**Курсов проект по База Данни**

На Стефан Христов Димитров Проверил: доц. Диян Динев

Трети Курс, ФИТА

КСТ 1Б

19621416



**Задание – Музей**

Да се проектира и реализирабаза от данни поизбрана от студента тематика, която съдържаминимум 6таблици,нормализирани доIII-танормална форма-на СУБД по избор (Oracle, MS SQL Server, MySQL,...), както ипрограмен интерфейс (реализиран на C#, PHP, JAVA, ...)-във вид на web / desktop /mobile–по избор.

Програмниятинтерфейс трябва да реализира следните възможности:  
-Добавяне/редактиране/ изтриване на информация към таблиците;  
-Възможности за търсения и справки (мин. 5) по различни критерии;  
-Печат на справки

Реализирания проект е база данни и интерфейс на **музей**. Целта е да се записват в база данни групите на които идват в музея, работниците които работят със групите и техните позиции. Също така и компаниите които резервират самите групи, датата на пристигане на групите, броят хора в групата, цена на човек, залата в която е била групата и какъв вид предмети има в залата.

**Използвани средства**

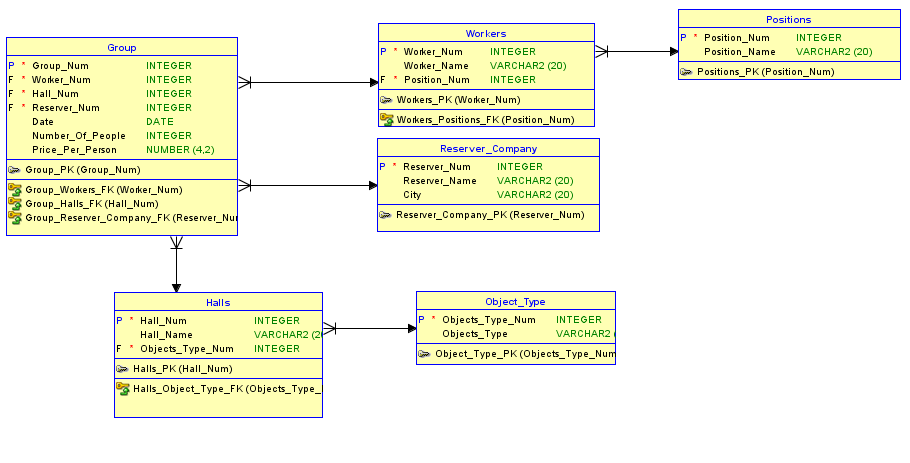
За реализиране на проекта са използвани Microsoft Access и C# Windows Form Application в Microsoft Visual Studio.

**Microsoft Access** е един от най- лесните начини за създаване на таблици за база данни тъй като не е нужно да се пише код. Също така се свързва лесно към Visual Studio тъй като няма нужда от профил с парола и име. Има малко по-различен синтаксис при създаването на търсения и справки от други SQL бази данни.

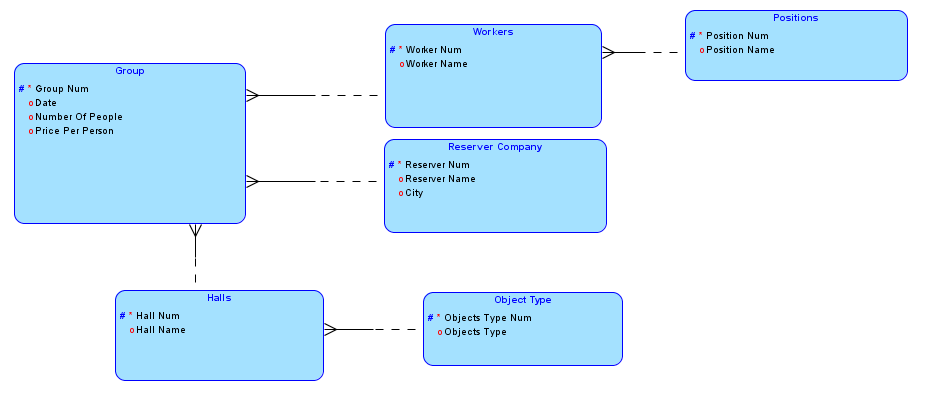
**Microsoft Visual Studio** е мощна интегрирана среда за разработка на софтуерни приложения за Windows и за платформата .NET Framework. Точно .NET Framework е използваната за създаването на интерфейс-а на тази база данни на езикът C# чрез проектът Windows Form Application. Този проект дава лесен начин за създаване на форми чрез които потребителя да манипулира базата данни

