

3.18. Formada sichqoncha ko'rsatgichi orqali chizish

Ushbu mavzuda qanday qilib, formaga sichqoncha yordamida chizish ko'rsatiladi. Ya'ni, sichqonchani chap yoki o'ng tugmasi bosilganda formaga chizish imkonini beruvchi dastur yozish masalasi qo'yilgan. Agar foydalanuvchi sichqoncha tugmasini qo'yib yuborsa, u holda chizish jarayoni to'xtatiladi. Loyihalashtirilayotgan formada chizmani tozalash uchun mo'ljallangan "Tozalash" tugmasini inobatga olish lozim.

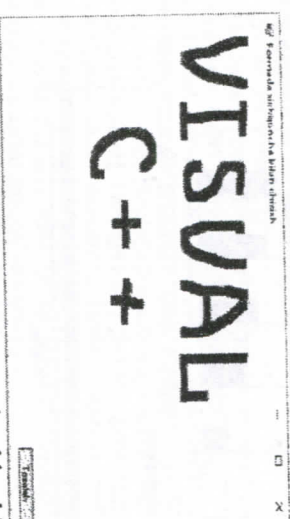
Ushbu vazifani bajaruvchi dasturni tuzish uchun Visual Studio 2012 dasturi ishga tushiriladi, asosiy oyna menyusida bo'limlardan **File->New->Project...** buyruqlari beriladi yoki **Ctrl+Shift+N** klavishalari bosiladi, ochilgan oynada loyihaga "Chizish" nomi berilib, **OK** tugmasi bosiladi. **ToolBox** panelidan formaga **Button** komponentasi joylashtiriladi va quyidagi kodlar teriladi:

```
1. #pragma endregion
2. bool chizish;
3. private: System::Void Form1_Load(System::Object^ sender,
4. System::EventArgs^ e) {
5. this->Text = "Formada sichqoncha bilan chizish";
6. button1->Text = "Tozalash"; chizish = false; }
7. private: System::Void Form1_MouseDown(System::Object^ sender,
8. System::EventArgs^ e) {
9. // Sichqoncha tugmasi bosilgan bo'lsa chizish
10. chizish = true; }
11. private: System::Void Form1_MouseUp(System::Object^ sender,
12. System::EventArgs^ e) {
13. // Sichqoncha tugmasi qo'yib yuborilganda chizmaslik
14. chizish = false; }
15. private: System::Void Form1_MouseMove(System::Object^ sender,
16. System::EventArgs^ e) {
17. if(chizish==true) { Graphics^ grafika = CreateGraphics();
18. Grafika->FillRectangle(gcnw SolidBrush(Color::Red),
19. e->X, e->Y, 10, 10); delete grafika; }
20. private: System::Void button1_Click(System::Object^ sender,
21. System::EventArgs^ e) { Graphics^ grafika = CreateGraphics();
22. Grafika->Clear(this->BackColor); } };
```

Bu yerda, dasturning boshida false qiymatiga ega bo'lgan mantiqiy turdagi chizish o'zgaruvchisi e'lon qilingan. Ushbu o'zgaruvchi yoki sichqonchani harakatlantirganda (MouseMove hodisasi) formaga chizishni (chizish=true), yoki chizishga ruxsatni man etadi (chizish=false). Sichqonchani tugmasini bosilgan holda

harakatlantirilganda, CreateGraphics() metodidan foydalangan holda dastur nomlar sohasi System::Drawing Graphics grafik ob'ektni yaratadigan va 10x10 piksel o'lchamdagi, qizil rang bilan to'ldirilgan FillRectangle to'g'ri to'rtburchak chizadi. e.X, e.Y - sichqoncha ko'rsatgichining koordinatasi, hamda to'g'ri to'rtburchakning yuqori chap burchak koordinatasi ham hisoblanadi.

Quyidagi rasmda formaga chizish namunasini keltirilgan. Barcl formaga chizilganlarni o'chirish uchun Tozalash tugmasini bosish zarur

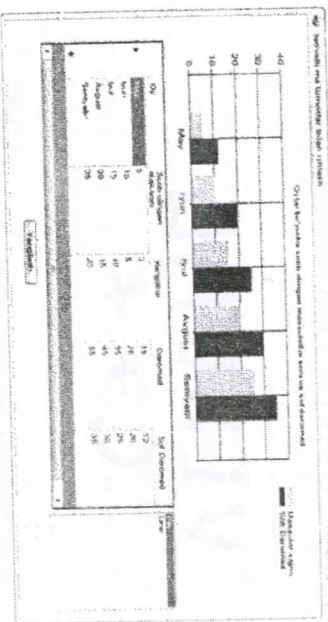


3.36- rasm. Formaga sichqoncha yordamida chizish

3.19. Jadvalli ma'lumotlar asosida Chart komponentasi yordamida grafik diagrammalar yaratish

Jadval tipidagi ma'lumotlar yaratish uchun DataTable tipidagi foydalaniladi. Bu tipda yaratilgan o'zgaruvchi ustun va satrlardan tashqari topadi. Jadval ko'rinishidagi ma'lumotlar asosida grafik diagramma qurish oson bo'ladi. Formaning grafik imkoniyatlarini o'zida jamlagay komponenta bu Chart komponentasidir. Ushbu komponenta yordamida jadvalli ma'lumotlarni ekranga grafik diagrammalar ko'rinishida tasvirlash mumkin. Ushbu mavzuning mazmunini har oyda sotib olingay mahsulotlar miqdorlarini jadvallarda shakllantirib, grafik diagramma ko'rinishida tasvirlovchi vizual dastur yaratish orqali tushuntirish o'tamiz. Topshiriqdagi variantingiz masalasining dasturini tuzish uchun dastur namuna sifatida xizmat qiladi. Lekin diagrammalarda ko'rinishlari va turlari har xil bo'lganligi uchun ushbu turlardan h foydalanib tuzilsa, dasturimizning tashqi ko'rinishlari chiroyli ko'rgazmali bo'ladi.

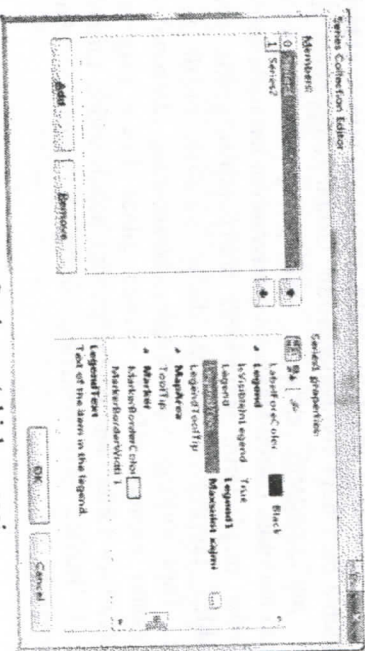
Ushbu vazifani bajaruvchi dasturni tuzish uchun Visual Studio 2012 dasturi ishga tushiriladi, asosiy oyna menyusu bo'limlaridan **File->New->Project...** buyruqlari beriladi yoki **Ctrl+Shift+N** klavishalari bosiladi, ochilgan oynada loyihaga **"Jadval_malumotlar"** nomi berilib, **OK** tugmasi bosiladi. Dasturning umumiy ko'rinishi quyidagicha bo'ladi.



3.37- rasm. Dasturning umumiy ko'rinishi oynasi

Dasturni tuzish quyidagi qadamlardan tashkil topgan:

1-qadam. Formaga **chart** komponentlari joylashtiriladi. Uning **Series** xossasiga kirib, **2-Series** qo'shildi. **Series1** tanlanadi va uning **LegendText** xossasiga **"Maxsulot xajmi"** matni yoziladi. **Series2** tanlanadi va uning **LegendText** xossasiga **"Soft daromat"** matni yoziladi. Ushbu **Series** qo'shish oynasi 3.38- rasmda keltirilgan.



3.38- rasm. Series qo'shish oynasi

2-qadam. Formaga **dataGridView**, **listBox** va **button** komponentlari joylashtiriladi. Ularning hech qanday xossasi o'zgartirilmaydi.

3-qadam. Formaning **load** hodisasi kodlari quyidagicha:

```
1. DataTable ^ Jdval;
2. private: System::Void Form1_Load(System::Object^ sender,
3. System::EventArgs^ e) {chart1->Titles->Add("Oylar bo'yicha sotib
4. olingan maxsulotlar soni va sof daromat");
5. listBox1->Items->Add("Pie"); listBox1->Items->Add("Line");
6. this->Text = "Grafik ma'lumotlar";
7. Jdval = gcnew DataTable();
8. //Ushbu jadvalda 2 ta ustun "Oy" va "Sotib olingan //max.soni";
9. Jdval->Columns->Add("Oy", String::typeid);
10. //C# da: Jdval.Columns.Add("Mesyas", //typeid(String));
11. //2-ustun bo'yicha ma'lumotlar long tipida bo'ladi:
12. Jdval->Columns->Add("Sotib olingan max.soni", Int64::typeid);
13. Jdval->Columns->Add("Xarajatlar", Int64::typeid);
14. Jdval->Columns->Add("Daromat", Int64::typeid);
15. Jdval->Columns->Add("Sof daromat", Int64::typeid);
16. //C# da: Jdval.Columns.Add("Ob'em prodaj", typeid(Long));
17. //Jdvalning 1- qatorini to'ldiramiz:
18. DataRow ^ qator = Jdval->NewRow();
19. qator["Oy"] = "May"; qator["Sotib olingan max.soni"] = 10;
20. qator["Xarajatlar"] = 3; qator["Daromat"] = 15;
21. qator["Sof daromat"] = Convert::ToInt64(qator["Daromat"]);
22. Convert::ToInt64(qator["Xarajatlar"]);
23. //Convert::ToInt64(dataGridView1->Rows[0]->Cells[0]->Value);
24. Jdval->Rows->Add(qator);
25. //Jdvalning 2- qatorini to'ldiramiz:
26. qator = Jdval->NewRow();
27. qator["Oy"] = "Iyun"; qator["Sotib olingan max.soni"] = 20;
28. qator["Xarajatlar"] = 5; qator["Daromat"] = 25;
29. qator["Sof daromat"] = Convert::ToInt64(qator["Daromat"]);
30. Convert::ToInt64(qator["Xarajatlar"]);
31. Jdval->Rows->Add(qator);
32. //3- qatorni qo'shamiz:
33. qator = Jdval->NewRow();
34. qator["Oy"] = "Iyul"; qator["Sotib olingan max.soni"] = 30;
35. qator["Xarajatlar"] = 10; qator["Daromat"] = 35;
36. qator["Sof daromat"] = Convert::ToInt64(qator["Daromat"]);
37. Convert::ToInt64(qator["Xarajatlar"]);
38. Jdval->Rows->Add(qator);
39. //4- qatorni qo'shamiz:
40. qator = Jdval->NewRow();
41. qator["Oy"] = "Avgust"; qator["Sotib olingan max.soni"] = 40;
42. qator["Xarajatlar"] = 15; qator["Daromat"] = 45;
43. qator["Sof daromat"] = Convert::ToInt64(qator["Daromat"]);
44. Convert::ToInt64(qator["Xarajatlar"]);
```



```

45. Jdaval->Rows->Add(qator);
46. //5- qatorni qo'shamiz:
47. qator = Jdaval->NewRow();
48. qator["Oy"] = "Sentyabr"; qator["Sotib olingan max.soni"] = 50;
49. qator["Xarajatlar"] = 20; qator["Daromat"] = 55;
50. qator["Sof daromat"] = Convert::ToInt64(qator["Daromat"]);
51. Convert::ToInt64(qator["Xarajatlar"]);
52. Jdaval->Rows->Add(qator);
53. dataGridView1->DataSource = Jdaval;
54. chart1->DataSource = Jdaval;
55. chart1->Series["Series1"]->XValueMember = "Oy";
56. chart1->Series["Series2"]->XValueMember = "Oy";
57. chart1->Series["Series1"]->YValueMembers = "Sotib olingan max.soni";
58. chart1->Series["Series2"]->YValueMembers = "Sof daromat";
59. chart1->Series["Series1"]->ChartType =
60. System.Windows.Forms.DataVisualization.Charting::
61. SeriesChartType::Column;
62. chart1->Series["Series2"]->ChartType =
63. System.Windows.Forms.DataVisualization.Charting::
64. SeriesChartType::Column;
65. // Diagrammning turi quyidagicha bo'lishi ham
66. // mumkin. masalan: pie, line va bohq.
67. chart1->Series["Series1"]->Color = Color::Aqua;
68. chart1->Series["Series2"]->Color = Color::Yellow;
69. chart1->DataBind(); }

```

Button1 tugmasini click xodisasining kodlari quyidagicha:

```

1. private: System::Void button1_Click(System::Object^ sender,
   System::EventArgs^ e) {dio(); }
2. void dio() {
3. this->Text = "Grafik ma'lumotlar";
4. Jdaval = gcnew DataTable();
5. Jdaval->Columns->Add("Oy", String::typeid);
6. Jdaval->Columns->Add("Sotib olingan max.soni", Int64::typeid);
7. Jdaval->Columns->Add("Xarajatlar", Int64::typeid);
8. Jdaval->Columns->Add("Daromat", Int64::typeid);
9. Jdaval->Columns->Add("Sof daromat", Int64::typeid);
10. DataRow ^ qator = Jdaval->NewRow();
11. qator["Oy"] = dataGridView1->Rows[0]->Cells[0]->Value;
   qator["Sotib
   olingan max.soni"] = dataGridView1->Rows[0]->Cells[1]->Value;
12. qator["Xarajatlar"] = dataGridView1->Rows[0]->Cells[2]->Value;
13. qator["Daromat"] = dataGridView1->Rows[0]->Cells[3]->Value;
14. qator["Sof daromat"] = Convert::ToInt64(dataGridView1->
   Rows[0]->Cells[3]->Value) - Convert::ToInt64(
   dataGridView1->Rows[0]->Cells[2]->Value);
15. dataGridView1->Rows->Add(qator);
16. Jdaval->Rows->Add(qator);
17. Jdaval->Rows->Add(qator);
18. qator = Jdaval->NewRow();

```

128

```

19. qator["Oy"] = dataGridView1->Rows[1]->Cells[0]->Value;
20. qator["Sotib olingan max.soni"] = dataGridView1->Rows[1]-
   >Cells[1]->Value;
21. qator["Xarajatlar"] = dataGridView1->Rows[1]->Cells[2]->Value;
22. qator["Daromat"] = dataGridView1->Rows[1]->Cells[3]->Value;
23. qator["Sof daromat"] = Convert::ToInt64(dataGridView1->
   Rows[1]->Cells[3]->Value) - Convert::ToInt64(
   dataGridView1->Rows[1]->Cells[2]->Value);
24. dataGridView1->Rows->Add(qator); qator = Jdaval->NewRow();
25. Jdaval->Rows->Add(qator); qator = Jdaval->NewRow();
26. qator["Oy"] = dataGridView1->Rows[2]->Cells[0]->Value;
27. qator["Sotib
   olingan max.soni"] = dataGridView1->Rows[2]->Cells[1]->Value;
28. qator["Xarajatlar"] = dataGridView1->Rows[2]->Cells[2]->Value;
29. qator["Daromat"] = dataGridView1->Rows[2]->Cells[3]->Value;
30. qator["Sof daromat"] = Convert::ToInt64(
   dataGridView1->Rows[2]->Cells[3]->Value) -
   Convert::ToInt64(
   dataGridView1->Rows[2]->Cells[2]->Value);
31. Convert::ToInt64(dataGridView1->Rows[2]->Cells[2]->Value);
32. Jdaval->Rows->Add(qator); qator = Jdaval->NewRow();
33. qator["Oy"] = dataGridView1->Rows[3]->Cells[0]->Value;
34. qator["Sotib olingan max.soni"] = dataGridView1->Rows[3]-
   >Cells[1]->Value;
35. qator["Xarajatlar"] = dataGridView1->Rows[3]->Cells[2]->Value;
36. qator["Daromat"] = dataGridView1->Rows[3]->Cells[3]->Value;
37. qator["Sof daromat"] = Convert::ToInt64(dataGridView1->Rows[3]-
   >Cells[3]->Value) - Convert::ToInt64(
   dataGridView1->Rows[3]->Cells[2]->Value);
38. dataGridView1->Rows->Add(qator); qator = Jdaval->NewRow();
39. Jdaval->Rows->Add(qator); qator = Jdaval->NewRow();
40. qator["Oy"] = dataGridView1->Rows[4]->Cells[0]->Value;
   qator["Sotib olingan max.soni"] = dataGridView1->Rows[4]-
   >Cells[1]->Value;
41. qator["Xarajatlar"] = dataGridView1->Rows[4]->Cells[2]->Value;
42. qator["Daromat"] = dataGridView1->Rows[4]->Cells[3]->Value;
43. qator["Sof daromat"] = Convert::ToInt64(
   dataGridView1->Rows[4]->Cells[3]->Value) - Convert::ToInt64(
   dataGridView1->Rows[4]->Cells[2]->Value);
44. Jdaval->Rows->Add(qator);
45. // chart1 komponentasiga Jdavalni eksport qilish
46. chart1->DataSource = Jdaval;
47. //chart1 komponentasidagi Series1 ning x kordinatasiga Oy so'zini
   yozish
48. chart1->Series["Series1"]->XValueMember = "Oy";
49. // chart1 komponentasidagi Series1 ning x kordinatasiga Oy
   so'zini yozish
50. chart1->Series["Series2"]->XValueMember = "Oy//chart1
   komponentasidagi
   Series1 ning y kordinatasiga Sotib olingan max.soni so'zini
   yozish
51. chart1->Series["Series1"]->YValueMembers = "Sotib olingan
   max.soni";

```

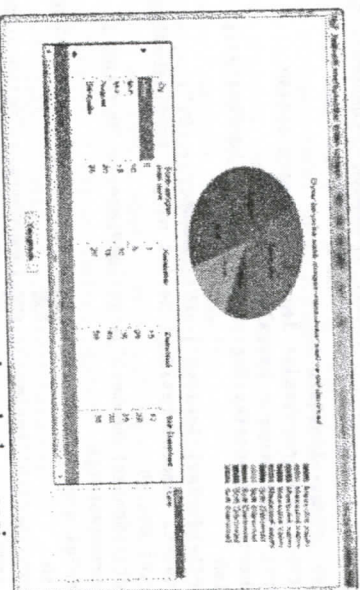
129


```

53. // chart1 komponentasidagi Serie2 ning y kordinatasiga Sof
daromat so'zini yozish
54. chart1->Series["Series2"]->YValueMembers = "Sof daromat";
55. chart1->Series["Series1"]->ChartType = System::Windows::Forms::
56. DataVisualization::Charting::SeriesChartType::Column;
57. chart1->Series["Series2"]->ChartType = System::Windows::Forms::
58. DataVisualization::Charting::SeriesChartType::Column;
59. chart1->Series["Series1"]->Color = Color::Aqua;
60. chart1->Series["Series2"]->Color = Color::Green;
61. //chart1->Series["Series1"]->IsVisibleInLegend = false;
62. chart1->DataBind();
63. //dataGridview1 komponentasiga jadvalni eksport qilish
64. dataGridview1->DataSource = jadval;
}
4-qadam. Jadvaldan kiritilgan qiymatlar natijaga darxol ta'sir
qilishi uchun dataGridview1 ning CellEndEdit xodisasiga dio()
funksiyasi chaqiriladi. Ahamiyat berilgan bo'lsa, button1 ga ham
chaqirilgan edi. Ushbu kod quyidagicha:
1. private: void dataGridview1_CellEndEdit(System::Object^
2. sender, System::Windows::Forms::DataGridViewCellEventArgs^ e) {
3. dio();
}
5-qadam. listBox1da diagrammani ko'rinishlarini o'zgartirish
mumkin. Buning uchun listBox ning SelectedIndexChanged
hodisasining kodlari quyidagicha bo'lishi kerak:
1. private: System::Void
listBox1_SelectedIndexChanged(System::Object^
2. sender, System::EventArgs^ e) {
3. if (listBox1->Text == "Transparent") return;
4. this->chart1->Series["Series1"]->ChartypeName=(listBox1->Text) ;
5. this->chart1->Series["Series2"]->ChartypeName=(listBox1->Text) ;
}

```

Dasturda diagrammani ko'rinishi quyidagicha:



3.39- rasm. Diagrammani sozlash oynasi

130

3.20. Veb browserda HTML jadvallarni tasvirlash va shakllantirish

Bizga ma'lumki HTML teglari yordamida veb ilovalar yaratiladi va bu ilovalar jamlanib, veb sayt tashkil etiladi. Veb ilovalarni yoki global tarmogga ulangan saytlarni amaliy dasturda ochish imkonini beruvchi komponenta bu WebBrowser komponentasi hisoblanadi. Ushbu komponenta orqali nafaqat saytlarni ochish, balki HTML kodlarida terilgan tekst (teg) larni ham chiqarish va tahrirlash imkoniyatlari mavjud. Quyidagi namunada tuziladigan dasturda HTML teglari ishlatilgan. Masalan: `<h1>-matnini 16 shriftida ko'rsatadi. <table>- jadval yaratadi. <tr>-jadvalning ustunini yaratadi. <td>- jadvalning satrlarini yaratadi. 8- jadvalda keltirilgan ma'lumotlar manli faylda tashkil qilinadi. Ushbu fayl o'qish uchun ochiladi va uning ma'lumotlari ma'lum o'zgaruvchilarga o'zlashtirilgan holda, sikl asnosida webBrowser komponentasi yordamida ekranga tartiblangan ko'rinishida chiqariladi.`

Ushbu vazifani bajaruvchi dasturni tuzish uchun Visual Studio 2012 dasturi ishga tushiriladi, asosiy oyna menu bo'limlaridan `File->New->Project...` buyruqlari beriladi yoki `Ctrl+Shift+N` klavishalari bosiladi, ochilgan oynada loyihaga `"HTML_xujjatlari"` nomi berilib OK tugmasi bosiladi. Dasturning umumiy ko'rinishi quyidagicha bo'ladi:

Nomen	Qar_nasi	Ismi	Tel_nomeri
1	o_zin	Malayev Oybek	5642925
2	Ukran	Malayev Nozibek	4443355
3	Akran	Malayev Ombek	1112233

3.30- rasm. Dasturning umumiy ko'rinish oynasi

131

Dasturni tuzish quyidagi qadamlardan tashkil topgan:

1-qadam. Formaga webBrowser komponentasini tashlang va

Formalning load hodisasining kodlari quyidagicha:

```
1. #pragma endregion
2. private: System::Void Form1_Load(System::Object^ sender,
3. System::EventArgs^ e) {
4. String^ s = <h1 align="center">O'lim haqi ma'lumot<br><table
5. border="1">";
6. s += <tr><td>Nomer</td>; s += <td>Qar turi</td>;
7. s += <td>Fam</td>; s += <td>Ismi</td>;
8. s += <tr><td>Tel nomer</td>;
9. char qar_turi[20] = ""; char Ismi[20] = "";
10. char Fam[20] = ""; char Tel[20] = "";
11. String^ ss = ""; FILE *oqi; int i=1, n; bool dd;
12. oqi = fopen("C:\\1.txt", "r");
13. if(oqi == NULL){MessageBox::Show("Not found");}
14. while(!feof(oqi)){
15. fgetc(qar_turi, 20, oqi); fgetc(Fam, 20, oqi);
16. fgetc(Ismi, 20, oqi); dd = fgetc(Tel, 20, oqi);
17. if(dd == 0){break;}
18. String^ Fq = gcnew String(qar_turi);
19. String^ Fi = gcnew String(Ismi);
20. String^ Ft = gcnew String(Fam);
21. s += <tr><td>+i.ToString() + <td>; s += <td>+Fq + <td>;
22. s += <td>+Fi + <td>; s += <td>+Ft + <td>;
23. s += <td>+Ft + <td>; i++; s += </table>;
24. webBrowser1->Navigate("about:" + s); }
```

2-qadam. Formada saytlarni qidirish uchun Button va TextBox komponentalari joylashtirildi va Button1 tugmasining click hodisasining kodlari quyidagicha:

```
1. private: System::Void Button1_Click(System::Object^ sender,
2. System::EventArgs^ e) { if (!this->textBox1->Text->Equals("")) {
3. this->webBrowser1->Navigate( this->textBox1->Text ); }
```

3-qadam. Manzillar oynasiga web sayt manzili yoziladigan TextBox1 hodisasi quyidagicha bo'ladi:

```
1. private: System::Void textBox1_KeyDown(System::Object^ sender,
2. System::Windows::Forms::EventArgs^ e) {
3. if ( e->KeyCode == System::Windows::Forms::Keys::Enter &&
4. !this->textBox1->Text->Equals( "" ) ) { this->webBrowser1->
5. Navigate(
6. this->textBox1->Text ); }
```

132

4-qadam. webBrowser1 komponentasining Navigated hodisasining kodlari quyidagicha:

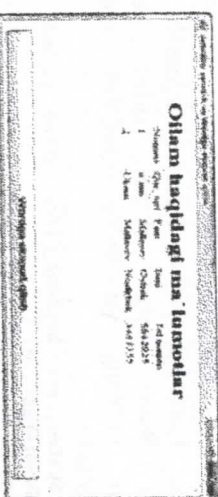
```
1. private: System::Void webBrowser1_Navigated(System::Object^
2. sender, System::Windows::Forms::WebBrowserNavigatedEventArgs^ e)
3. {
4. this->textBox1->Text = this->webBrowser1->Url->ToString(); }
```

Agar kompyuter internetga ulangan bo'lsa, sayt manzili maydoniga yozilgan sayt ochiladi.

3.21. Visual C++da MS Word imkoniyatlaridan foydalanib, jadvallar yaratish, ularni Word fayliga eksport qilish va taqdim etish

Visual C++ da Microsoft MS Office paketidagi amaliy dasturlarning imkoniyatlaridan foydalanish imkoniyatlari mavjud. Ushbu qo'llanmaning 2-bobida Visual Basic tilidagi InputBox kiritish oynasi imkoniyatlaridan foydalanilgan edi. Ushbu bo'limga esa Microsoft MS Office ning jadval yaratish usullari imkoniyatlaridan qanday foydalanish yo'llari o'rganiladi. Ish Visual Studio 2012 dasturini ishga tushirishdan boshlanadi. File menu bo'limidan New Project tanladi va yangi loyiha yaratiladi. Ushbu loyihaga MS Word bo'limidan Add Reference buyrug'i tanlanadi yoki Alt+F7 tugmalari boshiladi. Ochilgan Property Pages oynadan Com bo'limiga o'tiladi va Add new Reference tugmasi bosiladi. Ochilgan oynadan Microsoft Word 14.0 Object Library 8.5 belgilanadi va OK tugmachasi bosiladi. So'ngra ushbu kutubxona muvaffaqiyatli qo'shilganligi haqida ma'lumot chiqadi. Natijada, dasturda ushbu kutubxona funksiyalaridan foydalanish imkoniyati vujudga keladi.

