

МИНОБРНАУКИ РОССИИ

**Нижекамский химико-технологический институт (филиал)  
федерального государственного бюджетного образовательного учреждения  
высшего профессионального образования  
«Казанский национальный исследовательский технологический университет»  
(НХТИ ФГБОУ ВПО «КНИТУ»)**

**Кафедра ИСТ**

## **КУРСОВАЯ РАБОТА**

По дисциплине: Базы данных.

Тема: База данных для учёта площадей помещений ВУЗа.

Выполнил:

студент группы 3724  
Хазиев Р.Н.

Руководитель:

Саримов Н.Н.

г. Нижнекамск  
2022

## Содержание

1. Введение	3
2. Постановка задачи	4
3. Разработка структуры	5
4. Текст программы	9
5. Руководство пользователя	17
6. Тесты	20
7. Литература	21

## 1. Введение

Базами данных (БД) называют электронные хранилища информации, доступ к которым осуществляется с помощью одного или нескольких компьютеров. Обычно БД создаются для хранения и доступа к данным, содержащим сведения о некоторой предметной области, то есть некоторой человеческой деятельности или области реального мира. Проектируемая БД должна обладать определенными свойствами. Выделим основные:

*Целостность.* В каждый момент времени существования БД сведения, содержащиеся в ней, должны быть не противоречивы. Например, нельзя удалять запись, связанную с другой неудаляемой записью.

*Восстанавливаемость.* Данное свойство предполагает возможность восстановления БД после сбоя системы или отдельных видов порчи системы. В основном свойство восстанавливаемости обеспечивается дублированием БД и использованием техники повышенной надежности.

*Безопасность.* Безопасность БД предполагает защиту данных от преднамеренного и непреднамеренного доступа, модификации или разрушения. Применяется запрещение несанкционированного доступа, защита от копирования и криптографическая защита, а также необходимы и административные меры, например ограничение доступа к носителям информации.

*Эффективность.* Свойство эффективности обычно понимается как:

- минимальное время реакции на запрос пользователя;
- минимальные потребности в памяти;
- сочетание этих параметров.

Система управления базами данных (СУБД) – это программные средства, предназначенные для создания, наполнения, обновления и удаления баз данных. Можно выделить четыре этапа развития в СУБД. Первый этап развития связан с организацией баз данных на больших машинах, базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи. На втором этапе все СУБД были рассчитаны на создание БД в основном с монопольным доступом. Так как компьютер персональный, он не был подключен к сети, и база данных на нем создавалась для работы одного пользователя. В редких случаях предполагалась одновременная работа нескольких пользователей. После процесса персонализации начался обратный процесс интеграция – третий этап. Множится количество локальных сетей, все больше информации передается между компьютерами, остро встает задача согласования данных, хранящихся и обрабатывающихся в разных местах. Возникают задачи, связанные с параллельной обработкой транзакций – последовательностей операций над БД. Успешное решение этих задач приводит к появлению распределенных баз данных, сохраняющих все преимущества настольных СУБД и в тоже время позволяющие организовать параллельную обработку и поддержку целостности БД. Четвертый этап характеризуется появлением новой технологии доступа к данным – интранет. Основное отличие этого подхода от технологии

клиент – сервер состоит в том, что отпадает необходимость использования специального программного обеспечения.

Различают три основных вида СУБД: промышленные универсального назначения, промышленные специального назначения и разрабатываемые для конкретного заказчика. Специализированные СУБД создаются для управления базами данных конкретного назначения – бухгалтерские, складские, банковские и т.д. Универсальные СУБД не имеют четко очерченных рамок применения, они рассчитаны «на все случаи жизни» и, как следствие, достаточно сложны и требуют от пользователя специальных знаний. Специализированные и универсальные СУБД относительно дешевы, достаточно надежны и готовы к немедленной работе, а заказные СУБД требуют существенных затрат. Их подготовка к работе и отладка занимают значительный период. Однако, в отличие от промышленных заказные СУБД в максимальной степени учитывают специфику работы заказчика, их интерфейс обычно интуитивно понятен пользователям и не требует от них специальных знаний.

## 2. Постановка задачи

Разработать и реализовать базу данных *для учета площадей помещений ВУЗа*. БД должна содержать информацию о помещениях, подразделениях:

- название или номер помещения, вид помещения (аудитория, кабинет и т.п.), площадь, количество посадочных мест;
- название, вид оборудования в помещении, количество единиц, стоимость;
- название подразделения;
- ФИО, должность руководителя подразделения;
- ФИО, должность материально-ответственного лица.

