Содержание

[Введение……………………………………………………………………………….. 7](#_Toc137494289)

[1 Аналитический раздел……………………………………………………………8](#_Toc137494290)

[1.1 Описание деятельности ООО «Базис Про»………………………………… 8](#_Toc137494291)

[1.2 Анализ существующих технологий анализа ООО «Базис Про» и формирование требований…………………………………………………………… 10](#_Toc137494292)

[1.3 Анализ решений для разработки хранилища данных………………………… 12](#_Toc137494293)

[1.4 Проектирование архитектуры информационной системы…………………… 14](#_Toc137494294)

[1.5 Концептуальное моделирование данных информационной системы……….. 15](#_Toc137494295)

[2 Разработка информационно-аналитической системы………………………… 19](#_Toc137494296)

[2.1 Обоснование выбора платформы для создания системы…………………….. 19](#_Toc137494297)

[2.2 Реализация хранилища данных………………………………………………… 21](#_Toc137494298)

[2.3 Тестирование программного обеспечения…………………………………….. 28](#_Toc137494299)

[3 Информационная безопасность………………………………………………… 33](#_Toc137494300)

[3.1 Основные угрозы баз данных…………………………………………………... 33](#_Toc137494301)

[3.2 Особенности защиты баз данных………………………………………………. 35](#_Toc137494302)

[4 Безопасность жизнедеятельности……………………………………………… 36](#_Toc137494303)

[4.1 Микроклимат……………………………………………………………………. 37](#_Toc137494304)

[4.2 Производственное освещение………………………………………………….. 38](#_Toc137494305)

[4.3 Шум и вибрация………………………………………………………………… 40](#_Toc137494306)

[4.4 Безопасность технологических процессов и оборудования…………………. 41](#_Toc137494307)

[4.5 Отдых глаз………………………………………………………………………. 43](#_Toc137494308)

[Заключение……………………………………………………………………………. 44](#_Toc137494309)

[Перечень использованных информационных ресурсов…………………………… 45](#_Toc137494310)

[Приложение А (обязательное). Листинг программы……………………………… 48](#_Toc137494311)

# Введение

Хранение и обработка информации − ключевая задача для компаний любого размера. Это связано с тем, что ежедневно через нее проходят тысячи транзакций. Решения для хранения рутинной информации, такие как системы онлайн обработки транзакций (OLTP), не подходят для обработки и анализа данных для принятия решений [16].

В результате возникает необходимость в хранилище данных (ХД), чтобы использовать информацию без ущерба для работы OLTP-системы. Основная цель хранилища данных − поддержка принятия аналитических решений. Хранилища данных специально предназначены для создания отчетов, кубов для оперативной аналитической обработки (OLAP) и бизнес-аналитики для поддержки принятия решений в организации.

Хранилища данных предоставляют все необходимые механизмы для обеспечения постоянного доступа к важной для принятия решений информации в организации, что делает доступ к информации более удобным, а саму информацию − максимально надежной. Создание хранилища данных решает проблему сбора информации о текущей ситуации в компании для принятия решений.

Многолетний опыт использования этой технологии показывает её необычайную эффективность в улучшении качества управления, и, в конечном счете, в повышении конкурентоспособности предприятия.

В связи с этим было принято решение разработать для ООО «БАЗИС ПРО», г. Ростов-на-Дону систему информационно-аналитического обеспечения, которая позволит автоматизировать труд руководителей высшего и среднего звена организации и решать самые разнообразные задачи управления. **1 Аналитический раздел**

**1.1 Описание деятельности ООО «Базис Про»**

ООО «БАЗИС ПРО» − компания, основным видом деятельности которой является производство санитарно-технических работ, монтаж отопительных систем и систем кондиционирования воздуха.

Также развиваются следующие направления бизнеса:

* торговля оптовая станками;
* производство электромонтажных работ;
* производство кровельных работ;
* торговля оптовая машинами и оборудованием для добычи полезных; ископаемых и строительства;
* разборка и снос зданий.

Сегодня компания «БАЗИС ПРО» − это команда, состоящая из двадцати сотрудников. Компания имеет один офис в центре города.

В ООО «БАЗИС ПРО» для основной деятельности предприятия используются специализированное оборудование, для обслуживания клиентов и введение базы. Структура компании представлена на рисунке 1.

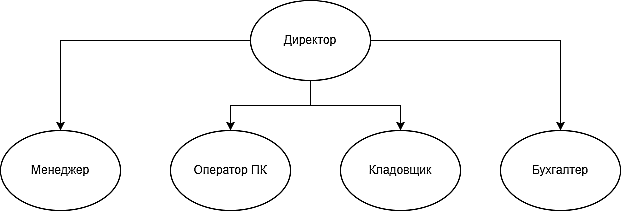


Рисунок 1 − Структура компании

Парк вычислительной техники ООО «Базис Про», насчитывает около 20 персональных компьютеров для работников и 4 персональных компьютеров для администрации. Все компьютеры связаны между собой локальной сетью, компьютеры администрации имеют возможность выхода в Интернет.

Для формирования наиболее точных требований к системе необходимо изучить предметную область компании. Рассмотрим подробнее процесс продажи на рисунках 2 и 3.

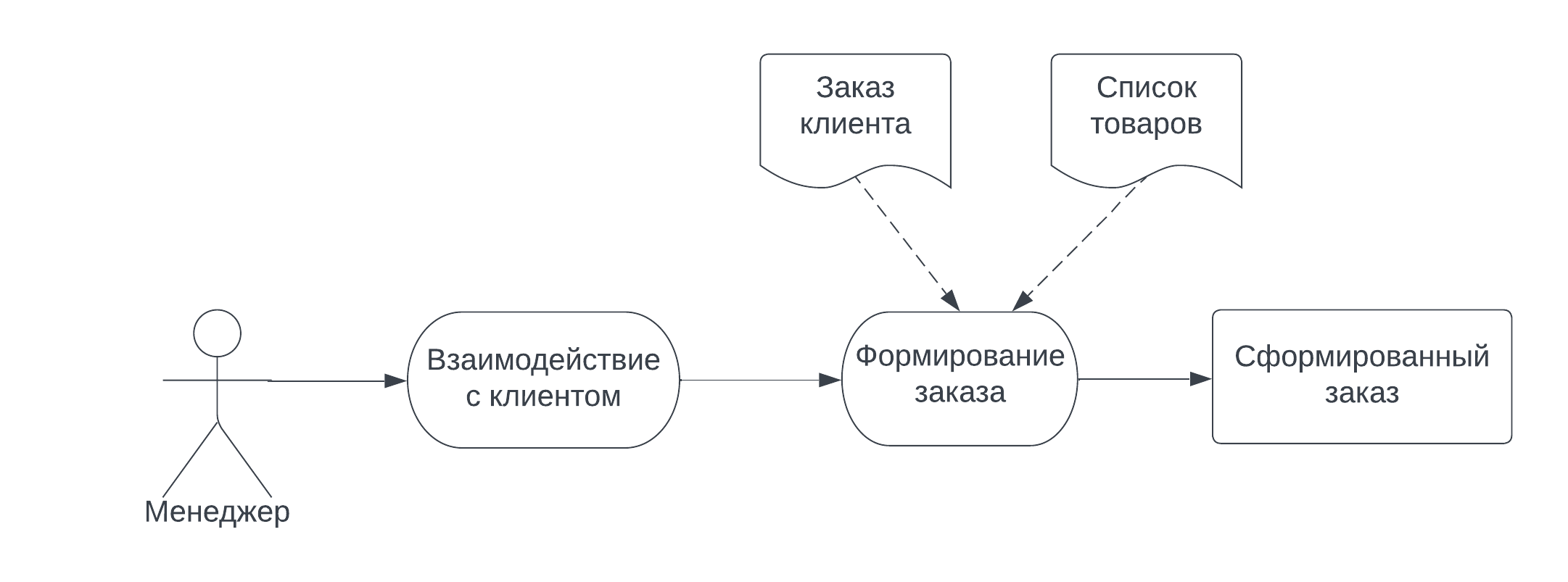


Рисунок 2 − Процесс формирование заказа

Данная диаграмма показывает процесс формирования заказа, клиенты делают заказ в офисе компании при помощи менеджера.

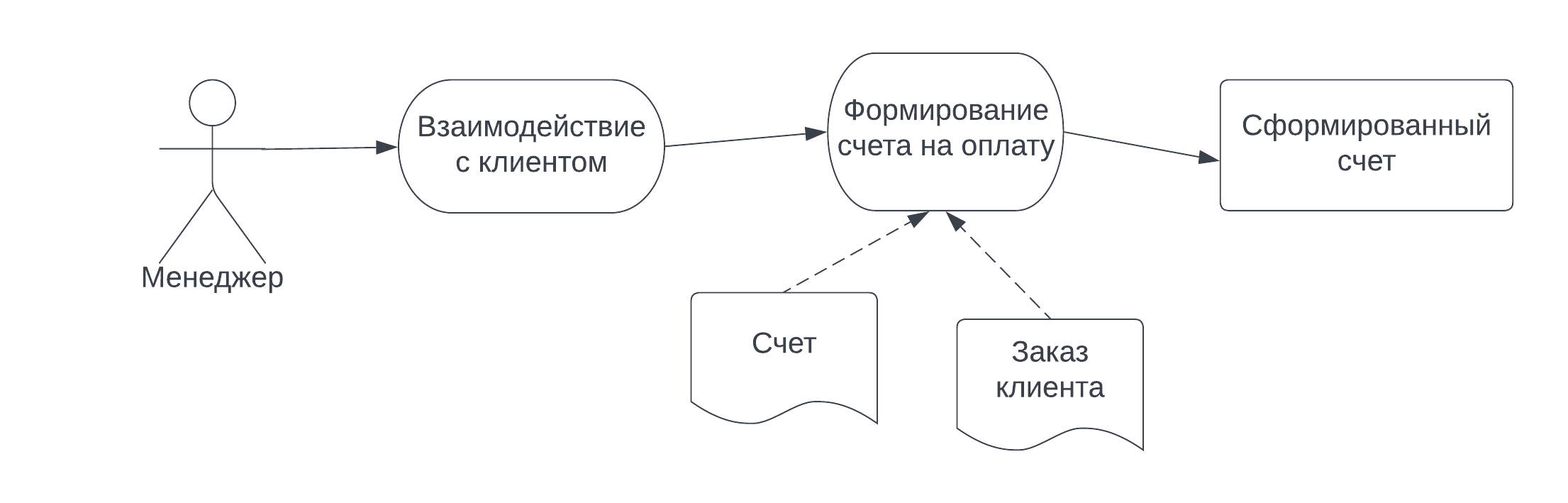


Рисунок 3 − Процесс формирования счета

Данная диаграмма показывает процесс формирования счета на оплату, менеджер формирует счет исходя из заказа клиента.

Отчеты и документы о продажах заполняются вручную. Во-первых, данный процесс занимает много времени и увеличивает трудоемкость, во-вторых данные поступают в базу данных очень медленно, что не дает проводить анализ над актуальными данными бизнеса.

Необходимо учитывать данные особенности при формировании требований к разрабатываемой информационно-аналитической системе, сделать данные процессы более эффективными для экономии времени и затрат компании.

## 1.2 Анализ существующих технологий анализа ООО «Базис Про» и формирование требований

Автоматизация процессов анализа данных в компании требует использования инструментов обработки информации. Такие инструменты позволяют более эффективно обрабатывать данные, анализировать и принимать на их основе решения для развития бизнеса. Для этого требуется провести анализ существующих инструментов, используемых в компании, и в конце выбрать наиболее подходящее инструментальное решение [5].

Для точного анализа была создана диаграмма формирования отчета в компании на текущий момент, она представлена на рисунке 4.

