**Аннотация**

В дипломной работе рассмотрены вопросы анализа предметной области, определена цель автоматизации, проведен обзор аналогов, сформулирована потребность в разработке собственной автоматизированной системы (ПИС). В пояснительной записке представлены методологии и технологии проекта, структура и архитектура автоматизированной системы, результаты проектирования компонентов ПИС, мероприятий по защите данных.

Пояснительная записка дипломной работы содержит 39 страниц, в том числе 6 приложений, 18 таблиц, 6 рисунков.

**Содержание**

Введение 5

1 Анализ требований к программно-информационной системе (ПИС) 6

1.1 Описание объекта автоматизации 6

1.1.1 Структура и функциональная модель предприятия 7

1.1.2 Цель проекта 8

1.1.3 Описание архитектуры проекта ПИС на основе методики Захмана 9

1.2 Обзор аналогов 10

1.3 Требования к ПИС 12

1.3.1 Методологии и технологии проекта 12

1.3.2 Структура и архитектура ПИС 13

2 Моделирование компонентов ПИС 17

2.1 Выбор инструментальных средств 17

2.2 Формирование функциональной составляющей ПИС 21

2.3 Проектирование информационного обеспечения ПИС 22

2.3.1 Внешний уровень архитектуры БД 22

2.3.1.1 Формализованное описание предметной области 22

2.3.1.2 Пользователи ПИС 24

2.3.2 Концептуальный уровень архитектуры БД 26

2.3.2.1 Информационно-логическая модель предметной области 26

2.3.2.2 Даталогическая модель БД 27

2.3.3 Внутренний уровень архитектуры БД 27

2.3.4 Макеты экранных форм для ввода и вывода информации 28

3 Реализация мероприятий по защите данных 32

3.1 Технология создания объектов БД. Поддержка целостности данных 32

3.2 Назначение прав доступа пользователям ПИС 32

3.3 Резервное копирование и восстановление БД 33

Заключение 34

Список использованных источников 35

Приложение А (обязательное) Контекстная диаграмма в нотации DFD 36

Приложение Б (обязательное) Контекстная диаграмма в нотации IDEF0 37

Приложение В (обязательное) Диаграмма 1-го уровня в нотации IDEF0 38

Приложение Г (обязательное) ER-диаграмма 39

Приложение Д (обязательное) Даталогическая модель БД 30

Приложение Е (обязательное) SQL-скрипты 41

**Введение**

Целью дипломной работы является проектирование компонентов программно-информационной системы (ПИС) «Моделирование движения городского пассажирского транспорта».

Программно-информационная система предназначена для расчёта и анализа нагрузки на транспортную сеть – выявление критический моментов (длительного ожидания общественного транспорта на остановочном пункте) соответствия кол-во автобусов пассажирообороту на заданных маршрутах.

Для достижения цели автоматизации необходимо выполнить следующие задачи:

- провести анализ предметной области и определить назначение разработки ПИС;

- выполнить обзор аналогов, сделать вывод о необходимости собственной разработки;

- определить методологии и технологии проекта, структуру и архитектуру автоматизированной системы;

- разработать проекты компонентов ПИС: функциональной составляющей, внемашинного информационно обеспечения – сформировать информационно-логическую модель предметной области, даталогическую модель реляционной базы данных, макеты экранных форм для ввода и вывода информации;

- определить состав и уровни доступа персонала системы, предложить проектные решения по их реализации;

- осуществить тестирование ряда полученных проектных решений.

Дипломная работа выполнена в рамках исследований выпускной квалификационной работы бакалавра.

**Анализ требований к программно-информационной системе (ПИС)**

**1.1 Описание объекта автоматизации**

Отдел управления транспортом в определённом населённом пункте (далее НП) вносит изменения в транспортную сеть города. НП состоит из структурных единиц, о которых необходимо знать: код, название, номер региона, координаты города (далее все координаты несут в себе значение широты и долготы).

Далее, реквизиты всех значимых информационных объектов предметной области будем указывать в скобках.

Транспортная сеть состоит из остановочных пунктов (далее ОП) (код, номер НП, название, координаты), маршрутов (код, номер НП, название, интервал движения автобусов в минутах, количество транспорта на маршруте). Каждому маршруту должен быть присвоен тип транспортного средства (далее ТС) (код, название, вместимость, описание), также к каждому маршруту должны относиться ОП, через которые проходит данный маршрут.

Отдел управления транспортом решает внести изменения в транспортную сеть города на основе аналитических данных, откуда возникает потребность в автоматизации процесса формировании и анализа расчёта нагрузки транспортной сети города. Руководителем отдела принято решение, что для решения этой задачи необходимо использовать программно-информационную систему, которая может быть приобретена на рынке программных продуктов, или специально разработана.

1.1.1 Структура и функциональная модель предприятия

Структура автоматизируемой системы с управлением включает в себя следующие элементы [1]:

1 Система управления:

- работник отдела (роль оператора ПИС) осуществляет прием сведений о транспортной сети города;

- руководитель отдела (роль руководителя, лица, принимающего решение) работает со сформированным отчётом и результатами анализа.

2 Объект управления (автоматизации): сведения о результатах расчёта нагрузки.

3 Информационные потоки: из внешней среды в отдел, локальные (внутренние) распорядительные документы отдела, сведения о состоянии объекта управления представлены на рисунке 1 [2 ,3]. Номера документов на рисунке 1 соответствуют номерам, указанным в таблице 1.

1,2,3,4,5,6,7

Руководитель отдела

Работник отдела

8

9

Рисунок 1 – Структура системы управления. Информационные потоки предприятия

Таблица 1 – Входные и выходные документы предприятия в рамках проекта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер, название документа/ фрагмента документа | Вид документ | Качественные реквизиты | Количественные показатели | Прина  длежн  ость к  инфор  мацион  ному  потоку  (ИП) |
| 1. Сведения о НП | Входной | Название, названии координат | Значение координат широты и долготы НП | ИП1 |
| 1. Типы ТС | Входной | Название, описание | Вместимость ТС | ИП1 |
| 1. Единицы измерения | Входной | Название, кр Название |  | ИП1 |
| 1. Маршруты | Входной | Номер, название | Интервал движения, кол-во транспорта на маршруте | ИП1 (ИП3) |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1. Остановочные пункты | Входной | Номер, название | Координаты широты и долготы ОП | ИП1 (ИП3) |
| 1. Расстояние между ОП | Входной | Номер документа | Номер первого ОП, номер второго ОП, расстояние между ними | ИП1 |
| 1. Связь маршрута и остановки | Входной | Номер документа | Номер маршрута, номер остановки | ИП1 (ИП3) |
| 1. Кол-во людей на остановке в заданный момент времени | Входной | Номер | Номер ОП, кол-во людей | ИП1 (ИП3) |
| 1. Отчёт | Выходной | Название | Номер | ИП2 |

На основе проведенного анализа деятельности сотрудников организации определена существующая в предметной области последовательность этапов обработки информации. Эти результаты отражают точку зрения руководителей разных уровней, являются основой для формирования функциональной модели процесса и содержат следующий перечень выполняемых операций по обработке данных (таблица 1):

1 Обработка внешних документов (1, 2, 3, 4, 5, 6, 7).

2 Обработка локальных документов предприятия (8).

3 Формирование отчета (9).

Модель потоков данных предприятия представлена контекстной диаграммой в нотации DFD (приложение А) [4]. На диаграмме отображены: основной (главный) процесс обработки данных; внешние сущности; потоки данных, отражающие обобщенную совокупность обрабатываемых документов [5].

Функциональная модель деятельности предприятия в рамках решаемой задачи автоматизации представлена в нотации методологии IDEF0 и включает в себя [6]:

 контекстную диаграмму (приложение Б);

 диаграмму 1-го уровня (приложение В).

1.1.2 Цель проекта

Целью разработки ПИС является автоматизация расчёта и анализа нагрузки на транспортную сеть города.

Внедрение ПИС **«**Моделирование движения городского пассажирского транспорта**»** позволит исключить человеческий фактор и повысить прозрачность принятия решения о вносимых изменениях в транспортную сеть города.

Для достижения поставленной цели необходимо провести работы по проектированию компонентов ПИС, качество результатов которых является залогом успеха всего проекта автоматизации.

1.1.3 Описание архитектуры проекта ПИС на основе методики Захмана

В данном описании представлена архитектура проекта, разработанная с использованием методики Захмана. Методика Захмана является широко применяемым подходом к архитектуре информационных систем, который предлагает систематический подход к проектированию и описанию архитектуры, учитывая множество аспектов, включая бизнес-требования, потребности пользователей, технические аспекты и ограничения системы.

Описание текущей архитектуры ПИС «Моделирование движения городского пассажирского транспорта» представлено в таблице 1.1.