*Основные требования к функциям системы:*

1. Поиск помещения по названию, номеру, виду;
2. Подсчет общей площади помещений по видам, подразделениям, общего количества посадочных мест, общей стоимости оборудования в помещениях;
3. Печать инвентаризационных ведомостей оборудования по помещениям, подразделениям.

### 3. Разработка структуры

Отправной точкой для процесса разбиения может служить отношение, находящееся в первой нормальной форме. Говорят, что отношение находится в 1НФ если каждый его элемент имеет и всегда будет иметь атомарное значение. Универсальное отношение находится в 1НФ. Процесс разбиения отношения с целью уменьшения вероятности возникновения аномалии называется декомпозицией. Основой для осуществления декомпозиции является понятие функциональной зависимости между атрибутами.

Пусть даны 2 атрибута А и В, говорят что В функционально зависит от А если для каждого значения А существует ровно 1 связанное с ним значение В в каждый момент времени.

Атрибуты А и В могут быть составными, т.е. они могут представлять не 1-ые атрибуты, а группы атрибутов.

В каждой конкретной ситуации ФЗ определяется путем анализа и детализации свойств всех атрибутов и выяснения того, как атрибуты соотносятся между собой.

Процесс декомпозиции может быть усложнен из-за наличия избыточных функциональных зависимостей.

Избыточной функциональной зависимостью называется зависимость, которая может быть получена на основе других имеющихся функциональных зависимостей.

Набор функциональных зависимостей, в котором нет избыточных ФЗ наз. минимальным покрытием. Минимальное покрытие может оказаться не единственным.

Существуют следующие виды избыточных функциональных зависимостей которые необходимо удалить для получения min покрытия:

1) транзитивная зависимость. Если В функционально зависит от А ( $A \rightarrow B$ ) и С функционально зависит от В ( $B \rightarrow C$ ), то функциональная зависимость С от А ( $A \rightarrow C$ ) является избыточной транзитивной зависимостью.

2) добавление1 Если  $A \rightarrow B$ , то  $\langle A, Z \rangle \rightarrow B$  является корректной, но избыточной функциональной зависимостью.

3) добавление2 Если  $A \rightarrow B$ , то  $\langle A, Z \rangle \rightarrow \langle B, Z \rangle$  является корректной, но избыточной функциональной зависимостью.

4) псевдотранзитивность Если  $X \rightarrow Y$ , и  $\langle Y, W \rangle \rightarrow Z$ , то зависимость  $\langle X, W \rangle \rightarrow Z$  является избыточной функциональной зависимостью.

Кроме этих правил для облегчения процесса декомпозиции могут быть использованы след. 2 правила:

1) Объединение Если  $A \rightarrow B$  и  $A \rightarrow C$ , то  $A \rightarrow \langle B, C \rangle$

2) Разложение Если  $A \rightarrow \langle B, C \rangle$ , то  $A \rightarrow B$  и  $A \rightarrow C$ .

Нормальная форма Бойса-Кодда (НФБК). Обобщенный алгоритм метода декомпозиции.

Возможный ключ представляет собой атрибут или набор атрибутов, которые могут быть использованы для данного отношения в качестве первичного ключа. Первичный ключ всегда является возможным ключом, однако не исключено наличие других возможных ключей.

Если  $A \rightarrow B$  и В не зависит функционально от любого подмножества А, то говорят что А представляет собой детерминант В.

Говорят что отношение находится в НФБК, тогда и только тогда когда каждый детерминант отношений является возможным ключом.

Обобщенный алгоритм метода декомпозиций состоит в следующем:

1) Построение универсального отношения.

2) Определение всех функциональных зависимостей между атрибутами универсального отношения.

3) Получение min покрытия.

4) Определение всех детерминант и возможных ключей отношений.

5) Определение того находится ли отношение в НФБК, если «да», то проектирование завершается, иначе отношение не находящееся в НФБК разлагаются на 2 отношения по следующему правилу:

а) пусть отношение  $R(A,B,C,D,E,...)$  не находится в НФБК, тогда в нем определяется функциональная зависимость которая является причиной того, что отношение не находится в НФБК, т.е. левая часть ФЗ является детерминантом, но не является возможным ключом.

б) пусть это будет  $C \rightarrow D$ , тогда формируется 2 новых отношения  $R_1(A,B,C,E,...)$   $R_2(C,D)$ . Здесь в  $R_1$  — отсутствует правая часть ФЗ и ФЗ была полностью использована  $R_2$ .

в) Если имеется цепочка ФЗ вида:  $A \rightarrow B \rightarrow C \rightarrow D$ , то процесс разложения следует начинать с последнего звена цепи.

