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 описания внешнего вида разрабатываемой системы

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 28 листов, 2 рисунка, 7 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.

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| [ВВЕДЕНИЕ](#Введение) | | 6 |
| [1 Теоретическое обоснование проекта](#Введение) | | 8 |
|  | [1.1 Анализ предметной области](#Анализ) | 8 |
|  | [1.2 Нормализация базы данных](#Нормализация) | 10 |
|  | [1.3 Выбор инструментария разработки](#Инструментарий) | 12 |
| [2 Проектная часть](#Проект) | | 21 |
|  | [2.1 Разработка базы данных](#Разработка) | 21 |
|  | [2.2 Организация защиты базы данных и СУБД](#Защита) | 23 |

|  |  |
| --- | --- |
| [ЗАКЛЮЧЕНИЕ](#Заключение) | 24 |
| [СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#Источники) | 26 |
| [ПРИЛОЖЕНИЕ А (обязательное) Скриншоты реализованного модуля](#ПриложениеА) | 27 |

**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 курсовой работы является разработка программного модуля   
он-лайн касса.

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.

Модель данных представлена на рисунке 1.

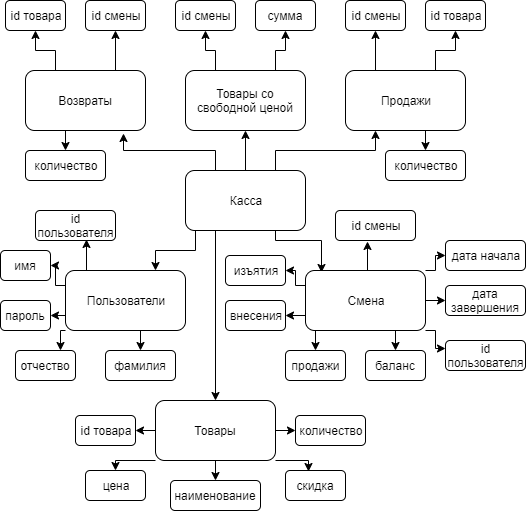


Рисунок 1 — Модель данных

ER-диаграмма базы данных представлена на рисунке 2.

