

Создание приложения для управления базой данных городской транспортной системы

Выполнил студент группы ИУ7-65Б: Михаил Дмитриевич Коротыч

Научный руководитель: Кирилл Андреевич Кивва

Цель и задачи

Цель данной работы: реализовать базу данных системы городского общественного транспорта и приложение для удобного просмотра и манипулирования (редактирования, обновления).

Задачи:

1. Провести анализ предметной области.
2. Формализовать задание, определить необходимую функциональность.
3. Провести анализ СУБД.
4. Описать структуру базы данных, включая объекты, из которых она состоит.
5. Создать и заполнить БД.
6. Спроектировать и реализовать приложение для доступа к БД.
7. Исследовать производительность доступа к базе данных.

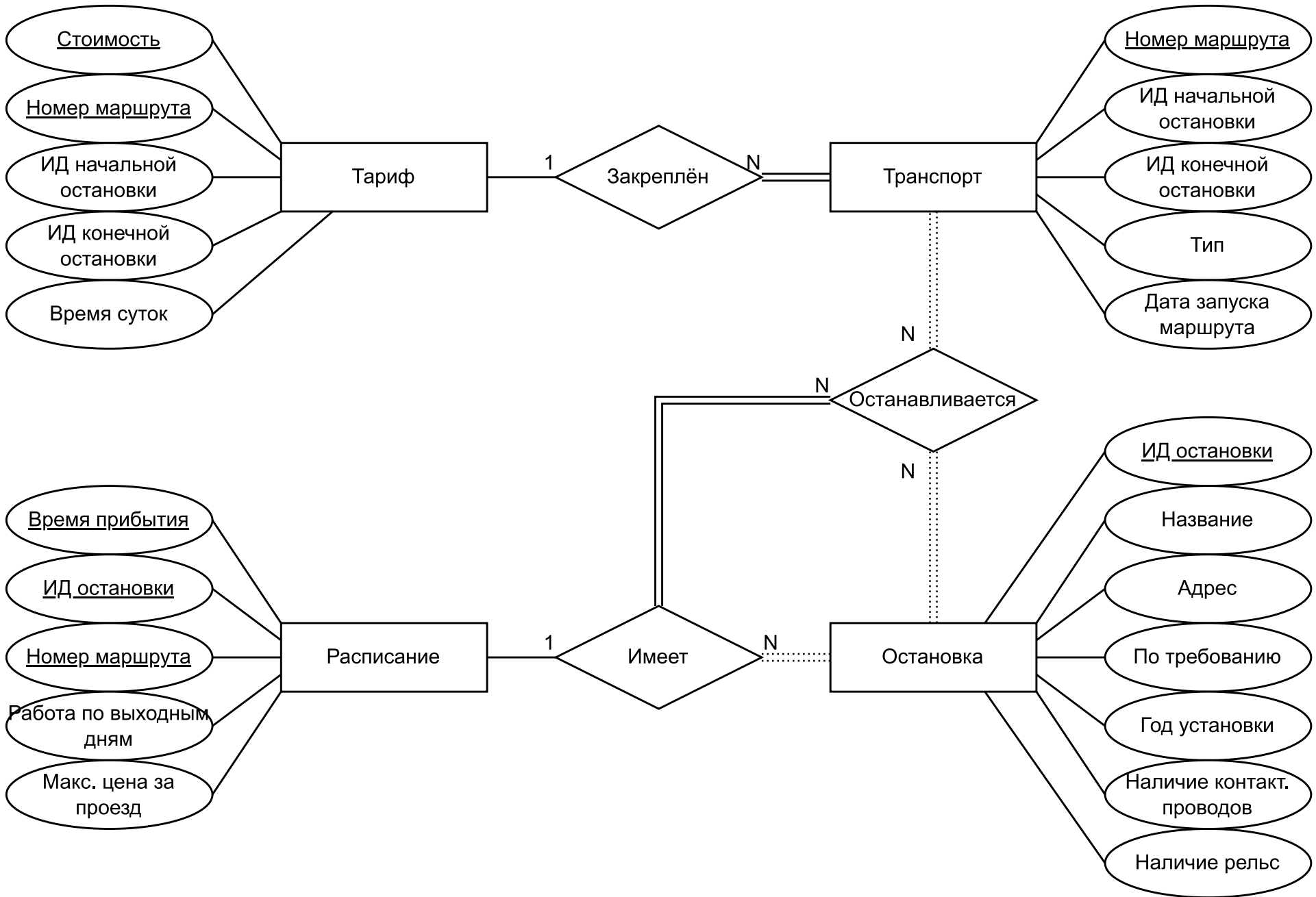
Особенности общественного транспорта

- наличие **маршрута** регулярных перевозок;
- наличие выделенных **остановок** (оборудованное для посадки, высадки пассажиров и ожидания транспортных средств);
- движение регламентируется **графиком** (расписанием), который зависит от протяжённости маршрута, частоты расположения остановок, перекрёстков;
- Наличие (но не обязательно) **оплаты** проезда.

Формализация

Категория	Сведения
Расписание	<ul style="list-style-type: none">• время прибытия;• идентификатор конечной остановки;• номер маршрута;• максимальная цена за проезд;• ходит ли маршрут по выходным дням.
Тариф	<ul style="list-style-type: none">• стоимость;• номер маршрута;• идентификатор начальной и конечной остановки;• время суток по прибытии.
Остановка	<ul style="list-style-type: none">• идентификатор остановки;• название;• адрес;• остановка по требованию;• год установки;• наличие электрической сети и рельс;
Конкретный транспорт	<ul style="list-style-type: none">• номер маршрута;• идентификаторы начальной и конечной остановки;• тип;• дата запуска маршрута;