Dasturning umumiy ko'rinishi quyidagicha bo'ladi:



3.31- rasm. Dasturning umumiy ko'rinishi oynasi

133

1-qadam. **Form1ga** asosiy dastur oynasida matnli fayl ma'lumotlarini chiqarish uchun **webBrowser** komponentasi tashlanadi va **Form1** ning **load** hodisasi kodlari quyidagicha bo'ladi:

2-qadam. **Formlga** asosiy dastur oynasida matnli fayl ma'lumotlarini Word faylga eksport qilish uchun **Button** komponentasini tashlang va uning **OnClick** xodisasi kodlarini quyidagicha bo'ladi:

```
1. private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
2.     // 4 ta satrli massiv e'lon kilish
3.     array<String>^ f1 = ["tsini", "tsina", "tsind", "tsini"];
```

[illegible]


```

52. default->InsertAfter(Fq[i - 1]);
53. Vord1->ActiveDocument->Tables[1]->Cell(1, 3)->
54. default->InsertAfter(Fq[1 - 1]);
55. Vord1->ActiveDocument->Tables[1]->Cell(1, 4)->
56. default->InsertAfter(Fq[1 - 1]);
57. Vord1->ActiveDocument->Tables[1]->Cell(1, 5)->
58. default->InsertAfter(Fq[1 - 1]);
59. // C# tilida, quyidagicha yoziladi:
60. // Vord1.ActiveDocument.Tables[1].Cell(1, 2).
61. // Range.InsertAfter(Tel[i - 1]);
62. }
63. // MS Word xujjatlari ilovalarini belgilash:
64. Object t3=Microsoft::Office::Interop::Word::MdwUnits::wdline;
65. //MS Word xujjatiga 1 ta satrni aniqlash :
66. Object ^ stroka9 = t3;
67. Vord1->Selection->MoveDown(t3, stroka9, t);
68. // Quyidagi matnni chop qilish:
69. Vord1->Selection->Text("Kafedra mudiri N.Rahimov");
70. /*xujjatni avtomatik saqlash, lekin buni foydalanuvchi hal
qilgan maqul*/:
71. Object ^ ImyaFayla = "C:\Matija.doc";
72. Vord1->ActiveDocument->SaveAs(ImyaFayla, t, t, t, t, t, t, t);
}

```

Dastur tahlili:

Ushbu dastur kodida 4 ta String tipidagi massiv e'lon qilindi. Dasturning kutubxona e'lon qilish bo'limiga fayllar bilan ishlash direktivasi (#include<fstream>) qo'shildi.

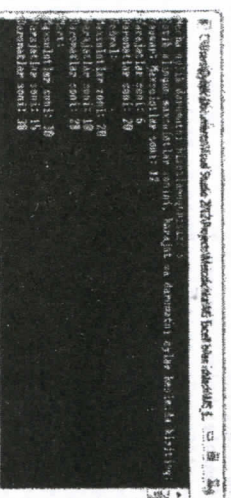
Word::Application sinf ob'ekti yaratildi. (auto Vord1 = new Microsoft::Office::Interop::Word::Application(); va uning Add metodidan foydalanib Word fayl ochildi. Matnli fayldagi ma'lumotlarni FILE tipidagi ko'rsatgich yordamida 4 ta massivga o'zlashtirildi. Massivdagi ma'lumotlar (Vord1->ActiveDocument->Tables[1]->Cell(1,2)->default->InsertAfter(Fq[i - 1]); faylga o'zlashtirildi.

3.22. Visual C++ da MS Excell imkoniyatlaridan foydalanib, diagrammalar yaratish va ularni turli kengaytmalarda saqlash

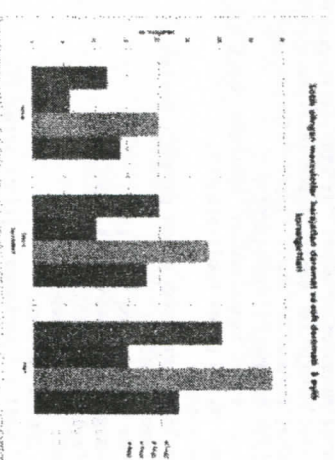
Visual C++ da Microsoft MS Office paketidagi amaliy dasturlarning imkoniyatlaridan foydalanish imkoniyatlari mavjud. Jumladan, jadvallar redaktori MS Excell da murakkab hisoblashlarni amalga oshirish imkoniyatlari mavjud. Ya'ni, dasturlash imkoniyatlarini ham oshiradi. Ushbu imkoniyatlardan qanday foydalanishni ushbu

bo'limda ko'rib chiqildi. Jumladan, Microsoft MS Excell ning diagrammalar yaratish usullari imkoniyatlaridan foydalanish yo'llari o'rganiladi. Ba'zi hisoblashlarni diagrammalar orqali namoyish qilish dasturning vizualligini o'rtiradi. Bu esa har doim MS Excell ning ma'lumotlari o'zgaranda vizual diagrammani ham o'zgaraganini ko'rish mumkin. Ish Visual Studio dasturini ishga tushirishdan boshlanadi. File menu bo'limidan New Project tanladi va yangi loyiha yaratiladi. Ushbu loyihaga MS Excell ning ob'ektlar kutubxonasi qo'shildi. Buning uchun Project menu bo'limidan Add Reference burug'i tanlanadi yoki Alt+F7 tugmalari bosiladi. Ochilgan Property Pages oynasidan Com bo'limiga o'tiladi va Add new Reference tugmasi bosiladi. Ochilgan oynadan Microsoft Excell 14.0 Object Library 8.5 belgilanadi va OK tugmachasi bosiladi. So'ngra ushbu kutubxona muvaffaqiyatli qo'shilganligi haqida ma'lumot chiqadi. Natijada, dasturda ushbu kutubxona funksiyalaridan foydalanish imkoniyati vujudga keladi.

Dasturning umumiy ko'rinishi quyidagicha bo'ladi:



3.32- rasm. Dasturning kompilyatsiya vaqtidagi oynasi



3.33- rasm. Dasturning kompilyatsiyadan keyingi diagramma oynasi

Dasturni tuzish quyidagi qadamlardan tashkil topgan:

1-qadam. Dasturga forma muhiti talab qilmaganligi uchun loyihani **Console Application** da tuziladi. **Visual C++** dasturi ishg'a tushiriladi. **File** menyusu bo'limidan **New Project** taniladi, **CLR** qismidan **Console Application CLR** muhitida yangi loyiha yaratiladi. MS Excell kutubxonasi qo'shiliadi (**Add Reference->Com->Add new Reference->Microsoft Excell 14.0 Object Library 8.5**). Dastur kodlari quyidagicha bo'ladi:

```
1. // MS_Excell_bilan_ishlash.cpp : main project file.
2. #include "stdafx.h"
3. #include "iostream"
4. using namespace System;
5. using namespace Microsoft::Office::Interop::Excel;
6. using namespace std;
7. int main(array<String^> args) {
8.     double harajat[12], daromat[12], maxsulot_soni[12];
9.     int n;
10.    Console::Write("Necha oylik daromatni hisoblamochisiz?");
11.    Console::WriteLine("Sotib olingan maxsulotlar sonini, harajat va daromatni oylar kesimida kiriting:");
12.    if(n<=12){
13.        for(int i=0;i<n;i++){
14.            if(i==0){Console::Write("Yanvar: ");}
15.            else if(i==1){Console::WriteLine("Febrl: ");}
16.            else if(i==2){Console::WriteLine("Mart: ");}
17.            else if(i==3){Console::WriteLine("Aprel: ");}
18.            else if(i==4){Console::WriteLine("May: ");}
19.            else if(i==5){Console::WriteLine("Iyun: ");}
20.            else if(i==6){Console::WriteLine("Iyul: ");}
21.            else if(i==7){Console::WriteLine("Avgust: ");}
22.            else if(i==8){Console::WriteLine("Sentyabr: ");}
23.            else if(i==9){Console::WriteLine("Oktabr: ");}
24.            else if(i==10){Console::WriteLine("Noyabr: ");}
25.            else if(i==11){Console::WriteLine("Dekabr: ");}
26.            Console::Write("Maxsulotlar soni: "); std::cin>>maxsulot_soni[i];
27.            Console::Write("Daromatlar soni: "); std::cin>>daromat[i];
28.            Sof_daromat[i]= daromat[i]-harajat[i];
29.            if(i==n-1){ // Excell::Application klasidan nusxa olish:
30.                Application ^ Xl1 = gcnew Application();
31.                Xl1->Visible = true;
32.                // mavjud usulda foydalanish uchun parametrlar berish:
33.                Object^ t = Type::Missing;
34.                // MS Excel da yangi varoq yaratish:
35.                Workbook ^ Kniiga = Xl1->Workbooks->Add(t);
36.                // Varoqni faylda e'lon qilish:
37.                Sheets ^ listi = Kniiga->Worksheets;
38.                // Birinchi listni tanlash:
39.                Worksheet ^ list = (Worksheet^)(listi->Item[1]);
40.                // Agar yana bitta list qo'shishni kerak bo'lsa quyidagi
41.                // kodni terish lozim:
```

138

```
42. // Worksheet^list = safe_cast<Worksheet^>(list->Item[0]);
43. // Worksheetni joyi parametrlari bilan Diagramma yaratishga buyurtma berish:
44. Chart ^ Grafik = (Chart^)(Xl1->Charts->Add(t, t, t));
45. if(n>1){
46.     list->Range["A1", t]->Value2 = "Oylar nomi";
47.     list->Range["A2", t]->Value2 = "Yanvar";
48.     list->Range["B1", t]->Value2 = "Sotib olingan maxsulot soni";
49.     list->Range["B2", t]->Value2 = maxsulot_soni[0];
50.     list->Range["C1", t]->Value2 = "Harajatlari";
51.     list->Range["C2", t]->Value2 = harajat[0];
52.     list->Range["D1", t]->Value2 = "Daromatlar";
53.     list->Range["D2", t]->Value2 = daromat[0];
54.     list->Range["E1", t]->Value2 = "Soft Daromatlar";
55.     list->Range["E2", t]->Value2 = Sof_daromat[0];
56.     //Grafika qurish uchun qiymatlar diapazonini berish:
57.     Grafik->SetSourceData(list->Range["A2", "E2"]);
58.     XlRowCol::XlColumns); }
59. if(n>2){list->Range["A3", t]->Value2 = "Febrl";
60.     list->Range["B3", t]->Value2 = maxsulot_soni[1];
61.     list->Range["C3", t]->Value2 = harajat[1];
62.     list->Range["D3", t]->Value2 = daromat[1];
63.     list->Range["E3", t]->Value2 = Sof_daromat[1];
64.     //Grafika qurish uchun qiymatlar diapazonini berish:
65.     Grafik->SetSourceData(list->Range["A2", "E3"]);
66.     XlRowCol::XlColumns); }
67. if(n>3){
68.     list->Range["A4", t]->Value2 = "Mart";
69.     list->Range["B4", t]->Value2 = maxsulot_soni[2];
70.     list->Range["C4", t]->Value2 = harajat[2];
71.     list->Range["D4", t]->Value2 = daromat[2];
72.     list->Range["E4", t]->Value2 = Sof_daromat[2];
73.     //Grafika qurish uchun qiymatlar diapazonini berish:
74.     Grafik->SetSourceData(list->Range["A2", "E4"]);
75.     XlRowCol::XlColumns); }
76. if(n>4){list->Range["A5", t]->Value2 = "Aprel";
77.     list->Range["B5", t]->Value2 = maxsulot_soni[3];
78.     list->Range["C5", t]->Value2 = harajat[3];
79.     list->Range["D5", t]->Value2 = daromat[3];
80.     list->Range["E5", t]->Value2 = Sof_daromat[3];
81.     //Grafika qurish uchun qiymatlar diapazonini berish:
82.     Grafik->SetSourceData(list->Range["A2", "E5"]);
83.     XlRowCol::XlColumns); }
84. if(n>5){
85.     list->Range["A6", t]->Value2 = "May";
86.     list->Range["B6", t]->Value2 = maxsulot_soni[4];
87.     list->Range["C6", t]->Value2 = harajat[4];
88.     list->Range["D6", t]->Value2 = daromat[4];
89.     list->Range["E6", t]->Value2 = Sof_daromat[4];
90.     //Grafika qurish uchun qiymatlar diapazonini berish:
91.     Grafik->SetSourceData(list->Range["A2", "E6"]);
92.     XlRowCol::XlColumns); }
93. if(n>6){
94.     list->Range["A7", t]->Value2 = "Iyun";
95.     list->Range["B7", t]->Value2 = maxsulot_soni[5];
```

139


```

96. list->Range["C7", t]->Value2 = harajat[5];
97. list->Range["D7", t]->Value2 = daromat[5];
98. list->Range["E7", t]->Value2 = Sof_daromat[5];
99. //Grafika qurish uchun qiyimatlar diapazonini berish:
100. Grafik->SetSourceData(list->Range["A2", "E7"]);
101. XlRowCol::xlColumns); }
102. if(n==7){
103. list->Range["A8", t]->Value2 = "Iyul";
104. list->Range["B8", t]->Value2 = maxsulot_soni[6];
105. list->Range["C8", t]->Value2 = harajat[6];
106. list->Range["D8", t]->Value2 = daromat[6];
107. list->Range["E8", t]->Value2 = Sof_daromat[6];
108. //Grafika qurish uchun qiyimatlar diapazonini berish:
109. Grafik->SetSourceData(list->Range["A2", "E8"]);
110. XlRowCol::xlColumns); }
111. if(n==8){
112. list->Range["A9", t]->Value2 = "Avgust";
113. list->Range["B9", t]->Value2 = maxsulot_soni[7];
114. list->Range["C9", t]->Value2 = harajat[7];
115. list->Range["D9", t]->Value2 = daromat[7];
116. list->Range["E9", t]->Value2 = Sof_daromat[7];
117. //Grafika qurish uchun qiyimatlar diapazonini berish:
118. Grafik->SetSourceData(list->Range["A2", "E9"]);
119. XlRowCol::xlColumns); }
120. if(n==9){
121. list->Range["A10", t]->Value2 = "Sentyabr";
122. list->Range["B10", t]->Value2 = maxsulot_soni[8];
123. list->Range["C10", t]->Value2 = harajat[8];
124. list->Range["D10", t]->Value2 = daromat[8];
125. list->Range["E10", t]->Value2 = Sof_daromat[8];
126. //Grafika qurish uchun qiyimatlar diapazonini berish:
127. Grafik->SetSourceData(list->Range["A2", "E10"]);
128. XlRowCol::xlColumns); }
129. if(n==10){
130. list->Range["A11", t]->Value2 = "Oktabr";
131. list->Range["B11", t]->Value2 = maxsulot_soni[9];
132. list->Range["C11", t]->Value2 = harajat[9];
133. list->Range["D11", t]->Value2 = daromat[9];
134. list->Range["E11", t]->Value2 = Sof_daromat[9];
135. //Grafika qurish uchun qiyimatlar diapazonini berish:
136. Grafik->SetSourceData(list->Range["A2", "E11"]);
137. XlRowCol::xlColumns); }
138. if(n==11){
139. list->Range["A12", t]->Value2 = "Noyabr";
140. list->Range["B12", t]->Value2 = maxsulot_soni[10];
141. list->Range["C12", t]->Value2 = harajat[10];
142. list->Range["D12", t]->Value2 = daromat[10];
143. list->Range["E12", t]->Value2 = Sof_daromat[10];
144. //Grafika qurish uchun qiyimatlar diapazonini berish:
145. Grafik->SetSourceData(list->Range["A2", "E12"]);
146. XlRowCol::xlColumns); }
147. if(n==12){
148. list->Range["A13", t]->Value2 = "Dekabr";
149. list->Range["B13", t]->Value2 = maxsulot_soni[11];

```

140

```

150. list->Range["C13", t]->Value2 = harajat[11];
151. list->Range["D13", t]->Value2 = daromat[11];
152. list->Range["E13", t]->Value2 = Sof_daromat[11];
153. //Grafika qurish uchun qiyimatlar diapazonini berish:
154. Grafik->SetSourceData(list->Range["A2", "E13"]);
155. XlRowCol::xlColumns); }
156. //Grafikni turini berish("stolbikova diagramma" (gistogramma));
157. Grafik->ChartType = xlChartType::xlColumnClustered;
158. //Grafikni legendasini ekranga chiqarish:
159. Grafik->HasLegend = true;
160. // Grafikning sarlavhasini ekranga chiqarish:
161. Grafik->HasTitle = true;
162. Grafik->ChartTitle->Caption = "Sotib olingan maxsulotlar";
163. harajatlarni daromat va sotil daromatni "n.ToString()+" oylik
164. korsetgichlarni";
165. // o'si x bo'yicha imzo:
166. Axis ^ GorizontalnayaOs = (Axis^)Grafik->Axes(
167. XlAxisType::xlCategory, XlAxisGroup::xlPrimary);
168. GorizontalnayaOs->HasTitle = true;
169. GorizontalnayaOs->HasMajorGridlines=true;
170. GorizontalnayaOs->AxisTitle->Text = "Oy nomlari";
171. // y o'qi bo'yicha imzo:
172. Axis ^ VertikalnayaOs = (Axis^)Grafik->Axes(
173. XlAxisType::xlValue, XlAxisGroup::xlPrimary);
174. VertikalnayaOs->HasTitle = true;
175. VertikalnayaOs->AxisTitle->Text = "Ko'rsatkichlar";
176. //Grafikni quyidagi kengaytmali fayl ko'rinishida saqlash:
177. XlIActiveChart->Export("C:\g\ExcelGrafik.jpg",t,t);
178. } } else {Console::Write("1 yilda 12 oy bor
179. Qaytadan ununib ko'ring"); goto bir;} return 0; }

```

Dastur tahlili:

Dasturda C++ ning kiritish (cin) operatori, Visual C++ ning chiqarish (Console::WriteLine()) operatori, Excel::Application sinfidan nusxa olish(Application ^ Xl1 = new ApplicationO;) operatori, MS Excell da yangi varoq yaratish (Workbook) operatori, varaqni faylda e'lon qilish (Sheets) operatori, Excell listini tanlash (Worksheet) operatori, diagrammani yaratish (Chart) operatori va diagrammani turli parametrlarini o'zgartiruvchi bir nechta funktsiyalar ishlatilgan. Algoritmni quyidagicha: 4 ta massiv (harajat[12], daromat[12], Sof_daromat[12], maxsulot_soni[12]) yaratilgan, ushbu massivlarga konsol muhitidan qiymat o'zlashtiriladi, ushbu massiv qiymatlarini Excell ning varag'idagi katakchalarga tartib bilan yoziladi, kataklar diapazoni belgilanadi va (Chart) operatori yordamida grafik quriladi.

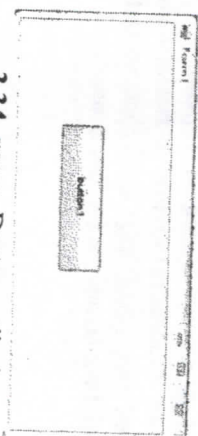
141

3.23. Visual C++ ning Windows Application muhitida komponentalarning joylashish vaziyatlarini nazorat qilish

I-dastur: Formada komponentalar dasturning tashqi ko'rinishlarini, unumdorligini oshirishda muhim ahamiyat kasb etadi. Bizga ma'lumki, dasturda ishlatiladigan komponentalarning joylashish pozitsiyalarini kompilyatsiyadan oldin o'zgartirish mumkin. Visual C++ muharririda kompilyatsiyadan keyin ham o'zgartirish imkoniyati mavjud. Hozirgi yaratiladigan dastur, **Button** komponentasini forma bo'ylab ko'chirish imkoniyatlarini ochib beradi. Buning uchun **Windows Form Application** muhitida yangi loyihani yaratiladi. Formaga bitta **Button** komponentasi joylashtiriladi. Dastur kodlari quyidagicha:

```
1. #pragma endregion
2. bool Peremehenie;
3. int MouseDownX; int MouseDownY;
4. private: System::Void Form1_Load(System::Object^ sender,
5. System::EventArgs^ e) { Peremehenie = false; }
6. private: System::Void button1_MouseDown(System::Object^ sender,
7. System::EventArgs^ e) { if (e->Button ==
8. System::Windows::Forms::MouseEventArgs::Left) {
9. MouseButtons::Left) { Peremehenie = true; MouseDownX = e->X;
10. MouseDownY = e->Y; } }
11. private: System::Void button1_MouseUp(System::Object^
12. sender, System::EventArgs^ e) { Peremehenie = false; }
13. MouseButtons::Left) { Peremehenie = false; }
14. private: System::Void button1_MouseMove(System::Object^
15. sender, System::EventArgs^ e) {
16. if (Peremehenie == true) {
17. auto Tochka = System::Drawing::Point();
18. Tochka.X = this->button1->Location.X + (e->X - MouseDownX);
19. Tochka.Y = this->button1->Location.Y + (e->Y - MouseDownY);
20. this->button1->Location = Tochka; } }
```

Dasturning tashqi ko'rinishi quyidagicha:



3.34- rasim. Dastur natijasi

Dastur tahlili:

Dasturda uchta global o'zgaruvchilar (**bool Peremehenie; int MouseDownX; int MouseDownY;**) e'lon qilindi. **bool Peremehenie** – mantiqiy o'zgaruvchi bo'lib, komponentani ko'chirishga ruxsat bor yoki yo'qligini aniqlash uchun ishlatildi. Agar foydalanuvchi **Button1** tugmasini sichqonchani chap tugmasi bilan bossa (**MouseDown, e->Button == Left**), u holda **Peremehenie = true** bo'ladi. Bu esa **Button1** tugmasini ko'chirish mumkinligini anglatadi. Agar foydalanuvchi sichqonchani tugmasini qo'yib yuborsa, (**Vaziyat MouseUp**) u holda **Peremehenie = true** bo'ladi. Qolgan 2 ta (**int MouseDownX; int MouseDownY;**) o'zgaruvchilar joriy vaziyatni to'ldirish uchun mo'ljallangan.

2-dastur: Operatsion tizimning maxsus ovozi fayllarini yuklovchi vizual dastur

Bizga ma'lumki, operatsion tizimda ishlaganda xatoliklar, amallarni yoki operatsiyalarni bajarilishi, dasturlarni o'rnatish va x.k.lar haqida ovozlarni eshitiлади. Ushbu ovozi fayllardan joriy dasturimizda foydalanishga harakat qilinadi. Buning uchun **Windows Form Application** muhitida yangi loyihani yaratiladi. Formaga 6 ta **Button** komponentasi joylashtiriladi. Dastur kodlari quyidagicha bo'ladi:

```
1. #pragma endregion
2. System::Media::SoundPlayer ^ Plier;
3. private: System::Void Form1_Load(System::Object^ sender,
4. System::EventArgs^ e) { this->Text = "Zvukovo'e signalo" OS";
5. Plier = gcnew System::Media::SoundPlayer();
6. button1->Text = "Zvuk torjesta \ta-dag"";
7. button2->Text = "Zvuk zaversheniya prosessa";
8. button3->Text = "Zvuk oshibki";
9. button4->Text = "Chastota zvuka = 1000 gs";
10. button5->Text = "Vxod v Windows XP";
11. button6->Text = "Vo'xod iz Windows XP";
12. private: System::Void button1_Click(System::Object^ sender,
13. System::EventArgs^ e) {
14. Plier->SoundLocation = "C:\\windows\\media\\tada.wav";
15. Plier->Play();
16. private: System::Void button2_Click(System::Object^ sender,
17. System::EventArgs^ e) {
18. System::Media::SystemSounds::Asterisk->Play();
19. private: System::Void button3_Click(System::Object^ sender,
20. System::EventArgs^ e) { }
```

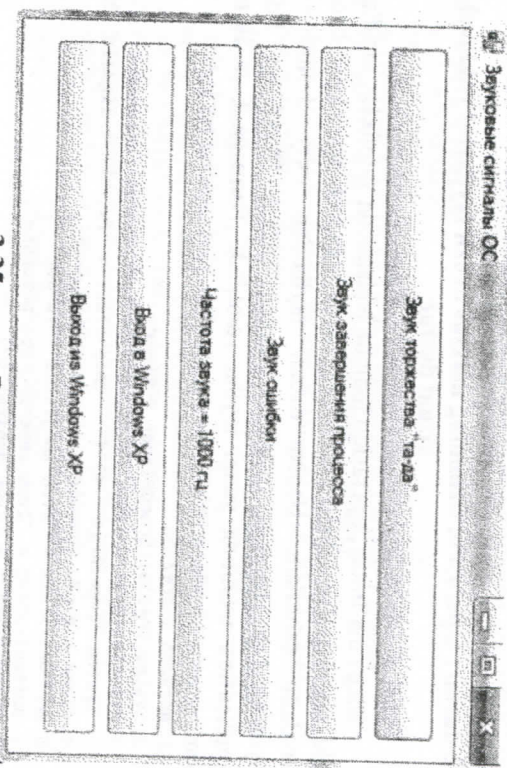


```

20. System::Media::SystemSounds::Beep->Play();
21. private: System::Void button4_Click(System::Object^ sender,
22. System::EventArgs^ e) {
23. Console::Beep(1000, 500);
24. private: System::Void buttons_Click(System::Object^ sender,
25. System::EventArgs^ e) {
26. Pleeer->SoundLocation = "C:\\Windows\\Media\\Windows
Startup.wav";
27. Pleeer->Play();
28. private: System::Void button6_Click(System::Object^ sender,
29. System::EventArgs^ e) {
30. Pleeer->SoundLocation = "C:\\Windows\\Media\\Windows User Account
Control.wav";
31. Pleeer->Play(); }

```

Dasturning tashqi ko'rinishi quyidagicha bo'ladi:



3.35- rasm. Dastur natijasi

3-dastur: Operasion tizinning video fayllarini yuklovchi dastur.

Ushbu dasturni tuzishda Windows Media Player dasturi ishlatiladi. Buni amalga oshirish uchun Windows Form Application muhitida yangi loyiha yaratiladi. Formaga menuStrip va Windows Media Player komponentlari joylashtiriladi. Lekin ToolStrip asboblari panelida Windows Media Player komponentasi yo'q. Uni o'rnatish uchun ToolStrip asboblari panelida sichqonchaning o'ng tugmasi

bosiladi va kontekst menyudan Choose Items buyrug'i tanlanadi. Ochiqilgan oynaning COM bo'limidan Windows.Media.Player ni tanlab, OK tugmachasi bosiladi. Natijada ToolStrip asboblari panelida ushbu komponenta paydo bo'ladi. menuStrip komponentasini sozlash 3.15- rasmida keltirilgan.

