09.02.03 УП, ПР-54

КУРСОВОЙ ПРОЕКТ

Разработка информационной системы   
«Регистрация продаж для булочной»

**по профессиональному модулю**

**ПМ.02 «Разработка и администрирование баз данных»**

Пояснительная записка

Выполнил Д.В.Волынчикова

|  |
| --- |
| М.А. Бодарева |

Преподаватели

**Содержание**

[Введение 3](#_Toc467878341)

[1 Аналитическая часть 4](#_Toc467878342)

[1.1 Описание предметной области 4](#_Toc467878343)

[1.2 Цели и задачи проекта 4](#_Toc467878344)

[1.3 Описание функциональности системы 5](#_Toc467878345)

[1.4 Классификация пользователей проекта 6](#_Toc467878346)

[1.5 Описание логической модели данных 6](#_Toc467878347)

[2 Технологическая часть 8](#_Toc467878348)

[2.1 Описание физической модели данных 8](#_Toc467878349)

[2.2 Реализация бизнес-логики серверной части приложения БД 10](#_Toc467878350)

[2.3 Описание уровня представления системы 16](#_Toc467878351)

[2.4 Описание тестирования системы; 21](#_Toc467878352)

[2.5 Описание системы безопасности 23](#_Toc467878353)

[2.6 Описание системы администрирования базы данных 23](#_Toc467878354)

[Заключение 25](#_Toc467878355)

[Использованные информационные источники 26](#_Toc467878356)

# **Введение**

Целью данного курсового проекта было создание информационной системы для булочной.

Количество различных точек быстрого питания стремительно растет. Любой организации занимающейся торговлей необходимо регистрировать произведенные продажи, а также вести учет движений товаров, хранить список продавцов и многое другое.

Для обширных сетей нет иного выхода, как хранить информацию на сервере, а для доступа к нему не обойтись без соответствующего приложения.

Разработанная система даст пользователю возможность работать с необходимыми ему данными.

Основанием для разработки проекта является решение ЦМК ПОИТ от 31 августа 2014 года №1 об утверждении тем курсовых проектов по профессиональному модулю «Разработка и администрирование баз данных» специальности 09.02.03 (углубленная подготовка) «Программирование в компьютерных системах»

1. **Аналитическая часть**

## Описание предметной области

Фирма занимается производством и продажей булочных изделий. Имеется ассортимент изделий (название, стоимость). Клиент может, как купить изделие, так и сделать отсроченный по времени заказ на изделие. Одна покупка/заказ (дата, итоговая стоимость) может включать несколько изделий в разном количестве. В случае заказа необходимо регистрировать дату, время изготовления.

Разработанная система должна хранить информацию о сырье, изделиях.

Система должна вести оформление продажи или заказа на изделия. Система должна обеспечивать поиск по различным критериям, а также создавать статистические отчеты в различных разрезах.

Разработанная система должна иметь систему безопасности, предоставляя доступ пользователям к объектам базы данных в соответствии с назначенными ролями. Также система должна иметь план обслуживания базы данных.

## Цели и задачи проекта

*Цели:*

* систематизация и закрепление полученных теоретических знаний и практических умений по междисциплинарным курсам профессионального модуля «Разработка и администрирование баз данных»;
* проверка овладения профессиональными компетенциями, сформированными в рамках изучения модуля;
* проверки овладения общими компетенциями, сформированными в рамках изучения модуля;
* развитие творческой инициативы, самостоятельности, ответственности и организованности.

*Задачи:*

* изучение и анализ особенностей конкретной предметной области, относящихся к теме курсового проекта;
* выбор и проектирование модели данных с использованием автоматизированных программных средств;
* выбор эффективных технологий разработки приложений баз данных;
* проектирование и реализация объектов серверной части приложения в среде SQL SERVER;
* создание и тестирование прототипа клиентской части приложения в среде Visual Studio;
* анализ возможных подходов и методов защиты данных с обоснованием выбранного метода;
* создание системы безопасности автоматизированной системы;
* разработка системы автоматизированного администрирования информационной системы;
* разработка программной документации.

## Описание функциональности системы

После изучения предметной области была выявлена необходимость в создании хранимых процедур для СУБД, позволяющие изменять, удалять, вставлять и отображать различные данные (заказов, клиентов, менеджеров, булочных изделий, их цен и видов).

Основным пользователем приложения будет считаться продавец-менеджер. Его главной задачей будет запись заказов с возможностью добавления клиентов для отсроченных по времени заказов.

