ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИМ. Н. П. ОГАРЁВА»**

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных и специальных (информационно-коммуникационных) дисциплин

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ**

на тему Разработка базы данных для клиентской части системы онлайн кассы

по ПМ.02 Разработка и администрирование баз данных

Автор курсового проекта Р. А. Куроедов

(подпись) (дата)

Обозначение курсового проекта КП–02069964–09.02.03–11–20

Направление подготовки (специальность) 09.02.03 Программирование в компьютерных системах

Руководители проекта:

Преподаватель Е. Е. Старушенкова

(подпись) (дата)

Проект защищён \_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата

Саранск

2020

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ИМ. Н. П. ОГАРЁВА»**

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра) общепрофессиональных и специальных (информационно-коммуникационных) дисциплин

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

Студент Куроедов Роман Александрович

1. Тема: Разработка базы данных для клиентской части системы онлайн кассы
2. Срок представления проекта к защите 23.03.2020 г.
3. Исходные данные для научного проектирования: техническая документация с описанием базы данных системы онлайн кассы, документация по использованию системы управления базами данных Microsoft SQL Server 2016
4. Содержание курсового проекта

ВВЕДЕНИЕ

4.1 Теоретическое обоснование проекта

4.2 Проектная часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А (обязательное) Скриншоты таблиц базы данных

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_ Е. Е. Старушенкова

подпись, дата

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ Р. А. Куроедов

подпись, дата

**РЕФЕРАТ**

Пояснительная записка содержит 31 лист, 6 рисунков, 8 таблиц,   
10 источников, 1 приложение.

БАЗА ДАННЫХ, КАССА, СКЛАД, ПРОДАЖА, ВОЗВРАТ, СМЕНА, АВТОРИЗАЦИЯ, ОНЛАЙН.

Объектом разработки является база данных для клиентской части системы онлайн кассы, включающий следующие функции:

* учет сотрудников предприятия;
* ведение сессионной статистики торгового предприятия;
* ведение статистики продажи товаров сотрудниками торгового предприятия;
* ведение статистики возвратов товаров клиентами торгового предприятия;
* обновление статистики склада торгового предприятия;
* учет товаров склада предприятия.

Цель работы ⎯ автоматизация и снижение уровня трудоёмкости следующих производственных процессов:

* учет сотрудников предприятия;
* ведение сессионной статистики торгового предприятия;
* ведение статистики продажи товаров сотрудниками торгового предприятия;
* ведение статистики возвратов товаров клиентами торгового предприятия;
* ведение учета и статистики склада торгового предприятия.

В процессе работы использовались знания и опыт, полученные при изучении учебного курса «Технология разработки и защиты баз данных».

В результате выполнения курсовой работы был получен программный модуль готовый к внедрению и расширению, который включает в себя модули продажи и возврата товаров, ведения сессионной статистики и авторизации пользователей, а также разработан удобный и понятный для пользователя интерфейс.

Эффективность проекта:

* уменьшение количества трудовых и временных ресурсов, необходимых для обслуживания клиентов торгового предприятия;
* уменьшение количества трудовых и временных ресурсов, необходимых, для ведения ресурсной и финансовой отчетности;
* компактное и мало затратное, с точки зрения пространственных и финансовых ресурсов, хранение необходимых предприятию данных;
* быстрый доступ к данным.

Для данной курсовой работы объектом исследования является онлайн касса. Предметом исследования курсовой работы является база данных.

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| [ВВЕДЕНИЕ](#Введение) | | 6 |
| [1 Теоретическое обоснование проекта](#Введение) | | 9 |
|  | [1.1 Анализ предметной области](#Анализ) | 9 |
|  | [1.2 Выбор инструментария разработки](#Инструментарий) | 10 |
| [2 Проектная часть](#Проект) | | 18 |
|  | [2.1 Разработка базы данных](#Разработка) | 18 |
|  | 2.2 Нормализация базы данных | 21 |
|  | [2.3 Организация защиты базы данных и СУБД](#Защита) | 23 |
|  | 2.4 Тестирование защищенности базы данных | 24 |

|  |  |
| --- | --- |
| [ЗАКЛЮЧЕНИЕ](#Заключение) | 27 |
| [СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#Источники) | 29 |
| [ПРИЛОЖЕНИЕ А (обязательное) Скриншоты таблиц](#ПриложениеА) базы данных | 30 |

**ВВЕДЕНИЕ**

Становление и развитие информационных технологий началось во второй половине 20 века. Тогда мало кто мог представить, что всего через пол века от этих самых информационных технологий общество будет зависеть настолько, что, лишившись их, впадет в хаос.

21 век по праву называется веком информационных технологий, так как они присутствуют на каждом шагу. Ими пользуются все от мала до велика: обычные люди — в повседневной жизни, частные предприятия — в своей деятельности и так далее. Так же информационные технологии широко применяются и на государственном уровне в сферах здравоохранения, политики, обороны, финансов и так далее. В следствие чего применение некоторых видов информационных технологий фиксируется на законодательном уровне. Ярким примером этого является применение контрольно-кассовой техники при осуществлении наличных денежных расчетов и (или) расчётов с использованием платёжных карт.

С апреля 2016 года вступил в силу ряд поправок к 54-ФЗ «О применении контрольно-кассовой техники при осуществлении наличных денежных расчетов и (или) расчётов с использованием платёжных карт». В обновлении закона понятие ЭКЛЗ исчезает. Вместо него появляется фискальный накопитель.

Фискальный накопитель — шифровальные (криптографические) средства защиты фискальных данных в опломбированном корпусе, содержащие ключ фискального признака, обеспечивающие запись перечня фискальных данных, в некорректируемом виде, их энергонезависимое долговременное хранение, формирование фискального признака, аутентификацию электронных документов, направляемых в контрольно-кассовую технику оператором фискальных данных, а также при необходимости пользователя обеспечивающие шифрование фискальных данных, в целях обеспечения конфиденциальности информации, передаваемой оператору фискальных данных (ОФД).

Кассовый программно-технический комплекс передает данные каждого оформленного чека в виде электронного документа оператору фискальных данных, который в свою очередь будет накапливать, хранить и передавать эти данные в ФНС. Оператор фискальных данных осуществляет ежедневную передачу в адрес налоговых органов фискальных данных, а также обеспечивает возможность проверки применения ККТ через интернет и достоверность оформленного кассового чека в виде электронного документа. Пользователь, приобретая новое оборудование, может самостоятельно поставить его на учет в налоговые органы, а также должен заключить договор с одним из ОФД на передачу данных для взаимодействия с налоговой.

Целью работы является автоматизация и снижение уровня трудоёмкости следующих производственных процессов:

* учет сотрудников предприятия;
* ведение сессионной статистики торгового предприятия;
* ведение статистики продажи товаров сотрудниками торгового предприятия;
* ведение статистики возвратов товаров клиентами торгового предприятия;
* ведение учета и статистики склада торгового предприятия.

Для выполнения курсовой работы поставлены следующие задачи:

1. изучить и проанализировать предметную область на основе материалов, предоставленных в 54-ФЗ «О применении контрольно-кассовой техники при осуществлении наличных денежных расчетов и (или) расчётов с использованием платёжных карт»;
2. разработать базу данных, позволяющую:

* осуществлять учет сотрудников предприятия;
* осуществлять ведение сессионной статистики торгового предприятия;
* осуществлять ведение статистики продажи товаров сотрудниками торгового предприятия;
* осуществлять ведение статистики возвратов товаров клиентами торгового предприятия;
* осуществлять ведение учета товаров и статистики склада торгового предприятия.