Dastur kodlari quyidagicha:

```

1. #pragma endregion
2. OpenFileDialog ^ openFileDialog1;
3. private: System::Void Form1_Load(System::Object^ sender,
4. System::EventArgs^ e) { openFileDialog1 = gcnew OpenFileDialog();
5. // PLEERni versiyasi
6. this->Text = "Windows Media Player, versiya = " +
7. axWindowsMediaPlayer1->VersionInfo; }
8. private: System::Void
9. ochishToolStripMenuItem_Click(System::Object^
10. sender, System::EventArgs^ e) { // PUNKT MENYUSI FOYDALANUVCHI
11. FAYLNI TANLAVDI:
12. openFileDialog1->ShowDialog();
13. // FAYLNING NOMINI PLEERGA JO'NATISH
14. axWindowsMediaPlayer1->URL = openFileDialog1->FileName;
15. // axWindowsMediaPlayer1->URL = "C:\\WINDOWS\\Media\\tada.wav";
16. // FAYLNI YUKLASH KOMANDASI
17. axWindowsMediaPlayer1->Controls->play();
18. // axWindowsMediaPlayer1->openPlayer(openFileDialog1->FileName);
19. private: System::Void
20. yopishToolStripMenuItem_Click(System::Object^ sender,
21. System::EventArgs^ e) { // KIRISH MENYU bo'limi
22. Application::Exit(); }
23. private: System::Void
24. ekranToolStripMenuItem_Click(System::Object^ sender,
25. System::EventArgs^ e) { // Ekran bo'ylab MENYU bo'limi.
26. if (axWindowsMediaPlayer1->playState ==
27. WMPPlayState::WMPPlaying)
28. axWindowsMediaPlayer1->fullScreen = true; }
29. private: System::Void
30. tuxatishToolStripMenuItem_Click(System::Object^ sender,
31. System::EventArgs^ e) { // To'xtatish MENYU bo'limi
32. axWindowsMediaPlayer1->controls->pause(); }
33. private: System::Void
34. boshlashToolStripMenuItem_Click(System::Object^ sender,
35. System::EventArgs^ e) { // Boshlash MENYU bo'limi
36. axWindowsMediaPlayer1->controls->play(); }
37. private: System::Void
38. ovozinOchirishToolStripMenuItem_Click(System::Object^ sender,
39. System::EventArgs^ e) { // Ovozni yoqish MENYU bo'limi
40. axWindowsMediaPlayer1->settings->mute = true; }
41. private: System::Void
42. ovozinYopishToolStripMenuItem_Click(System::Object^ sender,

```

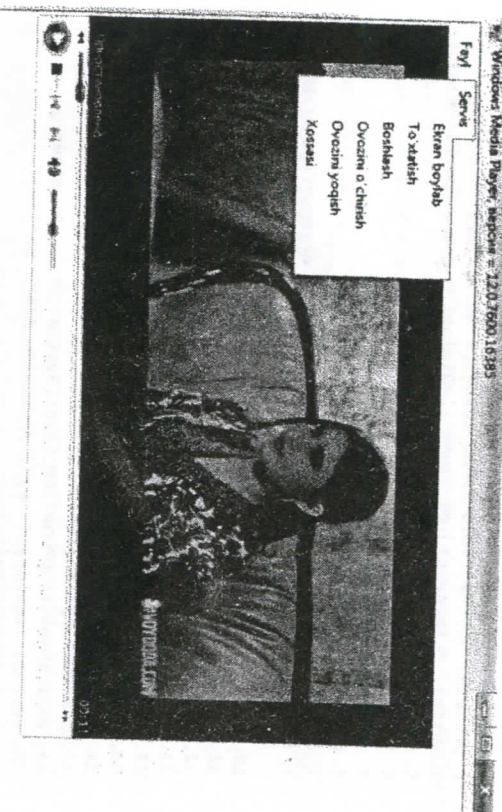


```

40. System::EventArgs^ e) { // Ovozni o'chirish MENU bo'limi
41. axWindowsMediaPlayer1->settings->mute = false; }
42. private: System::Void
43. xOSSasiToolStripMenuItem_Click(System::Object^ sender,
44. System::EventArgs^ e) { // Fayl xOSSasi MENU bo'limi
45. axWindowsMediaPlayer1->ShowPropertyPages(); }

```

Dasturning tashqi ko'rinishi quyidagicha bo'ladi:



3.36- rasm. Windows Media Player ni yuklash

Nazorat savollari:

1. Visual C++ da MS Excel fayllariga eksport va import qanday amalga oshiriladi?
2. Visual C++ da MS Word fayllariga eksport va import qanday amalga oshiriladi?
3. Visual C++ da diagrammalar bilan ishlash komponentalarini bilasizmi?
4. Visual C++ da satrlar bilan ishlovchi qanday funksiyalar bor?
5. Visual C++ da jadvalli tip nima?
6. DataGridView komponentasining vazifasi nima?
7. Visual C++ da ovozi fayllarga qanday murojaat qilinadi?
8. Visual C++ da video fayllarga qanday murojaat qilinadi?

9. Visual C++ da rasmi fayllarga qanday murojaat qilinadi?
10. Visual C++ da dastur hisobotlari qanday tayyorlanadi?

Bob xulosasi

Ushbu bobda **Windows Form Application** ilovasini yaratish, forma (Form) ni formalar bog'lash, berilganlarni lug'at (Dictionary) yordamida strukturali saqlash, bir prosedura orqali bir nechta hodisalarga ishlov berish, klaviatura hodisalarini qayta ishlash, matli maydonga kiruvchi ma'lumotlarni boshqarish, try ... catch istinosini qayta ishlash, matli xujjatni chop qilish, formada grafik shakllarni va funktsiya grafiklarni hamda sichqoncha ko'rsatgichi orqali chizish, veb browserda HTML jadvallarni tasvirlash, Visual C++da MS Word imkoniyatlaridan foydalanib, jadvallar yaratish, ularni Word faylga eksport qilish, Visual C++ da MS Excel imkoniyatlaridan foydalanib, diagrammalar yaratish va Visual C++ ning Windows Application muhitida komponentalarning joylashish vaziyatlarini nazorat qilish usullari ko'rib chiqildi. Ushbu usullar asosida amaliy dasturlar yaratish yo'llari ko'rsatildi. Har bir tuzilgan amaliy dastur tahlillari keltirildi. Ushbu vazifalarni bajarish uchun quyidagi komponentalardan foydalanildi: Form, Button, Label, TextBox va DateTimePicker, CheckBox, CheckedListBox, ComboBox, ListBox, TabControl, RadioButton, LinkLabel, OpenFileDialog, SaveFileDialog, Chart komponentalari. **MessageBox** va shu kabi xabarlar oynalaridan foydalanish qoidalarini keltirildi. **MouseHover** hodisasi kabi hodisalar o'rganildi, amaliy dasturlarda qo'llanildi va tahlil qilindi.

Nazariy savollar va amaliy topshiriqlar

1. Ob'ektni avtomatlashtirilishi lozim bo'lgan jarayonlarni topish. **Kutubxona**, dorixona, supermarket, firma va yuridik tashkilotlar misolida jarayonning strukturali va matematik modelini ishlab chiqish.
2. Kutubxona kitoblarining joriy reyting olib boruvchi avtomatlashtirilgan tizimini matematik loyihalash. Kutilayotgan natijalarni rejalashtirish.

Boshlab maqsadi



Ushbu bobda **MBBT**, **MS Access** dasturi, **SQL(Structured Query Language)** so'rovlar tili haqida ma'lumotlar va komponentalar va ularni ishlatishi yuzasidan quyida keltirilgan mavzular bo'yicha amaliy mashg'ulotlar keltirilgan:

1. «**MS Access**» dasturi yordamida talabalarning bilimlarini monitoring qilish ma'lumotlar bazasini yaratish texnologiyasi (konstruktor rejimida jadval yaratish, konstruktor rejimida so'rovlar yaratish, konstruktor rejimida makroslar yaratish, konstruktor rejimida formalar yaratish, formaga komponentalar joylashtirish)
2. **SQL** so'rovlar tili (**select** komandasi, **Like** va mantiqiy operatorlar (**and**, **or**, **not**), tartiblash (**order by**) va qisqartirish (**distinct**), **where** komandasi, jamlash operatorlari: **count**, **min**, **max**, **avg**, **sum** va **group by**, **SQL**da qism so'rovlar: **in**, **exists**, **not exists**, **SQL** funksiyalari: **lower** va **upper** haqida, jadvallarga yangi yozuv qo'shish: **INSERT** operatori, jadvallardan yozuvlarni o'chirish: **DELETE** operatori, **SQL** so'rovlarini bajarilishini bo'yicha maxsus jadval);
3. **Visual C++** da **MS Access** ning ma'lumotlar bazasini **SQL** so'rovlarini asosida tahrirlash usullari;
4. **Command** va **DataReader** sinf ob'ektlari yordamida **MS Access**da yaratilgan ma'lumotlar bazasining jadvalidagi barcha ma'lumotlarni o'quvchi dastur;
5. **Console Application** muhitida **MS Access** ma'lumotlar bazasini yaratuvchi dastur;
6. **Console Application** muhitida **MS Access** ma'lumotlar bazasining jadvallariga ma'lumotlar yozuvchi dastur;
7. **Command**, **DataReader** sinf ob'ektlari va **DataGridView** komponentalari yordamida **MBning** jadvalidan ma'lumotlarni o'quvchi vizual dastur;

8. **Command**, **Adapter** va **DataSet** sinf ob'ektlari hamda **DataGridView** komponentasi yordamida **MBning** jadvalidan ma'lumotlarni o'quvchi vizual dastur;
9. **MS Access** ning **MBdagi** jadval yozuvlarini yangilovchi vizual dastur;
10. **MS Access** ning **MBdagi** jadval yozuvlarini **SQL** so'rovlarini va "**Command**" sinf ob'ekti yordamida o'chirtuvchi vizual dastur.

Bob muharriri

- 4.1. Ma'lumotlar bazasini boshqarish tizimlari. **Microsoft Access** da ma'lumotlar bazasini yaratish.
- 4.2. **SQL (Structured Query Language)** so'rovlar tili.
- 4.3. **Visual C++** da **MS Access** ning ma'lumotlar bazasini **SQL** so'rovlarini asosida tahrirlash usullari.
- 4.4. Kichik loyihalar yaratish usullari va uning yuklanuvchi "инсталляционные" dasturlar paketini yaratish.

4.1. Ma'lumotlar bazasini boshqarish tizimlari. **Microsoft Access**da ma'lumotlar bazasini yaratish

Mavzuning maqsadi

MS Access 2010 da jadvallar, so'rovlar, formalar va makroslar yaratish usullarini o'rganish. **MS Access 2010** dasturida "**Talabalarning bilimlarini monitoring qiluvchi ma'lumotlar bazasini**" yaratish.

Informasion tizimlarni yaratish bo'yicha jadal harakatlar va ma'lumotlar hajmining tez sur'atlar bilan oshib borishi 60 yillar boshida maxsus "Ma'lumotlar bazasini boshqarish tizimi" (**MBBT**) deb ataluvchi dasturiy kompleksning yaratilishiga olib keldi.

Ma'lumotlar bazasi - biror sohaga oid o'zaro bog'langan ma'lumotlar yig'indisining disk tashuvchidagi tashkiliy jamlanmasidir. Boshqacha qilib aytganda, ma'lumotlar bazasi - bu kompyuter xoirasiga yozilgan ma'lum bir strukturali, o'zaro bog'langan va tartiblangan ma'lumotlar majmui bo'lib, u biror bir ob'ektning xususiyatini, holatini yoki ob'ektlar o'rtasidagi munosabatni ma'lum ma'noda tavsiflaydi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) - bu dasturiy va apparat vositalarining murakkab majmui bo'lib, ular yordamida foydalanuvchi ma'lumotlar bazasini yaratish va shu bazadagi ma'lumotlar ustida ish yuritish mumkin. MBBT o'z maxsus dasturlash tillariga ham ega bo'lib, bu tillarga buyruqli dasturlash tillari deyiladi.

MBBTga **Oracle, Clipper, Paradox, FoxPro, SQL, Access** va boshqalarni misol keltirish mumkin. MBBT asosiy xususiyatlari - bu nafaqat ma'lumotlarni kiritish va saqlashda ishlatiladigan proceduralar tarkibi bo'lmagan, ularning strukturatsini ham tasvirlaydi.

Ma'lumki, MB tushunchasi fanga kiritib kelgunga qadar, ma'lumotlardan turli ko'rinishda foydalanish juda qiyin edi. Dastur tuzuvchilar ma'lumotlarini shunday tashkil qilishni edi. Dastur qaratayotgan masala uchungina o'rini bo'lardi. Har bir yangi masalani hal qilishda ma'lumotlar qaytadan tashkil qilinardi va bu hol yaratilgan dasturlardan foydalanishni qiyinlashtirardi.

Shuni qayd qilish lozimki, MBni yaratishda ikkita muhim shartni hisobga olish zarur:

Birinchidan, ma'lumotlar turi, ko'rinishi, ularni qo'llaydigan dasturlarga bog'liq bo'lmasiligi lozim, ya'ni MBga yangi ma'lumotlarni kiritganda yoki ma'lumotlar turini o'zgartirganda, dasturlarni o'zgartirish talab etilmasiligi lozim.

Ikkinchidan, MBdagi kerakli ma'lumotni bilish yoki izlash uchun biror dastur tuzishga hojat qolmasin.

Shuning uchun ham MBni tashkil etishda ma'lum qonun va qoidalariga amal qilish lozim. Bundan buyon **axborot** so'zini **ma'lumot** so'zidan farqlaymiz, ya'ni **axborot** so'zini umumiy tushuncha sifatida qabul qilib, **ma'lumot** deganda aniq bir belgilangan narsa yoki hodisa sifatlarini nazarda tutamiz.

Kompyuter xotirasida har bir fayl yozuv deb ataladigan bir xil turdagi qismlardan iborat bo'ladi. **Yozuv** - o'zaro bog'langan ma'lumotlarning bir qismidir. Fayldagi yozuvlar soni, qaratilgan ma'lumotning o'lchoviga bog'liq. Har bir yozuv esa **maydon** deb ataladigan bo'laklardan tashkil topadi. **Maydon** ma'lumotlarning imkon boricha, qisqa to'planidan iborat bo'lishi lozim. Har bir **maydon**, o'zi ifodalaydigan ma'lumotlariga ko'ra, biror nomga ega bo'ladi. Fikrimizni misol bilan ifodalashga harakat qilamiz.

Masalan, biror oliy o'quv yurtining aniq fakultetida tahsil olayotgan biror guruh talabalari to'g'risidagi ma'lumotlar kiritilgan quyidagi jadvalni ko'raylik:

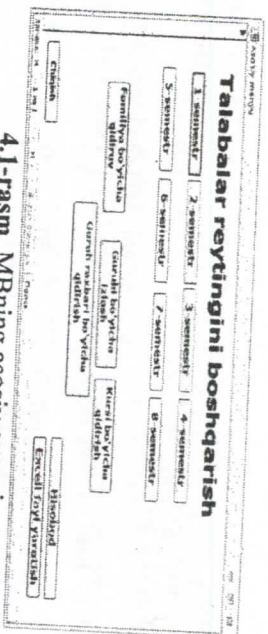
4.1.- jadval. MB dagi jadval ko'rinishi.

Familiyasi	Ismi	Tug'ilgan sanasi	Guruhi	Turar joyi	qiziqqan fani
Ochilov	Alisher	2.05.1978	5-M	S-1,15	Matem.
Qobulov	Farhod	2.12.1982	6-E	1.Sino,1	Adabiyot
Aminov	Sar'at	3.06.1980	5-M	S-2,12	Tarix
Tolipov	Jasur	24.05.1979	6-E	Beruni,2	Iqtisod

Bu misolda 4 ta yozuv bo'lib, ularning har biri 6 ta maydondan iborat. Mazkur maydonlarning har biri mos ravishda «**Familiyasi**», **Ismi**», «**Tug'ilgan sanasi**», «**Guruhi**», «**Turar joyi**» va «**qiziqqan fani**» deb nomlangan. Demak, yozuvdagi maydonlar soni yozuvga kiritiladigan ma'lumotlar hajmiga bog'liq. Fayldagi bu yozuvlar **birjamchi** hisoblanadi. Chunki biror yozuvdagi ixtiyoriy ma'lumotni boshqa yozuvdagi ma'lumotlar bilan taqqoslab aniqlash mumkin emas. Shuning uchun ham bizga kerakli bo'ladigan ikkilamchi yozuvlarni esa faqat amaliy dasturlar yordamida olish mumkin bo'ladi. Modomiki shunday ekan, MB tashkil qilish, ularga qo'shimcha ma'lumotlarni kiritish va mavjud MBdan foydalanish uchun maxsus MBlar bilan ishlaydigan dasturlar zarur bo'ladi. Bunday dasturlar majmui **ma'lumotlar bazasini boshqarish tizimi (MBBT)** deb yuritiladi. Aniqrog' qilib aytganda, MBBT - bu ko'plab foydalanuvchilar tomonidan MBni yaratish, unga qo'shimcha ma'lumotlarni kiritish va MBni birgalikda ishlatish uchun zarur bo'lgan dasturlar majmuidir. MBBTning tarkibida asosiy komponenti - bu **ma'lumotlar** bo'lsa, boshqa komponenti-**foydalanuvchilar**, **Hardware** - texnik va **Software** - dasturiy ta'minoti hisoblanadi. **Hardware** tashqi qo'shimcha xotiradan (disk, magnit lentasi) iborat bo'lsa, dastur qismini esa MB bilan foydalanuvchi o'rtasidagi mulogotni tashkil qilishni amalga oshiradi. MBning tuzilishi o'rganilayotgan ob'ektning ma'lumotlari ko'rinishi, ma'nosi, tuzilishi va hajmiga bog'liq bo'ladi.

4.4.1. «MS Access» dasturi yordamida talabalarining bilimlarini monitoring qilish ma'lumotlar bazasini yaratish texnologiyasi

MBning umumiy ko'rinishi quyidagicha:



4.1-rasm. MBning asosiy oynasi

Ushbu MBni yaratish algoritmi quyidagicha:

- 1- Har bir semester uchun alohida jadvallar yaratiladi.
- 2- Hamma jadvallarni birlashtiruvchi umumiy baza yaratiladi.
- 3- Jadvallar birga ko'p usulda bog'lanadi.
- 4- Umumiy jadvaldan ma'lumotlarni qidirish uchun so'rovlar (Familyasi, Guruhi, Kursi va Guruh rahbari bo'yicha) yaratiladi.
- 5- Hisobotlar tayyorlash uchun bitta makros yaratiladi va u (Hisobot) tugmasiga birlashtiriladi.
- 6- Har bir semester uchun alohida formalar yaratiladi.
- 7- Har bir semester formasiga, jadvaldagi ma'lumotlarni ko'rish uchun 3 ta (Familyasi bo'yicha, Guruhi bo'yicha va Umumiy bo'yicha), guruh talabalarini fanlardan olgan baholarini ko'rish uchun bitta, (Maxsus baholash) jadvalning Excel faylini yaratish uchun esa bitta (Excel faylini yaratish) tugmalar joylashtiriladi.
- 8- Asosiy oynaga 4.1-rasmida keltirilgandek tugmalar joylashtiriladi.
- 9- Semesterlar bo'yicha yaratilgan formalar ushbu tugmalarga birlashtiriladi.

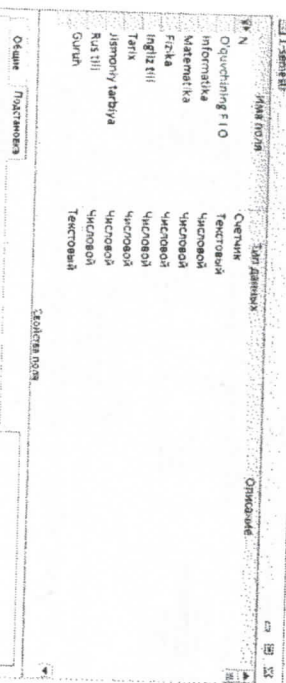
MBni yaratish uchun "Microsoft Access 2010" dasturida foydalaniladi:



4.2-rasm. "MS Access 2010" dastur muxariri

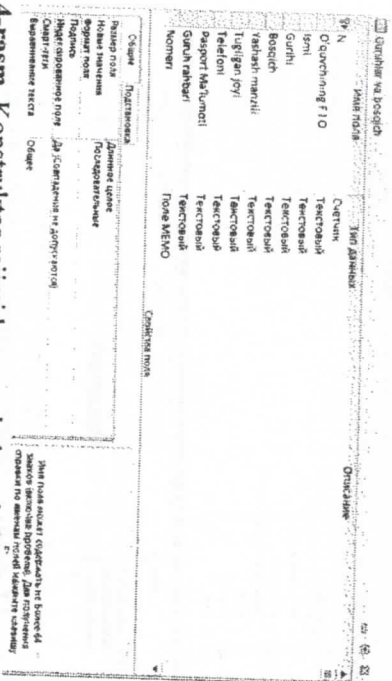
Bu oyundan: "Доступные шаблоны" bandidan "Новая база данных" buyrug'i tanlanadi. "Имя файла" bandini tanlab yaratilishi kerak bo'lgan ma'lumotlar bazasini nomi kiritilib, "Создать" tugmasi bosiladi. Agar uning formati ".mdb" bo'lishini xohlasangiz "Имя файла" bandidagi "папка" piktogrammasi bosiladi va ochilgan oynadan fayl nomi va kengaytmasini ixtiyoriy tanlash imkoniyati hosil bo'ladi.

1. "Конструктор режимa" jadval yaratish 1-semester jadvali:



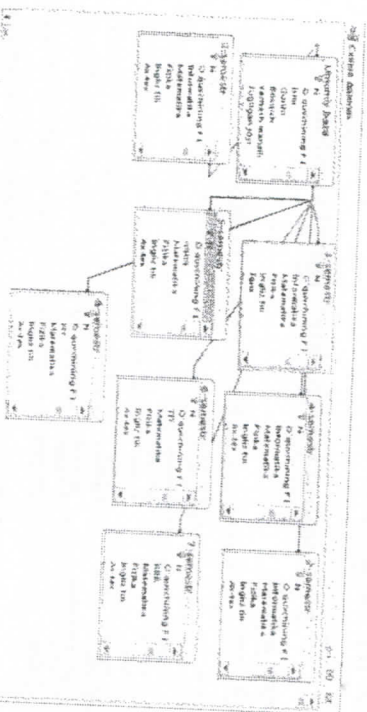
4.3-rasm. Конструктор rejimida 1-semester jadvalini yaratish

Shu kabi boshqa jadvallarni (2-semester, ... 8-semester) yaratib olinadi, chunki har bir semesterda fanlar o'zgarib boradi. Umumiy baza quyidagicha tuziladi:



4.4-rasm. Конструктор rejimida umumiy baza jadvalini yaratish

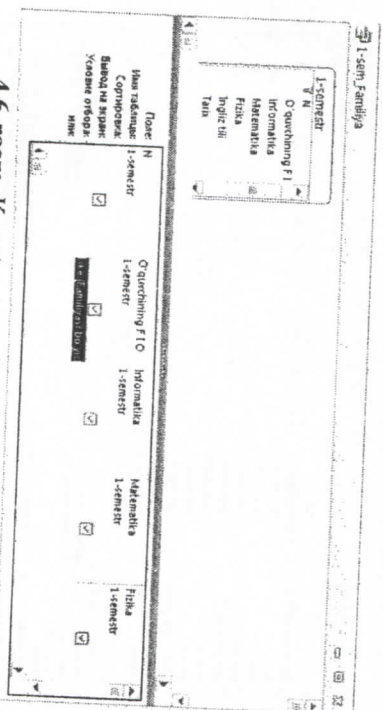
Bu jadval 1 - bosqichdan to 4 - bosqichgacha bo'lgan talabalarning ma'lumotlarini o'zida jamlaydi va boshqa yordamchi jadvalga avtomatik tarzda tarqatadi. Buni amalga oshirish uchun jadvallarni bog'lash lozim bo'ladi. U quyidagicha amalga oshiriladi:



4.5-rasm. Umumiy baza jadvaliga semester jadvallarini bog'lash

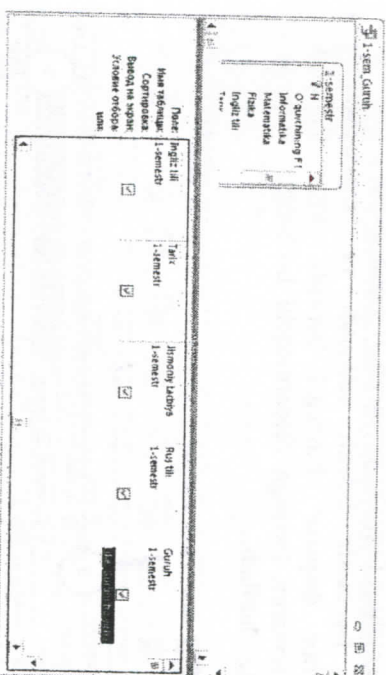
II. Konstruktor rejimida so'rovlar yaratish

Har bir semestrda alohida jadval bo'lgani uchun ularga alohida so'rovlar yaratiladi. Ya'ni, familiyasi bo'yicha qidirish so'rovini yaratish uchun 4.6-rasmdagi amallar ketma-ketligi bajariladi:



4.6-rasm. Konstruktor rejimida so'rovlar yaratish

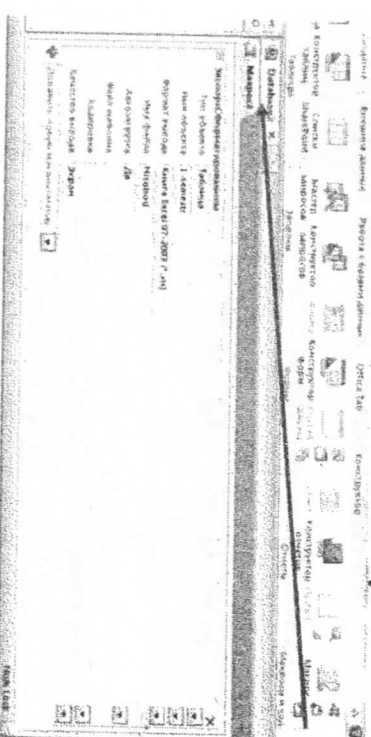
Guruh bo'yicha qidirish amalini bajarish uchun 4.7-rasmdagi amallar ketma-ketligi bajariladi:



4.7-rasm. Konstruktor rejimida so'rovlar yaratish

Shu kabi qolgan semester so'rovlari va umumiy jadvalning so'rovlari yaratiladi.

III. Konstruktor rejimida makroslar yaratish



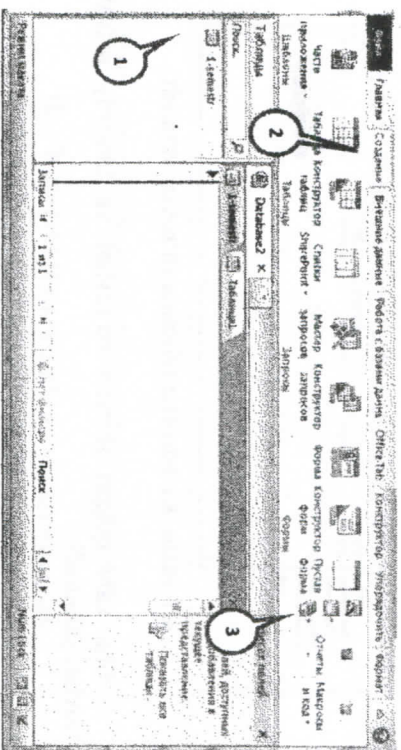
4.8-rasm. 1- semester uchun makros yaratish

Ushbu makros 1-semester jadvalini MS Excell dasturiga eksport qiladi. Xuddi shu tartibda qolgan semester jadvallari uchun ham makroslar yaratiladi.

IV. Konstruktor rejimida formalar yaratish

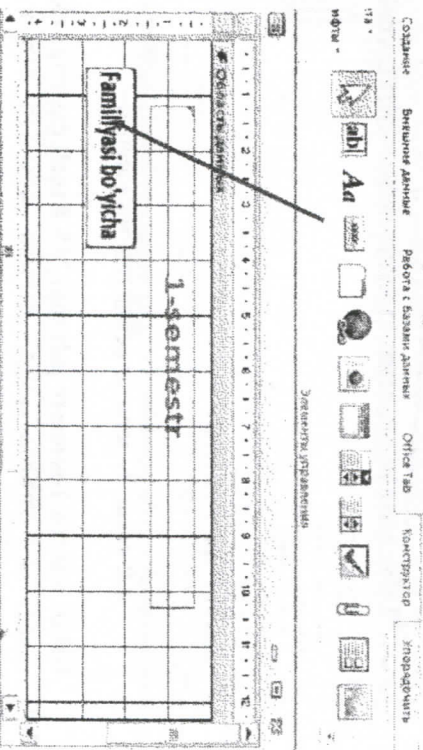
Har bir semestrda alohida jadval va so'rovlar bo'lgani uchun ularga alohida formalar yaratiladi. U quyidagicha:

1-semestr jadvali tanlanadi va "Создание" menyusidan "Пустая форма" buyrug'i beriladi. Agar tanlangan jadvalni ma'lumotlarini formaga chiqatmoqchi bo'lsangiz, u holda "Формы" buyrug'i beriladi.

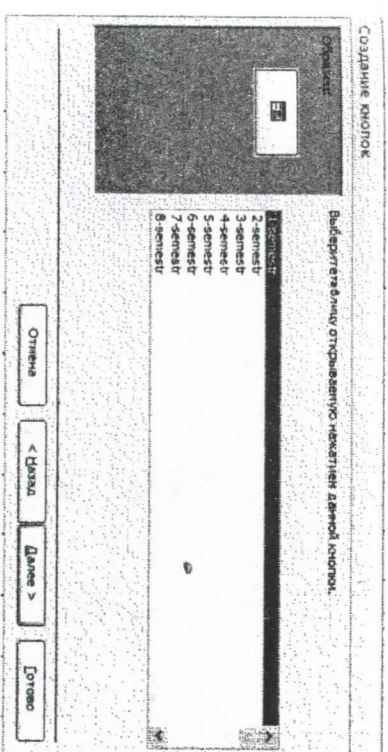


4.9-rasm. 1 - semestr uchun forma yaratish

1-semestr formasiga tugmalar o'rnatiladi.