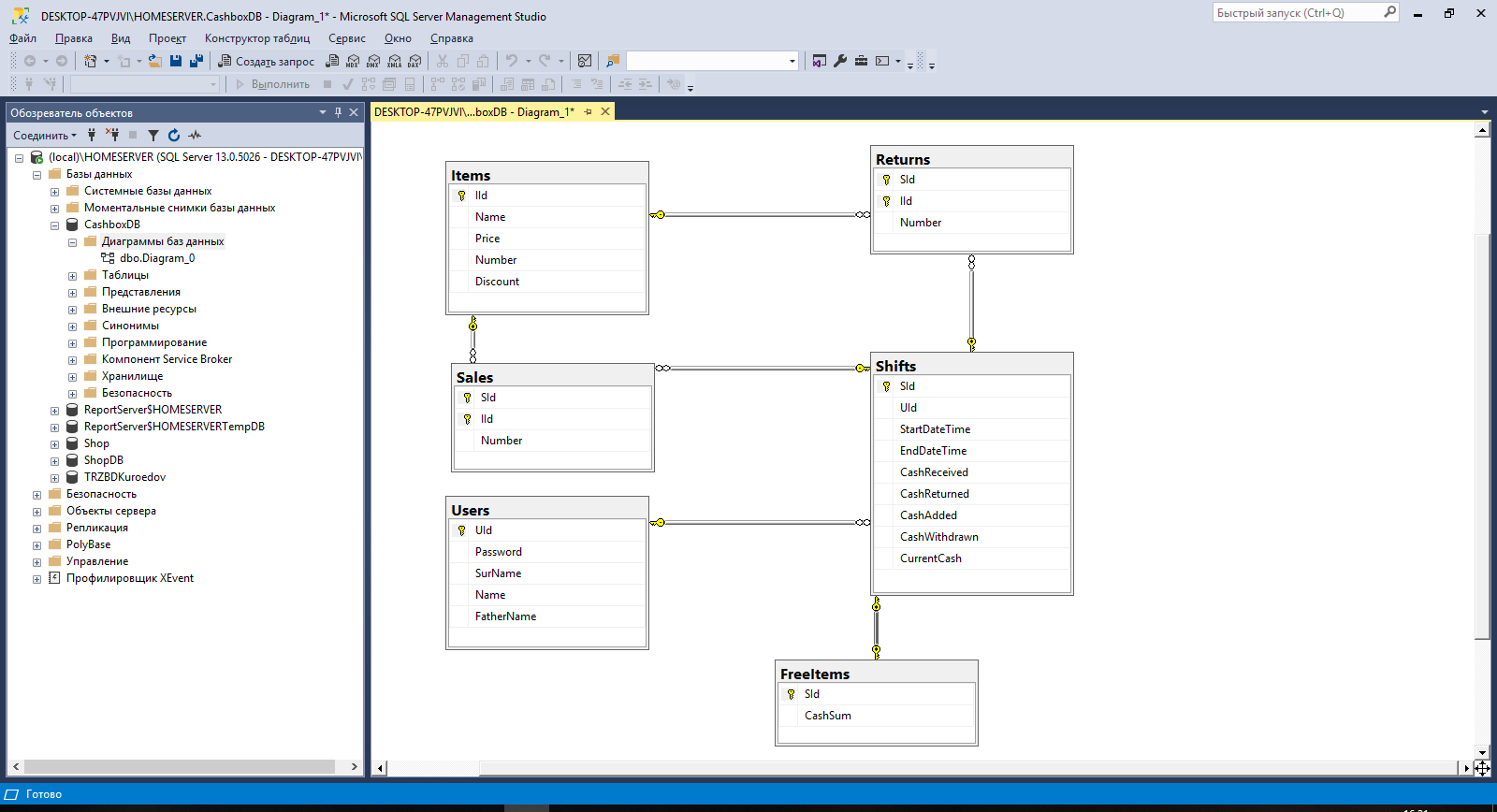


Рисунок 2 — ER-диаграмма базы данных

* 1. **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.

В общем случает для нормализации базы данных достаточно привести её таблицы с 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 1 — Соответствие таблиц нормальным формам.

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

* 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.** Неудивительно, что корпорация Oracle предлагает одноимённый продукт, с которого обычно начинается рассмотрение вариантов популярных СУБД. Oracle на данный момент имеет блестящую репутацию. Актуальная версия Oracle облачных сред и может быть размещена на одном или нескольких серверах, это позволяет управлять базами данных, которые содержат миллиарды записей. Некоторые из функций новейшей версии Oracle включают в себя использование как физических, так и логических структур. Это означает, что физическое управление данными не влияет на доступ к логическим структурам. Кроме того, безопасность в этой версии доведена до высочайшего уровня, потому что каждая транзакция изолирована от других.

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 — одна из самых популярных баз данных для веб-приложений. Фактически, является негласным стандартом для веб-серверов, которые работают под управлением операционной системы Linx.   
MySQL — это бесплатный пакет программ. Новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать как достоинством, так и недостатком — в зависимости от области внедрения. Эта СУБД позволяет выбирать различные движки для системы хранения, которые позволяют менять функционал инструмента и выполнять обработку данных, хранящихся в различных типах таблиц. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

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. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт и для операционной системы Linux. Одной из уникальных особенностей версии 2016 года является «Temporal Data Support» (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL Server поддерживает «Dynamic Data Masking» (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

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.** PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных   
веб-сайтов. Это была одна из первых разработанных систем управления базами данных, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использована на большинстве основных платформ, включая Linux. Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария. Движок базы данных может быть размещен в ряде сред, в том числе виртуальных, физических и облачных.

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.** Еще одна бесплатная база данных, которая имеет коммерческую версию — MongoDB. Она предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования. Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных, могут возникнуть проблемы производительности, при попытке использовать её таким образом. Однако, движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны. Последняя версия имеет новую подключаемую систему движков хранения. Документы могут быть проверены в процессе обновления или выполнения вставок, а функции текстового поиска были улучшены. Новая способность частичного индексирования может привести к более высокой производительности, уменьшая размер индексов.

Достоинства MongoDB:

* скорость и простота в использовании;
* движок поддерживает JSON и другие традиционные документы NoSQL;
* данные любой структуры могут быть сохранены/прочитаны   
  быстро и легко.

Недостатки MongoDB:

* SQL не используется в качестве языка запросов;
* инструменты для перевода SQL-запросов в MongoDB доступны, но их следует рассматривать именно как дополнение;
* установка может занять много времени.

