Національний Авіаційний Університет

Факультет комп’ютерних систем

Кафедра комп’ютерних систем та мереж

**Домашня робота**

**З дисципліни: Інформаційні системи та структури даних**

Київ 2010

**Інформаційна система будівельної організації**

Будівельна організація займається будівництвом різного роду об'єктів: житлових будинків, лікарень, шкіл, мостів, доріг і т.д. по договорах із замовниками (міська адміністрація, відомства, приватні фірми й т.д.). Кожна з перерахованих категорій об'єктів має характеристики, властиві тільки цій або декільком категоріям: наприклад, до характеристик житлових будинків відноситься поверховість, тип будівельного матеріалу, число квартир, для мостів унікальними характеристиками є тип пролітної будови, ширина, кількість смуг для руху.

Структурно-будівельна організація складається з будівельних правлінь, кожне будівельне правління веде роботи на одному або декількох ділянках, очолюваних начальниками ділянок, яким підпорядковується група виконробів, майстрів і техніків. Кожній категорії інженерно-технічного персоналу (інженери, технологи, техніки) і робітників (муляри, бетонники, обробники, зварники, електрики, шофери, слюсарі і ін.) також є властиві характерні тільки для цієї групи атрибути. Робітники поєднується в бригади, якими керують бригадири. Бригадири вибираються із числа робітників, майстри, виконроби, начальники ділянок і правлінь призначаються із числа інженерно-технічного персоналу.

На кожній ділянці зводиться один або кілька об'єктів, па кожному об'єкті роботу ведуть одна або кілька бригад. Закінчивши роботу, бригада переходить до Іншого об’єкту на цій, або іншій ділянці. Будівельному правлінню надається будівельна техніка (підйомні крани, екскаватори, бульдозери й т.д.), що розподіляються по об'єктах.

Технологія будівництва того або Іншого об'єкта припускає виконання певного набору видів робіт, необхідних для спорудження даного типу об'єкта. Наприклад, для житлового будинку –це зведення фундаменту, цегельні роботи, прокладання водопостачання й т.д. Кожен вид робіт на об'єкті виконується однією бригадою. Дляорганізації робіт на об'єкті складаються графіки робіт, що вказують у якому порядку й у які строки виконуються ті або інші роботи, а також кошторис, що визначає які будівельні матеріали й у яких кількостях необхідні для спорудження об'єкта. За результатами виконання робіт складається звіт із зазначенням строків виконання робіт і фактичних витрат матеріалів.

**Види запитів в інформаційній системі:**

1. Одержати перелік будівельних правлінь й/або ділянок та їх керівників.

2. Одержати список фахівців інженерно-технічного складу позначеної ділянки або будівельного правління із зазначенням їх посад.

3. Одержати перелік об'єктів, що зводяться зазначеним будівельним правлінням й/або ділянкою, і графіки їхнього зведення.

4. Одержати склад бригад, що працювали (працюють) на будівництві зазначеного об'єкта.

5. Одержати перелік будівельної техніки, наданої зазначеному будівельному правлінню.

6. Одержати перелік будівельної техніки, виділеної на зазначений об’єкт або тієї, що працювала там упродовж зазначеного періоду часу.

7. Одержати графік і кошторис на будівництво зазначеного об'єкта.

8. Одержати звіт про спорудження зазначеного об’єкта.

9. Одержати перелік об’єктів, що зводяться у деякому будівельному правлінні або в цілому по організації, та на яких у позначений період часу виконувався зазначений вид будівельних робіт.

10. Одержати перелік видів будівельних робіт, по яких мало місце перевищення строків виконання на зазначеній ділянці, будівельному правлінні або в цілому по організації.

11. Одержати перелік будівельних матеріалів, по яких мало місце перевищення кошторису на зазначеній ділянці, будівельному правлінні або в цілому по організації.