**1** **Теоретическое обоснование проекта**

* 1. **Анализ предметной области**

В настоящее время торговля осуществляется повсеместно. Ей подвержены любые предприятия: как малые, в которых может состоять лишь   
1 человек, так и гиганты, насчитывающие в своем штате миллионы сотрудников. Их объединяет то, что любая торговая деятельность облагается налогами со стороны государств. Налоговые организации для точного учета требуют от предприятий отчетность по своей финансовой деятельности. Бумажная отчетность может быть актуальна для малых предприятий, однако крупным компаниям неудобно отправлять тонны бумаги в налоговые инстанции, ровно, как и этим самым инстанциям неудобно обрабатывать информацию, хранящуюся во множественных кипах бумаг; куда удобнее предоставлять отчетность в режиме онлайн. Помимо этого, торговым предприятиям удобно автоматизировать их торговую деятельность не только из-за налоговой составляющей, но и со стороны уменьшения трудоемкости самих процессов торговли и управления запасами товаров и услуг. Эти задачи и призваны решить онлайн кассы.

Предметная область — онлайн касса.

Главными задачами онлайн кассы являются:

* ведение сессионной статистики;
* ведение статистики продаж;
* ведение статистики возвратов;
* ведение учета товаров и статистики складов;
* финансовая и налоговая отчетность.

Базовые сущности этой предметной области:

* Пользователи. Атрибуты — id, пароль, фамилия, имя и отчество;
* Смены. Атрибуты — id смены, id пользователя, даты и время начала и завершения смены, сумма продаж, сумма возвратов, сумма изъятий и внесений и баланс кассы;
* Товары. Атрибуты — id смены, id товара, наименование товара, количество и цена;
* Продажи. Атрибуты — id смены, id товара и количество проданного товара;
* Товары со свободной ценой. Атрибуты — id смены, сумма продаж/возвратов;
* Возвраты. Атрибуты возврата — id смены, id товара и количество возвращенного товара.
  1. **Выбор инструментария разработки**

Когда речь заходит о разработке баз данных, встаёт вопрос о выборе платформы, на базе которой будет осуществляться работа с базой данных. Платформа — это система управления базами данных (далее СУБД). Далее представлен сравнительный анализ современных СУБД.

**Oracle.** Компания Oracle предоставляет свою одноимённую СУБД, которая уже длительное время является негласным лидером по популярности среди других аналогов на рынке СУБД, имеющую блестящую репутацию. Актуальная версия Oracle идеально подходит для облачных сред хранения данных и может быть размещена на одном как на одном сервере, так и распределена по нескольким серверам, что позволяет работать с базами данных, содержащими более миллиарда записей. В инструментарий функций актуальной версии данной СУБД добавлена возможность использования и физических, и логических структур. Это дает возможность разграничить доступ между физическим управлением данными и логическими структурами. Также, благодаря тому, что каждая транзакция изолирована от других, безопасность в актуальной версии Oracle доведена до высочайшего уровня[3].

Достоинства Oracle**:**

* наличие интеграции передовых инноваций и впечатляющего функционал;
* высокая надежность.

Недостатки Oracle:

* стоимость Oracle может оказаться непомерно высокой, особенно для небольших организаций;
* высокая ресурсозатратность.

Данная СУБД идеально подходит для крупных организаций, которые работают с огромными базами данных и разнообразными функциями.

**MySQL.** MySQL по праву можно назвать одной из самых популярных СУБД для веб-приложений. Для веб-серверов, работающих на базе операционной системы Linux, данная СУБД является негласным стандартом.   
СУБД MySQL представляет собой бесплатный пакет программ. Команда разработчиков данной СУБД постоянно выпускают новые версии продукта, расширяя функционал и улучшая безопасность. Помимо этого есть и специальные платные версии для коммерческого использования. В бесплатной версии основными направлениями разработки являются скорость и надежность СУБД, а не полнота функционала, которая является и достоинством, и недостатком — в зависимости от области внедрения. Эта СУБД предоставляет возможность выбора различных движков системы хранения данных, что позволяет менять функционал и обрабатывать данные, хранящихся в различных типах таблиц. Благодаря поддержке большого количества типов таблиц достигается гибкость в работе с СУБД MySQL. В наличии также присутствуют простой и понятный интерфейс и простые в использовании пакетные команды, что обеспечивает удобство обработки огромных объемов данных. Система невероятно надежна и нересурсозатратна [4].

Достоинства MySQL:

* бесплатное распространение;
* наличие исчерпывающей документации;
* наличие множества функций, даже в бесплатной версии;
* пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно;
* поддержка набора пользовательских интерфейсов;
* возможность работы с другими базами данных, включая DB2 и Oracle.

Недостатки MySQL:

* высокая трудоёмкость при выполнении некоторых несложных задач, в то время как у других СУБД они автоматизированы. Например, создание инкрементных резервных копий;
* отсутствие встроенной поддержки XML или OLAP;
* для бесплатной версии доступна только платная поддержка.

Данная СУБД идеально подходит для: организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

**Microsoft SQL сервер.** Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Движок данной СУБД может работать как в облачных средах, так и в серверах локальных сетей. В дополнение к этому присутствует возможность одновременного комбинирования типов применяемых серверов. Начиная с версии 2016 года, данная СУБД была адаптирована и для операционной системы Linux. В актуальной версии Microsoft SQL Server присутствуют такие функции, как «Temporal Data Support» (временная поддержка данных), дающая возможность отслеживать изменения данных во времени, и «Dynamic Data Masking» (динамическую маскировку данных), предоставляющая механизм сокрытия конфиденциальных данных от неавторизованных пользователей [5].

ДостоинстваMicrosoft SQL сервер:

* простота использования;
* высокая скорость и стабильность работы;
* возможность регулирования и отслеживания уровня производительности, что помогает снизить использование ресурсов;
* наличие возможности визуализации на мобильных устройствах;
* высокая интеграция с другими продуктами Microsoft.

Недостатки Microsoft SQL сервер:

* цена для юридических лиц оказывается неприемлемой для большей части организаций;
* высокая ресурсозатратность;
* сообщается о проблемах с использованием службы интеграции для импорта файлов.

Данная СУБД идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

**PostgreSQL [7].** Еще одной часто используемой для ведения баз данных веб-сайтов бесплатной СУБД является. В настоящее время она хорошо развита, и предоставляет возможность управления как структурированными, так и неструктурированными данными. Данную СУБД можно разместить на большинстве основных платформ, включая Linux. Собственный инструментарий PostgreSQL предоставляет широкий функционал для импорта информации из других типов баз данных. Данная СУБД может быть размещена как в физической, так и в виртуальной, и в облачной средах [7].

Достоинства PostgreSQL:

* является масштабируемой и способна обрабатывать терабайты данных;
* поддерживает формат JSON;
* наличие множества предопределенных функций;
* доступен ряд интерфейсов.

Недостатки PostgreSQL:

* документация недостаточно точна, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете;
* конфигурация может смутить неподготовленного пользователя;
* скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

Идеально подходит для организаций с ограниченным бюджетом, но квалифицированными специалистами, когда требуется возможность выбрать свой интерфейс и использовать JSON.

**MongoDB.** Данная СУБД также является бесплатно, но в то же время имеет и коммерческую версию. Она предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядру Данной СУБД присуща гибкость. Работа с ядром при подключении базы данных к приложению осуществляется посредством драйверов MongoDB. Благодаря широкому выбору доступных драйверов, поиск драйвера, совместимого с требуемым языком программирования, является легкой задачей. При использовании MongoDB для обработки реляционных моделей данных могут возникнуть проблемы производительности, так как изначально данная СУБД не разрабатывалась для работы с ними. Напротив, так как СУБД предназначена для обработки данных, отличных от реляционных, она хорошо справляется там, где бессильны другие СУБД. Однако в актуальной версии присутствует новая подключаемая систему движков хранения данных. Благодаря способности частичного индексирования в актуальной версии повысился уровень производительности, за счет уменьшения размера индексов [6].

Достоинства MongoDB:

* скорость и простота использования;
* поддержка JSON и другие традиционные документы NoSQL;
* легкость и быстрота сохранения/чтения данных любой структуры.

Недостатки MongoDB:

* SQL не используется в качестве языка запросов;
* инструменты для перевода SQL-запросов в MongoDB доступны, но их следует рассматривать именно как дополнение;
* длительное время установки.

Данная СУБД подходит для организаций, работающих с разнородными данными, которые тяжело поддаются классификации. Для внедрения потребуются высококлассные специалисты.