**Модел**

**Релационен**



**Логически**



**Таблици**

**Таблица Group** – съдържа номера на групата, номера на работника работил с групата, номера на залата в която е била групата, номера на компанията която е запазила мястото за групата, датата на развеждане в музея, бройката хора в групата и цената за всеки човек.

**Таблица Workers** – съдържа номера на работника (така се свързва с Groups) името на работника и номера на позицията му.

**Таблица Positions** – съдържа номера на позицията (така се свързва с Workers) и името на позицията

**Таблица Reserver\_Company** – съдържа номера на тур оператора (така се свързва с Groups), името на самата компания и в кой град се намира компанията

**Таблица Halls –** съдържа номера на залата (така се свързва с Groups), името на залата и номера на типа предмети в залата

**Таблица Object\_Type** – съдържа номера на типа предмети (така се свързва с Halls) и какъв е вида на предметите

**Код на програмата**

**1. Създаване на таблиците чрез SQL код**

**Таблица Позиции**

create table Positions

(Position\_Num int, Position\_Name varchar2(20), constraint Positions\_PK primary key(Position\_Num)

);

**Таблица Работници**

create table Workers

(

Worker\_Num int,

Position\_Num int,

Worker\_Name varchar2(20),

constraint Worker\_PK primary key(Worker\_Num),

constraint Positions\_Worker\_FK foreign key(Position\_Num) references positions(Position\_Num)

);

**Таблица Резервиращи Компании**

create table Reserver\_Company

(

Reserver\_Num int,

Reserver\_Name varchar2(20),

City varchar2(20),

constraint Reserver\_PK primary key(Reserver\_Num));

**Таблица Тип Предмети**

create table Оbject\_type

(

Objects\_Type\_Num int,

Objects\_Type varchar2(20),

constraint Objects\_Type\_PK primary key(Objects\_Type\_Num)

);

**Таблица Зали**

create table Halls

(

Hall\_Num int,

Hall\_Name varchar2(20),

Objects\_Type\_Num int,

constraint Hall\_PK primary key(Hall\_Num),

constraint Object\_Halls\_FK foreign key(Objects\_Type\_Num) references object\_type(Objects\_Type\_Num)

);

**Таблица Групи**

create table Groups

(

Group\_Num int,

Worker\_Num int,

Hall\_Num int,

Reserver\_Num int,

Date\_of\_Group date,

Number\_of\_People int,

Price\_Per\_Person number(4,2),

constraint Group\_PK primary key(Group\_Num),

constraint Group\_Worker\_FK foreign key(Worker\_Num) references workers(Worker\_Num),

constraint Group\_Hall\_FK foreign key(Hall\_Num) references halls(Hall\_Num),

constraint Group\_Reserver\_FK foreign key(Reserver\_Num) references reserver(Reserver\_Num)

);

**1.2. Създаване на функциите чрез C# с помощта на SQL код**

**Проверяване на данните в таблиците**

private void button4\_Click(object sender, EventArgs e)

{

if (radioButton1.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Groups ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

else if (radioButton2.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Workers ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

else if (radioButton3.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Positions";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

else if (radioButton4.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Reserver\_Company";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

else if (radioButton5.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Halls";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

else if (radioButton6.Checked == true)

{

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from Object\_Type";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

}

}

**Функция за добавяне на Група**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Groups (Worker\_Num,Hall\_Num,Reserver\_Num,Date\_Of\_Group,Number\_of\_People, Price\_Per\_Person) " +

