Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Технологии разработки программного обеспечения

ОТЧЁТ

по лабораторной работе

на тему

Техническое задание проекта

Выполнил

Студент гр. 053502

Макаро М. В.

Проверил

Ассистент кафедры информатики

Гриценко Н.Ю.

Минск 2023

**содержание**

[1 Цель работы 3](#_Toc147164376)

[2 Техническое задание проекта 4](#_Toc147164377)

[2.1 Общие положения 4](#_Toc147164378)

[2.2 Функциональная карта системы 6](#_Toc147164384)

[2.3 Модель данных 10](#_Toc147164385)

[2.4 Интерфейсы пользователя 10](#_Toc147164386)

[2.5 Нефункциональные требования 15](#_Toc147164387)

ВВЕДЕНИЕ

Статистический центр – это организация, которая собирает, анализирует и публикует статистическую информацию. Это может быть информация о экономике, населении, социальном развитии и других аспектах жизни страны или региона. Статистические центры могут быть государственными или частными. Они играют важную роль в формировании экономической и социальной политики государства.

Статистические центры играют важную роль в современном мире, поскольку они предоставляют информацию о различных аспектах жизни общества. Эта информация может использоваться для принятия решений на разных уровнях – от местного до международного.

Например, статистические данные о населении могут помочь правительству планировать инфраструктуру, образование и здравоохранение. Данные о экономике могут помочь бизнесу принимать решения о развитии и инвестициях.

Кроме того, статистические центры помогают отслеживать изменения в обществе и предсказывать будущие тенденции. Это позволяет правительствам и организациям адаптироваться к новым условиям и улучшать свою работу.

Таким образом, статистические центры являются важным инструментом для анализа и понимания данных, которые помогают принимать обоснованные решения и улучшать жизнь людей.

Целью данного курсового проекта является разработка базы данных для статистического центра.

В соответствии с поставленной целью были определены следующие задачи:

– изучить теоретические материалы по выбранной предметной области;

– выделить основные потоки данных, протекающих во время исполнения процессов выбранной предметной области;

– выделить основные сущности, с помощью которых описываются имеющиеся объекты предметной области;

– определить отношения между выделенными сущностями;

– разработать базу данных для статистического центра.

1. обзор существующих аналогов

В мире существует огромное количество статистических центров и почти у всех из них свои базы данных. В современном информационном обществе роль статистических центров становится более актуальной и важной в контексте сбора, анализа и предоставления статистической информации для эффективного управления и развития общества. Почти все страны мира уделяют значительное внимание созданию и развитию государственных статистических центров, которые являются основными источниками объективной и достоверной статистической информации.

Помимо государственных институтов, существуют также коммерческие статистические центры, предоставляющие услуги сбора и анализа данных для бизнес-сектора. Эти организации могут предложить более гибкие и индивидуальные подходы к сбору данных, а также специализированные услуги, направленные на конкретные потребности клиентов.

* 1. Национальный статистический комитет Республики Беларусь

Национальный статистический комитет Республики Беларусь (Белстат) является государственным органом, ответственным за сбор, обработку и распространение статистической информации о социально-экономическом положении страны. Белстат занимается сбором данных от предприятий, организаций и домашних хозяйств, а затем анализирует и публикует полученную информацию. Основными направлениями деятельности Белстата являются экономика, демография, социальная сфера, наука и инновации, рынок труда, уровень жизни населения и др.

Двумя основными функциями Белстата являются: формирование и представление официальной статистической информации и обеспечение ведения баз данных официальной статистической информации. У Белстата разработаны собственные базы и банки данных для хранения официальной статистической информации. А так же собственные программный продукты для формирования и представления официальной статистики.

Вместе базы данных и программные продукты для представления статистической информации формируют интерактивную информационно-аналитическую систему баз данных (ИАС БД) для распространения официальной статистической информации. Она содержит статистическую информацию в динамике за ряд лет по более чем 700 показателям, характеризующим экономическое, демографическое, социальное и экологическое положение в Республике Беларусь. Например, такие показатели как:

– число населенных пунктов на начало периода;

– число родившихся на 1000 человек населения;

– число умерших детей до 1 года на 1000 родившихся;

– валовой внутренний продукт в текущих ценах;

– оплата труда работников в текущих ценах;

– расход топливно-энергетических ресурсов организациями.

Система предоставит Вам возможность самостоятельно формировать запрос, оперативно извлекать из базы данных нужную информацию, выбирать способ ее представления (таблица, график, картограмма), получать официальную статистическую информацию из международных источников, а также сравнить показатели, характеризующие социально-экономическое развитие Республики Беларусь, с показателями других стран мира. На рисунке 1.1 представлена одна из таблиц полученных с помощью информационно-аналитической системы баз данных.

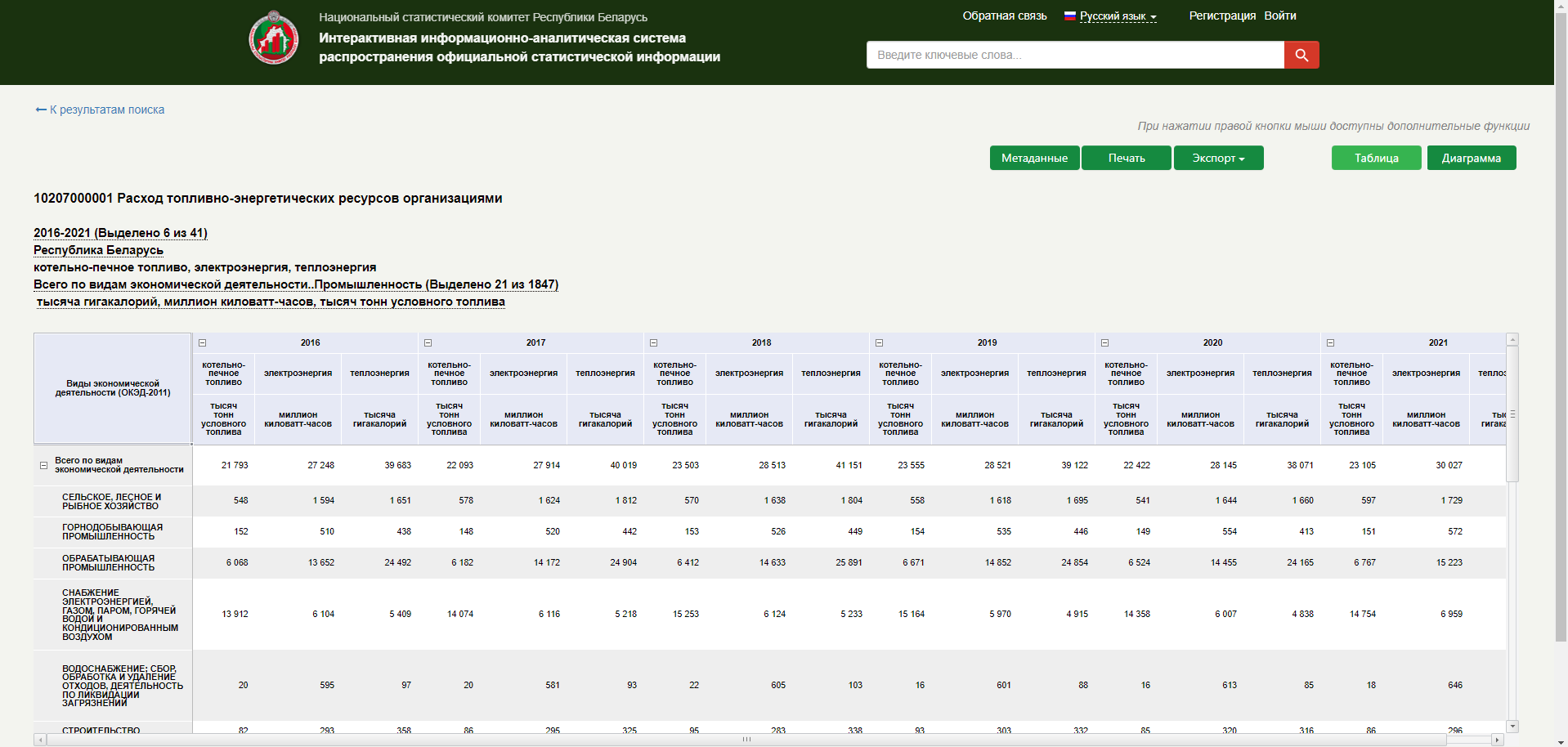


Рисунок 1.1 – Интерфейс ИАС БД

Таким образом можно выделить следующие плюсы ИАС БД:

– объективность и надежность данных;

– широкий спектр данных;

– систематичность и регулярность публикаций;

– поддержание стабильности.

Также наблюдаются следующие минусы:

– недостаток независимости;

– отсутствие свободы информации;

– отсутствие полной синхронизации с мировыми стандартами.

* 1. Федеральная служба государственной статистики

Федеральная служба государственной статистики (Росстат) является федеральным органом исполнительной власти, осуществляющим функции по формированию официальной статистической информации о социальных, экономических, демографических, экологических и других общественных процессах в Российской Федерации, а также в порядке и случаях, установленных законодательством Российской Федерации по контролю в сфере официального статистического учета [1].

Росстат предоставляет Витрину статистических данных (ВСД) – это интерактивный ресурс, который предоставляет пользователям удобный доступ к различным статистическим данным. Витрина статистических данных Росстата позволяет осуществлять следующие действия:

1 Искать и фильтровать данные. Пользователи могут осуществлять поиск и фильтрацию данных в соответствии с интересующими их параметрами.

2 Визуализировать данные. Витрина статистических данных предоставляет инструменты для визуализации данных. Это может включать в себя графики, диаграммы, карты и другие визуальные элементы, делая статистическую информацию более понятной и доступной.

3. Пользователи могут сравнивать различные статистические показатели между разными регионами, периодами времени и другими параметрами. Это полезно для выявления тенденций и анализа изменений во времени.

4 Экспортировать данные. Витрина предоставляет возможность экспорта данных в различные форматы, такие как таблицы, графики или файлы для дальнейшего использования в сторонних программных средствах или для подготовки отчетов.

5. Использовать интерактивные карты. Пользователи могут исследовать статистику по регионам, используя картографические средства.

На рисунке 1.2 представлена одна из таблиц полученных с помощью ВСД.

