

Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования  
**«Владимирский государственный университет**  
**имени Александра Григорьевича и Николая Григорьевича Столетовых»**  
(МИ ВлГУ)

Факультет ИТР  
Кафедра ПИН

***КУРСОВАЯ***  
***РАБОТА***

по курсу Системы управления базами данных

на тему: «Информационная система для учета пассажироперевозок»

Руководитель

доц. каф. ПИН

(уч. степень, звание)

Колпаков А.А.

(фамилия, инициалы)

(подпись)

(дата)

Члены комиссии

Студент ПИН-123

(группа)

(подпись)

(Ф.И.О.)

Зонин М.Д.

(фамилия, инициалы)

(подпись)

(Ф.И.О.)

(подпись)

(дата)

В данной курсовой работе разработана информационная система для учёта пассажироперевозок. Реализованы проектирование базы данных, анализ предметной области, создание хранимых процедур и триггеров в PostgreSQL. Проведено тестирование системы. Работа включает 7 иллюстраций, 7 приложений, 7 таблиц. Использовано 7 источников.

In this course work, an information system has been developed for passenger transportation accounting. Implemented database design, domain analysis, creation of stored procedures and triggers in PostgreSQL. The system has been tested. Does the job include ? illustrations, ? applications, ? tables'. Used ? sources.

## **Содержание**

Введение.....	1
1. А .....	1
2. Б.....	1
3. В .....	1
3.1 в .....	11
3.1 в .....	11
3.3 в .....	11
3.4 в .....	11
3.5 в .....	11
4. Г .....	11
5. Д .....	11
6. Е .....	11
6.1 е .....	11
6.2 е .....	11
6.3 е .....	11
Заключение .....	11
Список литературы .....	11
Приложения .....	11

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.	Зонин М.Д.			
Пров.	Колпаков А.А.			
Н. контр.				
Утв.				

**МИВУ 09.03.04 ПЗ**

Информационная система для учета  
пассажироперевозок

Лит.	Лист	Листов
У	2	76

**МИ ВлГУ  
ПИН-123**

## **Введение**

Информационные системы являются важнейшим инструментом управления в современных организациях, позволяя автоматизировать процессы, связанные с обработкой, хранением и анализом данных. Одной из сфер, где применение таких систем особенно актуально, являются пассажирские перевозки. В условиях увеличивающегося числа маршрутов, пассажиров и транспортных средств традиционные методы учета (бумажные журналы, разрозненные электронные таблицы) становятся малоэффективными и подверженными ошибкам.

Информационная система для учета пассажироперевозок предназначена для автоматизации процессов продажи билетов, управления автобусными маршрутами, учета пассажиров и водителей. Она обеспечивает формирование отчетности по выручке, количеству пассажиров, использованию транспортных средств и эффективности работы персонала. Применение такой системы позволяет повысить прозрачность деятельности компании, сократить количество ошибок при обработке информации и ускорить обслуживание клиентов.

Предметной областью данной курсовой работы является организация рейсовых перевозок пассажиров автобусами из города Мурома во Владимир, Москву, Нижний Новгород, Рязань, а также на промежуточные остановки. Для дальних маршрутов предусмотрено использование двух водителей, что требует учета графиков их работы и времени отдыха. Кроме того, стоимость билета зависит от конечного пункта назначения пассажира, что также должно учитываться при реализации функционала информационной системы.

Цель работы — разработка информационной системы, позволяющей автоматизировать учет пассажироперевозок, включая хранение сведений об автобусах, маршрутах, пассажирах, водителях, продаже билетов и отчетах.

Объект исследования — процесс организации пассажирских перевозок.

Предмет исследования — методы проектирования и реализации базы данных, обеспечивающей хранение и обработку информации о пассажироперевозках.

Изм.	Лист	№ докум.	Подп.	Дата

Для реализации проекта используется система управления базами данных PostgreSQL, обладающая высокой надежностью, гибкостью и широким набором инструментов для разработки. Клиентская часть информационной системы разработана в виде настольного приложения на языке C# с использованием технологии Windows Forms, что обеспечивает удобный и интуитивно понятный пользовательский интерфейс для взаимодействия с базой данных.

Курсовая работа состоит из следующих разделов:

Анализ технического задания — формулировка целей и задач, постановка требований к системе;

Разработка моделей данных — построение концептуальной, логической и физической моделей базы данных;

Разработка и реализация автоматизированной информационной системы — описание структуры базы данных, создание таблиц, связей, процедур и триггеров;

Тестирование системы — проверка корректности выполнения операций и формирования отчетов;

Изм.	Лист	№ докум.	Подп.	Дата

## **1. Анализ технического задания**

Цель технического задания — разработка информационной системы (ИС) для автоматизации учёта пассажирских перевозок автобусами (маршруты Муром в Владимир / Москва / Нижний Новгород / Рязань и промежуточные остановки), состоящей из клиентского приложения с графическим интерфейсом на C# (Windows Forms) и реализацией базы данных в PostgreSQL, пользовательских процедур/триггеров и набора отчётов для аналитики деятельности перевозчика.