**MariaDB.** Данная СУБД бесплатна, но есть и специальные платные версии. MariaDB предлагает широкий выбор плагинов расширений. В настоящее время MariaDB можно назвать наиболее быстро   
развивающейся СУБД. Данная СУБД является прямым аналогом СУБД MySQL. Ядро MariaDB предоставляет возможность выбора из нескольких систем хранения данных, тем самым обеспечивая более высокую степень оптимизации ресурсов. Благодаря этому повышается скорость выполнения запросов и обработки данных. Благодаря полному соответствию набора команд и API между MariaDB и MySQL, данные две СУБД являются полностью совместимыми [6].

Достоинства MariaDB:

* высокая скорость работы;
* наличие индикаторов обработки запросов;
* высокая гибкость настройки, благодаря расширяемой архитектуре и наличию большого количества плагинов;
* шифрование доступно в сети, сервере и уровне приложения.

Недостатки MariaDB:

* на данный момент стабильность ниже, чем у MySQL, поэтому даже на новых проектах можно рекомендовать [устанавливать My](http://drach.pro/blog/linux/item/124-mysql-for-fedora-25)SQL;
* как и во многих других бесплатных базах данных, необходимо платить за поддержку.

Данная СУБД идеальна как альтернатива MySQL, если MySQL не устраивает по каким-то причинам.

Работа с базой данных производится посредством программного модуля, разработанного в рамках курсового проектирования по профессиональному модулю «ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем».

Программный модуль разработан на платформе .NET Framework. Данная платформа предоставляет технологию ADO.NET для работы с базами данных. Эта технология предоставляет набор классов, через которые осуществляется механизм взаимодействия с базами данных [10].

Функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов [9]:

* Connection — установление подключения к источнику данных;
* Command — выполнение операций с данными в базе данных;
* DataReader — считывание полученных в результате запроса данных;
* DataSet — предназначен для хранения данных из базы данных и позволяет работать с ними независимо от самой базы данных;
* DataAdapter — посредник между DataSet и источником данных.

В основном работа с базой данных производится посредством этих объектов.

В дополнение к технологии ADO.NET платформа .NET Framework позволяет использовать технологию Entity Framework.

Entity Framework представляет специальную объектно-ориентированную технологию на базе.NET Framework для работы с данными. В отличие от технологии ADO.NET, технология Entity Framework представляет собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне осуществляются операции с таблицами, индексами, первичными и внешними ключами, то на концептуальном уровне, предоставляемом Entity Framework, осуществляется работа с объектами [2].

Понятие сущности является центральной концепцией Entity Framework. Сущность — это набор данных, ассоциированных с определенным объектом (класс). Следовательно, работа производится с объектами и их наборами,   
а не с таблицами [2].

Entity Framework использует запросы, составленные с помощью языка интегрированных запросов LINQ для работы с данными в базе данных. С помощью LINQ возможно не только извлекать определенные строки, хранящие объекты из базы данных, но и получать объекты, связанные различными ассоциативными связями [1].

Таким образом, так как большинство инструментов разработки программного модуля, посредством которого производится работа с разработанной базой данный, базируются на платформе .NET Framework, то для наибольшей интеграции компонентов системы в качестве СУБД целесообразно использовать Microsoft SQL Server. В дополнение к этому, для уменьшения трудоёмкости обеспечения взаимодействия приложения и базы данных лучшим выбором является использование технологии Entity Framework взамен языка SQL.

В ходе проектирования были решены следующие задачи:

* изучение и анализ предметной области;
* выбор инструментов, с помощью которых будет произведена разработка базы данных в проектной части данной курсовой работы.

Таким образом, после завершения всех перечисленных выше задач можно перейти к проектной части данной курсовой работы.

**2 Проектная часть**

**2.1 Разработка базы данных**

На основе анализа предметной области были разработаны модель данных, представленная на рисунке 1, и ER-диаграмма базы данных, представленная на рисунке 2.

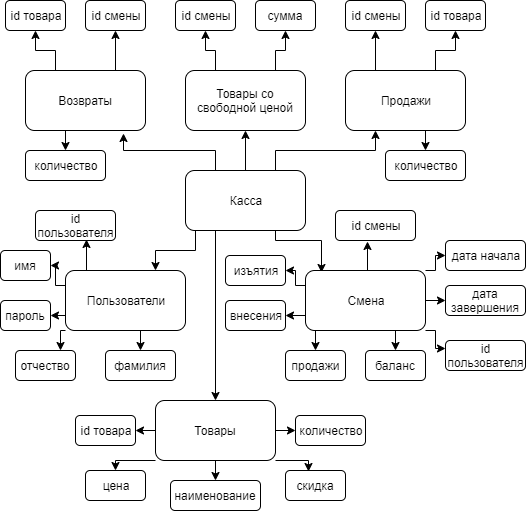


Рисунок 1 — Модель данных

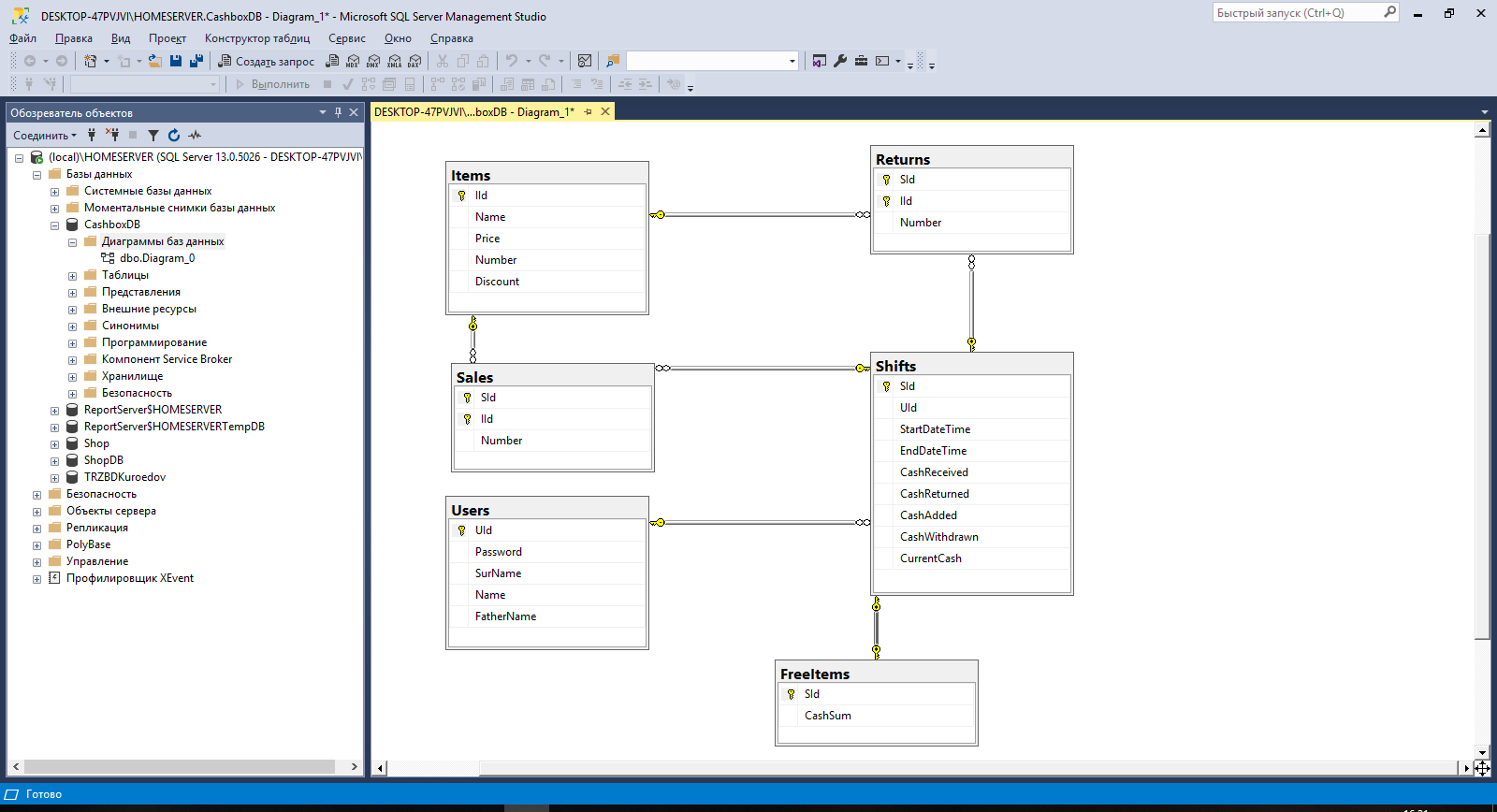


Рисунок 2 — ER-диаграмма базы данных

База данных включает в себя следующие объекты:

* таблица Items — таблица, хранящая данные о товарах, имеющихся в наличии;
* таблица Returns — таблица, хранящая данные о возвратах товара;
* таблица Sales — таблица, хранящая данные о продажах товара;
* таблица Shifts — таблица, хранящая данные о сменах;
* таблица FreeItems — таблица, хранящая данные о продажах товара со свободной ценой;
* таблица Users — таблица, хранящая данные об операторах системы.

