

**Рязанский станкостроительный колледж РГРТУ**

**МДК.01.01 Разработка программных модулей**

**Тема 6. Основы Entity Framework**

**Разработка многотабличных БД в среде Visual Studio на языке  
С#»**

**(версия 0.9.1)**

**Рязань 2022**

## Оглавление

Виды форм для многотабличных БД .....	3
<i>Форма список (таблица)</i> .....	3
<i>Форма - бланк</i> .....	3
<i>Связанные таблицы</i> .....	4
Создание модели и классов сущностей БД.....	4
Пример разработки и интерфейса БД Услуги.....	7
<i>Схема БД и модель БД Услуги</i> .....	7
<i>Разработка интерфейса главной формы</i> .....	7
<i>Создание окна Добавить запись</i> .....	8
Алгоритм действий создания окна Добавить запись .....	8
<i>Создание окна Изменить запись</i> .....	10
Алгоритм действий создания окна Изменить запись .....	10
Пример разработки и интерфейса БД Занятия .....	12
<i>Схема БД и модель БД Занятия</i> .....	12
<i>Разработка интерфейса главной формы</i> .....	12
Пример разработки и интерфейса БД Справочник.....	13
<i>Схема БД и модель БД Справочник</i> .....	13
<i>Разработка интерфейса главной формы</i> .....	14
<i>Создание окна Добавить запись</i> .....	14
Алгоритм действий создания окна Добавить запись .....	14
<i>Создание окна Изменить запись</i> .....	19
Алгоритм действий создания окна Изменить запись .....	19
<i>Удаления записи из таблицы Абонент</i> .....	21
Создание связанных таблиц.....	21
<i>Алгоритм создания связанных таблиц.</i> .....	22
Использование LINQ to Entities .....	22
Создание формы регистрации пользователя.....	24
<i>Практическая работа № 20</i> .....	26
<i>Практическая работа № 21</i> .....	43

## Виды форм для многотабличных БД

При разработке многотабличных БД одной из важных задач является отображение информации из БД. Очень часто начинающие разработчики – студенты используют следующую упрощенную концепцию – на каждой форме отображается одна таблица из БД, не учитываются связи между таблицами, т.е. каждая таблица заполняется сама по себе. В связанных таблицах для внешнего ключа не используют поля подстановки или другие методы организации связи. В итоге, например, формируя заказ вместо наименования клиента и товара пользователь должен выбирать коды этих элементов помня на память коды тысяч элементов что естественно невозможно. Реже на форме располагают 2-3 связанные таблицы, в табличном виде сваливая всю информацию в большую кучу.

Правильным подходом является продуманное отображение информации, т.е. главный критерий — это размещение информации в удобном виде для работы с ней пользователем этой программы и решение поставленных перед ним задач.

Рассмотрим наиболее типовые подходы при реализации интерфейса.

### Форма список (таблица)

Форма список отображает информацию в виде таблицы (списка). В таком виде обычно отображаются таблицы – справочники, полная или обобщённая информация из таблицы, сгруппированная информация из нескольких таблиц.

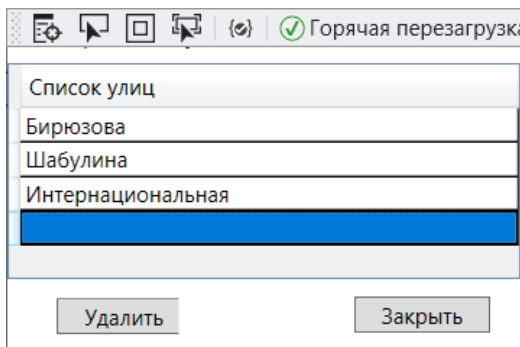


Рисунок 1 - Справочник улицы

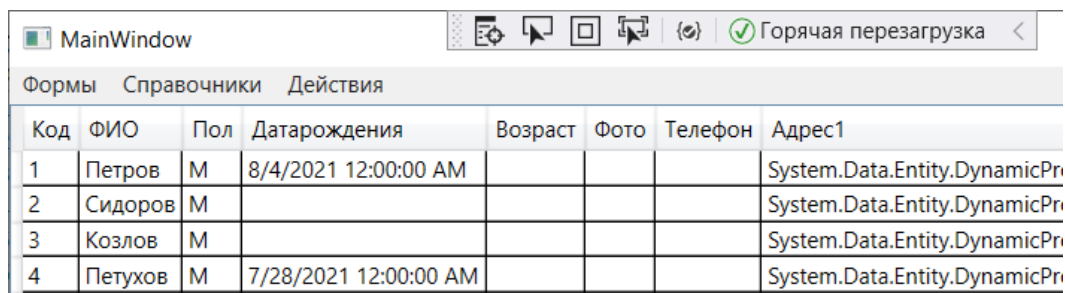


Рисунок 1 - Главная окно программы таблица абонент

### Форма - бланк

Эта форма похожа на анкету. На форме, как правило, располагаются поля одной записи из одной таблице, реже размещается связанная информация с данной записью из нескольких таблиц.

Фамилия  
Сидоров И.П.

Пол  
Муж

Возраст  
Выбор даты 15

Иин  
234567567344

Телефон

Добавить Отмена

Рисунок 2 - Форма – бланк

### Связанные таблицы

На одной форме можно расположить несколько таблиц используя оба описанных выше подхода. Главной задачей является сохранение связи между таблицами при отображении данных.

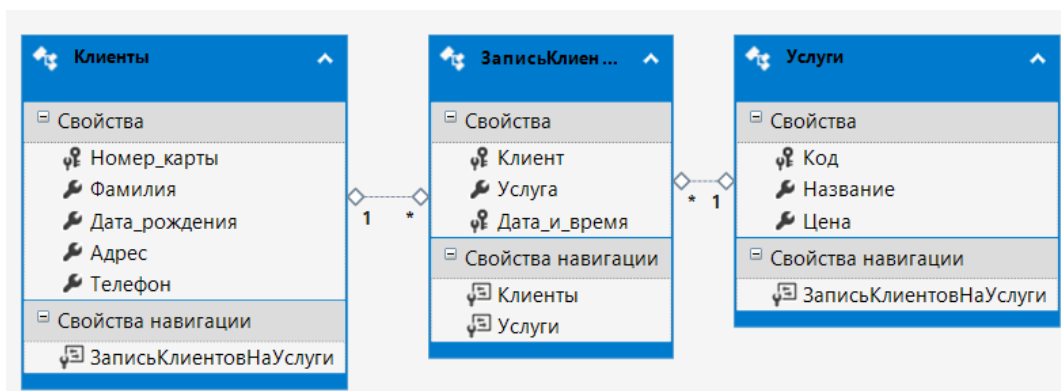
Код	ФИО	Пол	Датарождения	Возраст	Фото	Телефон
1	Петров	М				
2	Сидоров	М				
3	Козлов	М				

Код	Номер	Кодтип_телефона	Кодабоне
1	111111	1	1
2	222222	2	1

Одна из таблиц является главной, вторая подчиненной. Связь между ними, как правило, один ко многим. Одна запись главной таблицы связана с записями нескольким в подчиненной таблице. При отображении записи главной таблицы, подчиненная таблица отображает только те записи, которые связаны с выбранной записью из главной таблицы.

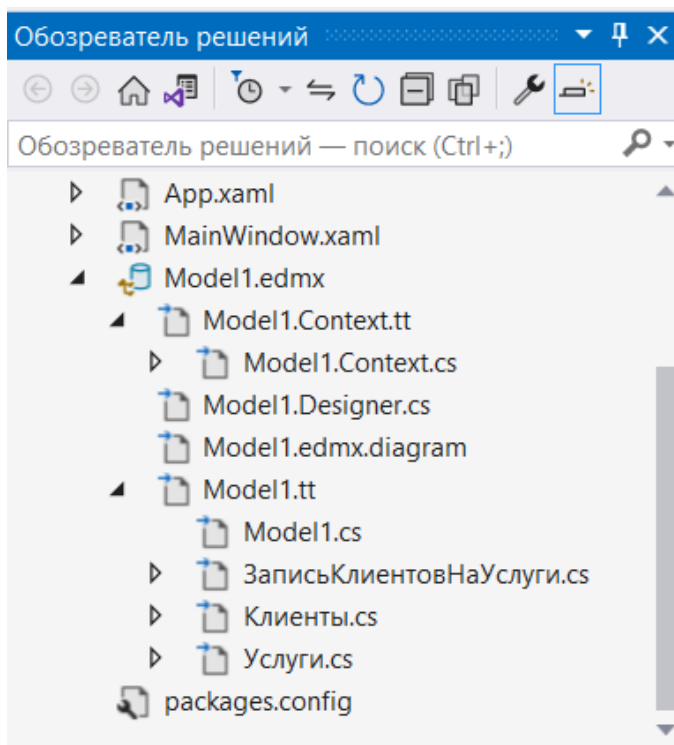
### Создание модели и классов сущностей БД

Подключение к БД осуществляется обычным образом. После подключения будет создана модель БД, которая соответственно содержится в элементе **ModelDataBase.edmx**:



Соответственно модель будет содержать все добавленные таблицы из БД и связи между ними.

В модели будут созданы классы сущностей:



Просмотрим класс сущности *Клиент*, в нем можно увидеть поля сущности и навигационное свойство, которые используются для связи с сущностью *ЗаписьКлиентовНаУслуги*:

```
public partial class Клиенты
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Клиенты()
    {
        this.ЗаписьКлиентовНаУслуги = new HashSet<ЗаписьКлиентовНаУслуги>();
    }

    Ссылка: 1
    public int Номер_карты { get; set; }
    Ссылка: 0
    public string Фамилия { get; set; }
    Ссылка: 0
    public System.DateTime Дата_рождения { get; set; }
    Ссылка: 0
    public string Адрес { get; set; }
    Ссылка: 0
    public string Телефон { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertyShouldBeKeyed")]
    Ссылка: 1
    public virtual ICollection<ЗаписьКлиентовНаУслуги> ЗаписьКлиентовНаУслуги { get; set; }
}
```

Обратите внимание, что навигационное свойство *ЗаписьКлиентовНаУслуги* представляет собой коллекцию связанных элементов, т.к. связь один ко многим.

Посмотрим класс сущности *ЗаписьКлиентовНаУслуги*, в нем можно увидеть поля сущности и навигационные свойства, которые используются для связи с сущностями Клиенты и Услуги:

```
public partial class ЗаписьКлиентовНаУслуги
{
    Ссылка: 1
    public int Клиент { get; set; }
    Ссылка: 1
    public int Услуга { get; set; }
    Ссылка: 1
    public System.DateTime Дата_и_время { get; set; }

    Ссылка: 0
    public virtual Клиенты Клиенты { get; set; }
    Ссылка: 0
    public virtual Услуги Услуги { get; set; }
}
```

Обратите внимание, что навигационные свойства *Клиенты* и *Услуги* представляют собой одиночные ссылки на связанный элемент.

В модели будет создан класс контекста данных для доступа к сущностям:

```
public partial class primerBD1Entities : DbContext
{
    Ссылка: 1
    public primerBD1Entities()
        : base("name=primerBD1Entities")
    {
    }

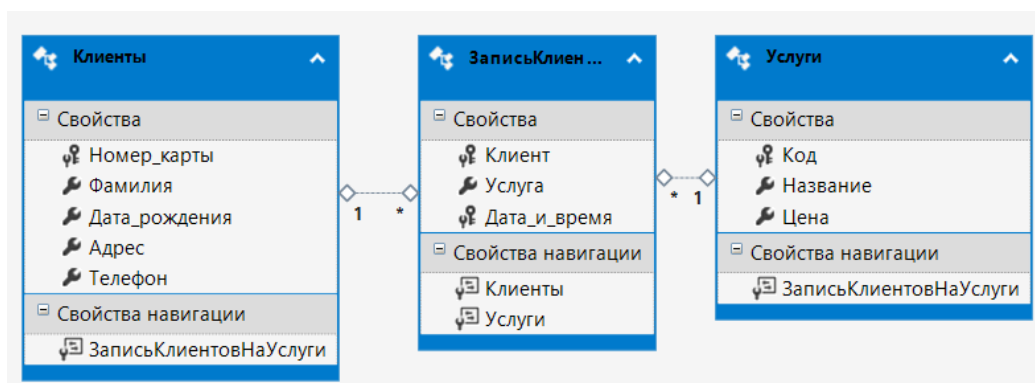
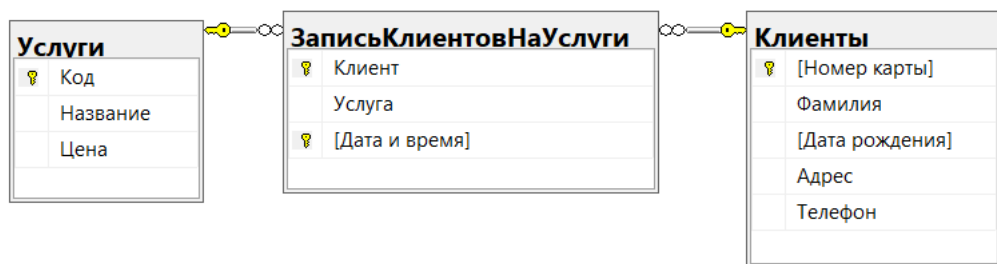
    Ссылка: 0
    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }

    Ссылка: 3
    public virtual DbSet<ЗаписьКлиентовНаУслуги> ЗаписьКлиентовНаУслуги { get; set; }
    Ссылка: 2
    public virtual DbSet<Клиенты> Клиенты { get; set; }
    Ссылка: 2
    public virtual DbSet<Услуги> Услуги { get; set; }

    //Добавляем статичную ссылку на контент
    private static primerBD1Entities context;
    Ссылка: 2
    public static primerBD1Entities GetContext()
    {
        if (context == null) context = new primerBD1Entities();
        return context;
    }
}
```

## Пример разработки и интерфейса БД Услуги

### Схема БД и модель БД Услуги



### Разработка интерфейса главной формы

После изучения БД на главной форме разместим информацию об оказанных услугах: фамилию клиента, телефон, услуга, цена услуги, дата оказания услуги.

Как видно из задания, информация, размещаемая на главной форме, берется из всех трех таблиц, также учтите, что клиент в таблице *ЗаписьКлиентовНаУслуги* представляет собой числовой код, вывести нужно фамилию клиента.

Для достижения поставленной задачи делаем представление, формирующее нужную информацию, и используем его на главной форме, либо при разметке интерфейса используем навигационные свойства для получения информации из связанных таблиц.