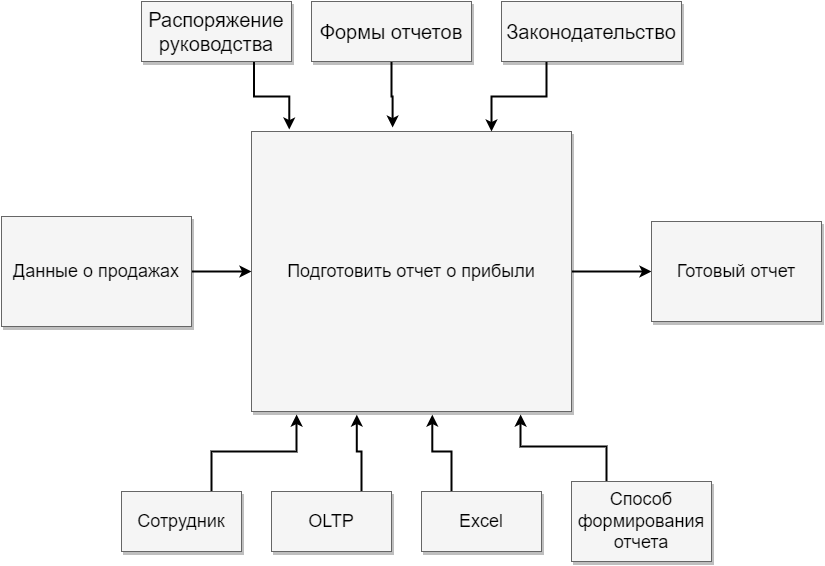


Рисунок 4− Диаграмма обработки данных

Диаграмма показывает:

* входные и выходные данные;
* механизмы для обработки данных;
* управление процессами.

Процесс формирования отчета начинается со сбора данных. Сотрудники собирают информацию из OLTP базы данных или Excel файлов. Для анализа данных в организации используется Microsoft Excel. Этот инструмент обладает достаточной функциональностью для построения анализа [1].

Так как данные для анализа в компании собираются из разных источников и обрабатываются сотрудниками, ответственными за определенные виды деятельности. Поэтому ежедневные отчеты, составляемые вручную, не содержат полной информации, часто содержат неактуальную и противоречивую информацию и не полностью отражают ситуацию в компании.

Совмещение в одной базе данных выполнения длинных аналитических запросов и выполнения транзакций десятками операторов неизбежно приводит к борьбе за вычислительные ресурсы. При этом до недопустимых величин может увеличиться время выполнения транзакций и время получения аналитических выборок [8,9].

Анализ предметной области определил основные задачи, которые необходимо автоматизировать при разработке информационной-аналитической системы (ИАС) для компании ООО «Базис Про». Были выявлены недостатки, касающиеся источников данных и скорости обработки информации, так как документы приходится получать вручную в разных местах. Для упрощения анализа данных и получения общей информации о продажах компании необходимо построить информационно-аналитическую систему, которая обеспечила бы:

* хранение и обработку данных;
* анализ основных бизнес-процессов предприятия;
* беспрерывный доступ к хранилищу данных.

## 1.3 Анализ решений для разработки хранилища данных

Хранилища данных предназначены для обработки и анализа данных для принятия решений на предприятии. В отличие от OLTP-систем, используемых для повседневной работы пользователей, данные в хранилище данных более стабильны и загружаются через регулярные промежутки времени из различных источников. Модель хранилища данных показана на рисунке 5.

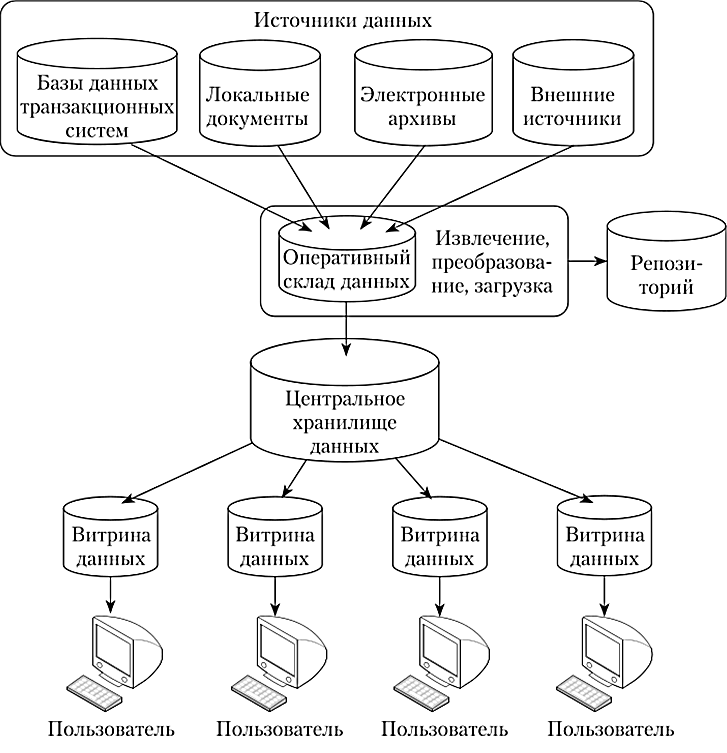


Рисунок 5− Модель хранилища данных

ХД предусматривает использование следующих компонентов:

* источники оперативной информации;
* инструменты для передачи и преобразования данных;
* реляционное хранилище;
* инструменты для анализа данных.

Оперативные данные, собранные из различных источников, очищаются, интегрируются и помещаются в реляционное хранилище. Эти данные могут быть проанализированы с помощью инструментов отчетности. Данные подготавливаются к передаче и преобразованию с помощью инструментов OLAP-анализа и затем могут быть помещены в специальную OLAP-базу данных или оставлены в хранилище.

Концепция OLAP представляет собой многомерную базу данных, так называемый куб, в котором пользователь может расширять интересующие его измерения, который показан на рисунке 6.

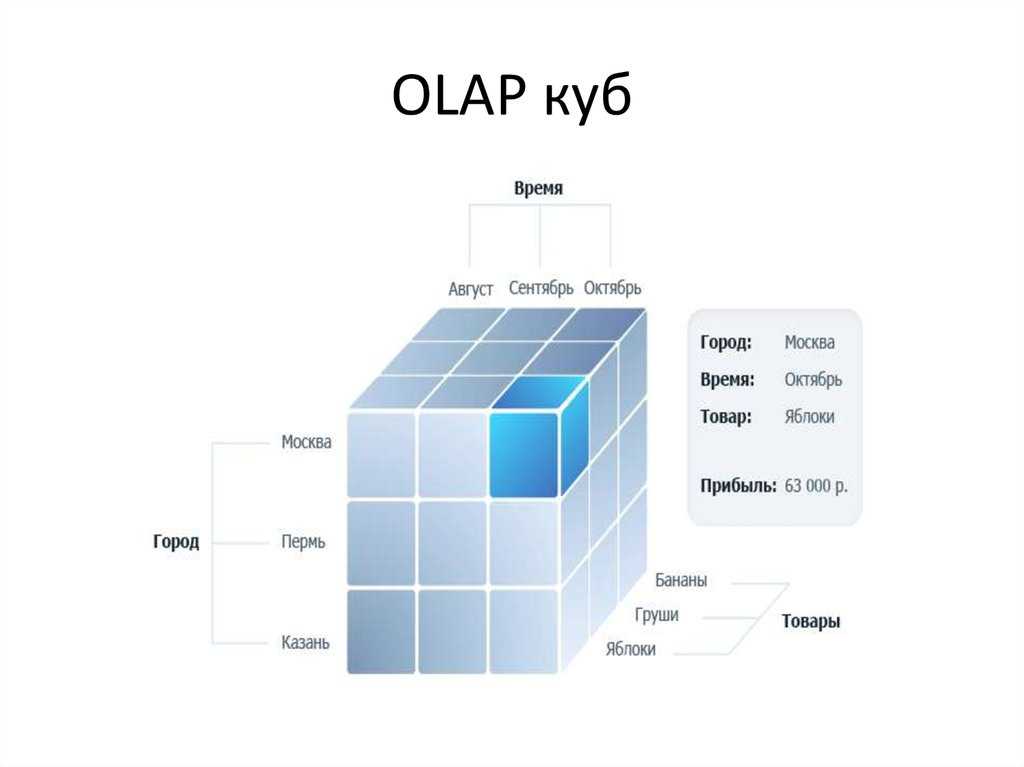


Рисунок 6− OLAP-концепция

Существует три типа OLAP:

* многомерный OLAP (MOLAP);
* реляционный OLAP (ROLAP);
* гибридный OLAP (HOLAP).

MOLAP часто называют просто OLAP, поскольку это классическая форма OLAP; MOLAP использует сводную базу данных для создания необходимой схемы многомерных данных, сохраняя как базовые данные, так и агрегаты.

ROLAP работает непосредственно с реляционными базами данных, при этом таблицы фактов и измерений хранятся в реляционных таблицах, а для хранения агрегатов создаются дополнительные реляционные таблицы.

HOLAP использует реляционные таблицы для хранения базовых данных и многомерные таблицы для агрегирования.

Решение, предоставленное компанией Microsoft, было наиболее подходящим, так как обработка данных ранее осуществлялась в программных продуктах Microsoft Office, было решено использовать подход ROLAP.

## 1.4 Проектирование архитектуры информационной системы

Архитектура системы − это структура программы или вычислительной системы, включающая программные компоненты, внешне видимые свойства этих компонентов и отношения между компонентами.

Сегодня на рынке информационных технологий представлен широкий спектр инструментальных средств, предназначенных для быстрой реализации компонентов архитектуры ИАС. Использование таких инструментов позволяет не разрабатывать аналитические приложения заново, а воспользоваться готовыми современными технологиями и, следовательно, сократить время и затраты на их создание [2].

Решение задачи обеспечения пользователей информацией в ИАС определяется в основном правильным подбором инструментов делового анализа. Но немаловажным является и выбор инструментов поддержки процессов извлечения, преобразования, загрузки и хранения данных [6].

Все системы информационно-аналитического взаимодействия построены на концепции хранилищ данных. В общем случае структура таких систем описывается тремя уровнями:

* извлечение, преобразование и загрузка;
* хранение данных;
* анализ и обработка данных.

Данные поступают из различных источников, изучаются, обрабатываются, преобразуются в нужный тип и загружаются в хранилище. Затем пользователи могут использовать специализированные инструменты для создания отчетов и проведения анализа. Архитектура системы показана на рисунке 7.

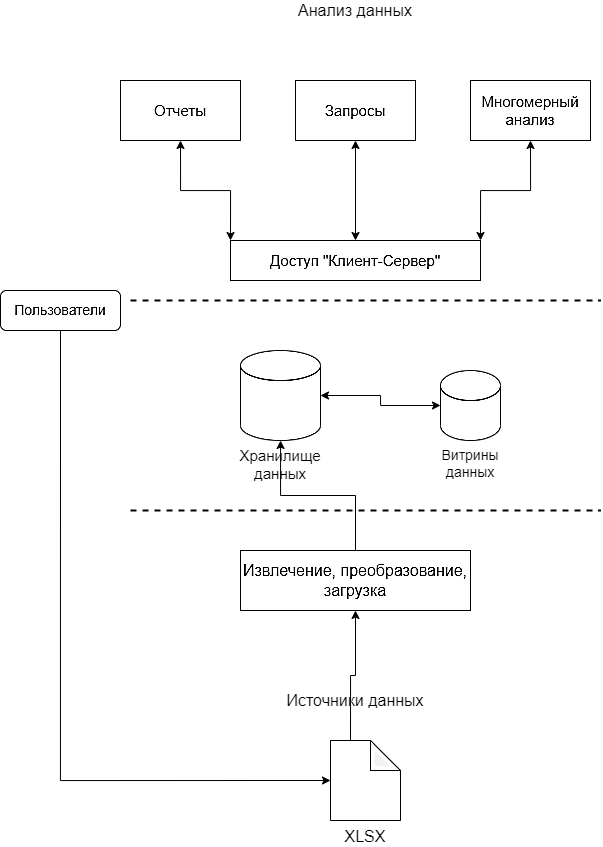
****

Рисунок 7− Архитектура системы

## 1.5 Концептуальное моделирование данных информационной системы

Логическое проектирование − это процесс создания модели будущей информационной системы, без учета физических аспектов или фактического типа системы управления базой данных. Это моделирование должно быть основано на концептуальной модели будущей ИС, определенной на этапе приобретения.