Таблица 1.1 – Ответы на вопросы Захмана

|  |  |
| --- | --- |
| Вопрос | Ответ |
| **1 Почему, зачем (цель, мотивация, motivation)** выполняется бизнес-процесс (БП)? | автоматизация расчёта и анализа нагрузки на транспортную сеть |
| **2 Кто (люди, пользователи, роли, people)** выполняет подпроцессы БП? | Конечные пользователи: Руководитель БП (городское управление транспортом, отдел маршрутов\*) – начальник этого отдела [ссылка на сайт] просмотр результатов обработки данных, их анализ и принятие решения, оператор – сотрудник отдела  Персонал ПИС: АБД, Прикладной программист |
| **3 Как (функции, function)** это делается? | Поддержка функций управления «учёт»:   1. Ведение справочных данных 2. Ведение данных о маршрутах 3. Ведение данных о пассажирах на остановочных пунктах   Поддержка функций управления «анализ»:   1. Имитация работы маршрутно-транспортной сети 2. Визуализация результата расчёта в виде отчёта 3. Визуализация отчёта в виде графика |
| **4 Что (существительные, сущности, data)** лежит в основе деятельности исследуемого бизнес-процесса? | 1. Типы транспортных средств (далее ТС), населённый пункт (далее НП), единицы измерения 2. Маршрут, остановочные пункты (далее ОП)   Кол-во людей на остановке в заданный момент времени |
| **5 Где (сеть, network)** происходят данные подпроцессы бизнес-процесса? | * На сервере предприятия |
| **6 Когда (время, период, time)** выполняют подпроцессы? | * В момент формирования новых или изменения текущих маршрутов |

**1.2 Обзор аналогов**

Тенденция автоматизации бизнес-процессов коснулась и сферы транспорта.

В настоящее время представлено небольшое число систем, позволяющих автоматизировать различные сферы транспортной деятельности. Наиболее популярными среди них являются AnyLogic и PTV Visum. Рассмотрим подробнее каждый аналог.

1.2.1 AnyLogic

AnyLogic — это программное обеспечение для имитационного моделирования, которое позволяет создавать модели различных систем и процессов. Оно обладает широким спектром функциональных возможностей, которые позволяют моделировать различные аспекты систем и процессов [7].

Основные функциональные возможности AnyLogic включают:

– создание моделей. AnyLogic позволяет создавать модели систем и процессов с помощью различных методов моделирования, ч, системная динамика и дискретно-событийное моделирование;

– визуализация моделей. AnyLogic предоставляет инструменты для визуализации моделей, включая возможность создания анимаций и графиков для отображения результатов моделирования;

– анализ моделей. AnyLogic предлагает широкий спектр инструментов для анализа моделей, включая статистический анализ, оптимизацию и визуализацию результатов;

– интеграция с другими программами. AnyLogic может быть интегрирован с другими программными продуктами, такими как Microsoft Excel, SPSS и MATLAB, что позволяет расширить возможности моделирования.

Как и у любого программного обеспечения, у AnyLogic есть свои недостатки. Некоторые из них включают:

– высокая стоимость. AnyLogic является коммерческим программным обеспечением, и его стоимость может быть высокой для некоторых пользователей;

– сложность использования. AnyLogic может быть сложным в использовании для новичков, особенно если они не знакомы с моделированием;

– ограниченная интеграция с другими инструментами. хотя AnyLogic может интегрироваться с другими программами, некоторые пользователи могут предпочесть использовать другие инструменты для своих нужд;

– перегруженный интерфейс. AnyLogic имеет избыточный функционал, который может затруднить процесс моделирования для некоторых пользователей.

1.2.2 PTV Visum

PTV Visum — это пакет программного обеспечения для моделирования и симуляции транспорта, который предоставляет полный набор функций для проектирования, анализа и оптимизации транспортных систем. Он поддерживает различные типы транспортных сетей, включая автомобильные, железнодорожные и системы общественного транспорта [8].

К функциональным возможностям PTV Visum относится:

– проектирование и анализ сети: PTV Visum позволяет пользователям создавать и изменять транспортные сети, рассчитывать время в пути и анализировать транспортные потоки;

– оптимизация маршрутов: PTV Visum можно использовать для оптимизации маршрутов для различных видов транспорта, таких как автобусы, поезда и автомобили;

– моделирование пассажиропотоков: PTV Visum моделирует движение пассажиров по транспортным сетям, принимая во внимание такие факторы, как время в пути, расстояние пешком и выбор вида транспорта.

Однако у PTV Visum есть и некоторые недостатки:

– программное обеспечение довольно сложное, и чтобы научиться эффективно его использовать, может потребоваться много времени;

– в пользовательском интерфейсе может быть сложно ориентироваться, особенно новичкам;

– программное обеспечение может быть дорогим, особенно для малого бизнеса или частных лиц.

Таким образом, анализ аналогов показал, что несмотря на то, что у разрабатываемой программно-информационной системы существуют аналоги, необходимо отметить, что они не обладают достаточным функционалом и возможностями. Создание приложения является необходимым условием повышения прозрачности принятия решения о внесении изменений в транспортную сеть НП.

**1.3 Требования к ПИС**

Классификация разрабатываемой ПИС выявила следующие характеристики:

 вид представления информации – фактографическая информация;

характер отражаемых процессов предметной области – управление транспортными потоками;

 метод организации взаимодействия компонентов ПИС – распределённая ПИС;

 способ организации информационных ресурсов – многомерная база;

 стандарт управления процессом – собственный стандарт предприятия;

 масштаб охвата функций управления: анализ расчёта нагрузки транспортной сети

1.3.1 Методологии и технологии проекта

В дипломной работе использованы следующие подходы и методы

1 Для проектирования функциональной составляющей ПИС:

- структурный (системный) подход для выделения функциональных подсистем, задач, функций;

- метод нисходящего проектирования, использующие его стандарты методологии SADT: IDEF0 – для построения функциональной модели автоматизируемого процесса; DFD – отображения схемы автоматизируемых информационных потоков (ИП) данных;

- ГОСТ 19 серии для формирования функциональной схемы ПИС и схем алгоритмов прикладной программы, описания функциональной составляющей (ГОСТ 19.005-85 ЕСПД. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов. ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению и др.).

2 Для проектирования моделей данных:

- формализованное, табличное описание (ИП) предметной области – структура, качественные и количественные реквизиты документов, фрагментов документов;

- формализованное, табличное описание классов объектов (сущностей) предметной области и отношений (связей) между ними на основе правил семантических объектных моделей, ориентированных на использование реляционной модели данных;

- нотация Ричарда Баркера для построения информационно-логической модели предметной области в виде ER-диаграммы [9];

- реляционная модель данных для формирования логической структуры базы данных;

- стандарты и правила языка SQL для описания физической модели.

3 Стандарты использования операционных систем Windows 7/8/8.1/10.

4 Стандарты и правила системы управления базами данных (СУБД) MySQL, отвечающей требованиям ANSI.

5 Стандарты механизмов доступа к данным: ADO (ActiveX Data Objects.)

6 Стандарты подключения внешних (периферийных) устройств – принтера.

7 Методы и стандарты документирования результатов проектирования:

- ГОСТы 19 и 34 серий (стандарт ГОСТ 19201-78 ЕСПД; стандарт ЕСПД 19503-79 «Руководство системного программиста. Требования к содержанию и оформлению» и др.);

- внутренний стандарт ОГУ - СТО 02069024.

Перечисленные методы, стандарты и средства составляют профиль разрабатываемой ПИС.

1.3.2 Структура и архитектура ПИС

Функциональная составляющая ПИС представлена в таблице 3.

Таблица 3 – Функциональная составляющая предметной области

|  |  |
| --- | --- |
| Компонент функциональной  составляющей ПИС  (функциональная подсистема,  функция/ задача) | Название, краткое описание |
| Функциональная подсистема 1 | Авторизация |
| *Функция 1* | Ведение данных пользователей, их прав |
| *Функция 2* | Проверка и предоставление доступа |
| Функциональная подсистема 2 | Ведение справочных материалов |
| *Функция 1* | Получение и отображение данных из базы (типы транспортных средств, населённые пункты, остановочные пункты, маршруты, единицы измерения) |
| *Функция 2* | Создание новой записи |
| *Функция 3* | Изменение записи |
| *Функция 4* | Удаление записи |
| *Функция 5* | Ведение истории действий пользователей |
| Функциональная подсистема 3 | Графическое представление информации |
| *Функция 1* | Интеграция карты из открытых источников на экран |
| *Функция 2* | Отображение населённых пунктов (НП) |
| *Функция 3* | Отображение остановочных пунктов (ОП) |
| *Функция 4* | Отображение маршрутов |
| *Функция 5* | Отрисовка результатов расчёта нагрузки |
| Функциональная подсистема 4 | Расчёт нагрузки |
| *Функция 1* | Формирование файла расчёта |
| *Функция 2* | Разбор файла расчёта, загруженного на сайт |
| *Функция 3* | Валидация файла в соответствии со справочными материалами |
| *Функция 4* | Имитация работ сети, получение результатов нагрузки |
| *Функция 5* | Формирование выходных данных |
| *Функция 6* | Формирование файла отчёта |

Компоненты обеспечивающих подсистем ПИС и их характеристики представлены в таблице 4.

