Министерство цифрового развития государственного управления,

информационных технологий и связи Республики Татарстан

государственное автономное профессиональное образовательное учреждение

«Международный центр компетенций –

Казанский техникум информационных технологий и связи»

ОТЧЕТ

ПО УЧЕБНОЙ ПРАКТИКЕ

ПМ.01 УП.01.01«Разработка, администрирование и защита БД»

Выполнил обучающийся Никифоренко Артур Дмитриевич

(ФИО)

группа 321 специальность 09.02.07 «Информационные системы и программирование. Квалификация: Программист»

Оценка \_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

Начало практики: 10.11.2022 г.              \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Музафаров Б.Р.

(Ф.И.О. подпись руководителя практики)

Окончание практики: 23.11.2022 г.            \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Музафаров Б.Р.

(Ф.И.О. подпись руководителя практики)

М.П.

Казань, 2022г.

СОДЕРЖАНИЕ

[1 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ 3](#_Toc120046180)

[1. 1 СУБД 3](#_Toc120046181)

[2. РАЗРАБОТКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 7](#_Toc120046182)

[2.1 Создание базы данных 7](#_Toc120046183)

[2.2 Создание таблиц 7](#_Toc120046184)

[2.3 Создание ER диаграммы 9](#_Toc120046185)

[2.4 Работа с таблицей 10](#_Toc120046186)

[2.5 Создание триггеров 11](#_Toc120046187)

[2.6 Создание представления 12](#_Toc120046188)

[2.7 Создание функции 13](#_Toc120046189)

[2.8 Создание резервной копии БД 14](#_Toc120046190)

[ЗАКЛЮЧЕНИЕ 16](#_Toc120046191)

# 1 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ

## 1 СУБД

База данных — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных. Большинство сайтов не могут функционировать без базы данных, поэтому СУБД используется практически повсеместно.

Основные функции СУБД:

* управление данными во внешней памяти (на дисках);
* управление данными в оперативной памяти с использованием дискового кэша;
* журнализация изменений (сохранение истории), резервное копирование и восстановление базы данных после сбоев;
* поддержка языков БД (язык определения данных, язык манипулирования данными).

Способы доступа к БД бывают:

* Клиент-серверные СУБД;
* Файл-серверные СУБД;
* Встраиваемые СУБД.

В клиент-серверных СУБД (Microsoft SQL Server, Oracle, Firebird, PostgreSQL, InterBase, MySQL и др.)

* Вся обработка данных ведётся в одном месте, на сервере, в том же месте, где хранятся (обычно) данные;
* К файлам данных имеет доступ только один сервер, одна система — это сама СУБД;
* Приложения-клиенты посылают запросы на обработку и получение данных из СУБД и получают ответы;
* Приложения-клиенты не имеют непосредственного доступа к файлам данных.

Все промышленные СУБД на данный момент являются именно клиент-серверными.

В файл-серверных СУБД (Paradox, Microsoft Access, FoxPro, dBase и др.), наоборот:

* Приложения имеют общий доступ ко всем файлам базы данных (хранящимся обычно в каком-то разделяемом файловом хранилище) и совместно обрабатывают эти данные;
* Каждое приложение самостоятельно обрабатывает данные.

На данный момент файл-серверная технология считается устаревшей, а её использование в крупных информационных системах — недостатком. Проблема в том, что файл-серверные СУБД не имеют многих преимуществ клиент-серверных, таких как кэширование данных, параллелизм запросов, высокая производительность и обладают рядом недостатков (сложности с поддержанием целостности базы, восстановлением, блокировками и т.д.), что приводит в свою очередь к пониженной надёжности и производительности. Состояние базы в файловых СУБД необходимо постоянно отслеживать и проводить операции по её «лечению» с помощью встроенных или сторонних утилит.

Встраиваемые СУБД (SQLite, Firebird Embedded, Microsoft SQL Server Compact и др.)

* Поставляются в составе готового программного продукта, не требуя процедуры самостоятельной установки;
* Предназначены для локального хранения данных приложения и не рассчитаны на коллективное использование в сети.

Встраиваемая бесплатная СУБД SQLite широко используется в известной мобильной ОС Android, разработанной в компании Google, и во многих мобильных приложениях.