Также необходимо предусмотреть роль, ответственную за меню, цены и персонал.

## Классификация пользователей проекта

В таблице 1 представлены созданные роли и их права на различные объекты базы данных.

Таблица - Группы пользователей и их права

|  |  |
| --- | --- |
| *Группы пользователей* | *Права групп пользователей* |
| Менеджер | Менеджер работает с заказами и клиентами, поэтому имеет полный доступ к таблицам: «Заказы», «Клиенты», «Заказы на изделия». Остальные таблицы (все, кроме «Менеджеры») он может только просмотреть. |
| Владелец заведения | Ответственен за булочные изделия, которые будут в продаже и их цены, поэтому она контролирует таблицы «Изделия», «Виды изделий», «Цены изделий», также учитывается прием на работу сотрудников (таблица «Менеджеры»). Остальные таблицы можно только просмотреть. |

## Описание логической модели данных

В качестве структуризации данных была создана концептуальная ER-диаграмма, отображающая семь основных сущностей (рис. 1).

Для хранения данных о товаре используются сущности «Виды изделий», «Изделия» и «Цены изделий». Менеджеры и клиенты были записаны в отдельные сущности.

При связывании таблиц сущность «Заказы» была разбита на две сущности («Заказы», «Заказы на изделия») во избежание связи «многие-ко-многим».

В качестве ограничений целостности каждой таблице был присвоен первичный ключ, некоторым полям разрешено хранить пустые значения, а также была выявлена уникальность, чтобы не допустить повторение данных.