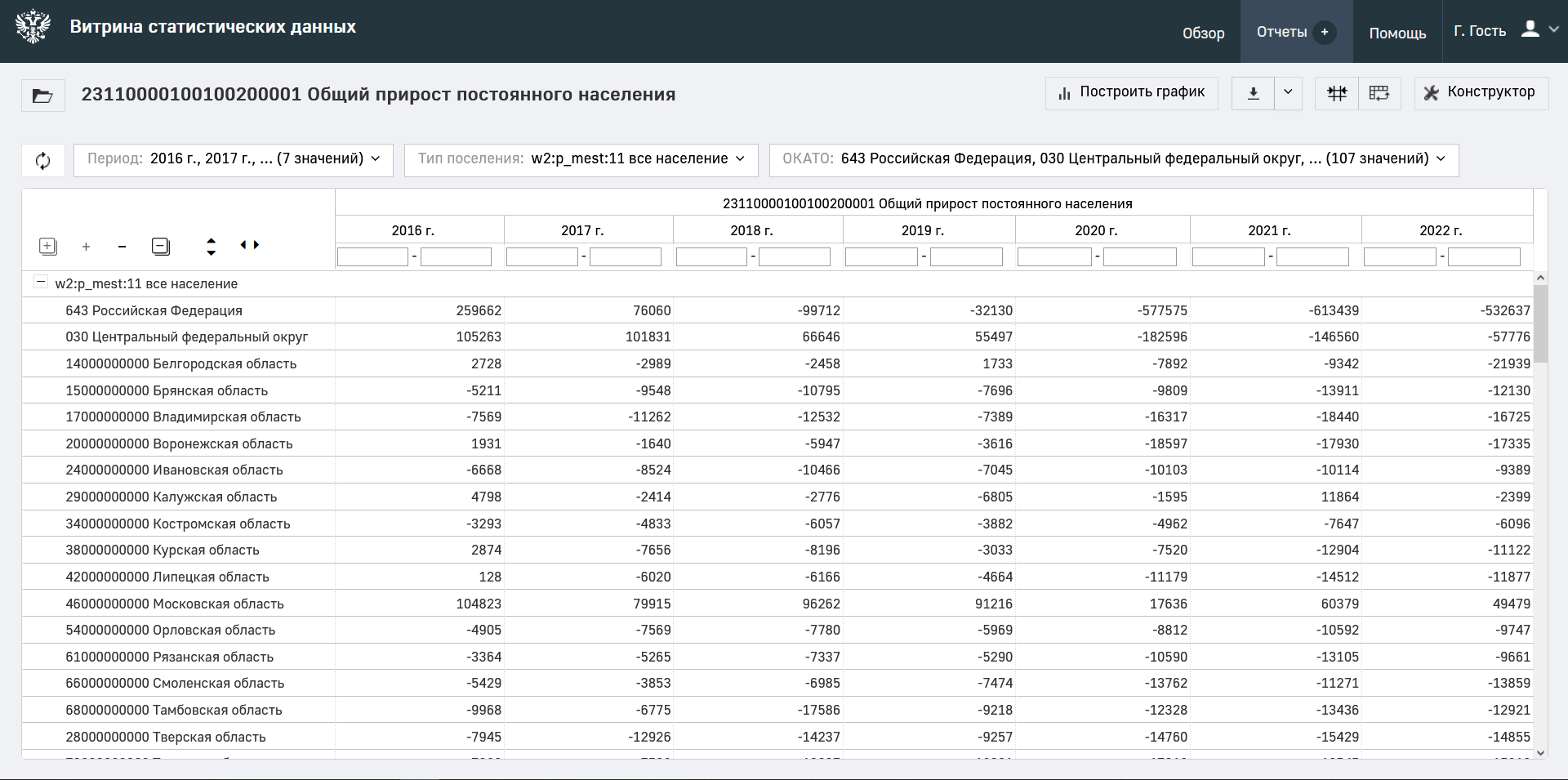


Рисунок 1.2 – Интерфейс ВСД

* 1. Всемирный банк

Всемирный банк – это международная финансовая организация, созданная с целью предоставления финансовой и технической помощи развивающимся странам.

Всемирный банк предоставляет различные инструменты и ресурсы для работы со статистическими данными. Такие как:

1 Данные Всемирного банка. Это онлайн-платформа, которая предоставляет доступ к разнообразным статистическим данным по странам и регионам мира. Пользователи могут проводить поиск, фильтрацию, а также скачивать данные для дальнейшего анализа.

2 Индикаторы мирового развития. Этот набор данных включает в себя ключевые показатели по мировому развитию, такие как данные о бедности, занятости, образовании, здравоохранении и другие. Индикаторы предоставляются в виде таблиц и графиков.

3 Открытые данные. Всемирный банк активно поддерживает инициативу открытых данных, предоставляя свободный доступ к широкому спектру статистических данных. Эти данные могут быть использованы для различных целей, включая исследования, образование и разработку.

4 База данных. Это веб-приложение, позволяющее пользователям исследовать, сравнивать и анализировать данные по различным показателям и временным периодам. База данных предоставляет гибкий интерфейс для работы с множеством статистических данных.

5. Библиотека микроданных. Этот ресурс предоставляет доступ к микроданным, которые могут быть использованы для более детального исследования структуры и характеристик статистических данных.

6 Всемирное интегрированное торговое решение. Этот инструмент предоставляет доступ к мировой торговой статистике. Он включает данные о торговле товарами и услугами для различных стран и регионов.

7 Интерфейс программирования приложений. Всемирный банк предоставляет интерфейс для доступа к статистическим данным, что позволяет разработчикам интегрировать данные в свои приложения и аналитические инструменты.

На рисунке 1.3 представлена таблица, полученная с помощью базы данных Всемирного банка.

Можно выделить следующие плюсы Всемирного банка:

– мировой охват данных;

– широкий спектр данных;

– доступность данных;

– международная методология.

Также наблюдаются следующие минусы:

– ограниченная детализация статистических данных;

– временная задержка результатов;

– ориентация на экономические аспекты;

– зависимость от добровольных отчетов.

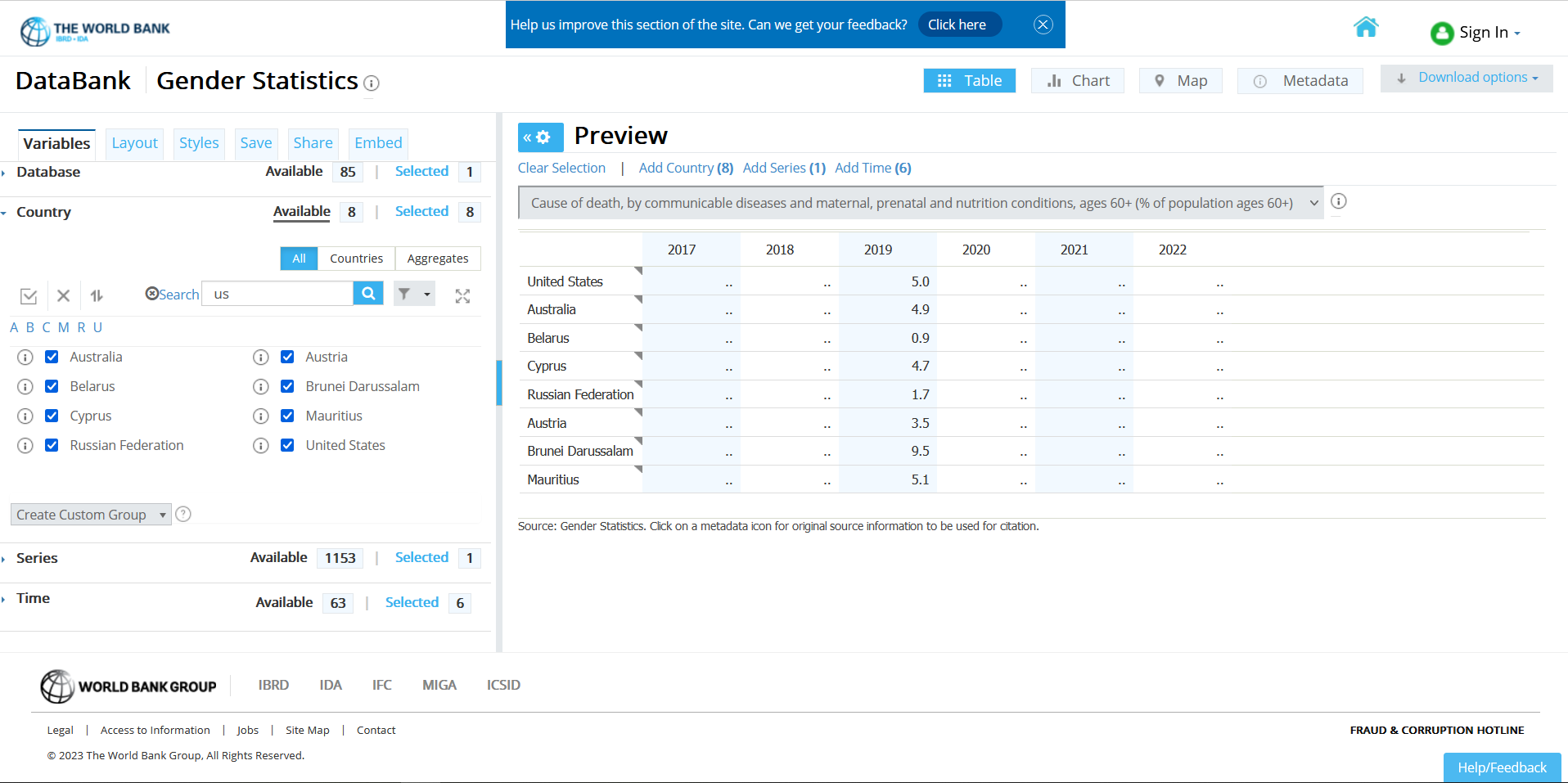


Рисунок 1.3 – Интерфейс базы данных Всемирного банка

1. Формирование требований к базе данных

База данных для статистического центра должна обладать рядом важных качеств, учитывая специфику и требования обработки статистических данных.

Необходимо, чтобы база данных была спроектирована и реализована таким образом, чтобы обеспечить максимальную надежность и целостность данных. Так как обеспечение надежности и целостности данных является фундаментальным аспектом управления информацией в базах данных. Надежные и целостные данные создают основу для доверия к информации в базе данных, что крайне важно для пользователей. Также существуют законы и нормы, регулирующие обработку данных, и обеспечение целостности данных становится важным аспектом для соблюдения этих требований. Это также способствует защите от потери информации в результате сбоев, сбросов или других технических проблем. Целостные данные минимизируют ошибки и проблемы, связанные с некорректными или несогласованными данными, что, в свою очередь, упрощает процессы работы с информацией и повышает производительность пользователей. Они также способствуют поддержанию консистентности в базе данных, предотвращая противоречия и несогласованность в информации.

В контексте статистических центров, где точность и достоверность данных критичны для оценки социально-экономических показателей, обеспечение надежности и целостности данных существенно влияет на качество предоставляемого сервиса. Ненадежные или поврежденные данные могут повлиять на репутацию организации или статистического центра, в то время как качественные и достоверные данные способствуют поддержанию положительной репутации.

Обеспечение высокой производительности для базы данных статистического центра представляет собой важный аспект для реализации по нескольким ключевым причинам. Во-первых, эффективность обработки запросов играет решающую роль в оперативном анализе данных и принятии решений, удовлетворяя ожидания пользователей. Высокая производительность обеспечивает отзывчивость системы, что существенно для оперативной работы аналитиков и исследователей. Во-вторых статистические центры оперируют большим объемом разнообразных данных, включая временные ряды, социально-экономические показатели и географическую информацию. Поэтому база данных должна быть способной эффективно обрабатывать масштабные объемы информации. Производительность базы данных также оказывает влияние на оптимизацию аналитических процессов, что важно для выявления тенденций и статистических закономерностей.

Необходимо обеспечить масштабируемость для базы данных статистического центра. Обеспечение масштабируемости для базы данных статистического центра представляет собой критически важный аспект с учетом нескольких основных причин. Прежде всего, статистические центры, занимающиеся обширными объемами данных о населении, экономике и социальных процессах, сталкиваются с постоянным ростом данных. Масштабируемость базы данных становится необходимой для эффективного управления этим ростом и обеспечения беспрепятственного функционирования.

Гибкость в обработке нагрузки также является важным аспектом, особенно учитывая временную изменчивость данных в статистических центрах, например, в периоды проведения переписи населения. Масштабируемость обеспечивает адаптацию к колебаниям в нагрузке, что поддерживает стабильность работы базы данных.

Масштабируемость также предоставляет гибкость для будущего развития системы, обеспечивая возможность легкого добавления новых ресурсов и соответствие изменяющимся требованиям. Это особенно важно в контексте статистических центров, где система должна адаптироваться к постоянно меняющейся информационной среде и оставаться устойчивой в долгосрочной перспективе.

Процесс нормализации [5] в базах данных играет ключевую роль в обеспечении эффективности и структурированности данных. В первую очередь, первая нормальная форма (1НФ) требует, чтобы каждая таблица имела уникальный первичный ключ, обеспечивая уникальную идентификацию каждой записи и исключая дублирование данных.

Вторая нормальная форма (2НФ) углубляет этот принцип, требуя полной зависимости всех неключевых атрибутов от всего первичного ключа. Это предотвращает возможные аномалии данных и способствует более структурированной базе данных.

Третья нормальная форма (3НФ) дополняет процесс, устраняя транзитивные зависимости между неключевыми атрибутами. Это обеспечивает гибкость в организации данных и снижает избыточность.

Применение этих принципов нормализации не только повышает структурированность данных, но и уменьшает вероятность ошибок в их обработке. Обеспечивая целостность и минимизацию избыточности данных, база данных становится более эффективной для выполнения запросов и управления информацией.

Данные организуются в виде таблиц, где каждая таблица представляет сущность, а строки этой таблицы представляют отдельные записи, каждая из которых уникально идентифицируется первичным ключом.

Необходимо использование внешних ключей для установления связей между таблицами. Это обеспечивает согласованность данных и позволяет эффективно связывать информацию из различных таблиц.

База данных должна быть нормализована согласно требованиям нормальных форм. До третьей нормальной формы включительно, что означает минимизацию избыточности данных и обеспечение их целостности.

Относительно количества сущностей, система должна содержать 25 сущностей. Каждая из этих сущностей представляет определенный объект или концепцию, а их атрибуты должны быть логически и связанно представлены внутри реляционной модели.