ER-модель



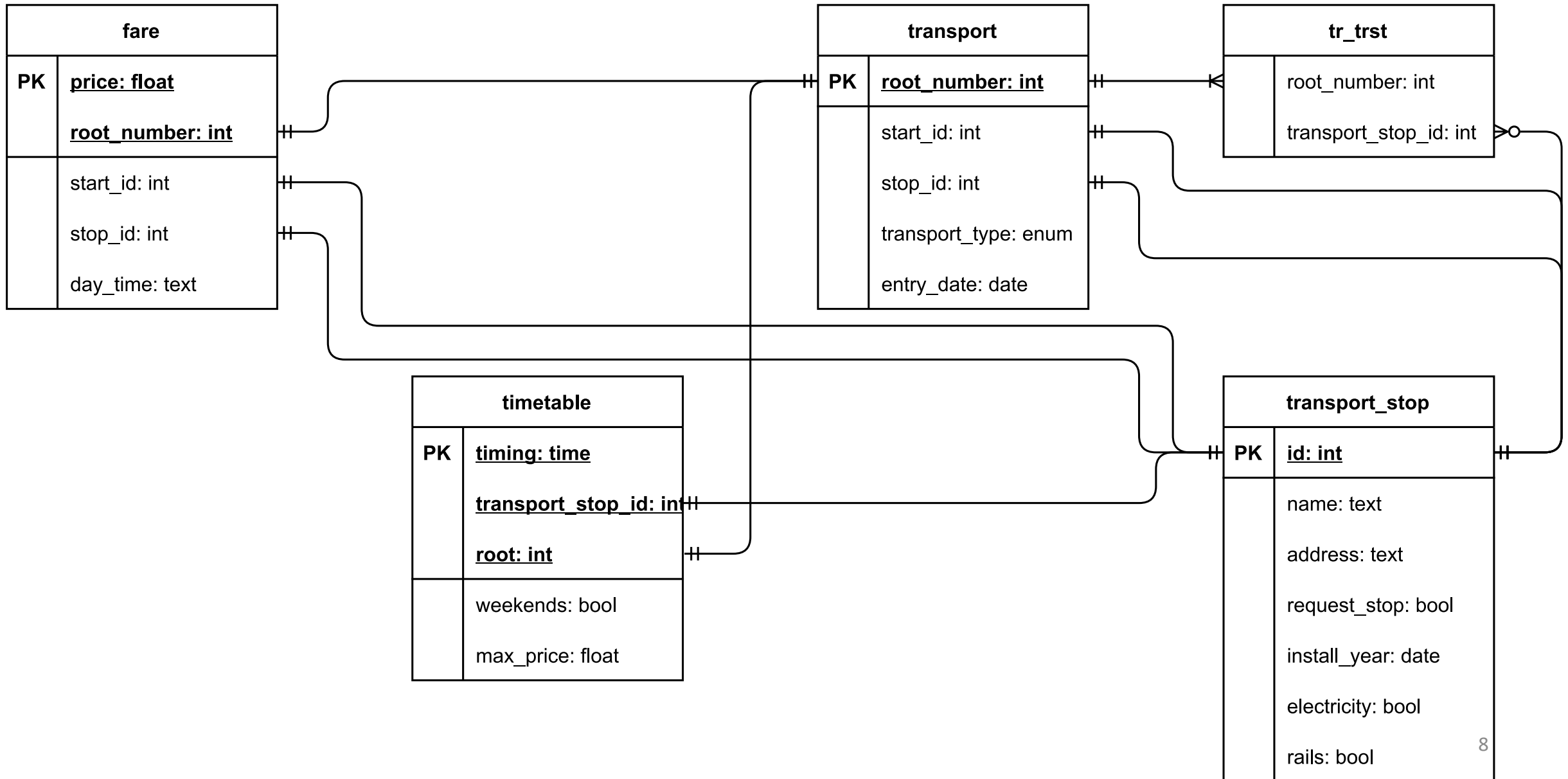
Пользователи и их права доступа

Тип пользователя	Права доступа
«Пассажир»	Просмотр информации о маршрутах, остановках, расписаниях и тарифах
«Диспетчер»	Возможность смотреть, а также изменять информацию об остановках, расписаниях, маршрутах и их тарифах
«Администратор»	Кроме всех прав доступа диспетчера может удалять и создавать новые таблицы

Сравнение моделей данных

Название модели	Достоинства	Недостатки
Иерархическая	<ul style="list-style-type: none">• строгая иерархия: один предок — много потомков;• подразумевает связь «один ко многим»;	дублирование данных для двух и более предков.
Сетевая	<ul style="list-style-type: none">• любое число предков;• типизированный набор данных;	высокая сложность системы из-за дополнительных связей
Реляционная	<ul style="list-style-type: none">• отсутствие иерархии на уровне элементов системы (таблиц);• Нет необходимости полностью знать структуру данных;	<ul style="list-style-type: none">• относительный медленный доступ к данным;• требуется относительно большой объём памяти

Схема разработанной базы данных



Model-View-Controller

MVC (Model–View–Controller, т. е. «Модель–Представление–Контроллер») — схема разделения данных и бизнес-логике по трём отдельным компонентам:

- Модель представляет данные приложения и реагирует на изменения контроллера;
- Контроллер реагирует на действия пользователя через представление, уведомляя модель об изменениях;
- Представление отображает модель пользователю.

