kuru nosaukt par **QualifLevel**, un papildināt ar šādu definīciju:

```
namespace DataModel
{
    public class QualifLevel
    {
        public static ArrayList Levels
        {
            get
            {
                if (_levels == null)
                {
                    _levels = new ArrayList();
                    _levels.Add(new QualifLevel(0, "nav definēts"));
                    _levels.Add(new QualifLevel(1, "pirmā līmeņa"));
                    _levels.Add(new QualifLevel(2, "bakalaura"));
                    _levels.Add(new QualifLevel(3, "maģistra"));
                    _levels.Add(new QualifLevel(4, "doktora"));
                }
                return _levels;
            }
        }
        private static ArrayList _levels;

        public const string ValueMember = "LevelId";
        public int LevelId { get { return _levelId; } }
        private int _levelId;

        public const string DisplayMember = "LevelDescription";
        public string LevelDescription { get { return _levelDescription; } }
        private string _levelDescription;

        private QualifLevel(int id, string descr)
        {
            _levelId = id;
            _levelDescription = descr;
        }
    }
}
```

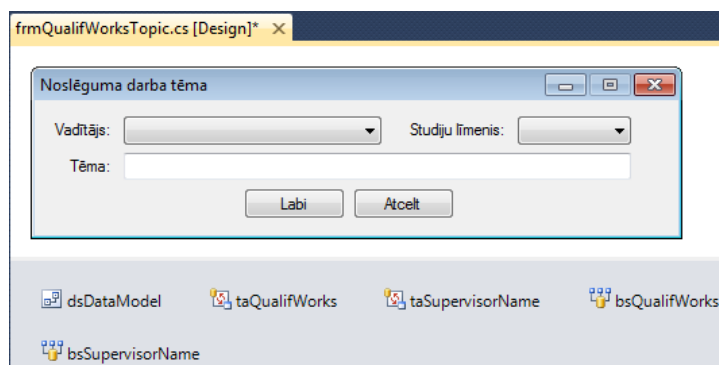
- Papildināt formas **frmQualifWorks** konstruktoru ar šādām pirmkoda rindām:

```
public frmQualifWorks()
{
    InitializeComponent();

    levelDataGridViewTextBoxColumn.DataSource = QualifLevel.Levels;
    levelDataGridViewTextBoxColumn.DisplayMember
        = QualifLevel.DisplayMember;
    levelDataGridViewTextBoxColumn.ValueMember = QualifLevel.ValueMember;
}
```

Jāuzsver, ka kombinēto sarakstu komponentos par saraksta avotu var tikt izmantotas ne tikai statiskas teksta virknes, bet arī datu avotu informācija. Šim nolūkam kontrolē tika uzstādīta: īpašība **DataSource**, norādot saraksta instanci, caur kuru notiks datu izgūšana; īpašība **DisplayMember**, kas nosaka, kurš datu avota lauks tiks atspoguļots kā saraksta elements uz displeja; īpašība **ValueMember**, kas nosaka, kurš datu avota lauks tiks izmantots par saraksta vērtību, t.i. lietotājs vizuāli redzēs **DisplayMember** lauka vērtības, bet īpašībā **SelectedValue** tiks saglabāta lauka **ValueMember** vērtība.

Kaut arī pašlaik jau tiek nodrošināta iespēja atspoguļot datus no tabulām **QualifWorks** un **Supervisors**, vēl ir jārealizē jaunu noslēguma darba tēmu reģistrācijas un labošanas funkcionalitāte. To realizēsim kā modālu dialogu, kas tiks atvērts pēc pogu **Pievienot** vai **Labot** nospiešanas:



Formas dati tiks piesaistīti tabulai **QualifWorks**, tāpēc formā būs jāizmanto šīs tabulas adaptera klase **QualifWorksTableAdapter**.

Formas kombinētajā sarakstā **Vadītājs**: tiks piedāvāts izvēlēties starp iepriekš ievadītajiem noslēguma darbu vadītājiem. Šim nolūkam būs nepieciešams izveidot jaunu tabulas adapteri, kura rezultāts būs saraksts ar laukiem **SupervisorID**, **Name** un **Surname**.