Исходные данные:

1. Организация выполняет рейсовые перевозки автобусов из Мурома в указанные города; автобусы делают промежуточные остановки — продажа билетов возможна на полный маршрут и на отдельные участки.
2. Используются разные автобусы с различным количеством посадочных мест.
3. Для маршрута требуется хранить: номер автобуса (или рег. номер), пункт отправления и назначения, номер маршрута, время отправления/прибытия, время в пути, список промежуточных остановок и время прибытия на каждую из них, количество мест / количество свободных мест.
4. Пассажир: ФИО, номер паспорта, год рождения, конечная станция (до какой станции едет).
5. Продажа билетов: регистрационный номер (id билета), пассажир, автобус/конкретный рейс, место (номер кресла), дата отправления (дата и время рейса), дата продажи, стоимость.
6. На дальние маршруты (в частности — в Москву) в автобусе в рейсе должен быть два водителя (туда и обратно). Для остальных маршрутов необходимо контролировать время отдыха водителя при назначении обратного рейса.

Изм.	Лист	№ докум.	Подп.	Дата

7. В БД требуется хранить изображения (минимум одно поле для изображений — например фото автобуса).
8. В БД должны быть реализованы хранимые процедуры и/или триггеры для реализации бизнес-логики (продажа билета, контроль мест, ограничения по водителям и т.д.).
9. В БД должно быть не менее 50 тестовых записей суммарно.
10. Используемые технологии: СУБД PostgreSQL, SQL, C#.

Для выполнения поставленной цели необходимо решить следующие задачи:

1. Выявить все ключевые сущности, включая маршруты, автобусы, пассажиров, водителей, билеты и остановки, а также определить их основные характеристики. Требуется четко определить основные бизнес-процессы системы, такие как планирование рейсов, продажа билетов, учет водителей и формирование отчетности.
2. Разработка концептуальной, логической и физической моделей данных. Начинается работа с построения ER-диаграммы для наглядного представления сущностей и связей между ними. Затем концептуальная модель преобразуется в логическую путем определения таблиц и их атрибутов. Завершается этап формированием физической модели данных с учетом особенностей СУБД PostgreSQL, включая типы данных, ограничения и индексы.
3. Проектирование структуры базы данных, где определяет состав таблиц, первичные и внешние ключи. Устанавливаются необходимые ограничения целостности данных, такие как NOT NULL, UNIQUE и CHECK. Дополнительно предусматриваются поля для хранения изображений.
4. Реализация базы данных в СУБД PostgreSQL осуществляется через написание DDL-скриптов для создания всех таблиц, связей и ограничений. Затем разработка и реализация хранимых процедур и триггеров

Изм.	Лист	№ докум.	Подп.	Дата	Лист
					МИВУ 09.03.04 ПЗ 6

для контроля количества свободных мест при продаже и отмене билетов, расчета стоимости билета в зависимости от участка маршрута, реализации логики назначения водителей с проверкой наличия двух водителей на дальних маршрутах и контроля времени отдыха, а также для других бизнес-правил, обеспечивающих целостность и корректность данных.

5. Наполнение базы данных добавляют не менее 50 суммарных записей во все таблицы для демонстрации работоспособности системы и проведения тестирования. Разработка механизмов формирования сводных отчетов через реализацию SQL-запросов для получения объема выручки за заданный период, количества перевезенных пассажиров, статистики по водителям и автобусам.
6. Тестирование разработанной информационной системы, в ходе которого проверяется корректность выполнения всех операций с данными, убеждаются в правильной работе хранимых процедур и триггеров, а также подтверждают корректность формирования всех необходимых отчетов.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

## **2. Анализ предметной области и аналогов**

### **2.1 Анализ предметной области**

Предметная область данной работы — сфера пассажирских автобусных перевозок, осуществляемых междугородними рейсами. Основной целью автоматизации является повышение эффективности учета и управления процессами, связанными с планированием маршрутов, продажей билетов, управлением автопарком и персоналом, а также формированием аналитической отчетности.