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

**MariaDB.** Эта СУБД является бесплатной, но, как и многие другие бесплатные приложения, предлагает платные версии. Есть множество доступных плагинов расширений. Пожалуй, это самая быстро-развивающаяся СУБД на данный момент. MariaDB — это, фактически, ответвление от СУБД MySQL, разрабатываемое сообществом под лицензией GNU GPL. Разработку и поддержку MariaDB осуществляет компания MariaDB Corporation Ab и фонд MariaDB Foundation. Толчком к созданию стала необходимость обеспечения свободного статуса СУБД, в противовес политике лицензирования MySQL компанией Oracle. Система лицензирования MariaDB обязывает участников, желающих добавить свой код в основную ветку СУБД, обмениваться своими авторскими правами с MariaDB Foundation для охраны лицензии и возможности создавать критические исправления для MySQL. Ядро базы данных позволяет делать выбор из нескольких систем хранения, и это делает использование ресурсов более оптимизированным, что повышает производительность запросов и обработки данных. В состав MariaDB включена подсистемы хранения данных XtraDB для возможности замены InnoDB, как основной подсистемы хранения. Также включены подсистемы Aria, PBXT и FederateX. Она полностью совместима с MySQL, и прекрасно подходит в качестве замены, т.к. полностью соответствует как наборкоманд, так и API. Многие разработчики MySQL были вовлечены в процесс разработки, а сейчас принимают участие в развитии.

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 отправки запросов к базам данных, установления подключения, получения ответов от базы данных и осуществления ряда других операций.

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:

* 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.

Главным образом, через эти объекты и будет идти работа с базой данных.

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. Если традиционные средства ADO.NET позволяют создавать подключения, команды и прочие объекты для взаимодействия с базами данных, то 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.

Центральной концепцией Entity Framework является понятие   
сущности (entity). Сущность представляет собой набор данных, ассоциированных с определенным объектом (класс). Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Отличительной чертой Entity Framework является использование запросов, составленных 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.

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.

**2 Проектная часть**

**2.1 Разработка 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 2 — 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 3 — Структура таблицы FreeItems

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

Таблица 4 — Структура таблицы Returns

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

Таблица 5 — Структура таблицы Sales

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

Таблица 6 — 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 7 — 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 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 Windows. Данный тип проверки подлинности использует пользователя операционной системы Windows, на которой установлен SQL Server как пользователя самого сервера. Помимо этого, дополнительной защитой данных является механизм авторизации, действующий на уровне клиентского приложения.