б) Повторение шагов 4-5 для каждого нового отношения полученного в результате декомпозиции до тех пор, пока все отношения не окажутся в НФБК.

7) Исключение избыточных отношений. Отношение называется избыточным, если:

а) все его атрибуты могут быть найдены в другом отношении;

б) все его атрибуты могут быть получены из нескольких других отношений с помощью серий операций соединений;

8) Проверка всех ФЗ полученных отношений по следующим направлениям:

а) одна и та же ФЗ не должна встречаться более чем в одном отношении;

б) набор ФЗ полученных в результате проектирования должен в точности совпадать с набором присутствующим в min покрытия после 3-го шага.

9) Если в результате получается min покрытие и декомпозиция могут быть получены несколько вариантов БД то выбирается вариант, наиболее отвечающий требованиям пользователя.

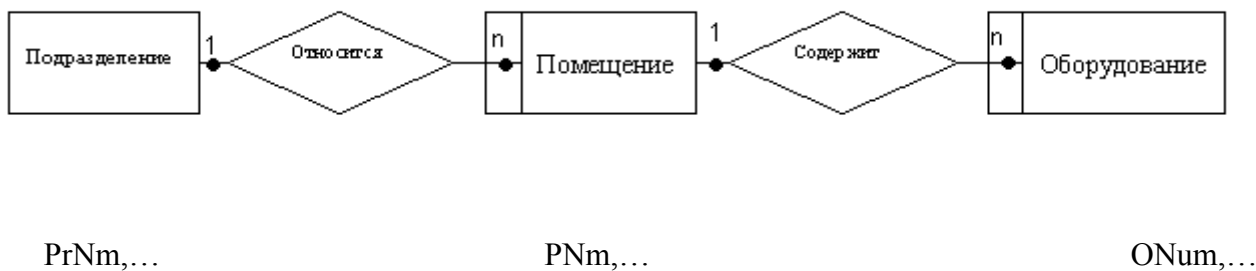
Проектирование базы данных для учёта площадей помещений ВУЗа методом сущностей и связей.

Выпишем все атрибуты, которые интересны нам для создания БД:

Название поля	Тип поля	Размер	Пояснение
PNm	A	30	Название помещения
PVid	A	30	Вид помещения
PS	S		Площадь помещения
PMest	S		Количество мест
PPr	S		Подразделение
ONm	A	30	Название оборудования
Ovid	A	30	Вид оборудования
OSht	S		Количество
OSt	S		Стоимость
ONum	S		Инвентаризационный номер
PrNm	A	30	Название подразделения
PrRNm	A	30	ФИО руководителя
PrRD	A	30	Должность руководителя
PrMNm	A	30	ФИО мат. ответственного лица
PrMD	A	30	Должность мат. ответственного лица

В данном случае сущностями являются **Помещения, Оборудование и Подразделения.**

ER-диаграмма:



Построим отношения по правилу:

Если степень бинарной связи 1:n и класс принадлежности n-связной сущности обязательный, то достаточно использовать два отношения – по одному на каждую сущность.

Получили предварительные отношения:

Подразделение (PrNm, ...)

Помещение (PNm, ... ,PrNm)

Оборудование (ONum, ... ,PNm)

Разместив оставшиеся неиспользованные атрибуты, получим окончательные отношения:

Подразделение (PrNm, PrRNm, PrRD, PrMNm, PrMD)

Помещение (PNm, PVid,PS,PMest,PPr)

Оборудование (ONum, OMest, ONm, Ovid,OSht,OST)

Проверим, находятся ли данные отношения в НФБК:

### 1. Помещения

Каждое Помещение имеет своё уникальное имя, по которому можно определить его Вид, Площадь, Количество мест, Подразделение.

<PNm> →<PVid,PS,PMest,PPr>

Ключи	Детерминанты
PNm	PNm

Так как единственный детерминант является возможным ключом, то данное отношение находится в НФБК.

### 2. Оборудование

Каждое Оборудование имеет свой уникальный Номер, по которому можно определить его Место,Название, Вид, Количество, Стоимость.

<ONum> →<OMest,ONm,Ovid,OSht,OST>

Ключи	Детерминанты
ONum	ONum

Так как единственный детерминант является возможным ключом, то данное отношение находится в НФБК.

### 3. Подразделение

Каждое Подразделение имеет своё уникальное Название, по которому можно определить его Руководителя, Должность руководителя, Материально ответственное лицо, Должность материально ответственного лица.