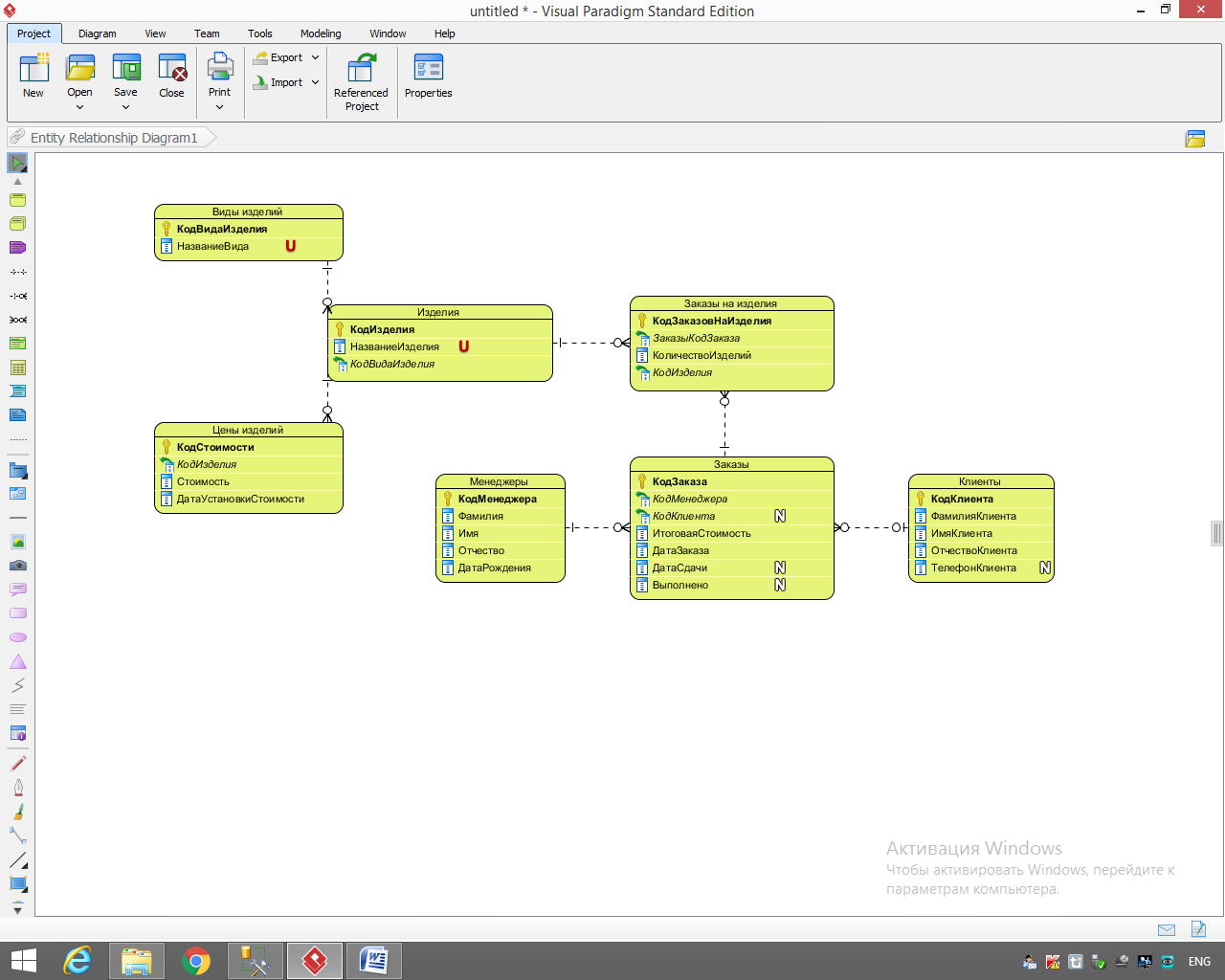


Рисунок - ER-диаграмма

1. **Технологическая часть**

## Описание физической модели данных

По созданной ER-диаграмме в среде SQL Server 2008 было реализовано семь таблиц:

* Buns –для записей о продаваемых булочных изделиях (таблица 2);
* OrdersBuns – содержит информацию о том, какие булочные изделия в каком заказе состоят (таблица 3);
* Clients – хранит данные о клиентах с отсроченными по времени заказами (таблица 4);
* TypesBuns - для записей о типах продаваемых булочных изделий (таблица 5);
* Managers - хранит данные о менеджерах, продающих товар (таблица 6);
* Orders – записывает основные данные о заказе (таблица 7);
* Price - для хранения цены товара (таблица 8);

Таблица - Структура таблицы Buns

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_buns | Первичный ключ таблицы Buns | int | - |
| id\_typeBuns | Внешний ключ таблицы TypesBuns | int | - |
| buns\_name | Название булочных изделий. Должно быть уникальным | varchar | 50 |

Таблица - Структура таблицы OrdersBuns

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_ordersBuns | Первичный ключ таблицы OrdersBuns | int | - |
| id\_buns | Внешний ключ таблицы Buns | int | - |
| id\_orders | Внешний ключ таблицы Orders | int | - |
| quantity | Количество булочных изделий | int | - |

Таблица - Структура таблицы Clients

| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| --- | --- | --- | --- |
| id\_clients | Первичный ключ таблицы Clients | int | - |
| client\_surname | Фамилия клиента | varchar | 50 |
| client\_name | Имя клиента | varchar | 50 |
| client\_patronymic | Отчество клиента | varchar | 50 |
| client\_phone | Телефон клиента | varchar | 50 |

Таблица - Структура таблицы TypesBuns

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_typesBuns | Первичный ключ таблицы TypesBuns | int | - |
| typeBuns\_name | Название типа булочного изделия. Должно быть уникальным | varchar | 50 |

Таблица - Структура таблицы Managers

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_managers | Первичный ключ таблицы Managers | int | - |
| manager\_surname | Фамилия менеджера | varchar | 50 |
| manager\_name | Имя менеджера | varchar | 50 |
| manager\_patronymic | Отчество менеджера | varchar | 50 |
| manager\_dateOfBirth | Дата рождения менеджера | date | - |
| manager\_login | Логин (имя входа) менеджера | varchar | 50 |

Таблица - Структура таблицы Orders

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_orders | Первичный ключ таблицы Orders | int | - |
| id\_clients | Внешний ключ таблицы Clients | int | - |
| id\_managers | Внешний ключ таблицы Managers | int | - |
| cost | Общая стоимость заказа | money | - |
| order\_date | Дата обращения клиента | date | - |
| deliveryTime | Дата сдачи заказа | date | - |
| done | Выполнение заказа | bit | - |

Таблица - Структура таблицы Price

|  |  |  |  |
| --- | --- | --- | --- |
| *Название поля* | *Описание* | *Тип поля* | *Размер* |
| id\_price | Первичный ключ таблицы Price | int | - |
| id\_buns | Внешний ключ таблицы Buns | int | - |
| price | Цена за ед. товара | money | - |
| price\_date | Дата установки цены | date | - |

## Реализация бизнес-логики серверной части приложения БД

Всего было создано 28 хранимых процедур, их названия, параметры и назначения представлены в таблице 9.