Таблица 1 — Структура таблицы Items

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Null | Комментарий |
| \*IId | varchar(50) | Нет | Id товара |
| Name | varchar(100) | Нет | Наименование |
| Price | decimal(18, 2) | Нет | Цена, руб. |
| Number | bigint | Нет | Количество |
| Discount | int | Нет | Скидка, % |

Таблица 2 — Структура таблицы FreeItems

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Null | Комментарий |
| \*SId | varchar(50) | Нет | Id смены |
| CashSum | decimal(18, 2) | Нет | Сумма продаж/возвратов |

Таблица 3 — Структура таблицы Returns

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Null | Комментарий |
| \*SId | bigint | Нет | Id смены |
| \*IId | varchar(50) | Нет | Id товара |
| Number | bigint | Нет | Количество |

Таблица 4 — Структура таблицы Sales

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Null | Комментарий |
| \*SId | bigint | Нет | Id смены |
| \*IId | varchar(50) | Нет | Id товара |
| Number | bigint | Нет | Количество |

Таблица 5 — Структура таблицы Shifts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Тип | Null | Дополнительно | Комментарий |
| \*Sid | bigint | Нет | Автоинкремент | Id смены |
| \*UId | varchar(50) | Нет |  | Id смены |
| StartDateTime | datetime | Нет |  | Дата начала |
| EndDateTime | datetime | Да |  | Дата окончания |
| CashReceived | decimal(18, 2) | Нет |  | Сумма продаж |
| CashReturned | decimal(18, 2) | Нет |  | Сумма возвратов |
| CashAdded | decimal(18, 2) | Нет |  | Сумма внесений |
| CashWithdrawn | decimal(18, 2) | Нет |  | Сумма изъятий |
| CurrentCash | decimal(18, 2) | Нет |  | Баланс кассы |

Таблица 6 — Структура таблицы Users

|  |  |  |  |
| --- | --- | --- | --- |
| Атрибут | Тип | Null | Комментарий |
| \*UId | varchar(50) | Нет | Id оператора |
| Password | varchar(50) | Нет | Пароль |
| Surname | varchar(50) | Нет | Имя |
| Name | varchar(50) | Нет | Фамилия |
| FatherName | varchar(50) | Нет | Отчество |

Скриншоты таблиц представлены в приложении А.

Так как связь базы данных и приложения осуществляется при помощи Entity Framework, то благодаря подходу Code First в приложении при помощи средств языка программирования можно создать контекст данных, при создании экземпляра которого, если база данных не существует на сервере, будет сгенерировано SQL-выражение для создания базы данных со структурой, соответствующей моделям, описанным в приложении. Выполнение запросов производится при помощи LINQ-запросов, которые, благодаря Entity Framework, транслируются в соответствующие SQL-выражения.

**2.2 Нормализация базы данных**

Обязательной составляющей разработки любой базы данных является нормализация таблиц базы данных. Нормализацией называется это процесс изменения информации и приведения общей структуры данных к нормальным формам. Благодаря нормализации, удаётся избежать структурных и логических проблем данных. Если в базе данных существуют одинаковые записи, то велика вероятность нарушения целостности данных при обновлении таблицы, но так структура процесса нормализации предполагает наличие связей между данными, нет необходимости в идентичных записях [8].

В общем случае для нормализации таблиц базы данных достаточно привести их к первой, второй и третьей нормальным формам. Для приведения к ним необходимо определить признаки каждой из трех нормальных форм:

1. признаки первой нормальной формы:

* отсутствие дублированных строк;
* хранение только атомарных значений в полях строки;
* хранение в поле данных лишь одного типа;
* отсутствие массивов и списков в любом виде;

1. признаки второй нормальной формы:

* наличие первичного ключа;
* наличие зависимости всех неключевых атрибутов от   
  первичного ключа;

1. признаки третьей нормальной формы:

* отсутствие зависимостей между неключевыми атрибутами.

Таблица 7 — Соответствие таблиц нормальным формам.

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Первичный ключ | Атрибуты | Нормальные формы |
| Items | IId | IId | 1НФ, 2НФ, 3НФ |
| Name |
| Price |
| Number |
| Discount |
| Sales | Sid+IId | Sid | 1НФ, 2НФ, 3НФ |
| IId |
| Number |
| Returns | Sid+IId | Sid | 1НФ, 2НФ, 3НФ |
| IId |
| Number |
| Users | UId | UId | 1НФ, 2НФ, 3НФ |
| Password |
| Surname |
| Name |
| FarherName |
|  |  |  |  |
| Окончание таблицы 7. | | | |
| FreeItems | SId | SId | 1НФ, 2НФ, 3НФ |
| CashSum |
| Shifts | Sid+UId | SId | 1НФ, 2НФ, 3НФ |
| UId |
| StartDateTime |
| EndDateTime |
| CashReceived |
| CashReturned |
| CashAdded |
| CashWithdrawn |
| CurrentCash |

**2.3 Организация защиты базы данных и СУБД**

Защита баз данных — это процесс, ориентированный на предотвращение угроз преднамеренного или непреднамеренного характера, который осуществляется с помощью компьютерных и иных средств контроля.

Для защиты СУБД и базы данных Microsoft SQL Server предусмотрены пять способов проверки подлинности пользователя:

* проверка подлинности Windows;
* проверка подлинности SQL Server;
* Active Directory — универсальная с поддержкой MFA;
* Azure Active Directory — пароль;
* Azure Active Directory — встроенная.

В данном проекте используется проверка подлинности SQL Server. Данный тип проверки подлинности использует заранее созданную внутреннюю учетную запись пользователя СУБД. Помимо этого, дополнительной защитой данных является механизм авторизации, действующий на уровне клиентского приложения.

**2.4 Тестирование защищенности базы данных**

После организации защиты базы данных необходимо протестировать реализованные механизмы защиты. Ниже приведена таблица тестирования защищенности базы данных.

Таблица 8 — Таблица тестирования.

|  |  |  |
| --- | --- | --- |
| Входные данные | Ожидаемый результат | Действительный результат |
| Уровень СУБД | | |
| Верные логин и пароль | Успешное подключение | Успешное подключение |
| Неверные логин и/или пароль | Сообщение об ошибке | Сообщение об ошибке |
| Уровень приложения | | |
| Верные логин и пароль | Старт смены | Старт смены |
| Неверные логин и/или пароль | Сообщение о неверных логине или пароле | Сообщение о неверных логине или пароле |

Результат авторизации в СУБД с неверными логином и паролем представлен на рисунке 3.