Логическая модель информационной системы отражает логические связи между атрибутами объектов вне зависимости от их содержания и среды хранения. Другими словами, логическая модель отображает логические связи между информационными данными в данной концептуальной модели.

Логическая модель данного программного продукта показана на рисунке 8.

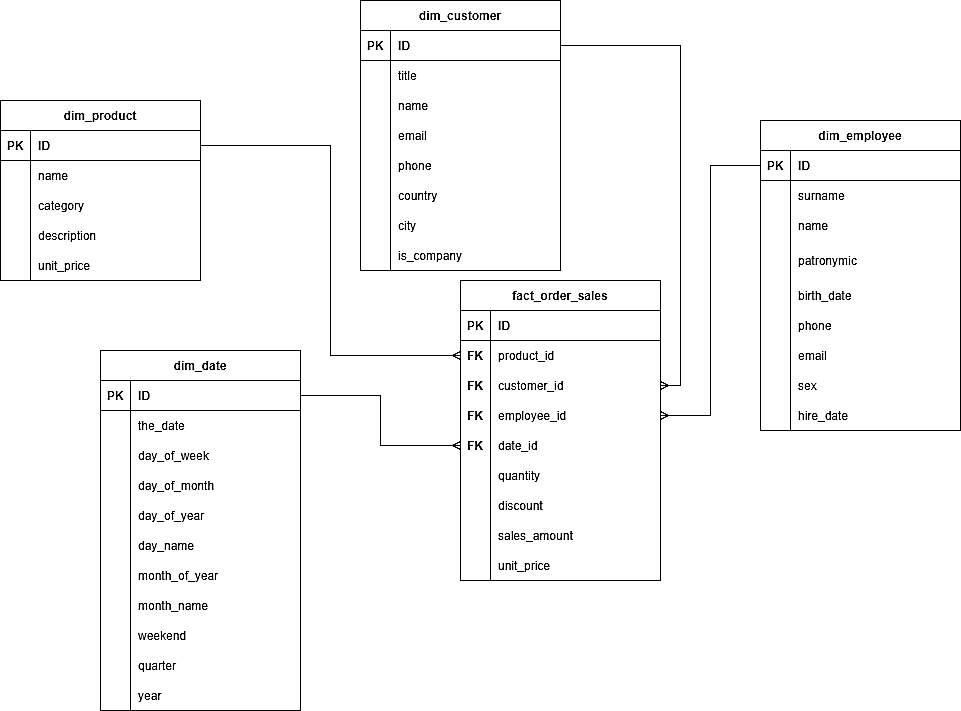


Рисунок 8− Логическая модель ХД

Перед внедрением хранилища данных (ХД) и OLAP необходимо выбрать схему, которая будет использоваться. На основании критериев, полученных на этапе разработки требований, рекомендуется ROLAP − схема многомерных данных, использующая звездообразную схему представления данных [15].

Эта схема представляет собой звезду с таблицей фактов в центре, окруженную таблицей измерений: схема звезды охватывает часть бизнес-деятельности предприятия, а таблица фактов связана с таблицей измерений внешним ключом. Схема показана на рисунке 9.

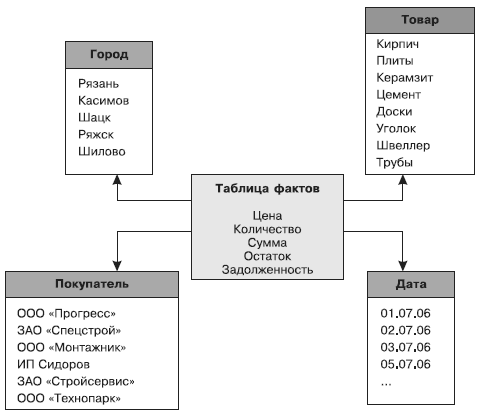


Рисунок 9 − Схема «Звезда»

Физическая модель данных используется конкретной СУБД для описания данных, а ограничения, присутствующие в логической модели данных, реализуются различными инструментами СУБД, такими как индексы, декларативные ограничения согласованности и триггеры.

Когда физическая модель данных реализуется реляционной СУБД, отношения, созданные на этапе генерации логической модели данных, преобразуются в таблицы, атрибуты преобразуются в столбцы таблиц, для ключевых атрибутов создаются уникальные индексы, а домены преобразуются в типы данных, принятые в конкретной СУБД.

Основными объектами логической модели данных являются сущности, атрибуты и взаимосвязи. Физическая модель данных, как правило, создается на основе логической, поэтому каждому объекту логической модели соответствует объект физической модели (хотя соответствие может быть неоднозначным). В физической модели данных сущности логической модели данных соответствует таблица, экземпляру сущности – строка в таблице, а атрибуту – колонка таблицы. Кроме перечисленных выше объектов, физическая модель может содержать объекты, тип которых зависит от СУБД: индексы, представления, последовательности, триггеры, процедуры и т.п. Если в логической модели данных не имеет большого значения, какой конкретно тип данных у атрибута, то в физической важно описать всю информацию о конкретных объектах.

Физическая модель показана на рисунке 10.

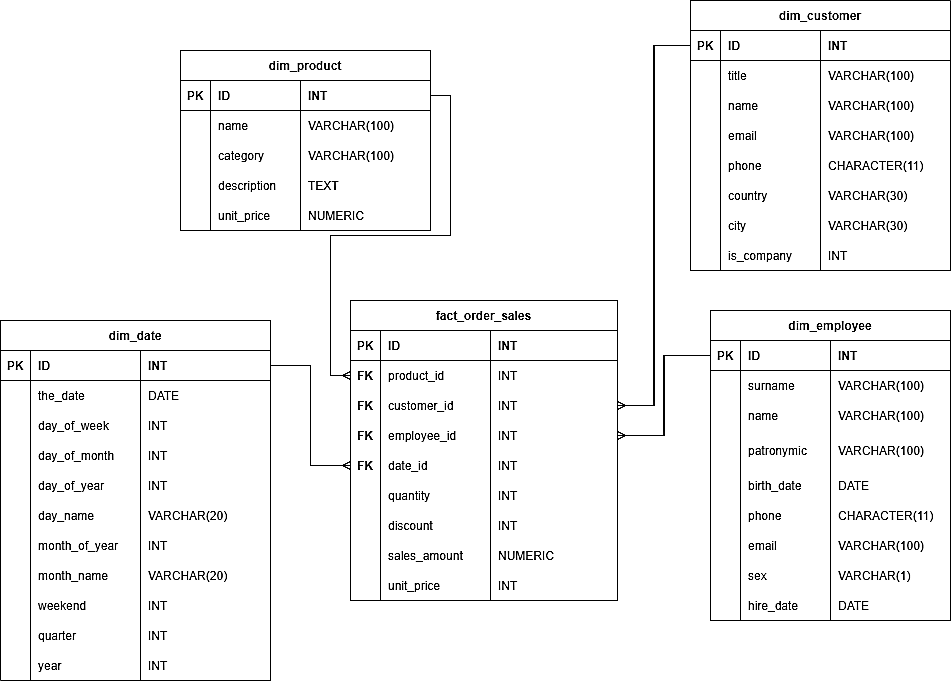


Рисунок 10 − Физическая модель ХД

**2 Разработка информационно-аналитической системы**

**2.1 Обоснование выбора платформы для создания системы**

Хранилище данных (ХД) − это центральное хранилище информации, используемое для анализа данных и принятия обоснованных решений в организации. Поскольку данные поступают из различных источников, часто в регулярном цикле, ХД обеспечивает такие преимущества, как возможность принятия обоснованных решений, интеграция данных из различных источников, исторический анализ данных, а также высокое качество, согласованность и точность данных.

Процесс создания хранилища данных включает в себя следующие шаги:

* определение бизнес-требований и целей хранилища данных;
* анализ источников данных и выбор тех, которые будут использоваться в хранилище;
* разработка модели данных, которая определяет, как данные будут храниться, и организовываться в хранилище;
* разработка процессов загрузки данных в хранилище;
* разработка процессов обновления данных в хранилище;
* разработка процессов анализа данных и создание отчетов;
* тестирование и внедрение хранилища данных.

Хранилища данных можно создавать самостоятельно или использовать готовые решения, предоставляемые различными поставщиками, например, Microsoft. Однако в обоих случаях процесс создания хранилища данных требует тщательного анализа бизнес-требований и источников данных, а также разработки модели данных и процессов загрузки и анализа данных.

Для разработки системы анализа данных было использовано программное обеспечение Microsoft SQL Server 2014 от компании Microsoft. Это реляционная система управления базами данных (РСУБД), разработанная компанией Microsoft. Она предназначена для создания, хранения, управления и анализа данных в организации. Компонентами Microsoft SQL Server 2014 являются:

SQL Server Database Engine − это основной компонент, который обеспечивает хранение и обработку данных в базе данных. Он включает в себя движок базы данных, который управляет хранением, обработкой и защитой данных [11].

SQL Server Analysis Services (SSAS) − это компонент, который обеспечивает анализ данных и создание многомерных кубов. Он позволяет пользователям анализировать данные из разных источников и создавать отчеты [17].

SQL Server Integration Services (SSIS) − это компонент, который обеспечивает интеграцию данных из разных источников в базу данных. Он позволяет пользователям создавать пакеты ETL (Extract, Transform, Load), которые загружают, очищают и преобразуют данные перед их загрузкой в базу данных.

SQL Server Reporting Services (SSRS) − это компонент, который обеспечивает создание и распространение отчетов. Он позволяет пользователям создавать отчеты на основе данных из базы данных и распространять их в различных форматах.

SQL Server Management Studio (SSMS) − это инструмент для управления базами данных SQL Server. Он позволяет пользователям управлять базами данных, создавать и изменять таблицы, индексы, процедуры и другие объекты базы данных [18].

SQL Server представляет решение, которое делает работу пользователей компании более безопасной, надежной и продуктивной. Различные инструменты SQL Server 2014 предоставляют разработчикам и пользователям дополнительные возможности, тем самым упрощая работу с разработкой решений для анализа данных и аналитических приложений. Благодаря этому, а также большому набору компонент представленное решение повышает эффективность хранения данных в различных компаниях.

Также для написания ETL-процесса обработки данных был выбран язык программирования Python и библиотека этого языка Pandas [7,12].

Python − это высокоуровневый, динамически типизированный, интерпретируемый язык программирования, используемый в самых разных областях, включая анализ данных, машинное обучение, веб-разработку, науку о данных и разработку программного обеспечения. Python был создан Гвидо ван Россумом в 1991 году. Разработанный Software Foundation и Гвидо ван Россумом, Python является полностью объектно-ориентированным языком программирования, в котором все является объектом. Python имеет множество преимуществ, таких как простота, читаемость кода, общность, эффективность и письменная форма. Он имеет множество преимуществ, включая переносимость программ [13].

Библиотека pandas, библиотека для языка программирования Python, предоставляет полезные инструменты для работы с данными, включая эффективное манипулирование данными и структуры данных для анализа. Она широко используется в науке о данных и машинном обучении благодаря своим мощным возможностям, таким как манипулирование таблицами, фильтрация, сортировка, группировка, агрегация и объединение. Библиотека pandas позволяет легко выполнять расширенный анализ и визуализацию данных, CSV, Excel, SQL и т.д. Она предоставляет удобный интерфейс для работы с данными из различных источников [19].

## 2.2 Реализация хранилища данных

На основе физической модели данных будет создано хранилище данных с помощью языка SQL. Реализация ХД не отличается от реализации баз данных, но имеет некоторые особенности. Одна из них заключается в том, что ХД содержит не оперативную информацию, поэтому не требуется создание журнала транзакций. Вместо этого можно выбрать простой режим восстановления, который характеризуется поддержкой минимальных требований для хранения журнала транзакций на диске и автоматическим освобождением места в журнале. Код создания БД представлен на рисунке 11.