Каждая СУБД основывается на какой-либо модели данных, это является одним из признаков классификации. По модели данных СУБД бывают:

* Иерархические. В этой модели данных используется представление БД в виде древовидной структуры, состоящей из данных разных уровней;
* Сетевые. Данная модель является расширением иерархического подхода. Иерархическая модель подразумевает, что запись-потомок может иметь строго одного предка, в то время как в сетевой структуре потомок может иметь любое количество предков;
* Реляционные. СУБД, ориентированные на организацию данных как набор связанных записей и атрибутов в двумерной таблице;
* Объектно-ориентированные. Для управления БД, основанными на объектной модели данных. Как правило основываются на объектно-ориентированных языках программирования;
* Объектно-реляционные. Объединяет в себе концепции реляционной модели с дополнительными объектно-ориентированными возможностями.

Сегодня по-прежнему наиболее популярными при создании веб-приложений и сервисов остаются реляционные базы данных. Для управления реляционными базами данных используется язык SQL (Structured Query Language — структурированный язык запросов). Изначально SQL был инструментом работы пользователя с базой данных, однако со временем язык усложнился и стал скорее инструментом разработчика, чем конечного пользователя.

Различные рейтинги самых популярных СУБД возглавляют Oracle, MySQL, Microsoft SQL Server, PostgreSQL.

# РАЗРАБОТКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

Дана предметная область: Хлебопекарня

Хлебопекарня занимается производством хлеба и хлебобулочных изделий, которые выпекаются в специальном оборудовании — печи. Готовый хлеб развозится по различным торговым точкам города, с которыми у хлебопекарни заключен долгосрочный договор на поставку хлебобулочных изделий. Также любое физическое или юридическое лицо может сделать предварительный заказ на выпечку большой партии изделий на некоторое мероприятие. Хлебопекарня, в зависимости от объема хлебобулочных изделий для торговых точек и наличия предварительных заказов, закупает у поставщиков соответствующий объем сырья и материалов, а также составляет график работы персонала. По результатам своей деятельности хлебопекарня производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики.

## Создание базы данных

Дана база данных под названием «BakeryShop».



Рисунок 2.1.1 Создание базы данных «BakeryShop»

## Создание таблиц

Дальше в БД при помощи команды CREATE TABLE были созданы таблицы «Product», «Row», «Cooking», «Treaty», «Outlet», «Delivery\_Treaty», «Order\_», «Order\_Status», «Order\_Products». После создания они были заполнены данными при помощи выражения INSERT INTO. Примеры создания таблиц изображены на рисунках 2.2.1-2.2.4:

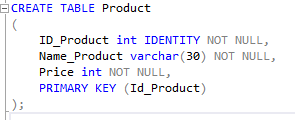


Рисунок 2.2.1 Таблица «Product»

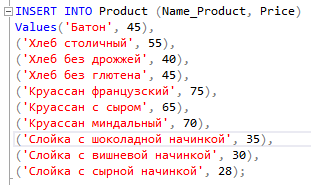


Рисунок 2.2.2 Заполнение данными таблицы «Product»

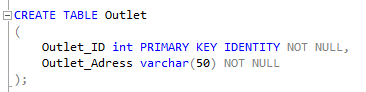


Рисунок 2.2.3 Таблица «Outlet»

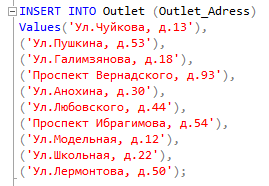


Рисунок 2.2.4 Заполнение данными таблицы «Outlet»

## Создание ER диаграммы

После создания таблиц была сформирована диаграмма:

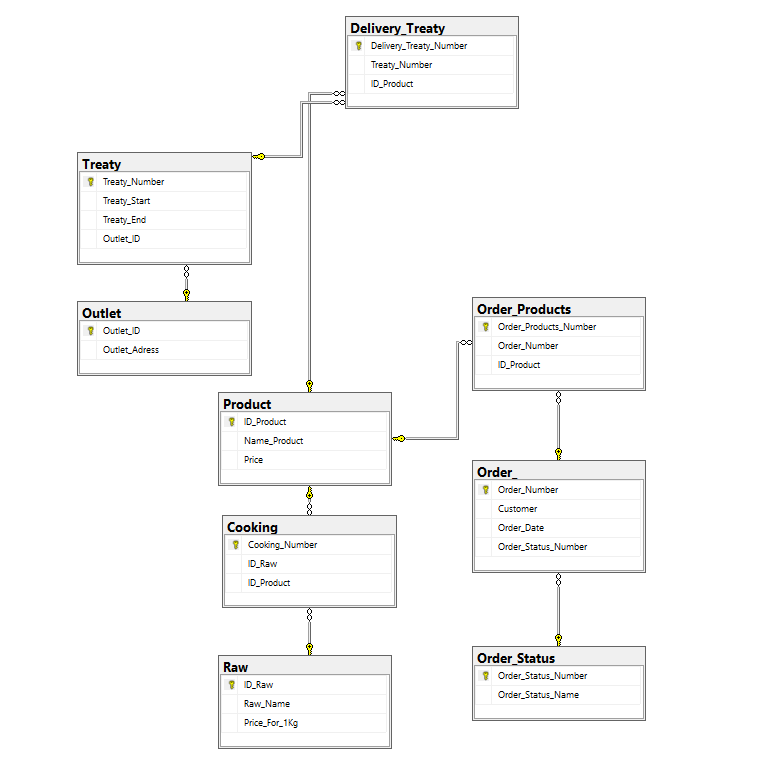


Рисунок 2.3.1 Диаграмма БД

## Работа с таблицей

В индивидуальном задании была дана задача на создание запросов для редактирования структуры таблиц.

Для выполнения этой задачи можно воспользоваться выражением ALTER TABLE. С помощью ALTER TABLE можно провернуть самые различные сценарии изменения таблицы.

В таблицу «Product» был добавлен новый столбец под названием «Description»:



Рисунок 2.4.1 Добавление столбца «Description» в таблицу «Product»

Получен итоговый результат:

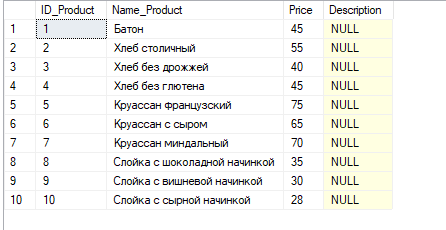


Рисунок 2.4.2 Результат изменения таблицы «Product»

## Создание триггеров

Триггеры представляют специальный тип хранимой процедуры, которая вызывается автоматически при выполнении определенного действия над таблицей или представлением, в частности, при добавлении, изменении или удалении данных, то есть при выполнении команд INSERT, UPDATE, DELETE.

Дана задача на создание триггеров, которые будут вызываться при добавлении, удалении и изменении заказов.

Была создана таблица «Order\_History», в которой будут храниться операции по удалению, добавлению и изменению заказов.

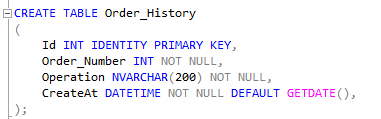


Рисунок 2.5.1 Создание таблицы «Order\_History»

Далее были созданы триггеры:

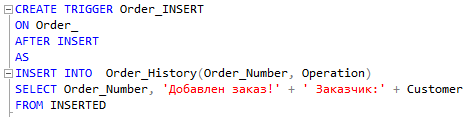


Рисунок 2.5.2 Создание триггера для добавления

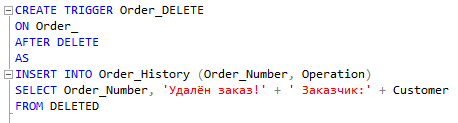


Рисунок 2.5.3 Создание триггера для удаления

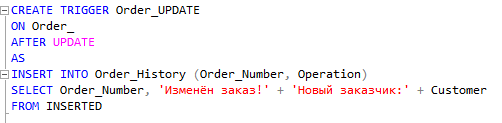


Рисунок 2.5.4 Создание триггера для изменения

На рисунке 2.5.5 виден результат после того, как были выполнены добавление, изменение и удаление:

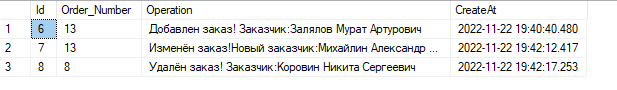


Рисунок 2.5.5 Результат работы триггеров

## Создание представления

Представления или Views представляют виртуальные таблицы. Но в отличии от обычных стандартных таблиц в базе данных представления содержат запросы, которые динамически извлекают используемые данные.