Ключевыми сущностями предметной области являются:

1. Маршруты: Определяют основные направления движения автобусов, включающие пункт отправления, пункт назначения и последовательность промежуточных остановок. Для каждого маршрута устанавливаются время отправления и прибытия, а также общая продолжительность пути.
2. Рейсы: Конкретные выполнения маршрутов в определенное время и с использованием конкретного автобуса и водителя (или водителей). Рейс характеризуется датой и временем отправления, доступным количеством мест и текущим количеством свободных мест.
3. Автобусы: Транспортные средства, используемые для перевозок. Для каждого автобуса фиксируется его идентификатор (например, номер), количество посадочных мест и, возможно, дополнительные характеристики, такие как модель или год выпуска, а также изображение.
4. Водители: Персонал, управляющий автобусами. Важными аспектами являются их ФИО и контроль за рабочим временем, особенно на дальних маршрутах, где требуется наличие второго водителя или соблюдение режима отдыха.
5. Пассажиры: Клиенты, приобретающие билеты на рейсы. Для пассажиров фиксируются основные персональные данные (ФИО, номер

Изм.	Лист	№ докум.	Подп.	Дата

паспорта, год рождения) и их конечная станция назначения в рамках выбранного рейса.

6. Билеты: Документы, подтверждающие право пассажира на проезд. Билет содержит информацию о пассажире, выбранном рейсе, номере места, дате отправления и дате продажи, а также стоимости. Стоимость билета зависит от длительности поездки (расстояния между пунктами).
7. Остановки: Географические пункты по маршруту следования автобуса, где происходит посадка и высадка пассажиров. Промежуточные остановки влияют на формирование стоимости билета и логистику рейса.

Основными бизнес-процессами, подлежащими автоматизации, являются:

1. Формирование и управление маршрутами и рейсами: Определение расписания, назначение автобусов и водителей на рейсы.
2. Продажа билетов: Оформление билетов для пассажиров с учетом свободных мест и выбора конечной станции.
3. Учет пассажиров: Сбор и хранение информации о пассажирах.
4. Управление автопарком: Учет характеристик автобусов.
5. Управление персоналом: Контроль за водителями, их графиком работы и режимом отдыха.
6. Формирование отчетности: Получение сводных данных о выручке, пассажиропотоке, эффективности использования автобусов и водителей.
7. Система должна обеспечить целостность данных, предотвращать коллизии при продаже билетов (например, двойное бронирование места) и контролировать соблюдение правил назначения водителей.

## **2.2 Анализ существующих аналогов**

Изм.	Лист	№ докум.	Подп.	Дата

Для автоматизации пассажирских перевозок используются различные программные решения:

1. АСУ «Автовокзал» – комплексная система, применяемая на крупных автовокзалах для продажи билетов, учета рейсов и анализа пассажиропотока.
2. Онлайн-сервисы (Busfor, Tutu.ru, Яндекс.Путешествия) – позволяют бронировать и покупать билеты через интернет, обеспечивая удобный доступ к расписанию и маршрутам.
3. Решения на базе 1С («1С:Транспортная логистика», «ВИТ:Автобус») – предназначены для крупных перевозчиков и включают широкий функционал: планирование рейсов, учет топлива, расчет зарплаты водителей.

Таблица 1 – Сравнительная таблица

Критерий	АСУ «Автовокзал» (крупные корпоративные решения)	Онлайн-сервисы (Busfor, Tutu.ru)	1С-решения (Транспортная логистика)	Разрабатываемая система
Целевая аудитория	Крупные автовокзалы перевозчики	Конечные потребители, крупные платформы по продаже билетов	Крупные и средние предприятия, даже сложным учреждениям	Небольшие и средние перевозчики
Функционал	Максимально полный, зачастую избыточный	Ограничечен бронированием и покупкой билетов	Широкий, интегрированный с бухгалтерией и складом	Сфокусированный на ключевых задачах: учет рейсов, билетов, водителей

Стоимость	Высокая (лицензии, внедрение, поддержка)	Комиссия с продаж или подписка перевозчика	Очень высокая (лицензии 1С, внедрение, доработки)	Низкая (бесплатная СУБД PostgreSQL)
Гибкость и кастомизация	Сложная и дорогая	Отсутствует, работа вносила правилами платформы	Высокая, лифицированная 1С-специалистов	Высокая (открытый SQL-код, возможность адаптации)
Данные и контроль	Данные принадлежат компании, полный контроль	Данные на стороне платформы, зависимость от него	Данные принадлежат компании, полный контроль	Полный контроль над данными на своем сервере
Внедрение и поддержка	Длительное, требуется привлечения специалистов	Не требуется	Длительное и дорогое	Быстрое развертывание, техподдержка силами штатного SQL-администратора

Продолжение таблицы 1

Ключевые недостатки	Дороговизна, сложность для больших компаний	Нет внутреннего учета, комиссия, зависимость от ком	Сложность, избыточность функционала, цена	Ориентация на средний и малый бизнес, отсутствие GUI
---------------------	---	---	---	--

Ключевые преимущества	Полнофункциональность, интеграция с оборудованием	Широкая аудитория, пристота для пассажира	Мощная отчетность, интеграция с 1С	Простота, низкая стоимость, целевой функционал, полный контроль
-----------------------	---	---	------------------------------------	---