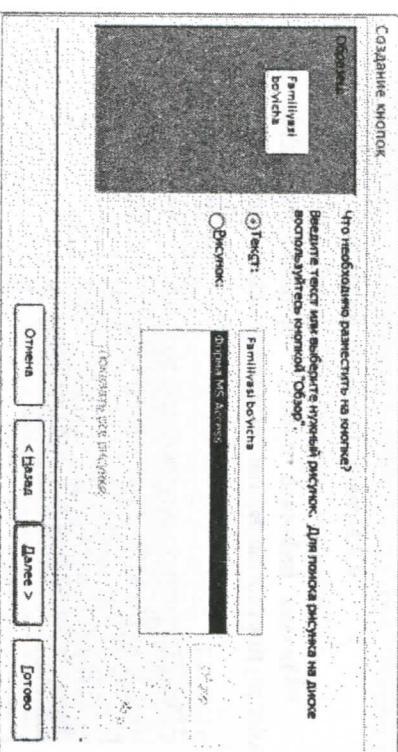


4.10-rasm. 1 - semestr formasiga tugmalar o'rnatish



4.11-rasm. Formaga formatlarni birlashtirish

Kerakli jadval tanlangandan so'ng "Далее" tugmasi bosiladi:



4.12-rasm. Tugma xossalarni belgilash

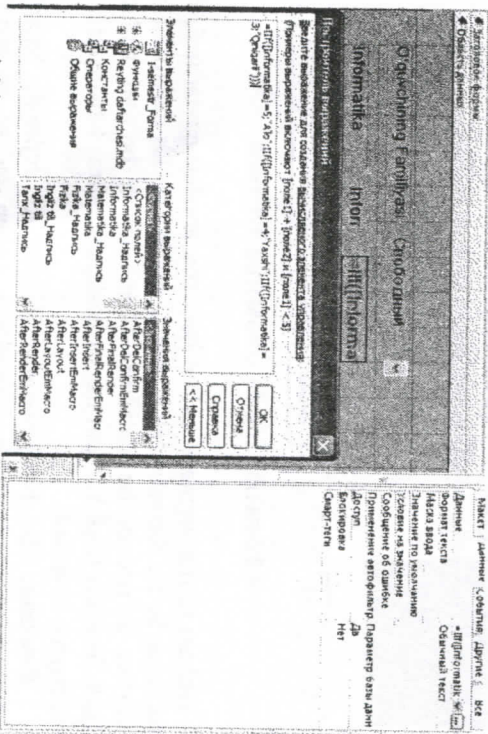
Qolgan to'rtta tugmalar uchun ham shu amallar qaytariladi, faqat forma kerak bo'lsa formalar bilan ishlash, makros bo'lsa makroslar bilan ishlash va so'rov bo'lsa so'rovlar bilan ishlash bo'limi tanlanadi. Natijada formada hosil bo'lgan tugma bosilganda nima tanlangan bo'lsa ushbu buyrug' bajariladi.

Eslatma: Har bir jarayonlar konstruktor rejimida bajariladi. Masalan, tayyor jadvallarni o'zgartirish uchun jadval tanlanadi va

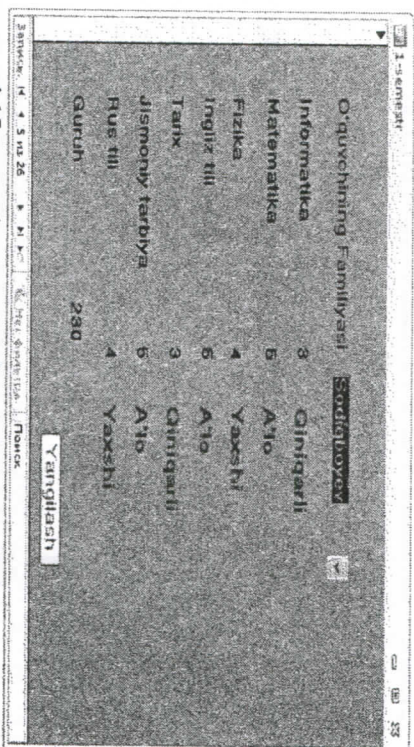
Formaga tashqi dizayn berilib, klaviaturadagi **F5** tugmasi bosilgandan so'ng forma ishchi rejimga o'tadi va quyidagi ko'rinishga keladi:

4.13-rasm. I-semestr forma oynasi

“Maxsus baholash oynasi” tugmasini yaratishda talabalarni baholar reytingi formasini har bir semestr uchun yaratiladi:



4.1.4-rasm. 1-semestr formasida formulalar bilan ishlash

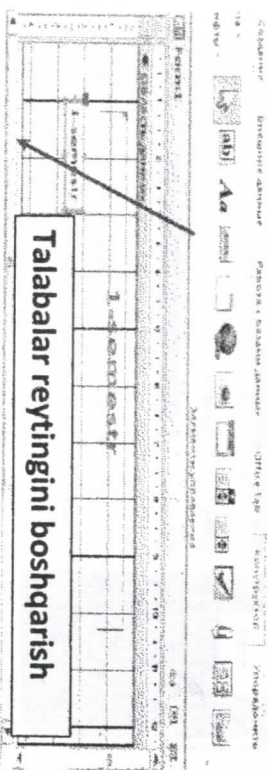


4.15-rasm. 1 - semestr uchun baholar reytingi formasi

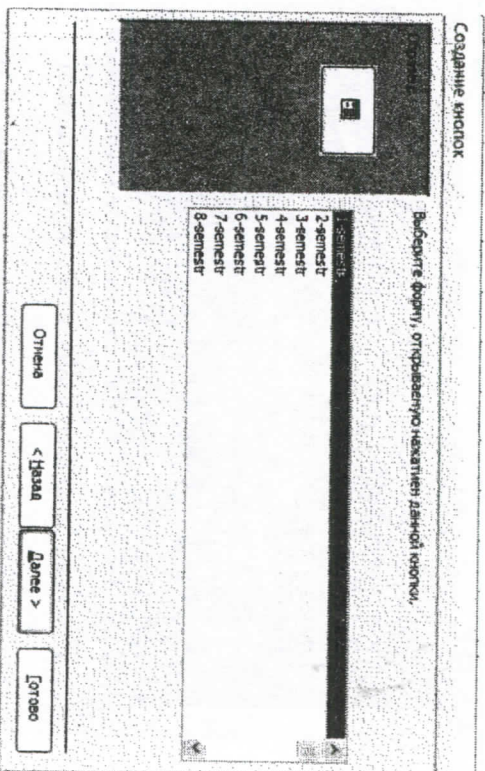
Ushbu amallar boshqa semestr jadvallari uchun ham bir xildir.

V. Formaga komponentalar joylashtirish

Asosiy oyna formasida tugmalarga semestr formalarini biriktirish. Forma yaratilganda **"konstruktor"** menyusi paydo bo'ladi. Konstruktor dan knopka tanlanadi.

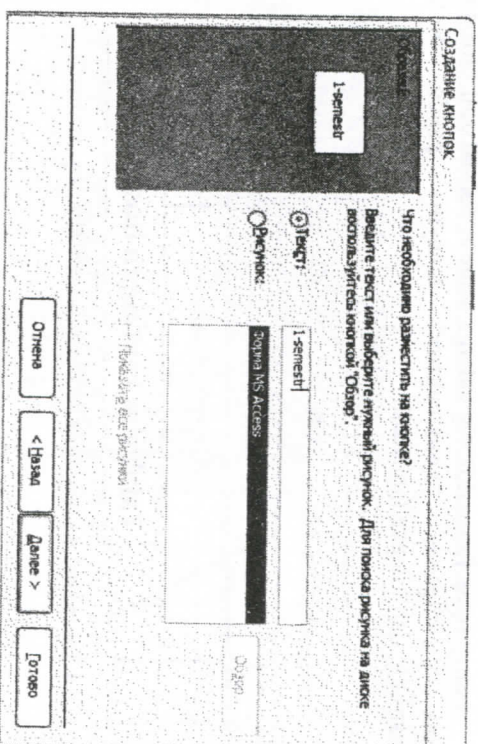


4.1.6-rasm. Asosiy formaga komponentalar joylashtirish.



4.17-rasm. Formaga formatlarni biriktirish

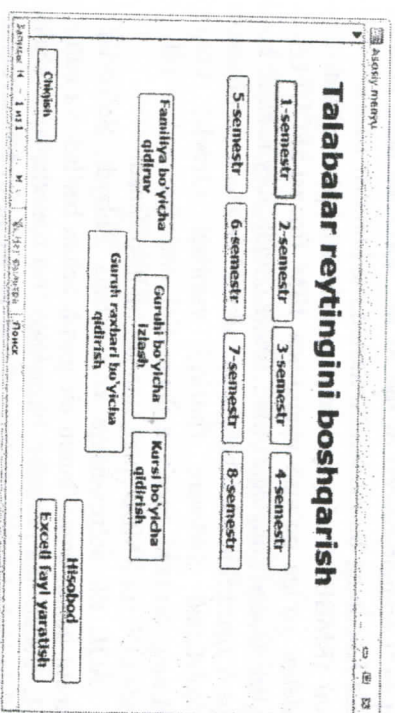
Kerakli forma tanlangandan so'ng "Далее" tugmasi bosiladi:



4.18-rasm. Tugma xossalarni belgilash oynasi

Yaratilgan MBda umumiy jadval yaratilgan. U orqali barcha jadvallar tashkil topadi. Bu jadvalda talabalarning umumiy ma'lumotlari

joylashgan. 1- bosqichdan to 4-bosqichgacha bo'lgan talabalarning familiyasi, guruh, bosqichi va guruh rahbari bo'yicha qidirish imkoniyatlari to'rtta (familiyasi, guruh, bosqichi va guruh rahbari bo'yicha qidirish) tugmalarga joylashtirilgan. "Hisobot" va "Excel fayl yaratish" tugmalarining vazifasi umumiy jadvalda joylashgan ma'lumotlar asosida hisobotlar va Excel fayl yaratishdan iboratdir. Dasturdan chiqib ketish uchun "Chiqish" tugmasi joylashtirilgan. MBda 12 ta jadval, 30 ta so'rov, 12 ta makros va 11 ta formadan tashkil topgan. Uning interfeysi quyidagicha:



4.19-rasm. MBning asosiy oynasi

Mishtakamlash uchun mashqlar

1. Mahalla ma'lumotlar bazasi;
2. Maktab o'quvchilarini ma'lumotlar bazasi;
3. Shifoxonaning qabul bo'limini ma'lumotlar bazasi;
4. Fakultetning ma'lumotlar bazasi.

Nazorat savollari:

1. MS Access 2010 dasturining maqsad va vazifalari?
2. MS Access 2010da jadval yaratishning nechta usuli bor?
3. MS Access 2010 da so'rov yaratishning nechta usuli bor?
4. MS Access 2010 da forma yaratishning nechta usuli bor?
5. MS Access 2010 da makros yaratishning nechta usuli bor?
6. Formaga tugmachalar qaysi rejimda joylashtiriladi?

4.2. SQL(Structured Query Language) tili so'rovlari. SQL haqida

SQL – bu so'rov tili ko'p operatorlardan tashkil topgan bo'lib, bu operatorlar orqali foydalanuvchilar va dasturlar Oracle (MBBT) dagi ma'lumotlar bazasiga murojaatni amalga oshirishi mumkin. Oracle utilitalari yoki har xil dasturlar SQL operatorlarisiz bazaga murojaatni amalga oshirishi mumkin, lekin so'rovlarni amalga oshirishda bu so'rov tilidan foydalanmaslikning iloji yo'q.

1970 yil iyun oyida E.F.Kodd o'zining "A Relational Model of Data for Large Shared Data Banks" maqolasini ommaga taqdim etdi. Bu maqola "Communications of the ACM" jurnalida chop etildi. Hozirgi kunda Kodning bu modeli "Relasion ma'lumotlar bazasini boshqarish tizimi (RMBBT)" ning yakuniy modeli deb qabul qilindi. Kodning modelini yo'lga qo'yish maqsadida IBM firmasi SEQUEL(Structured English Query Language) tilini ishlab chiqdi. Keyinchalik bu til SQL tiliga o'zgartirildi, lekin haligacha "sikvel" deb ham yuritilmogda. 1979 yil Relational Software (hozirgi vaqtdagi Oracle) korporatsiyasi SQLning birinchi tijoriy ishlatmasini ommaga taqdim etdi. Hozirgi kunda SQL tili RMBBTning standart tili hisoblanadi.

SQL tili so'rov-natija ko'rinishida ishlaydi. So'rovlar har bir element uchun emas, butun bir guruh uchun beriladi va natija olinadi. **SQL** uchun ma'lumotlar bazasidagi ma'lumotlar qay shaklda, qay tartibda joylashganini umuman ahamiyati yo'q, foydalanuvchilar ham bu ma'lumotlarni bilishi shart emas. Faqatgina operatorlarni to'g'ri yozish orqali istalgan ma'lumotlarni chiqarish mumkin bo'ladi.

SQL tili barcha ma'lumotlar bazasini boshqarish tizimlari uchun umumiy standart til hisoblanadi. Bundan kelib chiqadiki, agar siz bu tilni bir marotaba o'rganib olsangiz, istalgan MBBTlari bilan ishlay olasiz. Bitta **MBBT**da yaratilgan biror **SQL** operatorlar yig'indisi (kichik so'rov dasturi)ni istalgan **MBBT**ga ko'chirish mumkin bo'ladi.

SQL operatorlari orqali quyidagi vazifalarni bajarish mumkin:

- Ma'lumotlarni so'rov orqali olish.
- Jadvalning qatorlariga ma'lumot qo'shish, qatorlarini o'chirish va yangilash.
- Ob'ektlarni yaratish, o'zgartirish va o'chirish.
- Ma'lumotlar bazasi va ob'ektlarga ruxsatlarni o'rnatish.
- Ma'lumotlar bazasi foydalanuvchilarini hosil qilish va baza xavfsizligini ta'minlash.

162

Ikki xil turdagi **SQL** mavjud: interaktiv va o'rnatilgan (встроенный). **SQL** ning bu ikki turini ishlashi bir xil, lekin har xil joyda ishlatiladi.

Interaktiv **SQL** deganda — ma'lumotlar bazasiga so'rov orqali murojaat qilib, shu zahoti natijani olish tushuniladi. Ya'ni, bunda ketma-ketlik asosida jarayon sodir bo'ladi. So'rov-natija rejimida ishlaydi.

O'rnatilgan **SQL** deganda — so'rovlar yig'indisi biror dasturlash tilida ishlatilishi tushuniladi. **Pascal, Delphi, Java** va **C++** tillarida bazaga murojaat qilib, natijani biror o'zgartuvchiga yuklab qo'yamiz va kerakli joyda bu natijani ishlatamiz. Ya'ni bunda so'rov berib, darhol natijani ololmaymiz. Natija faqat dasturning davom etishi uchun olinadi va talab etilgan joyda ishlatiladi.

select komandasi

Misol sifatida quyidagi jadvaldan foydalanamiz.

NAME	SURNAME	YEAR	NUM
David	Beckham	7	1
Eric	Contana	17	2
Wayne	Rooney	10	3
Rio	Ferdinand	4	4
Nemane	Vidic	6	5

Bu jadval "**misol**" deb nomlanib, 4 ta ustun va 6 ta qatordan iborat. Yuqoridagi qatorda faqat qator nomlari aks ettirilgan, ular ma'lumot vazifasini bajarmaydi. 5 ta qatordan iborat ma'lumot, bizning jadvalimizda joylashgan va biz bu ma'lumotlarni **SQL** so'rov tili orqali har xil ko'rinishda chiqarib olishimiz mumkin. Albatta haqiqiy ma'lumotlar bazasida bunday kam ma'lumotlar saqlanmaydi, biz misollarni shu kichik ma'lumotlarda bajaramiz.

Demak birinchi komanda bu — **SELECT**.

```
select * from misol;
```

Bu so'rovning ma'nosi, "**misol**" jadvalidagi barcha ma'lumotlarni chiqar deganidir. "*" barcha ustunlardagi ma'lumotlarni degani. "**select**" — esa ekranga chiqar degani. Natijani ko'ramiz:

163

NAME	SURNAME	YEAR	NUM
David	Beckham	7	1
Eric	Contana	17	2
Wayne	Rooney	10	3
Rio	Ferdinand	4	4
Nemane	Vidic	6	5

Agar bizga barcha ustun ma'lumotlar emas, faqatgina ba'zi birlari kerak bo'lsa, kerakli ustun nomlarini vergul orqali nomma-nom yozamiz.

select name, surname, year from misol;

NAME	SURNAME	YEAR
David	Beckham	7
Eric	Contana	17
Wayne	Rooney	10
Rio	Ferdinand	4
Nemane	Vidic	6

Ma'lumotlar bazasidan, ma'lumotlarni chiqarishda mantiqiy amallar orqali(+, -, *, /, () ma'lumotlarga o'zgartirish kiritib, ekranga chiqarishimiz mumkin bo'ladi.

*select name, year+15, num*10 from misol;*

Natijani ko'rsangiz, so'rov tili orqali, ma'lumotlarni o'zgartirganimizni ko'rishingiz mumkin bo'ladi.

NAME	YEAR+15	NUM*10
David	22	10
Eric	32	20
Wayne	25	30
Rio	19	40
Nemane	21	50

yoki bo'lmasa quyidagicha

*select year+10*num, name from misol;*

YEAR+10*NUM	NAME
17	David
37	Eric
40	Wayne
44	Rio
56	Nemane

Agar bazadagi ustun nomlari sizga yoqmasa, siz ularni o'zingiz xohlaganday nom bilan ekranga chiqarishingiz mumkin bo'ladi, bunda ma'lumotlar bazasidagi ustun nomlari o'zgaraydi, faqat ekranda o'zgaragan ko'rinishi chiqadi. Bunda bizga "as" kalit so'zi yordam beradi.

select name as ism, surname as familiya from misol;

ISM	FAMILIYA
David	Beckham
Eric	Contana
Wayne	Rooney
Rio	Ferdinand
Nemane	Vidic

Agar nomlarni qo'shtirmoq ichiga olsangiz, qo'shtirmoqdagi katta kichik harflar farqli bo'lib qoladi.

select name as "ISM", surname as "FAMILYA" from misol;

ISM	FAMILYA
David	Beckham
Eric	Contana
Wayne	Rooney
Rio	Ferdinand
Nemane	Vidic

Like va mantiqiy operatorlar (and, or, not)

Navbatdagi darsimizning asosiy qismi "Like" operatoriga bag'ishlanadi. Bu orqali so'rovlarni ma'lum bir shablon orqali qurishingiz mumkin bo'ladi. Hozirgacha biz o'rgangan so'rovlar aniqlik asosida tuzildi va bita harfga adashib so'rov tuzib qo'ysangiz, kerakli ma'lumotlarni olishga qiyinalishingiz aniq. Bu darsda quyidagi "test" nomli jadvaldan foydalanamiz.

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000
Pablo	Aimar	30	25000
Carlos	Puyol	23	27000
Sir	Alex	50	
Phil	John	16	12000

Agar siz biror "x" ustundagi ma'lumot bo'yicha qolgan ustundagi ma'lumotlarni chiqarmoqchi bo'lsangiz va "x" ustundagi ma'lumotni nomini to'liq bilmasangiz, "like" operatoridan foydalanish sizga yengillik yaratadi. Demak nazariyadan amaliyotga o'tamiz, misol "test" jadvalidagi "ISM" ustundagi ma'lumotlar ichidan "S" harfdan boshlanadigan ismlarning to'liq ismini va familiyasini chiqaruvchi so'rov tuzish kerak:

```
select ism, yosh, familiya from test where ism like 'S%';
```

ISM	YOSH	FAMILIYA
Sir	50	Alex

Bu yerda "like" kalit so'zidan keyin, biror shablon beriladi, "e%" belgisi nol yoki bir necha simvol degan ma'noni beradi, bundan kelib chiqadiki, so'rovda "S" harfdan boshlanuvchi (bu harfdan keyin istalgancha simvol bo'lishi mumkin yoki umuman bo'lmasligi ham mumkin) ismlarni chiqarib beradi. Y qorida ketirilgan belgilarni("e%", "_") aralash holda ham ishlatish mumkin. Misol,

"familiya" ustundagi ma'lumotlar ichidan ikkinchi simvol "e" dan iborat bo'lgan va bu simvoldan keyin istalgancha simvol bo'lgan familiyani to'liq holda chiqaruvchi so'rov quyidagicha bo'ladi:

```
select familiya from test where familiya like 'e%';
```

FAMILIYA
Beckham

Bir necha shartlarni birlashtiruvchi mantiqiy operatorlarni boshlaymiz. Bu operatorlar "where" operatoridan keyin keldigan shartlarni mantiqiy tomondan birlashtirib beradi. Mantiqiy amallarda 2 ta tushuncha mavjud, bular: rost (true) va yolg'on (false). Barcha shartning natijasi yoki rost yoki yolg'on bo'lishi mumkin. Shu natijaga qarab mantiqiy amallar ishlatiladi. Bu mantiqiy amallar 3 hildir.

1. **And** – mantiqiy "va" operatori. Bir necha shartlarning barchasi "true" (rost) bo'lsa, oxirgi natija "true" bo'ladi. Misol, jadvaldagi "yosh" ustuni 20 dan katta bo'lgan va "maosh" ustuni 30000 dan katta bo'lgan insonlarning ismi va familiyasini chiqaruvchi so'rov:

```
select ism, familiya from test where (yosh>20) and (maosh>30000);
```

ISM	FAMILIYA
David	Beckham

Ikki shart bajariladigan (albatta yoshi 20 dan katta bo'lishi va maoshi 30000 dan katta bo'lishi shart) ustunlar ekranga chiqariladi.

2. **Or** – mantiqiy "yoki" operatori. Bir necha shart orasidan istalgan bitasi "true" (rost) bo'lsa, oxirgi natija rost bo'ladi va shart bajarilgan hisoblanadi. Misol, jadvaldan ismi "David" bo'lgan yoki maoshi 20000 dan kam bo'lganlar haqida ma'lumot olish so'rovi:

```
select * from test where (ism='David') or (maosh<20000);
```

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000
Phil	John	16	12000

3. Not – inkor ma'nosini bildiruvchi mantiqiy amal. Barcha buyruqlarni inkorini ishlatadi. Misol, yoshi 16, 21, 30 ga teng bo'lmagan insonlarning ismini chiqarish:

select ism as name from test where yosh not in (16, 21, 30);

NAME
David
Carlos
Sir

Agar "not" kalit so'zini olib tashlasak, yoshi 16, 21, 30 ga teng bo'lgan bo'lib qolar edi.

Tartiblash (order by va qisqartirish(distinct))

Ma'lumotlar har xil va hamma insonlarga kerak, shu bilan birga bu ma'lumotlar har kimga har xil ta'sir qiladi. Shuning uchun korxonalarda so'ragan insonlarga qarab har xil taqdim etiladi. Bu, bir joyga esa albatta ortiqcha ishini yuzaga keltiradi. Buni oldini olish uchun SQL maqolada quyidagi "test" jadvali kerak bo'ladi:

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	
David	Beckham	21	30000
Pablo	Aimar	21	55000
Carlos	Puyol	30	25000
Sir	Alex	23	27000
Phil	John	50	
		16	12000

SQL tilida tartiblash operatori vazifasini "order by" bajaradi. Misol ko'ramiz, **FAMILIYA** ustuni bo'yicha tartiblab, jadvaldagi ma'lumotlarni to'liq ekranga chiqaruvchi so'rov:

168

*select * from test order by familiya;*

ISM	FAMILIYA	YOSH	MAOSH
Pablo	Aimar	30	
Sir	Alex	50	25000
David	Beckham	21	
Phil	John	16	55000
David	Luiz	23	12000
Carlos	Puyol	23	30000
		23	27000

Natijadan ko'rinib turibdiki, "FAMILIYA" ustuni alfavit bo'yicha tartiblanib, ekranga chiqimogda.

"Order by" operatori ikki xil kalit so'z bilan ishlatilishi mumkin. Bular: desc, asc. "Desc" kalit so'zi, ustunni kamayish ketma-ketligida tartiblab ekranga chiqaradi, ya'ni eng katta qiymatdan boshlab eng kichikka qadar. Misol, "ISM" ustunini kamayish tartibida tartiblab chiqarish so'rovi tuzish:

*select * from test order by ism desc;*

ISM	FAMILIYA	YOSH	MAOSH
Sir	Alex	50	
Phil	John	16	12000
Pablo	Aimar	30	25000
David	Luiz	23	30000
David	Beckham	21	55000
Carlos	Puyol	23	27000

Natijadan ko'rinadiki, alfaviting quyi qismidan yuqori qismiga qarab ko'tarilib tartiblanmogda. "Ass" kalit so'zi esa kichikdan-katta (alfavit bo'yicha yuqori harfdan quyi harfga qarab) tomonga qarab tartiblab chiqariladi. Agar hech qaysi kalit so'z berilmasa, avtomat "ass" kalit so'zini ishlatib, tartiblab chiqaradi.

Agar bir necha ustunni tartiblab chiqarish kerak bo'lsa, "order by" operatoridan keyin ustunlar nomi vergul orqali ketma-ket yozib chiqariladi. Bunda natija, eng oldin yozilgan ustun bo'ylab tartiblanib, keyin ikkinchi bo'lib yozilgan ustun bo'yicha tartiblab ekranga

169

chigaradi. Misol, avval "ISM" ustuni, keyin "Familya" ustuni bo'yicha tartiblab, barcha ma'lumotlar ekranga chiqarilsin.

*select * from test order by ism, familiya;*

ISM	FAMILIYA	YOSH	MAO SH
Carlos	Puyol	23	27000
David	Beckham	21	55000
David	Luiz	23	30000
Pablo	Aimar	30	25000
Phil	John	16	12000
Sir	Alex	50	

Bu tartiblash, agar birinchi tartiblanayotgan ustun bir nechta turli hil ma'lumotlardan tashkil topgan bo'lsa foydali hisoblanadi.

Keyingi operatorimiz bir hil ma'lumotlarni qisqartirib, natijani chiqaradigan operator "distinct" haqidadir. Agar chiqarayotgan ma'lumotlar orasida bir xil ma'lumot bo'lsa, bu ma'lumotni bitta qilib chiqaradi. Misol, "ISM" ustundagi bir xil bo'lmagan ismlarni chiqarish so'rovini ko'ramiz:

select distinct ism from test;

ISM
Pablo
Sir
David
Carlos
Phil

Bundan xulosa chiqarish mumkinki, "distinct" operatorini ishlatdik, bir xil qatorlarni umuman ko'rmas ekamiz, lekin ustundagi ma'lumotlar bir xil bo'lishi mumkin. Bu operator faqat qator uchun ishlaydi.

SQL so'rov tilida jadvallarni birlashtirish(UNION)

SQL so'rov tili haqidagi maqolalarni kiritishni davom ettiramiz. Bunda biz ikki jadvalni bir-biri bilan birlashtirishni ko'rsatib o'tamiz, buning uchun bizga quyidagi ikkita(test1, test2) jadvallar kerak bo'ladi.