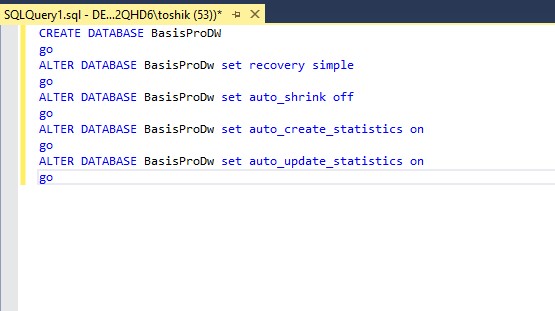


Рисунок 1 − Код создания базы данных

После создания базы данных необходимо создать таблицы, которые в данном случае называются таблицами измерений и таблицами фактов.

Таблица измерений является одним из ключевых элементов хранилища данных и используется для анализа данных. Эта таблица содержит описательные данные, которые используются для анализа данных в таблице фактов. Таблица измерений содержит информацию об измерениях для анализа данных, такую как дата, время, местоположение, продукт, клиент и т.д.

Таблица измерений содержит ключи, которые связывают с таблицей фактов, и атрибуты, которые описывают измерения. Атрибуты могут быть иерархическими. Это означает, что они могут содержать подуровни, чтобы данные можно было анализировать на разных уровнях детализации. Например, атрибут "дата" может содержать подуровни "год", "месяц" и "день", что позволяет анализировать данные за различные временные интервалы.

Создание таблицы измерений «Date», код представлен на рисунке 12.

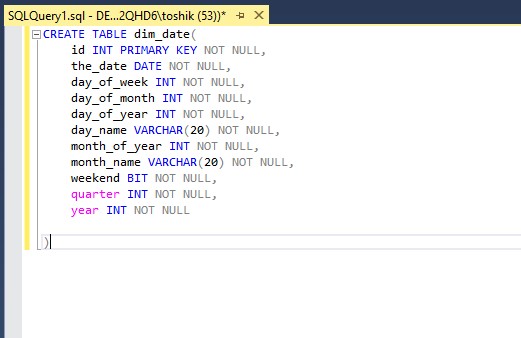


Рисунок 12 − Код создания таблицы "dim\_date"

Создание таблицы измерений «Employee», код представлен на рисунке 13.

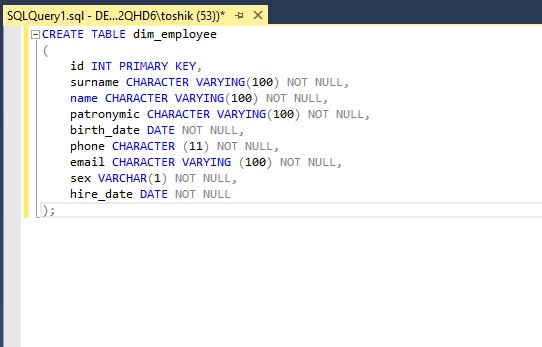


Рисунок 13 − Код создания таблицы "dim\_employee"

Создание таблицы измерений «Customer», код представлен на рисунке 14.

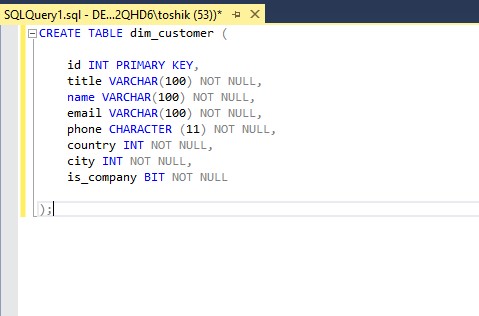


Рисунок 14 − Код создания таблицы "dim\_customer"

Создание таблицы измерений «Product», код представлен на рисунке 15.

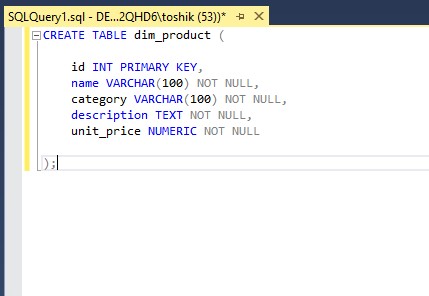


Рисунок 15 − Код создания таблицы "dim\_product"

Следующим шагом является создание таблицы фактов. Таблица фактов является главной таблицей хранилища данных и содержит информацию об объектах или событиях, совокупность которых будет анализироваться позже. Она содержит фактические данные, отражающие события или операции, произошедшие в определенный момент времени.

Таблица фактов содержит таблицы измерений и ключи, связанные с измерениями, которые представляют собой значения, которые можно агрегировать и анализировать. Например, таблица фактов для интернет-магазина содержит ключи измерений, относящиеся к клиентам, продуктам, датам и времени заказа, количеству продуктов, проданных в каждом заказе, и общей стоимости каждого заказа.

Создание таблицы фактов «Sales», код представлен на рисунке 16.

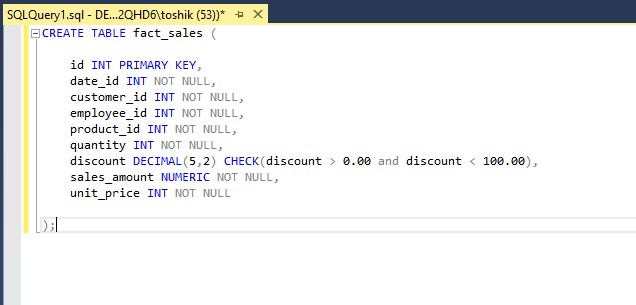


Рисунок 16 − Код создания таблицы "fact\_sales"

После создания всех таблиц их необходимо связать между собой с помощью связей на основе первичных и внешних ключей. Отношение в базе данных - это связь между таблицами, которая позволяет объединять данные из разных таблиц. Связи обычно создаются с использованием ключевых полей таблицы. Ключевое поле - это поле или группа полей, которые однозначно идентифицируют каждую запись в таблице. Существует несколько типов связей в базе данных

* один-к-одному: каждая запись в таблице связана с записью в другой таблице.
* один-ко-многим: каждая запись в одной таблице связана с несколькими записями в другой таблице.
* многие-ко-многим: каждая запись в одной таблице связана с несколькими записями в другой таблице, и наоборот.

Связи между таблицами позволяют объединять данные из разных таблиц, что делает базы данных более эффективными и гибкими. Они также упрощают анализ данных и принятие решений, поскольку данные из нескольких таблиц могут быть легко запрошены.

Связи между таблицами могут быть установлены с помощью внешних ключей. Внешний ключ − это атрибут или набор атрибутов, который ссылается на первичный или уникальный ключ в другой таблице. Внешний ключ − это как указатель на строку в другой таблице.

Создание связей в БД представлено на рисунке 17.

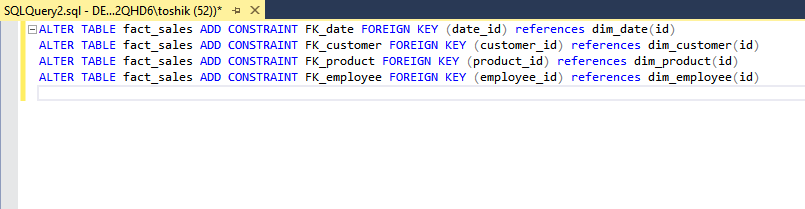


Рисунок 17 − Код создания связей в БД

После создания хранилища необходимо заполнить его данными. Загрузка данных в ХД может выполняться ручным и программным способом. При ручном способе выполняется скрипт языка T-SQL. В нашем случае было принято решение создать ETL-процесс на языке программирования Python, который будет обрабатывать входящий Excel файл. Этот файл содержит страницы соответствующие определенной таблице в базе данных, которые содержат конкретные экспортированные данные. Пример данных на рисунках 18 и 19.

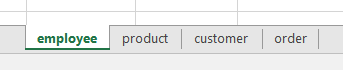


Рисунок 18 − Название страниц Excel файла

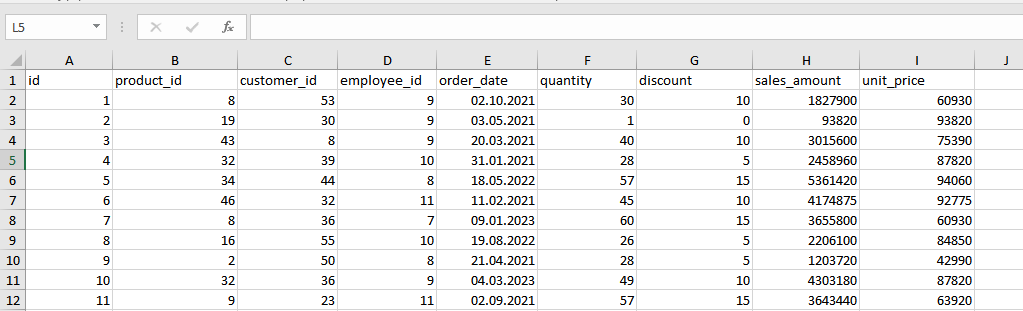


Рисунок 19 − Страница Excel файла "order"

ETL (Extract, Transform, Load) − это процесс, который используется для переноса данных из разных источников в одно центральное хранилище данных. ETL состоит из трех основных этапов:

* извлечение (Extract).

На этом этапе данные извлекаются из разных источников, таких как базы данных, файлы, API и т.д. Данные могут быть извлечены как полностью, так и частитчно, в зависимости от объема данных и доступности источника.

* преобразование (Transform).

На этом этапе данные преобразуются в нужный формат для загрузки в центральное хранилище данных. Преобразование может включать в себя очистку данных, объединение данных из разных источников, преобразование типов данных и т.д.

* загрузка (Load).

На этом этапе данные загружаются в центральное хранилище данных. Загрузка может быть выполнена как целиком, так и частями, в зависимости от объема данных и доступности хранилища данных.

Для реализации ETL-процесса используются специальные инструменты, такие как ETL-системы и библиотеки для языков программирования. Например, библиотека Pandas для языка Python в нашем случае, она предоставляет удобный интерфейс для работы с данными. Пример готовой таблицы представлен на рисунке 20.

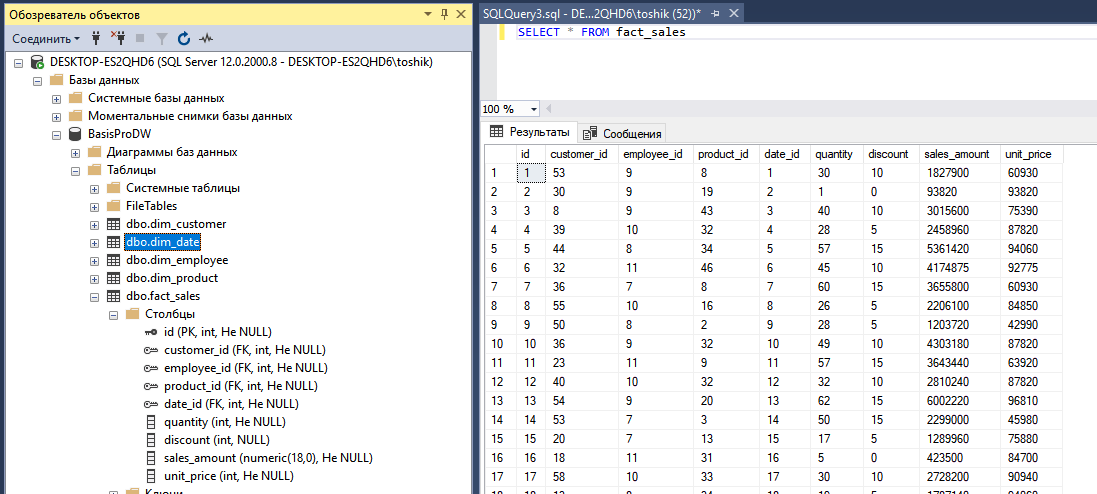


Рисунок 20 − Таблица "fact\_sales"

Кубы OLAP представляют собой, по сути, мета-отчеты. Разрезая мета-отчеты (кубы, то есть) по измерениям, аналитик получает, фактически, интересующие его "обычные" двумерные отчеты (это не обязательно отчеты в обычном понимании этого термина − речь идет о структурах данных с такими же функциями).