Общие требования поддержания целостности данных, избегание избыточности и обеспечение удобства обработки запросов и управления информацией должны быть обеспечены с использованием реляционной модели данных.

Разрабатываемая в настоящий момент база данных, включающая в себя вышеописанные критерии, согласно требованиям текущего проекта должна быть реализована с использованием исключительно реляционной модели данных.

1. Проектирование моделей данных

В данном разделе рассматривается проектирование моделей данных c использованием как просто описания словами, так и диаграмм. Диаграммы рассматриваются по убыванию абстрактности до физической модели данных.

Проектирование моделей данных в ходе разработки программного обеспечения представляет собой ключевой этап, нацеленный на обеспечение эффективности, надежности и удобства поддержки системы. Оно направлено на формирование структурированного и ясного представления о том, как данные будут организованы, взаимодействовать и использоваться в рамках приложения. Модели данных на этом уровне также служат основой для обеспечения целостности данных, оптимизации операций с базой данных, обеспечения согласованности в команде разработчиков, обеспечения легкости внесения изменений, обеспечения безопасности и конфиденциальности данных, а также поддержки процессов тестирования и отладки. Грамотно разработанные модели данных создают прочную основу для будущего расширения системы, обеспечивая гибкость и адаптивность к изменяющимся требованиям.

При переходе на концептуальный уровень моделирования баз данных выделяются ключевые концепции и отношения, существующие в предметной области, вне зависимости от того, как они будут физически реализованы в базе данных. Этот уровень моделирования фокусируется на описании структуры данных и их взаимосвязей, освобожденных от технических деталей и специфик систем управления базами данных. Здесь выделяются сущности, их атрибуты, отношения, ключи, ассоциации и кардинальность. Важным аспектом на данном уровне является также возможность проведения нормализации для избежания избыточности данных и обеспечения их целостности.

Перейдя на логический уровень, фокус смещается на более детализированные аспекты, связанные с конкретной реализацией базы данных. Описываются структуры таблиц, ключи, индексы и другие технические детали, которые напрямую связаны с функционированием системы управления базами данных. Логический уровень обеспечивает более конкретное представление о том, как концепции концептуального уровня будут технически воплощены в реальной базе данных.

Логический уровень моделирования баз данных является абстрактным, высокоуровневым представлением данных и их взаимосвязей, что обеспечивает лучшее понимание структуры предметной области и формирует основу для последующего проектирования и реализации физической базы данных.

Разрабатываемую базу данных можно разбить на два зависимых участка:

– часть, общая для всех статистических характеристик;

– конкретные статистические характеристики.

Участок, который отвечает за хранение информации о рождении детей представлен диаграммой IDEF1X на рисунке 3.1.

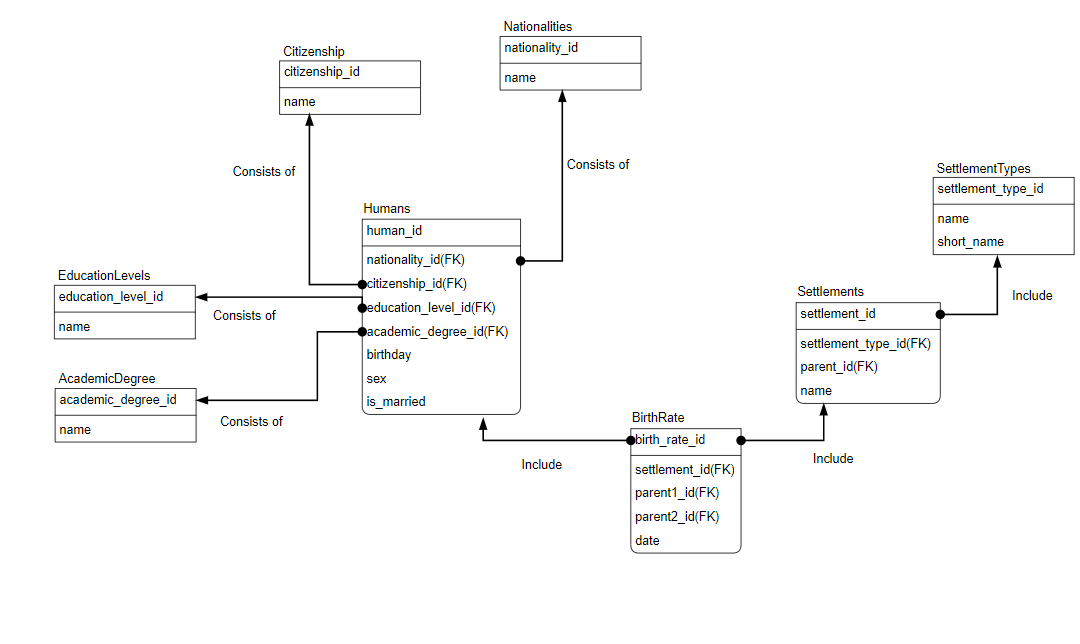


Рисунок 3.1 – Диаграмма IDEF1X показывающая хранение рождаемости

На данной диаграмме есть независимые таблицы, которые хранят название какого-то типа или какой-нибудь характеристики. Например, таблица с национальностями хранит все национальности определенные в системе. И есть зависимые таблицы, которые описывают такие сущности как человек, определенная территория и запись о рождении ребенка. Человек содержит в себе информацию интересную для статистики. Например, какой национальности родители ребенка, или какой уровень образования у родителей ребенка.

Участок, который отвечает за хранение смертей представлен диаграммой IDEF1X на рисунке 3.2.

В отличии от хранения рождения детей, для хранения смертей недостаточно сущностей территория и человек. Для этого еще необходима сущность причина смерти. Причина смерти описывается названием и кодом МКВ-10. МКБ-10 – международная классификация болезней десятого пересмотра. Представляет собой нормативный документ с общепринятой статистической классификацией медицинских диагнозов, которая используется в здравоохранении для унификации методических подходов и международной сопоставимости материалов [2].

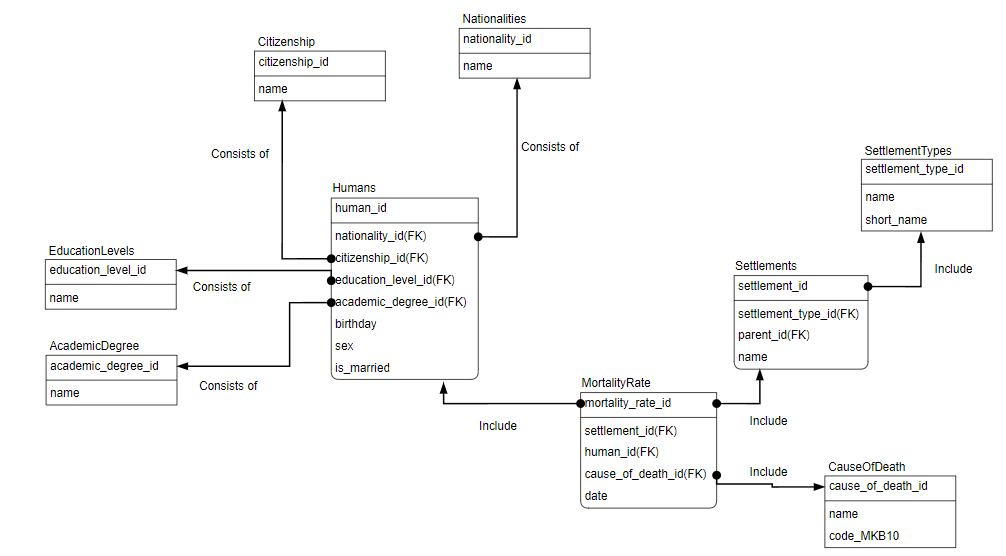


Рисунок 3.2 – Диаграмма IDEF1X показывающая хранение смертей

* 1. Определение сущностей и их связей

В результате исследования предметной области в ней были выделены сущности. Сущности и их описание перечислены в таблице 3.1.

Таблица 3.1 – Сущности системы

|  |  |
| --- | --- |
| Сущность | Описание |
| Ученая степень | Степень квалификационной системы в науке, позволяющей ранжировать научных деятелей на отдельных этапах академической карьеры. Решение о присуждении учёной степени базируется на оценке только научно-исследовательского уровня соискателя. Стаж в конкретной должности, педагогические достижения и иные показатели не учитываются, в отличие от ситуации присвоения учёных званий[3]. |
| Адрес | Уникальная структурированная иерархическая совокупность элементов, идентифицирующая местонахождение [4]. |
| Цели прибытия | Цель прибытия на новое место. Самые частые цели прибытия: работа или учеба. Но также могут встречаться более специфичные цели. |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| Сущность | Описание |
| Запись о прибытии | Запись о прибытие содержит важную информацию для статистики. И включает следующие данные: характеристики человека, который прибыл, дату прибытия, откуда и куда прибыл, а также цель прибытия. |
| Запись о рождении | Запись о рождении содержит важную информацию для статистики. И включает следующие данные: характеристики родителей, дату и время рождения, а также место рождения. |
| Причина смерти | Причины смерти все болезни, патологические состояния и травмы, которые привели к смерти или способствовали ее наступлению, а также обстоятельства происшествия или акта насилия, которые привели к таким травмам. В целях демографической статистики, симптомы или стадии смерти, такие как остановка сердца или астения, не считаются причинами смерти [5]. |
| График | Представляет определенные в системе графики. Описывается названием графика и описанием. |
| Гражданство | Правовая связь человека и государства, выражающаяся в совокупности их взаимных прав, обязанностей и ответственности [6]. Представляет определенные в системы гражданства. |
| Источник данных | Представляет источник данных, из которого были получены определенные данные. |
| Отдел | Представляет структурное подразделение какой-нибудь организации. |
| Запись о разводах | Запись о разводах содержит важную информацию для статистики. И включает следующие данные: характеристики людей, которые разводятся, дату развода, а также количество детей. |
| Уровень образования | Представляет собой уровень образования определенного человека. В Беларуси определены следующие уровни образования: дошкольное, общее среднее, профессионально-техническое, среднее специальное, высшее. |
| Должность | Представляет собой служебное положение работника, обусловленное кругом его обязанностей, должностными правами и характером ответственности [7]. |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| Сущность | Описание |
| Цель отбытия | Цель отбытия на новое место. Важная характеристика, по которой можно анализировать, почему люди покидают определенные места. Самые частые цели отбытия: работа или учеба. Но также могут встречаться более специфичные цели. |
| Человек | Представляет собой наиболее важные для статистики характеристики человека. Например, такие характеристики, как: дата рождения, пол, национальность, гражданство, уровень образования, ученую степень, и в браке ли находится человек. |
| Запись об отбытии | Запись об отбытии содержит важную информацию для статистики. И включает следующие данные: характеристики человека, который отбыл, дату отбытия, откуда и куда отбыл, а также цель отбытия. |
| Запись о браке | Запись о браке содержит важную информацию для статистики. И включает следующие данные: характеристики людей, которые вступают в брак, дату вступления в брак, место вступления в брак. |
| Запись о смерти | Запись о смерти содержит важную информацию для статистики. И включает следующие данные: характеристики человека, дату и время, место и причину. |
| Национальность | Принадлежность лица к этнической группе, характеризуемая родным языком, особенностями быта, традициями, обычаями, культурой, религией, родством и другими признаками, позволяющими лицу идентифицировать себя [8]. |
| Организация | Представляет собой основную информацию об организации, такую как: адрес, название, номер телефона и адрес электронной почты. |
| Тип территории | Представляет собой определенный тип территории. Например, страна, город, область, район, агрогородок, деревня. |
| Территория | Представляет собой конкретную территорию. Например, город Минск, Гродненская область, агрогородок Рожанка и т.д. |
| Пользователь | Представляет собой определенного пользователя системы. |

Продолжение таблицы 3.1

|  |  |
| --- | --- |
| Сущность | Описание |
| Статистический показатель | Количественно выраженное определенное свойство, качество совокупности в целом или ее частей [9]. |
| Интервал времени | Представляет собой непустой отрезок временной шкалы. |
| Единица измерения | Конкретная величина, определенная и установленная по договоренности, с которой сопоставляются другие величины того же рода, для того чтобы выразить их размер по отношению к указанной величине [10]. |

Как видно из таблицы 3.1 часть сущностей является общей для всей базы данных статистического центра. А часть сущностей является специфическими для определенных статистических показателей.

При разработке структуры базы данных выделены отношения между сущностями. Данные отношения можно увидеть в таблица 3.2.