Lai to realizētu:

- Projektam **QualifWorksClient** pievienot jaunu formas artefaktu un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	frmQualifWorksTopic
FormBorderStyle	FixedDialog
MaximizeBox	False
ShowInTaskbar	False
Size	483; 128
StartPosition	CenterParent
Text	Noslēguma darba tēma

- Projektā **DataModel** patstāvīgi izveidot jaunu tabulas adapteri (tikai datu izgūšanai ar metodi **Fill()**), nosaucot to par **SupervisorName**, kas realizē šādu SQL vaicājumu:

```
SELECT SupervisorID, Position + N' ' + Name + N' ' + Surname AS Supervisor
FROM Supervisors
ORDER BY Surname, Name
```

- Lai tabulas adaptera **SupervisorName** komponents tiku noģenerēts, nokompilēt risinājumu, un pēc tam uz formas **frmQualifWorksTopic** uzvietot komponentus **DataModelDataSet**, **QualifWorksTableAdapter**, **SupervisorNameTableAdapter** un divus **BindingSource**, un uzstādīt īpašības:

Īpašība:	Vērtība:
DataModelDataSet	
(Name)	dsDataModel
QualifWorksTableAdapter	
(Name)	taQualifWorks
SupervisorNameTableAdapter	
(Name)	taSupervisorName
BindingSource	
(Name)	bsQualifWorks
DataSource	dsDataModel

DataMember	QualifWorks
BindingSource	
(Name)	bsSupervisorName
AllowNew	False
DataSource	dsDataModel
DataMember	SupervisorName

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	lblSupervisor
Location	12; 15
Text	Vadītājs:

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **ComboBox** un uzstādīt īpašības:

Īpašība:	Vērtība:
(DataBindings) SelectedValue	bsQualifWorks - SupervisorID
(Name)	cboSupervisor
DataSource	bsSupervisorName
DisplayMember	Supervisor
DropDownStyle	DropDownList
Location	66; 12
Size	191; 21
ValueMember	SupervisorID

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	lblLevel
Location	275; 15
Text	Studiju līmenis:

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **ComboBox** un uzstādīt īpašības:

Īpašība:	Vērtība:
(DataBindings) SelectedValue	bsQualifWorks - Level
(Name)	cboLevel
DropDownStyle	DropDownList
Location	358; 12
Size	84; 21

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	lblTopic
Location	23; 42
Text	Tēma:

- Uz formas **frmQualifWorksTopic** uzvīetot kontroli **TextBox** un uzstādīt īpašības:

Īpašība:	Vērtība:
(DataBindings) Text	bsQualifWorks - Topic
(Name)	txtTopic
Location	66; 39
Size	376; 20

- Uz formas **frmQualifWorksTopic** uzvietot kontroli **Button** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	btnOk
Location	160; 65
Text	Labi

- Papildināt kontroles **btnOk** notikuma **Click** apstrādātāju ar šādu realizāciju:

```
private void btnOk_Click(object sender, EventArgs e)
{
    this.DialogResult = DialogResult.None;
    try
    {
        bsQualifWorks.EndEdit();
        taQualifWorks.Update(dsDataModel.QualifWorks);
        this.DialogResult = DialogResult.OK;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, this.Text, MessageBoxButtons.OK
            , MessageBoxIcon.Error);
    }
}
```

- Uz formas **frmQualifWorksTopic** uzvietot kontroli **Button** un uzstādīt īpašības:

Īpašība:	Vērtība:
(Name)	btnCancel
Location	241; 65
Text	Atcelt

- Papildināt kontroles **btnCancel** notikumu **Click** ar šādu realizāciju:

```
private void btnCancel_Click(object sender, EventArgs e)
{
    bsQualifWorks.CancelEdit();
    this.DialogResult = DialogResult.Cancel;
}
```

- Formas **frmQualifWorksTopic** konstruktoru papildināt ar šādām pirmkoda rindām:

```
public void frmQualifWorksTopic()
{
    InitializeComponent();

    cboLevel.DataSource = QualifLevel.Levels;
    cboLevel.DisplayMember = QualifLevel.DisplayMember;
    cboLevel.ValueMember = QualifLevel.ValueMember;

    taSupervisorName.Fill(dsDataModel.SupervisorName);
}
```

Šajā pirmkodā kontrolei **cboLevel** tiek izmantots mehānisms, kas ļauj piesaistīt kombinētā saraksta elementus no saraksta **QualifLevel.Levels**, kura elementi ir klases **QualifLevel** eksemplāri.