12. Одержати перелік видів будівельних робіт, виконаних зазначеною бригадою протягом позначеного періоду часу із вказівкою об'єктів, де ці роботи виконувалися.

13. Одержати перелік бригад, що виконували зазначений вид будівельних робіт протягом позначеного періоду часу із зазначенням об'єктів, де ці роботи виконувалися.

**Концептуальна модель бази даних**

База даних будівельної організації складається з таблиць:

**tblOrder** – таблиця що містить інформацію про закази поля:

Order\_id – ідентифікатор

Object – рід об’єкту

Organisetion – замовники

BuildManagement – будівельне управління

**tblChaBuild** – характеристика заказу на житловий будинок, лікарню та школу поля:

ChaBuild\_id – ідентифікатор

Floor – кількість поверхів

MaterialType – тип матеріалу

NApartament – число квартир

**tblChaRoad** – характеристика заказу на мости та дороги поля:

ChaRoad\_id – ідентифікатор

StructureType – тип пролітної будови

Width – ширина

Bands – кількість смуг

**tblGrafics** – графік зведення об’єкта поля:

Grafics\_id – ідентифікатор

JobType – вид роботи

TermPerformance – строк завершення

Kyshty – виділенні кошти

**tblMaterials** – матеріали поля:

Materials\_id – ідентифікатор

Materials – назва необхідного матеріала

**tblReport** – звіт виконання роботи поля:

Report\_id – ідентифікатор

JobType – вид роботи

Completion – реальне завершення роботи

Used – використанні кошти

**tblBrigade** – бригади поля:

Brigade\_id – ідентифікатор

Overman – бригадир

**tblNPlot** – номер ділянки поля:

NPlot\_id – ідентифікатор

Head – начальник ділянки

**tblTech** – інженерно-технічний персонал поля:

Tech\_id – ідентифікатор

Engineers – інженери

Technology – технологи

Technics – техніки

**tblMachines** – будівельна техніка поля:

Machines\_id – ідентифікатор

Bulldozers – бульдозери

Cranes – крани

Excavator – екскаватори

**tblWorkers** – робітники поля:

Workers\_id – ідентифікатор

Bricklayer – муляри

Concrete – бетонники

Handler – обробники

Welders – зварники

Electricity – електрики

Driver – водії

Locksmiths – слюсарі

**Фізична модель бази даних**

Для побудови фізичної моделі бази даних потрібно вказати типи даних полів таблиць, визначити первинні та зовнішні ключі, організувати відношення між таблицями. Відношення типу „один до одного” організується додаванням первинного ключа однієї із таблиць в іншу у якості зовнішнього. Відношення типу „один до багатьох” організується додаванням первинного ключа таблиці зі сторони „один” в таблицю „багато” в якості зовнішнього ключа. Відношення типу „багато до багатьох” організується через таблиці-связки, в які поміщаються первинні ключі двох таблиць в якості зовнішніх ключів.

**Створення таблиць та звя’зків**

Створення таблиць:

create table **rl\_10** ( NPlot\_id int not null, Machines\_id int not null, constraint PK\_RL\_10 primary key (NPlot\_id, Machines\_id)

)

go

create table **rl\_11** ( NPlot\_id int not null, Workers\_id int not null, constraint PK\_RL\_11 primary key (NPlot\_id, Workers\_id)

)

go

create table **rl\_9** ( NPlot\_id int not null, Tech\_id int not null, constraint PK\_RL\_9 primary key (NPlot\_id, Tech\_id)

)

go

create table **tblBrigade** ( Brigade\_id int not null, NPlot\_id int null, Grafics\_id int null, Overman varchar(255) not null, constraint PK\_TBLBRIGADE primary key nonclustered (Brigade\_id)

)

go

create table **tblChaBuild** ( ChaBuild\_id int not null, Order\_id int null, Floor int not null, MaterialType varchar(255) not null, NApartament int not null, constraint PK\_TBLCHABUILD primary key nonclustered (ChaBuild\_id)