Рассмотрим пример разметки интерфейса при использовании элемента DataGrid:

```

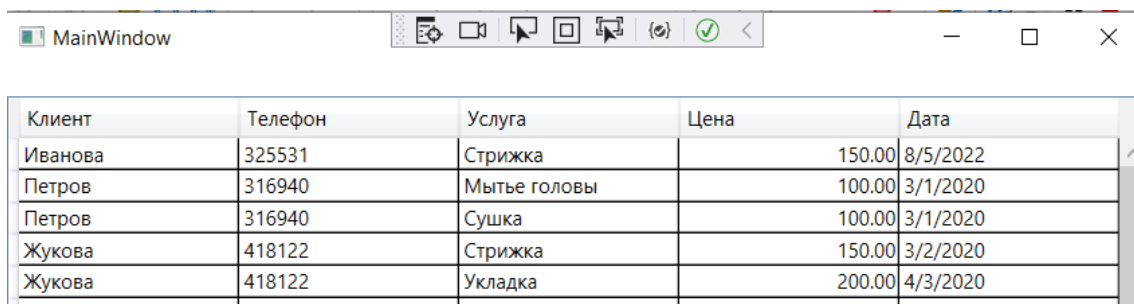
<DataGrid x:Name="dataGrid1" IsReadOnly="True" AutoGenerateColumns="False" Grid
    <DataGrid.Columns>
        <DataGridTextColumn Width="*" Header="Клиент"
            Binding="{Binding Клиенты.Фамилия}"/>
        <DataGridTextColumn Width="*" Header="Телефон"
            Binding="{Binding Клиенты.Телефон}"/>
        <DataGridTextColumn Width="*" Header="Услуга"
            Binding="{Binding Услуги.Название}"/>
        <DataGridTemplateColumn Width="*" Header="Цена">
            <DataGridTemplateColumn.CellTemplate>
                <DataTemplate>
                    <TextBlock Text="{Binding Услуги.Цена,StringFormat=f2}"
                        HorizontalAlignment="Right"/>
                </DataTemplate>
            </DataGridTemplateColumn.CellTemplate>
        </DataGridTemplateColumn>
        <DataGridTextColumn Width="*" Header="Дата"
            Binding="{Binding Дата_и_время,StringFormat=d}"/>
    </DataGrid.Columns>
</DataGrid>
    
```

Для отображения таблицы подключаемся к контексту, загружаем таблицу и получим информацию согласно реализованной разметке **DataGrid**:

```
//Получаем доступ к контексту данных
primerBD1Entities db = primerBD1Entities.GetContext();
```

Ссылка: 1

```
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    //Загружаем таблицу из БД
    db.ЗаписьКлиентовНаУслуги.Load();
    //Загружаем таблицу в DataGrid с отслеживания изменения контекста
    dataGrid1.ItemsSource = db.ЗаписьКлиентовНаУслуги.Local.ToBindingList();
    //Загружаем таблицу в DataGrid без отслеживания изменения контекста
    //dataGrid1.ItemsSource = db.ЗаписьКлиентовНаУслуги.ToList();
}
```



Клиент	Телефон	Услуга	Цена	Дата
Иванова	325531	Стрижка	150.00	8/5/2022
Петров	316940	Мытье головы	100.00	3/1/2020
Петров	316940	Сушка	100.00	3/1/2020
Жукова	418122	Стрижка	150.00	3/2/2020
Жукова	418122	Укладка	200.00	4/3/2020

### Создание окна **Добавить запись**

Окно добавить создается также, как и для однотабличных БД, но в данном случае в таблице **ЗаписьКлиентовНаУслуги** для удобства нужно использовать комбобоксы для выбора клиента по фамилии и услуги по названию, для этого в комбобоксы нужно загрузить соответствующие списки из таблиц. Далее рассмотрен как раз такой пример.

#### Алгоритм действий создания окна **Добавить запись**

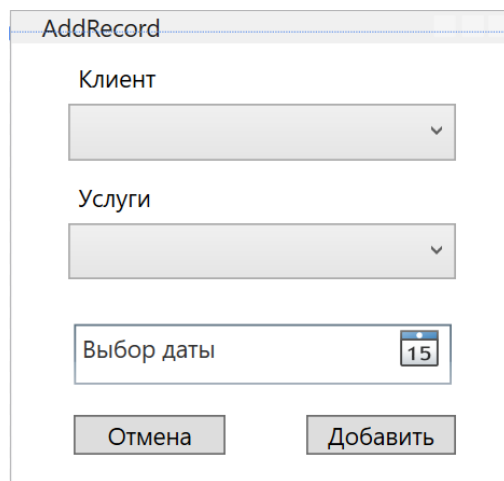
1. В главном окне размещаем компоненты для перехода в окно **ЗаписьКлиентовНаУслуги** и создаем соответствующий код для открытия окна.

```
private void Button_Add(object sender, RoutedEventArgs e)
{
    //Открываем форму Добавить
    AddRecord f = new AddRecord();
    f.ShowDialog();
    dataGrid1.Focus();
}
```

2. Создаем окно **ЗаписьКлиентовНаУслуги** см. пример:

```
<Button Content="Добавить" Click="Btn_Click_Add" ▸
<Button Content="Отмена" Click="Btn_Click_Cancel"
<ComboBox x:Name="cbClient" HorizontalAlignment="L
<ComboBox x:Name="cbUsługa" HorizontalAlignment="L
<DatePicker x:Name="dpDate" HorizontalAlignment="C
<Label Content="Клиент" HorizontalAlignment="Left"
<Label Content="Услуги" HorizontalAlignment="Left"
```





3. При загрузке формы загружаем таблицы Клиент и Услуги и провязываем их к комбобоксам.

```
//Получаем доступ к контексту данных
primerBD1Entities db = primerBD1Entities.GetContext();
ЗаписьКлиентовНаУслуги _record;
Ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    db.Клиенты.Load();
    cbClient.ItemsSource = db.Клиенты.ToList();
    //Так как таблица сложная нужно указать отображаемое поле
    cbClient.DisplayMemberPath = "Фамилия";
    db.Услуги.Load();
    cbUsluga.ItemsSource = db.Услуги.ToList();
    cbUsluga.DisplayMemberPath = "Название";
    dpDate.SelectedDate = DateTime.Now;
}
```

4. Создаем функцию Добавить, для сохранения внесенных данных в БД.

```
private void Btn_Click_Add(object sender, RoutedEventArgs e)
{
    _record = new ЗаписьКлиентовНаУслуги();
    //Проверяем ввод данных
    StringBuilder error = new StringBuilder();
    if (cbClient.SelectedItem == null)
    {
        error.Append("Выберите клиента");
    }
    if (cbUsluga.SelectedItem == null)
    {
        error.AppendLine("Выберите услугу");
    }
    if (error.Length > 0)
    {
        MessageBox.Show(error.ToString());
        return;
    }
}
```

```

        //Получаем выбранную запись
        //Приводит к типу клиента
        //Получаем номер карты – ключевое поле
        _record.Клиент = ((Клиенты)(cbClient.SelectedItem)).Номер_карты;
        _record.Услуга = ((Услуги)(cbUsluga.SelectedItem)).Код;
        _record.Дата_и_время = (DateTime)dpDate.SelectedDate;
        db.ЗаписьКлиентовНаУслуги.Add(_record);
        db.SaveChanges();
        Close();
    }

```

### Создание окна Изменить запись

Окно изменить имеет такой же вид как и окно Добавить запись, поэтому, чтобы не создавать окно заново можно создать копию окна Добавить запись.

### Алгоритм действий создания окна Изменить запись

1. Создаем глобальный класс для передачи ссылки на текущую запись.

```

public static class Data
{
    public static ЗаписьКлиентовНаУслуги Record;
}

```

2. В главном окне размещаем компоненты для перехода в окно *ЗаписьКлиентовНаУслуги* и создаем соответствующий код для открытия окна. Обратите внимание, что мы должны запомнить выбранную для редактирования запись, чтобы изменять именно её.

```

private void btnEdit_Click(object sender, RoutedEventArgs e)
{
    //Получаем ссылку на текущий элемент
    Data.Record = (ЗаписьКлиентовНаУслуги)dataGrid1.SelectedItem;
    //Открываем форму редактировать
    EditRecord winEditRecord = new EditRecord();
    winEditRecord.ShowDialog();
    dataGrid1.Items.Refresh();
    dataGrid1.Focus();
}

```

3. Создаем окно Изменить абонента или лучше делаем копии окна Добавить запись см. пример:

The screenshot shows a window titled 'EditRecord'. Inside, there are two dropdown menus. The first is labeled 'Клиент' and the second is labeled 'Услуги'. Below these is a date picker labeled 'Выбор даты' with the number '15' visible. At the bottom of the window, there are two buttons: 'Отмена' (Cancel) and 'Изменить' (Edit).

4. При загрузке формы загружаем таблицы Клиент и Услуги и привязываем их к комбобоксам. Загружаем информацию о текущей оказанной услуге: клиент, услуга и дата оказание услуги и отображаем всю эту информацию в соответствующих элементах окна.

```
//Получаем доступ к контексту данных
primerBD1Entities db = primerBD1Entities.GetContext();
ЗаписьКлиентовНаУслуги _record;
Ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    db.Клиенты.Load();
    cbClient.ItemsSource = db.Клиенты.ToList();
    //Так как таблица сложная нужно указать отображаемое поле
    cbClient.DisplayMemberPath = "Фамилия";
    db.Услуги.Load();
    cbUsluga.ItemsSource = db.Услуги.ToList();
    cbUsluga.DisplayMemberPath = "Название";
    //Отображаем текущую запись
    _record = Data.Record;
    //Через навигационные свойства задаем клиента и услугу
    cbClient.SelectedItem = _record.Клиенты;
    cbUsluga.SelectedItem = _record.Услуги;
    dpDate.SelectedDate = _record.Дата_и_время;
}
```

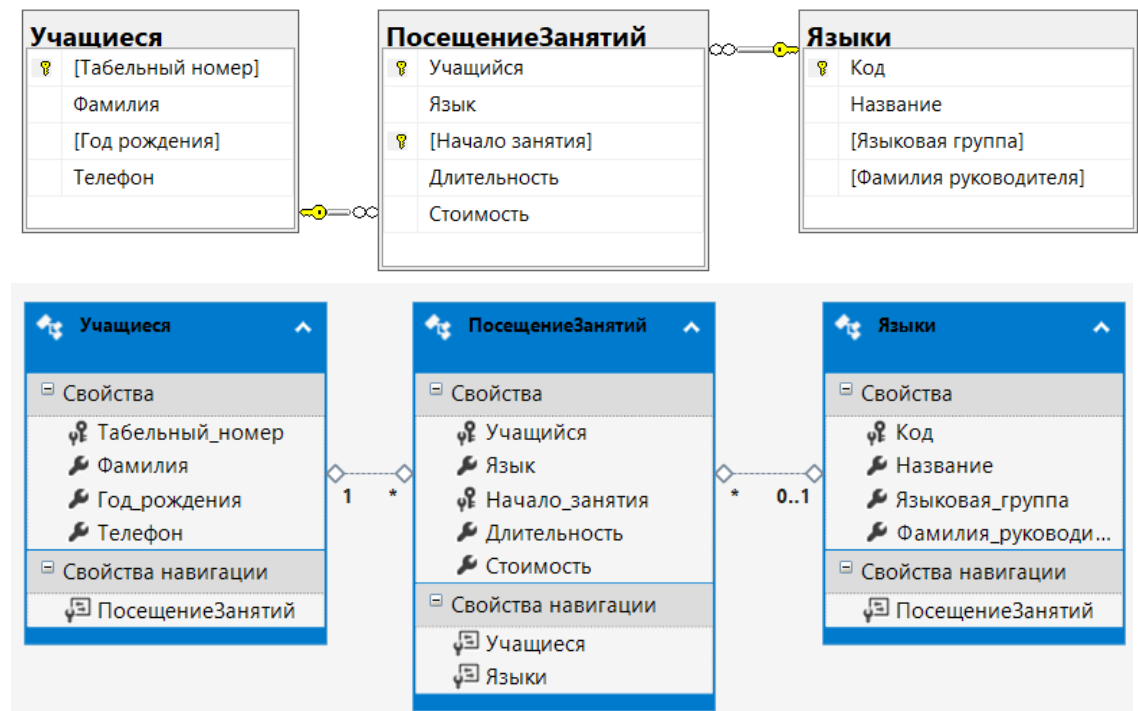
5. Создаем функцию Изменить, для сохранения внесенных данных в БД.

```
private void Btn_Click_Add(object sender, RoutedEventArgs e)
{
    //ЗаписьКлиентовНаУслуги record = new ЗаписьКлиентовНаУслуги();
    StringBuilder error = new StringBuilder();
    //Получаем выбранную запись
    //Приводит к типу клиента
    //Получаем номер карты – ключевое поле
    if (cbClient.SelectedItem == null)
    {
        error.Append("Выберите клиента");
    }
    if (cbUsluga.SelectedItem == null)
    {
        error.AppendLine("Выберите услугу");
    }
    if (error.Length > 0)
    {
        MessageBox.Show(error.ToString());
        return;
    }

    _record.Клиент = ((Клиенты)(cbClient.SelectedItem)).Номер_карты;
    _record.Услуга = ((Услуги)(cbUsluga.SelectedItem)).Код;
    _record.Дата_и_время = (DateTime)dpDate.SelectedDate;
    //db.ЗаписьКлиентовНаУслуги.Add(record);
    db.SaveChanges();
    Close();
}
```

## Пример разработки и интерфейса БД Занятия

### Схема БД и модель БД Занятия



### Разработка интерфейса главной формы

После изучения БД на главной форме разместим информацию об учащихся: табельный номер, фамилию, год рождения, полную сумму, потраченную на обучение.

Как видно из задания, информация, размещаемая на главной форме, берется из двух таблиц, также учтите, что полную стоимость нужно подсчитать, сложив стоимости за каждое занятие.

Для достижения поставленной задачи делаем представление, формирующее нужную информацию, и используем его на главной форме, либо необходимо в сущность Учащиеся добавить свойство для подсчёта суммы, потраченной на обучение.

