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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(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.

Оглавление

[1 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ 3](#_Toc120050311)

[1.1 Обоснование выбора СУБД 3](#_Toc120050312)

[2 РАЗРАБОТКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ 7](#_Toc120050313)

[2.1 Создание базы данных 7](#_Toc120050314)

[2.2 Создание таблиц 8](#_Toc120050315)

[2.3 Создание ER диаграммы 8](#_Toc120050316)

[2.4 Работа с таблицей 9](#_Toc120050317)

[2.5 Создание триггеров 10](#_Toc120050318)

[2.6 Создание представления 10](#_Toc120050319)

[2.7 Создание функции 11](#_Toc120050320)

[2.8 Создание резервной копии БД 12](#_Toc120050321)

[ЗАКЛЮЧЕНИЕ 13](#_Toc120050322)

* + 1. ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ
       1. Обоснование выбора СУБД

Всякая профессиональная деятельность, так или иначе, связана с информацией, с организацией ее сбора, хранения, выборки. Можно сказать, что неотъемлемой частью повседневной жизни стали базы данных, для поддержки которых требуется некоторый организационный метод, или механизм. Такой механизм называется системой управления базами данных (СУБД). Итак, введем основные понятия.

База данных (БД) - совместно используемый набор логически связанных данных (и их описание), предназначенный для удовлетворения информационных потребностей организации.

СУБД (система управления базами данных) - программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных, а также получать к ней контролируемый доступ.

Управление основными потоками информации осуществляется с помощью, так называемых, систем управления реляционными базами данных, которые берут свое начало в традиционных системах управления базами данных. Именно объединение реляционных баз данных и клиент-серверных технологий позволяет современному предприятию успешно управлять собственными данными, оставаясь конкурентоспособным на рынке товаров и услуг.

Реляционные БД имеют мощный теоретический фундамент, основанный на математической теории отношений. Появление теории реляционных баз данных дало толчок к разработке ряда языков запросов, которые можно отнести к двум классам:

- алгебраические языки, позволяющие выражать запросы средствами специализированных операторов, применяемых к отношениям;

- языки исчисления предикатов, представляющие собой набор правил для записи выражения, определяющего новое отношение из заданной совокупности существующих отношений. Следовательно, исчисление предикатов есть метод определения того отношения, которое желательно получить как ответ на запрос из отношений, уже имеющихся в базе данных.

В реляционной модели объекты реального мира и взаимосвязи между ними представляются с помощью совокупности связанных между собой таблиц (отношений).

Даже в том случае, когда функции СУБД используются для выбора информации из одной или нескольких таблиц (т.е. выполняется запрос), результат также представляется в табличном виде. Более того, можно выполнить запрос с применением результатов другого запроса.

Каждая таблица БД представляется как совокупность строк и столбцов, где строки (записи) соответствуют экземпляру объекта, конкретному событию или явлению, а столбцы (поля) - атрибутам (признакам, характеристикам, параметрам) объекта, события, явления.

В каждой таблице БД необходимо наличие первичного ключа - так именуют поле или набор полей, однозначно идентифицирующий каждый экземпляр объекта или запись. Значение первичного ключа в таблице БД должно быть уникальным, т.е. в таблице не допускается наличие двух и более записей с одинаковыми значениями первичного ключа. Он должен быть минимально достаточным, а значит, не содержать полей, удаление которых не отразится на его уникальности.

Связи между объектами реального мира могут находить свое отражение в структуре данных, а могут и подразумеваться, т.е. присутствовать на неформальном уровне.

Между двумя или более таблицами базы данных могут существовать отношения подчиненности, которые определяют, что для каждой записи главной таблицы (называемой еще родительской) возможно наличие одной или нескольких записей в подчиненной таблице (называемой еще дочерней).

Выделяют три разновидности связи между таблицами базы данных:

- "один-ко-многим";

- "один-к-одному";

- "многие-ко-многим".

Отношение "один-ко-многим" имеет место, когда одной записи родительской таблицы может соответствовать несколько записей дочерней. Связь "один-ко-многим" иногда называют связью "многие-к-одному". И в том, и в другом случае сущность связи между таблицами остается неизменной.

Связь "один-ко-многим" является самой распространенной для реляционных баз данных. Она позволяет моделировать также иерархические структуры данных.