Таблица - Хранимые процедуры

| *Имя\_ Процедуры* | *Параметры* | *Назначение* |
| --- | --- | --- |
| Buns\_Delete | @bunsId int | Удаление записи о булочном изделии |
| Buns\_Insert | @typeBunsId int, @name nvarchar(50) | Вставка записи о булочном изделии |
| Buns\_Select | @bunsId int, @typeBunsId int | Вывод записей об одном или нескольких булочных изделий, возможен отбор по типу изделий |
| Buns\_Update | @bunsId int,@typeBunsId int,@name nvarchar(50) | Обновление записи о булочном изделии |
| Clients\_Delete | @clientId int | Удаление записи о клиенте |
| Clients\_Insert | @surname nvarchar(50), @name nvarchar(50), @patronymic nvarchar(50), @phone nvarchar(50) | Вставка записи о клиенте |
| Clients\_Select | @clientsId int | Вывод одной или несколько записей о клиенте |
| Clients\_Update | @clientsId int, @surname nvarchar(50), @name nvarchar(50), @patronymic nvarchar(50), @phone nvarchar(50) | Обновление записи о клиенте |
| Managers\_Delete | @managerId | Удаление записи о менеджере |
| Managers\_Insert | @surname nvarchar(50),  @name nvarchar(50), @patronymic nvarchar(50), @dateOfBirth date | Вставка записи о менеджере |
| Managers\_Select | @managerId int | Вывод одной или несколько записей о менеджере |
| Managers\_Update | @managersId int, @surname nvarchar(50), @name nvarchar(50), @patronymic nvarchar(50), @dateOfBirth date | Обновление записи о менеджере |
| Orders\_Delete | @orderId int | Удаление записи о заказе |
| Orders\_Insert | @managersId int, @clientsId int, @orderDate date, @cost money, @done bit, @deliveryTime date | Вставка записи о заказе |
| Orders\_Select | @orderId int | Вывод одной или несколько записей о заказе |
| Orders\_Update | @ordersId int,@managersId int, @clientsId int, @orderDate date, @cost money, @done bit, @deliveryTime date | Обновление записи о заказе |
| OrdersBuns\_Delete | @orderBunsId int, @ordersId int | Удаление записи о заказанном изделии |
| OrdersBuns\_Insert | @bunsId int, @ordersId int, @quantity int | Вставка записи о заказанном изделии |
| OrdersBuns\_Select | @ordersId int | Вывод одной или несколько записей о заказанном изделии с указанием последней цены |
| OrdersBuns\_Update | @ordersBunsId int, @bunsId int, @ordersId int, @quantity int | Обновление записи о заказанном изделии |
| Price\_Delete | @priceId int | Удаление записи о цене |
| Price\_Insert | @bunsId int, @cost money, @dateInstallationCost date | Вставка записи о цене |
| Price\_Select | @priceId int, @bunsId int | Вывод одной или нескольких записей о цене, возможен отбор по конкретному изделию |
| Price\_Update | @priceId int, @bunsId int, @cost money, @dateInstallationCost date | Обновление записи о цене |
| TypesBuns\_Delete | @typeBunsId int | Удаление записи о типе изделия |
| TypesBuns\_Insert | @typeBuns nvarchar(50) | Вставка записи о типе изделия |
| TypesBuns\_Select | @typeBunsId int | Вывод одной или нескольких записей о типе изделия |
| TypesBuns\_Update | @typeBunsId int, @typeBuns nvarchar(50) | Обновление записи о типе изделия |

Код хранимых процедур Buns\_Select и Orders\_Update представлен в листингах 1 и 2.

Листинг 1

CREATE PROCEDURE [dbo].[Buns\_Select] (@bunsId INT = NULL,

@typeBunsId INT = NULL)

AS BEGIN

SET NOCOUNT ON

SELECT dbo.Buns.id\_buns AS [№],dbo.Buns.buns\_name AS [Наименование], dbo.TypesBuns.typeBuns\_name AS [Тип изделия]

FROM dbo.Buns LEFT OUTER JOIN dbo.TypesBuns ON dbo.TypesBuns.id\_typesBuns=dbo.Buns.id\_typeBuns

WHERE (@bunsId IS NULL OR dbo.Buns.id\_buns = @bunsId)

AND (@typeBunsId IS NULL OR dbo.TypesBuns.id\_typesBuns = @typeBunsId)

END

Листинг 2

CREATE PROCEDURE [dbo].[Orders\_Update] (@ordersId INT, @managersId INT, @clientsId INT = NULL, @orderDate DATE, @cost MONEY, @done BIT = NULL, @deliveryTime DATE = NULL)

AS BEGIN

SET NOCOUNT ON

UPDATE [Orders]