170

NAME	FAMIL	YEAR
Phil	Jones	1992
Tom	Cleverley	1989
Adnan	Januzaj	1995
Wayne	Rooney	1985
Ashley	Young	1985

NAME	FAMIL	NUMBER
Phil	Jones	4
Ashley	Young	18
Ander	Herrera	21

Jadvallar bir-birlari bilan har xil ko'rinishda birlashtirilishi mumkin. Dastlab ikki jadvalni to'liq bir biri bilan birlashtiramiz, buning uchun biz «Union» kalit so'zidan foydalanamiz. Bu birlashtirishda ikkita jadvaldagi barcha ma'lumotlar chiqariladi. Barcha ustundagi bir xil bo'lgan ma'lumotlar esa, bir marotaba chiqariladi, ya'ni bir xil ma'lumotlar ekranga chiqmaydi.

*select name, famili from test1
union
select name, famili from test2;*

NAME	FAMIL
Adnan	Januzaj
Ander	Herrera
Ashley	Young
Phil	Jones
Tom	Cleverley
Wayne	Rooney

Yuqoridagi so'rovda birinchi jadval (test1)ga ikkinchi jadval (test2) birlashtirilmogda. So'rovdan ko'rib turibizki, ekranga chiqarilayotgan ikkala jadval ustun nomi va tiplari bir xil bo'lishi lozim, agar ustunlar bir xil bo'lmasa quyidagi xatoilk paydo bo'ladi.

*select name, famili, year from test1
union
select name, famili from test2;*

171

select name, family, year from test1

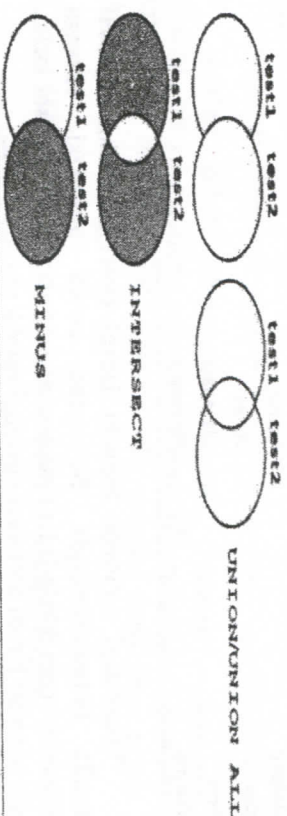
ERROR at line 1:
ORA-01789: query block has incorrect number of result columns

Keyingi birlashtirish, ikkala jadvaldagi barcha ma'lumotlarni chiqarish uchun ishlatiladi, bunda bizga «Union all» kalit so'zi yordam beradi. Bunda ikkala jadvaldagi hamma ma'lumotlar (bir xilari ham) chiqariladi.

*select name, family from test1
union all
select name, family from test2;*

NAME	FAMILY
Phil	Jones
Tom	Cleverley
Adnan	Januzaj
Wayne	Rooney
Ashley	Young
Phil	Jones
Ashley	Young
Ander	Herrera

Xulosa qiladigan bo'lsak, barcha birlashtirishlarni quyidagi rasmda ko'rish mumkin:



where komandasi.

SQL so'rov tilining asosiy shart qo'yish operatori (where) ni o'rganishda bizga "test" nomli quyidagi jadval kerak bo'ladi.

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000
Pablo	Aimar	30	25000
Carlos	Puyol	23	27000
Wayne	Rooney	18	21000
Sir	Alex	50	

Shu jadvalga so'rovlar berish orqali "where" operatorini o'rganamiz.

"where" operatori so'rovni ikkiga bo'lib beruvchi hisoblanadi. Undan so'ng so'rovga qo'yiladigan shartlar yoziladi. Uning umumiy ko'rinishi: *select * from test where ...;*

Har doim jadval nomidan so'ng "where" ishlatiladi. Misol ko'ramiz, "test" jadvalidagi, "maosh" ustuni 25000 ga teng bo'lgan barcha qatorlarni chiqarish kerak.

*select * from test where maosh=25000;*

ISM	FAMILIYA	YOSH	MAOSH
Pablo	Aimar	30	25000

Agar sonli ustun emas, balki biror matn yo so'zga tegishli bo'lgan ustunga shart qo'yib chiqarmoqchi bo'lsak, matn yoki so'z apostrof ichiga olinadi, aks holda xatolik yuzaga keladi.

*select * from test where ism='David';*

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000

Yuqoridagi shartlardan tashqari, yana quyidagi solishtirish belgilari orqali ham shart qo'yish mumkin bo'ladi.
= - tenglik belgisi

- > - kichik belgisi
 - >= - kichik yoki tenglikni anglatuvchi belgi
 - < - katta belgisi
 - <= - katta yoki teng belgisi
 - <> - teng emasni anglatuvchi belgi.
- Yuqoridagi belgilarga misol ko'ramiz. "test" jadvalidagi yoshi 20 dan kichik bo'lgan insonlarning familiyasini chiqaruvchi so'rov quyidagicha bo'ladi.

```
select familiya from test where yosh < 20;
```

ISM	FAMILIYA
Rooney	

Yoki bo'lmasa, ismi "David" bo'lmagan insonlar haqida ma'lumotni ekranga chiqarish so'rovi.

```
select * from test where ism <> 'David';
```

ISM	FAMILIYA	YOSH	MAOSH
Pablo	Aimar	30	25000
Carlos	Puyol	23	27000
Wayne	Rooney	18	21000
Sir	Alex	50	

Agar biz biror aniq qiymatni bilmasak, u holda qandaydir oraliq orqali bizga kerak qiymatni topib olishimiz mumkin bo'ladi. So'rovda biror oraliqni ishlatish uchun "between" va "and" operatorlaridan foydalanamiz. Misol, yoshi 20 dan 25 gacha bo'lgan insonlarning maoshini chiqaruvchi so'rov tuzish kerak.

```
select maosh from test where yosh between 20 and 25;
```

MAOSH
30000
55000
27000

Oraliq qiymatlar emas aksincha bir necha aniq qiymatlar berib, shu orqali kerakli ma'lumotlarni chiqarish kerak bo'lsa, "in" operatoridan

foydalanamiz. Misol, yoshi 18 va 30 ga teng bo'lgan ism va familiyalarni ekranga chiqaruvchi so'rov quyidagicha bo'ladi

```
select ism, familiya from test where yosh in (18, 30);
```

ISM	FAMILIYA
Pablo	Aimar
Wayne	Rooney

Agar biror ustunga hech qanday ma'lumot kiritilmagan bo'lsa, shu bo'sh ustunga tegishli qolgan ma'lumotlarni chiqarish kerak bo'lsa, "NULL" kalit so'zidan foydalanamiz. Misol, maoshi kiritilmagan ustunga tegishli bo'lgan boshqa ustun ma'lumotlarini chiqaramiz.

```
select * from test where maosh is null;
```

ISM	FAMILIYA	YOSH	MAOSH
Sir	Alex	50	

Jamlash operatorlari: count, min, max, avg, sum va group by

Hisobotlarni tuzishda ishlatiladigan va ma'lumotlarni ma'lum bir ko'rinishlarda jamlash uchun foydalaniladigan, "SQL" ning guruhli operatorlarini ishlatishda, bizga quyidagi "test" nomli jadval kerak bo'ladi.

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000
Pablo	Aimar	30	25000
Carlos	Puyol	23	27000
Sir	Alex	50	
Phil	John_	16	12000

Odada hisobotlar tuzishda, ma'lum bir jadvaldan foydalaniladi va bu jadvaldagi ma'lumotlarning umumiy hajmi katta rol o'ynaydi. Hajm - jadvaldagi qatorlar sonidir. Bu so'rovni topishda bizga "count"

operatori yordam beradi. Misol, "test" jadvalidagi barcha qatorlarning sonini topish so'rovi tuzilsin:

```
select count(*) from test;
```

COUNT(*)

6

Ko'rib turganingizdek qiyin emas. Bu operatorni "distinct" operatori bilan ham ishlatish mumkin. Misol, yoshi har xil bo'lgan insonlarning soni qanchaligini aniqlovchi so'rov tuzilsin.

```
select count(distinct yosh) from test;
```

COUNT(DISTINCT YOSH)

5

"distinct" operatori bo'lgani uchun, 23 yoshi bir marotaba hisoblanmoqda.

Keyingi operatorimiz "sum" deb nomlanadi va faqat raqam tipidagi ustundagi ma'lumotlar uchun ishlatiladi. Barcha ustundagi sonlarning yig'indisini topib beradi. Misol, ismi "David" bo'lgan insonlarning maoshi yig'indisi topilsin.

```
select sum(maosh) from test where ism='David';
```

SUM(MAOSH)

85000

Statistika ma'lumotlari bilan shug'ullanuvchi korxonalarda, raqamlarning o'rtacha qiymati katta ahamiyatga ega bo'ladi. SQL bu masalani ham birgina operator bilan yechmoqda, bu operator "AVG" operatori. Misol, ismi 5 ta harfdan iborat bo'lgan insonlarning o'rtacha yoshi hisoblansin.

```
select avg(yosh) from test where ism like '____';
```

AVG(YOSH)

24.6666667

176

Keyingi operatorlarimiz yuqoridagi operatorlardan ham soddalar. Bu operatorlarimiz, sonli ustunlarning maksimal va minimal qiymatlarini aniqlab beradi. Misol, yoshi eng katta va eng kichik insonni topish so'rovi tuzilsin.

```
select max(yosh) from test;
```

MAX(YOSH)

50

```
select min(yosh) from test;
```

MIN(YOSH)

16

Agar so'rovda guruhli operatorlardan boshqa operatorlar ham ishlatilsa, guruhli operatorlarni "guruh" ekanligini aytuvchi "group by" operatoridan foydalanish kerak. Agar bir xil ma'lumotlar bo'lsa, guruhlanadi, aks holda natijalar guruhlannagan holda chiqariladi. Misol, ism bo'yicha guruhlani, guruhlangan qatorlarni o'rtacha yoshini chiqaruvchi so'rov tuzish kerak.

```
select avg(yosh), ism from test group by ism;
```

AVG(YOSH)

ISM

30	Pablo
50	Sir
22	David
23	Carlos
16	Phil

Yana bitta misol ko'ramiz, bu misolni yaxshilab o'rganib chiqing. Jadvaldan insonlarning yoshi, ismi va familiyasiga oid ma'lumotlar chiqarilsin, agar yuqoridagi ma'lumotlari bir hil bo'lgan insonlar mavjud bo'lsa, ularning kichigi natija sifatida chiqarilsin.

```
select min(yosh), ism, familya from test group by ism, familya;
```

177

MIQYOSH	ISM	FAMILIYA
23	Carlos	Puyol
30	Pablo	Aimar
50	Sir	Alex
16	Phil	John
21	David	Beckham
23	David	Luiz

So'rovdan ko'rinib turibdiki, natijadagi ism va familiya ustuni guruhlanmoqda, lekin bir hil ismli va familiyali inson bo'lmagani uchun, "avg(yosh)=yosh" ga bo'lmogda va so'rov quyidagi oddiy so'rovga aylanmoqda.

select yosh, ism, familiya from test;

SQLda qism so'rovlar: in, exists, not exists.

SQL so'rov tilida qism so'rovlar tuzishda SQLning kalit so'zlaridan foydalanamiz. Buning uchun bizga quyidagi jadval kerak bo'ladi.

ID	NAME	YEAR
1	Marcos	1990
2	Antonio	1985
3	Jonny	1988
4	Shinji	1989
5	Rafaeli	1990
6	Rafaeli	1990

Dastlabki so'rovda «in» kalit so'zidan foydalanamiz. Qism dasturdan chiqqan natijalar, asosiy so'rovda, «in» kalit so'zi orqali tanlanib, mos kelganlari ekranga chiqariladi. Misol ko'raniz,

*select * from test where year in(select year from test where year>1988);*

ID	NAME	YEAR
6	Rafaeli	1990
5	Rafaeli	1990
1	Marcos	1990
4	Shinji	1989

Bu so'rovni qism so'rovidan natija chiqqach, quyidagicha kichik so'rovga aylanib qoladi (bazadagi axborotlardan kelib chiqqan holda).

*select * from test where year IN(1989, 1990);*

ma'nosi, 1989 va 1990 yillarda tug'ilgan o'yinchilarni ekranga chiqarib.

Qism so'rov tuzishda ishlatiladigan navbatdagi kalit so'zimiz «Exists» deb nomlanadi. Bu kalit so'z orqali biror ustun emas, butun boshli so'rov tekshiriladi. Agar qism dasturdan «true» natija chiqsa, ya'ni qism so'rovdan jadvalda mavjud bo'lgan biror axborot chiqsa, asosiy so'rov ishlaydi, aks holda hech qanday so'rov ishlamasdan, ekranga hech narsa chiqmaydi.

*select * from test where exists(select year from test where year=1990);*

ID	NAME	YEAR
1	Marcos	1990
2	Antonio	1985
3	Jonny	1988
4	Shinji	1989
5	Rafaeli	1990
6	Rafaeli	1990

agar qism so'rovdan natija olinmasa,

*select * from test where exists(select year from test where year=1990);*

asosiy so'rov ishlamaydi. Bu operator uchun qanday natija chiqqani qiziq emas, balki natija chiqqan yoki chiqmagani qiziq. Bunga teskari bo'lgan operator «Not exists» deb nomlanib, ishlashi ham «exists»ning teskari.

Yuqoridagi operatorlar oddiy so'rovlar uchun ham, qism so'rovlar uchun ham ishlataveriladi, lekin navbatdagi operatorlar faqat qism so'rovlar bilan ishlatiladi. Ular quyidagilardan iborat: ANY, ALL, SOME..

«ANY» va «SOME» bir xil operator bo'lib, istalganini ishlatishingiz mumkin, ikkisi ham bir xil natija chiqadi. Bu operatorlarning ishlashi, qism so'rovdan chiqqan natijani istalgani asosiy so'rov uchun qiymat bo'lsa, asosiy so'rov natija beraveradi. Bu operator «in» operatori bilan bir xil, faqat bular «=» dan boshqa «<», «>», «<=», «>» belgilarni ham ishlatish mumkin (in da faqat «=»).

*select * from test where id>any(select id from test where id>4);*

ID	NAME	YEAR
6	Rafaeli	1990

«ALL» qism so'rovdagi barcha natijalar, asosiy so'rovdagi shartini bajarishi lozim, shunda barcha shartdan o'tgan natija ekranga chiqadi.

*select * from test where id<all(select id from test where id>3);*

ID	NAME	YEAR
1	Marcos	1990
2	Antonio	1985
3	Jonny	1988

SQL funksiyalari: lower, upper haqida

So'rov tuzishda juda ko'p ishlatiladigan va juda kerakli bo'lgan ba'zi funksiyalar so'rov kodlarida tez-tez ishlatilib turiladi va shu bilan birga o'rganish ham qiyin emas. Bu kichik funksiyalarni o'rganishda yana o'sha biz bilgan "test" jadvali yordam beradi.

ISM	FAMILIYA	YOSH	MAOSH
David	Luiz	23	30000
David	Beckham	21	55000
Pablo	Aimar	30	25000
Carlos	Puyol	23	27000
Sir	Alex	50	
Phil	John	16	12000

Demak boshladik, dastlabki funksiyamiz "LOWER" deb ataladi. Yozuv ko'rinishidagi ustunlarni, biror qiymat bilan solishtirishda, kerakli natijaga erishish qiyinchilik tug'diradi (sababi, katta kichik yozuvlar qo'shtirmoq ichida farqlidir). Bu funksiya, solishtirilyotgan yozuvni barcha harflarini kichik harflarga o'tkazib beradi va siz bemaol kichik harflar bilan solishtirishingiz mumkin bo'ladi. Misol, familiyasi BECKHAM bo'lgan inson haqida ma'lumot chiqaramiz (biz, bazada bu familiya katta yoki kichik harfda yozilganini bilmaymiz va familiya ustundagi barcha familiyalarni kichik harfga o'tkazib tekshiramiz, ya'ni Beckham => beckham, Aimar => aimar, Luiz => luiz ko'rinishiga o'tkazilib, so'ng solishtiriladi):

*select * from test where lower(familya)=beckham;*

ISM	FAMILIYA	YOSH	MAOSH
David	Beckham	21	55000

Barcha familiyalarni katta harfga o'tkazib solishtiradi(Beckham => BECKHAM, Aimar => AIMAR, Luiz => LUIZ,... ko'rinishiga o'tkazilib solishtiriladi).

*select * from test where upper(familya)=BECKHAM;*

ISM	FAMILIYA	YOSH	MAOSH
David	Beckham	21	55000

Having bo'limi group by bo'limi bo'lgan hollarda ishlaydi. Having ni ishlatilishi deyarli where bilan bir hil. Faqat having bo'limidagi so'rov butun jadvalarga emas, where bo'limi orqali olingan natijaga yoziladi. Quyida birdan ortiq takrorlanuvchi familiyalar va ularning takrorlanishlar sonini chiqaruvchi so'rov keltirilgan:

Select familiya, count(familiya) as soni
From test Group by familiya
Having count(familiya)>2

Jadvallarga yangi yozuv qo'shish INSERT operatori orqali amalga oshiriladi:

INSERT INTO <jadvallar nomi>(<Maydonlar ro'yxati>)
VALUE(<qiymalar ro'yxati>)

Misol:

INSERT INTO test(familiya, ismi) Value("Mallayev", "Oybek");

Mayjud yozuvni tahrirlash(qiymaini yangilash) UPDATE operatori orqali amalga oshiriladi.

UPDATE <jadval nomi> SET <maydon>=<qiyma> Where <shart>

Misol: Mallayev Oybekning tug'ilgan kunini 1989 ga o'zgartirish.

Update test set tug_yili=1989 Where familiya="Mallayev" and ism="Oybek";

Jadvallardan yozuvlarni o'chirish DELETE operatori orqali amalga oshiriladi.

Delete from <jadval nomi>

Where <shart>

Misol: Ismi Oybekga teng bo'lgan qatorni o'chirish.

Delete from test Where ism="Oybek"

Bu so'rov orqali Ismi Oybek bo'lgan barcha qatorlar o'chiriladi.

SQL so'rovlarini baholalanishi bo'yicha maxsus jadval

2-jadval

MBBT NOMI				MAX
QISQACHA MAZMUNI				BALL
Jadvallar uchun (Har bir jadval uchun 1 balligacha)				4
Axborot modeli uchun (real bog'langanligini ko'rsatish, to'g'ri va teskari)				1
Yozuvlar soni uchun (har bir jadval uchun 10 ta yozuv, asosiy jadvalda 50 ta yozuv)				1
To'planadigan umumiy ball				6
AMALIY QISM				
t.r.	operator turi	so'rov soni (min)	qisqacha mazmuni	ball (max)
ASOSIY OPERATORLAR				
1	INSERT INTO	3	turli tipdagi va bir nechta yozuvlarni qo'shish	
2	SELECT	5	cheklangan sondagi yozuvlarni chiqarish	
3	DELETE	2	shart orqali o'chirish, hammasini o'chirish	
4	UPDATE	3	turli tipdagi va bir nechta yozuvlarni hamda hammasini	
AMALLAR				
5	matematik amallar	2	kamida ikkitasi birga bo'lsin	
6	munosabat amallar	3	kamida ikkitasi birga bo'lsin	
7	manitivy amallar	2	kamida ikkitasi birga bo'lsin	
8	arralash	2		
OPERATORLAR				
9	DISTINCT	2		
10	WHERE	2		
11	BETWEEN	2		
12	IN	2		
13	LIKE	4	fakat boshidan, oxiridan, ixtiyoriy joydan	
14	ISNULL	2	not bilan ham	
15	NOT	3	murakkab amallar bilan	
16	ORDER BY	2	xarima variantlar uchun	
17	HAVING	2		
STATIK FUNKSIYALAR				
18	MAX	1		
19	MIN	1		
20	COUNT	1		
21	SUM	1		
22	COUNT(*)	1		
23	AVG	1		
24	ARRALASH	3		

YORDAMCHI SO'ROVLAR (SUBQUERY - PODCHINENNOE ZAPROSO) BILAN ISHLASH

25	YORDAMCHI SO'ROVLAR	3	kamida 3 ta statik funksiyalar, amallar, operatorlardan foydalang
26	IN	2	
27	EXISTS	2	
28	ANY	2	
	ALL	2	
MAXSUS OPERATORLAR			
29	CAST	1	
30	CONVERT	2	
31	CASE	3	
32	COALESCE	1	
33	NULLIF	1	
BOG'LANISHLAR			
34	ODDIY	2	
35	INNER JOIN	2	
36	LEFT OUTER JOIN	2	
37	RIGHT OUTER JOIN	2	
38	FULL OUTER JOIN	2	
39	CROSS JOIN	2	
40	UNION	2	
YOZUV FUNKSIYALARI			
41	FIRST	1	
42	UCASE	1	
43	LCASE	1	
44	UPPER	1	
45	LOWER	1	
46	LAST	1	
47	MID	1	
48	SUBSTRING	1	
49	LEN	1	
50	NOW	1	
51	SARLAHANI NOMLASH	1	
52	FORMAT	1	

SQL so'rovlarini bajarilishi bo'yicha maxsus jadval

	INSERT INTO GA DOIR	3-jadval
1	INSERT INTO Millat (millat, fullinfo) VALUES ('', '')	
2	INSERT INTO Millat (millat, fullinfo) VALUES ('', '')	
3	INSERT INTO Fugaro (fugaro, ishStaji, maoshi, ishManzili, uyManzili, tugilganSana,	

	lavozimi, millati)	
	VALUES('121230000,1.1','1,1)	
	SELECT GA DOIR	
4	SELECT id, millat, fullinfo FROM Millat	
5	SELECT TOP N id, millat, fullinfo FROM Millat	
6	SELECT fugaro, ishStaji, maoshi FROM FUGARO	
	DELETE GA DOIR	
7	DELETE FROM Millat	
8	DELETE FROM Millat WHERE millat = ''	
9	INSERT INTO Millat(millat,fullinfo) VALUES ('TATAR','TATAR')	
	UPDATE ga dior	
10	UPDATE Millat SET millat = 'tatar' WHERE millat = 'TATAR'	
11	UPDATE Millat SET millat = 'tatar' WHERE millat = 'TATAR'	
12	UPDATE geografikManzili SET geografik M1 'Tashkent sh', pochtaIndex = '100' WHERE geografik M1 = 'Tashkent'	
	Amallar	
13	SELECT fugaro, maoshi*12 FROM Fugaro	
14	SELECT fugaro, ishStaji FROM Fugaro where Fugaro.maoshi>130000*4	
15	SELECT fugaro Fugaro.maoshi*0065 FROM Fugaro	
16	UPDATE Fugaro SET maoshi = maoshi*1.1	
17	SELECT fugaro, maoshi, ishStaji FROM Fugaro where (maoshi>1300*5 and ishStaji >=7)	
18	SELECT fugaro, ishStaji FROM Fugaro where not(maoshi<1300*5)	
19	DISTINCT	
20	SELECT DISTINCT geografik M1 FROM geografikManzili	
21	BETWEEN	
22	SELECT fugaro, maoshi, ishStaji FROM Fugaro WHERE ishStaji between 5 and 12;	
	IN ga dior	
23	SELECT fugaro, maoshi, ishStaji FROM Fugaro WHERE ishStaji not in(5,10)	
	LIKE ga dior	
24	SELECT fugaro, maoshi, ishStaji FROM Fugaro WHERE fugaro like 'A%'	
25	ISNULL	
26	SELECT fugaro, maoshi, ishStaji FROM Fugaro WHERE tugilganSana is not null	
27	SELECT geografik M1, kucha uy FROM geografikManzili WHERE pochtaIndex is null	
	NOT ga dior	
28	SELECT fugaro, maoshi, ishStaji FROM Fugaro where not maoshi>1300*5 and ishStaji >=7	
	ORDER BY ga dior	
29	SELECT fugaro FROM Fugaro order by fugaro desc	
30	SELECT fugaro FROM Fugaro order by fugaro	
31	group by	
32	SELECT millat FROM Fugaro group by millat	
	HAVING ga dior	
33	SELECT millat FROM Fugaro group by millat having millat>1	
34	Statik funksiyalar	
35	SELECT 'MAX'=MAX(maoshi), 'MIN'=MIN(maoshi) FROM Fugaro	

36	SELECT COUNT(fugaro) FROM Fugaro where ishStaji >= 5
37	SELECT SUM(maoshi) FROM Fugaro
38	SELECT COUNT(DISTINCT millat) FROM Fugaro
39	SELECT COUNT(*) FROM Fugaro
40	SELECT AVG(maoshi) FROM Fugaro
IN ga doir	
41	SELECT fugaro FROM Fugaro WHERE millat IN (SELECT id FROM millat where millat = 'uzbek')
42	EXISTS
43	SELECT Fugaro, fugaro FROM Fugaro WHERE EXISTS (SELECT id FROM millat where millat = 'turman')
44	ANY and ALL
45	SELECT millat FROM Fugaro group by millat having millat = ANY(SELECT id FROM millat)
46	SELECT millat FROM Fugaro group by millat having millat = ALL(SELECT id FROM millat)
CONVERT and CAST ga doir	
47	SELECT geografik MI FROM geografikManzil where CONVERT(int, telefonHome) > 0
48	SELECT CAST (millat as char) FROM Millat
CASE ga doir	
49	SELECT fugaro, case when maoshi <= 500000 then 'PAST' when (maoshi > 500000 and maoshi < 1000000) then 'URTACHA' else 'YAXSHI' end FROM Fugaro;
COALESCE ga doir	
50	SELECT geografik MI coalesce(telefonHome, -1 FROM geografikManzil
51	NULLIF
52	SELECT geografik MI NULLIF (telefonHome) AS NINN FROM geografikManzil
Bog'lanishlar	
53	SELECT Fugaro, millat FROM Fugaro, Millat where Fugaro.millat = Millat.id
54	SELECT Fugaro, millat FROM Fugaro inner join Millat ON Fugaro.millat = Millat.id
55	SELECT Fugaro, millat FROM Fugaro Full outer join Millat ON Fugaro.millat = Millat.id
56	SELECT Fugaro, millat FROM Fugaro Left outer join Millat ON Fugaro.millat = Millat.id
57	SELECT Fugaro, millat FROM Fugaro RIGHT outer join Millat ON Fugaro.millat = Millat.id
58	SELECT Fugaro, millat FROM Fugaro, Millat
59	SELECT Fugaro, millat FROM Fugaro cross join Millat
60	SELECT Fugaro, millat FROM Fugaro cross join Millat where Fugaro.millat = Millat.id

UNION ga doir	
61	(SELECT Fugaro FROM Fugaro where maoshi < 6*130000) UNION (SELECT Fugaro FROM Fugaro inner join Millat ON Fugaro.millat = Millat.id where Millat.millat = 'uzbek')
62	YOzuvi funksiyalari
63	SELECT First(fugaro) FROM fugaro
64	SELECT Ucase(fugaro) FROM Fugaro
65	SELECT Lcase(fugaro) FROM Fugaro
66	SELECT UPPER(fugaro) FROM Fugaro
67	SELECT LOWER(fugaro) FROM Fugaro
68	SELECT LAST(fugaro) FROM Fugaro
69	SELECT MID(fugaro, 1,1) FROM Fugaro
70	SELECT fugaro FROM Fugaro where SUBSTRING(fugaro, 1,1) = 'A'
71	SELECT lent(fugaro) FROM Fugaro
72	SELECT fugaro, Now() as SANA FROM Fugaro
73	SELECT Round(maoshi) FROM Fugaro
74	SELECT fugaro, FORMAT(Now(), 'YYYY-MM-DD') AS PerDate FROM Fugaro
Commit RollBack ga doir	
75	BEGIN TRANSACTION W1; INSERT INTO MILLAT(MILLAT, FULLINFO) VALUES ('TATAR12 5', 'TATAR12 5'); UPDATE GEOGRAFIKMANZIL SET GEOGRAFIK MI = 'TOSHKENT', POCHTAINDEX = '100' WHERE GEOGRAFIK MI = 'TASHKENT', COMMIT TRANSACTION W1;
76	BEGIN TRANSACTION WORK; INSERT INTO MILLAT(MILLAT, FULLINFO) VALUES ('TATAR-3'); UPDATE GEOGRAFIKMANZIL SET GEOGRAFIK MI = 'TASHKENT', POCHTAINDEX = '100' WHERE GEOGRAFIK MI = 'TSHKENT', ROLLBACK WORK;
77	BEGIN TRANSACTION WORK; INSERT INTO MILLAT(MILLAT, FULLINFO) VALUES('KKKK1'); DELETE FROM MILLAT WHERE MILLAT LIKE 'KK'; DECLARE @MAX1 INT; SELECT @MAX1 = MAX(MAOSHI) FROM FUGARO; IF(@MAX1 < 1300000) COMMIT WORK ELSE ROLLBACK WORK;
78	Procedure lar bilan ishlash ALTER PROCEDURE TEST as BEGIN DECLARE @i INT; SELECT @i = COUNT(fugaro) FROM Fugaro SELECT @i; END EXEC TEST


```

ALTER PROCEDURE TEST as
BEGIN DECLARE @i INT;
SELECT @i = COUNT(tugano) FROM Fugano return @i;
END
DECLARE @return_value int EXEC @return_value = TEST if (@return_value) < 10
Print N'10 dan kam' else
Print N'10 dan ko'p'

```

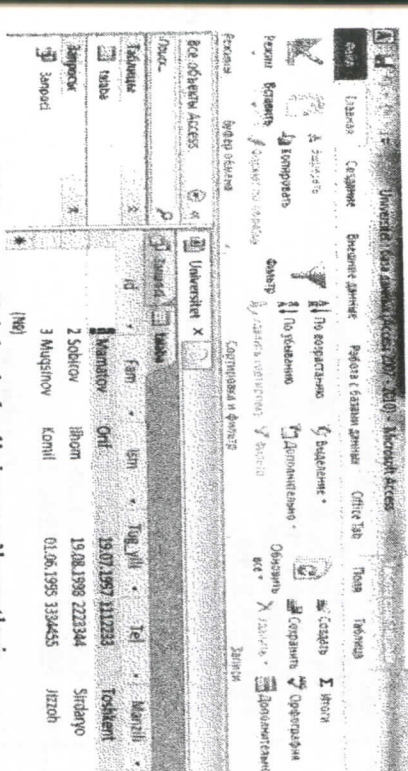
3. Ma'lumotlar bazasini yaratish bo'yicha uslubiy ko'rsatmalar

MS Access dasturida SQL so'rovlaridan foydalanish:

Oldingi bo'limlarda MS Access dasturida ma'lumotlar bazasini yaratish ko'rsatib o'tilgan edik. Endi esa MB dagi jadvallar asosida SQL so'rovlarni amalga oshirish ko'rsatiladi.