)

go create table **tblChaRoad** ( ChaRoad\_id int not null, Order\_id int null, Structuretype varchar(255) not null, Width int not null, Bands int not null, constraint PK\_TBLCHAROAD primary key nonclustered (ChaRoad\_id)

)

go

6

create table **tblGrafics** ( Grafics\_id int not null, Order\_id int null, Brigade\_id int null, JobType varchar(255) not null, TermPerformance datetime not null, Kyshty money not null, constraint PK\_TBLGRAFICS primary key nonclustered (Grafics\_id)

)

go

create table **tblMachines** ( Machines\_id int not null, Bulldozers varchar(255) null, Cranes varchar(255) null, Excavator varchar(255) null, constraint PK\_TBLMACHINES primary key nonclustered (Machines\_id)

)

go

create table **tblMaterials** ( Materials\_id int not null, Grafics\_id int null, Materials varchar(255) not null, constraint PK\_TBLMATERIALS primary key nonclustered (Materials\_id)

)

go

create table **tblNPlot** ( NPlot\_id int not null, Order\_id int null, Head varchar(255) not null, constraint PK\_TBLNPLOT primary key nonclustered (NPlot\_id)

)

go

create table **tblOrder** ( Order\_id int not null, NPlot\_id int null, ChaRoad\_id int null, ChaBuild\_id int null, Object varchar(255) not null, Organisetion varchar(255) not null, BuildManagement varchar(255) not null, constraint PK\_TBLORDER primary key nonclustered (Order\_id)

)

go

create table **tblReport** ( Report\_id int not null, Order\_id int null, JobType varchar(255) not null, Completion datetime null, Used money null, constraint PK\_TBLREPORT primary key nonclustered (Report\_id)

)

go

create table **tblTech** ( Tech\_id int not null, Engeneers varchar(255) null, Technology varchar(255) null, Technics varchar(255) null, constraint PK\_TBLTECH primary key nonclustered (Tech\_id)

)

go

create table **tblWorkers** ( Workers\_id int not null, Brigade\_id int null, Bricklayer varchar(255) null, Concrete varchar(255) null, Handler varchar(255) null, Welders varchar(255) null, Electricity varchar(255) null, Driver varchar(255) null, Locksmiths varchar(255) null, constraint PK\_TBLWORKERS primary key nonclustered (Workers\_id)

)

go

Створення зв’язків:

alter table rl\_10 add constraint FK\_RL\_10\_RL\_10\_TBLNPLOT foreign key (NPlot\_id) references tblNPlot (NPlot\_id)

go

alter table rl\_10 add constraint FK\_RL\_10\_RL\_14\_TBLMACHI foreign key (Machines\_id) references tblMachines (Machines\_id)

go

alter table rl\_11 add constraint FK\_RL\_11\_RL\_11\_TBLNPLOT foreign key (NPlot\_id) references tblNPlot (NPlot\_id)

go

alter table rl\_11 add constraint FK\_RL\_11\_RL\_13\_TBLWORKE foreign key (Workers\_id) references tblWorkers (Workers\_id)

go

alter table rl\_9 add constraint FK\_RL\_9\_RL\_15\_TBLNPLOT foreign key (NPlot\_id) references tblNPlot (NPlot\_id)

go

alter table rl\_9 add constraint FK\_RL\_9\_RL\_16\_TBLTECH foreign key (Tech\_id) references tblTech (Tech\_id)

go

alter table tblBrigade add constraint FK\_TBLBRIGA\_RL\_7\_TBLGRAFI foreign key (Grafics\_id) references tblGrafics (Grafics\_id)

go

alter table tblBrigade add constraint FK\_TBLBRIGA\_RL\_9\_TBLNPLOT foreign key (NPlot\_id) references tblNPlot (NPlot\_id)

go

alter table tblChaBuild add constraint FK\_TBLCHABU\_RL\_1\_TBLORDER foreign key (Order\_id) references tblOrder (Order\_id)