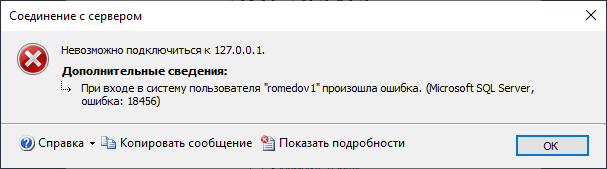
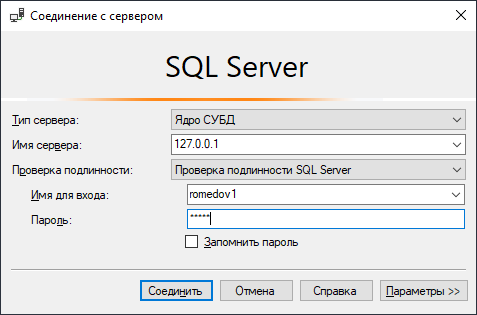
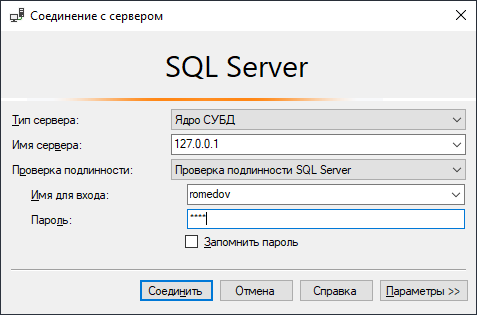


Рисунок 3 — Неверные логин и пароль СУБД

Результат авторизации в СУБД с верными логином и паролем представлен на рисунке 4.



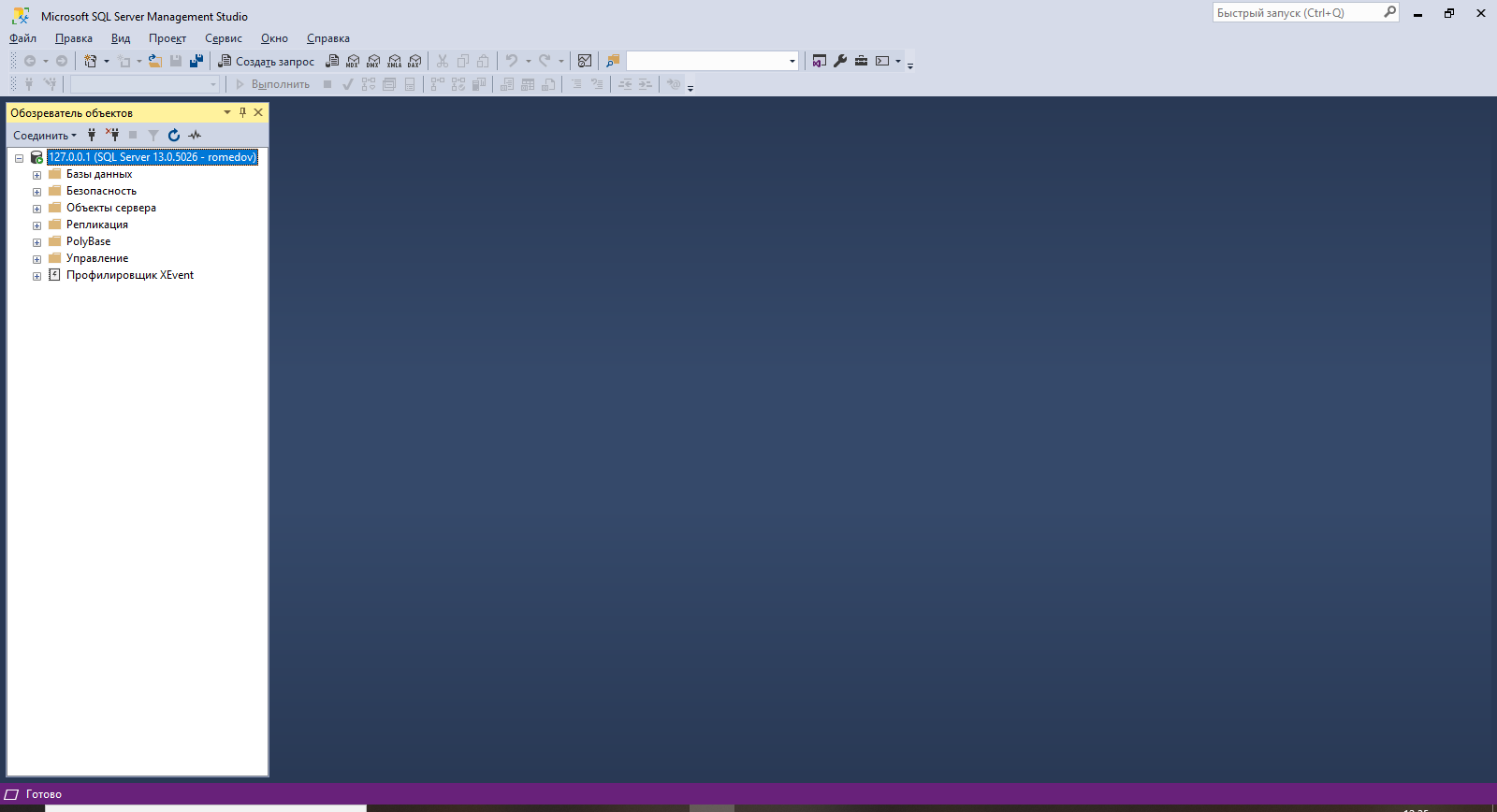
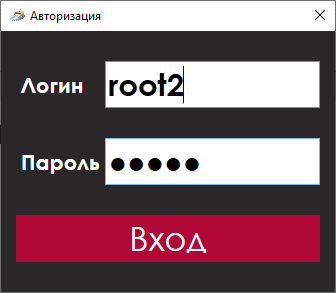


Рисунок 4 — Верные логин и пароль СУБД

Результат авторизации в приложении с неверными логином и паролем представлен на рисунке 5.



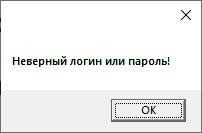


Рисунок 5 — Неверные логин и пароль приложения

Результат авторизации в приложении с верными логином и паролем представлен на рисунке 6.

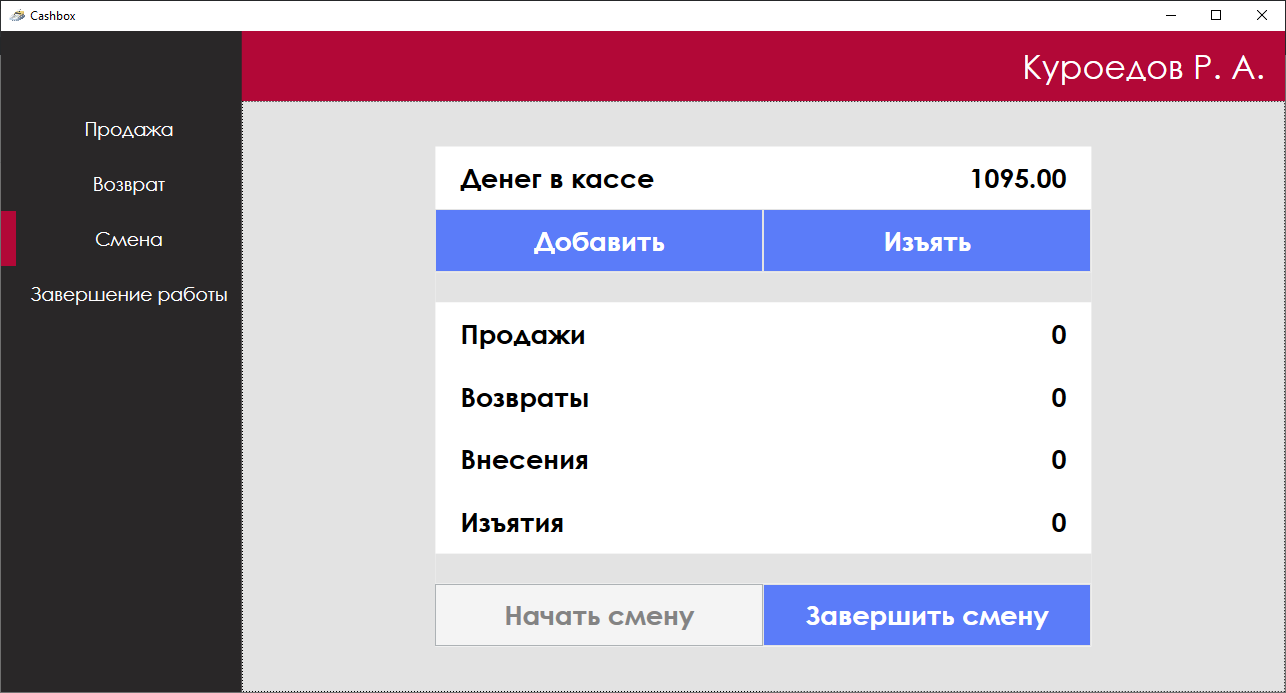
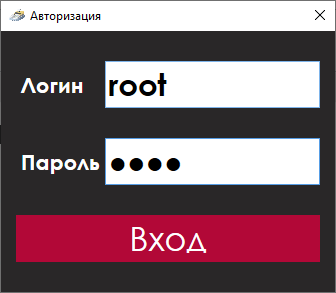