"values('" + textBox1.Text + " ','" + textBox2.Text + "','" + textBox3.Text + "','" + textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за добавяне на Работник**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Workers (Worker\_Name,Position\_Num) values('" + textBox1.Text + " ','" + textBox2.Text + "')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за добавяне на Позиция**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Positions (Position\_Name) values('" + textBox1.Text + " ')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за добавяне на Резервираща Компания**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Reserver\_Company (Reserver\_Name,City) values('" + textBox1.Text + " ','" + textBox2.Text + "')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за добавяне на Зала**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Halls (Hall\_Name,Objects\_Type\_Num) values('" + textBox1.Text + " ','" + textBox2.Text + "')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за добавяне на Тип Предмет**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "insert into Object\_Type (Objects\_Type) values('" + textBox1.Text + " ')";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуваща Група**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Groups set Worker\_Num= '" + textBox1.Text + "' ,Hall\_Num = '" + textBox2.Text + "' ,Reserver\_Num = '" + textBox3.Text + "'" +

",Date\_of\_Group = '" + textBox4.Text + "' ,Number\_Of\_People = '" + textBox5.Text + "' ,Price\_Per\_Person = '" + textBox6.Text + "' where Group\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуващ Работник** oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Workers set Worker\_Name= '" + textBox1.Text + "' ,Position\_Num= '" + textBox2.Text + "' where Worker\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуваща Позиция**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Positions set Position\_Name= '" + textBox1.Text + "' where Position\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close(); MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуваща Компания**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Reserver\_Company set Reserver\_Name= '" + textBox1.Text + "' ,City= '" + textBox2.Text + "' where Reserver\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуваща Зала**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Halls set Hall\_Name= '" + textBox1.Text + "' ,Objects\_Type\_Num= '" + textBox2.Text + "' where Hall\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за промяна на данните на съществуваща Тип на предмети**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "update Object\_Type set Objects\_Type = '" + textBox1.Text + "' where Objects\_Type\_Num = " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Група**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Groups where Group\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Работник**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Workers where Worker\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Позиция**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Positions where Position\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Резервираща Компания**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Reserver\_Company where Reserver\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Зала**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Halls where Hall\_Num= " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за изтриване на Тип на Предмет**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "delete from Object\_Type where Objects\_Type\_Num = " + Upd\_Bx.Text + " ";

cmd.ExecuteNonQuery();

oledbconnection.Close();

MessageBox.Show("WORKED");

**Функция за извеждане на Работници по Позиция**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select \* from (Workers inner join Positions on Workers.Position\_Num = Positions.Position\_Num)" +

" where Position\_Name = '"+textBox1.Text +"' ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на всички Групи обслужени от въведен Работник (Име)**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Groups.Group\_Num, Workers.Worker\_Name,Positions.Position\_Name, Halls.Hall\_Name, Reserver\_Company.Reserver\_Name," +

" Groups.Date\_Of\_Group, Groups.Number\_Of\_People " +

"from ((((Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num)" +

"inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num)" +

"inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num)" +

"inner join Positions on Workers.Position\_Num = Positions.Position\_Num)" +

"where Worker\_Name = '"+ textBox1.Text +"' ";

cmd.Parameters.AddWithValue("Worker\_Name", textBox1.Text);

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на Зали по Тип Предмети**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Halls.Hall\_Num, Halls.Hall\_Name, Object\_Type.Objects\_Type" +

" from (Halls inner join Object\_Type on Halls.Objects\_Type\_Num = Object\_Type.Objects\_Type\_Num)" +

" where Object\_Type.Objects\_Type = '" + textBox1.Text + "' ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на всички групи които са се състояли през Ноември**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Groups.Group\_Num, Workers.Worker\_Name, Halls.Hall\_Name, Reserver\_Company.Reserver\_Name, Groups.Date\_Of\_Group, Groups.Number\_Of\_People " +