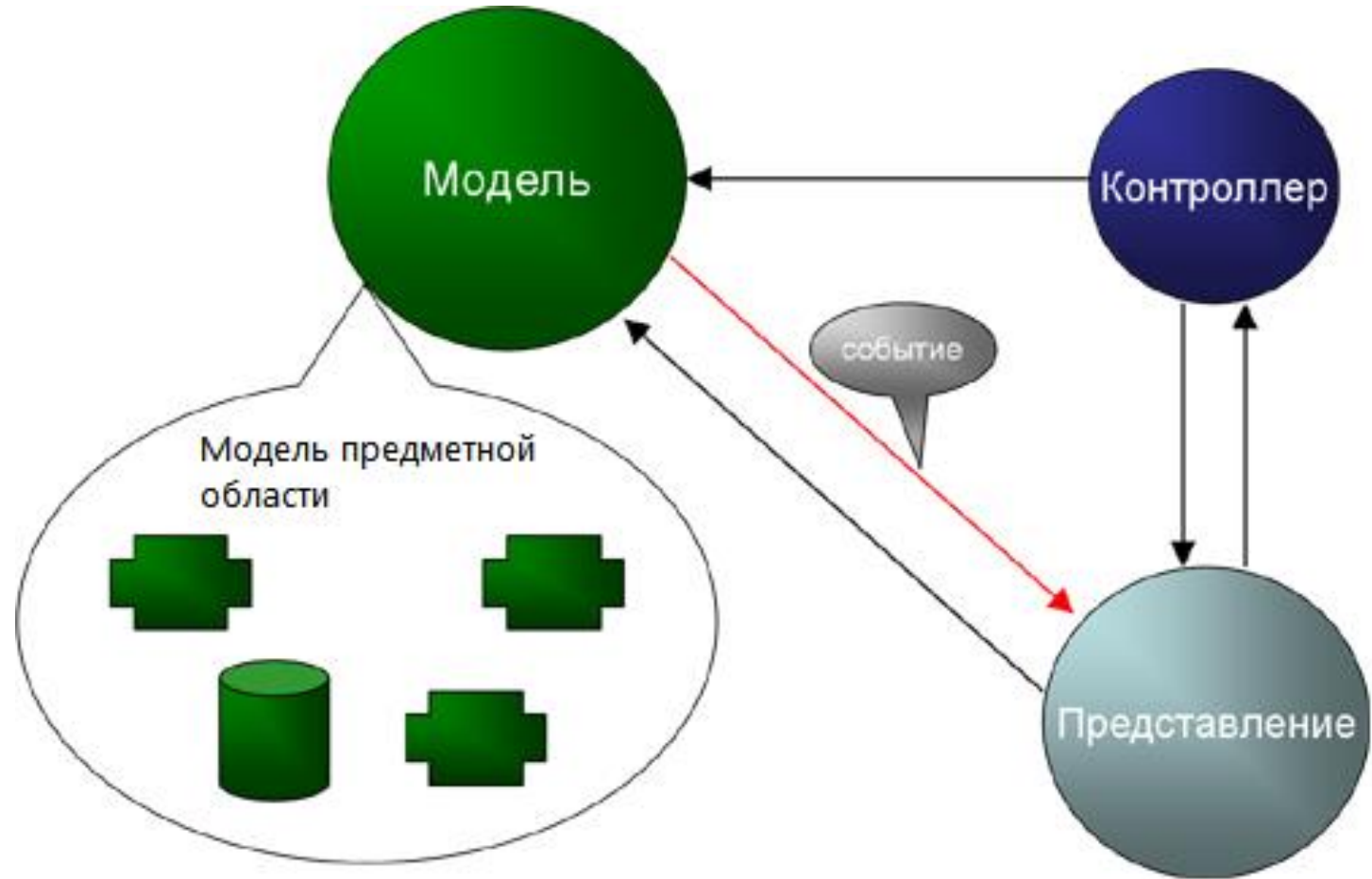
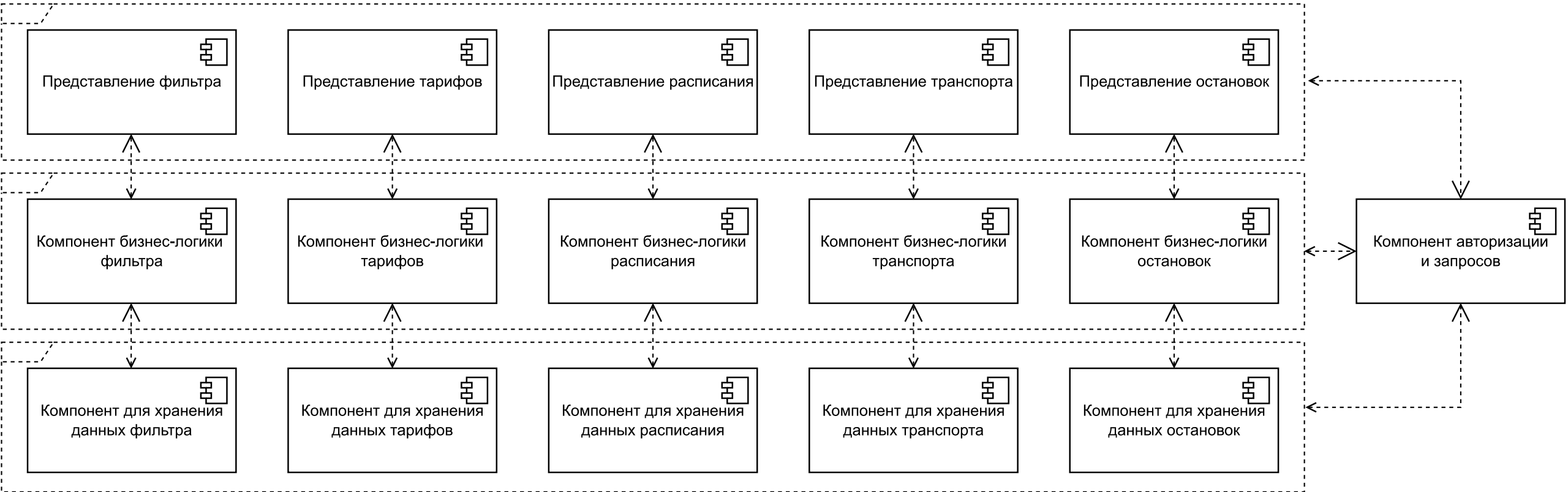
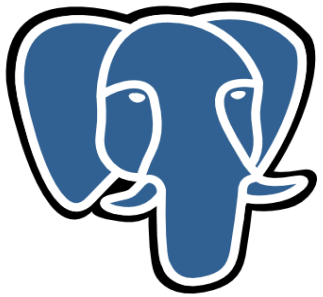