Таблица 3.2 – Отношения между сущностями

|  |  |  |
| --- | --- | --- |
| Название связи | Связываемые сущности | Пояснение |
| Содержит | Адрес – Территория | В адрес входит определенная территория. Помимо территории в адрес входит конкретная улица на территории, и конкретный дом на улице. |
| Содержит | Запись о прибытии – Цель прибытия | Запись о прибытии содержит в себе данные, важные для дальнейших статистических исследований. Цель прибытия является интересной для исследователей характеристикой. |
| Содержит | Запись о прибытии – Человека | Запись о прибытии содержит в себе данные, важные для дальнейших статистических исследований. Характеристики человека интересуют исследователей. |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| Название связи | Связываемые сущности | Пояснение |
| Содержит | Запись о прибытии – Территория | Запись о прибытии содержит в себе данные, важные для дальнейших статистических исследований. Откуда и куда прибыл человек являются интересными данными для исследователей. |
| Содержит | Запись об отбытии – Цель отбытия | Запись об отбытии содержит в себе данные, важные для дальнейших статистических исследований. Цель отбытия является интересной для исследователей характеристикой. |
| Содержит | Запись об отбытии – Человека | Запись об отбытии содержит в себе данные, важные для дальнейших статистических исследований. Характеристики человека интересуют исследователей. |
| Содержит | Запись об отбытии – Территория | Запись об отбытии содержит в себе данные, важные для дальнейших статистических исследований. Откуда и куда прибыл человек являются интересными данными для исследователей. |
| Содержит | Запись о разводе – Человек | Запись о разводе содержит в себе данные, важные для дальнейших статистических исследований. Характеристики человека интересуют исследователей. |
| Содержит | Запись о разводе – Территория | Запись о разводе содержит в себе данные, важные для дальнейших статистических исследований. Исследователей интересует где произошел развод. |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| Название связи | Связываемые сущности | Пояснение |
| Содержит | Запись о браке – Территория | Запись о браке содержит в себе данные, важные для дальнейших статистических исследований. Исследователей интересует где был заключен брак. |
| Содержит | Запись о браке – Человек | Запись о браке содержит в себе данные, важные для дальнейших статистических исследований. Характеристики человека интересуют исследователей. |
| Содержит | Запись о рождении – Территория | Запись о рождении ребенка содержит в себе данные, важные для дальнейших статистических исследований. Исследователей интересует где был рожден ребенок. |
| Содержит | Запись о рождении – Человек | Запись о рождении ребенка содержит в себе данные, важные для дальнейших статистических исследований. Характеристики родителей ребенка интересуют исследователей. |
| Содержит | Запись о смерти – Территория | Запись о смерти содержит в себе данные, важные для дальнейших статистических исследований. Исследователей интересует где умер человек. |
| Содержит | Запись о смерти – Человек | Запись о смерти содержит в себе данные, важные для дальнейших статистических исследований. Характеристики умершего человека интересуют исследователей. |
| Содержит | Запись о смерти – Причина смерти | Запись о смерти содержит в себе данные, важные для дальнейших статистических исследований. Исследователей интересует причина смерти человека. |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| Название связи | Связываемые сущности | Пояснение |
| Имеет | Человек– Национальность | Определенный человек имеет национальность. А национальность человека интересна в статистических исследованиях. |
| Имеет | Человек– Гражданство | Определенный человек имеет гражданство. А гражданство человека может быть интересно в статистических исследованиях. |
| Имеет | Человек– Уровень образования | Определенный человек имеет определенный уровень образования. А уровень образования человека может быть интересным в статистических исследованиях. |
| Имеет | Человек– Ученая степень | Определенный человек имеет определенную ученую степень. А ученая степень человека может быть интересной в статистических исследованиях. |
| Содержит | Территория– Территория | Территория может содержать в себе другую, более маленькую территорию. |
| Имеет | Пользователь– Должность | Каждый пользователь имеет определенную должность. |

* 1. Определение атрибутов сущностей

В данном пункте проводится анализ и описание атрибутов, характеризующих сущности, представленные в структуре базы данных. Каждая таблица базы данных представляет собой определенный тип сущности, а ее поля (атрибуты) определяют характеристики этой сущности. Данный раздел предоставляет подробное описание каждой сущности, выделяя ее ключевые атрибуты и их типы данных. Анализ атрибутов в контексте текущей базы данных позволяет лучше понять структуру данных, их взаимосвязи и обеспечивает основу для последующего проектирования запросов и взаимодействия с информацией в рамках разрабатываемого программного продукта.

Увидеть сущности и их атрибуты можно в таблице 3.3.

Таблица 3.3 – Атрибуты сущностей

|  |  |
| --- | --- |
| Сущность | Основные атрибуты |
| Positions | position\_id, name |
| Departments | department\_id, name |
| Users | user\_id, name, surname, patronymic, phone\_num, city\_telephone\_num, email, position\_id, department\_id |
| UnitsOfMeasure | unit\_of\_measure\_id, name, description |
| TimeIntervals | time\_interval\_id, name, interval\_in\_seconds |
| Charts | chart\_id, name, description |
| SettlementsType | settlement\_type\_id, name, short\_name |
| Settlements | settlement\_id, settlements\_type\_id, parent\_id, name |
| Addresses | address\_id, settlement\_id, street, house\_num |
| Organizations | organization\_id, address\_id, name, short\_name, phone\_num, city\_telephone\_num, email |
| Nationalities | nationality\_id, name |
| Citizenships | citizenship\_id, name |
| EducationLevels | education\_level\_id, name |
| AcademicDegrees | academic\_degree\_id, name |
| Humans | human\_id, birthday, sex, nationality\_id, citizenship\_id, education\_level\_id, academic\_degree\_id, is\_married |
| BirthRate | birth\_rate\_id, date, settlement\_id, parent1\_id, parent2\_id |
| CausesOfDeath | cause\_of\_death\_id, name, code\_MKB10 |
| MortalityRate | mortality\_rate\_id, human\_id, cause\_of\_death\_id, settlement\_id, date |
| Marriages | marriage\_id, human1\_id, human2\_id, settlement\_id, date |
| Divorces | divorce\_id, human1\_id, human2\_id, settlement\_id, date, children\_count |
| ArrivalGoal | arrival\_goal\_id, goal |
| ArrivalRate | arrival\_rate\_id, human\_id, date, from\_id, to\_id, goal\_id |
| LeavingGoal | leaving\_goal\_id, goal |
| LeavingRate | arrival\_rate\_id, human\_id, date, from\_id, to\_id, goal\_id |
| СonceptsAndDefinitions | concept\_and\_definition\_id, concept, definition |
| DataSources | data\_source\_id, name, reliability |
| StatisticalIndicators | statistical\_indicator\_id, name, last\_update, creation\_date, comment, contact\_person\_id |

1. разработка базы данных

Разработка базы данных – ключевой этап в создании информационной системы, направленной на эффективное хранение, обработку и управление данными. В данном разделе мы углубимся в процесс разработки базы данных, рассмотрим выбор технических средств, создание таблиц, разработку хранимых процедур, триггеров и индексов.

* 1. Выбор технических средств

В ходе процесса выбора системы управления базами данных (СУБД) для реализации разработанной модели данных, был проведен тщательный анализ нескольких популярных реляционных СУБД, с целью определения оптимального решения, соответствующего требованиям проекта. Рассмотрим основные аналоги, которые рассматривались в контексте принятия решения: Oracle Database, MySQL, SQLite иPostgreSQL и Microsoft SQL Server.

Oracle Database представляет собой мощную СУБД с высокой производительностью и обширными функциональными возможностями. Способна масштабироваться от небольших приложений до крупных предприятий, обрабатывая огромные объемы данных. Поддерживает транзакционную целостность данных, гарантируя надежность и согласованность в рамках транзакций. Имеет возможность интеграции с другими приложениями и обширные инструменты для управления базой данных, включая средства мониторинга и отчетности. Однако, она является проприетарным решением с высокими затратами на лицензирование, что может быть ограничивающим фактором для некоторых проектов.

MySQL – это открытая реляционная система управления базами данных, широко используемая для хранения, управления и обработки данных. MySQL, отличающаяся высокой производительностью, также она поддерживает транзакции. Способна масштабироваться от небольших веб-приложений до крупных корпоративных систем, обрабатывая различные объемы данных. Встроенные механизмы безопасности, такие как шифрование данных, аутентификация и авторизация, обеспечивают защиту информации. Обширное сообщество пользователей и разработчиков MySQL обеспечивает доступ к множеству ресурсов, включая форумы, документацию и обновления. При этом всем MySQL распространяется под лицензией General Public License (GPL), что делает ее бесплатной и доступной для широкого круга пользователей. Однако, она может иметь ограниченные функциональные возможности по сравнению с некоторыми другими СУБД.

SQLite – это встраиваемая реляционная система управления базами данных общего назначения, которая является самодостаточной и не требует отдельного сервера для своей работы. SQLite выделяется легкостью в развертывании, не требует сервера баз данных и прост в использовании. Однако, она не предназначена для крупных и сложных проектов, имеет ограниченные возможности масштабирования.

Microsoft SQL Server — система управления реляционными базами данных, разработанная корпорацией Microsoft [11]. MS SQL Server интегрирован с другими продуктами и технологиями Microsoft, такими как .NET, Azure, Microsoft Excel, что обеспечивает совместимость и удобство использования в экосистеме Microsoft. Способен масштабироваться от небольших веб-приложений до крупных корпоративных систем, обрабатывая различные объемы данных. Включает в себя инструменты для администрирования, мониторинга и оптимизации производительности баз данных. Но лицензионные ограничения могут быть затратными.

PostgreSQL – это мощная объектно-реляционная система управления базами данных с открытым исходным кодом. PostgreSQL поддерживает множество расширенных функций SQL, включая сложные запросы, хранимые процедуры, триггеры, оконные функции и многое другое. Способен масштабироваться от небольших проектов до крупных предприятий, обрабатывая сложные и объемные базы данных. Обеспечивает надежное хранение данных, поддержку транзакций, резервное копирование и механизмы восстановления после сбоев. PostgreSQL имеет активное сообщество пользователей и разработчиков, а также обширную документацию, что облегчает использование и поддержку. Но может быть тяжелее других СУБД в настройке.

Обобщим всё вышесказанное в таблицу 4.1.

Таблица 4.1 – Сравнение СУБД

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Oracle DB | MySQL | SQLite | PostgreSQL | Microsoft SQL Server |
| Высокая производительность | + | + | + | + | + |
| Масштабируемость | + | + | - | + | + |
| Наличие опыта у разработчиков | + | + | + | - | - |
| Поддержка транзакций | + | + | + | + | + |
| Свободная и открытая лицензия | - | + | + | + | - |
| Обширное сообщество и поддержка | + | + | + | + | + |

Таким образом в ходе рассмотрения различных аналогов систем управления базами данных и тщательного их анализа, в качестве СУБД для реализации данного курсового проекта была выбрана MySQL.

* 1. Разработка таблиц базы данных

В результате разработки вышеописанных моделей данных на основе технического задания получен список отношений и таблиц, соответствующих сущностям логической модели данных.

Таблица «Positions» описывает сущность должности. Здесь «position\_id» служит идентификатором, а «name» хранит название должности. Использование AUTO\_INCREMENT для «position\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «Departments» описывает сущность структурного подразделения. Здесь «department\_id» служит идентификатором, а «name» хранит название структурного подразделения. Использование AUTO\_INCREMENT для «department\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «Users» описывает сущность пользователя. Здесь «user\_id» служит идентификатором, «name» хранит имя, «surname» хранит фамилию, «patronymic» хранит отчество, «phone\_num» хранит номер телефона, «city\_telephone\_num» хранит номер городского телефона, «email» хранит адрес электронной почты, «position\_id» хранит идентификатор должности, а «department\_id» хранит идентификатор структурного подразделения. Использование AUTO\_INCREMENT для «user\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(position\_id) связывает пользователя с должностью, а FOREIGN KEY(department\_id) – со структурным подразделением.

Таблица «UnitsOfMeasure» описывает сущность единицы измерения. Здесь «unit\_of\_measure\_id» служит идентификатором, «name» хранит название, а «description» предоставляет описание данной единицы измерения. Использование AUTO\_INCREMENT для «unit\_of\_measure\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. Текстовое поле «description» оставлено nullable, что позволяет сохранять единицы измерения без подробных описаний, если они не нужны.