Отношение "один-к-одному" имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней. Это отношение встречается намного реже, чем отношение "один-ко-многим". Его используют, если не хотят, чтобы таблица БД "распухала" от второстепенной информации, однако для чтения связанной информации в нескольких таблицах приходится производить ряд операций чтения вместо одной, когда данные хранятся в одной таблице.

Отношение "многие-ко-многим" применяется в следующих случаях. Одной записи в родительской таблице соответствует более одной записи в дочерней. Или же одной записи в дочерней таблице соответствует более одной записи в родительской. Всякую связь "многие-ко-многим" в реляционной базе данных необходимо заменить на связь "один-ко-многим" (одну или более) с помощью введения дополнительных таблиц.

В качестве средства реализации БД была выбрана система MS SQL Server 20019 Express.

Microsoft SQL Server 20019 Express – это мощная и надежная система управления данными, обеспечивающая множество функций, защиту данных и высокую производительность для внедренных приложений-клиентов, «легких» веб-приложений и локальных хранилищ данных. SQL Server 2019 Express предназначен для упрощенного развертывания и быстрого создания прототипов; его можно получить бесплатно и свободно распространять вместе с приложениями.

SQL Server 2019 упрощает развёртывание, управление и оптимизацию данных предприятия и аналитических приложений. Как платформа управления данными предприятия, он предоставляет единую консоль управления, которая позволяет администраторам данных, находящимся в любом месте вашей организации, отслеживать, управлять и настраивать все базы данных и связанные службы по всему предприятию. Он предоставляет расширяемую инфраструктуру управления, которая может быть легко запрограммирована при помощи SQL Management Objects (SMO), позволяя пользователям переделывать и расширять их среду управления и независимым поставщикам программных продуктов (ISV) создавать дополнительные инструменты и функциональность для дальнейшего расширения возможностей, поставляемых по умолчанию.

2 РАЗРАБОТКА ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

Создание базы данных. Необходимо создать базу данных по предметной области “Автоцентр”. Создание таблиц. Необходимо было выявить все сущности связанные с описанием предметной области.

Создание ER диаграммы. Требовалось воспользоваться конструктором ER диаграмм в СУБД и создать её по заданной предметной области.

Работа с таблицей. Необходимо дополнить две созданные таблицы новыми колонками.

Создание триггеров. Требовалось создать триггеры сохранения всех проходящих операций для одной из таблиц. Триггер должен сохранять добавление и удаление в отдельную таблицу.

Создание представления. При помощи запроса требовалось разработать представление для одной из таблиц, где выводится вся информация о нём.

Создание функции. Разработать две функции для двух разных таблиц. Первая функция с сортировкой данных по времени, вторая с выводом последней операции пользователя.

* 1. Создание базы данных

При рассмотрении индивидуального задания можно выделить следующие сущности: Автоцентр, Автозапчасти, Автосервис, АвтосервисАвтозапчасти, Покупки, Машины, Клиент, История, Заказ.

* 1. Создание таблиц

Для создания таблиц необходимо было сформировать запрос со встроенной функцией CREATE TABLE изображенная на рисунке 2.2.1.

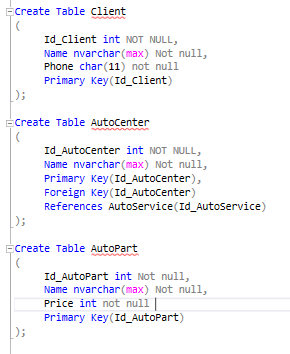


Рисунок 2.2.1 Запросы создания таблиц

* 1. Создание ER диаграммы

Благодаря инструментам SSMS 19 (SQL Server Management Studio), создается подробная диаграмма базы данных, изображенная на рисунке 2.3.1