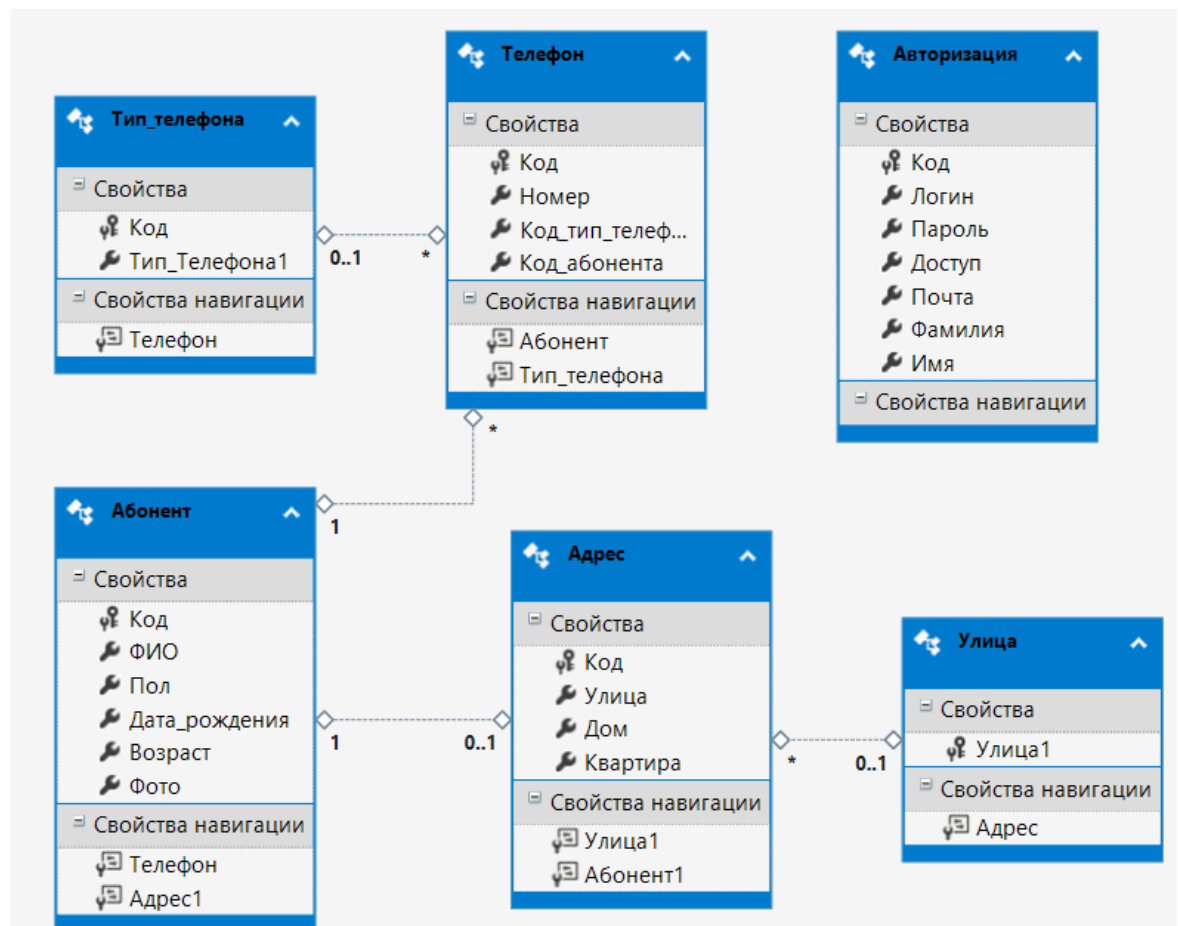
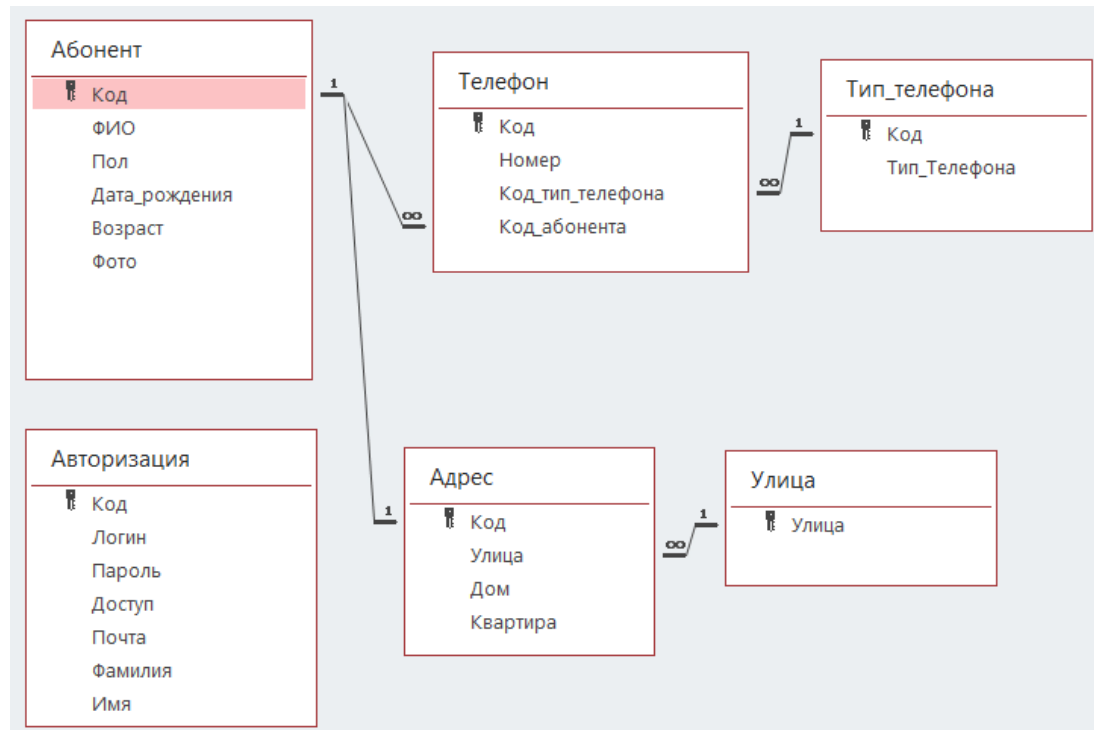
```
public int Сумма {
    get
    {
        int sum = 0;
        foreach (var item in ПосещениеЗанятий)
        {
            if(item.Стоимость != null)
                sum = sum + (int)item.Стоимость;
        }
        return sum;
    }
}
```

decimal? WpfApp1.ПосещениеЗанятий.Стоимость { get; set; }

Так как по сути на главной форме отображена одна таблица, добавление и изменение данных происходит как в однотабличной БД рассмотренной ранее.

## Пример разработки и интерфейса БД Справочник

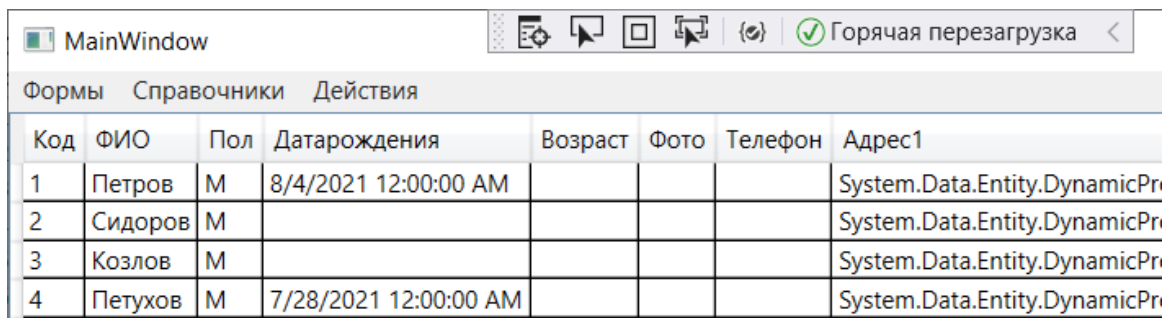
### Схема БД и модель БД Справочник



### Разработка интерфейса главной формы

После анализа текущей БД решаем, что главная информация – это информация об абоненте и эту таблицу будем размещать на главной форме.

Создаем интерфейс главного окна, где размещаем таблицу абонент, см. рисунок ниже.



The screenshot shows a Windows application window titled 'MainWindow'. It contains a table with 8 columns: 'Код' (Code), 'ФИО' (Full Name), 'Пол' (Gender), 'Датарождения' (Date of Birth), 'Возраст' (Age), 'Фото' (Photo), 'Телефон' (Phone), and 'Адрес1' (Address 1). The table has 4 rows of data. The first row shows '1', 'Петров', 'М', '8/4/2021 12:00:00 AM', and the last column contains 'System.Data.Entity.DynamicPr'. The second row shows '2', 'Сидоров', 'М', and the last column contains 'System.Data.Entity.DynamicPr'. The third row shows '3', 'Козлов', 'М', and the last column contains 'System.Data.Entity.DynamicPr'. The fourth row shows '4', 'Петухов', 'М', '7/28/2021 12:00:00 AM', and the last column contains 'System.Data.Entity.DynamicPr'. Above the table is a menu bar with 'Формы', 'Справочники', and 'Действия'. To the right of the menu bar is a toolbar with icons for settings, navigation, and a 'Горячая перезагрузка' (Hot Reload) button.

Код	ФИО	Пол	Датарождения	Возраст	Фото	Телефон	Адрес1
1	Петров	М	8/4/2021 12:00:00 AM				System.Data.Entity.DynamicPr
2	Сидоров	М					System.Data.Entity.DynamicPr
3	Козлов	М					System.Data.Entity.DynamicPr
4	Петухов	М	7/28/2021 12:00:00 AM				System.Data.Entity.DynamicPr

В данном примере разметка таблицы не проводилась, проведете ее самостоятельно. Пример программного кода приведен ниже:

```
СправочникEntities db = СправочникEntities.GetContext();
ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    //Занружаем таблицу и отображаем ее с отслеживанием контекста
    //Подключите библиотеку using System.Data.Entity;
    db.Абонент.Load();
    dataGrid1.ItemsSource = db.Абонент.Local.ToBindingList();
    //Или делаем тоже самое без отслеживания контекста
    //dataGrid1.ItemsSource = db.Абонент.ToList();
}
```

### Создание окна Добавить запись

Окно добавить создается также как и для однотабличных БД, но в данном случае у абонента есть адрес и телефоны. Поэтому можно создать более сложное окно, в котором можно сразу заполнить всю информацию для нового абонента. Далее рассмотрен как раз такой пример.

#### Алгоритм действий создания окна Добавить запись

1. В главном окне размещаем компоненты для перехода в окно добавить Абонента и создаем соответствующий код для открытия окна.

```
private void miAddRecord_Click(object sender, RoutedEventArgs e)
{
    Windows.AddRecord ar = new Windows.AddRecord();
    ar.Owner = this;
    ar.ShowDialog();
}
```

## 2. Создаем окно Добавить абонента см. пример:

```

<Grid.RowDefinitions>
    <RowDefinition Height="1*" />
    <RowDefinition Height="1*" />
    <RowDefinition Height="1*" />
    <RowDefinition Height="40" />
</Grid.RowDefinitions>
<GroupBox Header="Основная информация" HorizontalAlignment="Stretch" Grid.Row=
    <Grid>
        <Label Content="Код" HorizontalAlignment="Left" Margin="10,10,0,0" Ver
        <TextBox x:Name="txtId" IsEnabled="False" HorizontalAlignment="Left" H
        <Label Content="ФИО" HorizontalAlignment="Left" Margin="97,10,0,0" Ver
        <TextBox x:Name="tbFIO" HorizontalAlignment="Left" Height="23" Margin=
        <Label Content="Дата рождения" HorizontalAlignment="Left" Margin="10,5
        <DatePicker Name="dpDataR" HorizontalAlignment="Left" Margin="135,55,
        <Label Content="Пол" HorizontalAlignment="Left" Margin="361,10,0,0" Ve
        <ComboBox x:Name="cbGender" HorizontalAlignment="Left" Margin="396,11,
            <ComboBoxItem Content="М" HorizontalAlignment="Left" Width="92"/>
            <ComboBoxItem Content="Ж" HorizontalAlignment="Left" Width="92"/>
        </ComboBox>
        <Label Content="Возраст" HorizontalAlignment="Left" Margin="361,55,0,0
    </Grid>
</GroupBox>

```

Для адреса самый интересный элемент список комбобокс для выбора улицы из таблицы Улица. К этому элементу через свойство `ItemsSource` в коде программы привяжем таблицу Улица. В разметке XAML используя свойство `DisplayMemberPath`, укажем какое поле, будет отображаться в комбобоксе.



```

<GroupBox Header="Адрес" HorizontalAlignment="Stretch" Grid.Row="1">
    <Grid>
        <Label Content="Улица" HorizontalAlignment="Left" Margin="32,20,0,0">
            <ComboBox x:Name="cbStreet" HorizontalAlignment="Left" Margin="83,
                ✓ DisplayMemberPath="Улица1"/>
        <Label Content="Дом" HorizontalAlignment="Left" Margin="32,60,0,0">
            <TextBox x:Name="tbNumberBuilding" HorizontalAlignment="Left" Height="20">
        <Label Content="Квартира" HorizontalAlignment="Left" Margin="192,
            <TextBox x:Name="tbNumberApart" HorizontalAlignment="Left" Height="20">
        </Grid>
    </GroupBox>

```

3. При загрузке формы загружаем таблицу Улица и привязываем ее к комбобоксу для выбора улиц из списка и отключаем кнопку Добавить телефон, т.к. к несуществующему абоненту добавлять телефон нельзя.

```

СправочникEntities db = СправочникEntities.GetContext();
ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    btnAddPhone.IsEnabled = false;
    db.Улица.Load();
    cbStreet.ItemsSource = db.Улица.Local.ToBindingList();
}

```

4. Создаем функцию сохранить (Save), для сохранения внесенных данных в БД. Эта функция работает как на добавление записи, так и на обновление при повторном нажатии кнопки сохранить.

```

Абонент _abonent; //Глобальная объект абонент
Адрес _adress; //Глобальная объект адрес
//Функция сохранения
Ссылка: 2
void Save() {
    //Проверяем данные см. прошлую лекцию

    //Если запись новая, то добавляем ее
    if (_abonent == null)
    {
        //Сохраняем информацию главной таблицы Абонент
        _abonent = new Абонент();
        _abonent.ФИО = tbFIO.Text;
        _abonent.Пол = cbGender.Text;
        _abonent.Дата_рождения = dpDataR.SelectedDate;
        _db.Абонент.Add(_abonent);
        _db.SaveChanges();
        //Отображаем код абонента на форме
        txtId.Text = _abonent.Код.ToString();
        btnAddPhone.IsEnabled = true;
    }
}

```



```
else//Иначе обновляем информацию
{
    _abonent.ФИО = tbFIO.Text;
    _abonent.Пол = cbGender.Text;
    _abonent.Дата_рождения = dpDataR.SelectedDate;
    _db.SaveChanges();
}
//Если запись новая, то добавляем ее
if (_address == null)
{
    if (cbStreet.Text.Length != 0 ||
        tbNumberApart.Text.Length != 0 || tbNumberBuilding.Text.Length != 0)
    {
        //Сохраняем адрес
        _address = new Адрес();
        _address.Код = _abonent.Код;
        _address.Улица = cbStreet.Text;
        _address.Дом = tbNumberBuilding.Text;
        _address.Квартира = Convert.ToInt32(tbNumberApart.Text);
        _db.Адрес.Add(_address);
        _db.SaveChanges();
    }
}
else//Иначе обновляем информацию
{
    //Сохраняем адрес
    _address.Улица = cbStreet.Text;
    _address.Дом = tbNumberBuilding.Text;
    _address.Квартира = Convert.ToInt32(tbNumberApart.Text);
    _db.SaveChanges();
}
}
```

5. Реализуем кнопку Сохранить.

```
private void btnSave_Click(object sender, RoutedEventArgs e)
{
    Save();
}
```

6. Реализуем кнопку ОК.

```
private void btn_OK(object sender, RoutedEventArgs e)
{
    Save();
    Close();
}
```

7. Реализуем форму Добавить номер. К списку комбобокс Тип телефона., через свойство ItemsSource в коде программы привяжем таблицу Тип телефона. В разметке XAML используя свойство **DisplayMemberPath**, укажем какое поле, будет отображаться в комбобоксе

8. При загрузке окна Добавить телефон, к комбобоксу Тип телефона привязываем соответствующую таблицу.

```
СправочникEntities db = СправочникEntities.GetContext();
//ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    cbTipPhone.ItemsSource = db.Тип_телефона.ToList();
}
```

9. Реализуем кнопку ОК в окне Добавить телефон.

```
private void btnOK_Click(object sender, RoutedEventArgs e)
{
    Телефон phone = new Телефон();
    phone.Номер = tbPhone.Text;
    //указываем тип телефона через навигационную связь типа телефона
    //Тип_телефона typePhone = (Тип_телефона)(cbTipPhone.SelectedItem);
    //phone.Тип_телефона = typePhone;
    //Указываем тип телефона через код выбранного типа телефона
    phone.Код_тип_телефона = ((Тип_телефона)(cbTipPhone.SelectedItem)).Код;
    phone.Код_абонента = Data.IdAbonetn;
    db.Телефон.Add(phone);
    db.SaveChanges();
    Close();
}
```

10. Реализуем кнопку Добавить телефон в окне Добавление абонента.

```
private void btnAddPhone_Click(object sender, RoutedEventArgs e)
{
    //Получаем код абонента
    Data.IdAbonetn = _abonent.Код;
    //Добавляем телефон
    Windows.AddPhone p = new AddPhone();
    p.Owner = this;
    p.ShowDialog();
    //Отображаем телефоны
    //var phone = from p1 in db.Телефон where p1.Код_абонента == ab1.Код select p1;
    var phone = from p1 in _db.Телефон where p1.Код_абонента == _abonent.Код select new {
        Phone = p1.Номер, TipPhone = p1.Тип_телефона.Тип_Телефона1};
    DGPhone.ItemsSource = phone.ToList();
}
```

11. Для упрощения обновления информации в главной таблице делаем это при активации главного окна или при закрытии окна добавить/изменить.

```
private void Window_Activated(object sender, EventArgs e)
{
    dataGrid1.Items.Refresh();
    //Перезагружаем сущности если контекст не отслеживается
    //db.ChangeTracker.Entries().ToList().ForEach(p => p.Reload());
    //dataGrid1.ItemsSource = db.Абонент.ToList();
}
```

### Создание окна Изменить запись

Окно изменить имеет такой же вид как и окно Добавить запись, поэтому, чтобы не создавать окно заново можно создать копию окна Добавить запись.