- Papildināt formas **frmQualifWorksTopic** klasi ar šādām metodēm:

```
public void AddNew()
{
    // aktivizē jauna raksta pievienošanu
    bsQualifWorks.AddNew();
}

public void Edit(int topicId)
{
    // aktivizē esoša raksta labošanu
    taQualifWorks.FillByTopicID(dsDataModel.QualifWorks, topicId);
}
```

Metodes **Edit()** realizācijā tiek izsaukta tabulas adaptera metode **FillByTopicID()**, kas vēl nav realizēta. Šīs metodes mērķis būs aizpildīt lokālo datu kešatmiņu, norādot papildus nosacījumu – eksistējošas tēmas identifikatoru.

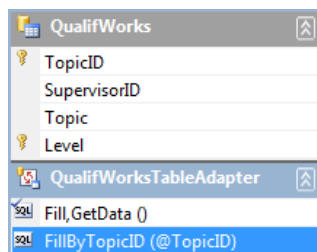
Lai realizētu metodi **FillByTopicID()**:

- Atvērt projekta **DataModel** artefaktu **DataModelDataSet.xsd** un izvēlēties tabulu **QualifWorks**.
- Uz tabulas nosaukuma atvērt konteksta izvēlni un izvēlēties punktu **Add | Query...**. Atveras tabulas adaptera vaicājumu konfigurēšanas vednis, kurā jāspiež poga **Next>**.
- Nākošajā dialogā jāatstāj noklusētā izvēles rūtiņa (**SELECT which returns rows**) un jāspiež **Next>**.
- Nākošajā dialogā jānorāda SQL vaicājums, kas atgriezīs rezultātus. Veidojot vaicājumu, jācenšas atgriezt tieši tos pašus laukus, kurus nodrošina pati tabula (t.i. metode **Fill()**). Pretējā gadījumā tiks izvadīts brīdinājums un var rasties sarežģījumi ar šo metožu rezultātu savstarpēju savietojamību.
- Par SQL vaicājumu norādīt šādu izteiksmi:

```
SELECT [TopicID], [SupervisorID], [Topic], [Level]
FROM [QualifWorks]
WHERE TopicID = @TopicID
```

Redzams, ka vaicājumā tiek norādīts viens parametrs - **@TopicID**. Pateicoties tam, metodei tiks noģenerēts parametrs ar nosaukumu **TopicID** un tipu, kas atbilst tabulas **QualifWorks** lauka **TopicID** tipam.

- Nospiešot pogu **Next>** un nākošajā dialogā atstāt tikai opciju **Fill a DataTable**, bet laukā **Method name:** norādīt **FillByTopicID**. Nospiežot pogu **Next>** un pēc tam **Finish** tabulas adapterim tiks pievienota jauna metode, kuras pirmkods tiks noģenerēts pēc risinājuma kompilēšanas:



Līdz ar to arī tagad veiksmīgi tiks nokompilēta arī formas **frmQualifWorksTopic** metode **Edit()**, kurā tiek izsaukta noģenerētā tabulas adaptera metode **FillByTopicID()**.

- Lai nodrošinātu jaunas tēmas izveidošanas, labošanas un dzēšanas funkcionalitāti, uz formas **frmQualifWorks** cilnes **tbpStudents** uzvietot trīs kontroles **Button** instances un uzstādīt īpašības:

Īpašība:	Vērtība:
Button	
(Name)	btnTopicAdd
Location	6; 327
Text	Pievienot
Button	
(Name)	btnTopicEdit
Location	87; 327
Text	Labot
Button	