Таблица «TimeIntervals» описывает сущность временного интервала. Здесь «time\_interval\_id» служит идентификатором, «name» хранит название интервала, а «interval\_in\_seconds» размер данного интервала в секундах. Использование AUTO\_INCREMENT для «time\_interval\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «Charts» описывает сущность графика. Здесь «chart\_id» служит идентификатором, «name» хранит название, а «description» предоставляет описание. Использование AUTO\_INCREMENT для «chart\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. Текстовое поле «description» оставлено nullable, что позволяет сохранять графики без подробных описаний, если они не нужны.

Таблица «SettlementsType» описывает сущность типа территории. Здесь «settlement\_type\_id» служит идентификатором, «name» хранит название, а «short\_name» предоставляет сокращение. Использование AUTO\_INCREMENT для «settlement\_type\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. Текстовое поле «short\_name» оставлено nullable, что позволяет сохранять типы территорий без подробных описаний, если они не нужны.

Таблица «Settlements» описывает сущность территории. Здесь «settlement\_id» служит идентификатором, «name» хранит название, «settlement\_type\_id» хранит идентификатор типа территории, а «parent\_id» хранит идентификатор на территорию, в которую включена данная территория. Использование AUTO\_INCREMENT для «settlement\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает территорию с родительской территорией, а FOREIGN KEY(settlement\_type\_id) – с типом территории.

Таблица «Addresses» описывает сущность адреса. Здесь «address\_id» служит идентификатором, «settlement\_id» хранит идентификатор территории, «street» хранит название улицы, а «house\_num» хранит номер дома. Использование AUTO\_INCREMENT для «address\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает адрес с конкретной территорией.

Таблица «Organizations» описывает сущность организации. Здесь «organization\_id» служит идентификатором, «address\_id» хранит идентификатор адреса, «name» хранит название организации, «short\_name» хранит краткое название, «phone\_num» хранит номер телефона, «city\_telephone\_num» хранит номер городского телефона, а «email» хранит адрес электронной почты. Использование AUTO\_INCREMENT для «organization\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(address\_id) связывает организацию с конкретным адресом. Текстовые поля «short\_name», «phone\_num», «city\_telephone\_num» и «email» оставлены nullable, что позволяет сохранять организации без эти необязательных полей.

Таблица «Nationalities» описывает сущность национальности. Здесь «nationality\_id» служит идентификатором, а «name» хранит название. Использование AUTO\_INCREMENT для «nationality\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «Citizenships» описывает сущность гражданства. Здесь «citizenship\_id» служит идентификатором, а «name» хранит название. Использование AUTO\_INCREMENT для «citizenship\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «EducationLevels» описывает сущность уровеня образования. Здесь «education\_level\_id» служит идентификатором, а «name» хранит название. Использование AUTO\_INCREMENT для «education\_level\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «AcademicDegrees» описывает сущность ученой степени. Здесь «academic\_degree\_id» служит идентификатором, а «name» хранит название. Использование AUTO\_INCREMENT для «academic\_degree\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «Humans» описывает сущность человека. Здесь «human\_id» служит идентификатором, «birthday» хранит дату рождения, «sex» хранит пол, «nationality\_id» хранит идентификатор национальности, «citizenship\_id» хранит идентификатор гражданства, «education\_level\_id» хранит идентификатор уровня образования, «academic\_degree\_id» хранит идентификатор ученой степени, а «is\_married» показвает, находится ли данный человек в браке. Использование AUTO\_INCREMENT для «human\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(nationality\_id) связывает человека с национальностью, FOREIGN KEY(citizenship\_id) – с гражданством, FOREIGN KEY(education\_level\_id) – с уровнем образования, а FOREIGN KEY(academic\_degree\_id) – с ученой степенью.

Таблица «BirthRate» описывает сущность записи о рождении. Здесь «birth\_rate\_id» служит идентификатором, «date» хранит дату и время рождения, «settlement\_id» хранит идентификатор территории, «parent1\_id» хранит идентификатор первого родителя, «parent2\_id» хранит идентификатор второго родителя. Использование AUTO\_INCREMENT для «birth\_rate\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает запись о рождении с территорией, FOREIGN KEY(parent1\_id) – с первым родителем, FOREIGN KEY(parent2\_id) – со вторым родителем.

Таблица «CausesOfDeath» описывает сущность причины смерти. Здесь «cause\_of\_death\_id» служит идентификатором, «name» хранит название, а «code\_MKB10» хранит код причины смерти в формате МКБ-10. Использование AUTO\_INCREMENT для «cause\_of\_death\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «MortalityRate» описывает сущность записи о смерти. Здесь «mortality\_rate\_id» служит идентификатором, «date» хранит дату и время смерти, «settlement\_id» хранит идентификатор территории, «human\_id» хранит идентификатор человека, «cause\_of\_death\_id» хранит идентификатор причины смерти. Использование AUTO\_INCREMENT для «mortality\_rate\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает запись о смерти с территорией, FOREIGN KEY(human\_id) – с человеком, FOREIGN KEY(cause\_of\_death\_id) – с причиной смерти.

Таблица «Marriages» описывает сущность записи о браке. Здесь «marriage\_id» служит идентификатором, «date» хранит дату заключения брака, «settlement\_id» хранит идентификатор территории, «human1\_id» хранит идентификатор первого человека, «human2\_id» хранит идентификатор второго человека. Использование AUTO\_INCREMENT для «marriage\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает запись о браке с территорией, FOREIGN KEY(human1\_id) – с первым человеком, FOREIGN KEY(human2\_id) – со вторым человеком.

Таблица «Divorces» описывает сущность записи о разводе. Здесь «divorce\_id» служит идентификатором, «date» хранит дату развода, «settlement\_id» хранит идентификатор территории, «human1\_id» хранит идентификатор первого человека, «human2\_id» хранит идентификатор второго человека, а «children\_count» хранит количество детей. Использование AUTO\_INCREMENT для «divorce\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(settlement\_id) связывает запись о разводе с территорией, FOREIGN KEY(human1\_id) – с первым человеком, FOREIGN KEY(human2\_id) – со вторым человеком.

Таблица «ArrivalGoal» описывает сущность цели прибытия. Здесь «arrival\_goal\_id» служит идентификатором, а «goal» хранит цель. Использование AUTO\_INCREMENT для «arrival\_goal\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «LeavingGoal» описывает сущность цели отбытия. Здесь «leaving\_goal\_id» служит идентификатором, а «goal» хранит цель. Использование AUTO\_INCREMENT для «leaving\_goal\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «ArrivalRate» описывает сущность записи о прибытии. Здесь «arrival\_rate\_id» служит идентификатором, «date» хранит дату прибытия, «from\_id» хранит идентификатор территории c которой прибыл человек, «to\_id» хранит идентификатор территории на которую прибыл человек, «human\_id» хранит идентификатор человека, «goal\_id» хранит идентификатор цели прибытия. Использование AUTO\_INCREMENT для «arrival\_rate\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(from\_id) связывает запись о прибытии с территорией с которой прибыл человек, FOREIGN KEY(to\_id) – с территорией на которую прибыл человек, FOREIGN KEY(human\_id) – с прибывшим человеком, а FOREIGN KEY(goal\_id) – с целью.

Таблица «LeavingRate» описывает сущность записи об отбытии. Здесь «leaving\_rate\_id» служит идентификатором, «date» хранит дату отбытия, «from\_id» хранит идентификатор территории c которой отбыл человек, «to\_id» хранит идентификатор территории на которую отбыл человек, «human\_id» хранит идентификатор человека, «goal\_id» хранит идентификатор цели отбытия. Использование AUTO\_INCREMENT для «leaving\_rate\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(from\_id) связывает запись об отбытии с территорией, с которой отбыл человек, FOREIGN KEY(to\_id) – с территорией на которую отбыл человек, FOREIGN KEY(human\_id) – с отбывшим человеком, а FOREIGN KEY(goal\_id) – с целью.

Таблица «СonceptsAndDefinitions» описывает сущность понятия и его определения. Здесь «concept\_and\_definition\_id» служит идентификатором, «concept» хранит понятие, а «definition» хранит определение. Использование AUTO\_INCREMENT для «concept\_and\_definition\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «DataSources» описывает сущность источника данных. Здесь «data\_source\_id» служит идентификатором, «name» хранит название, а «reliability» хранит коэффициент надежности. Использование AUTO\_INCREMENT для «data\_source\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей.

Таблица «StatisticalIndicators» описывает сущность статистических показателей. Здесь «statistical\_indicator\_id» служит идентификатором, «name» хранит название, «last\_update» хранит дату и время последнего обновления, «creation\_date» хранит дату и время создания, «comment» хранит комментарий, а «contact\_person\_id» хранит идентификатор пользователя. Использование AUTO\_INCREMENT для «statistical\_indicator\_id» обеспечивает автоматическую генерацию уникальных идентификаторов при добавлении новых записей. FOREIGN KEY(contact\_person\_id) связывает статистический показатель с ответственным за него пользователем. Текстовое поле «comment» оставлено nullable, что позволяет сохранять статистические показатели без комментариев, если они не нужны.

* 1. Разработка хранимых процедур

С точки зрения взаимодействия с базой данных, для сотрудников и клиентов статистического центра наиболее важными являются следующие возможности:

– добавить новую статистическую информацию;

– извлечь какие-нибудь показатели за определенный промежуток времени.

Для реализации этих наиболее важных функций базы данных статистического центра был реализован ряд хранимых процедур и функций. Далее мы опишем эти процедуры.

Процедура для добавления информации о рождении ребенка входит в категорию процедур для добавления новой статистической информации. Данная хранимая процедура принимает на вход:

– дату и время рождения ребенка;

– где родился ребенок;

– дату рождения первого родителя;

– пол первого родителя;

– национальность первого родителя;

– гражданство первого родителя;

– уровень образования первого родителя;

– ученую степень первого родителя;

– находится ли первый родитель в браке;

– дату рождения второго родителя;

– пол второго родителя;

– национальность второго родителя;

– гражданство второго родителя;

– уровень образования второго родителя;

– ученую степень второго родителя;

– находится ли второго родитель в браке.

Эта хранимая процедура создает или находит заданный тип людей для родителей, и создает запись о том, что в заданное время, в заданном месте родился ребенок с определенными родителями. В случае если указаны неизвестное место рождения, пол, национальность, гражданство, уровень образования или ученая степень, то данная процедура сообщает об ошибке.

Процедура для добавления информации о смерти человека входит в категорию процедур для добавления новой статистической информации. Данная хранимая процедура принимает на вход:

– дату и время смерти;

– населенный пункт, в котором умер человек;

– причину смерти;

– дату рождения;

– пол;

– национальность;

– гражданство;

– уровень образования;

– ученую степень;

– находится ли человек в браке;

Эта хранимая процедура создает запись о том, что в заданное время, в заданном месте, по заданной причине умер человек с определенными выше качествами. В случае если указаны неизвестное место смерти, пол, национальность, гражданство, уровень образования или ученая степень, то данная процедура сообщает об ошибке.

Функция для получения количества рожденных детей в заданный промежуток времени на заданной территории входит в категорию функций для извлечения статистической информации.

Данная функция принимает на вход:

– дату начала интересующего промежутка, включительно;

– дату конца интересующего промежутка, исключительно;

– территорию, на которой интересен данный показатель.

Эта функция возвращает количество рожденных детей в заданный промежуток времени на заданной территории. В случае если указана неизвестная территория, то данная функция сообщает об ошибке.

* 1. Разработка триггеров

В рамках разработки базы данных для статистического центра были разработаны ряд триггеров. Данные триггеры можно разделить на несколько групп:

– исправляющие входные данные;

– проверяющие входные данные;

– поддерживающие данные в консистентном состоянии.

В первую группу входят триггеры, которые в входной строке в начале и конце удаляют пробельные символы. Они защищают пользователя от некорректного ввода. Данные триггеры реализуют полезную способность базы данных очищать входные данные от ненужных символов, тем самым поддерживая строки в одном формате.

Во вторую категорию входят триггеры проверяющие входные данные. К ним относятся:

1 Триггер, который проверяет корректность записи о рождении ребенка. А именно, что один родитель является мужчиной, а другой женщиной.

2 Триггер, который проверяет корректность записи о браке. А именно, что один супруг является мужчиной, а другой женщиной.

3 Триггер, который проверяет корректность записи о браке. А именно, что люди, которые развелись, были до этого в браке.