go

alter table tblChaRoad add constraint FK\_TBLCHARO\_RL\_2\_TBLORDER foreign key (Order\_id) references tblOrder (Order\_id)

go

alter table tblGrafics add constraint FK\_TBLGRAFI\_RL\_5\_TBLORDER foreign key (Order\_id) references tblOrder (Order\_id)

go

alter table tblGrafics add constraint FK\_TBLGRAFI\_RL\_8\_TBLBRIGA foreign key (Brigade\_id) references tblBrigade (Brigade\_id)

go

alter table tblMaterials add constraint FK\_TBLMATER\_RL\_6\_TBLGRAFI foreign key (Grafics\_id) references tblGrafics (Grafics\_id)

go

alter table tblNPlot

add constraint FK\_TBLNPLOT\_RL\_18\_TBLORDER foreign key (Order\_id) references tblOrder (Order\_id)

go

alter table tblOrder add constraint FK\_TBLORDER\_RL\_19\_TBLNPLOT foreign key (NPlot\_id) references tblNPlot (NPlot\_id)

go

alter table tblOrder add constraint FK\_TBLORDER\_RL\_3\_TBLCHARO foreign key (ChaRoad\_id) references tblChaRoad (ChaRoad\_id)

go

alter table tblOrder add constraint FK\_TBLORDER\_RL\_4\_TBLCHABU foreign key (ChaBuild\_id) references tblChaBuild (ChaBuild\_id)

go

alter table tblReport add constraint FK\_TBLREPOR\_RL\_17\_TBLORDER foreign key (Order\_id) references tblOrder (Order\_id)

go

alter table tblWorkers add constraint FK\_TBLWORKE\_RL\_12\_TBLBRIGA foreign key (Brigade\_id) references tblBrigade (Brigade\_id)

go

**Виконання запитів**

1. Одержати перелік будівельних правлінь або ділянок та їх керівників

*select o.buildmanagement, o.nplot\_id, n.head*

*from tblorder as o, tblnplot as n*

*where o.nplot\_id=n.nplot\_id*

2. Одержати список фахівців інженерно-технічного складу позначеної ділянки або будівельного правління із зазначенням їх посад

*select distinct(o.buildmanagement), o.nplot\_id, t.engineers, t.technology, t.technics*

*from tblorder as o, tblnplot as n, tbltech as t*

*where o.nplot\_id=3*

3. Одержати перелік об'єктів, що зводяться зазначеним будівельним правлінням або ділянкою, і графіки їхнього зведення

*select distinct(o.object), o.buildmanagement, o.nplot\_id, g.jobtype, g.termperformance*

*from tblorder as o, tblgrafics as g*

*where o.order\_id=g.order\_id*

4. Одержати склад бригад, що працювали (працюють) на будівництві зазначеного об'єкта

select o.object, w.bricklayer, w.concrete, w.handler, w.welders, w.electricity, w.driver, w.locksmiths, w.workers\_id

from tblorder as o, tblworkers as w

where o.nplot\_id=w.nplot\_id

5. Одержати перелік будівельної техніки, наданої зазначеному будівельному правлінню

select o.buildmanagement, m.bulldozers, m.cranes, m.excavator, 3

from tblorder as o, tblmachines as m

where o.nplot\_id=3

order by 3

6. Одержати перелік будівельної техніки, виділеної на зазначений об’єкт або тієї, що працювала там упродовж зазначеного періоду часу

select o.object, o.buildmanagement, m.bulldozers, m.cranes, m.excavator, g.termperformance, m.nplot\_id

from tblorder as o, tblmachines as m, tblgrafics as g

where o.nplot\_id=m.nplot\_id and o.order\_id=g.order\_id and g.TermPerformance<'01.01.2012'