После создания базы данных хранилища, на основании загруженных данных был создан OLAP-куб, структура представлена на рисунке 21.

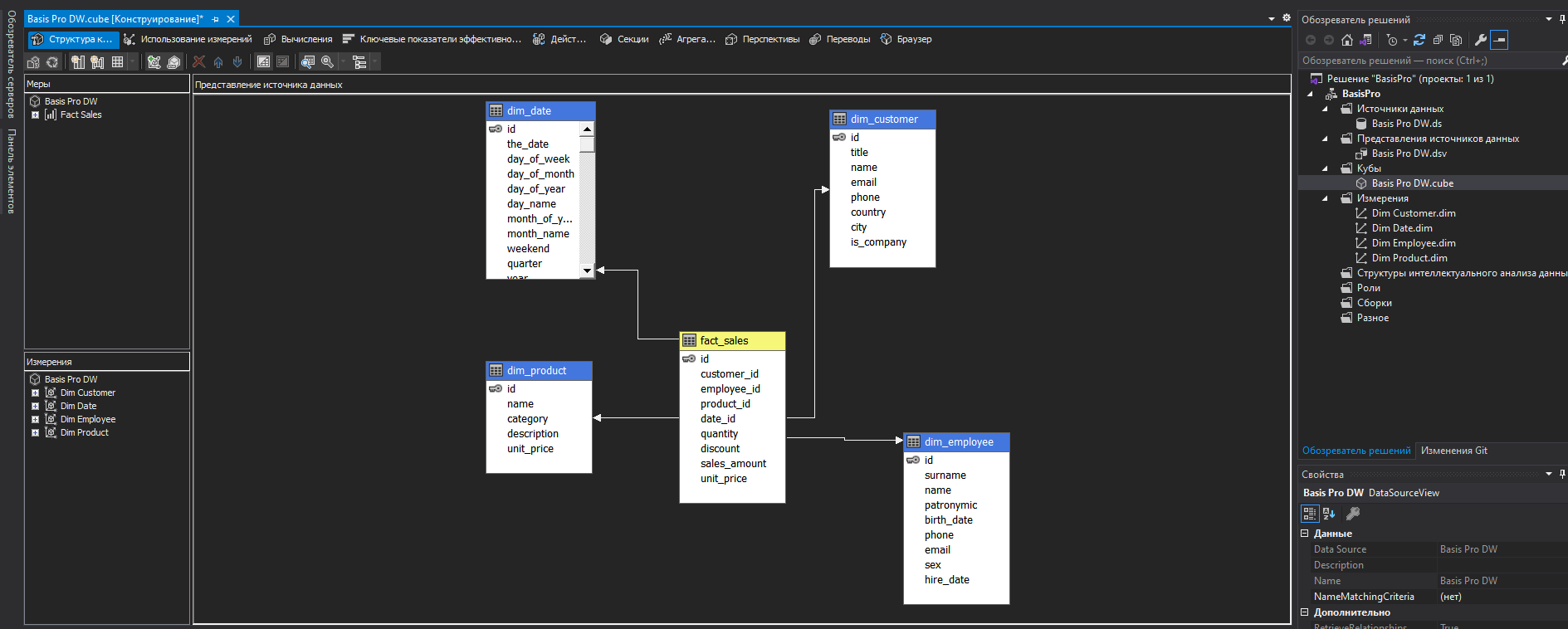


Рисунок 21 − Структура OLAP-куба

## 2.3 Тестирование программного обеспечения

Тестирование программного обеспечения (ПО) − это процесс проверки соответствия между фактическим и ожидаемым поведением программы и проводится по ограниченному набору тестов. Основная цель тестирования - выявить дефекты, ошибки и неисправности в программном продукте и убедиться, что программное обеспечение работает в соответствии с требованиями, ожиданиями и стандартами.

При тестировании программного обеспечения выполняются тестовые сценарии и анализируются результаты. Цель тестирования − выявить дефекты программного обеспечения и убедиться, что продукт работает правильно. Тестирование может быть автоматизированным или ручным.

Тестирование программного обеспечения может проводиться на различных уровнях, таких как модульное тестирование, интеграционное тестирование, системное тестирование и приемочное тестирование. Каждый уровень тестирования имеет свои характеристики и цели. Основными целями тестирования программного обеспечения являются:

* убедиться, что программный продукт соответствует требованиям и ожиданиям пользователей;
* проверка производительности программного продукта в различных средах;
* проверка безопасности программного продукта;
* проверка производительности программного продукта;
* проверка интерфейса программного продукта.

Тестирование программного обеспечения является важным этапом жизненного цикла разработки программного обеспечения. Оно выявляет дефекты и ошибки программного обеспечения на ранних стадиях процесса разработки, снижает затраты на исправление ошибок в будущем и повышает качество программного продукта.

Таблица 1 – Результаты тестирования

|  |  |  |
| --- | --- | --- |
| Название теста | Проверяемый случай | Результат теста |
|  | | |
| Проверка совместимости | Windows 10 | Совместима |
| Загрузка данных в хранилище | Загрузка данных с помощью ETL скрипта | Пройден |
| Ручная загрузка данных в хранилище | Загрузка данных с помощью SQL скрипта | Пройден |
| Проверка целостности данных хранилища | Проверка типов данных и корректности значений | Пройден |

Также было проведено тестирование SQL и Multidimensional Expressions (MDX) запросов к базе данных и OLAP кубу соответственно.

Существует объективная необходимость такого тестирования, в первую очередь проверки целостности данных, их достоверности, поскольку в бэкенде хранятся пользовательские данные, и данные необходимые для работы приложения.

SQL запрос на рисунке 22 показывает объем продаж товаров конкретным работником.

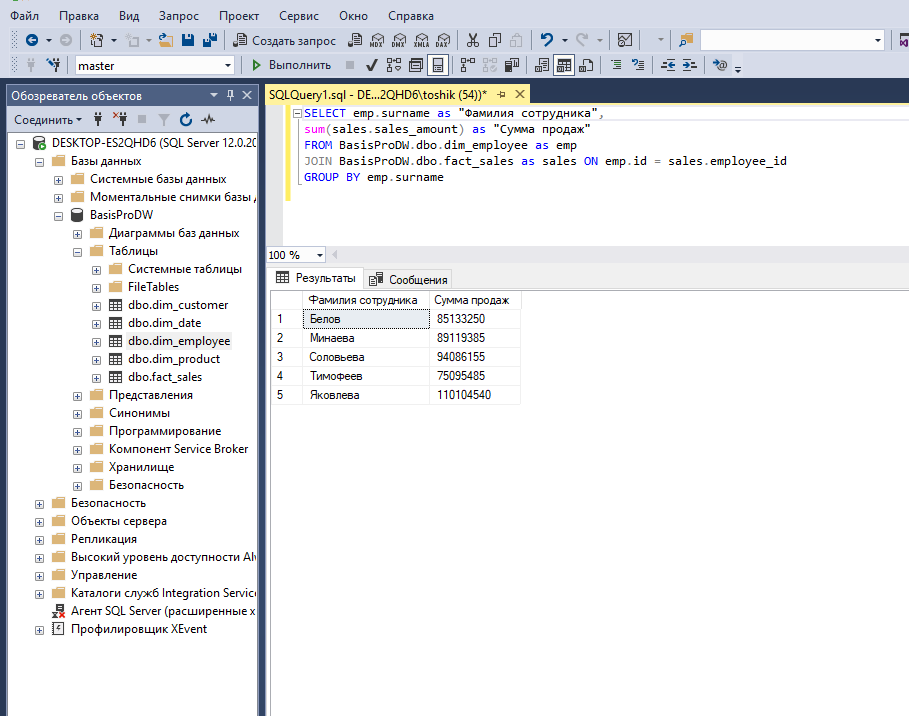


Рисунок 22 − SQL запрос к базе данных

При тестирование хранимых SQL-процедур необходимо проверить следующие параметры:

* имя хранимых процедур, имена параметров и их типы;
* вывод: сколько записей содержится в выводе, обрабатываются ли строки нулевой длины;
* функциональность процедуры, и ошибки функциональности;
* ввод данных в процедуру и проверка корректности вывода;
* параметры хранимых процедур: вызов процедуры с подачей невалидных данных. Негативное тестирование: запуск с невалидными параметрами;
* возвращаемые значения из процедуры, особенно в случае сбоя не должны возвращаться нулевые значения;
* проверка обработки ошибок: создание ошибки вручную, проверка сообщения об ошибке.

MDX запрос на рисунке 23 показывает объем продаж по месяцам 2022 года.

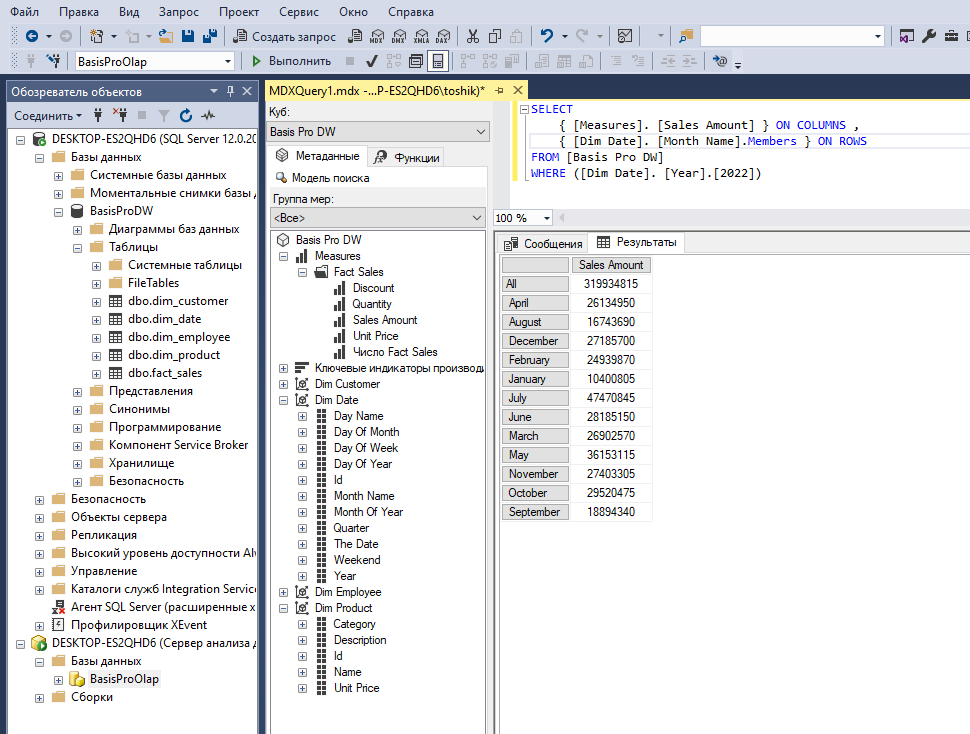


Рисунок 23 − MDX запрос к OLAP

С помощью SQL Server Reporting Services (SSRS) были создан отчет, представленный рисунке 24. Он показывает объём продаж разделенных по месяцам по конкретным категориям товаров.