Проведенный анализ показывает, что существующие решения на рынке обладают двумя ключевыми недостатками, которые открывают нишу для разрабатываемой системы:

1. Высокая стоимость и сложность: Крупные системы типа АСУ «Автовокзал» или 1С предназначены для предприятий с большими объемами перевозок и соответствующими бюджетами. Их функционал зачастую избытен для компаний, работающей на нескольких междугородних маршрутах. Затраты на лицензии, внедрение и поддержку делают их нерентабельными для малого и среднего бизнеса.
2. Зависимость и отсутствие контроля: Онлайн-платформы по продаже билетов (Busfor, Tutu.ru) решают проблему продаж, но полностью лишают перевозчика контроля над данными и процессами. Компания не может построить индивидуальную систему учета и отчетности под свои нужды, а также вынуждена платить комиссию с каждого проданного билета.

Разрабатываемая информационная система предлагает принципиально иной подход, обеспечивая следующие конкурентные преимущества:

1. Экономическая эффективность: Использование бесплатной и мощной СУБД PostgreSQL сводит стоимость владения к минимуму (затраты только на серверное оборудование и работу администратора). Это критически важно для небольших компаний с ограниченным бюджетом на ИТ.
2. Целевой функционал: Система не перегружена ненужными модулями. Она решает именно тот круг задач, который актуален для предметной

области: управление рейсами, продажа билетов с учетом промежуточных остановок, контроль за водителями и формирование отчетности. Это делает ее простой в освоении и использовании.

3. Полная независимость и контроль данных: Все данные хранятся на собственных серверах перевозчика. Компания не зависит от внешних платформ, не платит им комиссию и может быть уверена в сохранности и конфиденциальности своей информации.
4. Гибкость и адаптируемость: поскольку система строится на реляционной БД с открытым SQL-кодом (триггеры, процедуры), ее бизнес-логику можно точно настроить под уникальные требования заказчика. Например, можно легко изменить алгоритм расчета стоимости или правила контроля отдыха водителей без покупки дорогостоящих обновлений.
5. Решение конкретных проблем предметной области: в отличие от платформ по продаже билетов, система имеет встроенную логику для: Контроля назначения двух водителей на дальние рейсы. Проверки времени отдыха водителя перед назначением на обратный рейс. Расчета стоимости билета между промежуточными остановками. Гарантии отсутствия двойного бронирования мест.

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

### **3. Разработка моделей данных**

### **3.1 Концептуальная модель**

На этапе концептуального проектирования создается высокоуровневая модель данных, которая описывает основные сущности предметной области, их атрибуты и взаимосвязи. Эта модель не зависит от конкретной системы управления базами данных и служит для формализации и согласования понимания предметной области.

Для визуализации концептуальной модели была разработана ER-диаграмма в нотации Чена (см. Рисунок 1). В данной нотации сущности представлены прямыми угольниками, атрибуты — овалами, а связи между сущностями — ромбами. Ключевые атрибуты, уникально идентифицирующие экземпляры сущностей, подчеркнуты.



Рисунок 1 – Концептуальная модель

в концептуальной модели не понятно кто входит в систему, где-то должен быть логин пароль, в логической идея такая же, пассажир full name не должно быть ни где, отдельно имя фамилия отчество, у рейса не понятно время отправки есть дата есть, а прибытия не понятно

Изм.	Лист	№ докум.	Подп.	Дата

Автовгзал может быть как промежуточной остановке, проверить все поля, должен понимать, что ты там хранишь, маршрут табличка, есть остановки стопс и остановки на маршруте, что значит.

Каждое поле хранить отдельно, какие описывают, физическую переделать в связи с изменениями логической и тд.

SQL запросы, разработка триггеров и хранимых процедур, та часть которая отвечает за автоматизацию.

Основные сущности модели:

- Маршрут: Абстрактное описание пути следования (например, «Муром-Москва»).
- Остановка: Конкретный географический пункт на маршруте.
- Рейс: Конкретное выполнение маршрута в определенное время на определенном автобусе.
- Автобус: Транспортное средство с определенными характеристиками.
- Водитель: Член экипажа, управляющий автобусом.
- Пассажир: Клиент, приобретающий билет.
- Билет: Документ, подтверждающий право на проезд.
- Сотрудник: Пользователь системы (кассир, менеджер), который выполняет операции по продаже билетов и управлению рейсами.

Ключевые связи и их кардинальность:

- Маршрут Включает Остановку (M:N): Связь «многие-ко-многим» означает, что один маршрут состоит из многих остановок, и одна и та же остановка может входить в состав разных маршрутов.
- Маршрут Реализуется в Рейсах (1:N): Связь «один-ко-многим» показывает, что по одному маршруту может выполняться множество рейсов, но каждый рейс привязан только к одному маршруту.