Masalan: **talaba** jadvalida familiyasi M harfidan boshlanuvchi talabalarni chiqaruvchi so'rov yaratish.

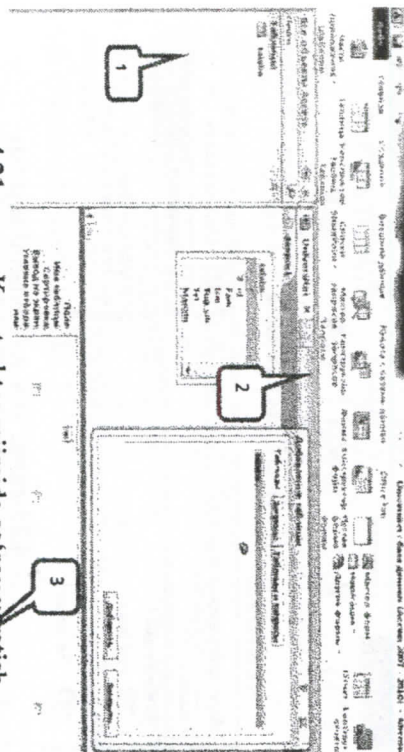
1. MS Access dasturi ishga tushiriladi va unda **Universitet** nomli MB yaratiladi;
2. MB da **talaba** jadvali yaratiladi;
3. **talaba** jadvalida **Fam, Ism, Tug_yili, Tel, Manzili** nomli maydonlarni tashkil etiladi va to'ldiriladi;



4.20-rasm. Talaba jadvalining ma'lumotlari

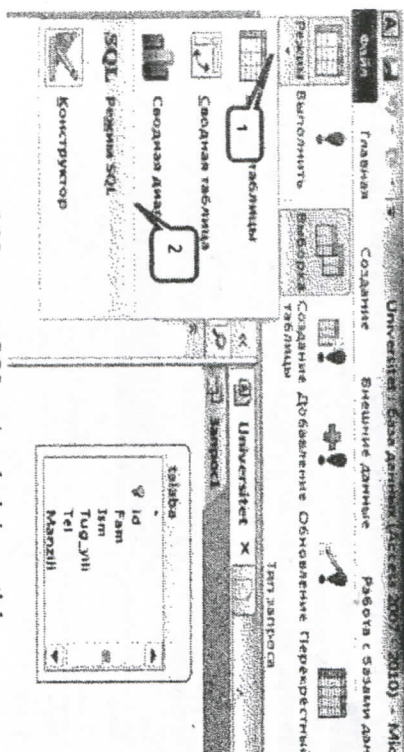
4. talaba jadvalidan SQL so'rovlarini yaratiladi:

“Создание” меню bo'limidan “Конструктор запросов” anlaniladi va 4.21 - rasmda ko'rsatilgan ketma – ketliklar bajariladi.



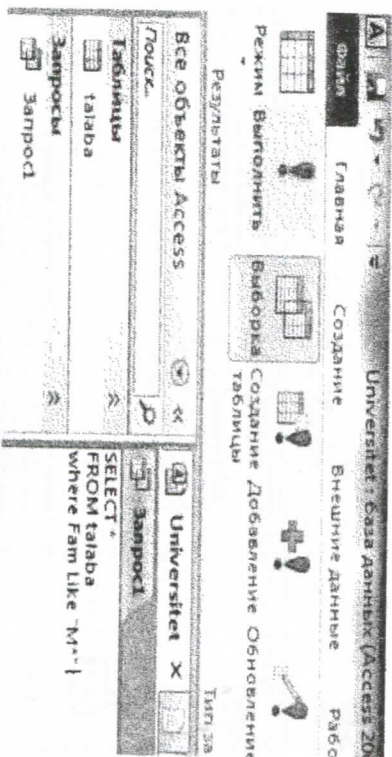
4.21-rasm. Конструктор rejimida so'rov yaratish

Jadval tanlangandan keyin, “Режим” aktivlashadi va 4.22- rasmda ko'rsatilgan ketma – ketliklar bajariladi.



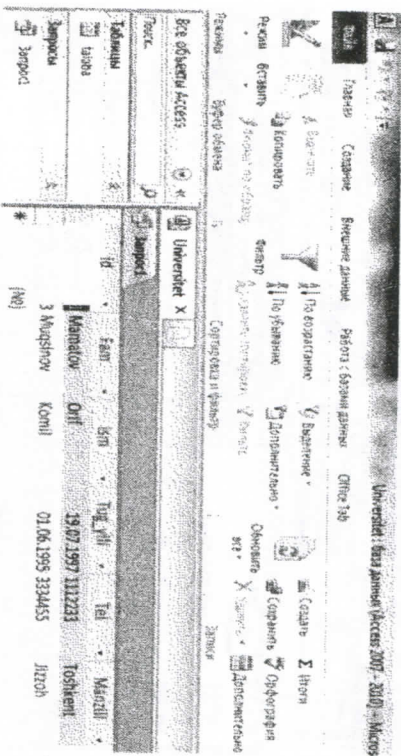
4.22-rasm. SQL so'rovlarini yaratish

So'rov yozilgandan keyin “Выполнить” tugmasi bosiladi.



4.23-rasm. SQL so'rovlarini bajarish

MS Accessda SQL so'rov yaratishda % o'runga * qabul qilingan. "Выполнить" tugmasi bosilgandan keyin quyidagi jadval paydo bo'ladi:



4.24-rasm. SQL so'rovining natijasi

Nazorat savollari:

1. SQL nima?
2. SQL ning maqsad va vazifasi nimalardan iborat?
3. Ma'lumotlarni chaqirish operatori nima?
4. Ma'lumotlarni yangilash operatori nima?

5. Ma'lumotlarni yozish operatori nima?
6. Ma'lumotlarni o'chirish operatori nima?

4.4. Visual C++ da MS Access ning ma'lumotlar bazasini SQL so'rovlari asosida tahrirlash usullari

Mavzuning maqsadi

Visual C++ ning Console Application multihitida MS Access ma'lumotlar bazasini yaratuvchi, unga ma'lumotlarni yozib va undan ma'lumotlarni o'qib hamda o'zlashtirilgan ma'lumotlarni tahrirlab ekraanga chiqaruvchi dastur tuzish.

I-dastur: Command va Datareader sinf ob'ektlari yordamida MS Accessda yaratilgan ma'lumotlar bazasining jadvalidagi barcha ma'lumotlarni o'quvchi dastur

I-dastur: Shunday dastur tuzaylikki, kam kodlarni ishlatib, ma'lumotlar bazasi (MB) ning jadvalidagi barcha ma'lumotlarni ekraanga chiqarsin. Buning uchun zamonaviy ADO.NET texnologiyasi ishlatiladi. Bunga 4 ta ob'ekt kerak bo'ladi. 1. Connection ob'ekti - MB bilan ulanishni ta'minlaydi. 2. Command ob'ekti - MB bilan SQL ifodalarni bog'lashni ta'minlaydi. 3. Dataset va Datareaders ob'ektlari yordamida so'rovlar natijasini ko'rish imkonini beradi.

Dasturda MB bo'yicha 4 ta asosiy xarakatlarni ko'rib chiqiladi: **Select** (MB jadvalidan yozuvlarni tanlash), **Insert** (bazaga yozish), **Update** (MBning jadvalidagi ma'lumotlarni yangilash), **Delete** (Jadvaldan ma'lumotlarni o'chirish uchun).

Visual Studio ni ishga tushiramiz va Console Application CLR multihitida ishlovchi loyihani yaratiladi va quyidagi kodlar kiritiladi:

1. // Access_console.cpp : main project file.
2. #include "stdafx.h"
3. using namespace System;
4. using namespace System::Data; // OleDb; // SqlServer;
5. int main(array<System::String^> args){
6. //Konsolning matniga rang berish;
7. Console::ForegroundColor = ConsoleColor::Yellow;
8. // Connection sinfi ob'ektlarni yaratish
9. auto Ulanish = new OleDbConnection();
10. // Ulanish qatoriga yuborish:


```

11. Ulanish->ConnectionString = "Data Source=\"Oybek.mdb\";user " +
    "ID=Admin;Provider=\"Microsoft.Jet.OLEDB.4.0\";";
12. Ulanish->Open();
13. // Command sinfi ob'ektni yaratish:
14. auto Komanda = gcnew OleDbCommand();
15. Komanda->Connection = Ulanish;
16. // Uni SQL-komandalarga jo'natish :
17. Komanda->CommandText = "Select * From [Talaba]";
18. // Xamma yozilganlar tanlanadi va FIO ustuni bo'yicha
    saralanadi:
19. // Komanda->CommandText = "Select * From [BD telefonov] order by
    FIO";
20. // "Nomer p/p" usutuni bo'yicha esa:
21. // Komanda->CommandText = "Select * From [BD telefonov] ORDER BY
    'Nomer p/p'";
22. // SQL-komandasini bajarish:
23. OleDbDataReader ^ CHitatel = Komanda->
    ExecuteReader(System::Data::CommandBehavior::CloseConnection);
24. Console::WriteLine("Jadval BD:\n");
25. while (CHitatel->Read() == true)
26. // siki xali xamma yozuvlarni o'qigani yo'q
27. // CHitatel->FieldCount - qatorlardagi maydonlar soni.
28. // Bu yerda 3 ta maydon: 0, 1 i 2.
29. // Minus prijmaet stroku vlevno:
30. Console::WriteLine("{0,-3} {1,-15} {2,-15}",
    CHitatel->GetValue(0),
    CHitatel->GetValue(1), CHitatel->GetValue(2));
31. CHitatel->Close(); Ulanish->Close();
32. CHitatel->Close(); Ulanish->Close();
33. // qandaydir tugmani bosilguncha kutib turadi
34. Console::Readkey(); return 0;

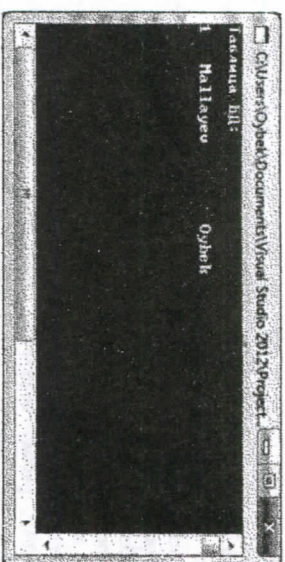
```

Dastur tahlili:

Dastur kodidan ko'rinish turibdiki, avval **Connection** sinfining Ulanish ob'ekti yaratildi va bog'lanish satriga yuborildi. Bog'lanish satiri **mdb** faylga hamma ruxsatni oldi. Bu esa fayl va kataloglarning yo'llarini aniq o'qish uchun mo'ljallangan. Keyin **Command** sinfi ob'ekti yaratildi va uni SQL komandasiga yuborildi.

Select * From [BD telefonov]

Bu holda BD telefonov jadvalidagi hamma yozuvlar tanlanadi. FIO bo'yicha (**ORDER BY FIO**) va nomeri bo'yicha (**ORDER BY 'Nomer p/p'**) saralanadi. Keyin SQL buyruqlarini bajarish uchun **DataReader** sinfi ob'ekti ishlatildi. Dastur natijasi 4.25-rasmida keltirilgan:



4.25-rasm. MIBning jadvalini konsol muhitida chiqarish

2-dastur: Console Application muhitida MS Access ma'lumotlar bazasini yaratuvchi dastur tuzish

Visual Studio 2012 ning **Console Application** muhitida **new BD.mdb** nomli fayl (Ma'lumotlar bazasi)ni yaratishda avval ushbu bazada hech qanday jadval bo'lmaydi. Jadvallarni dasturlash kodi yordamida yaratiladi. Ushbu dasturda **ADO.NET** texnologiyasi ishlatilmaydi.

Visual Studio dasturi ishga tushiriladi. **New Project** oynasidan **CLR** bo'limi tanlanadi va **Console Application CLR** muhitida yangi loyiha dasturi yaratiladi. **ADOX DDL** kutubxonasini loyihaga qo'shish uchun dasturning **Project** mensidan **Properties Add Reference** tanlanadi, ochilgan oynadan **COM** bo'limiga o'tiladi va **Microsoft ADO Ext. 2.8 for DDL and Security** belgilanib **OK** tugmachasi bosiladi. Natijada dasturdan **ADOX** kutubxonasi funksiyalariga murojaat qilish imkoniyatlari ochiladi.

```

1. // Access_mb_yaratish_Console.cpp : main project file.
2. #include "stdafx.h"
3. using namespace System;
4. using namespace System::Windows::Forms;
5. using namespace System::Data::OleDb;
6. using namespace System::Data::SqlClient;
7. int main(array<System::String ^> ^args){
8. ADOX::Catalog ^ Katalog = gcnew ADOX::Catalog();
9. try {
10. Katalog->Create("Provider=Microsoft.Jet. " + "OLEDB.4.0;Data
    Source=C:\\new_BD.mdb");
11. MessageBox::Show("Ma'lumotlar bazasi C:\\new_BD.mdb
    muvofaqiyatli yaratildi");
12. catch (System::Runtime::InteropServices::COMException ^
    Situasija)

```



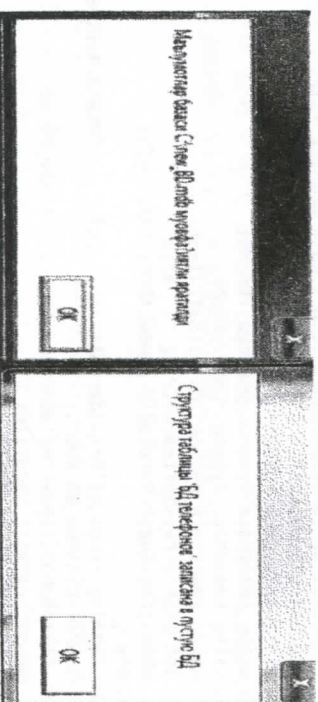
```

13. { MessageBox::Show(Situasiya->Message); }
14. finally
15. { Katalog = nullptr; }
16. auto Ulanish = gcnw OleDbConnection(
17. "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\new_BD.mdb");
18. // podklyucheniya ni ochish:
19. Ulanish->Open();
20. // Command sinfining obe'ktini SQL so'rovlari bilan yaratish
21. auto Komanda = gcnw OleDbCommand("CREATE TABLE [" +
22. "BD telefonov"] ([Nomer p/p] counter, [FIO] ch" +
23. "ar(20), [Nomer telefona] char(20))", Ulanish);
24. try { // SQL buyruqlarini bajarilishi:
25. Komanda->ExecuteNonQuery();
26. MessageBox::Show("Struktura tablisi 'BD telefonov' zapisana v
    pustuyu BD"); }
27. catch (Exception ^ Situasiya) { MessageBox::Show(
    Situasiya->Message); }
28. Ulanish->Close(); return 0; }

```

Dastur tahlili:

MessageBox ob'ektiaga murojaat qila olinishi uchun dasturga yana bitta DLL kutubxona qo'shildi. Buning uchun oldingi holatda tushuntirilganidek menuning **Project->Properties->Add Reference** buyrug'i berildi va NET bo'limidan **System.Windows.Forms.dll** belgilandi. Dastur kodiga **Using namespace System::Windows::Forms;** havolasi kiritildi. Dastur quyidagi ko'rinishda ishlaydi: **ADOX::catalog** sinfi yaratiladi, uning **Great** funksiyasi bilan MB yaratiladi. **try...catch** bilan esa qayta ishlanadi. Agar dastur ishga tushsa **new_BD.mdb** nomli ma'lumotlar bazasi yaratiladi. Lekin uning ichida hech qanday jadval yo'q va quyidagi xabarnoma chiqadi.



4.26-rasm. MB ni yaratilganligi haqida dasturning natija oynasi

194

3-dastur: Console Application muhitida MS Access ma'lumotlar bazasining jadvallariga ma'lumotlar yozuvchi dastur tuzish

Ma'lumotlar bazasi bilan mulqot qilish uchun 2- dasturni ishlatib, xohlaganicha so'rovlarni amalga oshirish mumkin. Masalan MB ning jadvallariga **INSERT** so'rovi yordamida ma'lumot yozish mumkin. Buning uchun **Command** ob'ekti yaratiladi va unga **SQL** so'rovlari beriladi. Buning uchun **insert** so'rovi komanda ob'ektiaga uzatiladi: **"Komanda->Connection = Ulanish"**;

Dastur kodi quyidagicha:

```

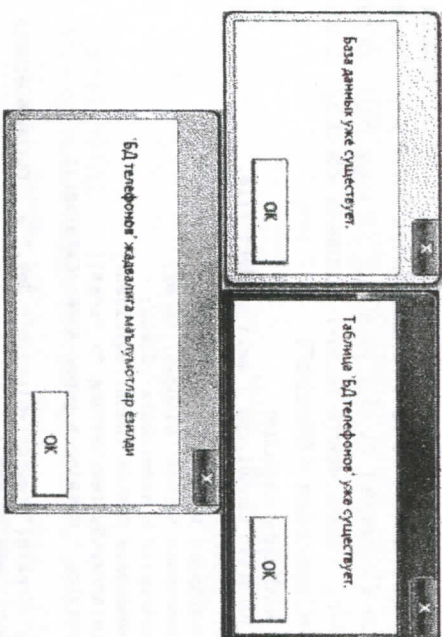
// Access_mb_yaratish_Console.cpp : main project file.
1. #include "stdafx.h"
2. using namespace System;
3. using namespace System::Windows::Forms;
4. using namespace System::Data::OleDb;
5. using namespace System::Data::SqlClient;
6. int main(array<System::String ^> args){
7. ADOX::Catalog ^ Katalog = gcnw ADOX::Catalog();
8. try {
9. Katalog->Create("Provider=Microsoft.Jet." + "OLEDB.4.0;Data
10. Source=C:\\new_BD.mdb");
11. MessageBox::Show("Ma'lumotlar bazasi C:\\new_BD.mdb
    muvaffaqiyatli yaratildi"); }
12. catch(System::Runtime::InteropServices::COMException ^
    Situasiya){
13. MessageBox::Show(Situasiya->Message); }
14. finally { Katalog = nullptr; }
15. auto Ulanish = gcnw OleDbConnection("Provider=
    Microsoft.Jet.OLEDB.4.0;Data Source=C:\\new_BD.mdb");
16. // podklyucheniya ni ochish:
17. Ulanish->Open();
18. //Command sinf obe'ktini SQL so'rovlari bilan yaratish
19. auto Komanda = gcnw OleDbCommand("CREATE TABLE [" + "BD
    telefonov"] ([Nomer p/p] counter, [FIO] ch" + "ar(20), [Nomer
    telefona] char(20))", Ulanish);
20. try { // SQL buyruqlarini bajarilishi:
    Komanda->ExecuteNonQuery(); MessageBox::Show("Struktura
    tabiliso" 'BD telefonov' zapisana v pustuyu BD"); }
21. catch (Exception ^ Situasiya) { MessageBox::Show(
    Situasiya->Message); }
22. Komanda = gcnw OleDbCommand("INSERT INTO [BD telefonov] (" +
    "FIO, [Nomer telefona] VALUES ('Sveta-X', '521-61-41')");
23. // MB jadvaliga yozish uchun ushbu komanda albatte //bo'lishi
    kerak:
24. Komanda->Connection = Ulanish;
25. // SQL buyruqlarini bajarish:
26. Komanda->ExecuteNonQuery();

```

195

28. MessageBox::Show("BD telefonov' jadvaliga ma'lumotlar yozildi");

Dastur natijasi quyidagicha:



4.27-rasm. Ma'lumotlar yozilganligi haqida dasturning natija oynasi

Mustakamlash uchun mashqlar

- Quyidagi topshiriqlar asosida MB jadvallardan bitisini tanlang va uni "Console Application" muhitida yarating.
 - Ushbu jadvalni SQL so'rovlar (**select, insert, update**) ni ishlatgan holda ma'lumotlar bilan to'ldiring, yangilang va ushbu ma'lumotlarni "Console Application" muhitida chiqaruvchi dastur tuzing.
- Mahalla ma'lumotlar bazasi;
 - Maktab o'quvchilarining ma'lumotlar bazasi;
 - Shifoxonaning qabul bo'limini ma'lumotlar bazasi;
 - Fakultetning ma'lumotlar bazasi;

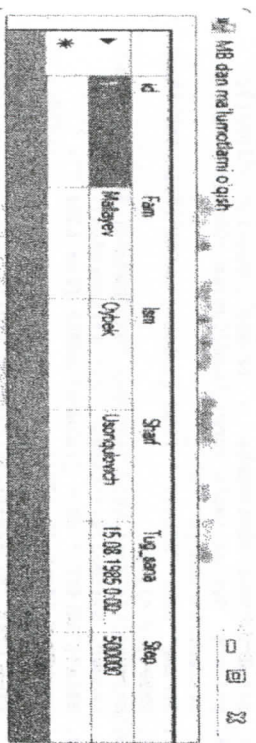
4-dastur: Command, DataReader sinf ob'ektlari va DataGridView komponentasi yordamida MBning jadvalidan ma'lumotlarni o'quvchi vizual dastur

Ushbu dasturni yaratishda Command, DataReader sinf ob'ektlari va DataGridView komponentasi kerak bo'ladi. DataGridView

komponentasidan MB ga ma'lumotlarni o'qish uchun foydalaniladi. Buning uchun Windows Application muhitida yangi loyiha yaratiladi va formaga DataGridView, Button komponentasi tashlanadi va Form1 da dastur kodi quyidagicha bo'ladi:

```
1. #pragma endregion
2. private: System::Void Form1_Load(System::Object^ sender,
3. System::EventArgs^ e) {
4. this->Text = "MB dan ma'lumotlarni o'qish";
5. auto Ulanish = gcnew OleDb::OleDbConnection("Data
6. Source='C:\\Oybek.mdb'; User =
7. 'ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
8. Ulanish->Open();
9. // Command obyekti yaratib, unga SQL-komandasi yuborish
10. auto Komanda = gcnew OleDb::
11. OleDbCommand("Select * From [talaba]", Ulanish);
12. // SQL-komandasi bajarish
13. auto Chitatel = Komanda->ExecuteReader();
14. // (CommandBehavior.CloseConnection)
15. auto Jadval = gcnew DataTable();
16. // Jadvalni to'ldirish
17. for(int i=0; i<6; i++)
18. {Jadval->Columns->Add(Chitatel->GetName(i));}
19. while (Chitatel->Read() == true)
20. {Jadval->Rows->Add(Chitatel->GetValue(0),
21. Chitatel->GetValue(1), Chitatel->GetValue(2),
22. Chitatel->GetValue(3), Chitatel->GetValue(4),
23. Chitatel->GetValue(5));}
24. // bu yerda 6 maydon: 0, 1, 2, 3, 4, 5 bor
25. Chitatel->Close(); Ulanish->Close();
26. dataGridView1->DataSource = Jadval; } }
```

Dastur ko'rinishi quyidagicha:



4.28-rasm. DataGridView komponentasiga MBdan ma'lumotlarni chiqarish

5-dastur: Command, Adapter va DataSet sinf ob'ektlari va DataGridView komponentasi yordamida MBning jadvalidan ma'lumotlarni o'quvchi vizual dastur

MB dan Adapter sinf ob'ekti bilan jadvallarini muhim ma'lumotlarini o'qish va ularni DataSet sinf ob'ektiga jo'natishni ko'rib chiqiladi. DataGridView komponentasidan foydalanib MB ning jadvalidagi ma'lumotlarni DataSet dan o'qish dasturiga juda qulay hisoblanadi. Misol sifatida quyidagi dasturni ko'rish mumkin:

Dasturning kodi:

```
1. #pragma endregion
2. void DBSet(String ^s){
3. this->Text = "MB dan jadvalni o'qish va so'rovlarini amalga
oshirish:";
4. auto Ulanish = gcnew OleDb::OleDbConnection( "Data
Source=\"C:\Oybek.mdbg\"; User = " +
5. "IDAdmin;Provider=\"Microsoft.Jet.OLEDB.4.0\";");
6. Ulanish->Open();
7. auto Komanda = gcnew OleDb::OleDbCommand(s, Ulanish);
8. // Adapter sinfi ob'ektini yaratamiz va SQL-so'rovini //amalga
oshiramiz
9. auto Adapter = gcnew OleDb::OleDbDataAdapter(Komanda);
10. // DataSet sinfi ob'ektini yaratamiz
11. auto NaborDannix = gcnew DataSet();
12. // DataSet ni SQL-so'rovini natijalari bilan //to'ldiramiz
13. Adapter->Fill(NaborDanno^x, "talaba");
14. // DataSet da satrlar ko'rinishi uchun XML ko'rinishiga
//o'tadi:
15. auto StrokXML = NaborDanno^x->GetXml();
16. // Komponentaga ma'lumot manbasini ko'rsatamiz:
17. dataGridView1->DataSource = NaborDannix;
18. // qidiriladigan ma'lumotlarning jadvalini nomini //ko'rsatamiz:
19. dataGridView1->DataMember = "talaba"; Ulanish->Close();
20. private: void Form1_Load(System::Object^ sender,
System::EventArgs^ e) {
21. String ^satr="SELECT * FROM [talaba] WHERE (Fam LIKE '%M%')";
22. DBSet(satr); }
23. private: void textBox1_TextChanged( System::Object^
sender, System::EventArgs^ e){ if(checkBox1->Checked){
24. DBSet("SELECT * FROM [talaba] WHERE (Fam LIKE '"+
textBox1->Text+"')");}
25. else if(checkBox2->Checked){ DBSet("SELECT * FROM [talaba] WHERE
(Fam LIKE '"+ textBox1->Text+"')");}
26. else if(checkBox3->Checked){ DBSet("SELECT * FROM [talaba]
WHERE (Telefoni LIKE '"+textBox1->Text+"')");}
```

198

27. else { MessageBox::Show("qidirish turlaridan bittasini tanlang"); } } } };

Dasturning ko'rinishi quyidagicha:

ID	Fam	Ism	Otak	Shif	Ushabdomoch	Tog_sana	Telefon
1	Makguyev	Oqbek	Oqil	Ushabdomoch	20.07.1987	1274455	
2	Salmov			Qadrovich	23.05.86	2315466	
3	Ramov	Korni		Ladovochi	24.08.1977	7778622	

4.29-rasm. MBdan barcha ma'lumotlarni chiqarish holati

ID	Fam	Ism	Otak	Shif	Ushabdomoch	Tog_sana	Telefon
1	Makguyev	Oqbek	Oqil	Ushabdomoch	20.07.1987	1274455	
2	Salmov			Qadrovich	23.05.86	2315466	
3	Ramov	Korni		Ladovochi	24.08.1977	7778622	

4.30-rasm. MBdan barcha ma'lumotlarni qidirish holati

Dastur tahlili:

Dasturdan ko'rinish turibdiki, Connection sinf ob'ekti yordamida MB bilan bog'lanish amalga oshirildi. Jadvaldan hamma yozuvlarni

199

o'qish va qidirish uchun SQL so'rovlarini (Select, Like)dan foydalanildi. Dastur formasiga 3 ta CheckBox va TextBox komponentlari turli parametrlar asosida qidirishni tashkil etish maqsadida joylashtirildi. Dastur kodida DBSet() nomli funksiya yaratildi va uning yordamida MBdan parametrlari qidirish amalga oshirildi. TextBox komponentasining onchange xodisasisiga ma'lum shartlar (Fam, Ism, Tel bo'yicha) asosida DBSet() funksiyasi chaqirildi (if(checkbox1->Checked) DBSet("SELECT * FROM [talaba] WHERE (Fam LIKE '"+textBox1->Text+"%')");).