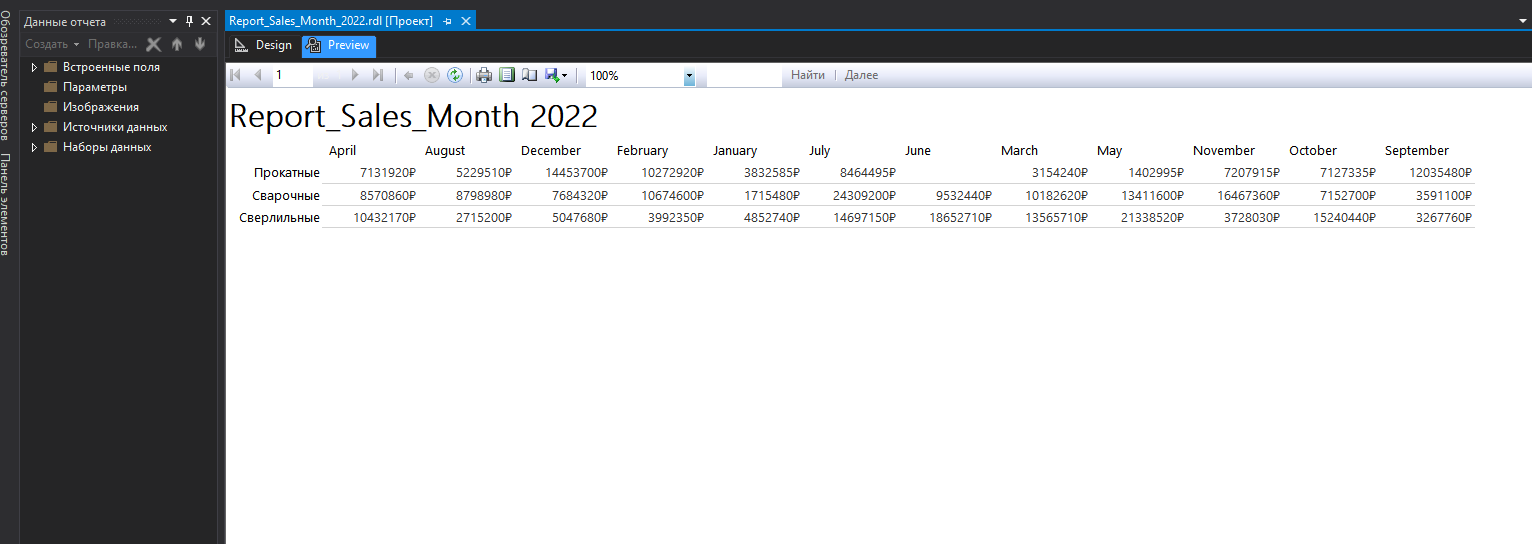


Рисунок 24 – Отчет Reporting Services

Дальнейшее взаимодействие с отчетами SSRS может осуществляться с помощью специального сервера, доступ к которому можно получить через браузер, что представлено на рисунке 25.

Даже то, что отчеты, реализуемые средствами SSRS, позволяют работать только с отчетами на специальном сервере, данное решение успешно выполняет функции необходимые для компании.

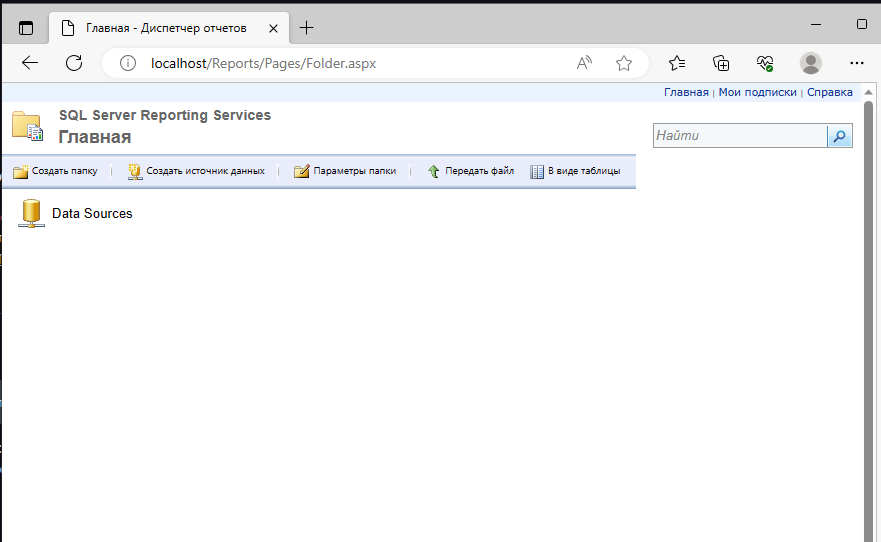


Рисунок 25 − Работа с отчетами через браузер

**3 Информационная безопасность**

Информационная безопасность − это комплекс мер, направленных на защиту информации от несанкционированного доступа, использования, раскрытия, модификации или уничтожения. Информационная безопасность включает в себя правовые, организационные и технические меры, направленные на обеспечение конфиденциальности, целостности и доступности информации.

Конфиденциальность относится к защите информации от несанкционированного доступа или раскрытия. Сюда входят меры по ограничению доступа к информации только авторизованных пользователей, шифрование данных и контроль доступа.

Целостность − это защита информации от несанкционированного изменения. Сюда входят такие меры, как контроль целостности данных, обнаружение и предотвращение несанкционированных изменений, и резервное копирование данных.

Доступность − это защита информации от несанкционированного перехвата или уничтожения. Она включает меры по предоставлению авторизованным пользователям доступа к информации, предотвращению отказа в обслуживании и резервному копированию данных [10].

Информационная безопасность − важный аспект современного мира, а информация - один из самых ценных ресурсов. Она включает в себя множество мер, таких как шифрование данных, контроль доступа, мониторинг безопасности и обучение пользователей. Информационная безопасность также должна рассматриваться на всех этапах жизненного цикла информационной системы, включая проектирование, разработку, эксплуатацию и утилизацию.

## 3.1 Основные угрозы баз данных

Основные угрозы информационной безопасности баз данных включают:

* внедрение кода: инъекционные недостатки, такие как SQL, NoSQL, возникают, когда ненадежные данные отправляются интерпретатору как часть команды или запроса;
* SQL-инъекции: это распространенный метод взлома веб-приложений и сайтов, работающих с реляционными базами данных. Атакующий может прочитать содержимое любых таблиц, удалить, изменить или добавить данные, а также есть вероятность получения возможности работы с локальными файлами и выполнения произвольных команд на атакуемом сервере;
* неограниченные привилегии базы данных: это угроза, когда пользователь получает доступ к базе данных с неограниченными правами, что может привести к краже, изменению или удалению данных;
* плохой аудиторский след: это угроза, когда не ведется достаточно полный и точный учет действий пользователей в базе данных, что может привести к несанкционированному доступу к данным;
* открытые резервные копии баз данных: это угроза, когда резервные копии баз данных не защищены должным образом, что может привести к краже или утечке конфиденциальной информации;
* нарушение целостности данных: это угроза, когда данные в базе данных изменяются или удаляются без разрешения, что может привести к потере или искажению данных;
* нарушение конфиденциальности данных: это угроза, когда конфиденциальная информация в базе данных становится доступной для несанкционированного использования или раскрытия;
* нарушение доступности данных: это угроза, когда данные в базе данных становятся недоступными для авторизованных пользователей, что может привести к проблемам в работе системы;
* нарушение безопасности сети: это угроза, когда злоумышленники получают доступ к базе данных через уязвимости в сетевой инфраструктуре.

## 3.2 Особенности защиты баз данных

Защита баз данных является важным аспектом информационной безопасности организации. Для этого необходимы правовые, организационные и технические меры, обеспечивающие защиту информации от несанкционированного доступа, уничтожения, модификации, блокирования, копирования, обмена, распространения и других незаконных действий, связанных с этой информацией [3,4].

Хранилище данных состоит из двух элементов: хранимых данных (собственно базы данных) и управляющего программного обеспечения (СУБД). Безопасность хранимой информации не может быть гарантирована без обеспечения безопасного управления данными; вопросы безопасности в СУБД можно разделить на две категории: зависимость от данных и независимость от данных.

Уязвимости, не зависящие от данных, могут возникнуть в любом программном обеспечении и могут быть вызваны, например, неправильно подобранным по времени обновлением программного обеспечения, неиспользуемыми функциями, недостаточной подготовкой администраторов программного обеспечения и т.д.

Большинство аспектов программного обеспечения полагаются на данные. Однако многие уязвимости могут косвенно опираться на данные. Например, большинство РСУБД поддерживают запросы к данным с использованием ряда языков запросов, включая набор функций, доступных пользователю, или произвольных функций на языке программирования [14].

**4 Безопасность жизнедеятельности**

Безопасность жизнедеятельности – это раздел науки, изучающий опасные и вредные производственные факторы, уровни техногенного воздействия на человека в процессе труда и разрабатывающий методы и средства повышения безопасности технологических процессов, основные направления снижения риска и последствий проявления опасных и вредных производственных факторов.

Здоровые условия труда программиста в значительной степени обеспечиваются техническим прогрессом, совершенствованием техники и технологии производства. Решение проблемы безопасности состоит в обеспечении нормальных условий деятельности людей, защите человека и окружающей его среды от воздействия вредных факторов.

Поддержание оптимальных условий деятельности и отдыха создает предпосылки наибольшей работоспособности и продуктивности.

Вредный производственный фактор − производственный фактор, воздействие которого на работающего, в определённых условиях, приводит к заболеванию или снижению работоспособности.

Опасный производственный фактор − производственный фактор, воздействие которого на работающего, в определённых условиях, приводит к травме или другому ухудшению здоровья.

На техника-программиста негативное действие могут оказывать следующие вредные и опасные факторы:

* плохой микроклимат помещения: повышенная и пониженная температура воздуха; чрезмерная запыленность и загазованность воздуха;
* повышенная и пониженная влажность воздуха;
* недостаточная освещенность рабочего места;
* превышающий допустимые нормы шум;
* повышенный уровень ионизирующего излучения;
* повышенный уровень статического электричества;
* опасность поражения электрическим током;
* нарушение эргономических норм при работе с компьютером и др.

## 4.1 Микроклимат

Микроклимат − комплекс физических факторов внутренней среды помещений, оказывающий влияние на тепловой обмен организма и здоровье человека. К параметрам микроклимата производственного помещения относится: температура воздуха (20-25 0С), скорость движения воздуха (0,2-0,3 м/с), относительная влажность (40-60 %) барометрическое давление (760 мм.рт.ст) и тепловое излучение от нагретых поверхностей.

Оптимальные микроклиматические условия характеризуются сочетанием параметров микроклимата, которые при длительном и систематическом воздействии на человека обеспечивают сохранения нормального функционального и теплового состояния организма без напряжения реакций терморегуляции. Они обеспечивают ощущение теплового комфорта и создают предпосылки для высокого уровня работоспособности.

Согласно СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы" представлены в таблице 2.

Таблица 2 − Допустимые и оптимальные величины показателей микроклимата на рабочих местах

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Период  года | Температура,  С | | Влажность,  % | | Скорость движ. воздуха, м/с | |
| Оптимальная | Допустимая | Оптимальная |  | Оптимальная | Допустимая |
| Холодный  (tср <=10oC) | 22-24 | 20 - 25 | 60-40 | Холодный  (tср <=10oC) | 22-24 | 20 – 25 |
| Теплый  (tср>10oC) | 23-25 | 21 - 28 | 60-40 | Теплый  (tср >10oC) | 23-25 | 21 – 28 |

## 4.2 Производственное освещение

Корректно спроектированное и исполненное производственное освещение улучшает работу с зрительным материалом, снижает уровень усталости, значительно повышает эффективность труда, а также благоприятно воздействует на уютность производственной среды и на психическое состояние работника, повышает уровень безопасности и снижает риски производственного травматизма.

Недостаточная яркость освещения приводит к напряжению зрения, ухудшает концентрацию и предрасполагает к ранней усталости.

Избыточно яркий свет может вызвать покраснение и разбухание глаз, раздражение и облупливание кожи вокруг глаз. Проектирование неуместной направленности света на рабочем месте может создать резкие тени, причины дезориентации работника. Все эти факторы могут спровоцировать несчастные случаи или профзаболевания, поэтому необходим оптимальный расчет освещения.