Изм.	Лист	№ докум.	Подп.	Дата

- Автобус Выполняет Рейс (1:N): Связь «один-ко-многим» показывает, что один автобус из автопарка может быть назначен на выполнение множества рейсов (в разное время), однако каждый конкретный рейс выполняется строго на одном автобусе.
- Рейс Управляет Водителем (M:N): Связь «многие-ко-многим», так как на одном рейсе (особенно дальнем) может быть несколько водителей, и один водитель может работать на многих рейсах.
- Рейс Продажа на Билет (1:N): На один рейс продается много билетов, но каждый билет действителен только для одного рейса.
- Пассажир Покупает Билет (1:N): Один пассажир может приобрести множество билетов (например, на разные рейсы или для группы), но каждый конкретный билет регистрируется на одного и только одного пассажира.
- Сотрудник Оформляет Билет (1:N): Один сотрудник (кассир) может оформить множество билетов, но каждый проданный билет связан только с одним сотрудником, который его оформил.
- Сотрудник Создает Рейс (1:N): Один сотрудник (менеджер) может создать и запланировать множество рейсов, но каждый рейс регистрируется в системе одним ответственным сотрудником.

### **3.2 Логическая модель**

Логическая модель данных является следующим шагом после концептуального проектирования и представляет собой детальное описание структуры будущей базы данных без привязки к конкретной СУБД. На этом этапе абстрактные сущности и связи преобразуются в конкретные таблицы, атрибуты, первичные и внешние ключи. Целью логической модели является приведение структуры данных к нормализованному виду для обеспечения их целостности и минимизации избыточности.

Изм.	Лист	№ докум.	Подп.	Дата

Визуальным представлением логической модели является диаграмма схемы данных (см. Рисунок 2), на которой подробно показаны все таблицы, их поля и связи, реализованные через механизм внешних ключей.