### Алгоритм действий создания окна Изменить запись

1. В главном окне размещаем компоненты для перехода в окно изменить Абонента и создаем соответствующий код для открытия окна. Обратите внимание, что мы должны запомнить код выбранной для редактирования записи, чтобы изменять именно её.

```
private void miEditRecord_Click(object sender, RoutedEventArgs e)
{
    //Получаем код изменяемой записи
    var row = (Абонент)dataGrid1.SelectedItem;
    //и запоминаем в глобальной переменной
    Data.IdAbonetn = row.Код;
    Windows.EditRecord winEditRecord = new Windows.EditRecord();
    winEditRecord.Owner = this;
    winEditRecord.ShowDialog();
}
```

2. Создаем окно Изменить абонента или лучше делаем копии окна Добавить запись см. пример:

Номер телефона	Тип телефона
111111	Мобильный
222222	Домашний

Для лучшего отображения списка телефонов делаем разметку таблицы для отображения телефона и типа телефона. Обратите внимание, чтобы не отображать код Типа телефона, через навигационное свойство получаем название типа телефона.

```
<GroupBox Header="Телефоны" HorizontalAlignment="Stretch" Grid.Row="2">
    <DataGrid x:Name="DGPhone" AutoGenerateColumns="False" IsReadOnly="True">
        <DataGrid.Columns>
            <DataGridTextColumn Header="Номер телефона"
                Binding="{Binding Номер}" Width="1*"/>
            <DataGridTextColumn Header="Тип телефона"
                Binding="{Binding Тип_телефона.Тип_Телефона1}" Width="1*"/>
        </DataGrid.Columns>
    </DataGrid>
</GroupBox>
```

3. При загрузке формы загружаем таблицу Улица и провязываем ее к комбобоксу для выбора улиц из списка. Загружаем информацию о текущем абоненте, его адресе, его список телефонов и отображаем всю эту информацию в соответствующих элементах окна.

```
СправочникEntities _db = СправочникEntities.GetContext();
Абонент _abonent; //Глобальная объект абонент
Адрес _adress; //Глобальная объект абонент
Ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    _db.Улица.Load();
    cbStreet.ItemsSource = _db.Улица.Local.ToBindingList();
    //Загружаем информацию об абоненте
    _abonent = _db.Абонент.Find(Data.IdAbonetn);
    tbId.Text = _abonent.Код.ToString();
    tbFIO.Text = _abonent.ФИО;
    cbGender.Text = _abonent.Пол;
    dpDataR.SelectedDate = _abonent.Дата_рождения;
    //Загружаем информацию об адресе
    _adress = _db.Адрес.Find(Data.IdAbonetn);
    if (_adress != null)
    {
        cbStreet.Text = _adress.Улица;
        tbNumberBuilding.Text = _adress.Дом;
        tbNamberApart.Text = _adress.Квартира.ToString();
    }
    //Отображаем телефоны
    //_db.Телефон.Load();
    //DGPhone.ItemsSource = _db.Телефон.Local.ToBindingList().
    //    Where(p => p.Код_абонента == _abonent.Код);
    //Отображаем телефон используя навигационное свойство
    DGPhone.ItemsSource = _abonent.Телефон.ToList();
}
```

4. Создаем функцию сохранить (Save), для сохранения внесенных данных в БД.

Эта функция не изменяется!

## 5. Реализуем кнопку Сохранить.

Эта функция не изменяется!

## 6. Реализуем кнопку ОК.

Эта функция не изменяется!

## 7. Реализуем форму Добавить номер.

Форма уже добавлена!

## 10. Реализуем кнопку Добавить телефон в окне Добавление абонента.

Эта функция не изменяется!

### Удаления записи из таблицы Абонент

Для данной БД для любого абонента информация об адресе и телефонах является его личной информацией, поэтому при удалении абонента должна также удалиться связанная с ним информация.

Для достижения этой цели при проектировании БД, при создании связи между этими таблицами нужно было включить каскадное удаления, в противном случае удалить абонента при наличии связанной информации будет невозможно.

```
private void DeleteRecord_Click(object sender, RoutedEventArgs e)
{
    //Получаем текущую строку
    var row = (Абонент)dataGrid1.SelectedItem;
    db.Абонент.Remove(row);
    db.SaveChanges();
    dataGrid1.Items.Refresh();
    //Перезагружаем сущности если контекст не отслеживается
    //db.ChangeTracker.Entries().ToList().ForEach(p => p.Reload());
    //dataGrid1.ItemsSource = db.Абонент.ToList();
}
```

## Создание связанных таблиц

На одной форме можно расположить несколько таблиц. Главной задачей является сохранение связи между таблицами при отображении данных.

Код	ФИО	Пол	Датарождения	Возраст	Фото	Телефон
1	Петров	M				
2	Сидоров	M				
3	Козлов	M				

Одна из таблиц является главной, вторая подчиненной. Связь между ними, как правило, один ко многим. Одна запись главной таблицы связана с записями несколькими в подчиненной таблице. При отображении записи главной таблицы, подчиненная таблица отображает только те записи, которые связаны с выбранной записью из главной таблицы.

В рассмотренном окне есть главная таблица Абонент и подчиненная таблица телефоны.

### **Алгоритм создания связанных таблиц.**

1. Размещаем в окне программы две таблицы, при необходимости делаем нужную разметку.

2. При загрузке формы отображаем информацию в таблицах.

```
СправочникEntities _db = СправочникEntities.GetContext();
Ссылка: 1
private void Window_Loaded(object sender, RoutedEventArgs e)
{
    //Отображаем таблицу Абонент
    _db.Абонент.Load();
    DGAbonent.ItemsSource = _db.Абонент.Local.ToBindingList();
    //Выделяем первую запись
    DGAbonent.SelectedItem = DGAbonent.Items[0];
    //Получаем первую запись таблицы Абонент
    Абонент abonent = _db.Абонент.FirstOrDefault();
    //Получаем телефоны указанного абонента
    //var phone = from p in db.Телефон where p.Код_абонента == abonent.Код select p;
    //DGPhone.ItemsSource = phone.ToList();
    //Либо используем навигационное свойство
    DGPhone.ItemsSource = abonent.Телефон.ToList();
}
```

3. При смене абонента изменяется список его телефонов.

```
//Смена выделения ячеек
Ссылка: 1
private void DGAbonent_SelectedCellsChanged(object sender, Sele
{
    //Получаем текущую запись таблицы Абонент
    Абонент abonent = (Абонент)DGAbonent.SelectedItem;
    //Получаем телефоны указанного абонента
    DGPhone.ItemsSource = abonent.Телефон.ToList();
}
```

### **Использование LINQ to Entities**

При работе с многотабличной БД можно также формировать запросы к БД используя LINQ. Для формирования запроса с использованием нескольких таблиц также как и в обычном запросе используется оператор **JOIN**.

Например, в рассмотренной БД справочник хотим вывести список абонентов с их адресами.

**В обычном SQL это будет такой запрос:**

```
SELECT Абонент.Код, Абонент.ФИО, Абонент.Пол, Абонент.[Дата_рождения],
Адрес.Улица, Адрес.Дом, Адрес.Квартира
FROM (Абонент INNER JOIN Адрес ON Абонент.Код = Адрес.[Код_абонента])
```

***В LINQ это будет такой запрос:***

```
var zp1 = from p in db.Абонент join c in db.Адрес on p.Код equals c.Код
select new {p.Код, p.ФИО, p.Пол, p.Дата_рождения,
            c.Улица, c.Дом, c.Квартира};
```

***В обычном SQL это будет такой запрос:***

```
SELECT Абонент.Код, Абонент.ФИО, Абонент.Пол, Абонент.[Дата_рождения],
Адрес.Улица, Адрес.Дом, Адрес.Квартира
FROM (Абонент LEFT OUTER JOIN Адрес ON Абонент.Код =
Адрес.[Код_абонента])
```

***В LINQ это будет такой запрос:***

```
var zp2 = from p in db.Абонент
join c in db.Адрес on p.Код equals c.Код into ps
from x in ps.DefaultIfEmpty()
select new
{
    p.Код, p.ФИО, p.Пол, p.Дата_рождения,
    x.Улица, x.Дом, x.Квартира
};
```



## Создание формы регистрации пользователя

Рассмотрим последовательность действий и код программы для создания формы регистрации. Регистрацию осуществляет администратор и вводятся логин, пароль, имя, фамилия, адрес электронной почты, режим доступа пользователя см. рисунок 12. Информация о пользователях храниться в таблице *Авторизация*. При регистрации запоминаем уровень пользователя А – администратор, П – пользователь.

Также необходимо проверить правильность ввода адреса электронной почты и сложность пароля:

- длина пароля должна быть не менее 8 символов;
- в числе символов пароля обязательно должны присутствовать:
  1. Буквы строчные(a, b, c, d, e, f и т.д.);
  2. Буквы заглавные(A, B, C, D, E, F и т.д.);
  3. Цифры(0, 1, 2, 3, 4 и т.д.);
  4. Спецсимволы(@, #, \$, &, \*, % и т.д.);

Рисунок 12. Пример формы регистрации

## Алгоритм создания формы регистрации

1. Создаем форму регистрации см. рисунок 12. Для доступа к таблице *Авторизация* через *Панель элементов* добавьте компоненты для работы с БД.

авторизацияTableAdapter1

2. На форме1 при открываем форму регистрация.

```
private void РегистрацияПользователяToolStripMenuItem_Click(
{
    //Открываем форму
    registrsia fr = new registrsia();
    fr.ShowDialog();
}
```



## 7. На форме *Регистрация* пишем код кнопки *Регистрация*.

```
private void РегистрацияBTN_Click(object sender, EventArgs e)
{
    //Примеры проверки полей
    if (фамилияТВ.Text.Length == 0 || имяТВ.Text.Length == 0)
    { MessageBox.Show("Заполните все поля"); return; }
    if (фамилияТВ.Text.Length == 0) { MessageBox.Show("Введите фамилию"); return; }
    if (имяТВ.Text.Length == 0) { MessageBox.Show("Введите имя"); return; }
    if (доступСВ.Text != "Администратор" && доступСВ.Text != "Пользователь")
    { MessageBox.Show("Уровень доступа"); return; }
    if (логинТВ.Text.Length == 0) { MessageBox.Show("Введите Логин"); return; }
    if (парольТВ.Text.Length < 8)
    { MessageBox.Show("Введите пароль не менее 8 символов"); return; }
    if (почтаТВ.Text.Length == 0) { MessageBox.Show("Введите почту"); return; }
    //Проверка адреса
    try
    {
        var addr = new System.Net.Mail.MailAddress(почтаТВ.Text);
    }
    catch { MessageBox.Show("Введите правильный почтовый адрес"); return; }
    //Проверка сложности пароля
    int p1,p2,p3,p4,p5;
    p1 = p2 = p3 = p4 = p5 = 0;
    for (int i = 0; i < парольТВ.Text.Length; i++)
    {
        if (Char.IsLetter(парольТВ.Text[i]) && Char.IsLower(парольТВ.Text[i])) p1 = 1;
        if (Char.IsLetter(парольТВ.Text[i]) && Char.IsUpper(парольТВ.Text[i])) p2 = 1;
        if (Char.IsDigit(парольТВ.Text[i])) p3 = 1;
        if (Char.IsPunctuation(парольТВ.Text[i])) p4 = 1;
        if (Char.IsSeparator(парольТВ.Text[i])) p5 = 1;
        //Можно вместо p4 и p5 Проверить наличие конкретных символов
        char[] s = { '!', '@', '#', '$', '%', '^' };
        if (парольТВ.Text.IndexOfAny(s) != -1) p5 = 1;
    }
    if (p1 != 1 || p2 != 1 || p3 != 1 || p4 != 1 || p5 != 1)
    {
        MessageBox.Show("Сложность пароля не соответствует заданным критериям");
        return;
    }

    //Вставка записи в БД
    авторизацияTableAdapter1.Insert(логинТВ.Text, парольТВ.Text, доступСВ.Text.Substring(0,1),
        почтаТВ.Text, фамилияТВ.Text, имяТВ.Text);
    Close();
}
```

## ***Практическая работа № 20***

### **Создание приложения с многотабличной БД**

1. Спроектировать и создать многотабличную БД в SQL server. Для удобства дальнейшей разработки заполните БД несколькими связанными записями (5-10 в зависимости от логики важности информации).
2. Разработать интерфейс программы Visual Studio C#, который содержит:
  - Главную форму.
  - Формы справочников. Для форм справочников реализовать операции добавления, удаления записей.
3. Разработать SQL запросы с участием одной или нескольких таблиц по заданию. Использовать представления или LINQ.
4. Реализовать функцию Поиск

## Варианты заданий 1 Подгруппа

### **Вариант 1**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия, Разряд, Цех

**Таблица 2.** Справочник тарифов.

Структура таблицы: Разряд, Тариф (руб./ час.)

**Таблица 3.** Табель.

Структура таблицы: Табельный номер, Отработанное время в часах, Номер месяца

**Таблица 4.** Список цехов.

Структура таблицы: Цех, Наименование цеха.

**Задание 2.** С помощью SQL-запроса сформировать список работников цеха. Список должен содержать поля: Наименование цеха, Фамилия, Разряд, Тариф.

**Задание 3.** Создать SQL-запрос для подсчета количества работников, работающих в одном цехе, который должен содержать поля: Цех, Количество работников.

### **Вариант 2**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник цехов-изготовителей.

Структура таблицы: Код цеха, Наименование цеха

**Таблица 2.** Справочник характеристик продукции.

Структура таблицы: Код продукции, Наименование продукции, Единица измерения, Код цеха-изготовителя, Код группы продукции, Цена (рублей за единицу)

**Таблица 3.** План выпуска продукции в натуральном выражении.

Структура таблицы: Код продукции, Количество, Номер месяца

**Таблица 4.** Группа продукции.

Структура таблицы: Код группы продукции, Наименование группы продукции.