Таблица 4 – Состав обеспечивающих подсистем

|  |  |
| --- | --- |
| Наименование обеспечивающей подсистемы | Описание |
| 1 | 2 |
| Организационное обеспечение | техническое задание |
| персонал ПИС:   * + оператор - работник отдела   + руководитель - руководитель отдела |
| Правовое обеспечение | лицензия на использование программного средства |
| документы, определяющие уровни доступа пользователей |
| Техническое обеспечение | локальный компьютер |
| принтер |
| Программное обеспечение | Windows 10 |
| MySQL 8.2.0 |
| прикладная программа |
| Microsoft Visual Studio Code |
| Браузер Chrome |
| Информационное обеспечение | внемашинное:   * справочник типов транспортных средств, единиц измерений |
| внутримашинное:   * база данных * экранные формы ввода/вывода данных в окно браузера * файл шаблон с примером заполнения данных для расчёта |
| Лингвистическое обеспечение | - используемый язык ОС: русский язык;  - используемый язык программирования: язык Python, язык запросов SQL; |
| Технологическое обеспечение | руководство оператора |
| руководство прикладному программисту |
| руководство администратору БД |
| Математическое обеспечение | Набор правил на основе сети Петри [10] для расчёта нагрузки на транспортную сеть. |

На рисунке 2 предоставлена схема распределённой программно-информационной системы.

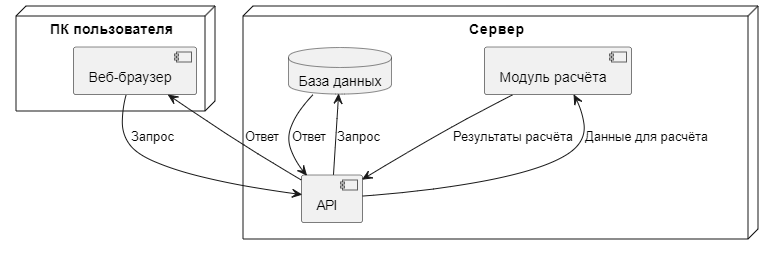


Рисунок 2 – Архитектура ПИС

Реализация ПИС «Моделирование движения городского пассажирского транспорта» предполагает распределённое взаимодействие компонентов системы: компоненты обработки устанавливаются и функционируют на отдельном сервере, а взаимодействие происходят на компьютерах пользователей по средствам общения через сеть.

В первом разделе на основе анализа предметной области и обзора аналогов определена необходимость разработки собственной ПИС, представлены методы проекта, архитектура и структура разрабатываемой программно-информационной системы.

**2 Моделирование компонентов ПИС**

**2.1 Выбор** **инструментальных средств**

Выбор инструментальных средств проекта осуществлен на основе теоретического анализа характеристики известных программных продуктов.

Результаты анализа современных систем управления базами данных (СУБД) приведены в таблице 5 [11, 12, 13].

Таблица 5 – Общие характеристики СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Характеристика | MySQL | Microsoft SQL Server | Oracle |
| 1 | Фирма производитель | MySQL AB (до 2008), Sun Microsystems (2008-2010),  Oracle (с 2010) | Sybase, Ashton-Tate, Microsoft | Oracle |
| 2 | Последняя версия | 5.6.27 | 14 | 12.1.0.2 |
| 3 | Модель данных | Реляционная | Реляционная | Объектно-реляционная |
| 4 | Формат файлов | myd, mrg, tmd | mdf, ndf, sqb | ora, trm |
| 5 | Поддерживаемые объекты БД | Диаграммы, таблицы, пользователи, роли, правила | Диаграммы, таблицы, представления, хранимые процедуры, триггеры, пользователи, роли, правила | Таблицы, пользователи привилегии, роли, хранимые процедуры, триггеры |
| 6 | Технология создания БД | Визуально, SQL-скрипты | Визуально, SQL-скрипты | SQL-скрипты |
| 7 | Операционная система | Microsoft Windows, Unix, Linux, Mac OS X | Microsoft Windows | Microsoft Windows, Unix, Linux, Mac OS X |
| 8 | Инструменты для администри-рования БД | phpMyAdmin, Workbench, Navicat | Интегрированная поддержка Microsoft SQL Server 2014 | dbForge Studio for Oracle, Oracle SQL Developer, OraDeveloper Studio |
| 9 | Лицензия | GNU GPL | Microsoft EULA | Коммерческая |

В таблице 6 приведены рекомендуемые требования к аппаратному обеспечению для работы с СУБД.

Таблица 6 – Требования к аппаратному обеспечению СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Аппаратное обеспечение | MySQL | Microsoft SQL Server | Oracle |
| 1 | Процессор | Intel x86, Amd x64 | Intel x86, Amd x64 | Intel x86, Amd x64 |
| 2 | Оперативная память | 256МБ | 512МБ | 1.5ГБ |
| 3 | Объем жесткого диска | 512МБ | 1ГБ | 6ГБ |

В таблице 7 представлены возможности СУБД [11, 12, 13].

Таблица 7 – Сравнение возможностей СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Характеристика | MySQL | Microsoft SQL Server | Oracle |
| 1 | Независимость от платформы | Да | Только семейство ОС Windows | Да |
| 2 | Условия использования и распространения | Полностью бесплатна для использования и распространения | Необходимо приобретать лицензию на использование | Необходимо приобретать лицензию на использование |
| 3 | Возможность создания сервера БД | Есть | Есть | Есть |
| 4 | Средства поддержки ограничений целостности БД | Есть | Есть | Есть |
| 5 | Возможность настройки реализации прав доступа для отдельных пользователей | Есть | Есть | Есть |
| 6 | Наличие встроенных средств для создания резервной копии БД | Есть | Есть | Есть |
| 7 | Возможности по защите данных | Большое количество функций обеспечивающих безопасность по умолчанию | Отвечает всем требованиям по защите данных | Универсальные средства защиты |
| 8 | Язык интерфейса | Русский, английский | Русский, английский | Русский, английский |
| 9 | Ориентация на Интернет технологии | Есть | Есть | Есть |
| 10 | Объемы обрабатываемых данных | Ориентирована на небольшие объемы данных (несколько гигабайт) | Обработка данных не выше 1 терабайта | Рассчитана на работу с огромными объемами данных и большим числом пользователей |

Для разработки распределённой ПИС выбрана СУБД MySQL. Несмотря на то, что СУБД обладает недостатком (зависимость от платформы – п. 7 таблицы 5), она имеет следующие преимущества: содержит широкий ассортимент объектов БД (п. 5 таблицы 6); имеет развитые средства защиты данных (п. 7 таблицы 7) и, в тоже время, не является слишком сложной для администрирования. Кроме того, различные инструментальные средства разработки прикладных программ поддерживают механизмы доступа к базам данных формата MySQL.

Результаты сравнительного анализа инструментальных средств разработки прикладных программ приведены в таблице 8 [11, 12, 13].

Таблица 8 – Сравнительные характеристики средств разработки приложений

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Характеристика | NetBeans | Microsoft Visual Studio Code | Borland Delphi |
| 1 | Фирма производитель | NetBeans Community | Microsoft | Embarcadero Technologies |
| 2 | Последняя версия | 8.0.2 | 14.0 | 10 |
| 3 | Операционная система | Microsoft Windows, Linux, Mac OS X | Microsoft Windows | Microsoft Windows |
| 4 | Встроенные языки | Java, HTML5, PHP, C/C++ | Basic .NET, C++, C#, F# | Object Pascal |
| 5 | Поддержка стандарта SQL | Да | Да | Да |
| 6 | Поддержка ООП | Да | Да | Да |
| 7 | Механизмы доступа к БД | JDBC | ADO | BDE, ADO, Express |
| 8 | Лицензия | LGPLv2.1, GPLv2 with Classpatch exception | лицензия MIT и проприетарная | Проприетарная |

В таблице 9 приведены рекомендуемые требования к аппаратному обеспечению для работы со средами разработки приложений.

Таблица 9 – Требования к аппаратному обеспечению

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Аппаратное обеспечение | NetBeans | Microsoft Visual Studio Code | Borland Delphi |
| 1 | Процессор | Intel x86, Amd x64 | Intel x86, Amd x64 | Intel x86, Amd x64 |
| 2 | Оперативная память | 512МБ | 2ГБ | 1ГБ |
| 3 | Объем жесткого диска | 512МБ | 2ГБ | 1ГБ |

Общие характеристики инструментов для разработки приложений баз данных приведены в таблице 10.

Таблица 10 – Общие характеристики инструментов для работы с СУБД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п.п. | Характеристика | phpMyAdmin | Интегрированная поддержка Microsoft SQL Server | Oracle SQL Developer |
| 1 | Разработчик | phpMyAdmin Developer Team | Sybase,  Ashton-Tate,  Microsoft | Oracle Corporation |
| 2 | Последняя версия | 4.5.1 | 14 | 4.0.3 |
| 3 | Операционная система | Windows, Linux, Mac Os X | Windows, Unix, OS/2 | Windows, Linux, Mac OS X |
| 4 | Язык интерфейса | Русский, Английский | Русский, Английский | Английский |
| 5 | Визуальное проектирование | Поддерживает | Поддерживает | Поддерживает |
| 6 | Лицензия | GNU GPL | Microsoft EULA | Проприетарная OTN SQL Developer License |

В качестве средства разработки приложений выбрана Microsoft Visual Studio Code (по результатам анализа, приведенного в таблице 8), а так же инструмент для работы с базами данных Интегрированная поддержка phpMyAdmin (по результатам анализа, приведенного в таблице 10).