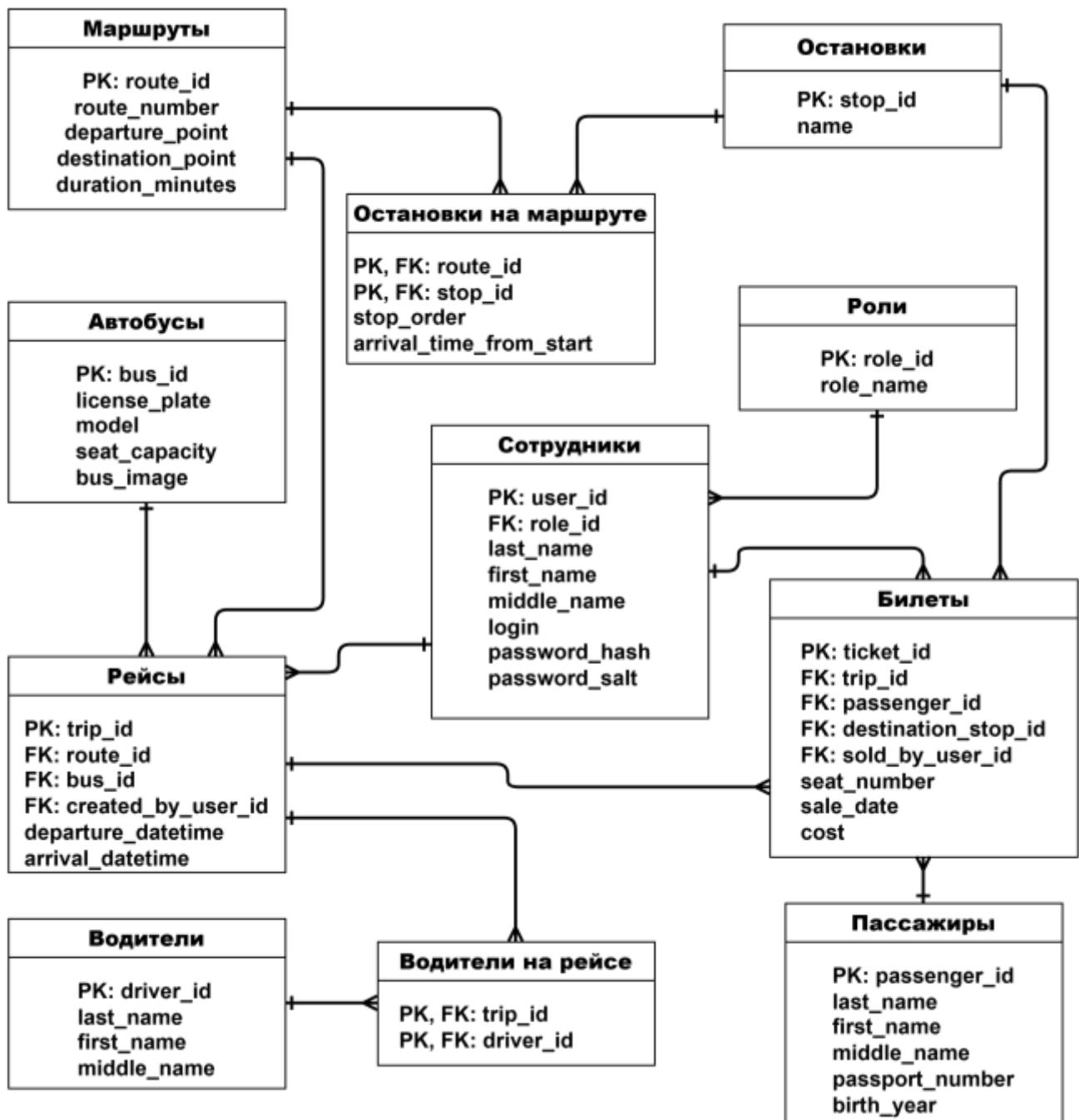


Рисунок 2 – Логическая модель

Описание структуры таблиц и их назначения:

Для обеспечения гибкости и минимизации дублирования данных информация разделена на справочные таблицы (хранищие относительно статичные дан-

Изм.	Лист	№ докум.	Подп.	Дата

ные, например, список водителей или автобусов) и операционные таблицы (фиксющие события, например, создание рейса или продажу билета).

## 1. Справочные таблицы:

– Роли (Roles): Таблица вынесена для централизованного управления ролями пользователей («кассир», «менеджер»). Это позволяет легко добавлять новые роли или изменять названия существующих без модификации основной таблицы сотрудников.

– Сотрудники (Users): Хранит учетные данные пользователей для аутентификации в системе и разграничения прав доступа. user\_id (PK) является уникальным идентификатором, на который ссылаются другие таблицы для отслеживания действий сотрудников.

– Маршруты (Routes), Остановки (Stops), Автобусы (Buses), Водители (Drivers), Пассажиры (Passengers): Эти таблицы служат едиными справочниками для соответствующих сущностей, что исключает повторный ввод одних и тех же данных и обеспечивает их согласованность во всей системе.

## 2. Операционные таблицы:

– Рейсы (Trips): Ключевая таблица, фиксирующая каждое конкретное выполнение рейса.

trip\_id (PK): Уникальный идентификатор, позволяющий однозначно ссылаться на данный рейс.

route\_id (FK): Устанавливает связь с таблицей Маршруты, чтобы определить, по какому именно маршруту (например, «Муром-Москва») выполняется данный рейс.

bus\_id (FK): Указывает, какой конкретный автобус из автопарка назначен на этот рейс.

created\_by\_user\_id (FK): Связывает рейс с сотрудником-менеджером, чтобы отслеживать, кто несет ответственность за его создание и планирование.

– Билеты (Tickets): Фиксирует факт продажи каждого билета.

trip\_id (FK): Однозначно определяет, на какой именно рейс продан данный билет.

Изм.	Лист	№ докум.	Подп.	Дата

`passenger_id` (FK): Указывает, какому пассажиру принадлежит билет.

`destination_stop_id` (FK): Позволяет хранить конечную станцию пассажира, что необходимо для расчета стоимости и определения свободных мест на последующих участках маршрута.

`sold_by_user_id` (FK): Связывает продажу с конкретным кассиром, что важно для ведения отчетности и анализа эффективности работы персонала.

### 3. Ассоциативные (связующие) таблицы:

– Остановки на маршруте (`Route_Stops`): Эта таблица необходима для реализации сложной связи «многие-ко-многим» между Маршрутами и Остановками.

`route_id` (FK) и `stop_id` (FK) вместе определяют, какая остановка включена в какой маршрут.

Использование составного первичного ключа (`route_id`, `stop_id`) на физическом уровне гарантирует, что одну и ту же остановку нельзя дважды добавить в один маршрут, обеспечивая целостность данных.

– Водители на рейсе (`Trip_Drivers`): Реализует связь «многие-ко-многим» между Рейсами и Водителями, позволяя назначать на один рейс нескольких водителей (согласно заданию).

`trip_id` (FK) указывает, к какому рейсу относится назначение.

`driver_id` (FK) указывает, какой именно водитель назначен.

Составной первичный ключ (`trip_id`, `driver_id`) предотвращает ошибочное двойное назначение одного и того же водителя на один и тот же рейс.