"from (((Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num)" +

"inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num)" +

"inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num)" +

"where Date\_Of\_Group > #11/1/2022# and Date\_Of\_Group < #11/30/2022#";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на всички Групи които са били във въведена Зала**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Groups.Group\_Num, Workers.Worker\_Name, Halls.Hall\_Name, Object\_Type.Objects\_Type, Reserver\_Company.Reserver\_Name," +

" Groups.Date\_Of\_Group, Groups.Number\_Of\_People " +

"from ((((Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num)" +

"inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num)" +

"inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num)" +

"inner join Object\_Type on Halls.Objects\_Type\_Num = Object\_Type.Objects\_Type\_Num)" +

"where Hall\_Name = '" + textBox1.Text + "' ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на първите пет въведени групи**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Groups.Group\_Num, Workers.Worker\_Name, Halls.Hall\_Name, Reserver\_Company.Reserver\_Name," +

" Groups.Date\_Of\_Group, Groups.Number\_Of\_People " +

"from (((Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num)" +

"inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num)" +

"inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num)" +

"where Group\_Num <= 5 ";

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**Функция за извеждане на всички Групи запазени от дадена Компания**

oledbconnection.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = oledbconnection;

cmd.CommandText = "select Groups.Group\_Num, Workers.Worker\_Name, Halls.Hall\_Name, Reserver\_Company.Reserver\_Name, Reserver\_Company.City," +

" Groups.Date\_Of\_Group, Groups.Number\_Of\_People " +

"from (((Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num)" +

"inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num)" +

"inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num)" +

"where Reserver\_Name = '" + textBox1.Text + "' ";

cmd.Parameters.AddWithValue("Reserver\_Name", textBox1.Text);

cmd.ExecuteNonQuery();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

oledbconnection.Close();

**3. Забележки**

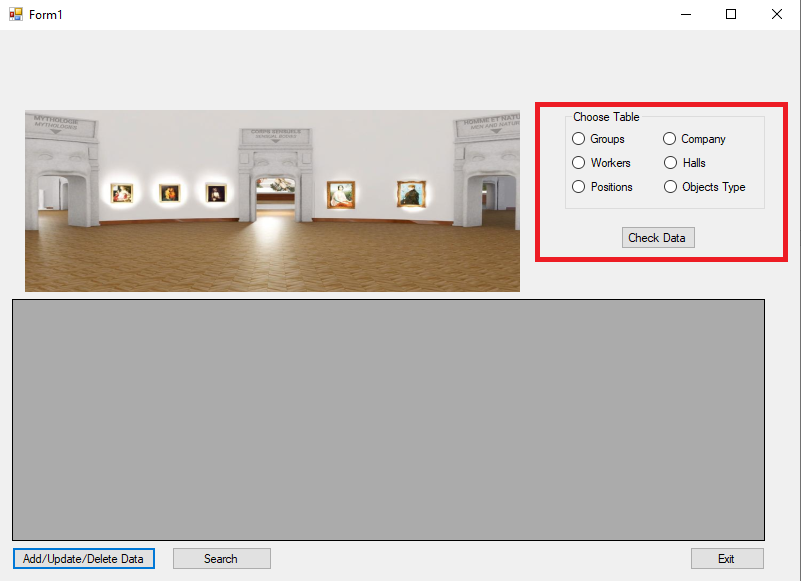
* Тъй като база данни е направена във Microsoft Access номерата на данните (ID) в таблицата са автоматично инкрементирани.
* Има и малка разлика в SQL кода с това че връзката между няколко таблици изисква да има скоби пр.- "from **(((**Groups inner join Workers on Groups.Worker\_Num = Workers.Worker\_Num**)**" + "inner join Halls on Groups.Hall\_Num = Halls.Hall\_Num**)**" + "inner join Reserver\_Company on Groups.Reserver\_Num = Reserver\_Company.Reserver\_Num**)**
* oledbconnection е нужно за да може да се свърже базата данни с приложението тази връзка е инициализирана при зареждането на определената форма.

**Ръководство за потребителя**

Интерфейса е разделен на три форми.

**1.Първата форма** има 4 функционалности – проверяване на данните в различните таблици, отиване към втората форма (за добавяне, изтриване и променяне на данни в таблицата), отиване към третата форма (за справки и търсения) и излизане от интерфейса.