**2.2 Формирование функциональной составляющей ПИС**

Иерархия функций ПИС «Моделирование движения городского пассажирского транспорта» сформирована на основе функциональной модели (приложение В) и представлена на рисунке 3.

К концептуальному уровню проекта ПИС относится схема иерархии функций [1]. Исходными данными для моделирования иерархии функций является функциональная модель. К задачам функциональной модели добавляются задачи авторизации пользователя и ведения справочных данных. Как правило, эти задачи решаются средствами СУБД.

Главные узлы иерархии функций ПИС – это задачи системы, Ф – функции.

Функции являются листьями дерева иерархии, они пронумерованы, т.е. их число конечно.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| АПИС | Авторизация пользователя |  |  | Ф1 |
| Ведение справочных данных |  |  |  |
|  | Тип транспортного средства | Добавление/Обновление | Ф2 |
|  |  | Поиск/Просмотр | Ф3 |
|  |  |  |  |
|  | Единица измерения | Добавление/Обновление | Ф4 |
|  |  | Поиск/Просмотр | Ф5 |
|  |  |  |  |
|  | Населённый пункт | Добавление/Обновление | Ф6 |
|  |  | Поиск/Просмотр | Ф7 |
|  |  |  |  |
|  | Остановочный пункт | Добавление/Обновление | Ф8 |
|  |  | Поиск/Просмотр | Ф9 |
|  |  |  |  |
|  | Маршрут | Добавление/Обновление | Ф10 |
|  |  | Поиск/Просмотр | Ф11 |
| Графическое представление информации |  |  |  |
|  | Интеграция карты из открытых источников |  |  |
|  |  | Поиск/Просмотр | Ф12 |
|  |  |  |  |
|  | Отображение населённых пунктов |  |  |
|  |  | Поиск/Просмотр | Ф13 |
|  |  |  |  |
|  | Отображение остановочных пунктов |  |  |
|  |  | Поиск/Просмотр | Ф14 |
|  |  |  |  |
|  | Отображение маршрутов |  |  |
|  |  | Поиск/Просмотр | Ф15 |
|  |  |  |  |
|  | Отрисовка результатов расчёта нагрузки |  |  |
|  |  | Поиск/Просмотр | Ф16 |
| Расчёт нагрузки |  |  |  |
|  | Формирование файла расчёта |  |  |
|  |  | Поиск/Просмотр | Ф17 |
|  |  |  |  |
|  | Разбор файла расчёта, загруженного на сайт |  |  |
|  |  | Поиск/Просмотр | Ф18 |
|  |  |  |  |
|  | Валидация файла в соответствии со справочными материалами |  |  |
|  |  | Поиск/Просмотр | Ф19 |
|  |  |  |  |
|  | Расчёт нагрузки |  |  |
|  |  | Поиск/Просмотр | Ф20 |
|  |  |  |  |
|  | Формирование выходных данных |  |  |
|  |  | Поиск/Просмотр | Ф21 |
|  |  |  |  |
|  | Формирование файла отчёта |  |  |
|  |  | Поиск/Просмотр | Ф22 |
|  |  |  |  |

Рисунок 3 – Иерархия функций

Одним из ключевых компонентов этой системы является диаграмма вариантов использования, которая описывает основные сценарии взаимодействия между различными участниками процесса.

Диаграмма представляет собой графическое изображение, на котором отражены актеры (конечные пользователи, внешние системы и т.д.), их варианты использования и связи между ними. Данная диаграмма представлена на рисунке 2.1.

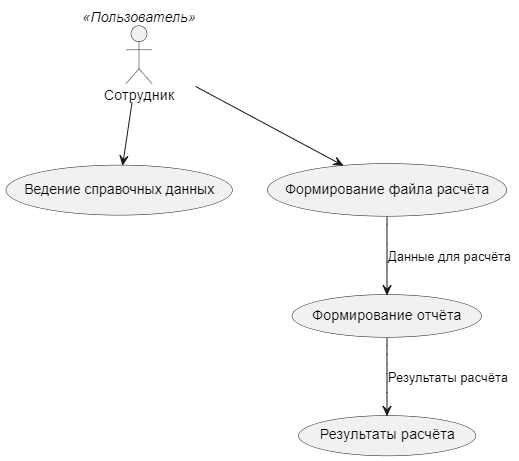


Рисунок 2.1 – Диаграмма вариантов использования

Полученный результат становится основой для моделирования основных интерфейсных форм прикладного программного обеспечения, состава программных модулей, процедур и функций программно-информационной системы, проверки формируемой модели данных.

2.2.1 Внешний уровень архитектуры БД

Функциональная схема ПС представлена на Рисунок . На нём перечислены основные задачи / функции АС; источники (слева) и - приёмники данных (справа).

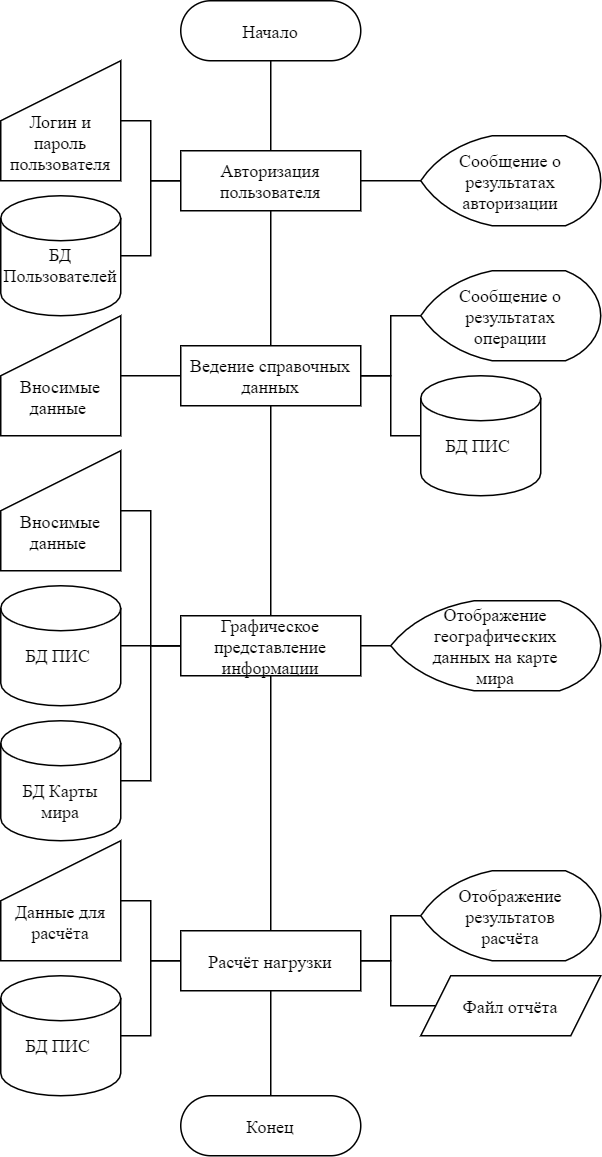


Рисунок 2.2 - Функциональная схема ПС

**2.3 Проектирование информационного обеспечения ПИС**

2.3.1 Внешний уровень архитектуры БД

2.3.1.1 Формализованное описание предметной области

Формализованное описание классов объектов предметной области ПИС представлено в таблице 11.

Класс объектов может иметь несколько уникальных ключей (УИ1, УИ2 и т.д.); среди них выбирается один – кандидат в первичный ключ (П).

Физические характеристики – это тип, длина значения свойства; логические ограничения – это диапазон значений, прописные, строчные буквы для символьных свойств и т.п.; процессы – это процессы, которые разрешены для значения свойства класса объектов: генерация, ввод, возможность обновления, просмотра.

В таблице использованы сокращения: У – уникальный идентификатор, П – кандидат в первичный ключ (главный уникальный идентификатор), м.б. – значение может быть, д.б. - должно быть, Г – генерация данных, Вв – ввод данных, Пр – просмотр данных, Об – обновление данных.