**Задание 2.** Создать SQL-запрос для подсчета количества наименований продукции, входящих в одну группу. Запрос должен содержать поля: Группа продукции, количество наименований продукции.

**Задание 3.** С помощью SQL-запроса сформировать План выпуска продукции за месяц в стоимостном выражении, включив в запрос поля: Наименование группы продукции, Наименование продукции, Номер месяца, Стоимость.

**Вариант 3**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник окладов.

Структура таблицы: Должность, Оклад (руб./мес.)

**Таблица 2.** Штатное расписание.

Структура таблицы: Табельный номер, Фамилия И.О., Должность, Отдел

**Таблица 3.** Табель учета отработанного времени.

Структура таблицы: Табельный номер, Отработанное время в днях, Номер месяца

**Таблица 4.** Справочник отделов.

Структура таблицы: Отдел, Наименование отдела, Начальник

**Таблица 5.** Структура предприятия.

Структура таблицы: Структурное подразделение, Отдел.

**Задание 2.** Создать SQL-запрос для подсчета количества сотрудников, работающих в одном отделе. Запрос должен содержать поля: Отдел, Количество сотрудников.

**Задание 3.** С помощью SQL-запроса вывести фамилии сотрудников, отработавших целый месяц.

**Вариант 4**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** План выпуска.

Структура таблицы: Код изделия, Количество изделий, Код цеха

**Таблица 4.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей.

**Считается, что в одном цехе может собираться несколько изделий.**

**Таблица 5.** Цех.

Структура таблицы: Код цеха, Наименование цеха, Начальник.

**Задание 2.** С помощью SQL-запроса подсчитать какое количество различных деталей, входит в изделие. Запрос должен содержать поля: Наименование изделия, Количество деталей из которых состоит изделие.

**Задание 3.** С помощью SQL-запроса подсчитать стоимость каждого изделия. Запрос содержит поля: Наименование изделия, Стоимость изделия.

**Вариант 5**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код вида работ, Наименование вида работ, Единица измерения

**Таблица 2.** Справочник расценок.

Структура таблицы: Код работы, Код вида работ, Наименование работы, Расценка (рублей за единицу измерения)

**Таблица 3.** Справочник строительных объектов.

Структура таблицы: Код объекта, Наименование объекта, Населенный пункт, Адрес объекта, Дата начала работ

**Таблица 4.** Объемы работ на объекте.

Структура таблицы: Код объекта, Код работы, Физический объем работы (в единицах измерения)

**Таблица 5.** Справочник окончания работ.

Структура таблицы: Код объекта, Дата окончания работ.

**Задание 2.** С помощью SQL-запроса подсчитать количество объектов в одном городе.

Запрос содержит поля: Населенный пункт, Количество.

**Задание 3.** Создать SQL-запрос для определения итоговой стоимости всех работ одного вида. Запрос должен содержать поля: Наименование вида работ, Итоговая стоимость

**Вариант 6**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник расценок.

Структура таблицы: Код работы, Наименование работы, Расценка (руб. за единицу измерения)

**Таблица 2.** Справочник бригад.

Структура таблицы: Код бригады, Табельный номер, Фамилия И.О.

**Таблица 3.** Наряды.

Структура таблицы: Номер наряда, Код бригады, Табельный номер, Код работы, Выполненный объем (в единицах измерения), КТУ (коэффициент трудового участия может иметь значения от 0 до 1)

**Таблица 4.** Справочник бригадиров.

Структура таблицы: Табельный номер, Фамилия И.О.

**Задание 2.** С помощью SQL-запроса подсчитать количество членов одной бригады.

Запрос должен содержать поля: Код бригады, Количество членов бригады.

**Задание 3.** Составить SQL-запрос для просмотра сведений о бригадах. Запрос должен содержать поля: Фамилия бригадира, Код бригады, Объем работ бригады

**Вариант 7**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей в изделии

**Таблица 4.** План выпуска.

Структура таблицы: Код изделия, План выпуска (количество изделий).

**Задание 2.** С помощью SQL-запроса просмотреть сведения о тех изделиях, наименования которых начинаются с указанной буквы алфавита.

**Задание 3.** С помощью SQL-запроса подсчитать стоимость всех деталей, входящих в изделие и стоимость изделия. Запрос должен содержать поля: Наименование изделия, Стоимость деталей, Стоимость изделия. Стоимость деталей рассчитывается по формуле:

$$\text{стоим.деталей} = \sum_{i=1}^n \text{цена} * \text{количество},$$

где  $i$  – номер детали, входящей в изделие, а  $n$  – количество деталей.

$$\text{стоимость изделия} = \text{стоимость деталей} + \text{стоимость сборки}$$

**Вариант 8**

**Задание 1.** Создать таблицы:

**Таблица 1.** Движение.

Структура таблицы: Код детали, Наименование детали, Количество

**Таблица 2.** Накладная на поступление.

Структура таблицы: Номер накладной, Код детали, Количество поступило, Дата прихода

**Таблица 3.** Требование на выдачу.

Структура таблицы: Номер требования, Код детали, Количество выдано, Дата выдачи

**Таблица 4.** Справочник цен.

Структура таблицы: Код детали, Цена

**Задание 2.** С помощью SQL-запроса подсчитать количество деталей, которые получены складом в течение одного месяца. Запрос должен содержать поля: Наименование детали, Количество.

**Задание 3.** Создать SQL-запрос для определения стоимости всех деталей, отпущенных по одному требованию. Запрос должен содержать поля: Номер требования, Стоимость деталей.

**Вариант 9**

**Задание 1.** Создать таблицы:

**Таблица 1:** Справочник групп товаров.

Структура таблицы: Код группы, Наименование группы

**Таблица 2.** Ценник.

Структура таблицы: Код товара, Наименование товара, Код группы, Ед. измерения, Цена

**Таблица 3.** Поступления.

Структура таблицы: Номер накладной, Код товара, Количество поступило, Дата поступления, Код поставщика

**Таблица 4.** Продажи.

Структура таблицы: Номер чека, Код товара, Количество продано, Дата продажи

**Таблица 5.** Остаток.

Структура таблицы: Код товара, Остаток на конец недели

**Таблица 6.** Поставщик.

Структура таблицы: Код поставщика, Наименование поставщика, Адрес, Телефон

**Задание 2.** Составить SQL-запрос для получения сведений о выполненных поставках.

Запрос должен содержать поля: Код поставщика и его наименование, Наименование товара, Дата поступления, Количество поступило.

**Задание 3.** С помощью SQL-запроса подсчитать общее количество проданных товаров за месяц. Запрос должен содержать поля: Наименование товара, Всего продано.

**Вариант 10**

**Задание 1.** Создать таблицы:

**Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 2.** С помощью SQL-запроса вывести сведения о тех объектах, которые вступят в строй в следующем году. Запрос должен содержать поля: Наименование организации, Наименование объекта, Год ввода в эксплуатацию

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные о тех строительных организациях, адреса которых начинаются на «Л», «М» и «Т». Запрос должен содержать поля: Наименование организации, Адрес, Наименование объекта.

**Вариант 11**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код работы, Марка автомобиля, Наименование работы, Стоимость работы

**Таблица 2.** Заказы.

Структура таблицы: Дата, Номер заказа, Клиент, Марка автомобиля, Код работы, Код исполнителя. **В одном заказе может быть несколько работ.**

**Таблица 3.** Сведения о клиентах.

Структура таблицы: Клиент, Наименование объекта, Адрес объекта.

**Таблица 4.** Справочник исполнителей работ.

Структура таблицы: Код исполнителя, Наименование организации, Адрес, Телефон

**Задание 2.** Создать SQL-запрос, позволяющий просмотреть сведения о работах тех исполнителей, адреса которых начинаются на М, С, Т. Запрос должен содержать поля: Наименование работы, Код исполнителя, Адрес исполнителя.

**Задание 3.** С помощью SQL-запроса подсчитать стоимость выполнения всего заказа.

Запрос должен содержать поля: Номер заказа, Стоимость.

**Вариант 12**

**Задание 1.** Создать таблицы:

**Таблица 1.** Лицевой счет.

Структура таблицы: Табельный номер, Фамилия И.О., Отдел, Должность

**Таблица 2.** Справочник начислений и удержаний.

Структура таблицы: Код, Наименование начисления/удержания. Таблица должна содержать перечень видов начислений и удержаний (например, к видам начислений относятся: оплата по окладу, премия, оплата по больничным листам, отпускные и т.д., к видам удержаний: подоходный налог, погашение ссуды, по исполнительным листам и т.д.). Коды начислений имеют значения от 1 до 100. Коды удержаний имеют значения от 101 до 200

**Таблица 3.** Ведомость начислений.

Структура таблицы: Номер месяца, Табельный номер, Код начисления, Сумма начислено

**Таблица 4.** Ведомость удержаний.

Структура таблицы: Номер месяца, Табельный номер, Код удержания, Сумма удержано

**Таблица 5.** Начальник отдела.

Структура таблицы: Табельный номер, Отдел

**Задание 2.** С помощью SQL-запроса подсчитать сколько человек работает в каждом отделе. Запрос должен содержать поля: Отдел, Количество работников.

**Задание 3.** С помощью SQL-запроса сформировать список работников, фамилии которых начинаются на заданную букву.



**Вариант 13**

**Задание 1.** Создать таблицы:

**Таблица 1.** Каталог товаров.

Структура таблицы: Код товара, Наименование товара, Цена

**Таблица 2.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код клиента, Код товара, Количество

**Таблица 3.** Клиенты.

Структура таблицы: Код клиента, Наименование клиента, Город, Адрес, Телефон.

**Задание 2.** С помощью SQL-запроса подсчитать стоимость заказа. Запрос должен содержать поля: Номер заказа, Стоимость.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть заказы товаров по заданному месяцу. Запрос должен содержать поля: Номер месяца, Наименование товара, Всего заказано.

**Вариант 14**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник услуг.

Структура таблицы: Код услуги, Наименование, Цена

**Таблица 2.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код услуги, Стоимость услуги, Форма оплаты. Форма оплаты может быть наличными или по безналичному расчету.

**Таблица 3.** Клиенты.

Структура таблицы: Код клиента, Фамилия клиента, Адрес клиента, Телефон (Номер телефона должен содержать код города)

**Таблица 4.** Реестр заказов.

Структура таблицы: Номер заказа, Код клиента, Стоимость заказа.

**Задание 2.** С помощью SQL-запроса подсчитать стоимость заказов по каждому клиенту. Запрос должен содержать поля: Фамилия клиента, Стоимость заказа.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные обо всех клиентах, телефон которых содержит код города Рязань.

**Вариант 15**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник марок автомобилей.

Структура таблицы: Код марки автомобиля, Марка автомобиля, Стоимость перевозки 1 ткм, Грузоподъемность.

**Таблица 2.** Автомобили.

Структура таблицы: Инвентарный номер, Марка автомобиля

**Таблица 3.** Заказы.

Структура таблицы: Код заказа, Дата заказа, Код заказчика, Инвентарный номер автомобиля, Масса груза (тн.), Расстояние (км.)

**Таблица 4.** Заказчики.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 2.** Составить SQL-запрос, выводящий сведения об автомобилях, марка которых начинается на «В», «З», «К».

**Задание 3.** Создать SQL-запрос, позволяющий просмотреть сведения о заказах на январь и февраль.

## Варианты заданий 2 Подгруппа

### **Вариант 1**

**Задание 1.** Создать таблицы:

Таблица 1: Справочник групп товаров.

Структура таблицы: Код группы, Наименование группы

**Таблица 2.** Ценник.

Структура таблицы: Код товара, Наименование товара, Код группы, Ед. измерения, Цена

**Таблица 3.** Поступления.

Структура таблицы: Номер накладной, Код товара, Количество поступило, Дата поступления

**Таблица 4.** Продажи.

Структура таблицы: Номер чека, Код товара, Количество продано, Дата продажи

**Таблица 5.** Поставщик.

Структура таблицы: Код поставщика, Код товара, Адрес, Телефон.

**Задание 2.** Составить SQL-запрос, подсчитывающий в каком количестве каждый товар поступил в магазин в январе. Запрос должен содержать следующие поля: Наименование товара, Количество поступило.

**Задание 3.** С помощью SQL-запроса вывести наименование товара, остаток которого равен нулю.

**Вариант 2**

**Задание 1.** Создать таблицы:

**Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 2.** С помощью SQL-запрос определить для каждого региона количество строительных объектов. Запрос должен содержать следующие поля: Наименование региона, Количество строительных объектов.

**Задание 3.** Создать SQL-запрос для получения следующих Сведений: Наименование региона, Наименование отрасли, Наименование объекта, Год ввода в эксплуатацию.

**Вариант 3**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник строительных механизмов.

Структура таблицы: Код механизма, Наименование, Стоимость одного машиночаса работы

**Таблица 2.** Заказчики.

Структура таблицы: Код заказчика, Наименование организации, Форма оплаты (наличными/безналичными)

**Таблица 3.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код заказчика, Код механизма, Объем работы в часах (таблица хранит заказы текущего года)

**Таблица 4.** Сведения о заказчиках.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 2.** С помощью SQL-запроса определить итоговую стоимость работ по каждому строительному механизму. Запрос должен содержать поля: Код механизма, Наименование, Стоимость одного машиночаса работы, Итоговый объем в часах, Стоимость объема работ.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть заказы, выполненные по безналичному расчету. Запрос должен содержать поля: Номер заказа, Форма оплаты.

**Вариант 4**

**Задание 1.** Создать таблицы:

**Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 2.** Составить SQL-запрос, позволяющий увидеть финансирования каждой отрасли в данном году. Запрос должен содержать следующие поля: Наименование отрасли, Объем финансирования.

**Задание 3.** Создать SQL-запрос для получения следующих сведений: Наименование отрасли, Строительная организация, Строительный объект, Годовой объем финансирования.

**Вариант 5**

**Задание 1.** Создать таблицы:

**Таблица 1.** Группа автомобилей.

Структура таблицы: Код группы, Наименование группы, Норма амортизации (норма амортизации задается в % от первоначальной стоимости на 100 км пробега и может иметь значения от 0.01% до 0.05%)

**Таблица 2.** Автомобили.

Структура таблицы: Государственный номер, Марка автомобиля, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Пробег на начало года, Табельный номер материально ответственного лица

**Таблица 3.** Месячные пробеги.

Структура таблицы: Государственный номер, Пробег, Наименование месяца

**Таблица 4.** Материально ответственные лица (МОЛ),.

Структура таблицы: Табельный номер, ФИО, Телефон.

**Задание 2.** С помощью SQL-запроса подсчитать количество автомобилей по маркам. Запрос должен содержать следующие поля: Марка автомобиля, Количество.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть сведения о количестве автомобилей, закрепленных за каждым материально ответственным лицом.

**Вариант 6**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник марок автомобилей.

Структура таблицы: Код марки автомобиля, Марка автомобиля, Стоимость перевозки 1 ткм

**Таблица 2.** Автомобили.

Структура таблицы: Инвентарный номер, Код марки автомобиля

**Таблица 3.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код заказчика, Инвентарный номер автомобиля, Масса груза (тн.), Расстояние (км.)

**Таблица 4.** Заказчики.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 2.** С помощью SQL-запроса подсчитать общую стоимость перевозок на заданную дату. Запрос должен содержать поля: Дата заказа, Стоимость.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные о тех заказчиках, наименования которых начинаются на «А», «В» и «С».