(Name)	btnTopicDelete
Location	168; 327
Text	Dzēst

- Katrai no šīm pogām izveidot šādus notikuma **Click** apstrādātājus:

```
private void btnTopicAdd_Click(object sender, EventArgs e)
{
    frmQualifWorksTopic frm = new frmQualifWorksTopic();

    frm.AddNew();

    if (frm.ShowDialog() == DialogResult.OK)
        taQualifWorksAndSupervisors.Fill(
            dsDataModel.QualifWorksAndSupervisors);
}

private void btnTopicEdit_Click(object sender, EventArgs e)
{
    frmQualifWorksTopic frm = new frmQualifWorksTopic();
    // ar šīm manipulācijām tiek iegūts datu kešatmiņas
    // tekošā raksta objekts
    DataGridView rowView
        = (DataGridView)bsQualifWorksAndSupervisors.Current;
    if (rowView == null)
        return; // ja raksts nav izvēlēts (vai arī tādu nav),
        // tad neturpināt

    DataModel.DataModelDataSet.QualifWorksAndSupervisorsRow row
    = (DataModel.DataModelDataSet.QualifWorksAndSupervisorsRow) rowView.Row;

    frm.Edit(row.TopicID);

    if (frm.ShowDialog() == DialogResult.OK)
    {
        // saglabā raksta pozīciju
        int index = bsQualifWorksAndSupervisors.Position;
        taQualifWorksAndSupervisors.Fill
            (dsDataModel.QualifWorksAndSupervisors);
        // atjauno raksta pozīciju, jo metode Fill() to uzstāda uz 1
        bsQualifWorksAndSupervisors.Position = index;
    }
}

private void btnTopicDelete_Click(object sender, EventArgs e)
{
    DataGridView rowView =
        (DataGridView)bsQualifWorksAndSupervisors.Current;
    if (rowView == null)
        return;
    DataModel.DataModelDataSet.QualifWorksAndSupervisorsRow row
    = (DataModel.DataModelDataSet.QualifWorksAndSupervisorsRow)
        rowView.Row;

    int index = bsQualifWorksAndSupervisors.Position;
    taQualifWorks.Delete(row.TopicID);
    taQualifWorksAndSupervisors.Fill(
        dsDataModel.QualifWorksAndSupervisors);
    bsQualifWorksAndSupervisors.Position = index;
}
```

- Papildināt formas **frmQualifWorks** metodes **Idle()** realizāciju ar šādām pirmkoda rindām:

```
btnTopicDelete.Enabled = btnTopicEdit.Enabled =
    bsQualifWorksAndSupervisors.Current != null;
```

- Pārbaudīt noslēguma tēmas pievienošanas un rediģēšanas funkcionalitāti, pievienojot vairākas noslēguma darbu tēmas, studiju līmeņu un to vadītājus.

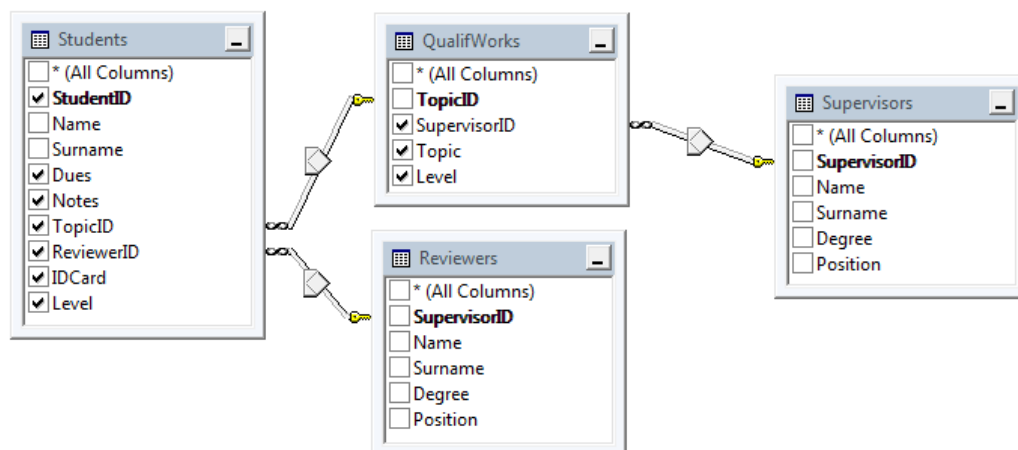
Lai nodrošinātu studentu un viņu izvēlēto noslēguma darba tēmu saraksta atspoguļošanu, uz formas **frmQualifWorks** cilnes **tbpStudents** jāuzvieto kontrole **DataGridView**, par kuras avotu jānorāda virtuāla tabula **StudentsForGridView**, ar šādu SQL izteiksmi:

```

SELECT Students.Surname + N', ' + Students.Name AS Student, Supervisors.Name + N', ' + Supervisors.Surname AS Supervisor, QualifWorks.Topic, Reviewers.Name + N', ' + Reviewers.Surname AS Reviewer, Students.Notes, Students.IDCard, Students.Dues, Students.Level AS StudentLevel, Students.StudentID, Students.TopicID, QualifWorks.Level AS TopicLevel, QualifWorks.SupervisorID, Students.ReviewerID
FROM Students LEFT OUTER JOIN QualifWorks ON QualifWorks.TopicID = Students.TopicID LEFT OUTER JOIN Supervisors ON Supervisors.SupervisorID = QualifWorks.SupervisorID LEFT OUTER JOIN Supervisors AS Reviewers ON Reviewers.SupervisorID = Students.ReviewerID
ORDER BY Student

