

```

Для Стр = Объект.НачСтрока По КонСтр Цикл
    СтруктураСтроки = ПодготовитьСтроку (ЛистФайла, Стр) ;

    Если ПустаяСтрока (СтруктураСтроки.КодТовара) = Ложь ИЛИ
ПустаяСтрока (СтруктураСтроки.НаименованиеТовара) = Ложь Тогда
        НовСтрока = ДокОбъект [ИмяТЧ].Добавить () ;

        Если ТипЭнч (Объект.Документ) =
Тип ("ДокументСсылка.РеализацияТоваровУслуг") Тогда
            ПараметрыНоменклатуры = СоздатьНайтиНоменклатуру (СтруктураСтроки) ;
        Иначе
            ПараметрыНоменклатуры = СоздатьНайтиНоменклатуру (СтруктураСтроки,
Истина) ;

        КонецЕсли;

        Номенклатура = ПараметрыНоменклатуры.Номенклатура;

        Если Номенклатура.Пустая() Тогда
            Возврат;
        КонецЕсли;

```

Рисунок 4: Фрагмент кода для кнопки "Загрузить"

```

ВЫБРАТЬ
    ТоварыНаСкладахОстатки.Номенклатура,
    СУММА(ТоварыНаСкладахОстатки.ВНаличииОстаток) КАК ВНаличииОстаток,
    спрНоменклатура.МинКол
ИЗ
    РегистрНакопления.ТоварыНаСкладах.Остатки КАК ТоварыНаСкладахОстатки
    ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Номенклатура КАК спрНоменклатура
    ПО ТоварыНаСкладахОстатки.Номенклатура = спрНоменклатура.Ссылка

СГРУППИРОВАТЬ ПО
    ТоварыНаСкладахОстатки.Номенклатура,
    спрНоменклатура.МинКол

```

Рисунок 5: Запрос, для создания отчёта

### Список использованных источников

1. Радченко М.Г. Хрусталева Е.Ю. 1С: Предприятие 8.2 Практическое пособие разработчика. Примеры и типовые приемы. – М.: ООО «1С-Публишинг», 2009.-874с.
2. Хрусталева Е.Ю. Разработка сложных отчетов в 1С. Предприятии 8. Система компоновки данных. – М.: ООО «1С-Публишинг», 2008.-515с.
3. Внешняя обработка – URL: [http://v8.1c.ru/overview/Term\\_000000601.htm](http://v8.1c.ru/overview/Term_000000601.htm) (дата обращения: 20.10.2018)

20.53.19

## РАЗРАБОТКА ФУНКЦИЙ УЧЁТА РАБОЧЕГО ВРЕМЕНИ С ПРИМЕНЕНИЕМ СРЕДСТВ T-SQL

**Автор:** Макарова Анастасия Владимировна, студентка третьего курса Филиала «Протвино» государственного университета «Дубна»

**Научный руководитель:** к.т.н., доцент кафедры Кульман Татьяна Николаевна

### Аннотация

В работе рассматривается способ работы с базой данных SQL Server и решение некоторых проблем, связанных с типом данных `datetime`. Для решения проблем создается приложение, позволяющее отмечать приход и уход сотрудников. Работа с базой данных осуществляется посредством Visual Studio 2015 и Transact - SQL.

### Annotation

The article discusses the way to work with a SQL Server database and the solution of some problems associated with the `datetime` data type. To solve problems, an application is created that allows you to mark the arrival and departure of employees. Work with the database is carried out through Visual Studio 2015 and Transact - SQL.

**Ключевые слова:** базы данных, SQL Server, T-SQL, Visual Studio 2015, тип данных `datetime`, проблемы с `datetime`.

**Keywords:** data base, SQL Server, T-SQL, Visual Studio 2015, the `datetime` data type, the problems with `datetime`.

Неотъемлемой частью жизни современного человека являются информационные технологии. Сейчас деятельность человека, по возможности, заменяется разными программами или оборудованием, иными словами, происходит автоматизация процессов. Так, например, облегчается поиск и хранение информации, её изменение и удаление. Каждое учреждение имеет базу данных, в которой хранится информация о нём, его сотрудниках, потребителях. Базы данных состоят из таблиц, количество которых может быть достаточно большим. Хранение информации в таком виде упрощает её поиск, изменение, удаление, увеличивает защищённость хранимых данных. Таким образом, повысилась важность умения правильно и грамотно разработать систему хранения информации, что, несомненно, показывает актуальность выбранной темы.