Таблица 11 – Формализованное описание классов объектов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Класс объектов/  Свойство | Ключ | Физические характеристики | Обязательность значения | Логические ограничения | Процессы |
| 1 | 2 | 3 | 4 | 5 | 6 |
| НАСЕЛЁННЫЙ ПУНКТ |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| название |  | символы, 250 | да | Не больше 250 символов | Вв, Пр, Об |
| регион |  | число, 1000 | да | > 0 и < 1000 | Вв, Пр, Об |
| широта |  | число, -11.154645 | да | > -90 и < 90 | Вв, Пр, Об |
| Долгота |  | число, 13.176645 | да | > -180 и < 180 | Вв, Пр, Об |
| ТИП ТРАНСПОРТНОГО СРЕДСТВА |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| название |  | символы, 250 | да | Не больше 250 символов | Вв, Пр, Об |
| вместимость |  | символы, 20 | да | > 0 | Вв, Пр, Об |
| описание |  | символы, 20 | нет |  | Вв, Пр, Об |
| ОСТАНОВОЧНЫЙ ПУНКТ |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| название |  | символы, 250 | да | Не больше 250 символов | Вв, Пр, Об |
| широта |  | число, -11.154645 | да | > -90 и < 90 | Вв, Пр, Об |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Долгота |  | число, 13.176645 | да | > -180 и < 180 | Вв, Пр, Об |
| МАРШРУТ |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| название |  | символы, 250 | да | Не больше 250 символов | Вв, Пр, Об |
| интервал движения |  | число, 100 | нет | > 0 | Вв, Пр, Об |
| количество транспорта на маршруте |  | число, 20 | нет | > 0 | Вв, Пр, Об |
| СВЯЗЬ МАРШРУТА И ОП |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| ЕДИНИЦА ИЗМЕРЕНИЯ |  |  |  |  |  |
| код | УИ1, ПК | число, 10 | да | > 0 | Г, Пр |
| название |  | символы, 250 | да | Не больше 250 символов | Вв, Пр, Об |
| краткое название |  | символы, 15 | да | Не больше 15 символов | Вв, Пр, Об |

Формализованное описание связей между классами объектов приведём в таблице 12. В таблице использованы следующие сокращения: Гл – главный, П – подчиненный, КО – класс объектов; д.б. – должна быть, м.б. – может быть, М – мощность связи «много».

Таблица 12 – Формализованное описание связей между классами объектов

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Классы объектов со стороны | | Опциональ-ность связи со стороны | | Имя связи со стороны | | Тип связи со стороны | |
| Гл КО | ПО | Гл КО | ПО | Гл КО | ПО | Гл КО | ПО |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| НАСЕЛЁННЫЙ ПУНКТ | ОСТАНОВОЧНЫЙ ПУНКТ | м.б. | д.б. | включает в себя | включает в себя | 1 | М |
| НАСЕЛЁННЫЙ ПУНКТ | МАРШРУТ | м.б. | д.б. | включает в себя | включает в себя | 1 | М |
| ТИП ТРАНСПОРТНОГО СРЕДСТВА | МАРШРУТ | м.б. | д.б. | соответствует | имеет | 1 | М |
| СВЯЗЬ МАРШРУТА И ОП | МАРШРУТ | м.б. | д.б. | включает в себя | соответствует | 1 | М |
| СВЯЗЬ МАРШРУТА И ОП (далее ДОК) | ОСТАНОВОЧНЫЙ ПУНКТ | м.б. | д.б. | включает в себя | соответствует | 1 | М |

Необходимо проверить связи, путем чтения с обеих сторон, используя правило чтения:

1. Каждому МАРШРУТУ должен соответствовать один или более ДОК
2. Каждый ДОК должен относиться к одному и только одному МАРШРУТУ
3. Каждый ОП может относиться к одному или более ДОК
4. Каждой ДОК должен соответствовать один и только один ОП
5. Каждому МАРШРУТУ должен соответствовать один или более ОП
6. Каждый ОП должен относиться к одному или более МАРШРУТУ
7. Каждый ОП должен относиться к одному НП
8. Каждому НП может относиться один или более ОП
9. Каждый Маршрут должен относиться к одному НП
10. Каждому НП может относиться один или более Маршрут

2.3.1.2 Пользователи ПИС

В ходе анализа предметной области выявляется и описывается состав пользователей разрабатываемой ПИС, а также их уровни доступа к базе данных. В рамках дипломной работы выделяются следующие роли пользователей и их уровни доступа к данным:

- администратор базы данных (АБД). Обладает всеми правами, может осуществлять все действия с базой данных, объектами базы данных и их поколениями (версиями, копиями);

- прикладной программист (ПП). Обладает всеми правами на все действия с объектами БД и их поколениями, кроме операций удаления объектов и данных;

- конечный пользователь - оператор (КПО), работник отдела, выполняющий операции добавления, обновления, поиска и просмотра данных;

- конечный пользователь – руководитель предприятия (КПР), руководитель отдела. Пользователю доступна только операция чтения данных.

Формализованное описание состава пользователей ПИС и их уровни доступа представлены в таблице 13. В таблице использованы сокращения операций, осуществляемых с данными: R - read (чтение); I - insert (добавление); U - update (обновление); D - delete (удаление); E - execute (обращение к хранимой процедуре).

Таблица 13 – Состав и уровни доступа пользователей ПИС

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Классы объектов/свойства | Пользователи | | | |
| АБД | ПП | КПО | КПР |
| 1 | 2 | 3 | 4 | 5 |
| НАСЕЛЁННЫЙ ПУНКТ |  |  |  |  |
| код | RIUDE | RIUE | R | R |
| название | RIUDE | RIUE | R | R |
| регион | RIUDE | RIUE | R | R |
| широта | RIUDE | RIUE | R | R |
| Долгота | RIUDE | RIUE | R | R |
| ТИП ТРАНСПОРТНОГО СРЕДСТВА |  |  |  |  |
| код | RIUDE | RIUE | R | R |
| название | RIUDE | RIUE | R | R |
| вместимость | RIUDE | RIUE | R | R |

Продолжение таблицы 13

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| описание | RIUDE | RIUE | R | R |
| ОСТАНОВОЧНЫЙ ПУНКТ |  |  |  |  |
| код | RIUDE | RIUE | RI | R |
| название | RIUDE | RIUE | RI | R |
| широта | RIUDE | RIUE | RI | R |
| Долгота | RIUDE | RIUE | RI | R |
| МАРШРУТ |  |  |  |  |
| код | RIUDE | RIUE | RI | R |
| название | RIUDE |  | R | R |
| интервал движения | RIUDE | RIUE | R | R |
| количество транспорта на маршруте | RIUDE | RIUE | R | R |
| СВЯЗЬ МАРШРУТА И ОП |  |  |  |  |
| код | RIUDE | RIUE | RI | R |
| ЕДИНИЦА ИЗМЕРЕНИЯ |  |  |  |  |
| код | RIUDE | RIUE | R | R |
| название | RIUDE | RIUE | R | R |
| краткое название | RIUDE | RIUE | R | R |

Уровни доступа конечных пользователей соответствуют разрешенным в предметной области операциям обработки значений свойств классов объектов (таблица 11, 6 столбец).

2.3.2 Концептуальный уровень архитектуры БД

2.3.2.1 Информационно-логическая модель предметной области

ER-диаграмма предметной области по методологии Ричарда Баркера приведена в приложении Г [9].

На ER-диаграмме представлено 5 классов объектов. Основной вид связи между классами объектов - 1:М с обязательной опциональностью на стороне «много».

В таблице 14 приведены результаты тестирования проектных решений с помощью перекрестной проверки полученной модели предметной области (перечень классов объектов – таблица 11) заявленному составу функций ПИС (рисунок 3).

В таблице 14 приведены сокращения:

- Ф – функции;

- НП – населённый пункт;

- ТТП – тип транспортного средства;

- ОП – остановочный пункт;

- М – маршрут;

- СВЯ – связь маршрута и остановочного пункта;

- ЕИ – единица измерения;

Таблица 14 – Перекрестная проверка

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ф | Классы объектов | | | | | |
| НП | ТТП | ОП | М | СВЯ | ЕИ |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Ф1 |  |  |  |  |  |  |
| Ф2 |  | IUD |  |  |  |  |
| Ф3 |  | R |  |  |  |  |
| Ф4 |  |  |  |  |  | IUD |
| Ф5 |  |  |  |  |  | R |
| Ф6 | IUD |  |  |  |  |  |
| Ф7 | R |  |  |  |  |  |
| Ф8 |  |  | IUD | R |  |  |
| Ф9 |  |  | R |  |  |  |
| Ф10 |  |  | R | IUD |  |  |
| Ф11 |  |  |  | R |  |  |
| Ф12 | R |  |  |  |  |  |
| Ф13 | R |  |  |  |  |  |
| Ф14 |  |  | R |  |  |  |
| Ф15 |  |  |  | R |  |  |
| Ф16 | R | R | R | R | R |  |
| Ф17 | R |  | R | R | R |  |
| Ф18 | R | R | R | R | R |  |
| Ф19 | R | R | R | R | R |  |
| Ф20 | R | R | R | R | R |  |
| Ф21 | R | R | R | R | R |  |
| Ф22 | R | R | R | R | R | R |

Рассмотренная таблица позволяет сделать выводы о модели предметной области:

- модель не является избыточной, поскольку в таблице отсутствуют пустые столбцы – все смоделированные классы объектов используются для той, или иной функции;

- модель является достаточной, поскольку в таблице отсутствуют пустые строки – для реализации каждой функции выделены определенные классы объектов.

2.3.2.2 Даталогическая модель БД

Метод нисходящего проектирования требует далее формирования даталогической модели (ДЛМ) реляционной базы данных (РБД) на основе ER-диаграммы. Результат в виде графического изображения логической структуры реляционной базы данных приведен в приложении Д.