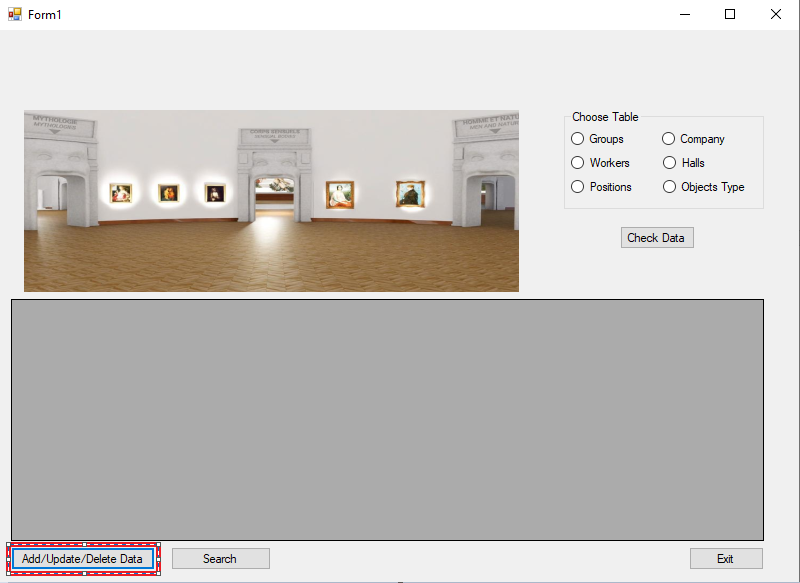
**1.1** Проверяване на данните в таблиците става чрез радио бутони и един нормален бутон.



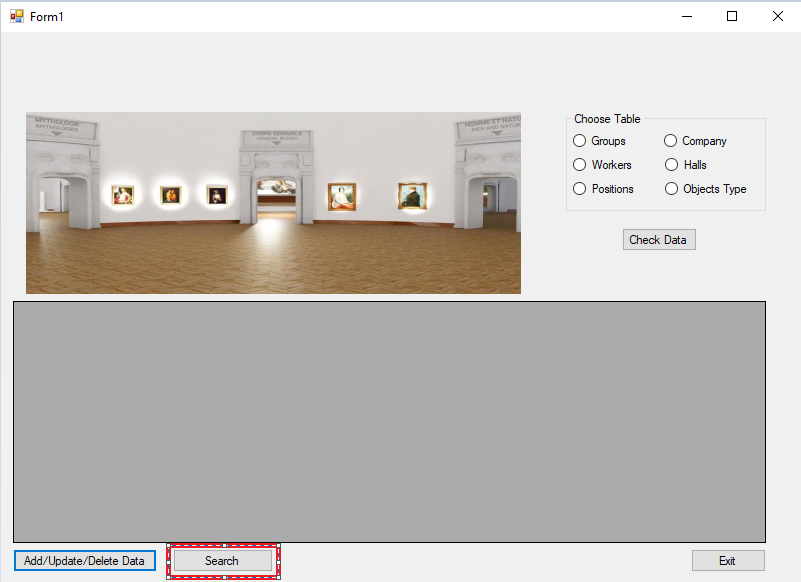
При избирането на един от радио бутоните и натискането на бутона **Check Data** в решетката за данни се изпечатват всички данни от дадена таблицаКартина, която съдържа маса

Описанието е генерирано автоматично

**1.2** Преминаване към формата за добавяне променяне и изтриване на данни става чрез натискането на бутона **Add/Update/Delete Data**

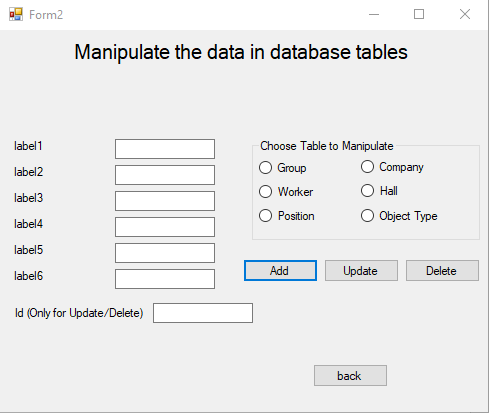


**1.3** Преминаване към формата за справки и търсения става чрез натискането на бутона **Search**



**1.4** Излизане от приложението е чрез бутона **Exit** това е нужно тъй като ако се затвори без този бутон, приложението продължава да работи в background-а.