SET id\_clients = @clientsId, id\_managers = @managersId,

order\_date = @orderDate, cost=@cost, deliveryTime = @deliveryTime, done = @done

WHERE id\_orders = @ordersId

RETURN @@ERROR

END

Для автоматизирования некоторых задач были созданы четыре триггера. Подробности о них можно посмотреть в таблице 10*.*

Таблица - Триггеры

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя\_Триггера* | *Тип* | *Объект\_привязки* | *Назначение* |
| OrdersTrInsert | after insert | Таблица Orders | Ограничение целостности дат на вставку записи. Дата сдачи заказа не должна быть раньше самого заказа |
| OrdersTrUpdate | after update | Таблица Orders | Ограничение целостности дат на обновление записи. Дата сдачи заказа не должна быть раньше самого заказа |
| OrdersBunsTrInsert | after insert | Таблица OrdersBuns | Обновление общей стоимости заказа из таблицы Orders при вставке записи в таблицу OrdersBuns |
| OrdersBunsTrUpdate | after update | Таблица OrdersBuns | Обновление общей стоимости заказа из таблицы Orders при обновлении записи в таблице OrdersBuns |

Код тригеров OrdersTrInsert и OrdersBunsTrUpdate представлены в листингах 3 и 4.

Листинг 3

CREATE TRIGGER [dbo].[OrdersTrInsert] ON [dbo].[Orders] AFTER INSERT

AS BEGIN

IF EXISTS (SELECT \*

FROM dbo.Orders INNER JOIN inserted

ON orders.id\_orders = inserted.id\_orders

WHERE orders.order\_date > inserted.deliveryTime)

BEGIN

PRINT 'Дата сдачи заказа должна быть позже даты совершения заказа'

ROLLBACK TRANSACTION

END

ELSE

BEGIN TRANSACTION

UPDATE dbo.Orders

SET orders.deliveryTime = inserted.deliveryTime

FROM dbo.Orders INNER JOIN inserted

ON orders.id\_orders =inserted.id\_orders

COMMIT TRANSACTION

END

Листинг 4

CREATE TRIGGER [dbo].[OrdersBunsTrInsert] ON [dbo].[OrdersBuns] AFTER INSERT

AS BEGIN

CREATE TABLE #CostTable (id\_costTable INT ,priceCost INT )

INSERT #CostTable

SELECT dbo.OrdersBuns.id\_orders,

SUM(dbo.OrdersBuns.quantity \* dbo.Price.price) AS priceOrder

FROM dbo.OrdersBuns INNER JOIN dbo.Price

ON dbo.OrdersBuns.id\_buns = dbo.Price.id\_buns

WHERE dbo.Price.price\_date = (SELECT MAX(price\_date) AS maxDate

FROM dbo.Price AS price2

WHERE (id\_buns = dbo.Price.id\_buns))

GROUP BY dbo.OrdersBuns.id\_orders

UPDATE dbo.Orders

SET orders.cost = #CostTable.priceCost

FROM dbo.Orders INNER JOIN #CostTable

ON dbo.Orders.id\_orders = #CostTable.id\_costTable

END

Также было создано представление для отображения последней цены конкретного булочного изделия, его код представлен в листинге 5.

Листинг 5

SELECT dbo.Price.price AS Цена, dbo.Buns.id\_buns

FROM dbo.Price INNER JOIN dbo.Buns ON dbo.Price.id\_buns = dbo.Buns.id\_buns

WHERE (dbo.Price.price\_date =

(SELECT MAX(price\_date) AS maxDate

FROM dbo.Price AS price2

WHERE (id\_buns = dbo.Price.id\_buns)))

GROUP BY dbo.Price.id\_buns, dbo.Price.price, dbo.Buns.id\_buns

## Описание уровня представления системы

Авторизация в системе предусмотрена с помощью простой формы, отображенной на рисунке 2.

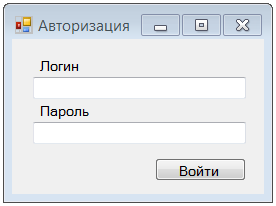
**

Рисунок - Форма авторизации

После входа в систему открывается главная форма, которая является родительской по отношению к остальным формам. В верхнем меню находятся кнопки для удобного вызова необходимых форм (рисунок 3).