$\langle \text{PrNm} \rangle \rightarrow \langle \text{PrRNm}, \text{PrRD}, \text{PrMNm}, \text{PrMD} \rangle$

Ключи	Детерминанты
PrNm	PrNm

Так как единственный детерминант является возможным ключом, то данное отношение находится в НФБК.



#### 4. Текст программы

```
unit Unit1;

interface

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

type
  TForm1 = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    ADOConnection1: TADOConnection;
    ADOQuery1: TADOQuery;
    DSPomeshenia: TDataSource;
    DSPodrazdelenia: TDataSource;
    DSOborudovanie: TDataSource;
    DBGrid2: TDBGrid;
    DBGrid3: TDBGrid;
    Button1: TButton;
    Button2: TButton;
    ADOQuery2: TADOQuery;
    ADOQuery3: TADOQuery;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Edit1: TEdit;
    BitBtn3: TBitBtn;
    Edit2: TEdit;
    BitBtn4: TBitBtn;
    BitBtn5: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4;

{$R *.dfm}
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
Form3.show;
ADOQuery1.insert;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
ADOQuery1.Delete;
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
Form2.Show;
end;
```

```
// Процедура подсчета общей площади помещений
procedure TForm1.BitBtn2Click(Sender: TObject);
```

```
var
i:integer;
sum:real;
begin
sum:=0;
ADOQuery2.First;
for i:=1 to ADOQuery2.RecordCount do
begin
sum:=sum+ADOQuery2.FieldByName('PMest').AsFloat;
ADOQuery2.Next;
end;
Edit1.Text:=FloatToStr(sum);
ADOQuery2.First;
end;
```

```
// Процедура подсчета общей стоимости оборудования
```

```
procedure TForm1.BitBtn3Click(Sender: TObject);
var
i:integer;
sum:real;
begin
sum:=0;
ADOQuery1.First;
for i:=1 to ADOQuery1.RecordCount do
begin
sum:=sum+ADOQuery1.FieldByName('OSht').AsFloat*ADOQuery1.FieldByName('OSt').AsFloat;
ADOQuery1.Next;
end;
Edit2.Text:=FloatToStr(sum);
ADOQuery2.First;
```

```

end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
Form4.show;
end;

procedure TForm1.BitBtn5Click(Sender: TObject);
begin
close;
end;

end.

unit Unit2;

interface

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

type
TForm2 = class(TForm)
BitBtn1: TBitBtn;
DBGrid1: TDBGrid;
DataSource1: TDataSource;
Edit1: TEdit;
Edit2: TEdit;
BitBtn2: TBitBtn;
Edit3: TEdit;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
Edit4: TEdit;
BitBtn5: TBitBtn;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
procedure BitBtn1Click(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

var
Form2: TForm2;

```

implementation

uses Unit1;

{\$R \*.dfm}

//Процедуры поиска

procedure TForm2.BitBtn1Click(Sender: TObject);

begin

if (Edit1.Text="")

then

begin

ShowMessage('Внимание! '+#13+'Введите название!');

exit;

end;

Form1.ADoQuery2.Close;

Form1.ADoQuery2.SQL.Clear;

Form1.ADoQuery2.SQL.Add('Select \* FROM Pomeshenia where PNm like '+'"%'+Edit1.Text+'%");

Form1.ADoQuery2.Open;

end;

procedure TForm2.BitBtn2Click(Sender: TObject);

begin

if (Edit2.Text="")

then

begin

ShowMessage('Внимание! '+#13+'Введите вид!');

exit;

end;

FORM1.ADoQuery2.Close;

Form1.ADoQuery2.SQL.Clear;

Form1.ADoQuery2.SQL.Add('Select \* FROM Pomeshenia where PVIId like '+'"%'+Edit2.Text+'%");

Form1.ADoQuery2.Open;

end;

// Процедура подсчета площади

procedure TForm2.BitBtn3Click(Sender: TObject);

var

i:integer;

sum:real;

begin

sum:=0;

Form1.ADOQuery2.First;

for i:=1 to Form1.ADOQuery2.RecordCount do

begin

sum:=sum+Form1.ADOQuery2.FieldByName('PS').AsFloat;

Form1.ADOQuery2.Next;

end;

Edit3.Text:=FloatToStr(sum);

Form1.ADOQuery2.First;

```

end;

procedure TForm2.BitBtn4Click(Sender: TObject);
begin
  if (Edit3.Text="")
  then
    begin
      ShowMessage('Внимание! '+#13+'Введи подразделение!');
      exit;
    end;

FORM1.ADoQuery2.Close;
Form1.ADoQuery2.SQL.Clear;
Form1.ADoQuery2.SQL.Add('Select * FROM Pomeshenia where PPr like '+'%'+Edit4.Text+'%');
Form1.ADoQuery2.Open;

end;

procedure TForm2.BitBtn5Click(Sender: TObject);
begin
close;
end;

end.

unit Unit3;

interface

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

type
  TForm3 = class(TForm)
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }

```

```

end;

var
  Form3: TForm3;

implementation

uses Unit1;

{$R *.dfm}

// Процедура добавления оборудования
procedure TForm3.BitBtn1Click(Sender: TObject);
begin
  if (DBEdit1.Text=""or
      (DBEdit2.Text=""or
      (DBEdit3.Text=""or
      (DBEdit4.Text=""or
      (DBEdit5.text=""
      then
        begin
          ShowMessage(' Внимание! '+#13+'Одно или несколько полей не заполнено!');
          exit;
        end;

  Form1.ADOQuery1.FieldByName('OMest').AsString:=DBEdit1.Text;
  Form1.ADOQuery1.FieldByName('ONm').AsString:=DBEdit2.Text;
  Form1.ADOQuery1.FieldByName('OVid').AsString:=DBEdit3.Text;
  Form1.ADOQuery1.FieldByName('OSht').AsInteger:=StrToInt(DBEdit4.Text);
  Form1.ADOQuery1.FieldByName('OSt').AsInteger:=StrToInt(DBEdit5.Text);

  Form1.ADOQuery1.post;
close;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
close;
end;

end.

unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Grids, DBGrids, QRCtrls, QuickRpt, ExtCtrls,
  DB;

type

```

```

TForm4 = class(TForm)
  QuickRep1: TQuickRep;
  ColumnHeaderBand1: TQRBand;
  DetailBand1: TQRBand;
  TitleBand1: TQRBand;
  QRSysData1: TQRSysData;
  QRLabel1: TQRLabel;
  QRLabel2: TQRLabel;
  QRLabel3: TQRLabel;
  QRLabel4: TQRLabel;
  QRLabel5: TQRLabel;
  QRLabel6: TQRLabel;
  QRLabel7: TQRLabel;
  QRSysData2: TQRSysData;
  QRDBText1: TQRDBText;
  QRDBText2: TQRDBText;
  QRDBText3: TQRDBText;
  QRDBText4: TQRDBText;
  QRDBText5: TQRDBText;
  QRDBText6: TQRDBText;
  DBGrid1: TDBGrid;
  BitBtn1: TBitBtn;
  Edit1: TEdit;
  BitBtn2: TBitBtn;
  Edit2: TEdit;
  DataSource1: TDataSource;
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  Label1: TLabel;
  Label2: TLabel;

  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;

implementation

uses Unit1;

{$R *.dfm}

procedure TForm4.BitBtn1Click(Sender: TObject);
begin
  if (Edit1.Text='')

```

```

then
    begin
        ShowMessage('Внимание! '+#13+'поле не заполнено!');
        exit;
    end;

Form1.ADoQuery1.Close;
Form1.ADoQuery1.SQL.Clear;
Form1.ADoQuery1.SQL.Add('Select * FROM Oborudovanie where OMest like
'+""'+Edit1.Text+'%");
Form1.ADoQuery1.Open;
end;

procedure TForm4.BitBtn2Click(Sender: TObject);
begin
    if (Edit2.Text=")
then
    begin
        ShowMessage('Внимание! '+#13+'поле не заполнено!');
        exit;
    end;

Form1.ADoQuery1.Close;
Form1.ADoQuery1.SQL.Clear;
Form1.ADoQuery1.SQL.Add('Select * FROM Oborudovanie where OVID like '+""'+Edit2.Text+'%");
Form1.ADoQuery1.Open;
end;

// Процедура формирования ведомости
procedure TForm4.BitBtn3Click(Sender: TObject);
begin
    QuickRep1.PreviewModal;
end;

procedure TForm4.BitBtn4Click(Sender: TObject);
begin
    close;
end;

end.

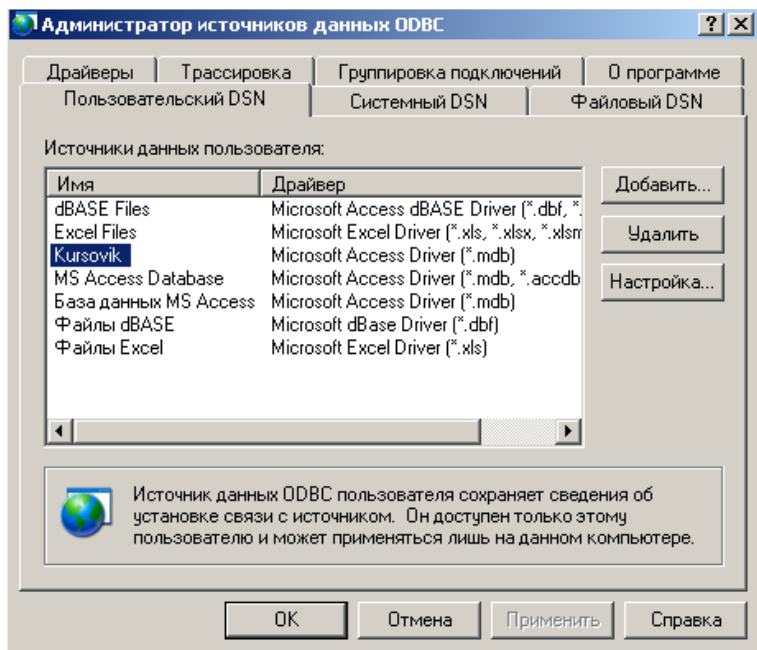
```