Схема полученной БД соответствует как минимум 3НФ, поскольку во всех реляционных отношениях значения всех атрибутов являются атомарными, первичные ключи не составные и отсутствуют транзитивные зависимости между не ключевыми атрибутами и первичными ключами.

**2.3.3 Внутренний уровень архитектуры БД**

Следующий этап проектирования РБД – формирование физической модели базы данных. В таблицах 15-18 приведено технические описания реляционных таблиц на языке определения данных выбранной СУБД – язык SQL.

Таблица 15 – Техническое описание таблицы City

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя отношения в схеме РБД – Населённый пункт | | | | | |
| Имя поля | id | name | region | latitude | longitude |
| Ключ | PRIMARY KEY |  |  |  |  |
| Тип, длина | UNIQUEIDENTIFIER | VARCHAR(250) | INT | FLOAT | FLOAT |
| Обязательность значения | NOT NULL | NOT NULL | NOT NULL | NOT NULL | NOT NULL |
| Логическое ограничение |  | name > 0 | region > 0 и region < 1000 | > -90 и < 90 | > -180 и < 180 |
| Примеры данных | 1 | Оренбург | 56 | -11.154645 | 13.176645 |

Таблица 16 – Техническое описание таблицы TC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя отношения в схеме РБД – Тип транспортного средства | | | | |
| Имя поля | id | name | capacity | description |
| Ключ | PRIMARY KEY |  |  |  |
| Тип, длина | UNIQUEIDENTIFIER | VARCHAR(100) | INT | VARCHAR(300) |
| Обязательность значения | NOT NULL | NOT NULL | NOT NULL |  |
| Логическое ограничение |  | > 0 | >0 |  |
| Примеры данных | 1 | Пазик | 60 | Что за тип транспорта |

Таблица 17 – Техническое описание таблицы BusStop

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя отношения в схеме РБД – Остановочный пункт | | | | | |
| Имя поля | id | city | name | latitude | longitude |
| Ключ | PRIMARY KEY | FOREIGN KEY |  |  |  |
| Тип, длина | UNIQUEIDENTIFIER | UNIQUEIDENTIFIER | VARCHAR(250) | FLOAT | FLOAT |
| Обязательность значения | NOT NULL | NOT NULL | NOT NULL | NOT NULL | NOT NULL |
| Логическое ограничение |  |  | > 0 | > -90 и < 90 | > -180 и < 180 |
| Примеры данных | 1 | 1 | ОГУ | -11.154645 | 13.176645 |

Таблица 18 – Техническое описание таблицы Route

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя отношения в схеме РБД – Маршрут | | | | | | |
| Имя поля | id | city | TC | interval | amount | busstop |
| Ключ | PRIMARY KEY | FOREIGN KEY | FOREIGN KEY | INT | INT | FOREIGN KEY |
| Тип, длина | UNIQUEIDENTIFIER (UI) | UI | UI | NOT NULL | NOT NULL | UI |
| Обязательность значения | NOT NULL | NOT NULL | NOT NULL | >0 | >0 | NOT NULL |
| Логическое ограничение |  |  |  | 60 | 60 |  |
| Примеры данных | 1 | 1 | 1 | INT | INT | 1 |

2.3.4 Макеты экранных форм для ввода и вывода информации

На рисунке 4 представлен пример файла отчёта

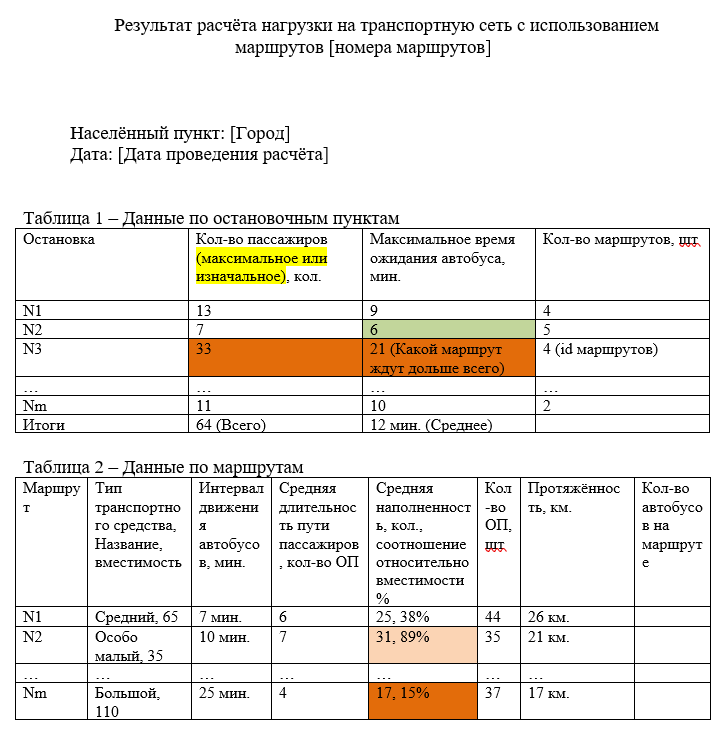


Рисунок 4 – пример файла отчёта

На рисунке 5 представлен макет экранной формы отображения данных

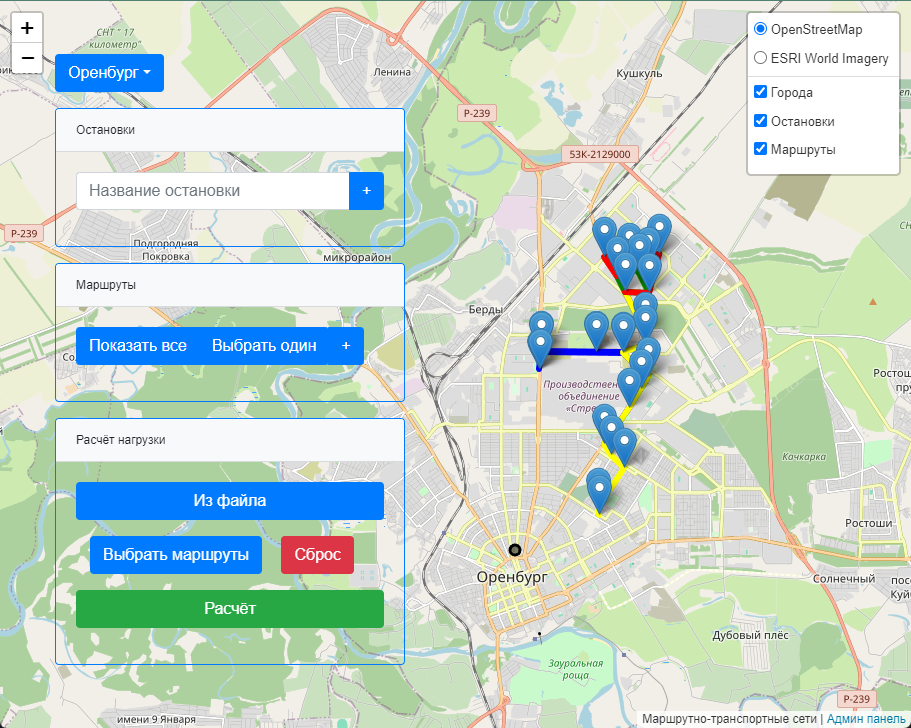


Рисунок 5 – экранная форма отрисовки НП, ОП и маршрутов

На рисунке 6 представлен макет экранной формы ведения справочных материалов

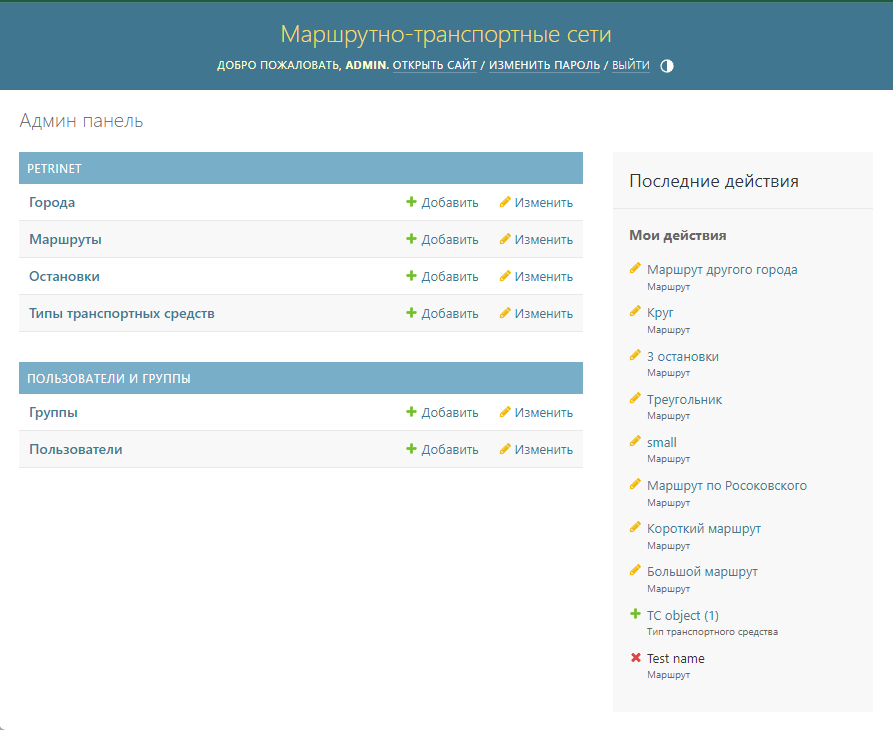


Рисунок 6 – экранная форма ведения справочных материалов

Во втором разделе приведены решения, полученные в результате проектирования компонентов ПИС:

- иерархия функций (рисунок);

- описание классов объектов предметной области и связей между ними (таблицы);

- информационно-логическая модель предметной области (ER-диаграмма). ER-диаграмма проверена путем чтения и проверки правил предметной области;

- даталогическая модель реляционной базы данных (схема), схема базы данных проверена на соответствие 3НФ;

- физическое описание таблиц базы данных на языке SQL.