7. Одержати графік і кошторис на будівництво зазначеного об'єкта

select o.object, g.jobtype, g.kyshty

from tblorder as o, tblgrafics as g

where o.order\_id=g.order\_id

8. Одержати звіт про спорудження зазначеного об’єкта

select o.object, r.jobtype, r.used

from tblorder as o, tblreport as r

where o.order\_id=r.order\_id and o.object='Дім'

9. Одержати перелік об’єктів, що зводяться у деякому будівельному правлінні або в цілому по організації, та на яких у позначений період часу виконувався зазначений вид будівельних робіт

select o.object, o.buildmanagement, g.jobtype, g.termperformance from tblorder as o, tblgrafics as g where o.order\_id=g.order\_id and g.termperformance= 01.01.2012

10. Одержати перелік видів будівельних робіт, по яких мало місце перевищення строків виконання на зазначеній ділянці, будівельному правлінні або в цілому по організації

select o.object, o.BuildManagement, r.jobtype, g.termperformance, r.completion

from tblorder as o, tblreport as r, tblgrafics as g

where o.order\_id=g.order\_id and o.order\_id=r.order\_id

and g.termperformance < r.completion and g.jobtype=r.jobtype

11. Одержати перелік будівельних матеріалів, по яких мало місце перевищення кошторису на зазначеній ділянці, будівельному правлінні або в цілому по організації

select o.object, o.BuildManagement, m.materials, g.kyshty, r.used

from tblorder as o, tblmaterials as m, tblgrafics as g, tblreport as r

where o.order\_id=g.order\_id and o.order\_id=r.order\_id and g.grafics\_id=m.grafics\_id and

g.kyshty<r.used and g.jobtype=r.jobtype

12. Одержати перелік видів будівельних робіт, виконаних зазначеною бригадою протягом позначеного періоду часу із вказівкою об'єктів, де ці роботи виконувалися

select o.object, g.jobtype, g.termperformance, b.brigade\_id

from tblorder as o, tblgrafics as g, tblbrigade as b

where o.nplot\_id=b.nplot\_id and o.order\_id=g.grafics\_id and b.brigade\_id=3

13. Одержати перелік бригад, що виконували зазначений вид будівельних робіт протягом позначеного періоду часу із зазначенням об'єктів, де ці роботи виконувалися

select o.object, g.jobtype, b.brigade\_id, g.termperformance

from tblorder as o, tblgrafics as g, tblbrigade as b

where o.nplot\_id=b.nplot\_id and o.order\_id=g.grafics\_id and g.jobtype='ремонт'

**Лістинг програми**

unit Unit1;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, DBCtrls, StdCtrls, Menus, DB, Grids, DBGrids, ADODB;

type TForm1 = class(TForm) ADOConnection1: TADOConnection; ADOConnection2: TADOConnection; ADOTable1: TADOTable; ADOQuery1: TADOQuery; DBGrid1: TDBGrid; DataSource1: TDataSource; DataSource2: TDataSource; MainMenu1: TMainMenu; ComboBox1: TComboBox; DBNavigator1: TDBNavigator; OpenDialog1: TOpenDialog; SaveDialog1: TSaveDialog; File1: TMenuItem; N1: TMenuItem; N2: TMenuItem; Button6: TButton; Button7: TButton; procedure ComboBox1Change(Sender: TObject); procedure N2Click(Sender: TObject); procedure N1Click(Sender: TObject); procedure FormShow(Sender: TObject); procedure Button6Click(Sender: TObject); procedure Button7Click(Sender: TObject); private { Private declarations } public { Public declarations } end;

var Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4, Unit5;