## 5. Руководство пользователя

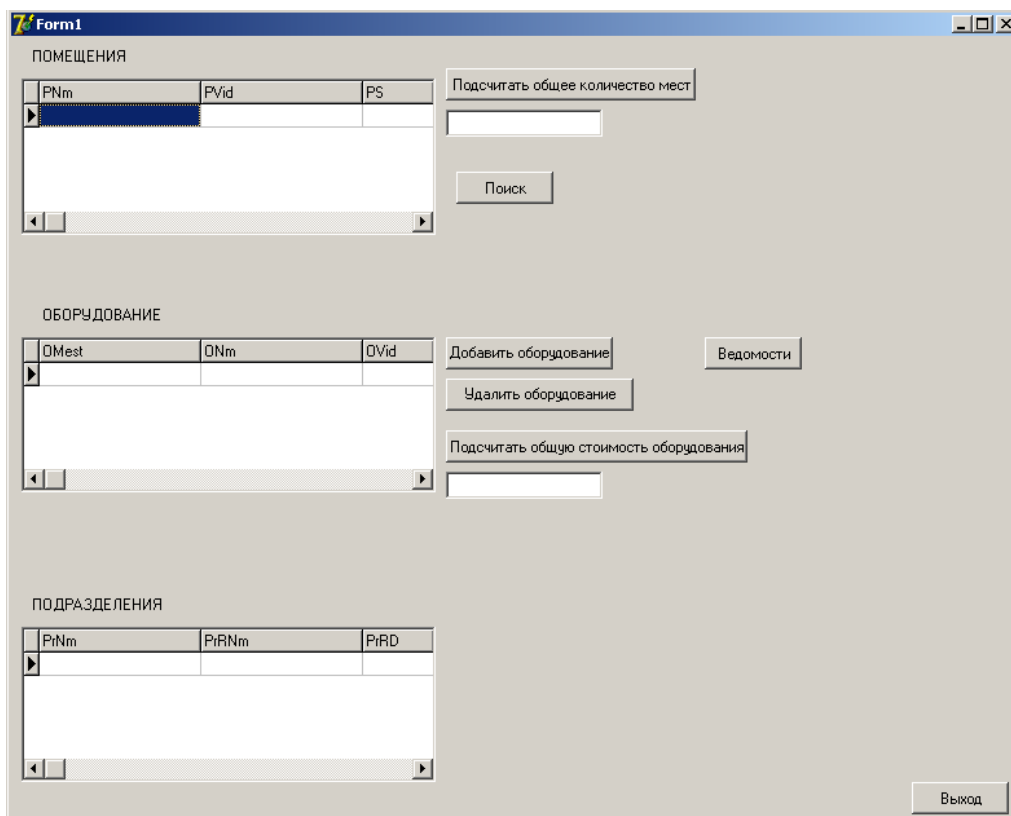
### Настройка ODBC

Перед запуском базы данных необходимо провести настройку: Пуск –Панель управления – Администрирование–ODBC Administrator. Затем добавляем источник данных в нашем случае:Kursovik