**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**

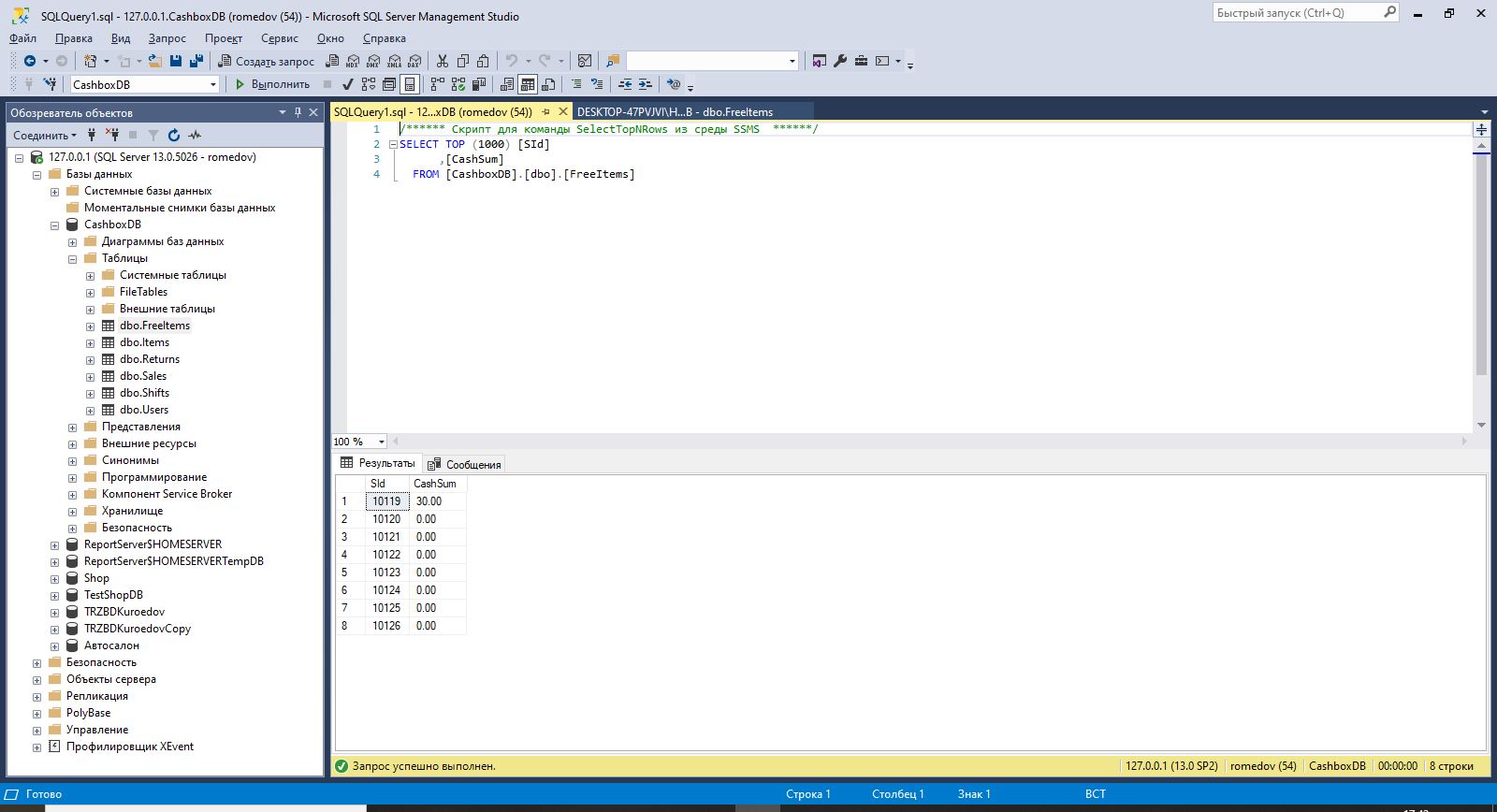


Рисунок А.1 ⎯ Таблица «FreeItems»