**Вариант 7**

**Задание 1.** Создать таблицы:

**Таблица 1.** Группа основных средств.

Структура таблицы: Код группы, Наименование группы, Годовая норма амортизации (% от первоначальной стоимости)

**Таблица 2.** Подразделение.

Структура таблицы: Код подразделения, Наименование подразделения, Табельный номер материально ответственного лица (МОЛ)

**Таблица 3.** Основные средства.

Структура таблицы: Инвентарный номер, Наименование, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Код подразделения

**Таблица 4.** МОЛ.

Структура таблицы: Табельный номер, ФИО, Адрес, Телефон.

**Задание 2.** Создать SQL-запрос для поиска средств начальная стоимость, которых больше 20000, но меньше 10000.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные Материально-ответственных лиц, фамилии которых начинаются на «И», «П» и «С».

**Вариант 8**

**Задание 1.** Создать таблицы:

**Таблица 1.** Автомобили.

Структура таблицы: Государственный номер, Марка автомобиля, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Пробег на начало года, Стоимость автомобиля на начало года, Табельный номер материально ответственного лица (всего 15 автомобилей)

**Таблица 2.** Месячные пробеги.

Структура таблицы: Государственный номер, Пробег, Номер месяца

**Таблица 3.** МОЛ.

Структура таблицы: Табельный номер, ФИО, телефон

**Таблица 4.** Группа автомобилей.

Структура таблицы: Код группы, Наименование группы, Норма амортизации (% от первоначальной стоимости на 100 км пробега и может иметь значения от 0.01% до 0.05%)

**Задание 2.** С помощью SQL-запроса подсчитать общий пробег автомобилей за месяц по группам. Запрос должен содержать поля: Номер месяца, Наименование группы, Общий пробег.

**Задание 3.** Составить SQL-запрос, выводящий данные об остаточной стоимости каждого автомобиля. Запрос должен содержать поля: Государственный номер, Остаточная стоимость.

**Вариант 9**

**Задание 1.** Создать таблицы:

**Таблица 1.** МОЛ – материально ответственные лица.

Структура таблицы: Табельный номер, ФИО, Дата рождения, Адрес, Телефон

**Таблица 2.** Подразделение.

Структура таблицы: Код подразделения, Наименование подразделения, ФИО материально ответственного лица (МОЛ)

**Таблица 3.** Основные средства.

Структура таблицы: Инвентарный номер, Наименование, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Код подразделения

**Таблица 4.** Группа основных средств.

Структура таблицы: Код группы, Наименование группы, Годовая норма амортизации (% от первоначальной стоимости).

**Задание 2.** Создать SQL-запрос для поиска средств, которые введены в эксплуатацию в этом году. Запрос должен содержать поля: Наименование, Дата ввода в эксплуатацию.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные МОЛ, родившихся в летние месяцы.

**Вариант 10****Задание 1.**

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** План выпуска.

Структура таблицы: Код изделия, План выпуска (количество изделий).

**Таблица 4.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей в изделии

**Задание 2.** С помощью SQL-запроса сформировать План выпуска изделий, который содержать поля: Код изделия, Наименование изделия, Количество деталей в изделии, Стоимость деталей на план выпуска изделия.

**Задание 3.** С помощью SQL-запроса подсчитать какое количество различных деталей, входит в изделие. Запрос должен содержать поля: Наименование изделия, Количество деталей из которых состоит изделие.

**Вариант 11**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код работы, Марка автомобиля, Наименование работы, Стоимость работы

**Таблица 2.** Справочник исполнителей работ.

Структура таблицы: Код исполнителя, Фамилия И.О.

**Таблица 3.** Заказы.

Структура таблицы: Дата, Номер заказа, Клиент, Марка автомобиля, Код вида работ, Код исполнителя

**Таблица 4.** Сведения об исполнителях.

Структура таблицы: номер, ФИО, Дата рождения, Адрес, Телефон.

**Задание 2.** С помощью SQL-запроса вывести сведения о стоимости выполненных работ по исполнителям. ФИО, стоимость выполненных работ.

**Задание 3.** Создать SQL-запрос для вывода сведений о стоимости заказов по каждому клиенту. Запрос должен содержать поля: Клиент, Стоимость заказов.



**Вариант 12**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия И.О., Номер отдела

**Таблица 2.** Ведомость начислений.

Структура таблицы: Номер месяца, Табельный номер, Начислено, Удержано (ввести данные по двум месяцам, значение удержано не заполнять)

**Таблица 3.** Структура предприятия.

Структура таблицы: Номер отдела, Наименование отдела, Структурное подразделение.

**Задание 2.** Создать SQL-запрос, подсчитывающий количество сотрудников, работающих в одном отделе. Запрос должен содержать поля: Наименование отдела, Количество работников.

**Задание 3.** Составить SQL-запрос, имеющий следующие поля: Табельный номер, Номер месяца, Начислено, Удержан ПН, Выдано

**Вариант 13**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник цехов-изготовителей.

Структура таблицы: Код цеха, Наименование цеха

**Таблица 2.** Справочник характеристик продукции.

Структура таблицы: Код продукции, Наименование продукции, Единица измерения, Код цеха-изготовителя, Код группы продукции, Цена (рублей за единицу)

**Таблица 3.** План выпуска продукции в натуральном выражении.

Структура таблицы: Код продукции, Количество, Номер месяца

**Таблица 4.** Группа продукции.

Структура таблицы: Код группы продукции, Наименование группы продукции.

**Задание 2.** Создать SQL-запрос для подсчета количества наименований продукции, выпускаемых цехом. Запрос должен содержать поля: Наименование цеха, количество наименований продукции.

**Задание 3.** Составить SQL-запрос для подсчета стоимости продукции по каждой Группе продукции. Запрос должен содержать поля: Наименование группы продукции, Стоимость продукции.

**Вариант 14**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия И.О., Адрес, Серия паспорта, Номер паспорта, Оклад, Количество детей, Отдел, Дата рождения

**Таблица 2.** Журнал учета отработанного времени.

Структура таблицы: Номер месяца, Табельный номер, Отработанное время (в днях)

**Таблица 3.** Справочник рабочих дней.

Структура таблицы: Наименование месяца, Плановое количество рабочих дней

**Таблица 4.** Зарплата.

Структура таблицы: Фамилия И.О., Номер месяца, Начислено, Подоходный налог.

**Задание 2.** Создать SQL-запрос для подсчета количества работников, работающих в каждом отделе. Запрос должен содержать следующие поля: Отдел, Количество работников.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть возраст сотрудников.

Запрос должен содержать поля: Фамилия, Возраст.

**Вариант 15**

**Задание 1.** Создать таблицы:

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код вида работ, Наименование вида работ, Единица измерения

**Таблица 2.** Справочник расценок.

Структура таблицы: Код работы, Код вида работ, Наименование работы, Расценка (рублей за единицу измерения)

**Таблица 3.** Справочник строительных объектов.

Структура таблицы: Код объекта, Наименование объекта, Населенный пункт, Адрес объекта, Дата начала работ

**Таблица 4.** Объемы работ на объекте.

Структура таблицы: Код объекта, Код работы, Физический объем работы (в единицах измерения)

**Таблица 5.** Справочник окончания работ.

Структура таблицы: Код объекта, Дата окончания работ.

**Задание 2.** С помощью SQL-запроса вывести список работ, срок окончания которых предусмотрен в 3 квартале этого года.

**Задание 3.** Создать SQL-запрос для определения итоговой стоимости всех работ одного вида. Запрос должен содержать поля: Наименование вида работ, Итоговая стоимость

***Практическая работа № 21***  
**Создание запросов с многотабличной БД**  
**Задание**

1. Открыть программу, разработанную в практической работе №21.
2. Добавить формы для добавления, редактирования текущей записи в главной форме, с учетом связанной информации.
3. Реализовать удаление записи.
4. Разработать по варианту задания SQL запросы с участием одной или нескольких таблиц. Использовать представления или LINQ.

## Варианты заданий 1 Подгруппа

### **Вариант 1**

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия, Разряд, Цех

**Таблица 2.** Справочник тарифов.

Структура таблицы: Разряд, Тариф (руб./ час.)

**Таблица 3.** Табель.

Структура таблицы: Табельный номер, Отработанное время в часах, Номер месяца

**Таблица 4.** Список цехов.

Структура таблицы: Цех, Наименование цеха.

**Задание 1.** Составить SQL-запрос на добавление записей в таблицу Цех.

**Задание 2.** Составить SQL-запрос на перевод работника, имеющего заданный табельный номер в другой цех.

**Задание 3.** С помощью SQL-запроса сформировать ведомость начислений зарплаты и упорядочить ее по фамилии. Ведомость должна содержать поля: Табельный номер, Фамилия, Номер месяца, Начислено.

**Задание 4.** Составить SQL-запрос для получения справки о средней зарплате сотрудников, работающих в одном цехе. В справку выводить Наименование цеха, Средняя зарплата по цеху.

**Задание 5.** С помощью SQL-запроса вывести фамилии сотрудников, оклад которых выше среднего оклада по предприятию.

### **Вариант 2**

**Таблица 1.** Справочник цехов-изготовителей.

Структура таблицы: Код цеха, Наименование цеха

**Таблица 2.** Справочник характеристик продукции.

Структура таблицы: Код продукции, Наименование продукции, Единица измерения, Код цеха-изготовителя, Код группы продукции, Цена (рублей за единицу)

**Таблица 3.** План выпуска продукции в натуральном выражении.

Структура таблицы: Код продукции, Количество, Номер месяца

**Таблица 4.** Группа продукции.

Структура таблицы: Код группы продукции, Наименование группы продукции.

**Задание 1.** Составить SQL-запрос на добавление записей в таблицу Группа продукции.

**Задание 2.** Создать SQL-запрос на увеличение цены, выпускаемой продукции на 50%.

**Задание 3.** Составить SQL-запрос для расчета суммарного планового количества (за все месяцы) по каждой продукции. Запрос должен содержать поля: Наименование продукции, Плановое количество продукции.

**Задание 4.** Составить SQL-запрос для подсчета стоимости продукции по каждой Группе продукции. Запрос должен содержать поля: Наименование группы продукции, Стоимость продукции.

**Задание 5.** С помощью SQL-запроса вывести Наименование той продукции, количество которой ниже среднего количества.

**Вариант 3**

**Таблица 1.** Справочник окладов.

Структура таблицы: Должность, Оклад (руб./мес.)

**Таблица 2.** Штатное расписание.

Структура таблицы: Табельный номер, Фамилия И.О., Должность, Отдел

**Таблица 3.** Табель учета отработанного времени.

Структура таблицы: Табельный номер, Отработанное время в днях, Номер месяца

**Таблица 4.** Справочник отделов.

Структура таблицы: Отдел, Наименование отдела, Начальник

**Таблица 5.** Структура предприятия.

Структура таблицы: Структурное подразделение, Отдел.

**Задание 1.** С помощью SQL-запроса сформировать Ведомость начислений зарплаты за месяц и упорядочить ее по возрастанию. Ведомость должна содержать поля: Табельный номер, Фамилия И.О., Должность, Отдел, Начислено, Номер месяца.

**Задание 2.** Создать SQL-запрос для подсчета суммарных начислений сотрудникам по должностям. Запрос должен содержать поля: Должность, Итого начислено по должностям.

**Задание 3.** Составить SQL-запрос на увеличение оклада сотрудников в 2 раза.

**Задание 4.** Составить SQL-запрос на добавление записей в таблицу Структура.

**Задание 5.** Составить SQL-запрос для подсчета сотрудников по отделам в разрезе Структурного подразделения. Запрос должен содержать поля: Структурное подразделение, Наименование отдела, Количество сотрудников. Данные упорядочить по убыванию.

**Вариант 4**

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** План выпуска.

Структура таблицы: Код изделия, Количество изделий, Код цеха

**Таблица 4.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей.

**Считается, что в одном цехе может собираться несколько изделий.**

**Таблица 5.** Цех.

Структура таблицы: Код цеха, Наименование цеха, Начальник.

**Задание 1.** Составить SQL-запрос на уменьшение стоимости сборки за счет механизации процесса в 1,2 раза.

**Задание 2.** Составить SQL-запрос на добавление записей в таблицу Цех.

**Задание 3.** Создать SQL-запрос, для вычисления средней стоимости изделий, выпускаемых цехом. Запрос должен содержать поля: Наименование цеха, Средняя стоимость изделий.

**Задание 4.** С помощью SQL-запроса сформировать План выпуска изделий, который содержать поля: Код изделия, Наименование изделия, Количество деталей в изделии, Стоимость сборки, Стоимость планового выпуска изделия.

**Задание 5.** С помощью SQL-запроса вывести наименование изделия, стоимость которого ниже средней по заводу.

**Вариант 5**

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код вида работ, Наименование вида работ, Единица измерения

**Таблица 2.** Справочник расценок.

Структура таблицы: Код работы, Код вида работ, Наименование работы, Расценка (рублей за единицу измерения)

**Таблица 3.** Справочник строительных объектов.

Структура таблицы: Код объекта, Наименование объекта, Населенный пункт, Адрес объекта, Дата начала работ

**Таблица 4.** Объемы работ на объекте.

Структура таблицы: Код объекта, Код работы, Физический объем работы (в единицах измерения)

**Таблица 5.** Справочник окончания работ.

Структура таблицы: Код объекта, Дата окончания работ.

**Задание 1.** Создать SQL-запрос для просмотра сведений обо всех объектах, на которых предусмотрено выполнение работ одного вида. Запрос должен содержать поля: Наименование объекта, Вид работ, Наименование работы. Данные упорядочить по убыванию Кода вида работ.

**Задание 2.** Составить SQL-запрос на добавление записей в таблицу Справочник окончания работ.

**Задание 3.** Составить SQL-запрос для подсчета количества месяцев строительства. Запрос должен содержать поля: Наименование объекта, Количество месяцев.

**Задание 4.** С помощью SQL-запроса вывести список работ, срок окончания которых предусмотрен в 3 квартале этого года.

**Задание 5.** С помощью SQL-запроса вывести Наименование работ с максимальной расценкой.

**Вариант 6**

**Таблица 1.** Справочник расценок.

Структура таблицы: Код работы, Наименование работы, Расценка (руб. за единицу измерения)

**Таблица 2.** Справочник бригад.

Структура таблицы: Код бригады, Табельный номер, Фамилия И.О.

**Таблица 3.** Наряды.

Структура таблицы: Номер наряда, Код бригады, Табельный номер, Код работы, Выполненный объем (в единицах измерения), КТУ (коэффициент трудового участия может иметь значения от 0 до 1)

**Таблица 4.** Справочник бригадиров.

Структура таблицы: Табельный номер, Фамилия И.О.

**Задание 1.** С помощью SQL-запроса для расчета среднего КТУ для каждой бригады. Запрос должен содержать поля: Код бригады, Средний КТУ.

**Задание 2.** Создать SQL-запрос, позволяющий увидеть максимальный коэффициент трудового участия каждого члена бригады в данной работе. Запрос должен содержать поля: Фамилия И.О., КТУ. Данные упорядочить по убыванию КТУ.