Спроектированы уровни доступа пользователей, проведена перекрестная проверка иерархии функций и модели данных (ER-диаграммы).

**3** **Реализация мероприятий по защите данных**

**3.1 Технология создания объектов БД. Поддержка целостности данных**

На основании физической модели данных (таблицы 15-18) разработаны SQL-команды для реализации реляционных таблиц базы данных (приложение Е).

Структура каждой команды Create Table содержит:

- описание первичных ключей, их значения определены как обязательные (конструкция Not Null);

- явно описаны ссылки внешних ключей подчиненных таблиц на первичные ключи главных по связи таблиц (конструкции Foreign Key);

- описаны выявленные в предметной области ограничения (таблица 11): типы и длина полей, обязательность (опциональность) значений (конструкция Not Null), ограничения.

Таким образом, поддерживаются ограничения целостности реляционной базы данных, определяемые предметной областью и реляционной моделью данных.

Физическая модель данных реализуется в среде СУБД MySQL, для администрирования используется PHPMyAdmin, так как она позволяет работать в текстовом и графическом режиме [11].

Последовательность создания таблиц: сначала главные, затем подчиненные. В среде СУБД выполняется скрипт с описанием структуры таблицы (команда Create Table), в результате получаем созданный объект базы.

**3.2 Назначение прав доступа пользователям ПИС**

Права доступа пользователей ПИС описаны в таблице 13. Для их реализации необходимо в среде СУБД выполнить следующие виды работ:

1 Создание пользователя. Осуществляется с помощью SQL-команды: CREATE USER 'имя\_пользователя'@'localhost' IDENTIFIED BY 'пароль'.

2 Назначить пользователю привилегии. Выполняются SQL-команды вида:

GRANT [SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE USER] ON ИМЯ\_БД.ИМЯ\_СУЩНОСТИ TO 'имя\_пользователя'@'localhost'.

SQL-скрипты создания пользователей и описание их привилегий представлены в приложении Е.

**3.3 Резервное копирование и восстановление БД**

Базовая структура синтаксиса Transact-SQL для полного резервного копирования базы данных:

BACKUP DATABASE *database*

TO *backup\_device* [ **,**...*n* ]

[ WITH *with\_options* [ **,**...*o* ] ] ;

где *database -* база данных для резервного копирования

*backup\_device*  - указывает список от 1 до 64 устройств резервного копирования, используемых для создания резервной копии. Можно указать как физическое устройство резервного копирования, так и соответствующее логическое устройство, если оно уже определено. Для указания физического устройства резервного копирования используйте параметр DISK или TAPE.  
{ DISK | TAPE } **=***имяфизическогоустройства\_резервного \_копирования*

*with\_options*  - при необходимости можно указать один или несколько дополнительных параметров:

DESCRIPTION **=** { **'***text***'** | **@***text\_variable* }  
Задает произвольное текстовое описание резервного набора данных. В этой строке может содержаться до 255 символов.

NAME **=** { *backup\_set\_name* | **@***backup\_set\_name\_var* }  
Указывает имя резервного набора данных. Длина имени не может превышать 128 символов. Если параметр NAME не указан, то имя является пустым.

Пример:

USE Base;

GO

BACKUP DATABASE Base

TO DISK = 'Z:\SQLServerBackups\ Base.Bak'

WITH FORMAT,

MEDIANAME = 'Z\_SQLServerBackups',

NAME = 'Full Backup of Base ';

GO

Для восстановления базы данных используется команда RESTORE, для которой необходимо указать логическое или физическое устройство резервного копирования, из которого будет восстанавливаться копия.

Пример:

RESTORE DATABASE Base

FROM DISK = 'Z:\SQLServerBackups\ Base.bak' ;

В третьем разделе рассмотрены вопросы создания объектов базы данных в среде Microsoft SQL Server 2014, полученных в результате проектных решений. Рассмотрены вопросы защиты данных средствами указанной СУБД.

**Заключение**

Для достижения поставленной цели – создания проекта автоматизации процесса расчёта и анализа нагрузки на транспортную сеть:

а) проведен анализ предметной области и определены требования к программно-информационной системе (ПИС);

б) проведен обзор аналогов;

в) определены методологии и технологии проекта;

г) спроектирована структура и архитектура ПИС (рисунок 2);

д) смоделированы компоненты ПИС:

- сформирована функциональная составляющая ПИС (Приложение А, Приложение Б, Приложение В);

- спроектированы компоненты информационного обеспечения ПИС: информационно-логическая модель предметной области (Приложение Г); построена даталогическая модель БД (Приложение Д); спроектированы макеты экранных форм для вывода информации (рисунки 4, 5); разработана физическая модель базы данных (таблицы 15-32, приложение Е);

- рассмотрены вопросы защиты данных: определены пользователи ПИС (таблица 13); назначены права доступа пользователям ПИС, реализованы ограничения целостности реляционной базы данных, представлены действия по резервному копированию и восстановлению базы данных в среде СУБД MySQL (приложение Е);

е) проведено тестирование сформированной иерархии функций и модели данных.

Таким образом, поставленная цель достигнута – данные проектные решения могут быть использованы для создания ПИС.

**Список использованных источников**

9 Ричард Баркер CASE Method - Моделирование взаимосвязей между сущностями. [Электронный ресурс] – Режим доступа: WWW.URL: <ftp://ftp.opennet.ru/pub/docs/case/casemeth.zip>.

10 Федотов И.Е. **Некоторые приемы параллельного программирования**: Учебное пособие. - М.: Изд-во МГИРЭА(ТУ), 2008. - 188 с.  
ISBN 978-5-7339-0724-6.

11 MySQL. [Электронный ресурс] – Режим доступа: https://www.php.net/manual/ru/book.mysql.php.

12 Microsoft SQL Server. [Электронный ресурс] – Режим доступа: https://www.microsoft.com/ru-ru/sql-server/sql-server-2019.

13 Oracle. [Электронный ресурс] – Режим доступа: https://www.oracle.com/cis/.

**Приложение А**

***(обязательное)***

**Контекстная диаграмма в нотации DFD**

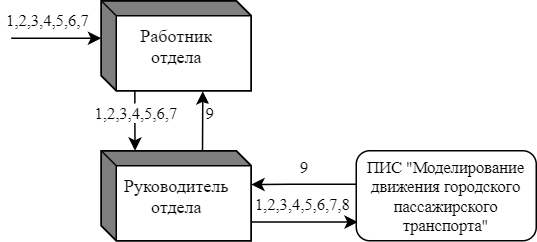


Рисунок А.1 – Контекстная диаграмма потоков данных в нотации DFD

**Приложение Б**

***(обязательное)***

**Контекстная диаграмма в нотации IDEF0**



Рисунок Б.1 – Контекстная диаграмма функциональной модели предприятия в нотации IDEF0

**Приложение В**

***(обязательное)***

**Диаграмма 1-го уровня в нотации IDEF0**

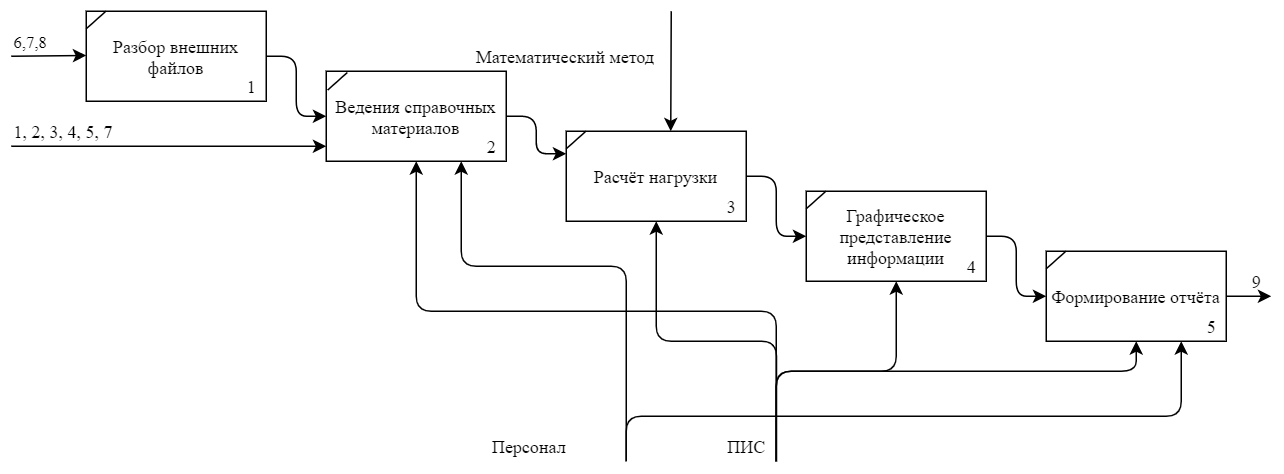


Рисунок В.1 – Функциональная модель ПИС. Диаграмма первого уровня в нотации IDEF0