Освещение рабочего места должно удовлетворять нормам, представленным в СанПин 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». Требования по освещенности, предъявляемые к помещениям, оборудованным ПЭВМ:

* помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение.
* окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.;
* рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева;
* рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева;
* для внутренней отделки интерьера помещений, где расположены ПЭВМ, должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка - 0,7 - 0,8; для стен - 0,5 - 0,6; для пола - 0,3 - 0,5.
* искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения;
* освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать бликов на поверхности экрана.
* следует ограничивать прямую блёскость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м2;
* следует ограничивать отраженную блёскость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м2 и яркость потолка не должна превышать 200 кд/м2;
* показатель ослепленности для источников общего искусственного освещения в производственных помещениях должен быть не более 20. Показатель дискомфорта в административно-общественных помещениях не более 40;
* яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м2, защитный угол светильников должен быть не менее 40 градусов;
* светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов;
* следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3: 1 - 5: 1, а между рабочими поверхностями и поверхностями стен и оборудования 10: 1;
* для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами (ЭПРА).
* в качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных;
* общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов, коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1,4;
* коэффициент пульсации не должен превышать 5%;
* для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших лампах.

## 4.3 Шум и вибрация

Шум в помещении создается внутренними источниками:

* вентилятор охлаждения блока питания;
* жесткий диск;
* вентилятор охлаждения процессора;
* люминесцентные лампы;

В помещение тип шумов можно классифицировать как:

* широкополосный (по спектральному составу);
* непостоянный (по временным характеристикам).

Допустимые уровни звукового давления и уровней звука на рабочем месте для помещения вычислительного центра согласно СН 2.2.4/2.1.8.562-96 "Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки" приведены в таблице 4.2. Допустимый уровень звукового давления и уровень звука определяется в зависимости от характеристики шума и назначения помещения.

Предельно допустимые величины нормируемых параметров вибрации рабочих мест при длительности вибрационного воздействия 480 мин (8 ч) приведены в таблице 3.

Таблица 3 − Допустимые значения уровней звукового давления в октавных полосах частот

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вид трудовой деятельности | Уровни звукового давления (дБ) в октавных полосах со среднегеометрическими частотами, Гц | | | | | | | | | Уровень звука в дБА |
| 31,5 | 63 | 125 | 250 | 500 | 1000 | 2000 | 4000 | 8000 |
| Эксплуатация ПЭВМ | 86 | 50 | 61 | 54 | 49 | 45 | 42 | 40 | 38 | 50 |

Таблица 4 − Предельно допустимые величины нормируемых параметров вибрации рабочих мест

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Среднегеометрические частоты полос, ГЦ | 1,0 | 2,0 | 4,0 | 8,0 | 16,0 | 31,5 | 63,0 | Корректированные и эквивалентные корректированные значения и их уровни |
| Предельно допустимые значения виброскорости, дБ |  | 91 | 82 | 76 | 75 | 75 | 75 | 75 |

## 4.4 Безопасность технологических процессов и оборудования

Требования безопасности перед началом работы:

* привести в порядок одежду;
* осмотреть рабочее место, убрать все мешающие работе предметы;
* протереть рабочую поверхность клавиатуры, очистить экран;
* визуально проверить правильность подключения ПЭВМ к электросети;

Требования безопасности во время работы:

* убедиться в работоспособности ПЭВМ после выключения питающего напряжения электросети;
* запрещается работать на оборудовании со снятыми кожухами и крышками;
* запрещается трогать кабели и провода, соединяющие блоки ПЭВМ, перемещать устройства, находящиеся под напряжением;
* не оставлять без присмотра включенные ПЭВМ и отдельные устройства;
* запрещается производить самостоятельно любые виды ремонта и устранение неисправностей;
* не устанавливать неизвестные системы паролирования и самостоятельно проводить переформатирование диска;
* во время регламентированных перерывов с целью снижения нервно-эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинеза, предотвращения развития нозвотонического утомления целесообразно выполнять комплексы упражнений.

Требования безопасности в аварийных ситуациях:

* при нарушении работы ПЭВМ, перегорании предохранителей и т.п. аппаратура должна быть немедленно отключена;
* при временном отключении электроэнергии тумблера электропитания должны быть выключены;
* при появлении запаха гари, дыма в помещении или на рабочем месте сеть электропитания ПЭВМ и других устройств должна быть выключена и приняты меры к обнаружению источника загорания и тушению первичными средствами пожаротушения. Тушение загорания оборудования, находящегося под напряжением, производить только углекислотными или порошковыми огнетушителями. При работе с углекислотными огнетушителями не следует браться руками за раструб (температура до - 80 град. С);
* при обнаружении пожара или признаков возгорания немедленно сообщить об этом в пожарную охрану или привести в действие ручной извещатель пожарной сигнализации, а затем действовать в соответствии с планом эвакуации.

Требования безопасности по окончании работы:

* отключить ПЭВМ от сети;
* привести в порядок рабочее место;
* при сменной работе передать рабочее место в рабочем состоянии по смене, сделать запись в журнале учета работ и передачи смены. Если дальнейшей работы не будет, сдать рабочее место старшему по смене или ответственному за помещение.

## 4.5 Отдых глаз

Для программиста техника на производстве важно соблюдать правила отдыха глаз, чтобы избежать напряжения зрения и усталости глаз. Для этого можно использовать следующие рекомендации:

* регулярно делать перерывы в работе и отдыхать глаза, например, каждые 20-30 минут смотреть вдаль на 20-30 секунд;
* избегать резкого перехода зрения с яркого экрана на темный объект и наоборот;
* поддерживать правильное освещение на рабочем месте, чтобы избежать напряжения глаз и усталости;
* обеспечить комфортный микроклимат в помещении, чтобы избежать пересыхания слизистой оболочки глаз и усталости.

Соблюдение этих рекомендаций поможет сохранить здоровье глаз и повысить эффективность работы программиста техника на производстве.

# Заключение

В процессе выполнения дипломной работы была разработана информационно-аналитическая система, представляющая собой хранилище данных, для ООО «Базис Про», г. Ростов-на-Дону. Были собрана и проанализирована информации о компании, с учетом проведенного анализа предметной области были выявлены основные задачи, которые успешно реализованы в данном проекте.

Хранилище данных, лежащее в основе системы, обеспечивает:

* ускорение выполнение запросов;
* повышение качества и целостности данных;
* хранение исторических данных.

Разработанная информационная система − это эффективное средство обработки и анализа данных, она охватывает основные части бизнес процесса. Например, использование системы позволит оценить динамику продаж за разные временные периоды для различных категорий товаров, отдельных товаров.

В качестве дальнейшего развития системы, можно расширять хранилище данных по мере необходимости отчетов, внедрение оркестратора для автоматизации загрузки и обработки данных в хранилище.

Разработанное обеспечение отвечает требованиям заказчика и готово к использованию.

# Перечень использованных информационных ресурсов

1. Закон №152 ФЗ Федеральный закон от 28.07.2006, №152, (ФЗ о персональных данных)//URL: rg.ru, дата поступления 27.07.2006, опубликован 29.07.2006, вступает в силу 26.01.2007; СПС «КонсультантПлюс»). См.: Федеральный закон от 26.07.2006 г.
2. Законодательные материалы Российская Федерация. Законы. «О товарных знаках, знаках обслуживания и наименованиях мест происхождения товаров» // Собрание законодательства РФ. - 23.09.1992 - № 3520-1.
3. Законодательные материалы Российская Федерация. Законы. «О правовой охране программ для электронных вычислительных машин и баз данных». // Собрание законодательства РФ. - 23.09.1992 - № 3523-1.
4. Гражданский кодекс Российской Федерации (часть четвертая) от 24.11.2006 № 318-ФЗ (ред. от 11.06.2022) [Электронный ресурс] // Справочная правовая система «Консультант плюс»
5. Беспалов, Д. А. Администрирование баз данных и компьютерных сетей : учебное пособие / А. И. Костюк, Д. А. Беспалов ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2020. - 127 с. - ISBN 978-5-9275-3577-4. - Текст: электронный. - URL: https://znanium.com/catalog/product/1308403 (дата обращения: 13.05.2023).
6. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 2-е изд., перераб. и доп. — Москва: ФОРУМ: ИНФРА-М, 2021. — 416 с. — (Cреднее профессиональное образование). - ISBN 978-5-91134-655-3. - Текст : электронный. - URL: https://znanium.com/catalog/product/1190668 (дата обращения: 25.04.2023).
7. Маккинни, У. Python и анализ данных : практическое пособие / У. Маккинни ; пер. с анг. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2020. - 540 с. - ISBN 978-5-97060-590-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/2012523 (дата обращения: 13.03.2023).
8. Новиков, Б. А. Основы технологий баз данных : учебное пособие / Б. А. Новиков, Е. А. Горшкова, Н. Г. Графеева ; под ред. Е. В. Рогова. — 2-е изд. — Москва : ДМК Пресс, 2020. - 582 с. - ISBN 978-5-97060-841-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/1210665 (дата обращения: 14.05.2023).
9. Шитов, В. Н. Основы проектирования баз данных : учебное пособие / В.Н. Шитов. — Москва: ИНФРА-М, 2023. — 236 с. — (Среднее профессиональное образование). — DOI 10.12737/1855782. - ISBN 978-5-16-017461-7. - Текст: электронный. - URL: https://znanium.com/catalog/product/1855782 (дата обращения: 10.04.2023).
10. Сидорова, Н. П. Информационное обеспечение и базы данных : практикум по дисциплине «Информационное обеспечение, базы данных» / Н. П. Сидорова, Г. Н. Исаева, Ю. Ю, Сидоров. - Москва : Директ-Медиа, 2019. - 84 с. - ISBN 978-5-4475-9996-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1985726 (дата обращения: 30.04.2023).
11. Митин, А. И. Работа с базами данных Microsoft SQL Server: сценарии практических занятий / А. И. Митин. - Москва : Директ-Медиа, 2020. - 142 с. - ISBN 978-5-4499-0420-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/1985743 (дата обращения: 11.05.2023).
12. Титов, А. Н. Обработка данных в Python. Основы работы с библиотекой Pandas : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тазиева ; Минобрнауки России, Казан. нац. исслед. технол. ун-т. - Казань : Изд-во КНИТУ, 2022. - 116 с. - ISBN 978-5-7882-3164-8. - Текст : электронный. - URL: https://znanium.com/catalog/product/2069262 (дата обращения: 09.05.2023).
13. Титов, А. Н. Python. Обработка данных : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тазиева ; Минобрнауки России, Казан. нац. исслед. технол. ун-т. - Казань : Изд-во КНИТУ, 2022. - 104 с. - ISBN 978-5-7882-3171-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/2069264 (дата обращения: 16.04.2023)
14. Полищук, Ю. В. Базы данных и их безопасность : учебное пособие / Ю.В. Полищук, А.С. Боровский. — Москва : ИНФРА-М, 2023. — 210 с. — (Высшее образование: Специалитет). — DOI 10.12737/1011088. - ISBN 978-5-16-014924-0. - Текст : электронный. - URL: https://znanium.com/catalog/product/1905717 (дата обращения: 06.04.2023)
15. Общие сведения о схеме типа "звезда" и ее значении в Power BI // Microsoft URL: https://learn.microsoft.com/ru-ru/power-bi/guidance/star-schema (дата обращения: 14.05.2023).
16. Что такое аналитическая обработка данных, как она помогает строить прогнозы и принимать бизнес-решения // Mail.ru URL: https://mcs.mail.ru/blog/chto-takoe-analiticheskaya-obrabotka-dannykh (дата обращения: 14.05.2023).
17. Кондрашов, Ю. Н. Эффективное использование СУБД MS SQL Server: учебное пособие / Ю. Н. Кондрашов. — Москва : Финансовый университет, 2021. — 134 с. — ISBN 978-5-7942-1413-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/265970 (дата обращения: 14.01.2023)
18. Турманов, В. SQL для хранения, обработки и анализа данных : учебное пособие / В. Турманов, Б. Гайфуллин. — Москва : СОЛОН-Пресс, 2021. — 172 с. — ISBN 978-5-91359-463-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/322250 (дата обращения: 14.03.2023).
19. Никитина, Т. П. Программирование. Основы Python / Т. П. Никитина, Л. В. Королев. — Санкт-Петербург: Лань, 2023. — 156 с. — ISBN 978-5-507-45283-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/302714 (дата обращения: 20.02.2023).