**Задание 3.** Составить SQL-запрос на изменения расценок работ в 0,5 раз.

**Задание 4.** С помощью SQL-запроса сформировать Ведомость начислений бригады, имеющей следующую структуру: Табельный номер, Фамилия И.О., Объем работы, Расценка, Начислено сдельно (Расценка\* Выполненный объем \* КТУ).

**Задание 5.** С помощью SQL-запроса вывести Фамилию того работника, который имеет максимальные начисления.

**Вариант 7**

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей в изделии

**Таблица 4.** План выпуска.

Структура таблицы: Код изделия, План выпуска (количество изделий).

**Задание 1.** С помощью SQL-запроса сформировать План выпуска в стоимостном выражении, который должен содержать поля: Код изделия, Наименование изделия, Количество изделий, Стоимость деталей, Стоимость сборки, Стоимость всего.

**Задание 2.** С помощью SQL-запроса вывести Наименование изделия, состоящего из наибольшего количества деталей.

**Задание 3.** Составить SQL-запрос на уменьшение стоимости сборки за счет механизации процесса в 0,9 раза.

**Задание 4.** Составить SQL-запрос на добавление записей в таблицу План выпуска.

**Задание 5.** С помощью SQL-запроса вывести Наименование изделия с минимальной стоимостью.

**Вариант 8**

**Таблица 1.** Движение.

Структура таблицы: Код детали, Наименование детали, Количество

**Таблица 2.** Накладная на поступление.

Структура таблицы: Номер накладной, Код детали, Количество поступило, Дата прихода

**Таблица 3.** Требование на выдачу.

Структура таблицы: Номер требования, Код детали, Количество выдано, Дата выдачи

**Таблица 4.** Справочник цен.

Структура таблицы: Код детали, Цена

**Задание 1.** С помощью SQL-запроса подсчитать остаток деталей, оставшихся на складе.

Запрос должен содержать поля: Наименование детали, Остаток.

**Задание 2.** Составить SQL-запрос для просмотра движения детали на определенную дату.

Запрос должен содержать поля: Наименование детали, Дата прихода, Количество поступило, Дата выдачи, Количество выдано.

**Задание 3.** Составить SQL-запрос на добавление данных в таблицу Справочник цен.

**Задание 4.** С помощью SQL-запроса рассчитать стоимость деталей, хранящихся на складе.

**Задание 5.** С помощью SQL-запроса определить наименование детали, которая выдавалась чаще остальных.



**Вариант 9**

**Таблица 1:** Справочник групп товаров.

Структура таблицы: Код группы, Наименование группы

**Таблица 2.** Ценник.

Структура таблицы: Код товара, Наименование товара, Код группы, Ед. измерения, Цена

**Таблица 3.** Поступления.

Структура таблицы: Номер накладной, Код товара, Количество поступило, Дата поступления, Код поставщика

**Таблица 4.** Продажи.

Структура таблицы: Номер чека, Код товара, Количество продано, Дата продажи

**Таблица 5.** Остаток.

Структура таблицы: Код товара, Остаток на конец недели

**Таблица 6.** Поставщик.

Структура таблицы: Код поставщика, Наименование поставщика, Адрес, Телефон

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Поставщик.

**Задание 2.** С помощью SQL-запроса подсчитать общую стоимость, поступивших товаров за день. Запрос должен содержать поля: Дата, Наименование товара, Общая стоимость.

**Задание 3.** С помощью SQL-запроса подсчитать остаток каждого товара по группам в стоимостном выражении. Запрос должен содержать поля: Наименование группы, Стоимость остатка по группам.

**Задание 4.** Составить SQL-запрос для расчета стоимости остатка по каждому товару на конец заданного дня. Запрос должен содержать поля: Наименование товара, Остаток, Стоимость остатка.

Остаток товара на конец дня определяется как сумма Остатка на конец недели с общим количеством поступлений с начала недели до заданного дня, за вычетом количества продаж с начала недели до заданного дня

**Задание 5.** С помощью SQL-запроса вывести Наименование товара пользующегося наибольшим спросом.

**Вариант 10**

**Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Строительная организация.

**Задание 2.** С помощью SQL-запроса рассчитать годовой объем финансирования для каждой отрасли. Запрос должен содержать поля: Наименование отрасли, Объем финансирования.

**Задание 3.** Составить SQL-запрос, формирующий сведения о финансировании региона. Запрос должен содержать поля: Наименование региона, Объем финансирования.

**Задание 4.** С помощью SQL-запроса подсчитать средний годовой объем финансирования по каждой отрасли. Запрос содержит поля: Наименование отрасли, Средний объем. Данные представить в алфавитном порядке наименований отраслей.

**Задание 5.** С помощью SQL-запроса вывести Наименование отрасли с максимальным финансированием.

**Вариант 11**

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код работы, Марка автомобиля, Наименование работы, Стоимость работы

**Таблица 2.** Заказы.

Структура таблицы: Дата, Номер заказа, Клиент, Марка автомобиля, Код работы, Код исполнителя. **В одном заказе может быть несколько работ.**

**Таблица 3.** Сведения о клиентах.

Структура таблицы: Клиент, Наименование объекта, Адрес объекта.

**Таблица 4.** Справочник исполнителей работ.

Структура таблицы: Код исполнителя, Наименование организации, Адрес, Телефон

**Задание 1.** Составить SQL-запрос на добавление записей в таблицу Справочник исполнителей работ.

**Задание 2.** С помощью SQL-запроса подсчитать премию для исполнителя, если она равна 10% от стоимости работы. Запрос должен содержать поля: Наименование организации, Наименование работы, Премия.

**Задание 3.** С помощью SQL-запроса выполнить подсчет автомобилей занятых выполнением работ на одном объекте. Запрос должен содержать поля: Наименование объекта, Количество автомобилей.

**Задание 4.** С помощью SQL-запроса подсчитать количество заказов за летние месяцы по маркам автомобиля. Запрос должен содержать поля: Марка автомобиля, Количество заказов.

**Задание 5.** С помощью SQL-запроса вывести полные сведения о клиенте, заказ которого имеет максимальную стоимость.

**Вариант 12**

**Таблица 1.** Лицевой счет.

Структура таблицы: Табельный номер, Фамилия И.О., Отдел, Должность

**Таблица 2.** Справочник начислений и удержаний.

Структура таблицы: Код, Наименование начисления/удержания. Таблица должна содержать перечень видов начислений и удержаний (например, к видам начислений относятся: оплата по окладу, премия, оплата по больничным листам, отпускные и т.д., к видам удержаний: подоходный налог, погашение ссуды, по исполнительным листам и т.д.). Коды начислений имеют значения от 1 до 100. Коды удержаний имеют значения от 101 до 200

**Таблица 3.** Ведомость начислений.

Структура таблицы: Номер месяца, Табельный номер, Код начисления, Сумма начислено

**Таблица 4.** Ведомость удержаний.

Структура таблицы: Номер месяца, Табельный номер, Код удержания, Сумма удержано

**Таблица 5.** Начальник отдела.

Структура таблицы: Табельный номер, Отдел

**Задание 1.** С помощью SQL-запроса подсчитать общую сумму начислений по каждому коду. Запрос должен содержать поля: Код начислений, Сумма начислений.

**Задание 2.** Создать SQL-запрос, подсчитывающий сумму начисления и удержания за один месяц. Запрос должен содержать поля: Номер месяца, Всего начислено, Всего удержано.

**Задание 3.** Составить SQL-запрос для расчета суммы к выдаче для каждого работника. Запрос должен содержать поля: Фамилия, Номер месяца, Сумма к выдаче.

**Задание 4.** Составить SQL-запрос на перевод работника по заданному табельному номеру в другой отдел.

**Задание 5.** С помощью SQL-запроса вывести фамилию сотрудника с максимальными начислениями в заданный месяц.

**Вариант 13**

**Таблица 1.** Каталог товаров.

Структура таблицы: Код товара, Наименование товара, Цена

**Таблица 2.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код клиента, Код товара, Количество

**Таблица 3.** Клиенты.

Структура таблицы: Код клиента, Наименование клиента, Город, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Клиенты.

**Задание 2.** С помощью SQL-запроса подсчитать среднюю стоимость заказов для каждого клиента. Запрос должен содержать поля: Наименование клиента, Средняя стоимость заказа.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть данные обо всех клиентах Санкт-Петербурга и Москвы.

**Задание 4.** Составить SQL-запрос, подсчитывающий стоимость товаров с учетом НДС (НДС =0,2).

**Задание 5.** С помощью SQL-запроса вывести Наименование тех клиентов, стоимость заказа которых выше средней.

**Вариант 14**

**Таблица 1.** Справочник услуг.

Структура таблицы: Код услуги, Наименование, Цена

**Таблица 2.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код услуги, Стоимость услуги, Форма оплаты. Форма оплаты может быть наличными или по безналичному расчету.

**Таблица 3.** Клиенты.

Структура таблицы: Код клиента, Фамилия клиента, Адрес клиента, Телефон (Номер телефона должен содержать код города)

**Таблица 4.** Реестр заказов.

Структура таблицы: Номер заказа, Код клиента, Стоимость заказа.

**Задание 1.** С помощью SQL-запроса подсчитать количество заказов по наличному и безналичному расчету в первом квартале. Запрос должен содержать поля: Форма оплаты, Количество заказов.

**Задание 2.** Создать SQL-запрос для определения итоговой стоимости услуг, оказанных в зимние месяцы. Запрос должен содержать поля: Номер месяца, Стоимость.

**Задание 3.** Составить SQL-запрос на увеличение стоимости услуг на 0,15%.

**Задание 4.** С помощью SQL-запроса определить количество заказов по клиентам. Запрос должен содержать поля: Фамилия клиента, Количество заказов.

**Задание 5.** Создайте запрос, показывающий Фамилии клиентов, со стоимостью заказов ниже средней.

**Вариант 15**

**Таблица 1.** Справочник марок автомобилей.

Структура таблицы: Код марки автомобиля, Марка автомобиля, Стоимость перевозки 1ткм, Грузоподъемность.

**Таблица 2.** Автомобили.

Структура таблицы: Инвентарный номер, Марка автомобиля

**Таблица 3.** Заказы.

Структура таблицы: Код заказа, Дата заказа, Код заказчика, Инвентарный номер автомобиля, Масса груза (тн.), Расстояние (км.)

**Таблица 4.** Заказчики.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос для подсчета автомобилей по маркам. Запрос должен содержать поля: Марка автомобиля, Количество.

**Задание 2.** Создать SQL-запрос на добавление данных в таблицу Заказчики.

**Задание 3.** С помощью SQL-запроса вывести сведения о стоимости перевозок для каждого заказчика. Запрос должен содержать поля: Наименование заказчика, Стоимость заказа.

**Задание 4.** С помощью SQL-запроса вывести средние сведения о заказах по зимним месяцам в зависимости от марки автомобиля. Запрос должен содержать поля: Марка автомобиля, Средние сведения о заказах за месяц.

**Задание 5.** С помощью SQL-запроса вывести марку автомобиля, имеющего грузоподъемность выше средней.

## Варианты заданий 2 Подгруппа

### **Вариант 1**

**Таблица 1.** Справочник групп товаров.

Структура таблицы: Код группы, Наименование группы

**Таблица 2.** Ценник.

Структура таблицы: Код товара, Наименование товара, Код группы, Ед. измерения, Цена

**Таблица 3.** Поступления.

Структура таблицы: Номер накладной, Код товара, Количество поступило, Дата поступления

**Таблица 4.** Продажи.

Структура таблицы: Номер чека, Код товара, Количество продано, Дата продажи

**Таблица 5.** Поставщик.

Структура таблицы: Код поставщика, Код товара, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Поставщик.

**Задание 2.** С помощью SQL-запроса подсчитать стоимость поставок. Запрос должен содержать следующие поля: Наименование поставщика, Общая стоимость.

**Задание 3.** С помощью SQL-запроса подсчитать остаток каждого товара. Запрос должен содержать следующие поля: Код группы, Наименование товара, Остаток. Рассчитывается остаток для каждого товара по формуле:

Остаток = Всего поступило - Всего продано.

**Задание 4.** Составить SQL-запрос, позволяющий определить остаточную стоимость по каждому товару. Запрос должен содержать поля: Наименование товара, Остаточная стоимость.

**Задание 5.** С помощью SQL-запроса подсчитать общее количество проданных товаров за месяц. Запрос должен содержать следующие поля: Наименование товара, Количество продано за месяц.

**Вариант 2****Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Строительная организация.

**Задание 2.** С помощью SQL-запроса сосчитать объем финансирования по кварталам. Запрос должен содержать следующие поля: Наименование отрасли, Объем финансирования по кварталам.

**Задание 3.** Составить SQL-запрос, позволяющий увидеть финансирования каждой отрасли за год. Запрос должен содержать следующие поля: Наименование отрасли, Общий объем финансирования.

**Задание 4.** С помощью SQL-запроса просмотреть сведения о строительных организациях. Запрос должен содержать следующие поля: Наименование организации, Адрес, Количество объектов, на которых ведет работы, Общий объем финансирования.

**Задание 5.** С помощью SQL-запроса вывести наименование региона с минимальным финансированием в первом квартале.

**Вариант 3****Таблица 1.** Справочник строительных механизмов.

Структура таблицы: Код механизма, Наименование, Стоимость одного машиночаса работы

**Таблица 2.** Заказчики.

Структура таблицы: Код заказчика, Наименование организации, Форма оплаты (наличными/безналичными)

**Таблица 3.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код заказчика, Код механизма, Объем работы в часах (таблица хранит заказы текущего года)

**Таблица 4.** Сведения о заказчиках.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Сведения о заказчиках.

**Задание 2.** С помощью SQL-запроса подсчитать количество заказов в первом квартале по каждому заказчику. Запрос должен содержать поля: Наименование заказчика, Количество заказов.

**Задание 3.** Составить SQL-запрос на увеличение Стоимость одного машиночаса работы для заказчиков, форма оплаты которых безналичными на 0,2%.

**Задание 4.** С помощью SQL-запроса подсчитать общее количество заказов, выполненных за месяц. Запрос должен содержать следующие поля: Код механизма, Общее количество заказов.

**Задание 5.** С помощью SQL-запроса ввести наименование заказчика с максимальной стоимостью заказа.



**Вариант 4****Таблица 1.** Регионы.

Структура таблицы: Код региона, Наименование региона

**Таблица 2.** Отрасли.

Структура таблицы: Код отрасли, Наименование отрасли (жилье, здравоохранение, культура и т.д.)

**Таблица 3.** Строительный объект.

Структура таблицы: Наименование объекта, Код региона, Код отрасли, Код организации, Объем финансирования в 1 квартале, Объем финансирования во 2 квартале, Объем финансирования в 3 квартале, Объем финансирования в 4 квартале, Год ввода в эксплуатацию.

**Таблица 4.** Строительная организация.

Структура таблицы: Код организации, Наименование организации, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Строительный объект.

**Задание 2.** С помощью SQL-запроса сосчитать объем финансирования для региона по каждому кварталу. Полученные данные упорядочить в алфавитном порядке наименований регионов.

**Задание 3.** Составить SQL-запрос, позволяющий просмотреть на каких строительных объектах ведут работы те строительных организации, адреса которых начинаются на «В», «М» и «С».

**Задание 4.** С помощью SQL-запроса подсчитать годовой объем финансирования по каждому региону. Запрос должен содержать следующие поля: Наименование региона, Объем финансирования.