{$R \*.dfm}

procedure TForm1.FormShow(Sender: TObject);

begin

Form2.ShowModal;

end;

procedure TForm1.ComboBox1Change(Sender: TObject);

begin

if (Combobox1.Text='Номер ділянки') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblNPlot';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Характеристика заказу на мости та дороги') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblChaRoad';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Характеристика заказу на житловий будинок') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblChaBuild';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Таблиця що містить інформацію про закази') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblOrder';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Графік зведення обєкта ') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblGrafics';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Звіт виконання роботи') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblReport';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Бригади') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblBrigade ';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Інженерно-технічний персонал') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblTech ';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='будівельна техніка') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblMachines ';form1.ADOTable1.Active:=true;

end;

if (Combobox1.Text='Робітники') then

begin

form1.ADOTable1.Active:=false;

form1.ADOTable1.TableName:='tblWorkers ';form1.ADOTable1.Active:=true;

end;

end;

procedure TForm1.N2Click(Sender: TObject);

begin

close();

end;

procedure TForm1.N1Click(Sender: TObject);

begin ShowMessage('Anisimov Andriy');

end;

procedure TForm1.Button6Click(Sender: TObject);

begin

form5.ShowModal;

if (Form5.modalresult <> 6) then

begin

ShowMessage('Пароль неверный');

end

else

begin

form3.ShowModal;

end;

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

form4.ShowModal;

end;

end.

unit Unit2;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, ExtCtrls, StdCtrls, jpeg;

type TForm2 = class(TForm) Image1: TImage; Timer1: TTimer; procedure Image1Click(Sender: TObject); private { Private declarations } public { Public declarations } end;

var Form2: TForm2;

implementation

{$R \*.dfm}

procedure TForm2.Image1Click(Sender: TObject);

begin

close;

end;

end.

unit Unit3;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Menus, DB, ADODB, Grids, DBGrids;

type TForm3 = class(TForm) Label1: TLabel; DBGrid2: TDBGrid; ADOConnection1: TADOConnection; ADOConnection2: TADOConnection;

ADOTable1: TADOTable; ADOQuery1: TADOQuery; DataSource1: TDataSource; DataSource2: TDataSource; MainMenu1: TMainMenu; File1: TMenuItem; N1: TMenuItem; N2: TMenuItem; OpenDialog1: TOpenDialog; SaveDialog1: TSaveDialog; Label2: TLabel; Button1: TButton; Memo1: TMemo; Button2: TButton; Button3: TButton; Button4: TButton; Button5: TButton; procedure Button1Click(Sender: TObject); procedure Button5Click(Sender: TObject); procedure Button4Click(Sender: TObject); procedure Button2Click(Sender: TObject); procedure Button3Click(Sender: TObject); procedure N2Click(Sender: TObject); procedure FormCreate(Sender: TObject); private { Private declarations } public { Public declarations } end;

var Form3: TForm3;

implementation

{$R \*.dfm}

procedure TForm3.Button1Click(Sender: TObject);

begin

ADOQuery1.close; {Деактивируем запрос в качестве одной из мер предосторожности }

ADOQuery1.SQL.Clear; {Стираем любой предыдущий запрос}

If Memo1.Lines[0] <> '' {Проверяем на предмет пустого ввода} then

ADOQuery1.SQL.Add(Memo1.Text) {Назначаем свойству SQL текст Memo}

else

begin

messageDlg('Введіть sql запрос', mtError, [mbOK], 0);

exit;

end;

try {перехватчик ошибок}

ADOQuery1.Open; {Выполняем запрос и открываем набор данных}

except {секция обработки ошибок}

On e : EDatabaseError do {e - новый дескриптор ошибки}

messageDlg(e.message,mtError,[mbOK],0); {показываем свойство message объекта e}

end;{окончание обработки ошибки}

Memo1.ReadOnly:=false;

end;

procedure TForm3.Button5Click(Sender: TObject);

begin

Memo1.Clear;

end;

procedure TForm3.Button4Click(Sender: TObject);

begin

if SaveDialog1.Execute then

Memo1.Lines.SaveToFile(SaveDialog1.FileName);

end;

procedure TForm3.Button2Click(Sender: TObject);

begin

Memo1.Lines.SaveToFile(OpenDialog1.FileName)

end;

procedure TForm3.Button3Click(Sender: TObject);

begin

Memo1.ReadOnly:=true;

openDialog1.Filter:='Текстові файли|\*.txt';

if OpenDialog1.Execute and FileExists (OpenDialog1.FileName)

then Memo1.Lines.LoadFromFile(OpenDialog1.FileName);

end;

procedure TForm3.N2Click(Sender: TObject);

begin

close();

end;

procedure TForm3.FormCreate(Sender: TObject);

begin

ShowMessage('Anisimov Andriy');

end;

end.