**Приложение**

**(обязательное)**

**Листинг программы**

Листинг А.1 − Программный код базы данных

CREATE DATABASE BasisProDW

go

ALTER DATABASE BasisProDw set recovery simple

go

ALTER DATABASE BasisProDw set auto\_shrink off

go

ALTER DATABASE BasisProDw set auto\_create\_statistics on

go

ALTER DATABASE BasisProDw set auto\_update\_statistics on

go

CREATE TABLE dim\_date(

id INT PRIMARY KEY NOT NULL,

the\_date DATE NOT NULL,

day\_of\_week INT NOT NULL,

day\_of\_month INT NOT NULL,

day\_of\_year INT NOT NULL,

day\_name VARCHAR(20) NOT NULL,

month\_of\_year INT NOT NULL,

month\_name VARCHAR(20) NOT NULL,

weekend INT NOT NULL,

quarter INT NOT NULL,

year INT NOT NULL

)

CREATE TABLE dim\_employee

(

id INT PRIMARY KEY,

surname CHARACTER VARYING(100) NOT NULL,

name CHARACTER VARYING(100) NOT NULL,

patronymic CHARACTER VARYING(100) NOT NULL,

birth\_date DATE NOT NULL,

phone CHARACTER (11) NOT NULL,

email CHARACTER VARYING (100) NOT NULL,

sex VARCHAR(1) NOT NULL,

hire\_date DATE NOT NULL

);

CREATE TABLE dim\_customer (

id INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) NOT NULL,

phone CHARACTER (11) NOT NULL,

country VARCHAR(30) NOT NULL,

city VARCHAR(30) NOT NULL,

is\_company INT NOT NULL

);

CREATE TABLE dim\_product (

id SERIAL PRIMARY KEY,

name VARCHAR(100) NOT NULL,

category VARCHAR(100) NOT NULL,

description VARCHAR(8000) NOT NULL,

unit\_price NUMERIC NOT NULL

);

CREATE TABLE fact\_sales (

id INT PRIMARY KEY,

customer\_id INT NOT NULL,

employee\_id INT NOT NULL,

product\_id INT NOT NULL,

date\_id INT NOT NULL,

quantity INT NOT NULL,

discount INT,

sales\_amount NUMERIC NOT NULL,

unit\_price INT NOT NULL

);

ALTER TABLE fact\_sales ADD CONSTRAINT FK\_date FOREIGN KEY (date\_id) references dim\_date(id)

ALTER TABLE fact\_sales ADD CONSTRAINT FK\_customer FOREIGN KEY (customer\_id) references dim\_customer(id)

ALTER TABLE fact\_sales ADD CONSTRAINT FK\_product FOREIGN KEY (product\_id) references dim\_product(id)

ALTER TABLE fact\_sales ADD CONSTRAINT FK\_employee FOREIGN KEY (employee\_id) references dim\_employee(id)

Листинг А.2 − Программный код ETL процесса

import pyodbc  
import sqlalchemy  
import pandas as pd  
import urllib  
from datetime import datetime as dt  
  
# импорт библиотек  
  
class MissingValue(Exception):  
 pass  
  
# класс для создания ошибки пропущенных значений в датафрейме  
  
  
class Sql:  
 def \_\_init\_\_(self, database, server="DESKTOP-ES2QHD6"):  
 self.cnxn = pyodbc.connect("Driver={SQL Server Native Client 11.0};"  
 "Server="+server+";"  
 "Database="+database+";"  
 "Trusted\_Connection=yes;")  
 self.conn = "-- {} -- successfully".format(datetime.now()  
 .strftime("%d/%m/%Y"))  
 def query(self, req):  
 cursor = sql.cnxn.execute(req)  
 row = cursor.fetchone()  
 while row:  
 print(row[0])  
 row = cursor.fetchone()  
  
# создание класса для подключения к бд  
# и совершения запросов к ней  
  
  
def extract\_data(file\_path="data.xlsx"):  
 print("---Начало процесса extract---")  
 try:  
 df\_employee = pd.read\_excel(file\_path, sheet\_name='employee')  
 print("Успешно загружен лист - employee")  
  
 except ValueError:  
 print("Лист employee не найден")  
 raise  
  
 except:  
 print("Что-то пошло не так с листом employee")  
 raise  
  
 try:  
 df\_customer = pd.read\_excel(file\_path, sheet\_name='customer')  
 print("Успешно загружен лист - customer")  
  
 except ValueError:  
 print("Лист customer не найден")  
 raise  
  
 except:  
 print("Что-то пошло не так с листом customer")  
 raise  
  
 try:  
 df\_product = pd.read\_excel(file\_path, sheet\_name='product')  
 print("Успешно загружен лист - product")  
  
 except ValueError:  
 print("Лист product не найден")  
 raise  
  
 except:  
 print("Что-то пошло не так с листом product")  
 raise  
  
 try:  
 df\_order = pd.read\_excel(file\_path, sheet\_name='order')  
 print("Успешно загружен лист - order")  
  
 except ValueError:  
 print("Лист order не найден")  
 raise  
  
 except:  
 print("Что-то пошло не так с листом order")  
 raise  
  
 print('---Конец процесса extract---')  
  
  
 return df\_employee, df\_customer, df\_order, df\_product  
  
# загрузка данных из excel файла  
def transform\_data(df\_employee, df\_customer, df\_product, df\_order):  
 print('---Начало процесса transform---')  
  
 if sum(list(df\_employee.isnull().sum())) == 0:  
 print('Отсутствуют пустые значения - df\_employee')  
 else:  
 raise MissingValue('Пустые значения в df\_employee')  
 if sum(list(df\_customer.isnull().sum())) == 0:  
 print('Отсутствуют пустые значения - df\_customer')  
 else:  
 raise MissingValue('Пустые значения в df\_customer')  
 if sum(list(df\_product.isnull().sum())) == 0:  
 print('Отсутствуют пустые значения - df\_product')  
 else:  
 raise MissingValue('Пустые значения в df\_product')  
 if sum(list(df\_order.isnull().sum())) == 0:  
 print('Отсутствуют пустые значения - df\_order')  
 else:  
 raise MissingValue('Пустые значения в df\_order')  
 # проверка столбцов на пустые значения  
  
  
 df\_order.order\_date = df\_order.order\_date.apply(lambda x: dt.date(x))  
 df\_employee.hire\_date = df\_employee.hire\_date.apply(lambda x: dt.date(x))  
  
 # перевод типов столбцов из timestamp в datetime  
 df\_date = pd.DataFrame(columns=['id', 'the\_date', 'day\_of\_week', 'day\_of\_month', 'day\_of\_year', 'day\_name',  
 'month\_of\_year', 'month\_name', 'weekend', 'quarter', 'year'])  
  
  
 # создание таблицы для дат заказов  
  
  
 def is\_weekend(n):  
 if (int(n) == 5) or (int(n) == 6):  
 return 1  
 return 0  
 # функция, которая проверяет является ли день субботой или воскресеньем  
 def cal\_quarter(n):  
 if int(n) in [1, 2, 3]:  
 return 1  
 elif int(n) in [4, 5, 6]:  
 return 2  
 elif int(n) in [7, 8, 9]:  
 return 3  
 elif int(n) in [10, 11, 12]:  
 return 4  
  
 # функция, которая определяет квартал которому принадлежит дата  
 try:  
 cursor = sql.cnxn.execute("SELECT id FROM dim\_date ORDER BY id DESC")  
 row = cursor.fetchone()  
 num\_id = row[0]  
 except:  
 num\_id = 0  
  
 for n in range(len(df\_order)):  
 s = df\_order.order\_date[n].strftime('%w, %d, %j, %A, %m, %B, %Y')  
 s = s.split(',')  
 weekend = is\_weekend(s[0])  
 quarter = cal\_quarter(s[4])  
 df\_date = df\_date.append(  
 {'id': num\_id + 1, 'the\_date': df\_order.order\_date[n], 'day\_of\_week': int(s[0]) + 1, 'day\_of\_month': int(s[1]),  
 'day\_of\_year': int(s[2]), 'day\_name': s[3], 'month\_of\_year': int(s[4]),  
 'month\_name': s[5], 'weekend': weekend, 'quarter': quarter, 'year': int(s[6])}, ignore\_index=True)  
 num\_id += 1  
 df\_order.order\_date[n] = n + 1  
 df\_order = df\_order.rename(columns={'order\_date': 'date\_id'})  
  
 # запись данных в новый датафрейм df\_date  
  
 print('---Конец процесса transform---')  
 return df\_employee, df\_customer, df\_order, df\_product, df\_date  
  
# обработка данных  
def load\_data(df\_employee, df\_customer, df\_order, df\_product, df\_date):  
 print('---Начало процесса load---')  
 quoted = urllib.parse.quote\_plus("DRIVER={SQL Server};SERVER=DESKTOP-ES2QHD6;DATABASE=BasisProDW")  
 engine = sqlalchemy.create\_engine('mssql+pyodbc:///?odbc\_connect={}'.format(quoted))  
 # начало загрузки датафрейма  
 with engine.connect() as conn:  
 transaction = conn.begin()  
 try:  
 df\_employee.to\_sql("dim\_employee", con=engine, if\_exists='append', index=False)  
 transaction.commit()  
 print('Успешная загрузка датафрейма - df\_employee')  
  
 except Exception as E:  
 print(E)  
 transaction.rollback()  
 raise  
 # конец загрузки датафрейма  
 # данная конструкция создана для добавления данных транкзакциями  
 # для обеспечения целостности данных  
  
 with engine.connect() as conn:  
 transaction = conn.begin()  
 try:  
 df\_product.to\_sql("dim\_product", con=engine, if\_exists='append', index=False)  
 transaction.commit()  
 print('Успешная загрузка датафрейма - df\_product')  
  
 except Exception as E:  
 print(E)  
 transaction.rollback()  
 raise  
  
 with engine.connect() as conn:  
 transaction = conn.begin()  
 try:  
 df\_customer.to\_sql("dim\_customer", con=engine, if\_exists='append', index=False)  
 transaction.commit()  
 print('Успешная загрузка датафрейма - df\_customer')  
  
 except Exception as E:  
 print(E)  
 transaction.rollback()  
 raise  
  
 with engine.connect() as conn:  
 transaction = conn.begin()  
 try:  
 df\_order.to\_sql("fact\_sales", con=engine, if\_exists='append', index=False)  
 transaction.commit()  
 print('Успешная загрузка датафрейма - df\_order')  
  
 except Exception as E:  
 print(E)  
 transaction.rollback()  
 raise  
 with engine.connect() as conn:  
 transaction = conn.begin()  
 try:  
 df\_date.to\_sql("dim\_date", con=engine, if\_exists='append', index=False)  
 transaction.commit()  
 print('Успешная загрузка датафрейма - df\_date')  
  
 except Exception as E:  
 print(E)  
 transaction.rollback()  
 raise  
 print('---Конец процесса load---')  
# загрузка данных в хранилище  
if \_\_name\_\_ == "\_\_main\_\_":  
 sql = Sql("BasisProDW")  
 sql.conn  
 df\_employee, df\_customer, df\_order, df\_product = extract\_data()  
 df\_employee, df\_customer, df\_order, df\_product, df\_date= transform\_data(df\_employee, df\_customer, df\_product, df\_order)  
 load\_data(df\_employee, df\_customer, df\_order, df\_product, df\_date)