МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Базы данных»

Тема «Реализация базы данных железнодорожного вокзала с использованием средств мониторинга состояния СУБД»

**Исполнитель**

студент 2 курса 7 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. В. Тимошенко

подпись, дата

**Руководитель**

Ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кантарович В. С.

должность, учен. степень, ученое звание подпись, дата

Допущен к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ Кантарович В. С.

подпись дата инициалы и фамилия

Минск 2022

**Содержание**

[**Введение** 3](#_Toc130492348)

[**1. Аналитический обзор литературы по теме проекта** 4](#_Toc130492349)

[**1.1 Сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих по данной тематике технических научных решений** 4](#_Toc130492350)

[**1.2 Пример отечественных и зарубежных аналогов** 5](#_Toc130492351)

[**2. Изучение требований, определение вариантов использования** 6](#_Toc130492352)

[**2.1 Определение основных функциональных требований к базе данных** 6](#_Toc130492353)

[**2.2 Определение нефункциональных требований к базе данных** 6](#_Toc130492354)

[**2.3 Определение вариантов использования** 7](#_Toc130492355)

[**2.4 Вывод** 7](#_Toc130492356)

[**3. Анализ и проектирование модели базы данных. Описание информационных объектов и ограничений целостности** 8](#_Toc130492357)

[**3.1 Проектирование базы данных** 8](#_Toc130492358)

[**3.1.1 Таблица Points** 8](#_Toc130492359)

[**3.1.2 Таблица Routes** 9](#_Toc130492360)

[**3.1.3 Таблица Schedule** 9](#_Toc130492361)

[**3.1.4 Таблица Trains** 9](#_Toc130492362)

[**3.1.5 Таблица Vans** 10](#_Toc130492363)

[**3.1.6 Таблица Passengers** 10](#_Toc130492364)

[**3.1.7 Таблица Payments** 10](#_Toc130492365)

[**3.1.8 Таблица Tickets** 10](#_Toc130492366)

[**Заключение** 12](#_Toc130492367)

[**Список использованных литературных источников** 13](#_Toc130492368)

**Введение**

В нашем современном мире имеется огромный спрос пользователей на хранилища информации, также называемые базами данных, которые могли бы постоянно работать и позволять удобно пользоваться хранимой инфромацией, также необходимо учитывать необходимость в использовании средств мониторинга состояния хранилища данных. Особенно важным становится использование баз данных в железнодорожной отрасли для обработки и хранения данных о состоянии железнодорожных вокзалов и поездов.

В связи с актуальностью данной работы целью курсового проекта является разработка и реализация базы данных Oracle и интерфейса для неё.

К задачам курсового проекта относится: аналтический обзор литературы по теме проекта, изучение требований; определение вариантов использования; анализ и проектирование модели данных; описание информационных объектов и ограничений целостности; создание необходимых объектов; импорт и экспорт данных; описание требуемой технологии; тестирование производительности; формирование вывода по каждому разделу; заключение, включающее вывод по проделанной работе.

Для базы данных обычно требуется комплексное программное обеспечение, которое называется системой управления базами данных (СУБД). СУБД служит интерфейсом между базой данных и пользователями или программами, предоставляя пользователям возможность получать и обновлять информацию, а также управлять ее упорядочением и оптимизацией. СУБД обеспечивает контроль и управление данными, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка, а также резервное копирование и восстановление.

Спектр применения систем управления базами данных на сегодняшний день практически необъятен – базы данных используются в интернете, в производстве, в промышленности, в маркетинге, в мобильных устройствах, в финансовой и банковской сферах, на телевидении, в телекоммуникациях и рекламе.

**1. Аналитический обзор литературы по теме проекта**

**1.1 Сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих по данной тематике технических научных решений**

Существует множество методов решения этой задачи, которые могут быть классифицированы по следующим параметрам:

1. Производительность: скорость выполнения запросов и эффективность использования ресурсов.
2. Надежность: способность системы работать без сбоев и сбоев, а также ее способность восстановиться после сбоя.
3. Гибкость: способность системы адаптироваться к изменениям требований и удовлетворять различным потребностям пользователей.
4. Расширяемость: возможность системы расширяться и добавлять новые функциональные возможности.

В теоретической литературе активно обсуждаются различные модели данных, в том числе реляционная, иерархическая и сетевая модели. Однако в контексте реализации базы данных железнодорожного вокзала реляционная модель является наиболее предпочтительной, так как она хорошо подходит для описания связей между различными элементами информационной системы, такими как станции, поезда, расписания, билеты и т.д.