Рисунок 6 — Верные логин и пароль приложения

Таким образом в ходе разработки были решены следующие задачи:

1. создание модели данных на основе анализа предметной области;
2. создание ER-диаграммы базы данных;
3. разработка полнофункциональной базы данных, позволяющей:

* осуществлять учет сотрудников предприятия;
* осуществлять ведение сессионной статистики   
  торгового предприятия;
* осуществлять ведение статистики продажи товаров сотрудниками торгового предприятия;
* осуществлять ведение статистики возвратов товаров клиентами торгового предприятия;
* осуществлять ведение учета товаров и статистики склада торгового предприятия;

1. организация защиты базы данных и её тестирование.

Следовательно, так как все перечисленные выше задачи выполнены, можно переходить к заключительной части данной курсовой работы.

**ЗАКЛЮЧЕНИЕ**

С внедрением информационных технологий производственные процессы сильно упростились за счёт их автоматизации. Автоматизация наиболее востребована в финансовой сфере, ведь именно на обороте ресурсов и финансов всегда держалась и будет держаться экономика, являющаяся неотъемлемой сферой жизни человеческого общества. В условиях динамически расширяющегося производства обработка и хранение огромного количества данных требует слишком много времени и ресурсов.

На сегодняшний день эти действия автоматизированы, а для хранения данных используются специализированные информационные средства, что позволяет снизить трудоемкость производственных процессов и значительно снизить количество ресурсов, необходимых для хранения и обработки данных.

Целью данного курсового проекта являлась разработка физической модели базы данных для клиентской части системы онлайн кассы. Она была достигнута путем решения определенного круга задач.

Был проведен анализ предметной области, выявлены объекты и атрибуты. Для выполнения курсового проекта необходимо было провести анализ и выбор системы управления базами данных и язык запросов (были выбраны Microsoft SQL Server и LINQ). В ходе выполнения курсового проекта была создана модель данных, на основе которой была спроектирована ER-диаграмма базы данных. Ненормализованные данные в базе данных были приведены к третьей нормальной форме. Для защиты данных использовались механизмы:

* авторизация пользователя на уровне клиентского приложения;
* аутентификация Windows на уровне СУБД.

Основными задачами для разработки базы данных являлись:

* осуществление учета сотрудников предприятия;
* осуществление ведения сессионной статистики торгового предприятия;
* осуществление ведения статистики продажи товаров сотрудниками торгового предприятия;
* осуществление ведения статистики возвратов товаров клиентами торгового предприятия;
* осуществление ведения учета товаров и статистики склада торгового предприятия.

Итогом курсовой работы стала полнофункциональная база данных, полностью готовая к применению. Исходя из этого, можно сказать, что цель курсового проекта достигнута путем реализации поставленных задач. Разработанная база данных полностью удовлетворяет всем требованиям, поставленным на этапе постановки задач курсового проекта.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Stack Overflow [Электронный ресурс]. —  
    Режим доступа: https://ru.stackoverflow.com. — Дата доступа: 27.10.2019.
2. METANIT.COM [Электронный ресурс]. —   
   Режим доступа: https://metanit.com/sharp/entityframework/. — Дата доступа: 22.10.2019.
3. Дунаев, В. В. Базы данных. Язык SQL для студента / В. В. Дунаев. — Санкт-Петербург : БХВ-Петербург, 2017. — 376 с.
4. Маркин, А. В. Программирование на sql в 2 ч. Часть 1. / А. В. Маркин. — Москва : Издательство Юрайт, 2018. — 362 с.
5. Маркин, А. В. Программирование на sql в 2 ч. Часть 2. / А. В. Маркин. — Москва : Издательство Юрайт, 2018. — 362 с.
6. Тарасов, С. В. СУБД для программиста. Базы данных изнутри /   
   С. В. Тарасов. — Москва : Солон-пресс, 2017. — 1-19 с.
7. Васильев, А. Ю. Работа с Postgresql. Масштабирование и настройка /   
   А. Ю. Васильев. — Creative Commons Attribution-Noncommercial, 2017. — 70-120 с.
8. Коннолли, Т., Бегг, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. —   
   Москва : Вильямс, 2017. — 700-760 с.
9. Троелсен, Э. Язык программирования С# 7 и платформы .NET и .NET Core/ Э. Троелсен, Ф. Джепикс. — Киев : Издательство Диалектика, 2019. — 1328 с.
10. Тюкачев, Н. А. С#. Основы программирования / Н. А. Тюкачев,   
    В. Г. Хлебостроев, Джей Глинн. — Санкт-Петербург : Издательство Лань, 2018. — 272 с.

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Скриншоты таблиц базы данных**