## 3.3 Физическая модель

Физическая модель является финальным этапом проектирования базы данных и представляет собой ее точную схему, адаптированную для конкретной системы управления базами данных — PostgreSQL. На этом этапе определяются точные типы данных для каждого поля, создаются ограничения целостности и добавляются индексы для оптимизации производительности запросов. Физическая

Изм.	Лист	№ докум.	Подп.	Дата

модель служит прямым руководством для написания DDL-скрипта, создающего структуру базы данных.

Визуальным представлением физической модели является детализированная диаграмма схемы данных (см. Рисунок 3), на которой для каждого поля указан конкретный тип данных PostgreSQL и ключевые ограничения.

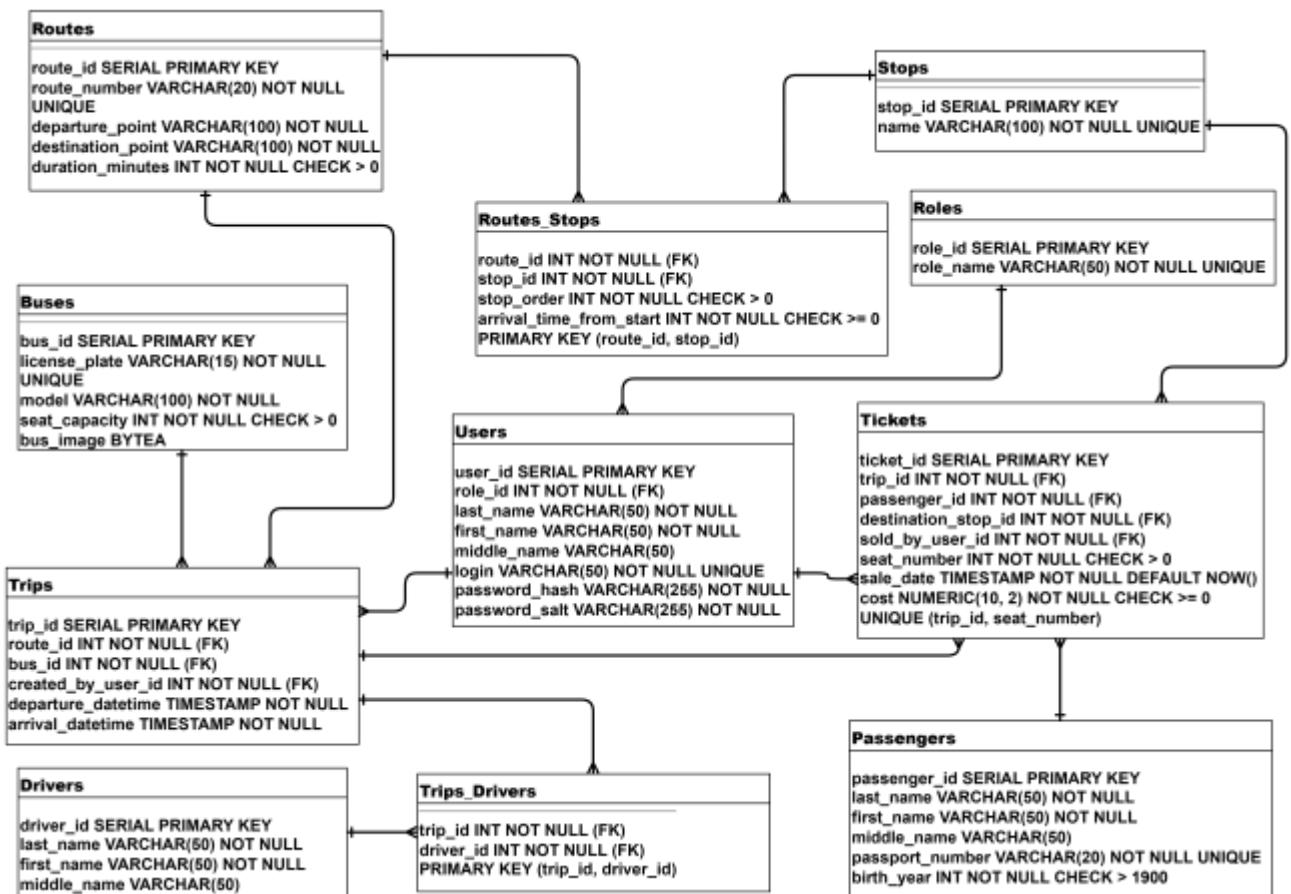


Рисунок 3 – Физическая модель

Ключевые решения, принятые при разработке физической модели:

### 1. Выбор типов данных:

Выбор оптимальных типов данных является критически важным для производительности системы и целостности информации.

- Первичные ключи (SERIAL): Для всех первичных ключей основных таблиц (user\_id, route\_id, bus\_id и т.д.) выбран тип SERIAL. Это специальный псевдотип PostgreSQL, который автоматически создает целочисленное поле (INTEGER) и привязанную к нему последовательность (SEQUENCE) для генерации уникаль-

Изм.	Лист	№ докум.	Подп.	Дата

ных, автоинкрементных значений. Данный подход является стандартным и надежным способом обеспечения уникальности записей без необходимости ручного управления идентификаторами.