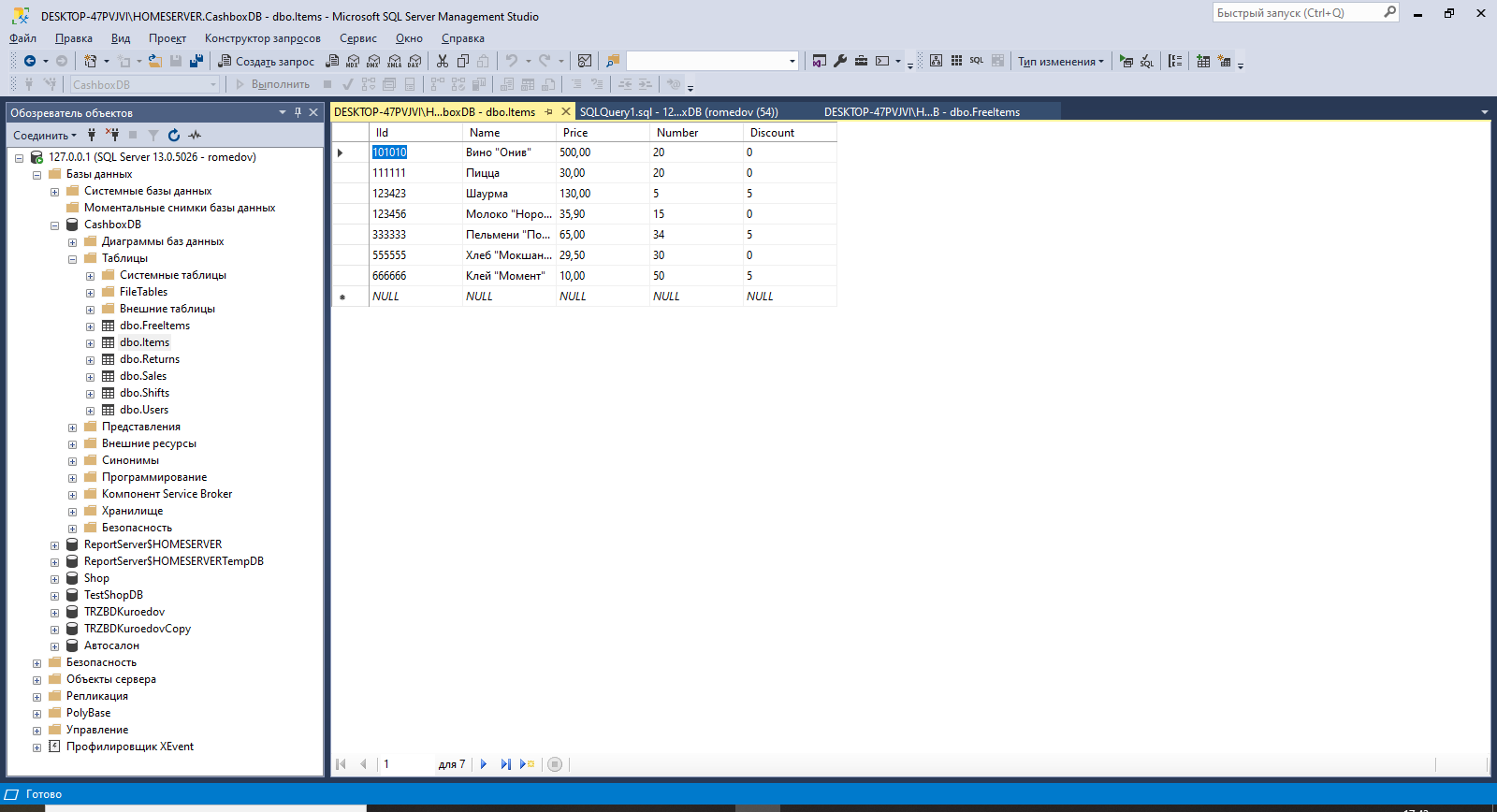


Рисунок А.2 ⎯ Таблица «Items»