**Задание 5.** С помощью SQL-запроса вывести наименование отрасли с объемом финансирования ниже среднего.

**Вариант 5****Таблица 1.** Группа автомобилей.

Структура таблицы: Код группы, Наименование группы, Норма амортизации (норма амортизации задается в % от первоначальной стоимости на 100 км пробега и может иметь значения от 0.01% до 0.05%)

**Таблица 2.** Автомобили.

Структура таблицы: Государственный номер, Марка автомобиля, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Пробег на начало года, Табельный номер материально ответственного лица

**Таблица 3.** Месячные пробеги.

Структура таблицы: Государственный номер, Пробег, Наименование месяца

**Таблица 4.** Материально ответственные лица (МОЛ),.

Структура таблицы: Табельный номер, ФИО, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Материально ответственные лица (МОЛ).

**Задание 2.** С помощью SQL-запроса подсчитать средний пробег автомобиля за год. Запрос должен содержать поля: Государственный номер, Марка автомобиля, Средний пробег.

**Задание 3.** Создать SQL-запрос рассчитывающий амортизацию для тех автомобилей, которые введены в эксплуатацию в текущем году. Запрос должен содержать поля: Государственный номер, Марка автомобиля, Первоначальная стоимость, Наименование группы, Норма амортизации, Пробег за год, Амортизация.

**Задание 4.** Составить SQL-запрос, выводящий данные о государственных номерах автомобилей с 0 остаточной стоимостью.

**Задание 5.** С помощью SQL-запроса вывести марку самого дешевого автомобиля.

**Вариант 6**

**Таблица 1.** Справочник марок автомобилей.

Структура таблицы: Код марки автомобиля, Марка автомобиля, Стоимость перевозки 1 ткм

**Таблица 2.** Автомобили.

Структура таблицы: Инвентарный номер, Код марки автомобиля

**Таблица 3.** Заказы.

Структура таблицы: Номер заказа, Дата заказа, Код заказчика, Инвентарный номер автомобиля, Масса груза (тн.), Расстояние (км.)

**Таблица 4.** Заказчики.

Структура таблицы: Код заказчика, Наименование заказчика, Адрес, Телефон.

**Задание 1.** С помощью SQL-запроса подсчитать количество заказов по каждому заказчику в течение квартала. Запрос должен содержать поля: Наименование заказчика, Количество заказов.

**Задание 2.** Создать SQL-запрос на добавление данных в таблицу Заказчики.

**Задание 3.** С помощью SQL-запроса подсчитать среднюю массу груза, перевезенную каждой маркой автомобиля. Запрос должен содержать поля: Марка автомобиля, Средняя масса груза.

**Задание 4.** С помощью SQL-запроса подсчитать суммарную стоимость заказов по каждому заказчику. Запрос должен содержать поля: Наименование заказчика, Стоимость заказа.

**Задание 5.** С помощью SQL-запроса вывести номер заказа с максимальной стоимостью.

**Вариант 7**

**Таблица 1.** Группа основных средств.

Структура таблицы: Код группы, Наименование группы, Годовая норма амортизации (% от первоначальной стоимости)

**Таблица 2.** Подразделение.

Структура таблицы: Код подразделения, Наименование подразделения, Табельный номер материально ответственного лица (МОЛ)

**Таблица 3.** Основные средства.

Структура таблицы: Инвентарный номер, Наименование, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Код подразделения

**Таблица 4.** МОЛ.

Структура таблицы: Табельный номер, ФИО, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу МОЛ.

**Задание 2.** С помощью SQL-запроса вывести сведения амортизационных отчислений за месяц, которые вычисляется по формуле:

Первоначальная стоимость \* Годовая норма амортизации / (12 \* 100).

Запрос должен содержать поля: Наименование группы, Амортизационные отчисления

**Задание 3.** Создать SQL-запрос для вывода списанных основных средств. Запрос должен содержать следующие поля: Инвентарный номер, Дата ввода в эксплуатацию, Срок окончания амортизации. Срок окончания амортизации вычисляется по формуле: Дата ввода в эксплуатацию + 365 \* 100 / Годовая норма амортизации.

**Задание 4.** Создать SQL-запрос для поиска основных средств с нулевой остаточной стоимостью.

**Задание 5.** С помощью SQL-запроса вывести фамилию МОЛ, за которым числится самое дорогое основное средство.

**Вариант 8****Таблица 1.** Автомобили.

Структура таблицы: Государственный номер, Марка автомобиля, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Пробег на начало года, Стоимость автомобиля на начало года, Табельный номер материально ответственного лица (всего 15 автомобилей)

**Таблица 2.** Месячные пробеги.

Структура таблицы: Государственный номер, Пробег, Номер месяца

**Таблица 3.** МОЛ.

Структура таблицы: Табельный номер, ФИО, телефон

**Таблица 4.** Группа автомобилей.

Структура таблицы: Код группы, Наименование группы, Норма амортизации (% от первоначальной стоимости на 100 км пробега и может иметь значения от 0.01% до 0.05%)

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Группа автомобилей.

**Задание 2.** С помощью SQL-запроса рассчитывающих по каждому автомобилю амортизацию за год. Запрос должен содержать поля: Наименование группы, Марка автомобиля, Государственный номер, Отчисления

**Задание 3.** Составить SQL-запрос, позволяющий производить передачу автомобиля от одного МОЛ другому.

**Задание 4.** Составить SQL-запрос для просмотра сведений об амортизационных отчислениях с начала года по каждой группе автомобилей. Запрос должен содержать следующие поля: Наименование группы, Сумма отчислений с начала года.

**Задание 5.** С помощью SQL-запроса вывести государственный номер автомобиля с максимальным пробегом на данный период.

**Вариант 9****Таблица 1.** МОЛ – материально ответственные лица.

Структура таблицы: Табельный номер, ФИО, Дата рождения, Адрес, Телефон

**Таблица 2.** Подразделение.

Структура таблицы: Код подразделения, Наименование подразделения, ФИО материально ответственного лица (МОЛ)

**Таблица 3.** Основные средства.

Структура таблицы: Инвентарный номер, Наименование, Код группы, Первоначальная стоимость, Дата ввода в эксплуатацию, Код подразделения

**Таблица 4.** Группа основных средств.

Структура таблицы: Код группы, Наименование группы, Годовая норма амортизации (% от первоначальной стоимости).

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Группа основных средств.

**Задание 2.** С помощью SQL-запроса вывести сведения амортизационных отчислений за месяц по каждому подразделению, которые вычисляются по формуле: Первоначальная стоимость \* Годовая норма амортизации / 12 \* 100. Запрос должен содержать поля: Наименование подразделения, Амортизационные отчисления.

**Задание 3.** Создать SQL-запрос для вывода основных средств, имеющих срок окончания амортизации в этом году. Запрос должен содержать следующие поля: Инвентарный номер, Дата ввода в эксплуатацию, Срок окончания амортизации. Срок окончания амортизации вычисляется по формуле: Дата ввода в эксплуатацию + 365 \* 100 / Годовая норма амортизации.

**Задание 4.** Создать SQL-запрос для поиска основных средств, остаточная стоимость которых не равна нулю.

**Задание 5.** С помощью SQL-запроса рассчитать отчисления по группам. Запрос должен содержать поля: Наименование группы, Отчисления.

**Вариант 10**

**Таблица 1.** Справочник изделий.

Структура таблицы: Код изделия, Наименование изделия, Стоимость сборки

**Таблица 2.** Справочник деталей.

Структура таблицы: Код детали, Наименование детали, Цена

**Таблица 3.** План выпуска.

Структура таблицы: Код изделия, План выпуска (количество изделий).

**Таблица 4.** Состав изделия.

Структура таблицы: Код изделия, Код детали, Количество деталей в изделии

**Задание 1.** С помощью SQL-запроса подсчитать итоговую стоимость по каждой детали с учетом планы выпуска изделий. Запрос содержит поля: Наименование детали, Итоговая стоимость.

**Задание 2.** Составить SQL-запрос на изменение стоимости сборки на 20%.

**Задание 3.** Составить SQL-запрос на добавление записей в таблицу Цех.

**Задание 4.** Создать SQL-запрос, для вычисления стоимости каждого изделия, выпускаемого цехом. Запрос должен содержать поля: Наименование цеха, Наименование изделия, Стоимость изделия.

**Задание 5.** С помощью SQL-запроса вывести наименование изделия, состоящего из наибольшего количества деталей.

**Вариант 11**

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код работы, Марка автомобиля, Наименование работы, Стоимость работы

**Таблица 2.** Справочник исполнителей работ.

Структура таблицы: Код исполнителя, Фамилия И.О.

**Таблица 3.** Заказы.

Структура таблицы: Дата, Номер заказа, Клиент, Марка автомобиля, Код вида работ, Код исполнителя

**Таблица 4.** Сведения об исполнителях.

Структура таблицы: номер, ФИО, Дата рождения, Адрес, Телефон.

**Задание 1.** Создать SQL-запрос на добавление данных в таблицу Сведения об исполнителях.

**Задание 2.** Создать SQL-запрос для подсчета количества заказов по каждой марке автомобиля. Запрос должен содержать поля: Марка автомобиля, Количество заказов.

**Задание 3.** Создать SQL-запрос для подсчета количества заказов в день. Запрос должен содержать поля: Дата, Наименование работы, Количество заказов. Дата вводится по запросу.

**Задание 4.** Составить SQL-запрос, позволяющий просмотреть полные данные о заказах, сделанных в первом квартале текущего года. Запрос должен содержать поля: Номер заказа, Дата заказа, Марка автомобиля, Стоимость заказа.

**Задание 5.** Вывести Наименование работы, которая выполнялась чаще всего.

**Вариант 12**

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия И.О., Номер отдела

**Таблица 2.** Ведомость начислений.

Структура таблицы: Номер месяца, Табельный номер, Начислено, Удержано (ввести данные по двум месяцам, значение удержано не заполнять)

**Таблица 3.** Структура предприятия.

Структура таблицы: Номер отдела, Наименование отдела, Структурное подразделение.

**Задание 1.** Составить SQL-запрос на добавление записей в таблицу Структура.

**Задание 2.** С помощью SQL-запроса обновить данные поля подоходный налог (Подоходный налог = Начислено\*0,13)

**Задание 3.** Создать SQL-запрос, подсчитывающий среднюю заработную плату по отделам. Запрос должен содержать поля: Наименование отдела, средняя заработная плата. Номер месяца вводится по запросу.

**Задание 4.** Составить SQL-запрос для подсчета сотрудников по отделам в разрезе Структурного подразделения.

**Задание 5.** С помощью SQL-запроса вывести Фамилия И.О. работника с максимальным начислением за первый месяц.

**Вариант 13**

**Таблица 1.** Справочник цехов-изготовителей.

Структура таблицы: Код цеха, Наименование цеха

**Таблица 2.** Справочник характеристик продукции.

Структура таблицы: Код продукции, Наименование продукции, Единица измерения, Код цеха-изготовителя, Код группы продукции, Цена (рублей за единицу)

**Таблица 3.** План выпуска продукции в натуральном выражении.

Структура таблицы: Код продукции, Количество, Номер месяца

**Таблица 4.** Группа продукции.

Структура таблицы: Код группы продукции, Наименование группы продукции.

**Задание 1.** Составить SQL-запрос на добавление записей в таблицу Группа продукции.

**Задание 2.** С помощью SQL-запроса сформировать План выпуска продукции за месяц в стоимостном выражении, включив в запрос поля: Наименование группы продукции, Наименование продукции, Номер месяца, Стоимость.

**Задание 3.** Создать SQL-запрос на увеличение цены, выпускаемой продукции на 15%.

**Задание 4.** Составить SQL-запрос для расчета суммарного планового количества (за все месяцы) по каждой продукции. Запрос должен содержать поля: Наименование продукции, Количество продукции.

**Задание 5.** С помощью SQL-запроса вывести по каждому цеху Наименование той продукции, количество которой ниже среднего количества, выпускаемого цехом.

**Вариант 14**

**Таблица 1.** Справочник работников.

Структура таблицы: Табельный номер, Фамилия И.О., Адрес, Серия паспорта, Номер паспорта, Оклад, Количество детей, Отдел, Дата рождения

**Таблица 2.** Журнал учета отработанного времени.

Структура таблицы: Номер месяца, Табельный номер, Отработанное время (в днях)

**Таблица 3.** Справочник рабочих дней.

Структура таблицы: Наименование месяца, Плановое количество рабочих дней

**Таблица 4.** Зарплата.

Структура таблицы: Фамилия И.О., Номер месяца, Начислено, Подоходный налог.

**Задание 1.** Составить SQL-запрос, позволяющий получить сведения о тех сотрудниках, которые отработали не целый месяц. Запрос должен содержать следующие поля:

Фамилия, Отработанное время (в днях), Номер месяца.

**Задание 2.** С помощью SQL-запроса сформировать ведомость начислений зарплаты и упорядочить ее по фамилии. Ведомость должна содержать поля: Отдел, Табельный номер, Фамилия, Номер месяца, Начислено ( $\text{Начислено} = \text{Оклад} / \text{Плановое количество рабочих дней} * \text{Отработанное время}$ ).

**Задание 3.** С помощью SQL-запроса сформировать ведомость удержаний подоходного налога. Ведомость должна содержать поля: Табельный номер, Номер месяца, Подоходный налог. Подоходный налог должен рассчитываться по формуле:  $\text{ПН} = (\text{Всего начислено} - \text{Стандартный вычет на работника} - \text{Стандартный вычет на 1-го ребенка} * \text{Количество детей}) * 0.13$  Стандартный вычет вводить по запросу или задать постоянной величиной в тексте запроса.

**Задание 4.** Составить SQL-запрос на перевод работника, имеющего заданный табельный номер в другой отдел.

**Задание 5.** С помощью SQL-запроса сформировать список работников, фамилии которых не начинаются на «А», «С» и «П».

**Вариант 15**

**Таблица 1.** Справочник видов работ.

Структура таблицы: Код вида работ, Наименование вида работ, Единица измерения

**Таблица 2.** Справочник расценок.

Структура таблицы: Код работы, Код вида работ, Наименование работы, Расценка (рублей за единицу измерения)

**Таблица 3.** Справочник строительных объектов.

Структура таблицы: Код объекта, Наименование объекта, Населенный пункт, Адрес объекта, Дата начала работ

**Таблица 4.** Объемы работ на объекте.

Структура таблицы: Код объекта, Код работы, Физический объем работы (в единицах измерения)

**Таблица 5.** Справочник окончания работ.

Структура таблицы: Код объекта, Дата окончания работ.

**Задание 1.** С помощью SQL-запроса подсчитать количество работ, которые выполняются на одном объекте в одном городе. Запрос содержит поля: Населенный пункт, Наименование объекта, Количество работ.

**Задание 2.** Создать SQL-запрос для просмотра сведений обо всех объектах, на которых предусмотрено выполнение работ одного вида. Запрос должен содержать поля: Наименование работы, Наименование объекта, Физический объем работы (в единицах измерения). Данные упорядочить по убыванию Физический объем работы.



**Задание 3.** Составить SQL-запрос на добавление записей в таблицу Справочник окончания работ.

**Задание 4.** Составить SQL-запрос для подсчета количества месяцев строительства. Запрос должен содержать поля: Наименование объекта, Количество месяцев.

**Задание 5.** С помощью SQL-запроса вывести Наименование работ с расценкой выше средней.