– Текстовые данные (VARCHAR(n)): Для строковых полей, таких как ФИО, названия моделей или номера паспортов, используется тип VARCHAR(n) с указанием разумной максимальной длины. Это позволяет эффективно хранить данные, не резервируя лишнего места (в отличие от CHAR(n)), и обеспечивает базовую валидацию на уровне длины строки.

– Числовые данные (INTEGER, NUMERIC): Для целочисленных значений (например, seat\_capacity, duration\_minutes) используется тип INTEGER, являющийся стандартным и производительным для целых чисел. Для финансовых данных (cost) выбран тип NUMERIC(10, 2), который гарантирует точное хранение десятичных дробей без ошибок округления, что абсолютно необходимо для денежных расчетов.

– Даты и время (TIMESTAMP): Для хранения точных моментов времени, таких как отправление рейса (departure\_datetime) или продажа билета (sale\_date), используется тип TIMESTAMP. Это позволяет производить точные вычисления и фильтрацию по временным интервалам.

– Бинарные данные (BYTEA): В соответствии с требованием ТЗ, для хранения изображения автобуса (bus\_image) используется тип BYTEA (byte array), который позволяет хранить бинарные данные непосредственно в базе данных.

## 2. Ограничения целостности (Constraints):

Ограничения обеспечивают соблюдение бизнес-правил и логическую не-противоречивость данных на уровне самой базы.

– PRIMARY KEY: Для каждой таблицы определен первичный ключ, обеспечивающий уникальность каждой записи. В связующих таблицах (Routes\_Stops, Trips\_Drivers) используется составной первичный ключ, который гарантирует уникальность комбинации полей и является наиболее корректным способом реализации связей "многие-ко-многим".

Изм.	Лист	№ докум.	Подп.	Дата

– FOREIGN KEY: Для реализации связей между таблицами используются внешние ключи. Установлены правила ON DELETE RESTRICT для предотвращения удаления записей, на которые есть ссылки (например, нельзя удалить маршрут, если на него запланированы рейсы), и ON DELETE CASCADE там, где это логически оправдано (например, при удалении рейса можно автоматически удалить связанные с ним назначения водителей).

– NOT NULL: Большинство ключевых полей имеют это ограничение, что предотвращает запись в базу данных неполной или некорректной информации (например, рейс не может существовать без даты отправления).

– UNIQUE: Используется для полей, значения которых должны быть уникальны в рамках таблицы, но не являются первичным ключом. Это обеспечивает соблюдение бизнес-логики: не может быть двух автобусов с одинаковым госномером (license\_plate) или двух пассажиров с одинаковым номером паспорта (passport\_number). Также UNIQUE (trip\_id, seat\_number) в таблице Tickets предотвращает двойную продажу одного и того же места на рейс.

– CHECK: Применяется для реализации бизнес-правил на уровне базы данных. Например, CHECK (cost >= 0) не позволит установить отрицательную стоимость билета, а CHECK (seat\_capacity > 0) гарантирует, что у автобуса есть хотя бы одно место.

### 3. Индексы (Indexes):

Для повышения производительности операций поиска и соединения таблиц (JOIN) PostgreSQL автоматически создает индексы для всех первичных и уникальных ключей. Дополнительно предполагается создание индексов для всех полей, являющихся внешними ключами (FOREIGN KEY), так как по ним часто будут происходить соединения таблиц. Также целесообразно индексировать поля, которые будут часто использоваться в условиях фильтрации (WHERE), например, departure\_datetime в таблице Trips. Индексы значительно ускоряют выполнение запросов, что критически важно для отзывчивости информационной системы.

Изм.	Лист	№ докум.	Подп.	Дата

## **4. Тестирование системы**

Изм.	Лист	№ докум.	Подп.	Дата

**МИВУ 09.03.04 ПЗ**

Лист  
23

**5.**

Изм.	Лист	№ докум.	Подп.	Дата

**МИВУ 09.03.04 ПЗ**

Лист  
24

**6.**

**6.1**

**6.2**

**6.3**

Изм.	Лист	№ докум.	Подп.	Дата

**МИВУ 09.03.04 ПЗ**

Лист  
25

## **Заключение**

Изм.	Лист	№ докум.	Подп.	Дата

**МИВУ 09.03.04 ПЗ**

Лист  
26

## **Список литературы**

Изм.	Лист	№ докум.	Подп.	Дата

**МИВУ 09.03.04 ПЗ**

Лист  
27

## **Приложение А**

Ссылка на репозиторий с кодом программы

Изм.	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04 ПЗ

Лист  
28