6-dastur: MS Access ning MBdagi jadval yozuvlarini yangilovchi vizual dastur

Ushbu dasturni tuzish uchun Visual C++ dasturida Windows Form Application multitida yangi loyiha yaratiladi. Formaga 2 ta Button va DataGridView komponentlari joylashtiriladi.

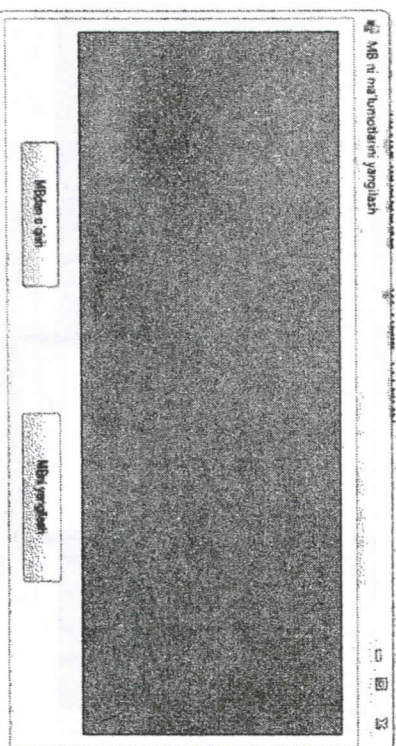
Dasturning kodlari quyidagicha:

```
1. #pragma endregion
2. DataSet ^ NaborDannix;
3. OleDb::OleDbDataAdapter ^ Adapter; OleDb::OleDbConnection ^
   Ulanish;
4. OleDb::OleDbCommand ^ Komanda;
5. private: System::Void Form1_Load(System::Object^ sender,
   System::EventArgs^ e){
6. NaborDannix = gcnew DataSet();
7. Ulanish = gcnew OleDb::OleDbConnection( "Data
   Source='C:\\Oybek.mdb\\';User =
   + "ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0';");
8. Komanda = gcnew OleDb::OleDbCommand();
9. button1->Text = "MBdan o'qish";button1->TabIndex = 0;
10. button2->Text = "MBni yangilash"; this->Text="MB ni
   ma'lumotlarini yangilash";}
11. private: System::Void button1_Click(System::Object^ sender,
   System::EventArgs^ e){
12. // MB dan o'qish:
13. if (Ulanish->State == ConnectionState::Closed) Ulanish->Open();
14. Adapter = gcnew OleDb::OleDbDataAdapter("Select * From
   [talaba]", Ulanish);
15. // DataSet ni SQL so'rovlarini natijasi bilan to'ldirish
16. Adapter->Fill(NaborDannix, "talaba");
17. String ^ Strokaxml = NaborDannix->GetXml();
18. dataGridView1->DataSource = NaborDannix;
19. dataGridView1->DataMember = "talaba";
20. Ulanish->Close();}
```

200

```
21. private: System::Void button2_Click(System::Object^ sender,
   System::EventArgs^ e) { // MB ni yangilash
22. Komanda->CommandText = "UPDATE [talaba] SET [Fam] = ?, [Ism] = ?,
   [Sharf] = ?, "+ "Tug_sana = ?, [Telefoni] = ? WHERE ([Id] = ?)";
23. // Ustun nomlari, tiplari, o'lchami, ustun nomlari
24. Komanda->Parameters->Add("Fam", OleDb::OleDbType::VarChar, 50,
   "Fam");
25. Komanda->Parameters->Add( "Ism", OleDb::OleDbType::VarChar, 50,
   "Ism");
26. Komanda->Parameters->Add( "Sharf", OleDb::OleDbType::VarChar,
   50, "Sharf");
27. Komanda->Parameters->Add( "Tug_sana", OleDb::OleDbType::Date ,
   50, "Tug_sana");
28. Komanda->Parameters->Add( "Telefoni", OleDb::OleDbType::VarChar
   , 50, "Telefoni"); Komanda->Parameters->Add(gcnew
29. OleDb::OleDbParameter("Original_id",
30. OleDb::OleDbType::Integer, 0, System::Data::ParameterDirection::
31. Input, false, (Byte)0, (Byte)0, "Id",
   System::Data::DataRowVersion::
32. Original, nullptr));
33. Adapter->UpdateCommand = Komanda; Komanda->Connection =
   Ulanish;
34. try { // Update o'zgargan qatorlar sonini qaytaradi:
35. int kol = Adapter->Update(NaborDannix, "talaba");
36. MessageBox::Show("Obnovleno " + kol + " zapisey"); }
37. catch (Exception ^ Situasija)
38. { MessageBox::Show(Situasiya->Message, "Nedorazumenie"); }
39. }
```

Dasturning tashqi ko'rinishi quyidagicha:



4.31-rasm. Dasturning boshlang'ich xolati

201

“MBdan o’qish” tugmasi bosilgandan dastur “Oybek.mdb” nomli MB bilan bog’lanadi va uning “talaba” nomli jadvalidagi barcha ma’lumotlarni ekran “dataGridView” komponentasi”ga chiqaradi va quyidagi ko’rinishga keladi:

ID	Fam	Ism	Shif	Tug_sana	Telefon
1	Madayev	Oyoq	Leonidovich	15.08.1985	1234455
2	Salimov	Ota	Odjonovich	20.07.1987	2345566
3	Gadayev	Sayid	Abduravich	24.08.1977	7778822

4.32-rasm. MB ga bog’lanib, yozuvlarini chiqargan xolati

Jadvaldagi ixtiyoriy qatorni o’zgartirib, “MBni yangilash” tugmasi bosilsa dastur “Oybek.mdb” nomli MB bilan bog’lanadi, uning “talaba” nomli jadvalidagi o’zgartirilgan qator ma’lumotlarini yangilaydi va quyidagi ko’rinishga keladi:

ID	Fam	Ism	Shif	Tug_sana	Telefon
1	Madayev	Oyoq	Leonidovich	15.08.1985	1234455
2	Salimov	Ota	Odjonovich	20.07.1987	2345566
3	Tobirov	Lotan	Haydarovich	14.03.1987	1112233

4.33-rasm. Jadvaldagi yozuvlarini yangilagan holati

Dastur tahlili:

MB dan ma’lumotlarni o’qish va yangilash dasturini tuzishda birinchi navbatda SQL so’rovi (Select * From[talaba])ni tanlandi. Hamda Adapter sinf ob’ekti yordamidan foydalanildi. Keyin “dataGridView1” komponentasining “DataSource” hossasiga olingan barcha yozuvlar ko’rsatildi. Natijada MBdan yozuvlarini o’qish va ularni o’zgartiruvchi vizual dastur yaratildi.

7-dastur: MS Access ning MB dagi jadval yozuvlarini SQL so’rovlarini va “Command” sinf ob’ekti yordamida o’chiruvchi vizual dastur

SQL so’rovlaridan maqsadli foydalanib, MB ning jadvalidagi yozuvlarini o’chirish mumkin. Buning uchun “Command” sinf ob’ekti kerak bo’ladi. Ushbu dasturni tuzish uchun “Visual Studio” dasturida “Windows Form Application” muhitida yangi loyiha yaratiladi. Formaga bitta “Button”, “TextBox” va “DataGridView” komponentalari joylashtiriladi.

Dasturning kodlari quyidagicha:

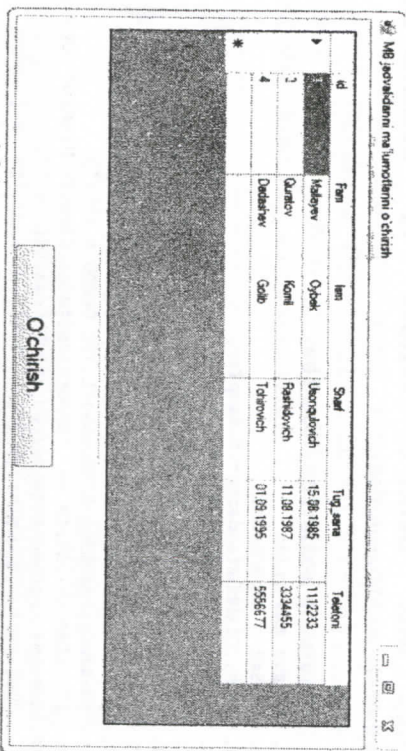
1. #pragma endregion
2. DataSet ^ NaborDannix;
3. OleDb::OleDbDataAdapter ^ Adapter;
4. OleDb::OleDbConnection ^ Ulanish;
5. OleDb::OleDbCommand ^ Komanda;
6. void MBdan_uchirish(){
7. Ulanish = gcnew OleDb::
8. OleDbConnection("Data Source='C:\\Oybek.mdb\\';User '+'
9. ID=Admin;Provider='Microsoft.Jet.OLEDB.4.0\\';");
10. Komanda = gcnew Data::OleDb::OleDbCommand(
11. "Delete Fam,Ism,Sharf,Tug_sana,Telefoni From [talaba] Where " +
12. "Fam = '"+textBox1->Text+"', Ulanish);
13. int i = Komanda->ExecuteNonQuery();
14. // i - kolichestvo udalennix zapisey
15. if (i > 0){ MessageBox::Show("Familiya ustunidagi bo'yicha
16. topilgan yozuvlar '"+textBox1->Text+"', o'chirildi");
17. MBdan_uqish(); }
18. if (i == 0) MessageBox::Show("Familiya ustunidagi bo'yicha
19. topilgan yozuvlar '"+textBox1->Text+"', topilmadi"); }
20. void MBdan_uqish(){


```

17. NaborDannix = gcnew DataSet();
18. Ulanish = gcnew OleDb::OleDbConnection( "Data
Source=\\C:\\Oybek.mdb\\;User " +
19. "ID=Admin;Provider=\\Microsoft.Jet.OLEDB.4.0\\;");
20. Komanda = gcnew OleDb::OleDbCommand();
21. // MB dan o'qish:
22. if (Ulanish->State == ConnectionState::Closed) Ulanish->Open();
23. Adapter = gcnew OleDb::OleDbDataAdapter("Select * From
[talaba]", Ulanish);
24. // DataSet ni SQL so'rovlarini natijasi bilan to'ldirish
25. Adapter->Fill(NaborDanno`x, "talaba");
26. String ^ StrokaxML = NaborDanno`x->GetXml();
27. dataGridView1->DataSource = NaborDanno`x;
28. dataGridView1->DataMember = "talaba"; }
29. private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e) {
30. this->Text="MB jadvalidanni
ma'lumotlarini o'chirish"; MBdan_uqish(); }
31. private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) { MBdan_uchirish(); }

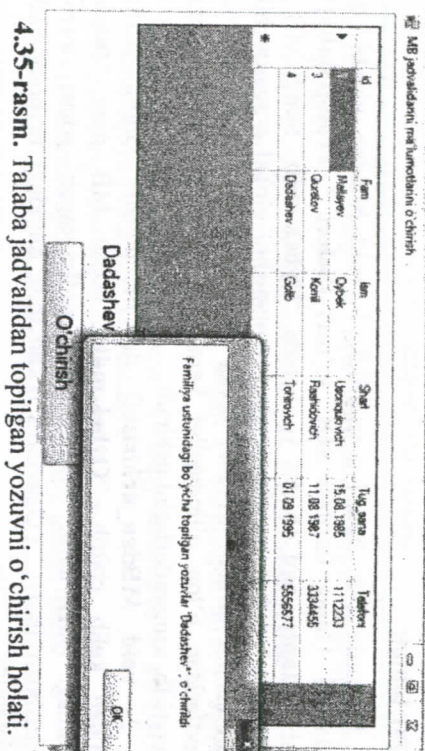
```

Dastur natijasi:



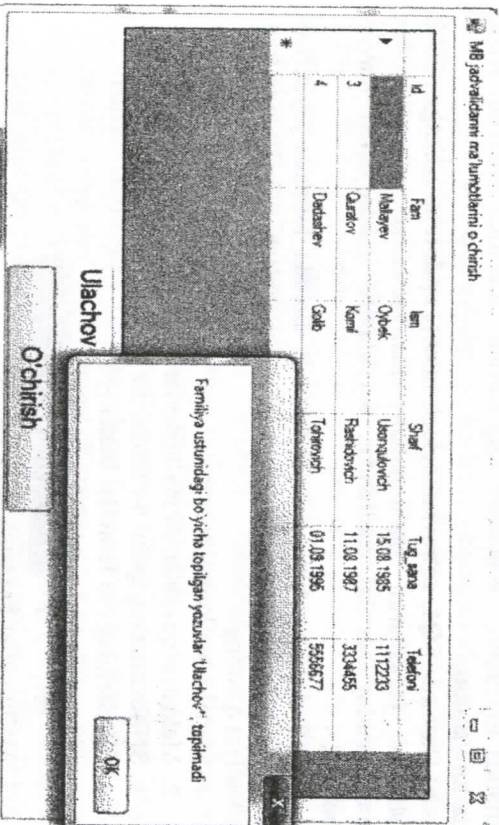
4.34-rasm. Dasturning boshlang'ich xolati.

"O'chirish" tugmasi tepasidagi bo'sh maydonchaga "Fam" ustunida mavjud yozuvlardan biri kiritiladi. "O'chirish" tugmasi bosilganda Agar kiritilgan yozuv "Fam" ustunining qaysidir qatorida mavjud bo'lsa, o'sha qator o'chiriladi.



4.35-rasm. Talaba jadvalidan topilgan yozuvni o'chirish holati.

Aks holda, "Bunday yozuv topilmadi" degan xabaroma chiqadi. Xabarnomadagi OK tugmasi bosilsa, dastur yana Oybek.mdb nomli MB bilan bog'lanib, uning "talaba" nomli jadvalidagi barcha ma'lumotlarni ekran (dataGridView komponentasi)ga chiqaradi va quyida ko'rinishga keladi:



4.36-rasm. MB dan yozuvlar topilmagan holat.

Dastur tahlili:

Dasturda 2 ta funksiya yaratildi: 1- void **MBdan_uchish()** deb nomlanadi. Uning vazifasi **SQL** so'rovi (**Select * From [talaba]**)ni tanlab, **Adapter** sinf ob'ekti yordamida "**Oybek.mdb**" nomi MB ga ulanishni ta'minlashdan iborat. Bunda ulanishni global o'zgaruvchilar orqali tashkili qilindi. Chunki ikkita ta funksiya uchun alohida sinf ob'ektlardan e'lon qilinishi kerak edi. Bu esa dastur hajmini kamaytirib, samaradorligini oshiradi.

2- void **MBdan_uchirish()** deb nomlanadi. Uning vazifasi nomidan kelib chiqib, "**Oybek.mdb**" nomi MB ning "**talaba**" jadvaldagi yozuvlarni, foydalanuvchi qidirayotgan yozuv asosida o'chirishdan iborat. Yozuvlarni o'chirishda quyidagi texnologiyadan foydalanildi:

```
Komanda = genew Data::OleDb::OleDbCommand("Delete
Fam,Ism,Sharf,Tug_sana,Telefoni From [talaba] Where " + "Fam =
"++textBox1->Text+""", Ulanish);
```

Mustahkamlash uchun mashqlar

a) Quyidagi topshiriqlar asosida 4, 5, 6 va 7- dasturlardan na'muna sifatida foydalanib, "**Windows Application**" muhitida ishlovchi vizual dastur yarating;

b) Dasturda **SQL** so'rovlar(**select, insert, update**)ni ishlatgan holda MB ni jadvallarini ma'lumotlar bilan to'ldirish, yangilash va o'chirishning har hil usullaridan foydalanish imkoniyatlari mavjud bo'lsin;

s) MB dagi har bita jadval, dasturda alohida formda joylashishi lozim va ularning tashqi ko'rinishlariga va ishlash qulayligiga katta ahamiyat qaratilgan.

1. Mahalla ma'lumotlar bazasi;
2. Maktab o'quvchilarini ma'lumotlar bazasi;
3. SHifoxonaning qabul bo'limini ma'lumotlar bazasi;
4. Fakultetning ma'lumotlar bazasi;

Nazorat savollari:

1. Ma'lumotlar bazasiga ta'rif bering?
2. Ko'pchilik foydalanuvchilar tomonidan MBni yaratish, to'ldirish va birgalikda foydalanish uchun mo'ljallangan dasturiy vositalar majmuasi nima deyiladi?

3. Fayllarni boshqarishda nima yordam beradi?

4. MB administratori.

5. Zamonaviy MBBTlar fayl tizimining qaysi muammosini hal qiladi?

6. MBBT dagi foydalanuvchilarga yaqin abstraksiya bosqichini ko'rsating.

7. MBBT arxitekturasining bosqichi bo'lmagan javob variantini ko'rsating.

8. MBBT dagi MBning barcha mantiqiy strukturasi ko'rsatuvchi abstraksiya bosqichini ko'rsating.

9. MBBTning vazifasiga nimalar kirmaydi?

10. Ma'lumotlar bazasi tizimi nimalarni o'z ichiga oladi?

11. MBBT dagi axborotni jismonan saqlashga yaqin abstraksiya bosqichini ko'rsating.

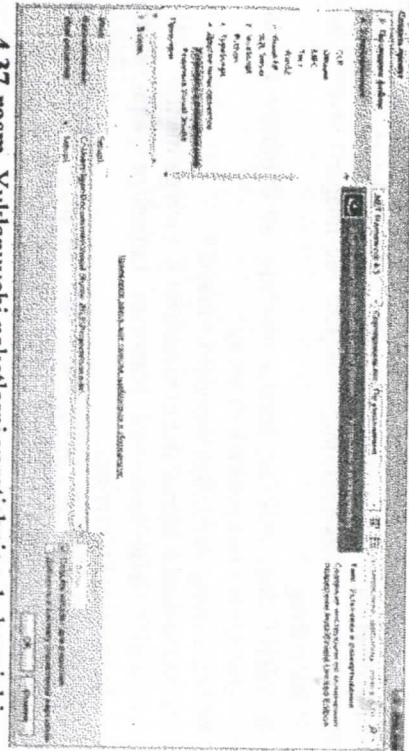
12. Ma'lum bir sust strukturga ega va qiyin shakllanuvchi fan sohasining sun'iy intellekt tizimi nima deb ataladi?

4.5. Loyihalarning yuklanuvchi "инсталляционные" dasturlar paketini yaratish

Bizga ma'lumki, avvalgi bo'limlarda yaratilgan kichik loyihalarni boshqa kompyuterlarda sinash yoki amaliyotga tadbiq qilish uchun ularning yuklanuvchi paketlarini yaratish kerak. Yuklanuvchi paketlar - bu loyihaning dastur kodlari yaratuvchisida saqlangan holda ularning "exe" formati (tizimga yuklanuvchi)dir. Yuklanuvchi paket o'zida bir nechta (**setup.exe, *.msi, *.dll, *.tlb**)kengaytmali fayllarni saqlaydi. Kichik loyihalarning operasion tizimga yuklanuvchi paketlarini yaratish uchun dasturlarni instal'atsiyalovchi maxsus dasturlar bor. Masalan **InstallShield, InnoSetup** va boshqalar. Lekin bunday dasturlarni internet orqali yuklab olishni va aktivlashtirishni talab qiladi.

Yuklanuvchi paketlarni yaratish uchun **Visual Studio** dasturi ishga tushiriladi. Yangi loyiha yaratiladi(**New Project**), **Other Project Types** bo'limidan **Setup and Deployment (Установка и распространение)** qismiga o'tiladi va **Visual Studio Installer va Setup Project** tanlanadi. **Location** qismiga yuklanuvchi paket saqlanishi kerak bo'lgan diskdan joy ko'rsatiladi va OK tugmachasi bosiladi. Masalan: **S:\New_instal**. Misol uchun, bir loyihani yuklanuvchi paketini yaratib ko'raylik.

1-qadam. Veb brouzerda quyidagi sahifa ochiladi. Bu sahifadan “**Har**
2” havolasi bosiladi.



4.37-rasm. Yuklanuvchi paketlarni yaratishning 1 - bosqichi

1-800-4-A-SECURE
InstallShield

InstallShield Limited Edition AA Visual Studio

- **Algorithmen** = Verfahren zur Lösung eines Problems (z.B. Berechnung der Summe aller Zahlen in einer Liste)
- **Strukturierte Programmierung** = Programmierung, die in eine Reihe von Schritten unterteilt ist, die in einer bestimmten Reihenfolge ausgeführt werden
- **Strukturierte Programmierung** = Programmierung, die in eine Reihe von Schritten unterteilt ist, die in einer bestimmten Reihenfolge ausgeführt werden
- **Strukturierte Programmierung** = Programmierung, die in eine Reihe von Schritten unterteilt ist, die in einer bestimmten Reihenfolge ausgeführt werden

Шар 2 силкаси

2-qadam.

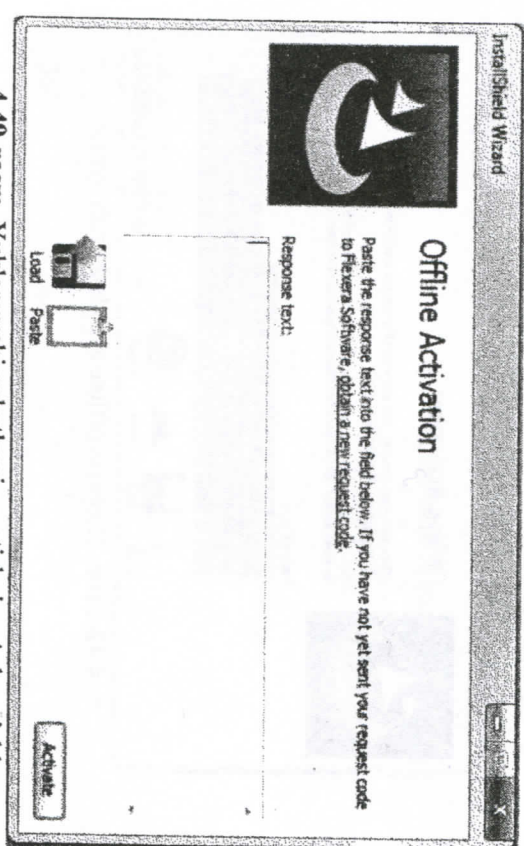
Undan keyin anketa to'ldiriladi.

[illegible]

4.39-rasm. Yuklanuvchi paketlarni yaratishning 3- bosqichi

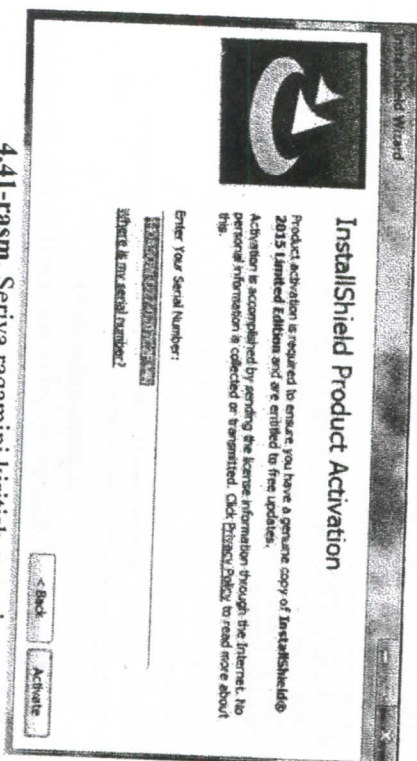
3-qadam.

Anketa to'ldirigandan keyin "**Download Now**" tugmasini bosib, dasturni yuklab olasiz va tizimga yuklaysiz. Yaratilgan tayyor loyiha dasturini ochasiz. Dastur ochilishida yangi yuklangan dastur lisenziya kaltitini talab qiliadi:



4.40-rasm. Yuklanuvchi paketlarni yaratishning 4- bosqichi

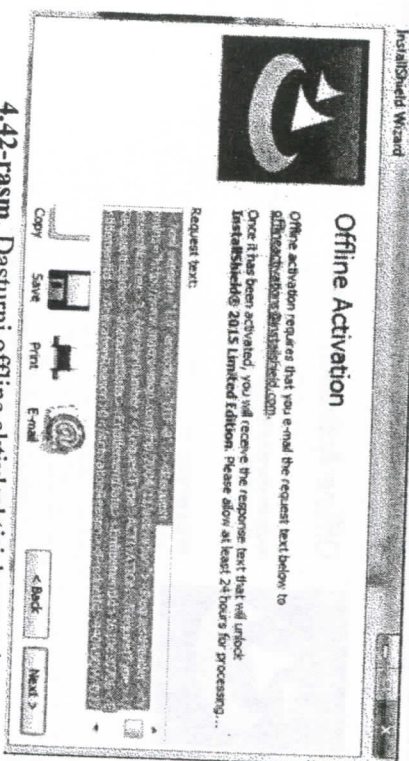
4-qadam.



4.41-rasm. Seriya raqamini kiritish oynasi

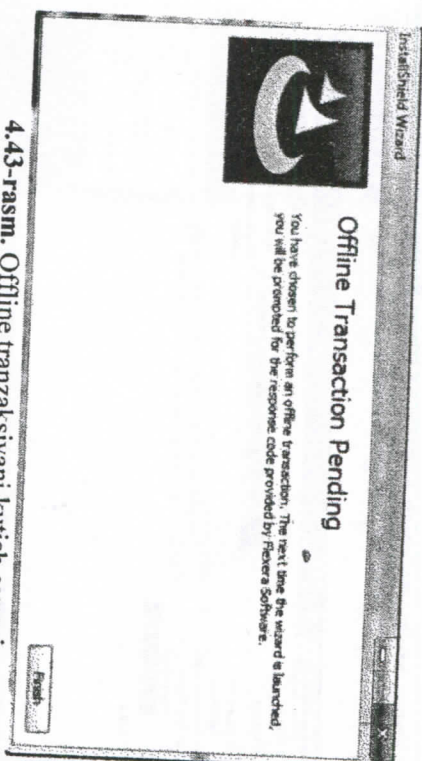
5-qadam.