В третью группу входят триггеры, которые поддерживают данные в консистентном состоянии. Сюда входят триггеры, которые изменяются время обновления статистического показателя, при изменении данных, которые формируют данный статистический показатель. Это является важной функцией, так как пользователям необходимо знать, насколько свежая информация присутствует в статистическом центре.

* 1. Разработка индексов

Индекс [14] – объект базы данных, создаваемый с целью повышения производительности поиска данных. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному критерию путём последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет искать строки, удовлетворяющие критерию поиска. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск – например, сбалансированного дерева.

Необходимость создания индексов может вызвать медленная работа запросов при выборке данных. Показать это может только тестирование на больших объёмах и показания при использовании реального приложения с реальными данными.

1. Тестирование базы данных

TODO:

ЗАКЛЮЧЕНИЕ

Использованные источники:

[1] <https://rosstat.gov.ru/mission>

[2] <https://rosoncoweb.ru/standarts/RUSSCO/2018/2018-mkb10.pdf>

[3] https://pravo.by/document/?guid=3871&p0=p30400560

[4] https://www.vagr.by/address-kadastr/

[5] <https://w3.unece.org/PXWeb2015/pxweb/ru/STAT/STAT__30-GE__06-Health/007_ru_GEHECOD_r.px/>

[6] <http://www.eycb.coe.int/compass/ru/chapter_5/5_2.html>

[7] <https://kodeksy-bel.com/dictionary/d/dolzhnost.htm>

[8] <https://01.xn--b1aew.xn--p1ai/document/201357>

[9] <https://www.hse.ru/data/353/323/1234/lect4-5ec2005.pdf>

[10] <https://unece.org/sites/default/files/2023-10/rec20_rev5R.pdf>

[11] <https://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:Microsoft_SQL_Server>

[14] Индексы базы данных [Электронный ресурс]. – Режим доступа: https://www.tutorialspoint.com/postgresql/postgresql\_indexes.htm. – Дата доступа: 15.11.2023.

Тема База данных для статистического центра.

Стандартный набор возможностей: выбрать тему, выбрать время, выбрать страну. Посмотреть данные ввиде таблицы, диаграмм или на карте. Так же есть вкладка метаданные – описывает то, о чем идет речь.

Получить данные за разные период

Метаданные – единица измерения, ответственное управление, ответственный сотрудник, периодичность сбора,

В идеале должна быть возможность добавить новую категорию.

Есть стандарты ISO OECD

Тут будут аналоги

Аналоги:

Информационно-аналитическая система распространения официальной статистической информации национального статистического комитета Республики Беларусь.

Функциональные возможности:

1. Доступ к официальной статистической информации в режиме реального времени;
2. Возможность просмотра и анализа данных в различных разрезах (временных, географических, отраслевых и т.д.);
3. Использование различных инструментов для обработки и анализа данных (графики, таблицы, карты и т.п.);
4. Формирование отчетов по заданным параметрам;
5. Интеграция с другими информационными системами;
6. Обеспечение конфиденциальности и безопасности данных.

The world bank.

1. Цель работы

Целью данной лабораторной работы является проведение анализа интересующей меня областей рынка, нахождения проблемы, которую может решить мой программный продукт, продумать функционал, который будет в данном программном продукте и составить техническое задание.

1. Техническое задание проекта
   1. Общие положения
      1. Термины и определения

ВИ – вариант использования или use case.

ПД – персональные данные.

ПП – программный продукт.

* + 1. Цель

Основной целью является создание программного продукта для учета рабочего времени сотрудников. Данный программный продукт должен решать изложенные дальше задачи.

Помочь рядовому сотруднику четко понимать проекты над которыми он работает и конкретные задачи в рамках этих проектов. Так же он должен наглядно показать сотруднику сколько времени он по итогу тратит на разные задачи или возможно показать, что большую часть рабочего времени он бездельничает.

Сделать процесс управления рабочим временем более эффективным для руководителей, позволяя им быстро анализировать данные о рабочем времени и принимать необходимые решения.

Предоставить возможность анализа данных о рабочем времени с целью повышения эффективности бизнеса, оптимизации рабочих процессов и улучшения ресурсного планирования.

Сократить затраты на административные процессы, связанные с учетом рабочего времени, благодаря автоматизации и оптимизации задач.

Гарантировать защиту конфиденциальности данных о рабочем времени сотрудников и обеспечить соблюдение законодательства.

* + 1. Предполагаемая аудитория

С точки зрения организаций данный программный продукт направлен на малые и средние предприятия. Так как данным организациям может быть нужен простой и доступный продукт для учета рабочего времени своих сотрудников без сложной инфраструктуры.

С точки зрения сотрудников образовательный уровень может сильно варьироваться внутри организации. Программный продукт должен быть простым в использовании и иметь интуитивный интерфейс, чтобы даже пользователи с ограниченным опытом в работе с компьютерами могли легко заполнять данные о своем рабочем времени. В зависимости от образования и навыков сотрудников может потребоваться проведение обучения или подготовительных мероприятий для корректного использования продукта. Предполагается, что руководители достаточно квалифицированы для эффективного использования предоставляемых инструментов для исполнения своих служебных обязанностей.

* + 1. Компоненты системы

Система должна состоять из трёх компонентов: сервера, клиента и клиент-серверной системы управления базой данных.

Сервер осуществляет запросы к базе данных, отправляет необходимые данные клиенту и является одним из уровней защиты персональных данных.

Клиент представляет из себя настольное приложение, которое предоставляет пользователю графический интерфейс для взаимодействия с программным продуктом.

Система управления базой данных и база данных используется для хранения, организации, обеспечения целостности и доступа к данным.

* + 1. Основные функциональные возможности

Основные функциональные возможности для рядовых сотрудников:

* принятие задания от руководителя;
* возможность сообщить о завершении задания руководителю;
* внесение данных о начале и окончании рабочего дня;
* внесение данных о задачах, которые выполнялись во время рабочего дня;
* внесение времени, затраченное на каждое задание во время рабочего дня;
* формирование заявления на отпуск;
* редактирование внесенных данных;
* просмотр своих записей о рабочем времени;
* просмотр отчета о своих задачах и проектах;
* получение напоминаний о заполнении отчета;
* получение напоминаний о задачах, время на выполнение которых скоро закончится;
* получение уведомлений о одобрении/отказе заявки на отпуск;
* получение уведомлений о ближайших командировках;
* получение уведомлений о изменении рабочего графика;
* получение уведомлений о назначении новой задачи;
* получение уведомлений о назначении нового проекта;
* получение уведомлений о изменении задач или проекта.

Основные функциональные возможности для руководителей:

* совпадают с возможностями рядового сотрудника;
* просмотр данных о рабочем времени подчиненных сотрудников;
* просмотр рабочего графика сотрудника;
* создание задач и проектов;
* назначение задач и проектов сотрудникам;
* отслеживание статусов задач и сроков выполнения;
* редактирование, удаление и просмотр проектов и задач;
* настройки и управление графиками работы сотрудников;
* отслеживание сверхурочных работ сотрудников;
* одобрение или отклонения заявления на отпуск;
* создание, редактирование, удаление и просмотр командировок;
* отправление сотрудников в командировку.

Основные функциональные возможности для администраторов:

* совпадают с возможностями руководителя;
* создание, редактирование, удаление, просмотр рядовых сотрудников;
* создание, редактирование, удаление, просмотр руководителей;
* создание, редактирование, удаления, просмотр праздников;
* создание, редактирование, удаление, просмотр должностей;
* создание, редактирование, удаление, просмотр отделов;
* очистка устаревших данных.

Общие функциональные возможности:

* проверка на корректность пользовательского ввода;
* сообщение об ошибках с связью.
  1. Функциональная карта системы

На рисунках 2.1, 2.2, 2.3 представлена функциональная карта ПП с точки зрения сотрудника.

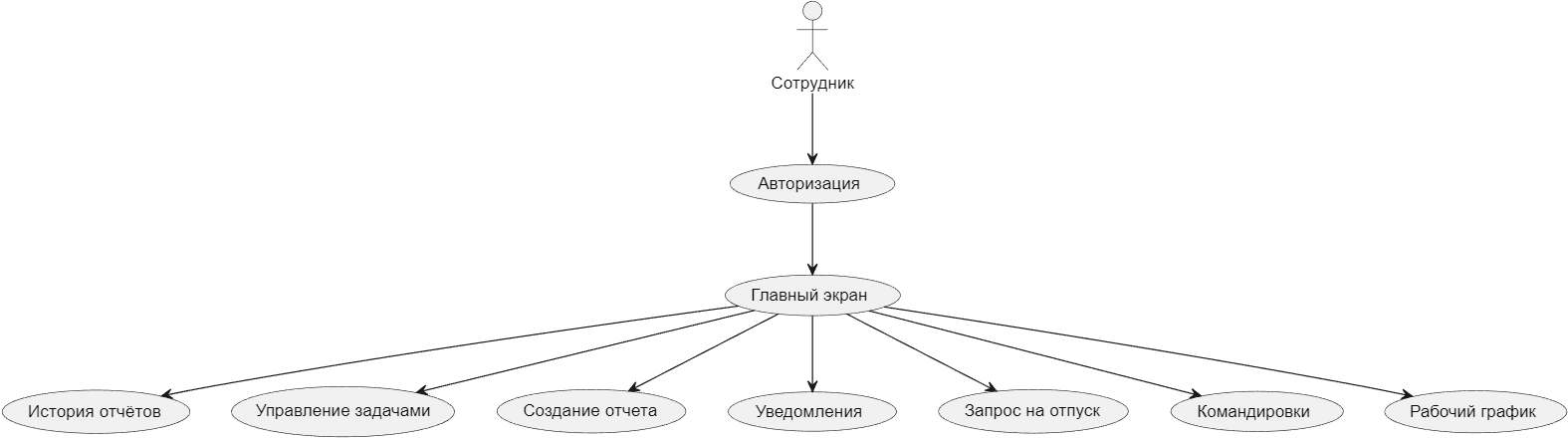


Рисунок 2.1 – Функциональная карта ПП с точки зрения сотрудника (часть 1)

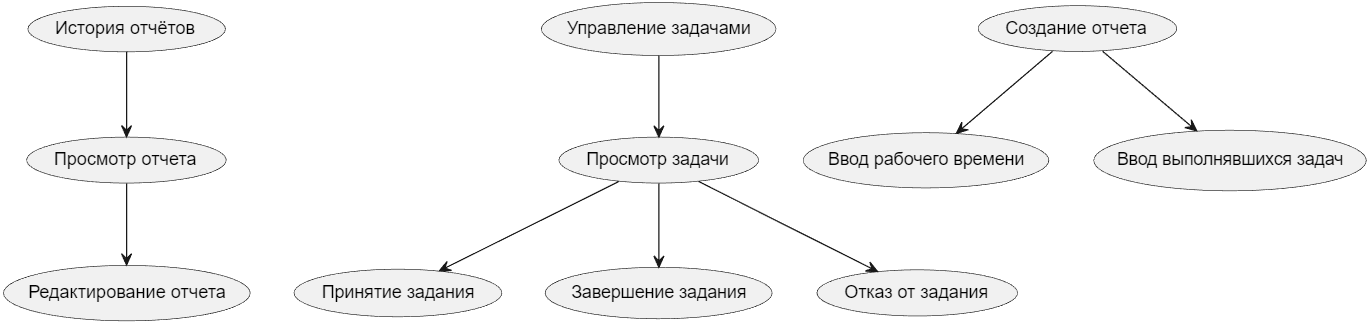


Рисунок 2.2 – Функциональная карта ПП с точки зрения сотрудника (часть 2)

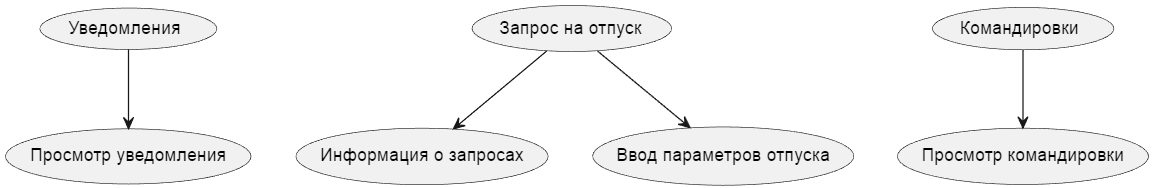


Рисунок 2.3 – Функциональная карта ПП с точки зрения сотрудника (часть 3)

На рисунках 2.4, 2.5, 2.6, 2.7 представлена функциональная карта ПП с точки зрения руководителя.