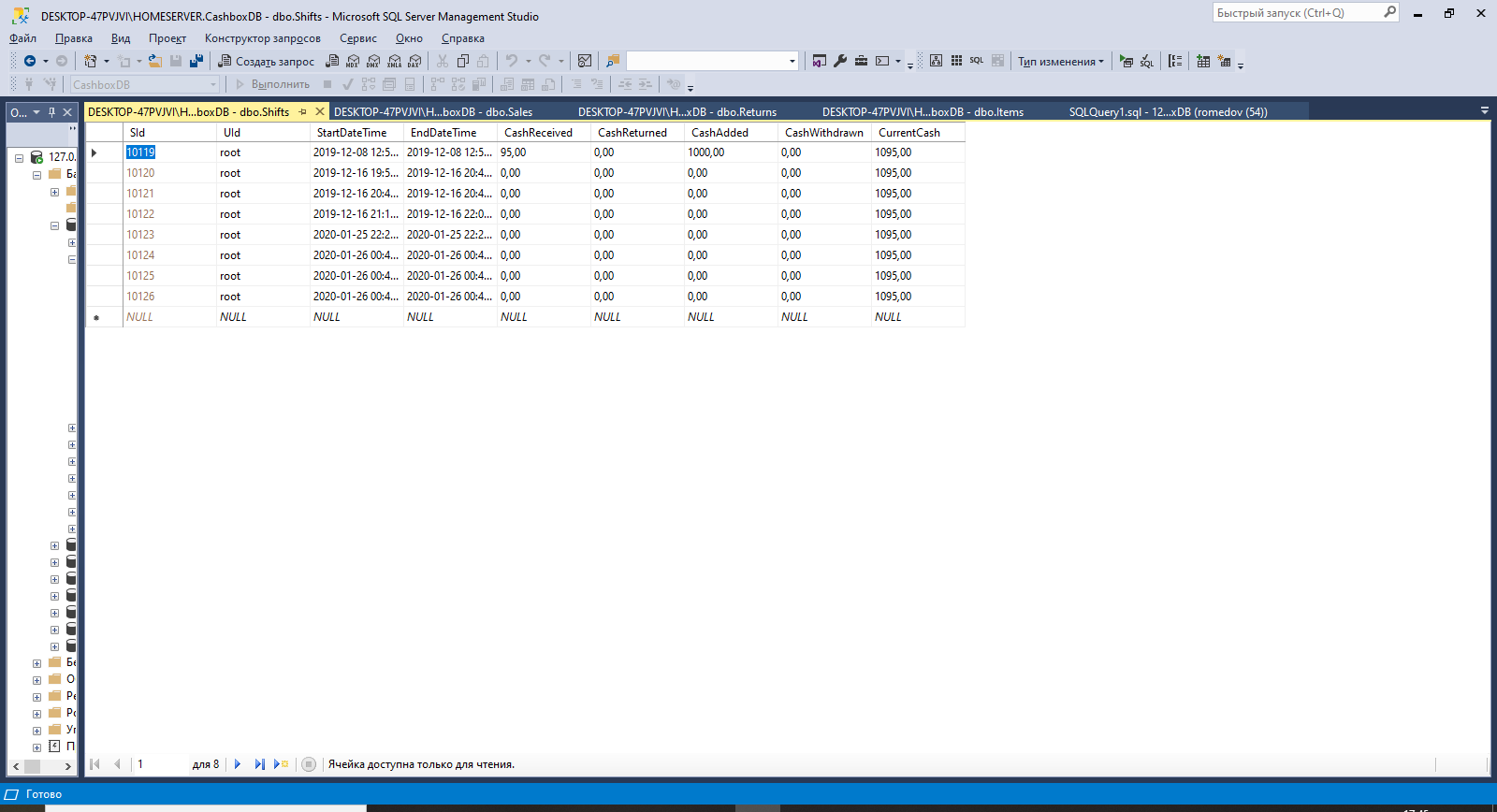


Рисунок А.3 ⎯ Таблица «Shifts»

Окончание ПРИЛОЖЕНИЯ А

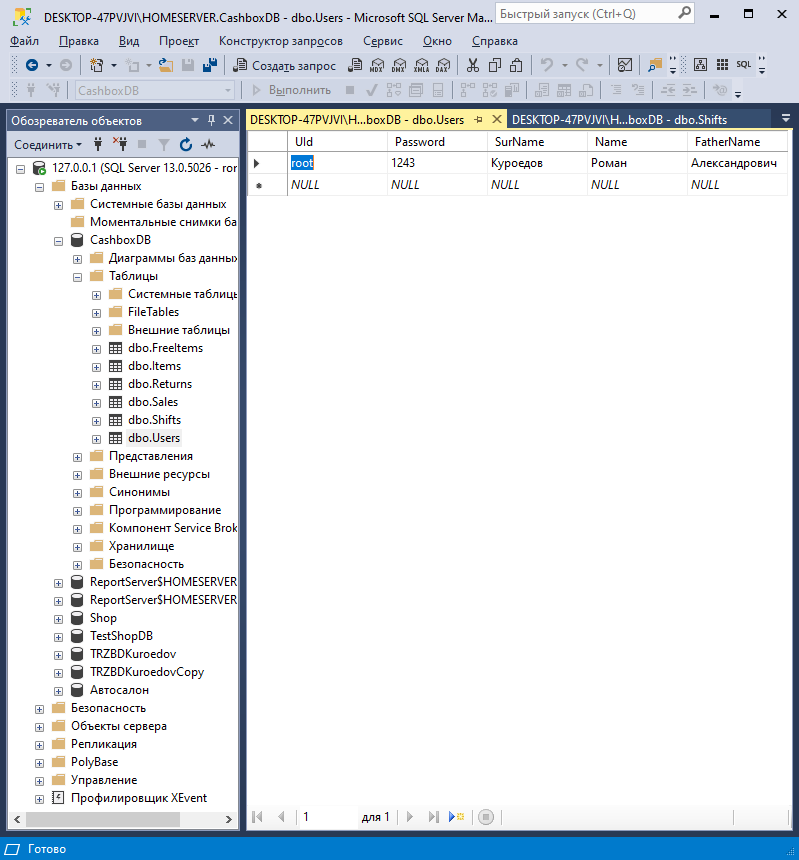


Рисунок А.4 ⎯ Таблица «Users»