**Приложение Г**

***(обязательное)***

**ER-диаграмма**

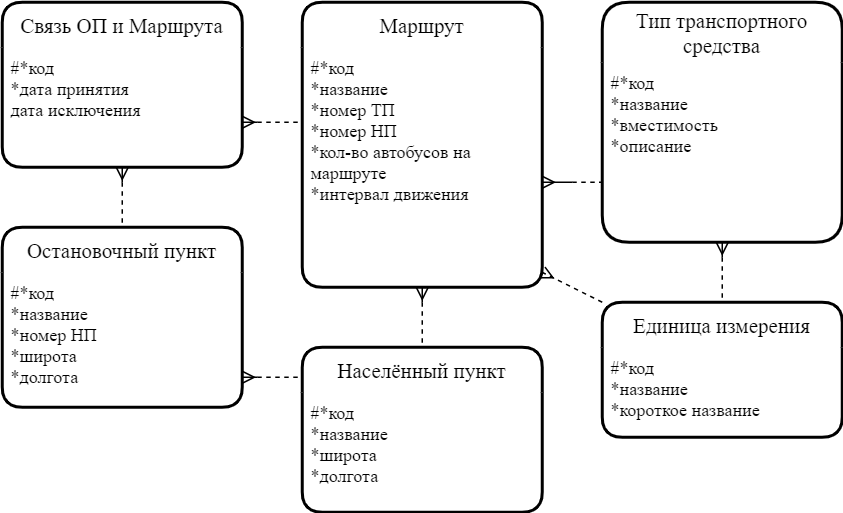


Рисунок Г.1 – ER-диаграмма

**Приложение Д**

***(обязательное)***

**Даталогическая модель БД**

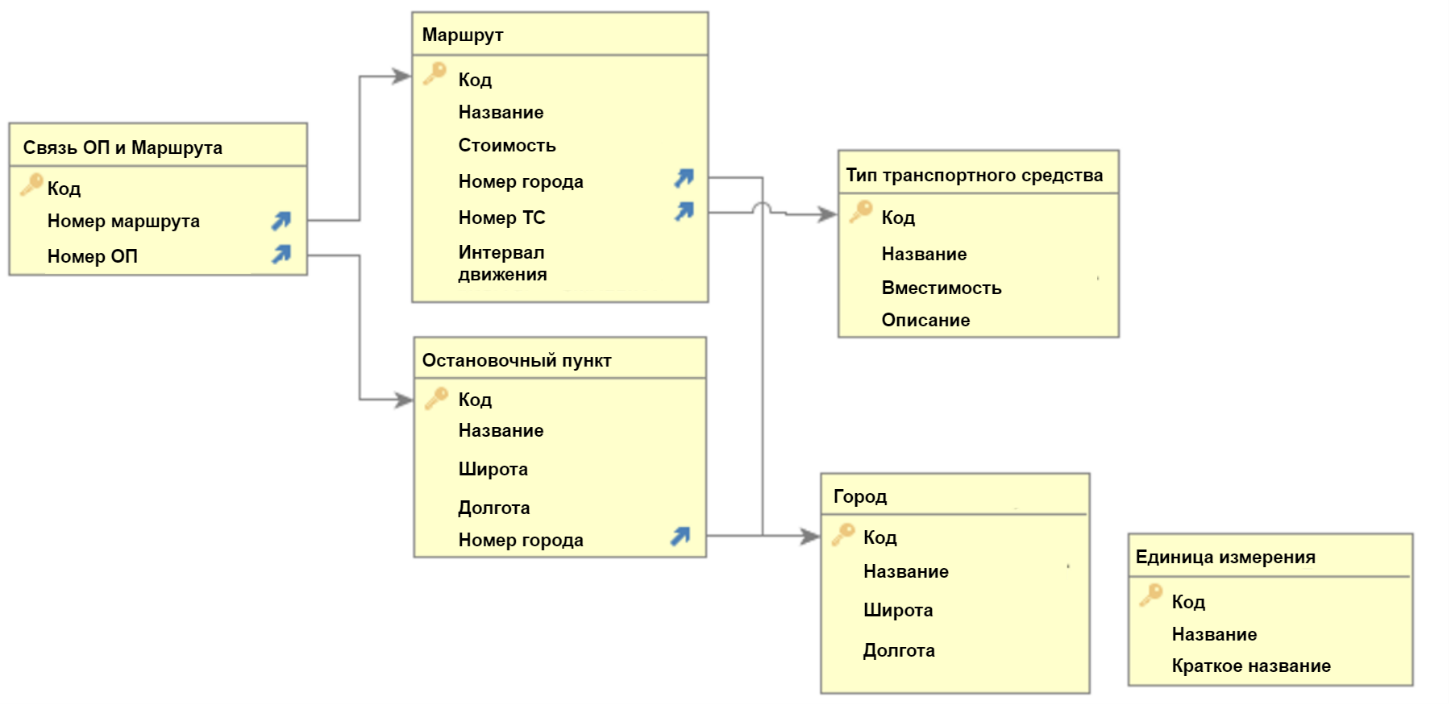


Рисунок Д.1 – Логическая структура базы данных

**Приложение Е**

***(обязательное)***

**SQL-скрипты**

BEGIN;

--

-- Create model BusStop

--

CREATE TABLE "PetriNET\_busstop" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(250) NOT NULL, "latitude" real NOT NULL, "longitude" real NOT NULL);

--

-- Create model City

--

CREATE TABLE "PetriNET\_city" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "region" smallint NOT NULL, "name" varchar(250) NOT NULL, "latitude" real NOT NULL, "longitude" real NOT NULL);

--

-- Create model EI

--

CREATE TABLE "PetriNET\_ei" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(250) NOT NULL, "short\_name" varchar(15) NOT NULL);

--

-- Create model TC

--

CREATE TABLE "PetriNET\_tc" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(250) NOT NULL, "capacity" smallint NOT NULL, "description" text NULL);

--

-- Create model Route

--

CREATE TABLE "PetriNET\_route" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(250) NOT NULL, "interval" smallint NULL, "amount" smallint NULL, "list\_coord" text NULL CHECK ((JSON\_VALID("list\_coord") OR "list\_coord" IS NULL)), "city\_id" bigint NOT NULL REFERENCES "PetriNET\_city" ("id") DEFERRABLE INITIALLY DEFERRED, "tc\_id" bigint NULL REFERENCES "PetriNET\_tc" ("id") DEFERRABLE INITIALLY DEFERRED);

CREATE TABLE "PetriNET\_route\_busstop" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "route\_id" bigint NOT NULL REFERENCES "PetriNET\_route" ("id") DEFERRABLE INITIALLY DEFERRED, "busstop\_id" bigint NOT NULL REFERENCES "PetriNET\_busstop" ("id") DEFERRABLE INITIALLY DEFERRED);

--

-- Add field city to busstop

--

CREATE TABLE "new\_\_PetriNET\_busstop" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(250) NOT NULL, "latitude" real NOT NULL, "longitude" real NOT NULL, "city\_id" bigint NOT NULL REFERENCES "PetriNET\_city" ("id") DEFERRABLE INITIALLY DEFERRED);

INSERT INTO "new\_\_PetriNET\_busstop" ("id", "name", "latitude", "longitude", "city\_id") SELECT "id", "name", "latitude", "longitude", NULL FROM "PetriNET\_busstop";

DROP TABLE "PetriNET\_busstop";

ALTER TABLE "new\_\_PetriNET\_busstop" RENAME TO "PetriNET\_busstop";

CREATE INDEX "PetriNET\_route\_city\_id\_e1af5c13" ON "PetriNET\_route" ("city\_id");

CREATE INDEX "PetriNET\_route\_tc\_id\_22990a62" ON "PetriNET\_route" ("tc\_id");

CREATE UNIQUE INDEX "PetriNET\_route\_busstop\_route\_id\_busstop\_id\_deffd66b\_uniq" ON "PetriNET\_route\_busstop" ("route\_id", "busstop\_id");

CREATE INDEX "PetriNET\_route\_busstop\_route\_id\_dc96dfb1" ON "PetriNET\_route\_busstop" ("route\_id");

CREATE INDEX "PetriNET\_route\_busstop\_busstop\_id\_54360cb4" ON "PetriNET\_route\_busstop" ("busstop\_id");

CREATE INDEX "PetriNET\_busstop\_city\_id\_92539046" ON "PetriNET\_busstop" ("city\_id");

COMMIT;