unit Unit4;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Menus, DB, ADODB, Grids, DBGrids;

type

TForm4 = class(TForm) Label1: TLabel; DBGrid2: TDBGrid; ADOConnection1: TADOConnection; ADOConnection2: TADOConnection; ADOTable1: TADOTable; ADOQuery1: TADOQuery; DataSource1: TDataSource; DataSource2: TDataSource; MainMenu1: TMainMenu; File1: TMenuItem; N1: TMenuItem; N2: TMenuItem; OpenDialog1: TOpenDialog; SaveDialog1: TSaveDialog; Label2: TLabel; Button1: TButton; Memo1: TMemo; Button3: TButton; procedure Button3Click(Sender: TObject); procedure Button1Click(Sender: TObject); procedure N2Click(Sender: TObject); procedure FormCreate(Sender: TObject); private { Private declarations } public { Public declarations } end;

var Form4: TForm4;

implementation

{$R \*.dfm}

procedure TForm4.Button3Click(Sender: TObject);

begin

Memo1.ReadOnly:=true;

openDialog1.Filter:='Текстові файли|\*.txt';

if OpenDialog1.Execute and FileExists (OpenDialog1.FileName)

then Memo1.Lines.LoadFromFile(OpenDialog1.FileName);

end;

procedure TForm4.Button1Click(Sender: TObject);

begin

ADOQuery1.close; {Деактивируем запрос в качестве одной из мер предосторожности }

ADOQuery1.SQL.Clear; {Стираем любой предыдущий запрос}

If Memo1.Lines[0] <> '' {Проверяем на предмет пустого ввода} then

ADOQuery1.SQL.Add(Memo1.Text) {Назначаем свойству SQL текст Memo}

else

begin

messageDlg('Введіть sql запрос', mtError, [mbOK], 0);

exit;

end;

try {перехватчик ошибок}

ADOQuery1.Open; {Выполняем запрос и открываем набор данных}

except {секция обработки ошибок}

On e : EDatabaseError do {e - новый дескриптор ошибки}

messageDlg(e.message,mtError,[mbOK],0); {показываем свойство message объекта e}

end;{окончание обработки ошибки}

Memo1.ReadOnly:=false;

end;

procedure TForm4.N2Click(Sender: TObject);

begin

close();

end;

procedure TForm4.FormCreate(Sender: TObject);

begin

ShowMessage('Anisimov Andriy');

end;

end.

unit Unit5;

interface

uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type TForm5 = class(TForm) Button1: TButton; Edit1: TEdit; procedure oK(Sender: TObject); procedure FormClose(Sender: TObject; var Action: TCloseAction); private { Private declarations } public { Public declarations } end;

var Form5: TForm5;

implementation

{$R \*.dfm}

procedure TForm5.oK(Sender: TObject);

begin

if (Edit1.Text = '0000') then ModalResult:=6;

close;

end;

procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if (Edit1.Text = '0000') then ModalResult:=6;

end;

end.

**Висновок**

При виконанні курсової роботи я пригадав як створювати: базу даних, концептуальну та фізичну модель БД, SQL запити, запити на створення таблиць та зв’язків між ними. Освіжив свої навики в SQL quarry analyzer, SQL Enterprise manager та Power designer. А також написав програму на мові Delphi, яка допомогла б користувачу з легкістю користуватися створеною мною базою даних.

27