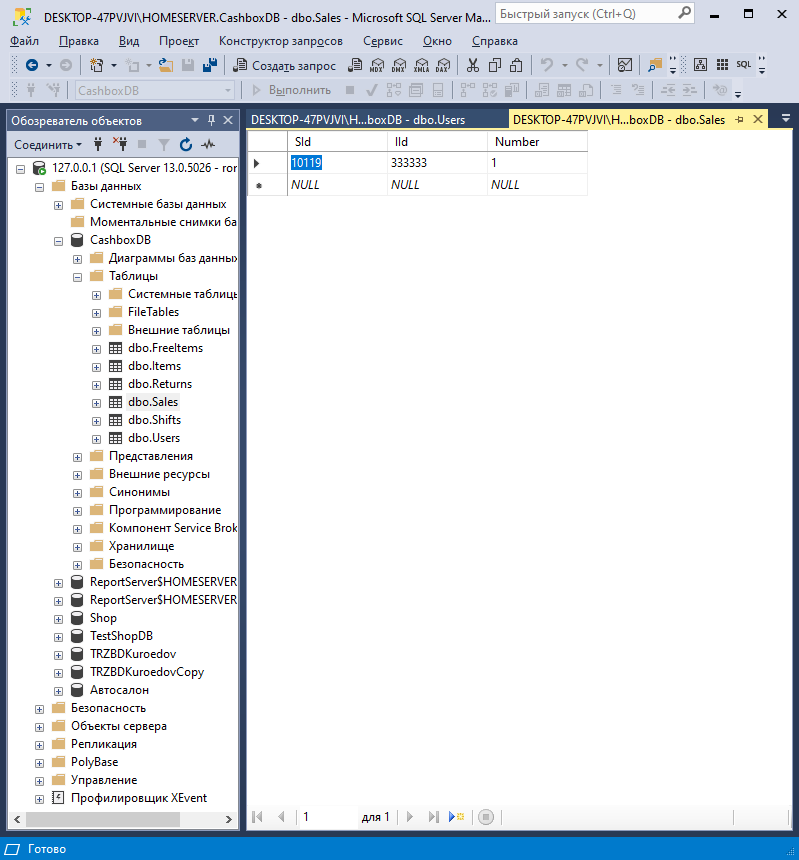


Рисунок А.5 ⎯ Таблица «Sales»

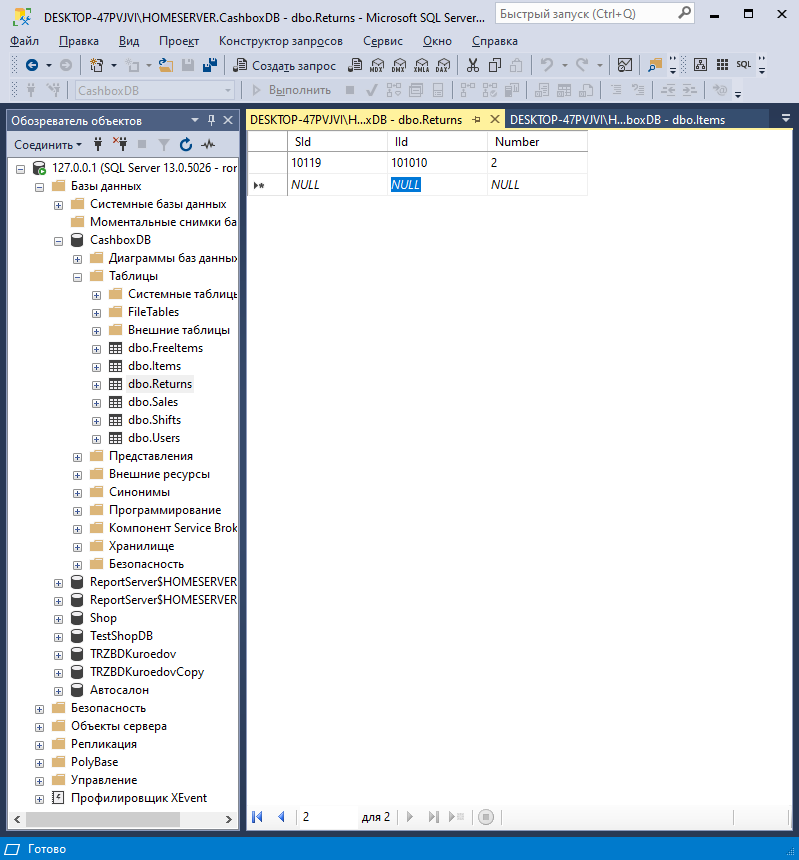


Рисунок А.6 ⎯ Таблица «Returns»