```

Jeb vizuāli:



kur tabula **Reviewers** ir sinonīms tabulai **Supervisors**.

- Patstāvīgi izveidot iepriekš minēto virtuālo tabulu **StudentsForGridView** (tikai datu izgūšanai ar metodi **Fill()**), uzvietot uz cilnes kontroles **DataGridView** un **BindingSource** un **StudentsForGridViewTableAdapter**, un uzstādīt īpašības:

Īpašība:	Vērtība:
StudentsForGridViewTableAdapter	
(Name)	taStudentsForGridView
BindingSource	
(Name)	bsStudentsForGridView
AllowNew	False
DataSource	dsDataModel
DataMember	StudentsForGridView
DataGridView	
(Name)	dgvStudents
AllowUserToAddRows	False
AllowUserToDeleteRows	False
DataSource	bsStudentsForGridView
Location	4; 10
MultiSelect	False
ReadOnly	True
Size	465; 126
SelectionMode	FullRowSelect

- Pēc datu avota piesaistes izdzēst visas kontroles **dgvStudents** kolonnas, izņemot: **Student**, **Supervisor**, **Topic** un **Reviewer**. Šīm kolonnām uzstādīt īpašības uz:

Īpašība:	Vērtība:
Kolonna: studentDataGridViewTextBoxColumn	
HeaderText	Students
AutoSizeMode	DisplayedCells

Īpašība:	Vērtība:
Kolonna: topic DataGridViewTextBoxColumn1	
HeaderText	Noslēguma darba tēma
AutoSizeMode	Fill

Īpašība:	Vērtība:
Kolonna: supervisor DataGridViewTextBoxColumn1	
HeaderText	Vadītājs
AutoSizeMode	DisplayedCells

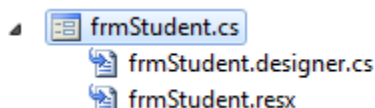
Īpašība:	Vērtība:
Kolonna: reviewer DataGridViewTextBoxColumn	
HeaderText	Recenzents
AutoSizeMode	DisplayedCells

- Papildināt formas **frmQualifWorks** notikuma **Load** apstrādātāju ar šādu pirmkoda rindu:

```
taStudentsForGridView.Fill(dsDataModel.StudentsForGridView);
```

Līdzīgu funkcionalitāti kā noslēguma darbu tēmu ievadei un labošanai, jānodrošina arī pašu studentu ievadei un sasaistei ar noslēguma darbu tēmām. Tomēr, lai pārbaudītu, kā strādā jau gatavu artefaktu pievienošana projektam, pievienot projektam **QualifWorkClient** arhīva **frmStudent.zip** saturu.

Šim nolūkam atarhivēt **frmStudent.zip** saturu pagaidu katalogā un projekta pārlūka logā uz projekta **QualifWorkClient** virsotnes konteksta izvēlnes izvēlēties punktu **Add | Existing Item...** un dialogā pāriet uz pagaidu katalogu, un izvēlēties visus trīs tajā esošos failus: **frmStudent.cs**, **frmStudent.Designer.cs** un **frmStudent.resx**. Nospiežot pogu **Add**, šie faili tiks iekopēti projekta katalogā un parādīsies arī projekta **QualifWorkClient** pārlūka logā:



Atverot formu **frmStudent.cs** formas redaktorā, parādīsies tās kontroļu izvietoījums:

Atliek tikai realizēt šīs formas izsaukšanas funkcionalitāti no formas **frmQualifWorks**. Šim nolūkam:

- Formas **frmQualifWorks** cilnē **tbpStudents** uzvietot trīs **Button** kontroles un uzstādīt īpašības:

Īpašība:	Vērtība:
Button	
(Name)	btnStudentAdd
Location	6; 142
Text	Pievienot
Button	
(Name)	btnStudentEdit
Location	87; 142
Text	Labot
Button	
(Name)	btnStudentDelete
Location	168; 142
Text	Dzēst

Pogām **Pievienot** un **Labot**, izveidot šādus notikuma **Click** apstrādātājus:

```
private void btnStudentAdd_Click(object sender, EventArgs e)
{
    frmStudent dlg = new frmStudent();

    dlg.AddNew();

    if (dlg.ShowDialog() == DialogResult.OK)
    {
        int index = bsStudentsForGridView.Position;
        taStudentsForGridView.Fill(dsDataModel.StudentsForGridView);
        if (-1 != index)
            bsStudentsForGridView.Position = index;
    }
}

private void btnStudentEdit_Click(object sender, EventArgs e)
{
    DataRowView rowView = (DataRowView)bsStudentsForGridView.Current;
    if (rowView == null)
        return;

    DataModel.DataModelDataSet.StudentsForGridViewRow row
        = (DataModel.DataModelDataSet.StudentsForGridViewRow) rowView.Row;

    frmStudent dlg = new frmStudent();

    dlg.Edit(row.StudentID);

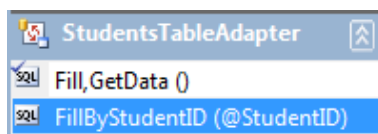
    if (dlg.ShowDialog() == DialogResult.OK)
    {
        int index = bsStudentsForGridView.Position;
        taStudentsForGridView.Fill(dsDataModel.StudentsForGridView);
        if (-1 != index)
            bsStudentsForGridView.Position = index;
    }
}
```

Pogas **btnStudentDelete** nospiešanas funkcionalitāti pagaidām nav jārealizē!

Mēģinot nokompilēt projektu, tiks izvadīta vismaz viena kļūda – importētās formas **frmStudent** metodē **Edit()** tiek izsaukta tabulas adaptera metode **FillByStudentID()**, kas pašlaik vēl nav realizēta.

- Projektā **DataModel** datu kopas diagrammā tabulai **Students** patstāvīgi realizēt metodi **FillByStudentID()**, par pamatu izmantojot šādu SQL izteiksmi:

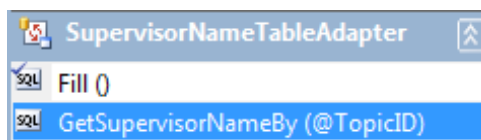
```
SELECT [StudentID], [Name], [Surname], [Dues], [Notes], [TopicID], [ReviewerID],  
[IDCard], [Level] FROM [Students] WHERE [StudentID] = @StudentID
```



- Tabulai **SupervisorName** patstāvīgi realizēt metodi **GetSupervisorNameBy()**, vednī par SQL vaicājuma tipu izvēloties opciju **SELECT which returns a single value**, un norādot šādu SQL vaicājumu:

```
SELECT Supervisors.Surname + ', ' + Supervisors.Name AS Supervisor  
FROM Supervisors INNER JOIN QualifWorks ON  
Supervisors.SupervisorID = QualifWorks.SupervisorID  
WHERE (QualifWorks.TopicID = @TopicID)
```

Šim vaicājumam ir viens parametrs **TopicID**, kas pēc tā vērtības nosaka un atgriež atbilstošu vadītāja vārdu un uzvārdu:



- Papildināt formas **frmQualifWorks** metodes **Idle()** realizāciju ar šādām pirmkoda rindām:

```
btnStudentEdit.Enabled = bsStudentsForGridView.Current != null;  
btnStudentDelete.Enabled = btnStudentEdit.Enabled;
```

Pārbaudīt studentu datu pievienošanas un rediģēšanas funkcionalitāti, pievienojot vairākus studentus, norādot tiem iepriekš ievadītas tēmas, kā arī nozīmējot recenzentus.