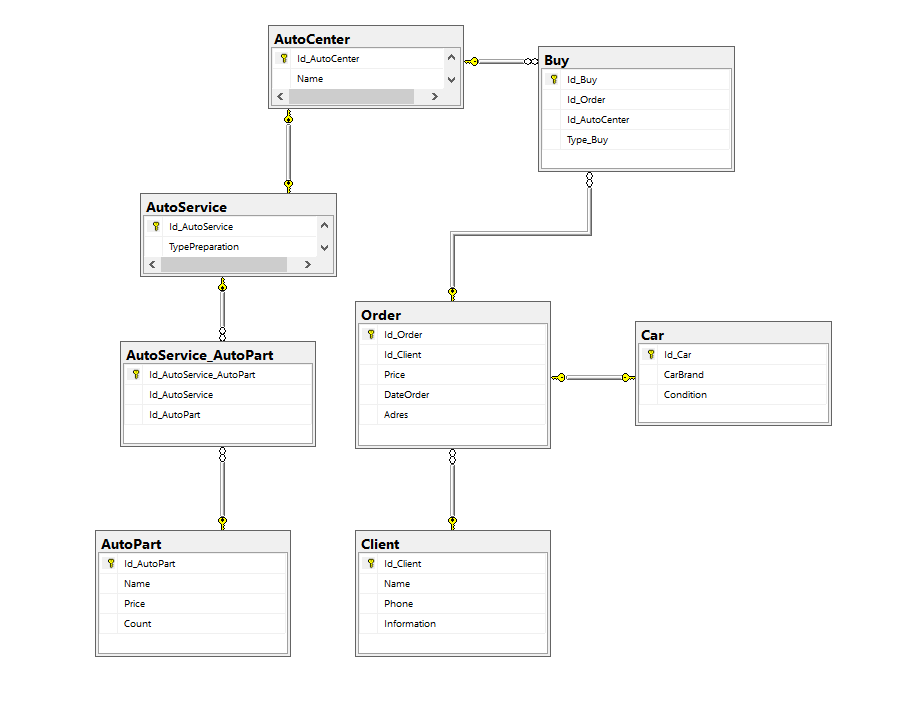


Рисунок 2.3.1 Диаграмма базы данных «Автоцентр»

* 1. Работа с таблицей

Для работы с таблицей потребовалось сформировать запросы добавления данных в таблицу и добавление дополнительных столбцов. Для добавления данных используется функция INSERT INTO изображенная на рисунке 2.4.1.

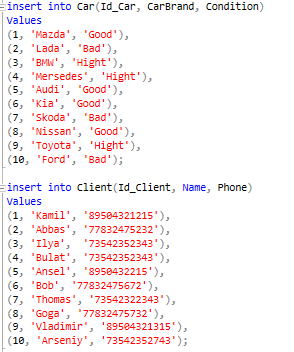


Рисунок 2.4.1 Запросы добавления данных в таблицу

Для добавления столбца используется функция ADD [Имя столбца] тип данных, но перед этим необходимо указать Alter Table [Название таблицы]. Добавление столбца изображено на рисунке 2.4.2.

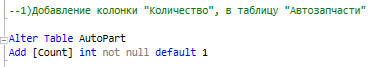


Рисунок 2.4.2 Запрос добавления столбца

2.5 Создание триггеров

Для создания триггера использовалась функция CREATE TRIGGER [Название триггера]. Чтобы запустить триггер необходимо написать запрос, изображенный на рисунке 2.5.1.

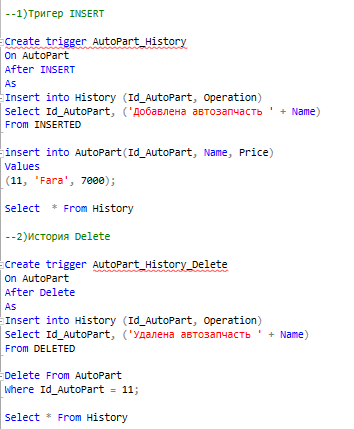


Рисунок 2.5.1 Триггер проходящих операций

* 1. Создание представления

Для создания представления использовалась функция CREATE VIEW [Название представления]. Представления сохраняются в базе данных в разделе представления. Создание представления изображено на рисунке 2.6.1.

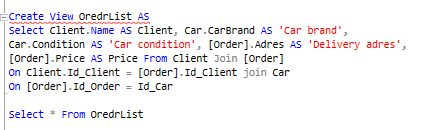


Рисунок 2.6.1 Создание представления

* 1. Создание функции

Для создания функции требуется разрешение CREATE FUNCTION на базу данных и разрешение ALTER для схемы, в которой создается функция. Если в функции указан определяемый пользователем тип, требуется разрешение EXECUTE на этот тип. CREATE FUNCTION принимает параметры, выполняет действия, такие как сложные вычисления, а затем возвращает результат этих действий в виде значения. Создание функция изображено на рисунке 2.7.1.