Базы данных имеют множество различных типов данных, работа с каждым из которых имеет свои особенности, и проблемы. На крупных предприятиях или в маленьких компаниях всегда большим значением, кроме численных типов, будут обладать типы для отображения даты и времени. Их востребованность связана с тем, что нужно учитывать приход и уход сотрудников с работы, составлять график работы, учитывать срок годности продукции и тому подобное. К таким типам данных можно отнести тип `date`, `time`, `datetime`, `smalldate` и другие. Если нужно ввести дату приёма сотрудника на работу, то можно воспользоваться `date`, а если нужно вести учёт прихода и ухода сотрудников, то здесь уже правильно будет использовать `datetime`.

`Datetime` является самым оптимальным типом, для отображения даты и времени, поскольку имеет формат (ГГГГ-ММ-ДД чч:мм:сс[.ннн]), который позволяет учитывать и дату и время, и достаточно большой диапазон дат (от 01.01.1753 до 31.12.9999). Именно по этим причинам, чаще всего в разработке таблиц баз данных используют данный тип. Но как говорилось выше, работа со всеми типами данных имеет свои особенности. К часто встречаемым особенностям, связанным с типом `datetime` можно отнести:

1. Учёт в расчётах выходных и праздников;
2. Проблема исключения из даты части, соответствующей столетию (проблема 2000 года);
3. Расчёт стажа работы и другие.

Для решения второй проблемы, нужно просто при вводе даты вводить год не двумя последними цифрами, а использовать все четыре. Что касается решения первой задачи, она решается в практической части данной работы.

В практической части разрабатывалась небольшая программа, позволяющая отмечать приход и уход сотрудников на работу, отображать график их работы, выводить информацию о праздничных дня и отчёт, позволяющий проанализировать рабочее время сотрудников за месяц. Вся эта информация впоследствии может помочь в расчётах заработной платы и премий. Данная программа была реализована на SQL Server и Visual Studio 2015. В среде Visual Studio 2015 использовались язык программирования C# и Windows Forms.

Перед началом создания таблиц в SQL Server следует продумать структуру таблиц и их связь между собой. После этого можно создавать проекты таблиц и диаграмму базы данных. Диаграмма базы данных демонстрирует связь таблиц. Диаграмма разработанной базы данных продемонстрирована на рисунке 1.