Реляционная модель базы данных – это стандарт, который используется для описания и управления данными в большинстве современных баз данных. Для реализации базы данных железнодорожного вокзала можно использовать различные СУБД, такие как Oracle, Microsoft SQL Server, PostgreSQL и другие.

В литературе также описываются различные методы обработки данных, такие как SQL-запросы и NoSQL-базы данных. В контексте данной темы SQL-запросы являются более предпочтительными, так как они хорошо подходят для организации транзакций, которые важны для системы бронирования билетов и работы с расписаниями.

Для реализации базы данных железнодорожного вокзала с использованием средств мониторинга состояния СУБД можно использовать специализированные программные средства, такие как системы мониторинга и управления базами данных. Одним из примеров такой системы является Nagios, которая позволяет мониторить состояние базы данных, отслеживать ошибки и предупреждать об их возникновении.

Другой подход - использование инструментов профилирования кода, которые позволяют выявить узкие места в работе приложения и оптимизировать его.

Среди существующих технических решений можно выделить информационную систему "Железнодорожный транспорт", разработанную компанией "ООО Интерконтиненталь". Система предназначена для автоматизации работы железнодорожного транспорта и включает в себя модуль управления железнодорожными станциями и вокзалами, а также модуль мониторинга и управления базами данных.

Другим примером является система управления железнодорожным транспортом "ТРАНСДИЗЕЛЬ", разработанная ООО "Трансдизель". Система включает в себя модули управления поездами, грузами, вагонами и другими ресурсами железнодорожного транспорта. В системе используется СУБД PostgreSQL, а для мониторинга состояния базы данных применяется средство мониторинга pgBadger.

**1.2 Пример отечественных и зарубежных аналогов**

Из примеров успешной реализации мониторинга СУБД для железнодорожных систем является проект "Система мониторинга энергопотребления на железнодорожном транспорте" разработанный в России на базе БД Oracle[1]. В данном проекте были решены задачи сбора, хранения и анализа данных по энергопотреблению на железнодорожных объектах, таких как вокзалы, станции, электроподстанции и др. Реализация системы осуществляется через использование сети сбора данных, оборудования для их сбора и передачи на сервера базы данных, а также средств анализа и визуализации полученных данных.

Еще одним примером является проект "Управление железнодорожным транспортом на основе мониторинга движения поездов" разработанный в Китае. В рамках данного проекта была разработана система мониторинга движения поездов, которая позволяет отслеживать расписание движения, скорость движения, статус оборудования на поездах и многое другое. Система использует технологию GPS для определения местоположения поездов, а также средства мониторинга состояния СУБД для обработки и анализа полученных данных.

**1.3 Вывод**

В заключение можно отметить, что реализация базы данных железнодорожного вокзала с использованием средств мониторинга состояния СУБД является актуальной и перспективной задачей, которая может быть решена с помощью различных технических решений и инструментов, такие как Oracle и PostgreSQL, а для мониторинга состояния базы данных – различные средства мониторинга, такие как Oracle Enterprise Manager, Nagios и pgBadger. Однако при выборе конкретного подхода необходимо учитывать особенности конкретной задачи и требования к системе, такие как надежность, масштабируемость и производительность, а также проводить сравнительный анализ существующих решений и аналогов.

**2. Изучение требований, определение вариантов использования**

**2.1 Определение основных функциональных требований к базе данных**

В ходе изучения требований было выявлено, что база данных железнодорожного вокзала должна включать информацию о поездах, вагонах, расписании движения, билетах и о пассажирах. Были определены следующие варианты использования:

1. Хранение информации о расписании прибытия и отправления поездов на определенную дату и время.
2. Предоставление информации о расписании движения поездов пассажирам.
3. Возможность просмотра и редактирования информации о поездах, включая название, маршрут, время отправления и прибытия, количество свободных мест.
4. Хранение информации о билетах на поезда, включая стоимость, маршрут, дату и время отправления, место в вагоне.
5. Возможность продажи билетов на поезда через систему онлайн-бронирования.
6. Ведение учета проданных билетов и свободных мест на каждый поезд.
7. Хранение информации о пассажирах, включая имя, фамилию, номер паспорта, контактную информацию и предоставляемые льготы.
8. Обеспечение защиты данных и доступа только авторизованных пользователей к базе данных.
9. Мониторинг состояния базы данных и обнаружение проблем в работе СУБД, включая информацию о возникающих ошибках и неполадках, о времени отклика системы, а также о загрузке процессора и объеме свободного места на диске.