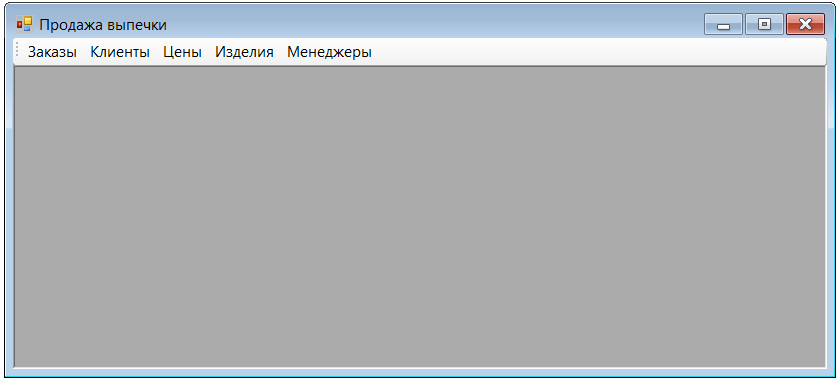
**

Рисунок - Главная форма

Продаваемые булочные изделия хранятся на одной форме с их типами, но разных вкладках (рисунок 4).

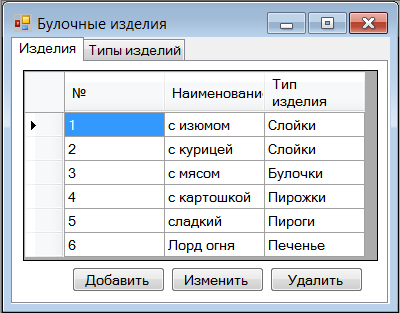
**

Рисунок - Форма булочных изделий

Форма с клиентами отображает данные о них (рисунок 5), а по нажатию на кнопки добавления/изменения вызывает форму для записи/обновления (рисунок 6).

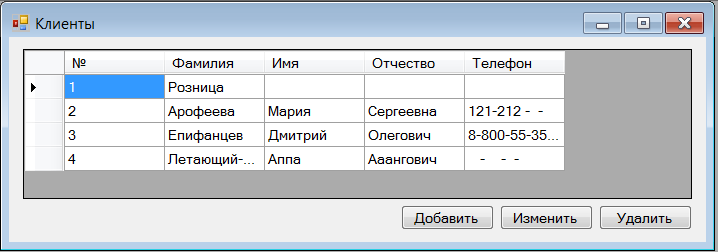
**

Рисунок - Форма клиентов

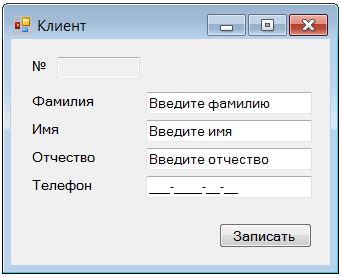
**

Рисунок - Форма добавления, обновления клиента

Список цен отображается на форме, изображенной на рисунке 7. Форма добавления/обновления записи представлена на рисунке 8.

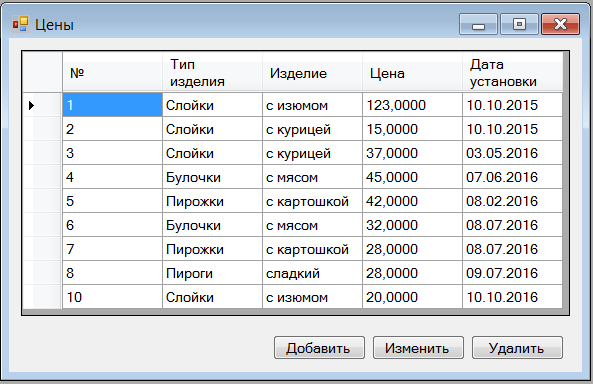
**

Рисунок - Форма цен

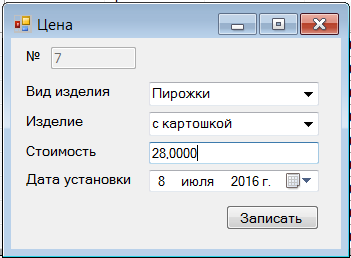
**

Рисунок - Форма вставки, обновления цены

Форма для отображения заказов и их добавления/обновления представлены на рисунках.