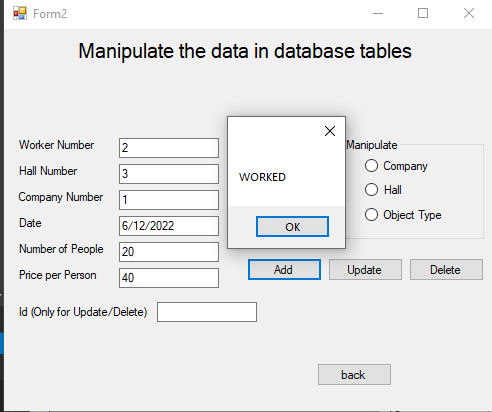
**2.Втората форма** има 4 функционалности – добавяне на данни, изтриване на данни и променяне на данни в отделните таблици на базата данни и връщане към първата форма

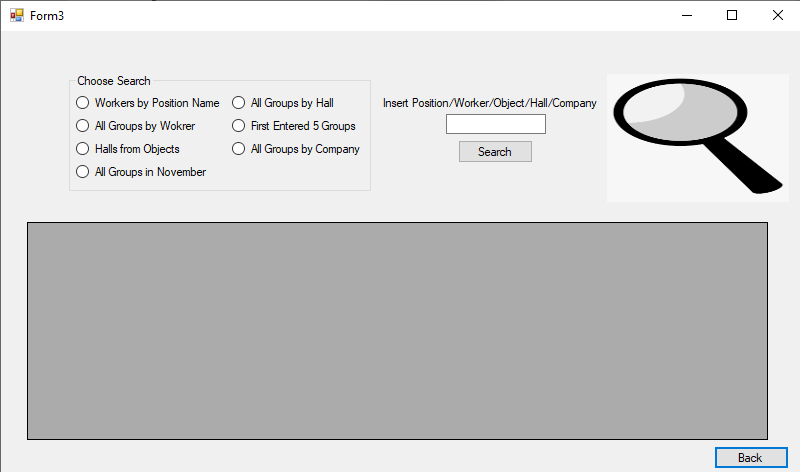


При избиране на коя таблица ще бъде манипулирана(чрез радио бутоните)надписите ще се променят и в някои случаи част от текстовите полета ще бъдат затворени (това е защото някои таблици имат по-малко елементи от други) Картина, която съдържа маса

Описанието е генерирано автоматично

Добавяне, променяне и изтриване на данни става чрез попълване на текстовите полета след избиране на таблица и натискане на съответните бутони – **Add, Update, Delete**. За променяне и истриване на определен обект е нужно да се допълнително поле – **Id**. След натискане на някой от трите бутона излиза текст WORKED за да покаже че съответната манипулация се е извършила успешно.



Бутона **Back** се използва за връщане към първата форм  
  
**3.Третата форма** има функционалностите за справки и търсения и връщане към Първата форма

Чрез радио бутоните се избира начина по който ще се прави справката или търсенето. В някои случаи е нужно да се попълни и текстово поле. След натискане на бутона **Search** в решетката на данни се изпечатва съответната справка

Картина, която съдържа маса

Описанието е генерирано автоматично

**Видовете търсения и справки са:**

* Извеждане на работниците според въведена позиция
* Извеждане на всички групи според въведено име на работник
* Извеждане на зали според въведен вид на предмети
* Извеждане на всички групи които са се състояли през месец Ноември
* Извеждане на всички групи които са били във въведена зала
* Извеждане на първите 5 групи
* Извеждане на всички групи които са били запазени от въведена компания