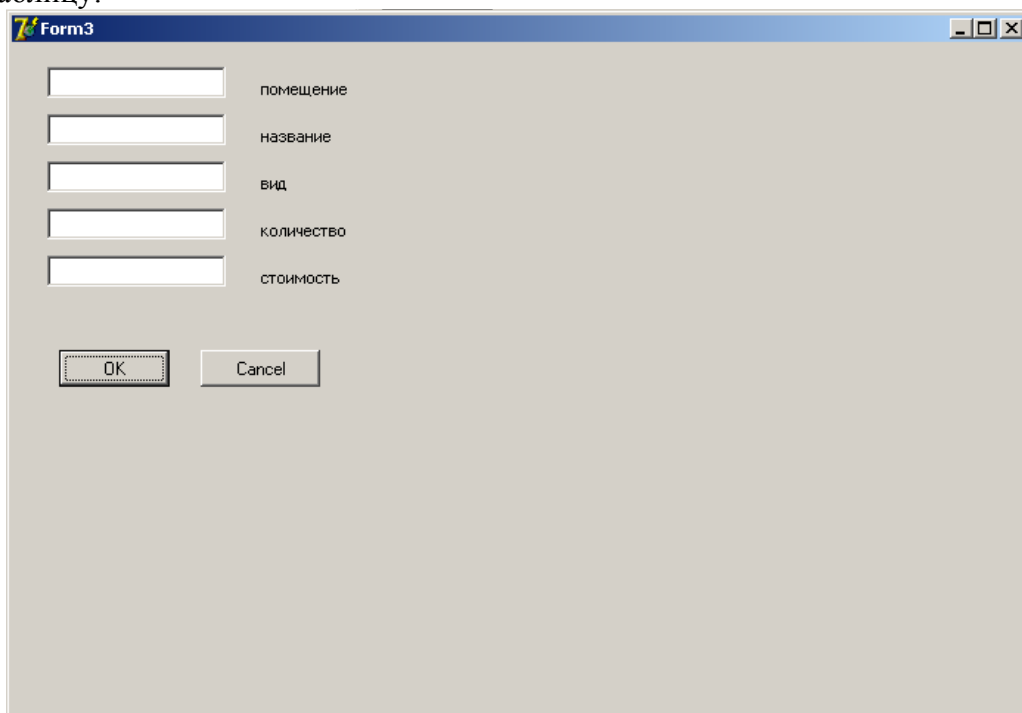
### Главное окно

На данной форме можно осуществить просмотр текущей базы данных.



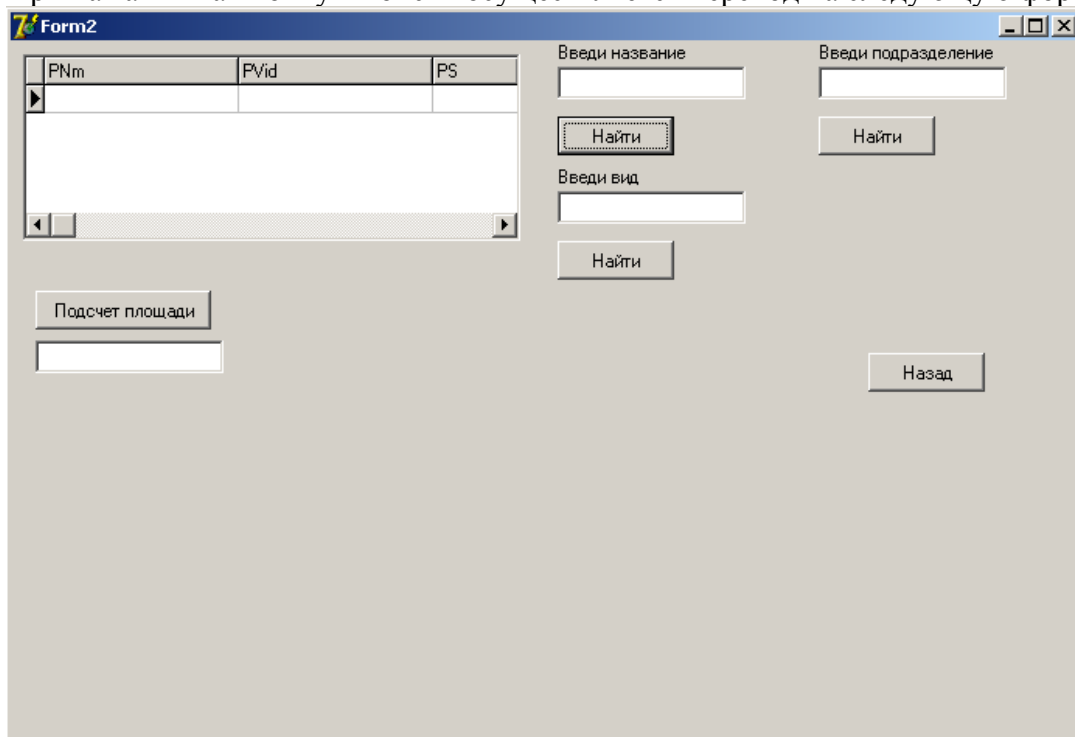
Так же имеются кнопки для подсчета общей площади помещений и общей стоимости оборудования находящихся в БД.