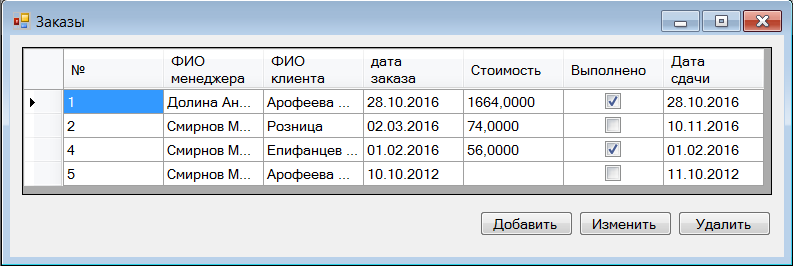
**

Рисунок - Форма заказов

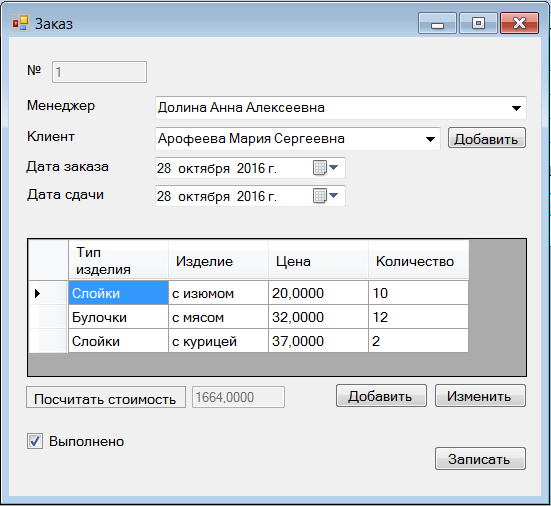
**

Рисунок - Форма вставки, добавления заказа

## Описание тестирования системы;

Сценарии тестирования и полученные результаты основных хранимых процедур и триггеров отображены в таблице 11.

Таблица - Тестирование объектов БД

| *Имя объекта тестирования* | *Параметры* | *Ожидаемый результат* | *Результат* |
| --- | --- | --- | --- |
| *Хранимые процедуры* | | | |
| Price\_Insert | @bunsId = 1, @cost = 20, @dateInstallationCost = '10-10-16' | Вставка новой записи | Вставка новой записи |
| Price\_Insert | @bunsId = 2,  @cost = 26, @dateInstallationCost = 10-10-2016 | Ошибка синтаксиса | Сообщение 102. Неправильный синтаксис около конструкции "-".  Выполнение данной инструкции было прервано. |
| Price\_Insert | @bunsId = 50, @cost = 20, @dateInstallationCost = '10.10.16' | Ошибка, т.к. записи с @bunsId = 50 не существует | Сообщение 547. Конфликт инструкции INSERT с ограничением FOREIGN KEY"FK\_\_Price\_\_id\_buns\_\_1273C1CD".  Выполнение данной инструкции было прервано. |
| OrdersBuns\_Update | @ordersBunsId = 3, @bunsId = 2,  @ordersId = 1,  @quantity = N'два' | Ошибка при преобразовании  типов данных | Сообщение 8114.Ошибка при преобразовании типа данных nvarchar к int.  Выполнение данной инструкции было прервано. |
| Clients\_Delete | @clientId = 6 | Удаление записи. | Удаление необходимой записи. |
| TypesBuns\_Insert | @typeBuns = N'Пирожки' | Ошибка при нарушении ограничения UNIQUE | Сообщение 2627.Нарушение "UQ\_\_TypesBun\_\_5CCDE1F9023D5A04" ограничения UNIQUE KEY. Не удается вставить повторяющийся ключ в объект "dbo.TypesBuns".  Выполнение данной инструкции было прервано. |
| Managers\_Update | @managersId = 3,@surname = N'Петров', @name = N'Павел',  @patronymic = N'Панфилович',  @dateOfBirth =  '1979-08-14' | Обновление записи | Обновление записи |
| *Триггеры* | | | |
| OrdersTrInsert | @managersId = 2, @clientsId = NULL, @orderDate ='10/10/2016', @cost = NULL,  @done = False, @deliveryTime = '10/10/2010' | Ошибка, выданная триггером OrdersTrInsert | Сообщение: «Дата сдачи заказа должна быть позже даты совершения заказа»  Сообщение 3902. Запрос COMMIT TRANSACTION не имеет соответствующей инструкции BEGIN TRANSACTION.  Выполнение данной инструкции было прервано. |

## Описание системы безопасности

Созданные роли базы данных и их доступ к объектам отображены в таблице 12.