Представления дают нам ряд преимуществ. Они упрощают комплексные SQL-операции. Они защищают данные, так как представления могут дать доступ к части таблицы, а не ко всей таблице. Представления также позволяют возвращать отформатированные значения из таблиц в нужной и удобной форме.

Для создания представления используется команда CREATE VIEW.

На рисунке 2.6.1 видно создание представления «Treaty\_», в котором будет отображен список всех договоров с суммой всех их товаров.

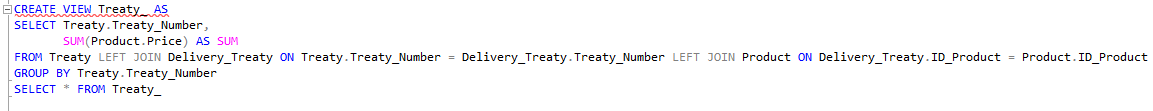


Рисунок 2.6.1 Создание представления «Treaty\_»

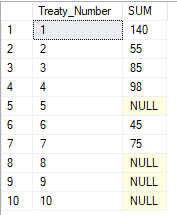


Рисунок 2.6.2 Результат создания представления «Treaty\_»

## Создание функции

В языках программирования обычно имеется два типа подпрограмм:

* хранимые процедуры;
* определяемые пользователем функции (UDF).

В отличие от хранимых процедур, функции всегда возвращают одно значение. В этом разделе мы рассмотрим создание и использование определяемых пользователем функций (User Defined Functions - UDF).

На рисунке 2.7.1 приведен пример создания функции «Select\_Orders» для отображения заказов в определённом промежутке времени. Функция будет принимать 2 значения: начальную и конечную дату.



Рисунок 2.7.1 Создание функции «Select\_Orders»

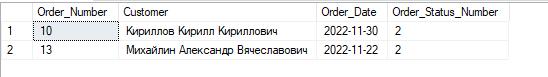


Рисунок 2.7.2 Результат выполнения функции «Select\_Orders»

На рисунке 2.7.3 приведен пример создания функции «Select\_Finished\_Orders» для отображения выполненных заказов. Функция не принимает никаких значений.

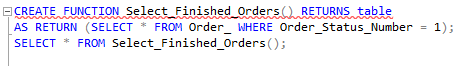


Рисунок 2.7.3 Создание функции «Select\_Finished\_Orders»

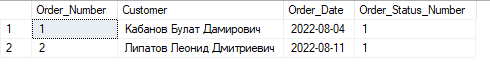


Рисунок 2.7.4 Результат выполнения функции «Select\_Finished\_Orders»

## 2.8 Создание резервной копии БД

Для создания резервной копии БД требуется нажать ПКМ по своей БД, выбрать «Задачи» и затем выбрать «Создать резервную копию».

Пример показан на рисунке 2.8.1:

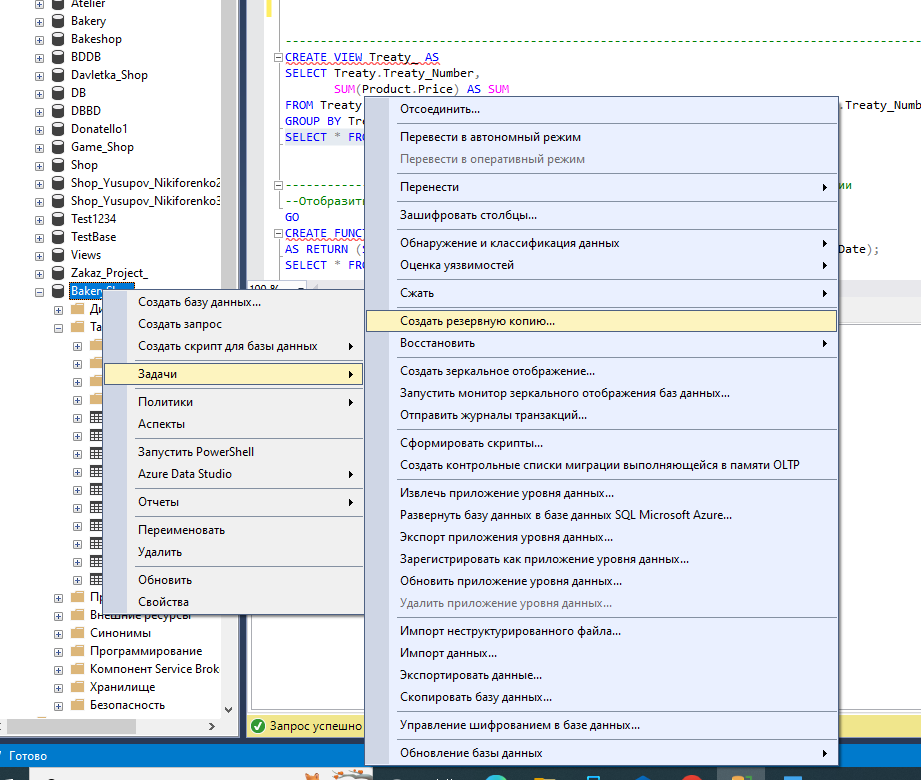


Рисунок 2.8.1 Первый этап создания резервной копии БД

Затем, по нажатию на «Создать резервную копию» откроется окно «Резервное копирование данных. В центральном окне расположена ссылка, по которой будет сохранена резервная копия. Снизу от окна надо нажать на кнопку «Ок». По нажатию на эту кнопку сохранение будет успешно завершено.

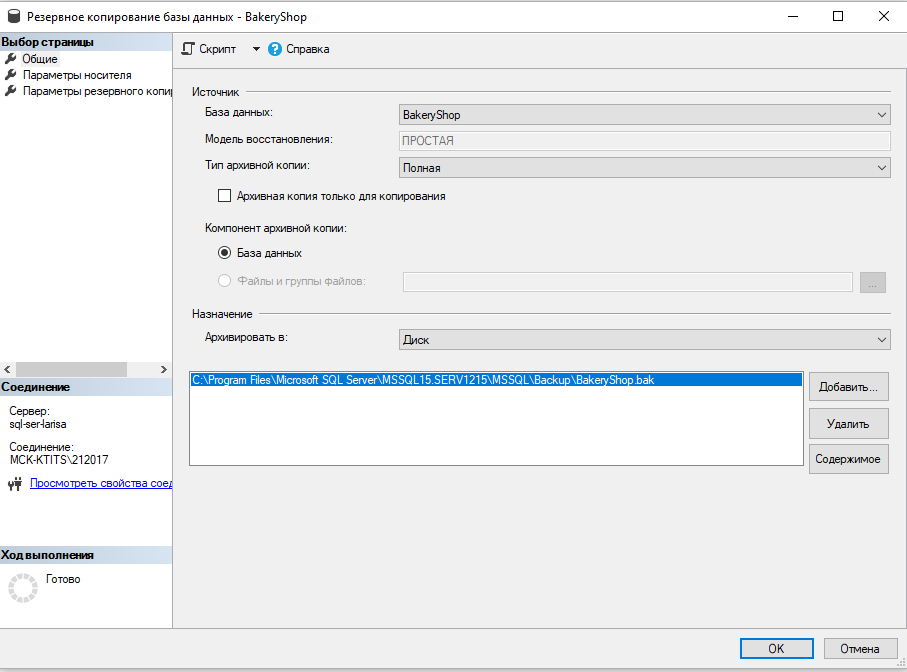


Рисунок 2.8.2 Окно «Резервное копирование базы данных»

# ЗАКЛЮЧЕНИЕ

В процессе прохождения учебной практики была разработана база данных – «Хлебопекарня». В ходе работы с ними были достигнуты цели и задачи:

* Изучение и анализ предметной области;
* Разработка самой базы данных в СУБД и заполнение её данными;
* Применение практических навыков проектирования баз данных с помощью СУБД;
* Использование выражений для редактирования структуры таблиц;
* Использование триггеров;
* Использование представлений;
* Использование функций;
* Сохранение резервной копии БД.

Индивидуальное задание было успешно выполнено.