С помощью кнопки «Добавить» оборудование можно добавить данные в соответствующую таблицу:



The screenshot shows a dialog box titled "Form3" with a standard Windows title bar. It contains five text input fields stacked vertically, each with a label to its right: "помещение", "название", "вид", "количество", and "стоимость". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

При нажатии на кнопку «Поиск» осуществляется переход на следующую форму:



The screenshot shows a dialog box titled "Form2" with a standard Windows title bar. On the left side, there is a table with three columns: "PNm", "Pvid", and "PS". The table has one empty row below the header. Below the table is a button labeled "Подсчет площади" and an empty text input field. On the right side, there are three search sections: "Введи название" with an input field and a "Найти" button; "Введи подразделение" with an input field and a "Найти" button; and "Введи вид" with an input field and a "Найти" button. At the bottom right, there is a "Назад" button.

Здесь можно осуществить поиск помещений по названию, виду и подразделению. А так же подсчет площади помещений по соответствующим критериям.

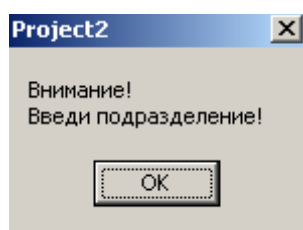
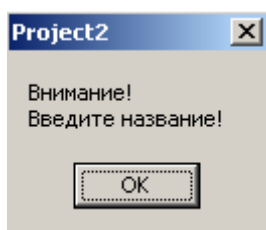
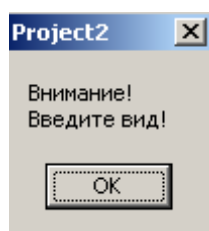
При нажатии кнопки «Ведомости» осуществляется переход на следующую форму:

The screenshot shows a software window titled "Form4" with a standard Windows-style title bar. On the left side, there is a table with three columns: "OMest", "ONm", and "OVid". The first row of the table is highlighted with a blue background and contains an asterisk "\*" in the "OMest" column. Below the table is a horizontal scrollbar. To the right of the table, there are two input fields. The first is labeled "По помещениям" and the second is labeled "По виду". Below each input field is a button labeled "Вывести". In the center of the window, there is a larger button labeled "Сформировать ведомость". In the bottom right corner, there is a button labeled "Назад". The window has a light gray background and a vertical scrollbar on the right side.

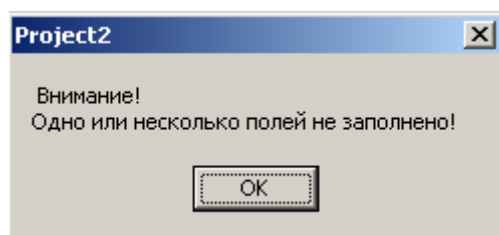
Здесь можно вывести на печать ведомости оборудования по необходимым критериям (помещению или виду). Сначала необходимо произвести поиск оборудования, а затем с помощью кнопки «Сформировать ведомость» можно осуществить печать ведомости.

## 6. Тесты

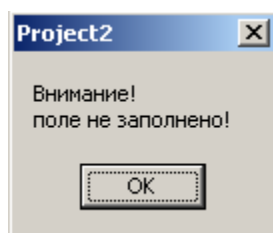
При попытке осуществления поиска по пустым полям, выводятся сообщения:



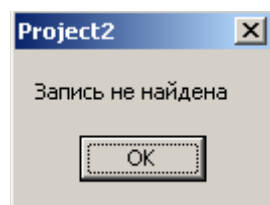
При пропуске какого либо поля при добавлении оборудования, выводится сообщение:



При попытке формирования ведомости без указания критерия, выводится сообщение:



Если при поиске не удастся найти необходимые значения, то выводится сообщение:



## Литература

1. «Базы Данных» А.Д. Хомченко, Санкт-Петербург «Корона принт» 2002;
2. Лекции по БД Саримов Н.Н.;
3. Базы данных: основы, проектирование, использование., Малыгина М.П, Петербург, 2004.-512 с.
4. Методы проектирования баз данных: Метод.указания/Казан. матем. Об-во. Сост. Н.Н.Саримов. Казань, 2000.3
5. Проектирование баз данных: Метод.указания/Нижекамский рег.уч.-мед. Центр; Сост. О.В.Ибущева, А.Р.Нигматуллина, Н.Н.Саримов. Нижекамск, 2007.