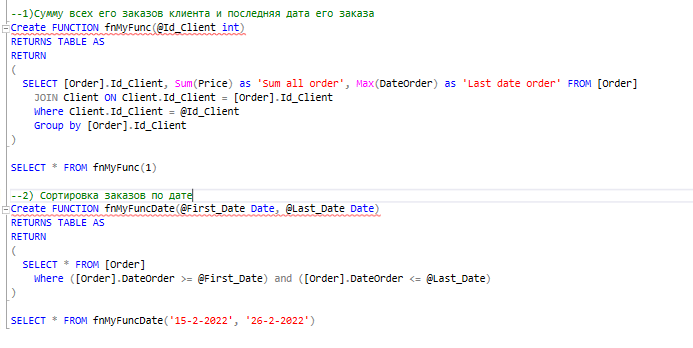


Рисунок 2.7.1 Создание функций

* 1. Создание резервной копии БД

Создание резервной копии БД происходит после подключения к базе данных. Для полного резервного копирования на диск необходимо развернуть элемент Базы данных, щелкнуть на название БД правой кнопкой мыши, навести указатель на пункт Задачи и выбрать действие Создать резервную копию. Процесс создания резервного копирования показан на рисунке 2.8.1.

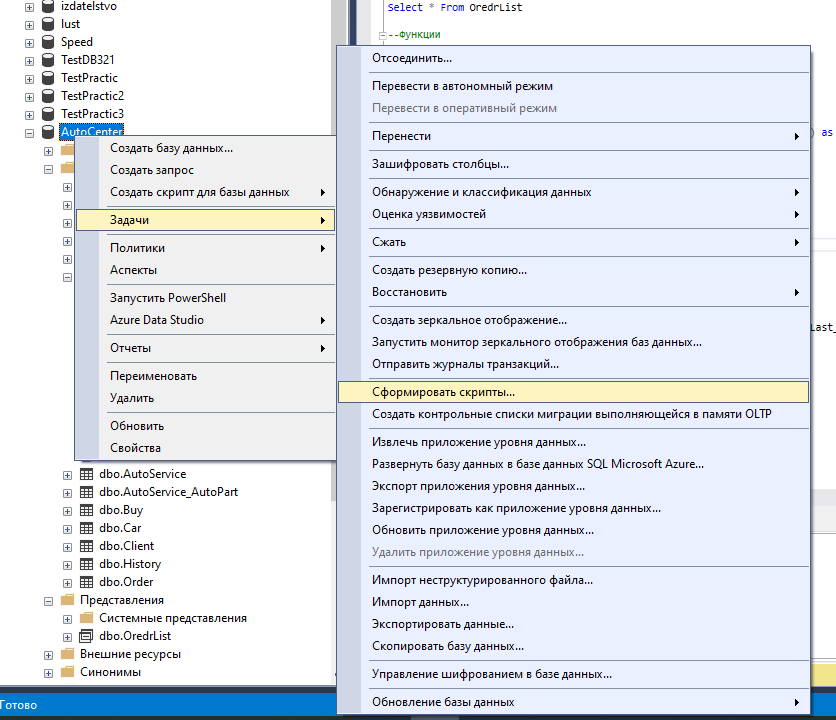


Рисунок 2.8.1 Создание резервной копии БД

ЗАКЛЮЧЕНИЕ

В процессе прохождения учебной практики была разработана база данных – «Автоцентр» с помощью языка запросов SQL и СУБД SQL Managment. 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;
* Умение применять встроенные функции СУБД такие, как JOIN, SUM, CREATE FUNCTION, DELETE TABLE;

По окончанию практики была достигнута главная цель – закрепление теоретических знаний, полученных в процессе обучения; приобретения практических навыков, компетенций и опыта деятельности по направлению подготовки; ознакомления на практике с вопросами профессиональной деятельности, направленными на формирование знаний, навыков и опыта профессиональной деятельности.

Данная практика является хорошим практическим опытом для дальнейшей самостоятельной деятельности. За время пройденной практики я познакомилась с новыми интересными фактами.