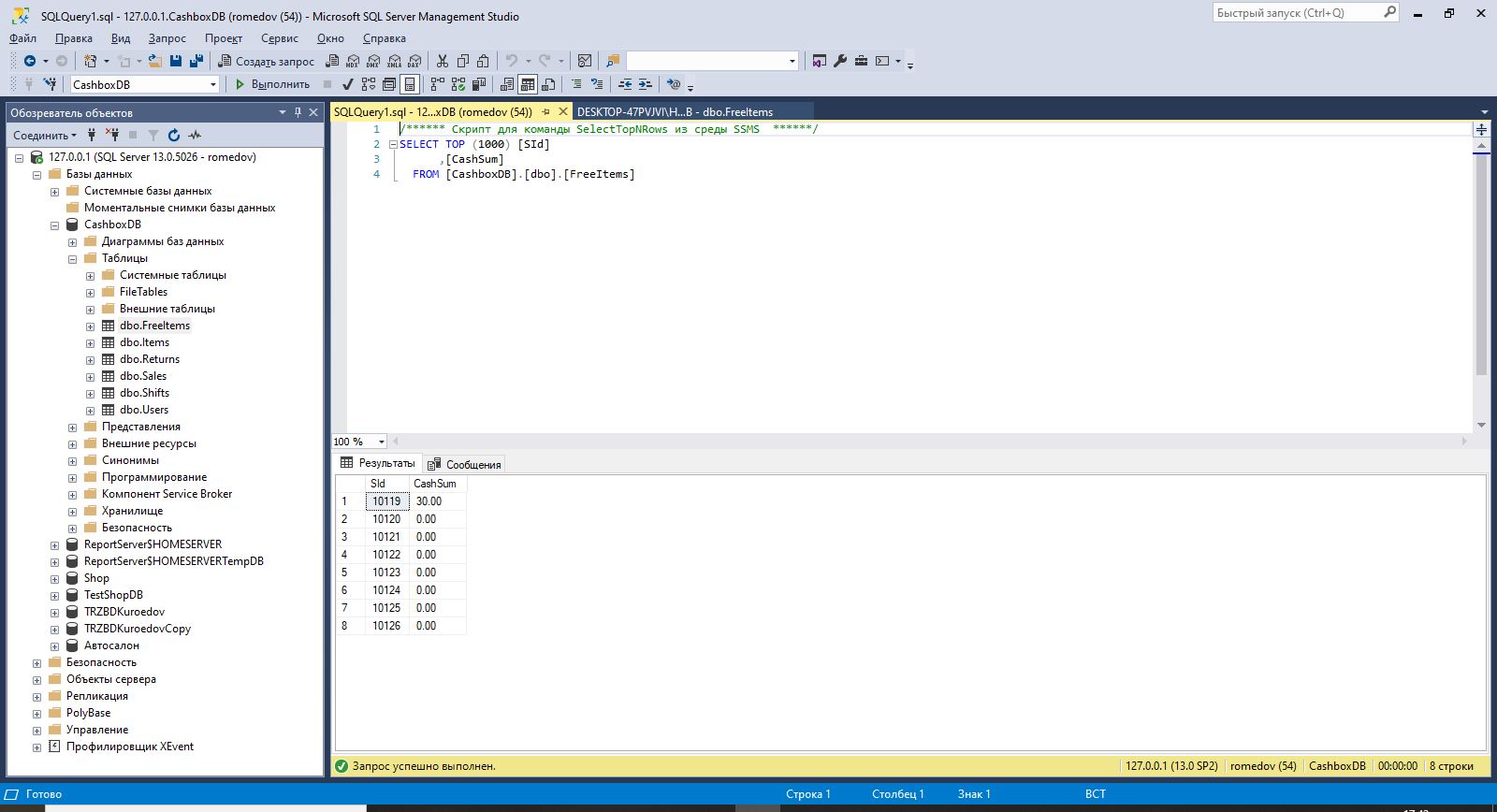


Рисунок А.1 ⎯ Таблица «FreeItems»

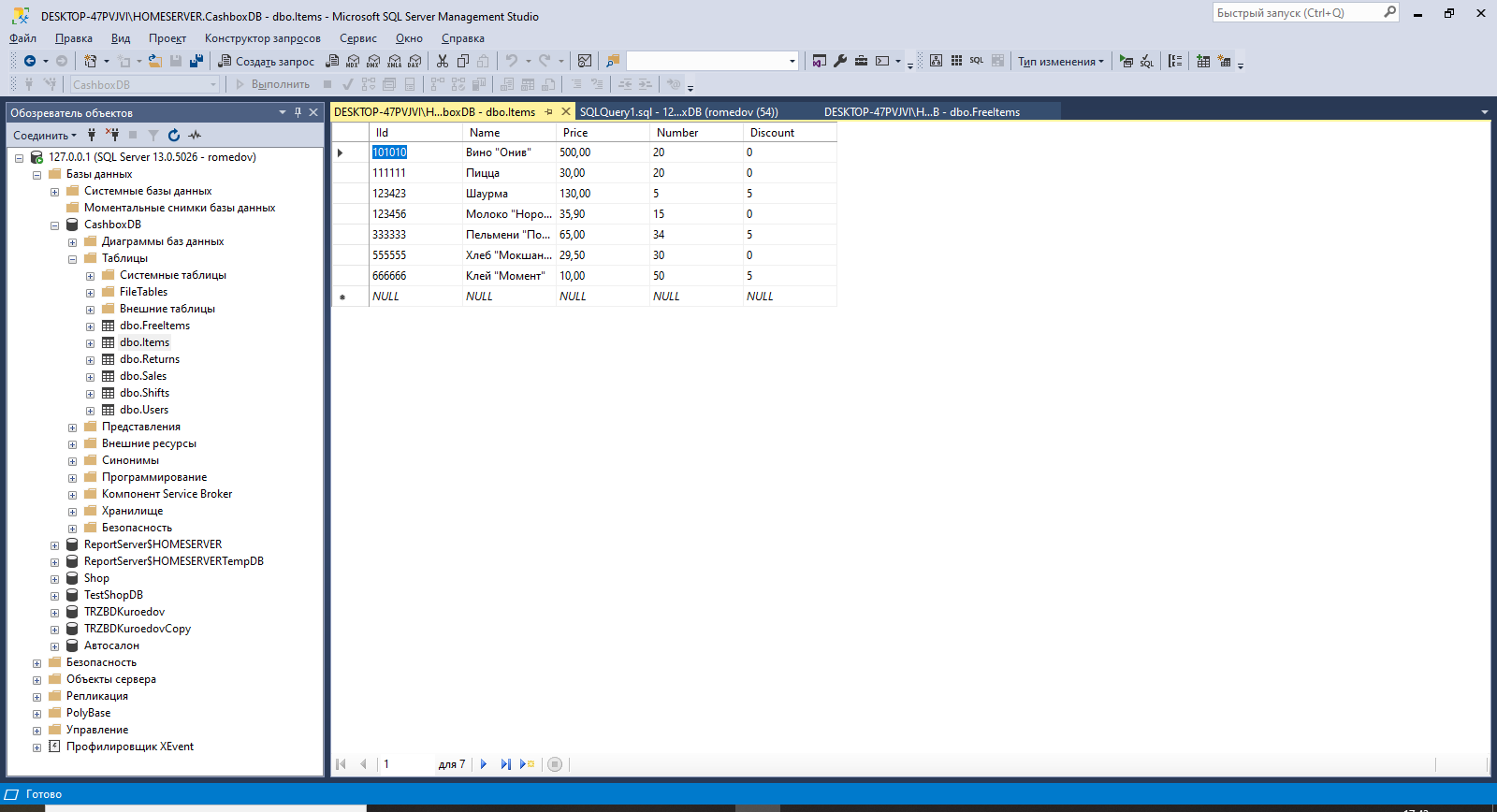


Рисунок А.2 ⎯ Таблица «Items»

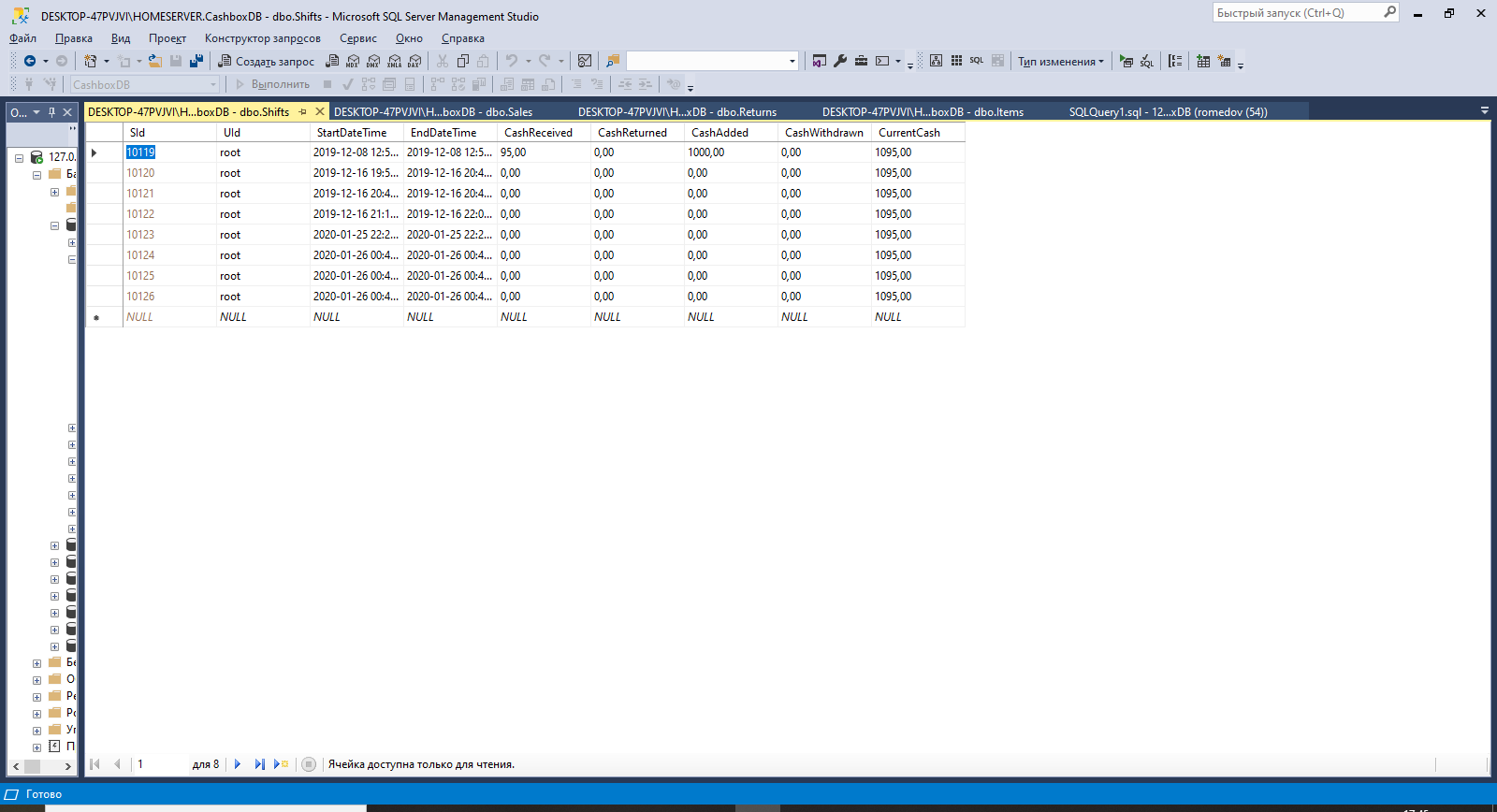


Рисунок А.3 ⎯ Таблица «Shifts»