Lisenzia kaitini anketangizda ko'rsatilgan pochta manziliidan olinadi, so'ralgan joyga yoziladi va "Activate" tugmachasi bosiladi.



4.42-rasm. Dasturni offline aktivlashirish oynasi

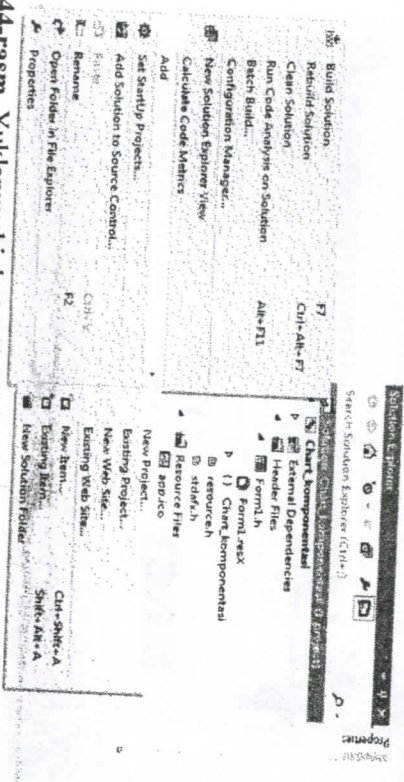
6-qadam.



4.43-rasm. Offline tranzaksiyani kutish oynasi

7-qadam.

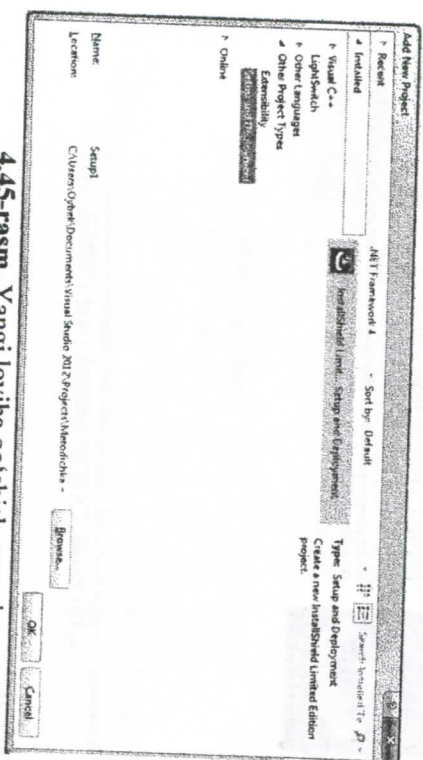
Finish tugmasi bosilgandan keyin. Yuklanuvchi paketini yaratish kerak bo'lgan dasturga yangi loyiha qo'shildi: (Solution Explorer-> loyiha tanlanadi va sichqonchani o'ng tugmasini bosib, kontekst menyudan->Add->New Project..)



4.44-rasm. Yuklanuvchi dasturga qo'shiladigan yangi loyihani tanlash oynasi

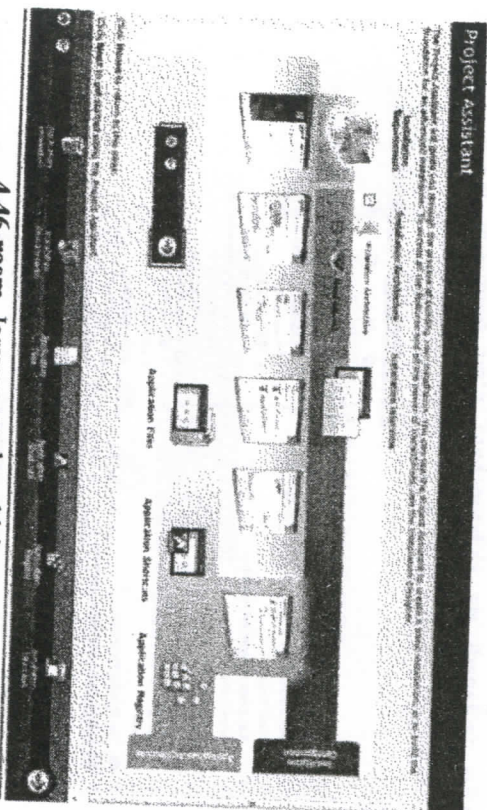
8-qadam.

Yangi loyihaga qo'shish oynasida **Other Project Types** bo'limidan **Setup and Deployment** qismiga o'tiladi va OK tugmasi bosiladi:



4.45-rasm. Yangi loyihaga qo'shish oynasi

9-qadam. Ushbu jarayondan keyin quyidagi jarayon bajariladi:

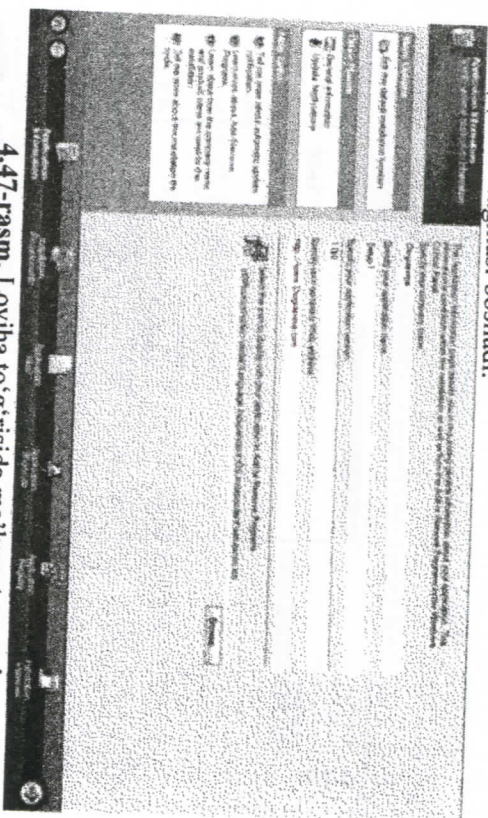


4.46-rasm. Jarayon yordamchisi oynasi

212

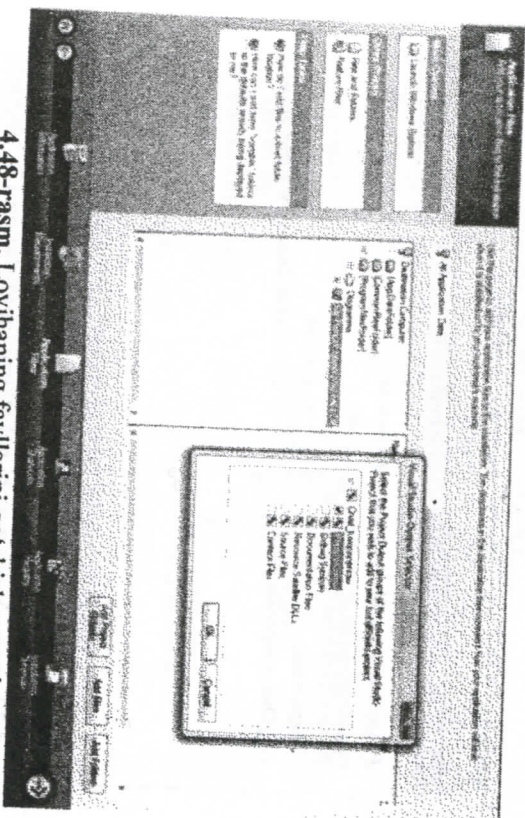
10-qadam.

"**Daree**" tugmasi bosiladi.



4.47-rasm. Loyihaga to'g'risida ma'lumot oynasi

11-qadam. "**Daree**" tugmasi bosiladi.

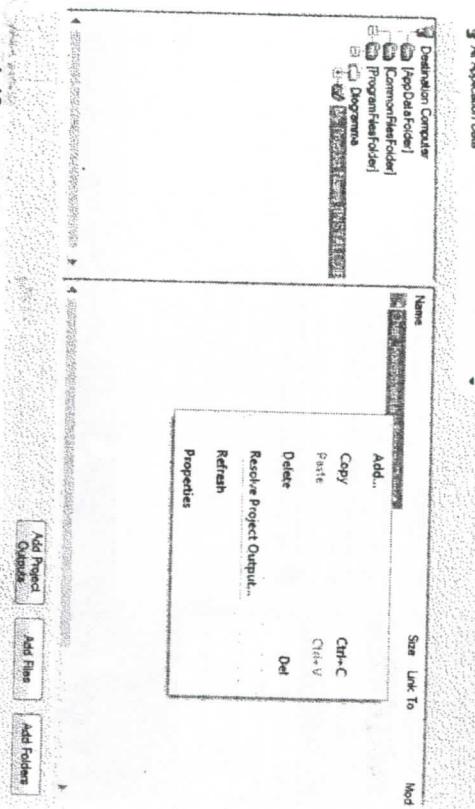


4.48-rasm. Loyihaning fayllarini qo'shish oynasi

213

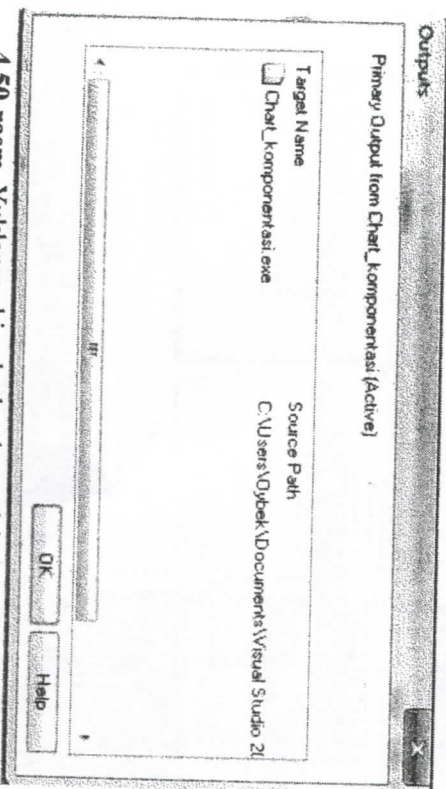
12-qadam.

“OK” va “Daree” tugmasi bosiladi.



4.49-rasm. Yuklanuvchi paketlarni yaratishning 12- bosqichi

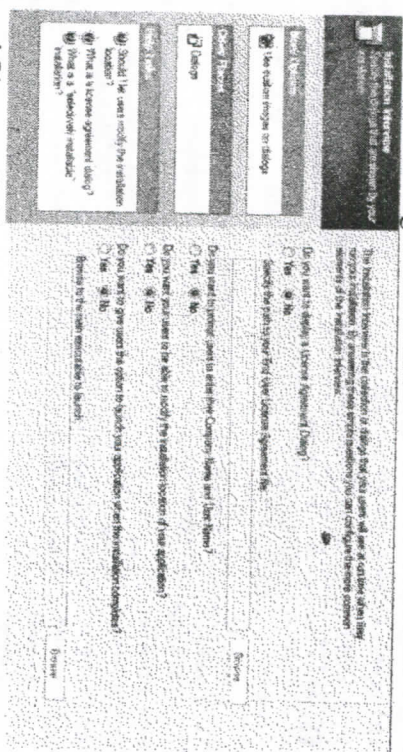
13-qadam. “OK” va “Daree” tugmasi bosiladi.



4.50-rasm. Yuklanuvchi paketlarni yaratishning 13- bosqichi

14-qadam.

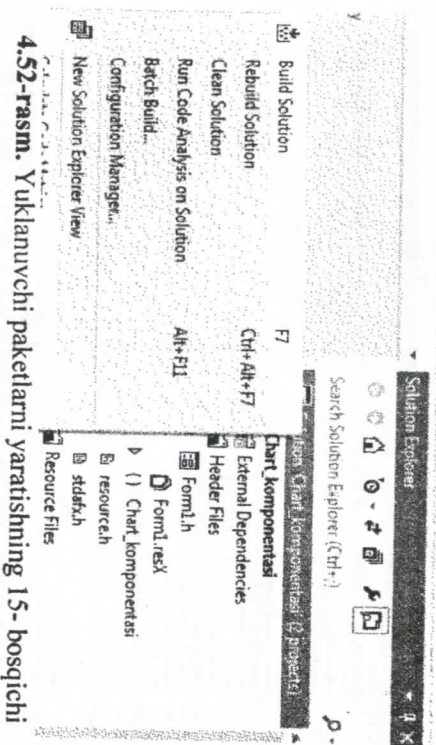
“OK” va “Daree” tugmasi bosiladi.



4.51-rasm. Yuklanuvchi paketlarni yaratishning 14- bosqichi

15-qadam.

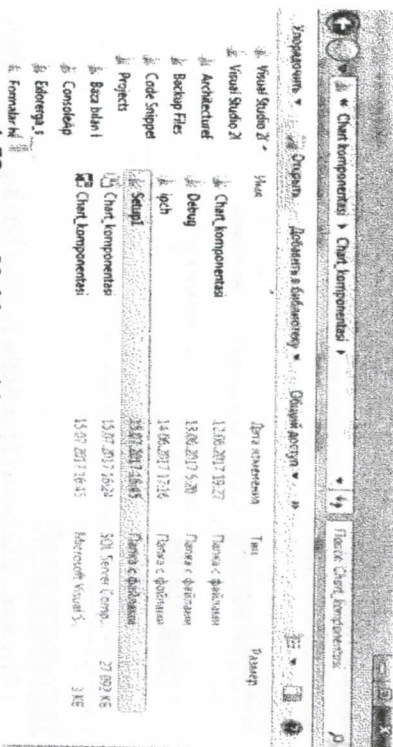
Kontekst menyudan “Build Solution” tanlanadi.



4.52-rasm. Yuklanuvchi paketlarni yaratishning 15- bosqichi

16-qadam.

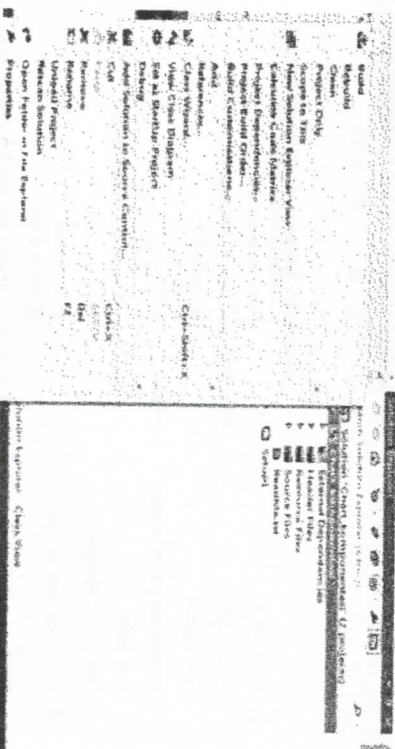
Bu jarayondan so'ng yuklanuvchi paket saqlangan katalog ichida setup1 nomi katalog paydo bo'ladi.



4.53-rasm. Yuklanuvchi paket saqlangan joy

17-qadam.

Ushbu katalogga quyidagicha o'tsa ham bo'ladi:



4.54-rasm. Yuklanuvchi paket saqlangan joyga o'tish

18-qadam.

Setup1\Express\DVD-5\DiskImages\DISK1 ushbu manzilda yuklanuvchi fayl joylashgan. Natijada ushbu loyihaning yuklanuvchi paketini tarqatish imkoniyati vujudga keldi.

Bob xulosasi

Ushbu bobda MS Access 2010 dasturida talabalarni baholash reytinglari haqidagi ma'lumotlar bazasini loyihalashtirish uchun jadvallar, so'rovlar, formalar, hisobotlar va makrosalar yaratish usullari keltirilgan. SQL(Structured query language) so'rovlarini ishlatish qoidalari yuzasidan ma'lumotlar keltirildi hamda ushbu ma'lumotlar asosida amaliy mashg'ulotlar tashkil qilindi va amaliy dasturlar tuzildi.

Nazoriy savollar va amaliy topshiriqlar

1. MS Access dasturida kitoblarning avtomatlashgan reytingini formallashtirish jadvalini shlab chiqish.
2. MS Access dasturida kitoblarning avtomatlashgan reytingini ma'lumotlar bazasini ishlab chiqish. MB da jadvallar, jadvallarning bog'lanishini, so'rovlarni va makroslarni tashkil etish.
3. SQL dasturida SQL so'rovlarini yordamida jadvalarni, jadvallar asosida ma'lumotlarni izlash usullarini ishlab chiqish.
4. MS Access dasturida SQL so'rovlarini yordamida kitoblarni qidirish usullari ishlab chiqish.
5. Visual Studio/Embarcadero dasturida komponentalar, formalar, kutubxonalarini tashkil etish. Boshqa dasturlash tilarida tuzilgan funksiyalardan foydalanish.
6. Visual Studio/Embarcadero dasturida yordamchi maxsus dasturlar yordamida loyihani interfeysini ishlab chiqish.
7. Dasturiy loyihani uchun ishlab chiqilgan algoritim funksiyalarini ob'ektlarga yo'naltirilgan dasturlash shartlari asosida tashkil etish.

8. Ma'lumotlarning universal strukturasi va klasslarini tashkil etish. Ular asosida DLL kutubxona fayllarini tashkil etish.
9. Visual C++ da qanday dastur yuklanuvchi dastur paketini yaratadi?
10. Visual C++ da yuklanuvchi dastur paketini yaratishning qanday turlarini bilasiz?
11. Visual C++ da yaratilgan loyihani boshqa kompyuterlarda ishlatish usullarini bilasizmi?
12. .net framework nima?
13. Visual C++ da .net framework nimaga kerak?
14. Mayjud loyihalarni birlashtirish qanday amalga oshiriladi?
15. Loyihaga serverni ulash qanday amalga oshiriladi?
16. Loyihaga MB ni ulash qanday amalga oshiriladi?
17. Tizimga yuklangan yangi loyiha dasturini sozlash mumkinmi?
18. Visual C++ da loyiha dasturining xavfsizligi qanday taminlanadi?

ADABIYOTLAR

1. O'zbekiston Respublikasini yanada rivojlantirish bo'yicha harakatlar strategiyasi to'g'risida. O'zbekiston Respublikasi Prezidentining PF-4947- son farmoni. Toshkent, 2017 yil 7 fevral.
2. Mirziyoev SH.M. Tanqidiy tahlil, g'atiy tartib-ittizom va shaxsiy javobgarlik - har bir rahbar faoliyatining kundalik qoidasi bo'lishi kerak. Mamlakatimizni 2016 yilda ijtimoiy-iqtisodiy rivojlantirishning asosiy yakunlari va 2017 yilga mo'ljallangan iqtisodiy dasturining eng muhim ustuvor yo'nalishlariga bag'ishlangan Vazirlar Mahkamasining kengaytirilgan majlisidagi ma'ruza, 2017 yil 14 yanvar. - Toshkent: «O'zbekiston», 2017.-104 b.
3. Mirziyoev SH.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. 2017.
4. Mirziyoev SH.M. Qonun ustuvorligi va inson manfaatlarini ta'minlash – yurt taraqqiyoti va xalq farovonligining garovi. 2017.
5. Nazirov SH.A., Qobulov R.V., Bobojanov M.R., Rahmatov Q.S. S va C++ tili. "Voriz-nashriyot" MCHJ, Toshkent 2013, 488 b.
6. Maxarov Q.T., Maxarov T.A. Visual Studio multitida dasturlash asoslari(ustubiy qo'llanma). Toshkent – 2017.
7. Maxarov Q.T., Nigmatova D.B., Navruzov E.R., Nayitqulov B.H. VC++ multitida obyektlar bilan ishlash. Toshkent – 2017
8. Horstmann, Cay S. C++ for everyone/Cay S. Horstmann. Printed in the United States of America - 2nd ed. 2010. – P. 562.
9. Zubarov V.V. MS Visual C++ 2010 v srede .NET (2012) Piter.
10. Boris Raхomov. C/C++ i Visual C++ 2010 dlya nachinayuchi. - SBP: BXV-Peterburg. 2011.
11. J.Aхmadaliev, R.Xoldorboev C++ dasturlash tilini o'rganish bo'yicha ustubiy qo'llanma(2015).
12. Norton I.-Beginning Visual C++ 2012/ I.Norton. Published simultaneously in Canada.-2012.-P. 988.
13. Давыдов, В. Visual C++. Разработка Windows-приложений с помощью МФС и API-функций / В. Давыдов. - М.: БХВ-Петербург, 2014. - 576 с.
14. Довбуш, Галина Visual C++ на примерах / Галина Довбуш, Анатолий Хомоненко. - М.: БХВ-Петербург, 2012. - 528 с.

15. Эйборов, В. MS Visual C++ 2010 в среде .NET / В. Эйборов. - М.: Питер, 2012. - 320 с.
16. Мешков, А. Visual C++ и МФС / А. Мешков, Ю. Тихомиров. - М.: БХВ-Петербург, 2013. - 546 с.
17. Панкова, Т. А. Языки и методы программирования. Создание простых GUI-приложений с помощью Visual C++. Учебное пособие / Т. А. Панкова, А. В. Панюков. - Москва: Мир, 2015. - 144 с.
18. Пахомов, Б. C/C++ и MS Visual C++ 2010 для начинающих / Б. Пахомов. - М.: БХВ-Петербург, 2011. - 736 с.
19. Пахомов, Борис C/C++ и MS Visual C++ 2012 для начинающих / Борис Пахомов. - Москва: СИНТЕГ, 2015. - 518 с.
20. Пахомов, Борис C/C++ и MS Visual C++ 2012 для начинающих / Борис Пахомов. - М.: "БХВ-Петербург", 2013. - 502 с.
21. Понамарев, В. Программирование на C++/C# в Visual Studio .NET 2003 / В. Понамарев. - М.: БХВ-Петербург, 2015. - 917 с.
22. Сидорина, Татьяна Самоучитель Microsoft Visual Studio C++ и МФС / Татьяна Сидорина. - М.: "БХВ-Петербург", 2014. - 848 с.
23. <http://labs.org.ru/visual-c/>
24. <https://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>
25. <https://professorweb.ru/my/programs/visual-studio/level1/>

MUNDARIJA

KIRISH	3
1- BOB. Visual Studio 2012 dasturida Visual C++ muhitining loyihla platformalarini tanlash usullari.....	6
1.1. Dasturlash nima? Visual Studio 2012 dasturini tuzilishi..	7
1.2. Texnika kodi va dasturlash tillari	15
1.3. Standartlovchi tashkilotlar	21
1.4. Dasturlash muhiti bilan tanishish	22
1.5. Zahira ko'nikmalari	23
1.6. Xatolar	26
1.7. Odatiy xato	27
1.8. Muammo yechimi: algoritim konstruksiyasi	28
Bob xulosasi	32
Nazariy savollar va amaliy topshiriqlar	32
2- BOB. VC++ ning Console Application muhiti va unda ishlash	34
2.1. Visual Studio 2012 dasturini tizimga o'rnatish.....	34
2.2. Console Application muhitida konsol ilovalar yaratish	39
2.3. Visual C++ muhitida strukturalar bilan ishlash	51
2.4. Visual C++ muhitida sinflar va ob'ektlar yaratish.....	56
2.5. Visual C++ muhitida konstruktor hamda destruktork yaratish.....	58
2.6. Visual C++ muhitida inkapsulyatsiyani qo'llash	59
2.7. Visual C++ muhitida polimorfizmni qo'llash	64
2.8. Visual C++ muhitida sinflar orasidagi munosabatni va merosxo'rlikni qo'llash	70
2.9. Visual C++ muhitida standart qo'llar kutubxonasi bilan ishlash	74
Bob xulosasi	74
Nazariy savollar va amaliy topshiriqlar	75
3- BOB. Visual C++ ning Windows Form Application muhiti va unda ishlash	77
3.1. Windows Form Application muhiti yaratish va yordamchi oynalarini sozlash	78
3.2. Form, Button, Label komponentlari va MessageBox xabarlar oynasi.....	79
3.3. MouseHower hodisasi	85

3.4. TextBox va DatePicker komponentalari.....	86
3.5. Forma (Form) ni formalarga bog'lash usullari	91
3.6. CheckBox, CheckedListBox, ComboBox, ListBox komponenta-lari va ularning xossalari	94
3.7. TabControl va RadioButton komponentalari.....	100
3.8. Berilganlarini lug'at (Dictionary) yordamida strukturali saqlash	102
3.9. Bir prosedura orqali bir nechta hodisalarga ishlov berish.....	104
3.10. Turli tipdagi fayllarning manbalariga LinkLabel komponentasi yordamida murojaatlar.....	105
3.11. Klaviatura hodisalarini qayta ishlash.....	106
3.12. Matnli maydonga kiruvchi ma'lumotlarni basqarish.....	109
3.13. try ...catch istisnosini qayta ishlash orqali matnli faylni o'qish va matnli faylga yozish	110
3.14. OpenFileDialog va SaveFileDialog komponentalaridan foydalanib fayllarni ochish va saqlash	112
3.15. Matnli xujjatni chop qilish	118
3.16. Formaga grafik fayldagi tasviri chiqarish	121
3.17. Formada grafik shakllarni va funkciya grafiklarini chizish.....	121
3.18. Formada sichqoncha ko'rsatgichi orqali chizish.....	124
3.19. Jadvalli ma'lumotlar asosida Chart komponentasi yordamida grafik diagrammalar yaratish	125
3.20. Veb browserda HTML jadvallarni tasvirlash va shakllantirish.....	131
3.21. Visual C++da MS Word imkoniyatlaridan foydalanib, jadvallar yaratish, ularni Word faylga eksport qilish va taqdim etish	133
3.22. Visual C++ da MS Excell imkoniyatlaridan foydalanib, diagrammalar yaratish va ularni turli kengaytmalarda saqlash	136
3.23. Visual C++ ning Windows Application muhitida komponentalarning joylashish vaziyatlarini nazorat qilish	142
Bob xulosasi	147
Nazariy savollar va amaliy topshiriqlar	147
4-BOB. Ma'lumotlar bazasini yaratish va ularga ishlov berish usullari.....	148

4.1. Ma'lumotlar bazasini boshqarish tizimlari. Microsoft Access da ma'lumotlar bazasini yaratish	149
4.2. SQL(structured query language) tili so'rovlari. SQL(Structured Query Language) haqida	162
4.3. Ma'lumotlar bazasini yaratish bo'yicha uslubiy ko'rsatmalar	188
4.4. Visual C++ da MS Access ning ma'lumotlar bazasini SQL so'rovlari asosida tahrirlash usullari.....	191
4.5. Loyihalarining yuklanuvchi "инсталляционные" dasturlar paketini yaratish	207
Bob xulosasi	217
Nazariy savollar va amaliy topshiriqlar	217
Adabiyotlar ro'yhati	219

M.YU.XAYDAROVA,
N.M.QURBONOV, O.U.MALLAYEV

VISUAL C++ DA KICHIK LOYIHALAR YARATISH

(O'quv qo'llanma).

Toshkent – «Aloqachi» – 2019

Muharrir:	Q.Matqurbonov
Tex. muharrir:	A.Tog'ayev
Musavvir:	B.Esanov
Musahhiha:	F.Tog'ayeva
Kompyuterda sahifalovchi:	Sh.To'xtamurodov

Nashr. lits. AI №176. 11.06.11.
Bosishga ruxsat etildi: 28.06.2019. Bichimi 60x841/16.
Shartli bosma tabog'i 14,5. Nashr bosma tabog'i 14,0.
Adadi 100. Buyurtma № .

«Nihol print» Ok da chop etildi.
Toshkent sh., M. Ashrafiy ko'chasi, 99/101.