Таблица - Роли пользовавателей и их доступ к объектам БД

|  |  |  |
| --- | --- | --- |
| *Объект базы данных* | *Роли пользователей* | |
| *Director* | *Manager* |
| Buns (таблица «Изделия») | Обновление, добавление, удаление, изменение | Выборка |
| Clients (таблица «Клиенты») | Выборка | Обновление, добавление, удаление, изменение |
| Orders (таблица «Заказы») | Выборка | Обновление, добавление, удаление, изменение |
| OrdersBuns (таблица «Заказы на изделия») | Выборка | Обновление, добавление, удаление, изменение |
| Price (таблица «Цены изделий») | Обновление, добавление, удаление, изменение | Выборка |
| TypesBuns (таблица «Типы изделий») | Обновление, добавление, удаление, изменение | Выборка |
| Managers (таблица «Менеджеры») | Обновление, добавление, удаление, изменение | - |

## Описание системы администрирования базы данных

Для графика обслуживания базы данных “SalesBuns” были созданы два плана с разным расписанием:

1. subplan\_1, который осуществляет резервное копирование журнала транзакций и проверку целостности БД по расписанию ниже:
   * тип расписания: повторяющееся;
   * частота выполнения: ежедневно, каждые два часа;
   * время выполнения: с 08:00:00 до 20:00:00;
   * дата окончания плана: не указана;
2. subplan\_2, который осуществляет разностное резервное копирование БД по расписанию ниже:
   * тип расписания: повторяющееся;
   * частота выполнения: ежедневно;
   * время выполнения: в 20:00:00;
   * дата окончания плана: не указана.

Проект графика обслуживания отображен на картинках

C:\Users\Диана\AppData\Roaming\Skype\volynchikova97\media_messaging\media_cache_v3\^3F042DA1F94B340F1D8B3B80375531F79A6D739EC9D2E9A7A7^pimgpsh_fullsize_distr.png

Рисунок - Краткое описание плана обслуживания

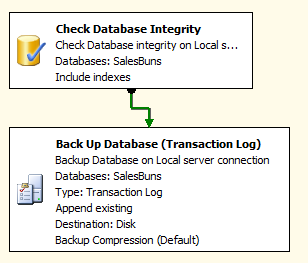


Рисунок - Графическое отображение Subplan\_1

# **C:\Users\Диана\AppData\Roaming\Skype\volynchikova97\media_messaging\media_cache_v3\^A71D2253F20AC068BCE8B0069B482107F6B2E96E85B891CABC^pimgpsh_fullsize_distr.png**

Рисунок - Графическое отображение Subplan\_2

# **Заключение**

В результате работы была спроектирована и реализована информационная система для регистрации продаж булочной.

Средствами SQL Server 2008 была создана база данных “SalesBuns”, содержащая 7 таблиц, 28 хранимых процедур, 4 триггера и 1 представление. Для нее были спланированы задания по обслуживанию базы данных в целях защиты данных от потерь. Для взаимодействия с базой средствами Visual Studio 2013 было создано клиентское приложение. Оно позволяет пользователям авторизоваться в нем, работать с данными в базе (просматривать, добавлять, изменять, удалять). В соответствии с определенными правами доступа пользователям ограничен некоторый функционал приложения (например, менеджер не может даже просмотреть данные о других менеджерах).

Созданная информационная система удовлетворяет поставленному заданию. Процесс разработки и реализации системы можно просмотреть в предоставленной пояснительной записке. В ней отображены основные этапы и ключевые моменты.

Помимо закрепления и систематизации знаний, полученных ранее при изучении модуля «Разработка и администрирование баз данных», появился опыт написания курсового проекта, что, несомненно, поможет при написании диплома.

**Использованные информационные источники**

1. Дьюсон Р. SQL Server 2008 для начинающих разработчиков. - СПб.: БХВ-Петербург, 2009. - 704 с.
2. Microsoft SQL Server 2008. Реализация и обслуживание. Учебный курс Microsoft / Пер.с англ. – М.: Издательско-торговый дом «Русская редакция»; СПб.: Питер, 2011. – 576 с.
3. Петкович Д. Microsoft SQL Server 2008. Руководство для начинающих: Пер. с англ. - СПб.: БХВ-Петербург, 2012. - 752 с.
4. Дейт К. Дж. Введение в системы баз данных. 8-е изд.: Пер. с англ. – К.; М.; СПб.: Издательский дом «Вильямс», 2008. – 1328 с.: ил.
5. Конноли Т., Бегг К., Страчан А. Базы данных: Проектирование, реализация и сопровождение. Теория и практика, 2-е изд. – М.: Издательский дом «Вильямс», 2003. – 1440 с.: ил.
6. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление.- 5-е изд., перераб. и доп.: Пер. с англ. – СПб.:БХВ-Петербург, 2004.–1040 с.: ил.
7. Стивенс Р. Программирование баз данных.: Пер. с англ. – М.: ООО «Бином-Пресс», 2003. – 384 с.: ил.