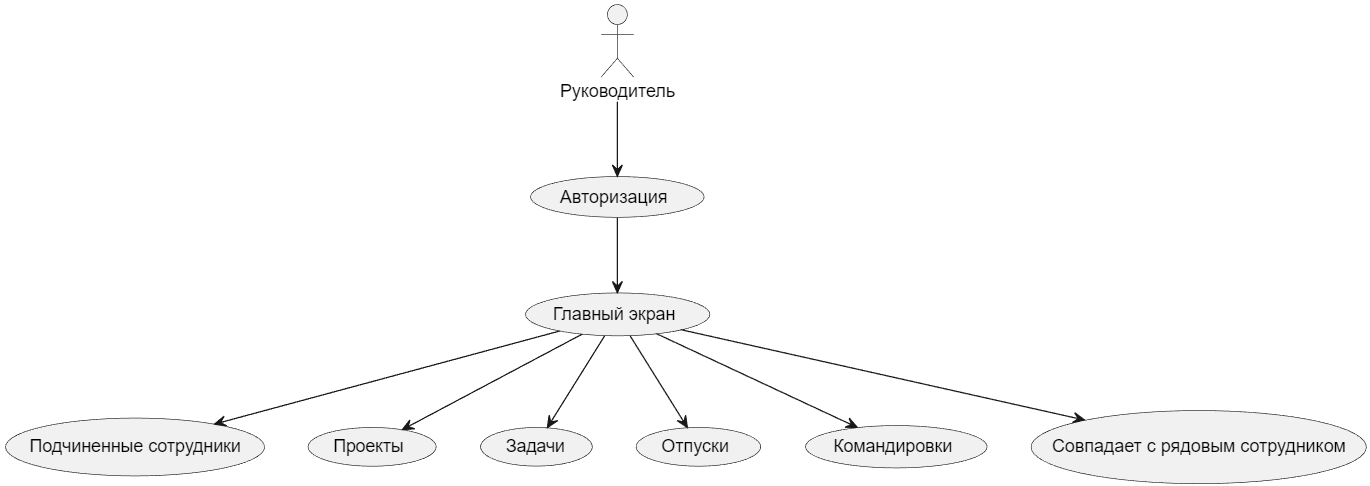


Рисунок 2.4 – Функциональная карта ПП от лица руководителя (часть 1)

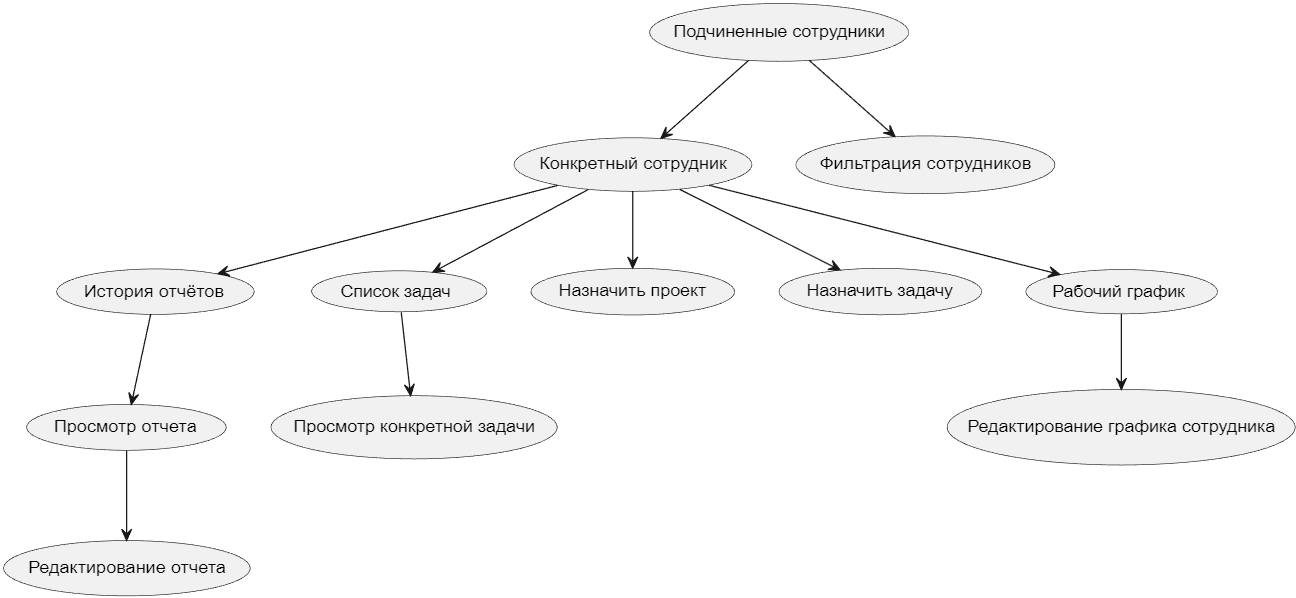


Рисунок 2.5 – Функциональная карта ПП от лица руководителя (часть 2)

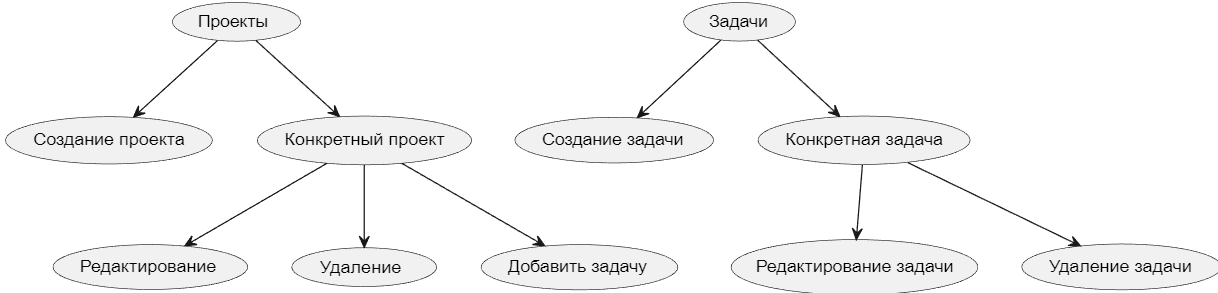


Рисунок 2.6 – Функциональная карта ПП от лица руководителя (часть 3)

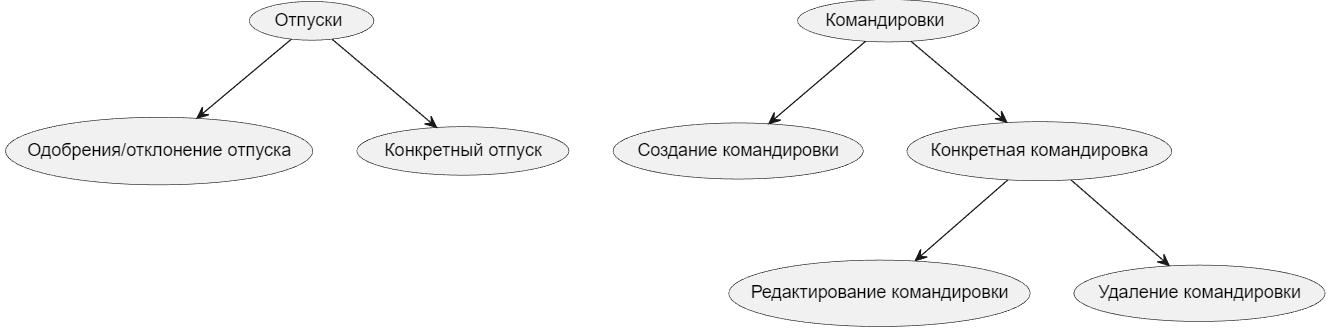


Рисунок 2.7 – Функциональная карта ПП от лица руководителя (часть 4)

На рисунках 2.8, 2.9, 2.10, 2.11 представлена функциональная карта ПП с точки зрения администратора.

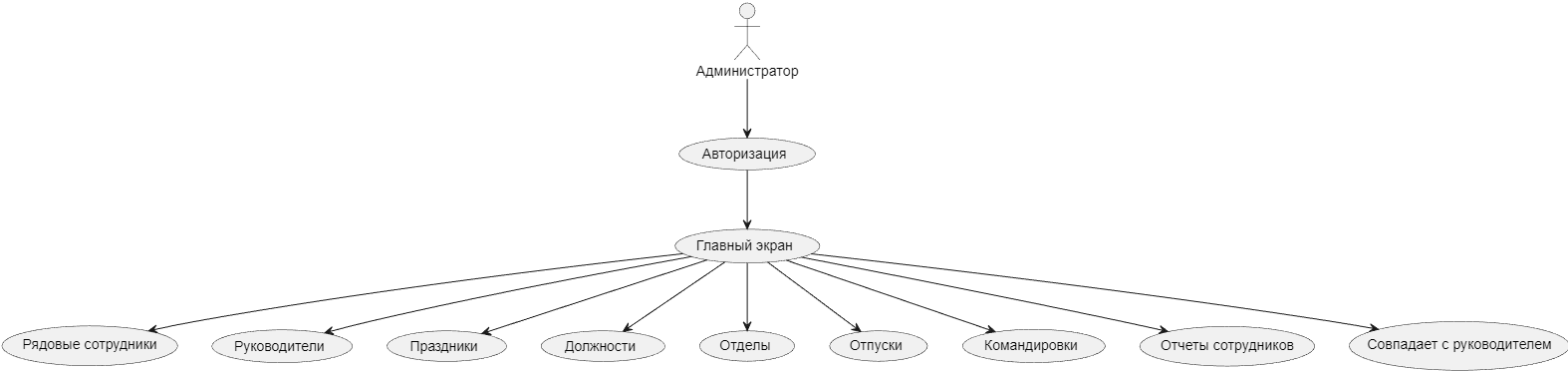


Рисунок 2.8 – Функциональная карта ПП от лица администратора (часть 1)

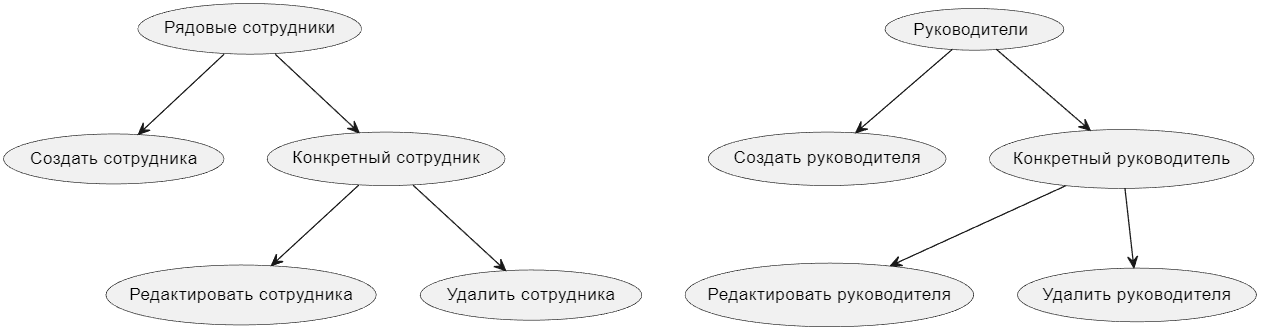


Рисунок 2.9 – Функциональная карта ПП от лица администратора (часть 2)



Рисунок 2.10 – Функциональная карта ПП от лица администратора (часть 3)



Рисунок 2.11 – Функциональная карта ПП от лица администратора (часть 4)

* 1. Модель данных

На диаграмме показаны основные сущности программного продукта и их взаимосвязь между собой. Данную диаграмму можно увидеть на рисунке 2.12.