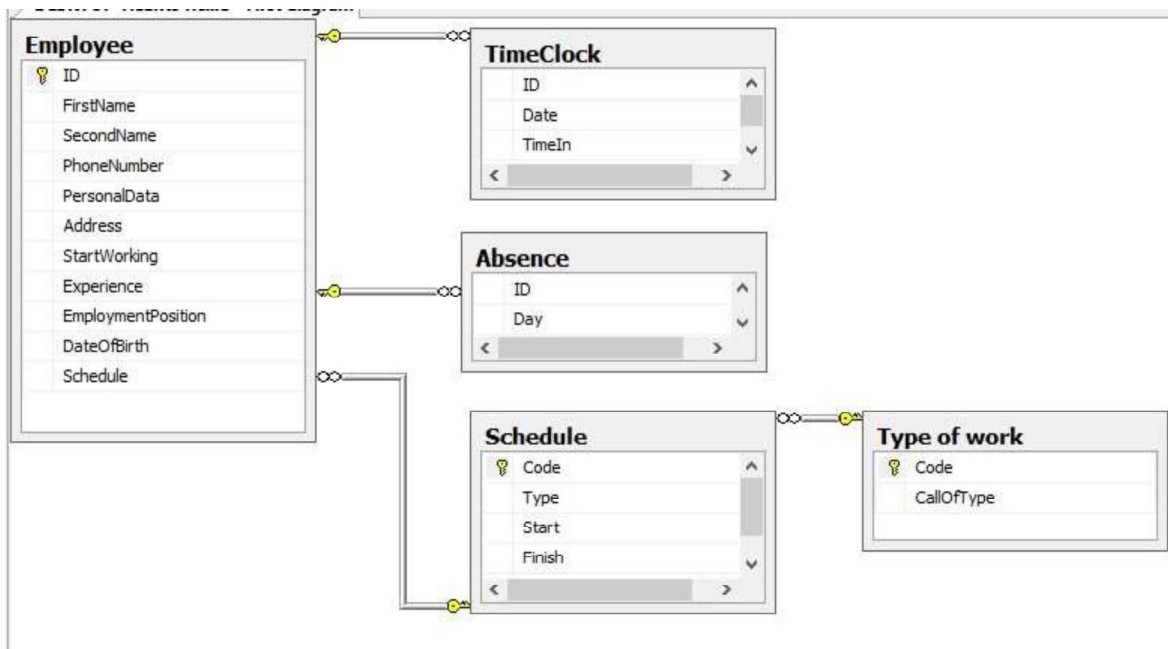


Рисунок 1 Диаграмма базы данных

Следует отметить, что зачастую пользователи не имеют прямого доступа к информации, хранящейся в информационных базах SQL Server, а работают с ней, используя сторонние программы, примером может служить Visual Studio. Способ работы с базой изображён на рисунке 2.

Так как, требовалось использование функции Transact – SQL для решения определённых задач, то сначала они были реализованы в среде SQL Server, а потом в Visual Studio. Пример функции T – SQL, создающей таблицу рабочих дней предприятия, приведён на рисунке 3.

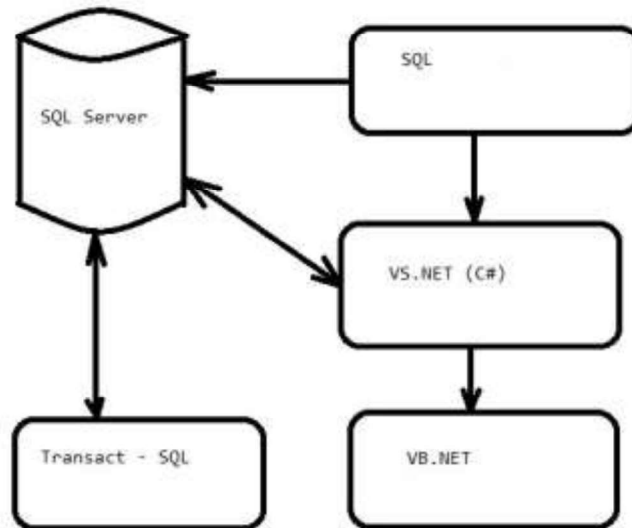


Рисунок 2 Схема взаимодействия программных средств

```

CREATE PROCEDURE popduedates AS
SET NOCOUNT ON
DECLARE @year integer, @insertday datetime

SELECT @year = YEAR(GETDATE()), @insertday = CAST(@year AS char(4)) + '0101'
TRUNCATE TABLE DUEDATES
WHILE YEAR(@insertday) = @year BEGIN
  IF ((SELECT DATEPART(dw, @insertday)) NOT IN (6,7))
  AND NOT EXISTS (SELECT * FROM HOLIDAYS WHERE HolidayDate = @insertday)
    INSERT DUEDATES VALUES (@insertday)
  SET @insertday = @insertday+1
END
  
```

Рисунок 3 Функция T-SQL

В приложении Visual Studio 2015 было разработано приложение типа Windows Forms, содержащие несколько форм. Перед началом создания форм следует задуматься над тем, как будет подключаться база SQL Server к приложению. Как правило, для этой цели применяется два способа: добавление источника данных в приложении, без использования программного кода, или обращение непосредственно при определённом действии, прописывая программный код. В работе использовались оба способа. Так, например, при работе пользователя на форме «Main», структура формы изображена на рисунке 4, необходимо заносить новые значения в таблицы и вычислять их, что осуществляется с помощью Transact – SQL. Для удобства используется средство

DateGridView, которое отображает данные, хранящиеся в таблице о сотрудниках. Фрагмент программного кода для кнопки «Start» приведён на рисунке 5.

Плюсы данной разработки заключаются в том, что она не узконаправленная, то есть, может работать на любом предприятии или компании. Также к плюсам можно отнести её достаточно простой и удобный интерфейс и то, что она не требует больших усилий при установке.