**2.2 Определение нефункциональных требований к базе данных**

В качестве нефункциональных требований, можно выделить:

1. Безопасность: База данных должна обеспечивать безопасность данных и защиту от несанкционированного доступа, в том числе использование механизмов авторизации и аутентификации.
2. Производительность: База данных должна иметь высокую производительность и обеспечивать быстрый доступ к данным для обеспечения эффективной работы вокзала.
3. Надежность: База данных должна быть надежной и стабильной, обеспечивать целостность данных и иметь возможность быстрого восстановления после сбоев.
4. Масштабируемость: База данных должна быть масштабируемой и гибкой, чтобы обеспечивать эффективное управление растущим объемом данных вокзала.
5. Удобство использования: База данных должна быть удобной и простой в использовании для обеспечения эффективной работы пользователей.

**2.3 Определение вариантов использования**

Вариантами использования базы данных являются:

1. Просмотр расписания отправления и прибытия поездов;
2. Просмотр информации о наличии мест на поездах;
3. Бронирование мест на поезд;
4. Просмотр информации о задержках и отменах поездов;
5. Просмотр истории покупок билетов;

**2.4 Вывод**

Для определения функиональных требований были рассмотрены варианты использования, которые позволяют эффективно применять СУБД для решения задач, схожих с поставленной. Были определены нефункциональные варианты использования, необходимые качественному продукту.

Таким образом, изучение требований и определение вариантов использования позволяет определить основные функциональные и нефункциональные требования к базе данных, а также выявить возможные варианты использования системы.

**3. Анализ и проектирование модели базы данных. Описание информационных объектов и ограничений целостности**

**3.1 Проектирование базы данных**

Для разработки веб-приложения в рамках курсового проекта понадобилась база данных с необходимой конфигурацией сущностей для хранения всей необходимой информации.

Структуру базу данных, ограничения целостности, связи и поля можно увидеть в приложении **!!!!!!!** и на рисунке 3.1.

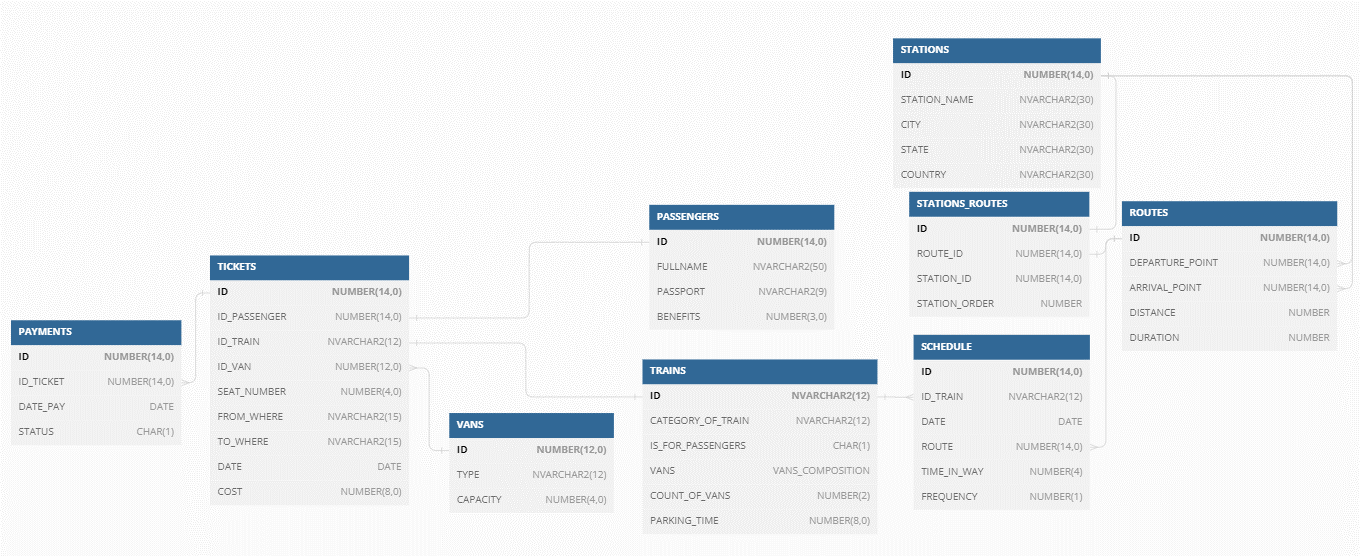


Рисунок 3.1 – Структура базы данных

В базе данных существует 8 таблиц, каждая из которых выполняет свою функцию.