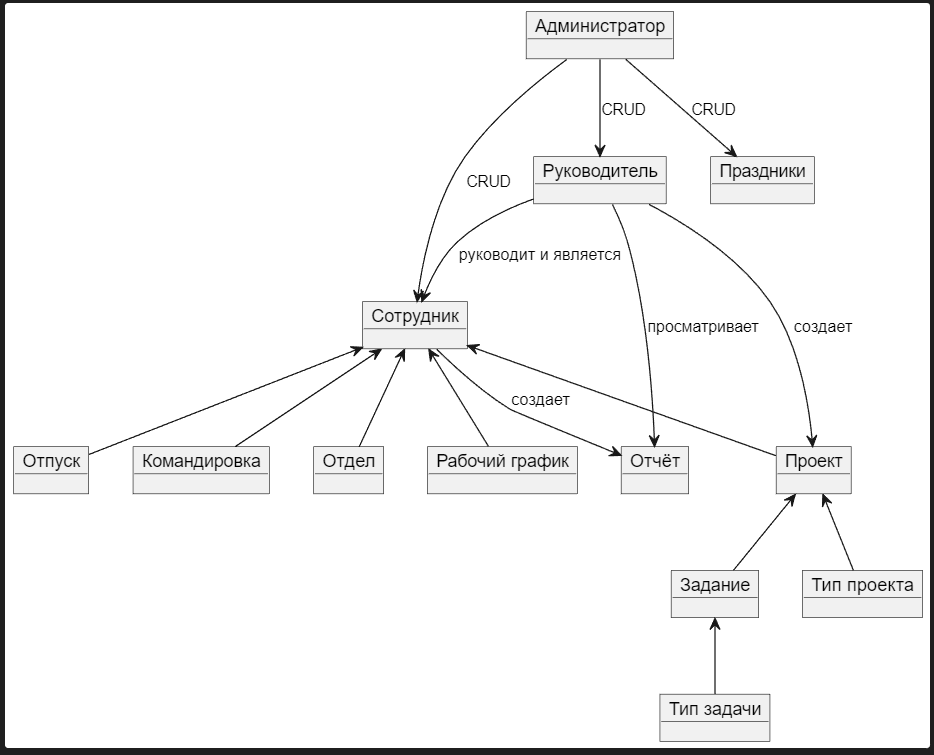


Рисунок 2.12 – Диаграмма с основными сущностями

* 1. Интерфейсы пользователя

На рисунке 2.13 представлена форма для авторизации.

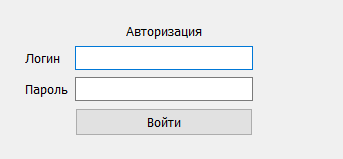


Рисунок 2.13 – Форма для авторизации

На рисунке 2.14 представлено главное окно.

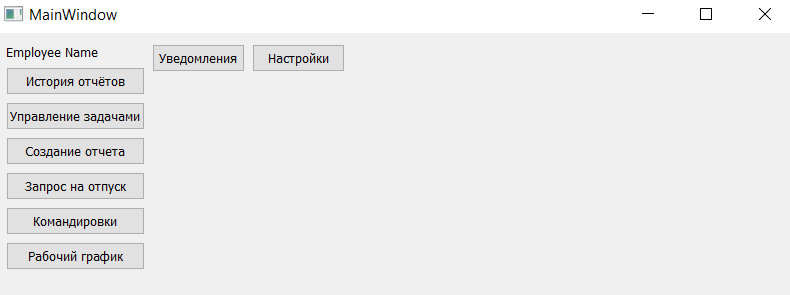


Рисунок 2.14 – Главное окно приложения

На рисунке 2.15 представлена форма для создания отчета.

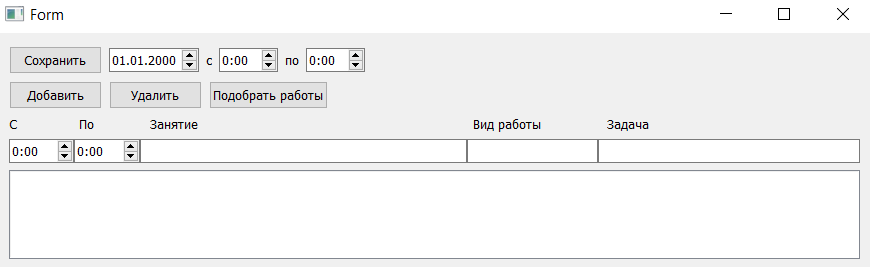


Рисунок 2.15 – Форма для создания отчета

На рисунке 2.16 представлена форма с историей отчетов.

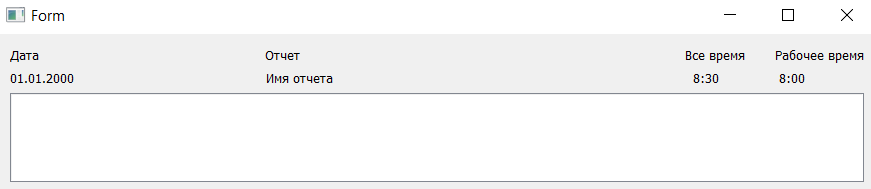


Рисунок 2.16 – Форма с историей отчетов

На рисунке 2.17 представлена форма для управления задачами.

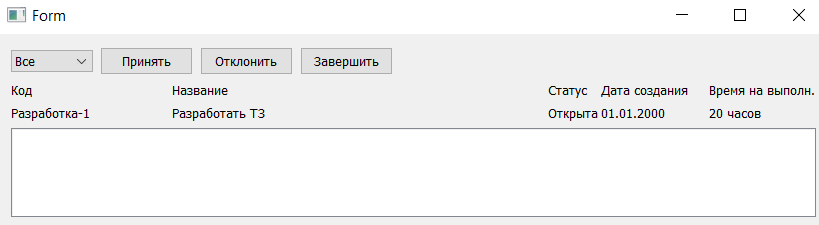


Рисунок 2.17 – Форма для управления задачами

На рисунке 2.18 представлена форма для показа конкретного задания.

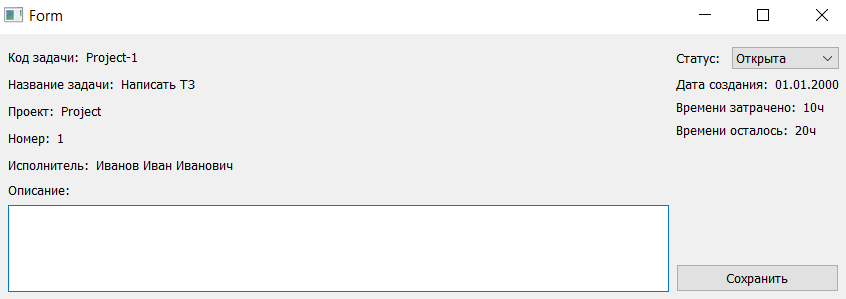


Рисунок 2.18 – Форма для показа конкретного задания

На рисунке 2.19 представлена форма со списком уведомлений.

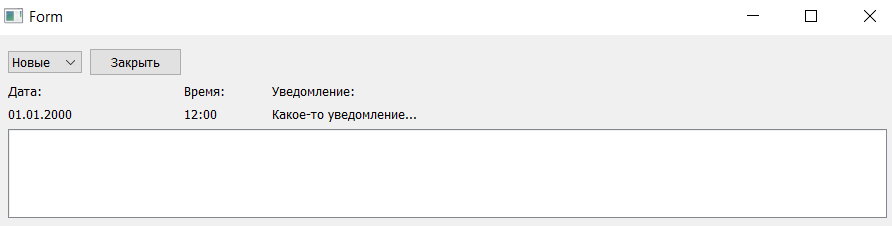


Рисунок 2.19 – Форма со списком уведомлений

На рисунке 2.20 представлена форма с конкретным уведомлением.

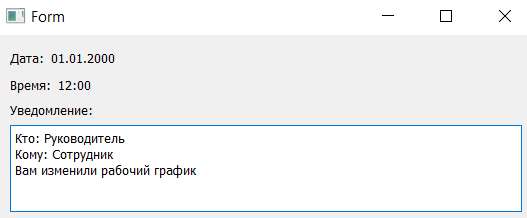


Рисунок 2.20 – Форма с конкретным уведомлением

На рисунке 2.21 представлена форма со списком отпусков.

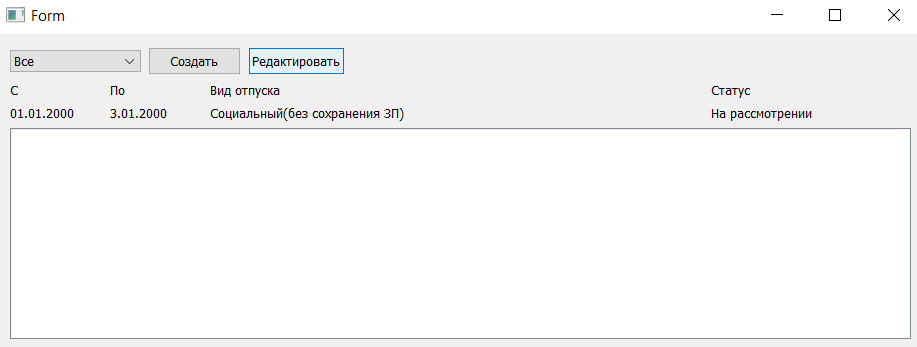


Рисунок 2.21 – Форма со списком отпусков

На рисунке 2.22 представлена форма для создания запроса на отпуск.

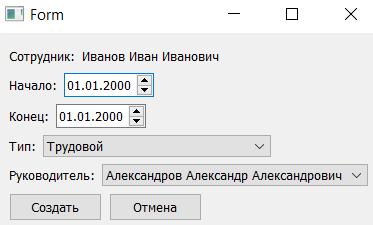


Рисунок 2.22 – Форма для создания запроса на отпуск

На рисунке 2.23 представлена форма со списком командировок.

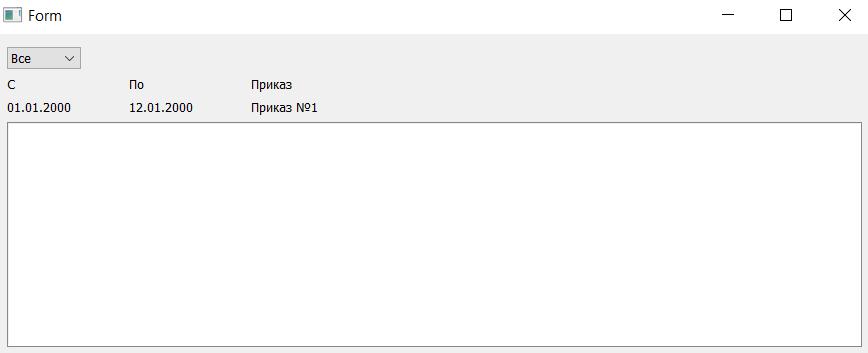


Рисунок 2.23– Форма со списком командировок

На рисунке 2.24 представлена форма с информацией о конкретной командировке.

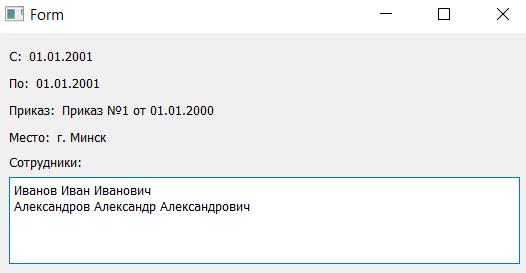


Рисунок 2.24 – Форма с информацией о конкретной командировке

На рисунке 2.25 представлена форма с информацией о рабочем графике.

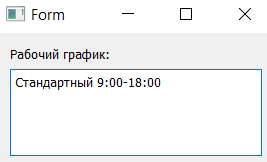


Рисунок 2.25 – Форма с информацией о рабочем графике

* 1. Нефункциональные требования
     1. Требования к техническому обеспечению

Программный продут должен быть кросс-платформенным и работать под операционными системами Windows 10 и Linux Mint. Разработка должна вестись с использованием фреймворка Qt 5.

* + 1. Требования к безопасности

Данный ПП должен обеспечивать необходимые механизмы технической защиты для соблюдения закона Республики Беларусь от 07.05.2021 № 99-З «О защите персональных данных» в целях обеспечения защиты прав и свобод физических лиц при обработке их ПД.

Система не должна позволять не администраторам получать доступ к интерфейсу администратора. Система не должна позволять не руководителям получать доступ к интерфейсу руководителя.

Система не должна позволять доступ к данным пользователям, не имеющим прав на их просмотр.

Пароли должны храниться в зашифрованном виде. Длина пароля должна быть не менее восьми символов.

Должна быть базовая защита от sql-инъекций.

Предполагается, что будет как минимум один администратор способный сделать резервную копию базы данных.

База данных должна быть клиент-серверной, для возможности развертывания базы данных у сторонней компании, которая может обеспечить безопасность персональных данных в соответствии с законодательством.

* + 1. Требования к производительности

Система должна стабильно работать с глубиной истории не менее 1 года.

Самый тяжелый запрос должен выполнятся не больше пяти секунд.