Окончание ПРИЛОЖЕНИЯ А

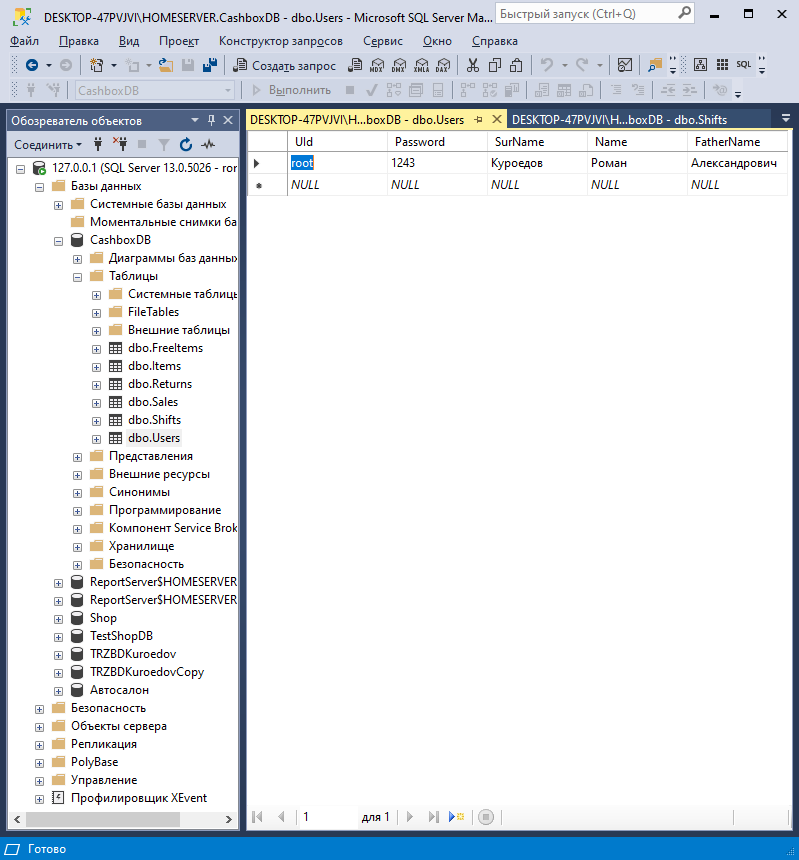


Рисунок А.4 ⎯ Таблица «Users»

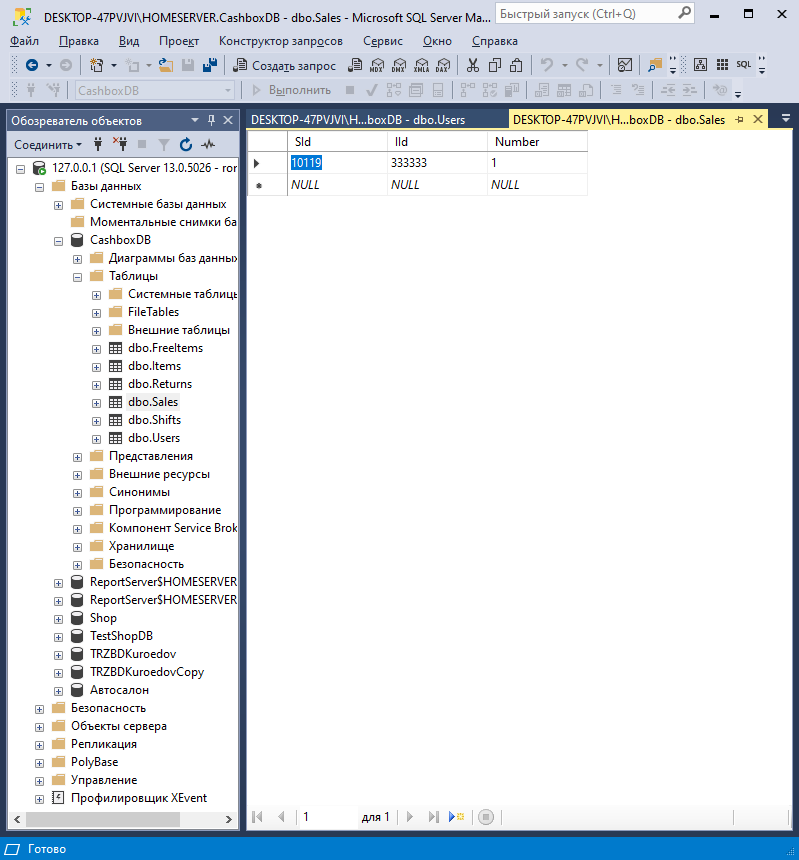


Рисунок А.5 ⎯ Таблица «Sales»

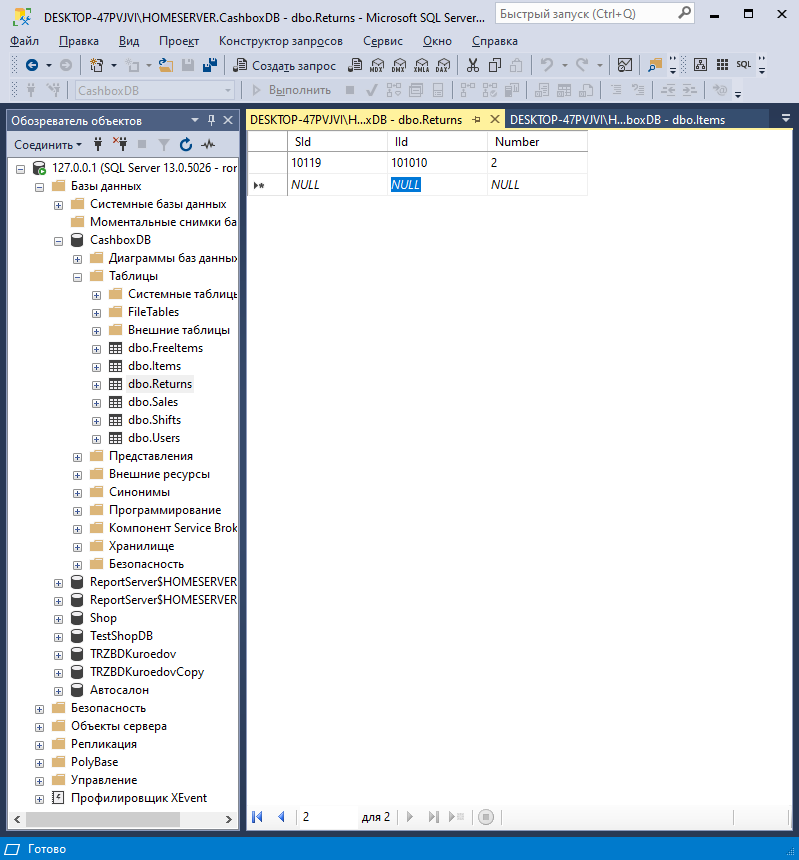


Рисунок А.6 ⎯ Таблица «Returns»