### **3.1.1 Таблица Stations**

Данная таблица используется для хранения информации о станциях остановки составов.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждой точки остановки.
* Cтолбец STATION\_NAME. Хранит название станции.
* Cтолбец CITY. Хранит название города, в котором расположена станция.
* Cтолбец STATE. Хранит название региона, в котором расположена станция.
* Cтолбец COUNTRY. Хранит название страны, в котором расположена станция.

Данная таблица содержит ограничение перичного ключа для столбца ID, а также все столбцы должны содержать какое-либо значение.

### **3.1.2 Таблица Routes**

Данная таблица используется для хранения информации о маршрутах передвижения составов.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого маршрута.
* Cтолбец DEPARTURE\_POINT. Хранит ID начальной точки.
* Cтолбец ARRIVAL\_POINT. Хранит ID конечной точки.
* Cтолбец DISTANCE. Хранит расстояние, проходящее составом.
* Cтолбец DURATION. Хранит время, проводимое в пути.

Данная таблица содержит ограничение перичного ключа для столбца ID. Также для столбцов DEPARTURE\_POINT и ARRIVAL\_POINT существуют ограничения целостности по внешнему ключу к таблице STATIONS.

### **3.1.3 Таблица Schedule**

Данная таблица используется для хранения информации о расписании передвижения составов.

В её состав входят следущие столбцы:

* cтолбец ID. Хранит уникальный идентификатор каждого маршрута.
* Cтолбец ID\_TRAIN. Хранит ID поезда, зарезервированного для маршрута.
* Cтолбец DATE. Хранит дату начала выдвижения состава по маршруту.
* Cтолбец ROUTE. Хранит ID маршрута передвижения состава.
* Cтолбец TIME\_IN\_WAY. Хранит время состава в пути.
* Cтолбец FREQUENCY. Хранит частоту использования данной ячейки расписания.

Данная таблица содержит ограничение перичного ключа для столбца ID. Также для столбцов ID\_TRAIN и ROUTE существуют ограничения целостности по внешнему ключу к таблице TRAINS и ROUTES соответсвенно.

### **3.1.4 Таблица Trains**

Данная таблица используется для хранения информации о поездах(составах).

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого состава.
* Cтолбец CATEGORY\_OF\_TRAIN. Хранит категорию данного состава.
* Cтолбец IS\_FOR\_PASSENGERS. Хранит символ, обозначающий возможность использования данного состава для перевозки пассажиров.
* Cтолбец VANS. Хранит массив ID вагонов, из которых состоит поезд.
* Cтолбец COUNTS\_OF\_VANS. Хранит количество вагонов в составе.
* Cтолбец PARKING\_TIME. Хранит время стоянки состава.

Данная таблица содержит ограничение перичного ключа для столбца ID.

### **3.1.5 Таблица Vans**

Данная таблица используется для хранения информации о вагонах.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого вагона.
* Cтолбец TYPE. Хранит тип вагона.
* Cтолбец CAPACITY. Хранит вместимость вагона.

Данная таблица содержит ограничение перичного ключа для столбца ID.

### **3.1.6 Таблица Passengers**

Данная таблица используется для хранения информации о пассажирах.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого пассажира.
* Cтолбец FULLNAME. Хранит ФИО пассажира.
* Cтолбец PASSPORT. Хранит паспортные данные пассажира.
* Cтолбец BENEFITS. Хранит льготы, применяемые к пассажиру.

Данная таблица содержит ограничение перичного ключа для столбца ID.

### **3.1.7 Таблица Payments**

Данная таблица используется для хранения информации о платежах.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого платежа.
* Cтолбец ID\_TICKET. Хранит ID билета, по которому произошла оплата.
* Cтолбец DATE\_PAY. Хранит дату оплаты.
* Cтолбец STATUS. Хранит статус оплаты.

Данная таблица содержит ограничение перичного ключа для столбца ID. Также для столбцов ID\_TICKET существует ограничение целостности по внешнему ключу к таблице TICKETS.

### **3.1.8 Таблица Tickets**

Данная таблица используется для хранения информации о билетах.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждого билета.
* Cтолбец ID\_PASSENGER. Хранит ID пассажира, которому был выдан билет.
* Cтолбец ID\_TRAIN. Хранит ID поезда, на который забронирован билет.
* Cтолбец ID\_VAN. Хранит ID вагона, в котором находится арендованное место.
* Cтолбец SEAT\_NUMBER. Хранит номер арендованного места.
* Cтолбец FROM\_WHERE. Хранит место посадки на поезд.
* Cтолбец TO\_WHERE. Хранит место выхода из поезда.
* Cтолбец DATE. Хранит дату поездки.
* Cтолбец COST. Хранит стоимость билета.