Рисунок 4 Структура формы "Main"

```
private void button1_Click(object sender, EventArgs e)
{
    //Пропущен кусок кода
    //Создание соединения с базой
    using (SqlConnection cs = new
SqlConnection(Properties.Settings.Default.conferenceConnectionString))
    {
        cs.Open();
        //Создание команды к базе
        SqlCommand cmd = cs.CreateCommand();
        cmd.CommandText = @"Select ID from Employee WHERE SecondName = @LastName";
        cmd.Parameters.AddWithValue("@LastName", LastName);
        //Выполнение команды
        object o = cmd.ExecuteScalar();
        if (o != null)
            if (o != DBNull.Value)
                strValue = o.ToString();
        cmd.CommandText = @"INSERT TimeClock (ID, Date, TimeIn) VALUES (@ID, @Date,
@TimeIn)";
        cmd.Parameters.AddWithValue("@ID", strValue);
        cmd.Parameters.AddWithValue("@Date", dateTimePicker1.Value.Date);
        cmd.Parameters.AddWithValue("@TimeIn", label6.Text);
        object c = cmd.ExecuteScalar();
        if (c != null)
            if (c != DBNull.Value)
                strValue = c.ToString();
    }
}
```

Рисунок 5 Фрагмент программного кода, создающего соединение с БД

В дальнейшем планируется сделать возможным реализацию данной функции посредством электронного пропуска.

#### Список использованных источников

1. Албахари Джозеф, Албахари Бен С# 7.0 Карманный справочник. – Пер. с англ. – СПб: ООО «Альфа - книга», 2017. – 224 с.



2. Кляйн Кевин, Кляйн Дэниел, Хант Бренд SQL.Справочник, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2010. – 656с.
3. Хендерсон Кен Профессиональное руководство по Transact - SQL – СПб.: Питер,2005. - 558 с.

47.35.01.

## ИСПОЛЬЗОВАНИЕ ЭМУЛЯТОРА ДЛЯ НАПИСАНИЯ КВАНТОВЫХ ПРОГРАММ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

**Автор:** Солод Борис Евгеньевич, студент первого курса Филиала «Протвино» государственного университета «Дубна»

**Научный руководитель:** к.т.н., доцент кафедры Кульман Татьяна Николаевна

### Аннотация

В работе рассматриваются квантовые компьютеры и квантовые вычисления. Кратко изложены основные понятия квантовых компьютеров, рассмотрены языки квантового программирования, эмуляторы квантовых вычислений. Реализован вариант программы для персонального компьютера (ПК) с помощью эмулятора квантового компьютера.

### Annotation

The paper deals with quantum computing and quantum computers. The basics of quantum computing are briefly described, programming languages for quantum computer, emulators of quantum computing are considered. Implemented a version of the program for the quantum computer emulator.

**Ключевые слова:** Квантовый компьютер, кубит, условия существования кубита, языки квантового программирования, язык Q#, эмулятор квантового компьютера.

**Keywords:** *Quantum computer, qubit, condition of existence qubit, quantum programming languages, language Q#, emulator quantum computer*

Целью работы является изучение квантовых компьютеров и языков квантового программирования, установка эмулятора квантового компьютера на домашнем компьютере, создание программы на языке квантового программирования Q#.

### Квантовый компьютер

Квантовый компьютер – вычислительное устройство, которое использует явления квантовой механики (квантовая суперпозиция, квантовая запутанность) для передачи и обработки данных. Не все операции можно выполнить с помощью квантового компьютера из-за ограничений квантовой физики.

Qubit (кубит) – квантовый бит. Классический бит, используемый в привычных всем компьютерах, может принимать значения либо "0", либо "1". У квантового бита в принципе нет определенного значения до того момента, как мы его измерили. Принято говорить, что он находится в определенном "состоянии". У кубита два состояния,  $|0\rangle$  и  $|1\rangle$  (эти скобки ученые договорились использовать, чтобы отличать состояния от обычных чисел 0 и 1). В общем случае кубит находится не в одном из этих основных состояний, а в некоторой их суперпозиции  $\alpha|0\rangle + \beta|1\rangle$ , которая и дает изображенную на рисунке 1 сферу.

То есть в момент измерения кубит может быть равен любому числу, удовлетворяющему условию  $a^2 + b^2 = 1$ . Например  $0,64 + 0,36 = 1$ .