Данная таблица содержит ограничение перичного ключа для столбца ID. Также для столбцов ID\_PASSENGER, ID\_TRAIN и ID\_VAN существуют ограничения целостности по внешнему ключу к таблице PASSENGERS, TRAINS и VANS соответсвенно.

### **3.1.9 Таблица Stations\_routes**

Данная таблица используется для хранения информации о билетах.

В её состав входят следущие столбцы:

* Столбец ID. Хранит уникальный идентификатор каждой станции в маршруте.
* Cтолбец ROUTE\_ID. Хранит ID маршрута, в котором привязана станция.
* Cтолбец STATION\_ID. Хранит ID станции, которая используется в маршруте.
* Cтолбец STATION\_ORDER. Хранит номер станции в маршруте.

Данная таблица содержит ограничение перичного ключа для столбца ID. Также для столбцов ROUTE\_ID, STATION\_ID существуют ограничения целостности по внешнему ключу к таблице ROUTES, STATIONS соответсвенно.

**3.2 Вывод**

В ходе анализа базы данных необходимо была составлена UML диаграмма базы данных, а также описана каждая таблица в диаграмме.

Таким образом, изучение требований и определение вариантов использования позволяет определить основные функциональные и нефункциональные требования к базе данных, а также выявить возможные варианты использования системы.

**4. Описание процедур импорта и экспорта. Описание технологии**

**4.1 Описание процедур импорта и экспорта**

Процедуры импорта и экспорта являются важными инструментами для перемещения данных между различными базами данных или для создания резервных копий базы данных.

JSON (JavaScript Object Notation) – это легкий формат обмена данными, который используется для представления структурированных данных в виде пар "имя/значение". Он широко используется в веб-приложениях и мобильных приложениях, а также в базах данных.

Использование формата JSON для импорта и экспорта данных предоставляет удобный и гибкий способ обмена данными между различными базами данных или веб-приложениями. Он позволяет представлять структурированные данные в виде пар "имя/значение" и легко преобразовывать данные в различные форматы, такие как XML или CSV.

Процедуры импорта и экспорта данных могут быть полезны для создания резервных копий базы данных, для переноса данных между различными окружениями или для обмена данными с другими приложениями. Они также обеспечивают удобный способ загрузки данных из внешних источников и экспорта данных для анализа и обработки в других приложениях.

В целом, процедуры импорта и экспорта данных являются важными инструментами для обеспечения эффективной работы базы данных и обеспечения ее надежности.

**4.2 Описание технологии**

Мониторинг состояния СУБД вляется важным инструментом для обеспечения высокой производительности и надежности базы данных. Он позволяет анализировать работу базы данных в реальном времени и выявлять проблемы, которые могут влиять на ее работу.

В данном проекте используется Oracle Enterprise Manager (OEM), который предоставляет средства мониторинга, управления и администрирования баз данных Oracle. OEM обеспечивает централизованное управление всеми аспектами работы базы данных, включая аппаратное обеспечение, операционную систему, СУБД и приложения.

Одной из главных функций OEM является мониторинг производительности базы данных. Он позволяет анализировать данные о нагрузке на базу данных, использовании ресурсов и производительности запросов. OEM также предоставляет возможность установки и настройки предупреждений и автоматических действий в случае возникновения проблем с производительностью.

Кроме того, OEM обеспечивает механизмы автоматического обнаружения и устранения проблем в работе базы данных. Он позволяет анализировать журналы ошибок, настраивать систему резервного копирования, мониторить доступность базы данных и многое другое.

Таким образом, использование Oracle Enterprise Manager в проекте "Реализация базы данных железнодорожного вокзала с использованием средств мониторинга состояния СУБД" обеспечивает высокую производительность и надежность базы данных, а также облегчает ее администрирование и управление.

**4.3 Вывод**

В данном разделе были описаны процедуры импорта и экспорта из JSON. Использование формата JSON для импорта и экспорта данных позволяет удобно и гибко обмениваться данными между различными приложениями и базами данных, что может упростить работу с данными и повысить эффективность работы в целом. Также была рассмотрена используемая технология.

# **Заключение**

**Список использованных литературных источников**

1. Жуков А. А. Мониторинг баз данных [Электронный ресурс] / Справочник Режим доступа:

СПб.: БХВ-Петербург, 2010. – 320 с. – Дата доступа 10.03.2023.

1. Статья "Development of Railway Station Management Information System Based on Internet of Things" (J. Lin, X. Zhang, H. Shao, C. Jiang, L. Yu, 2019) опубликованная в журнале "IEEE Access" – Дата доступа 11.03.2023.