Схема компонентов



Выбор СУБД



PostgreSQL

- многоверсионный параллелизм
- объектно-реляционность
- документация на русском

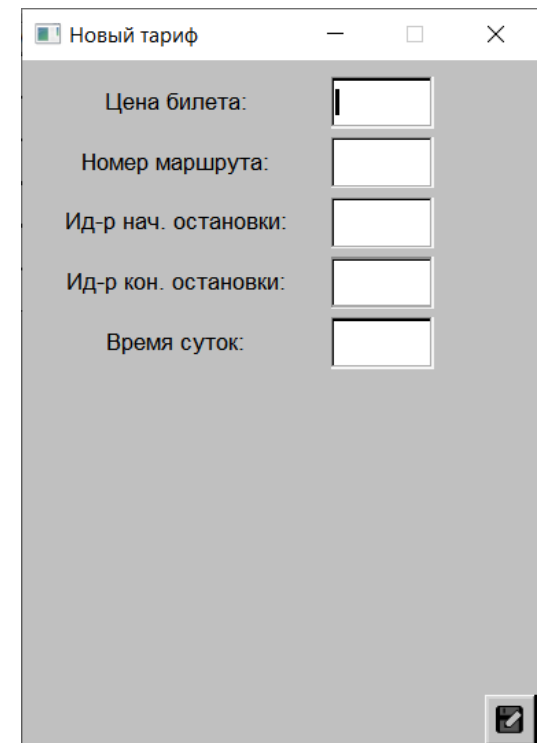
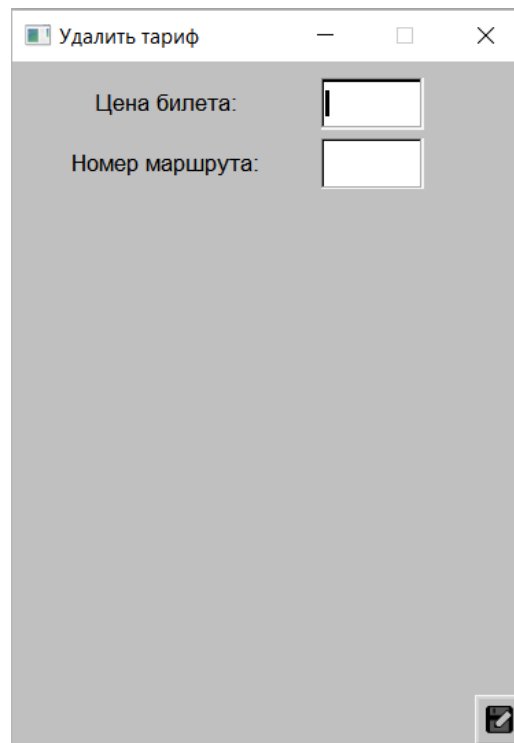
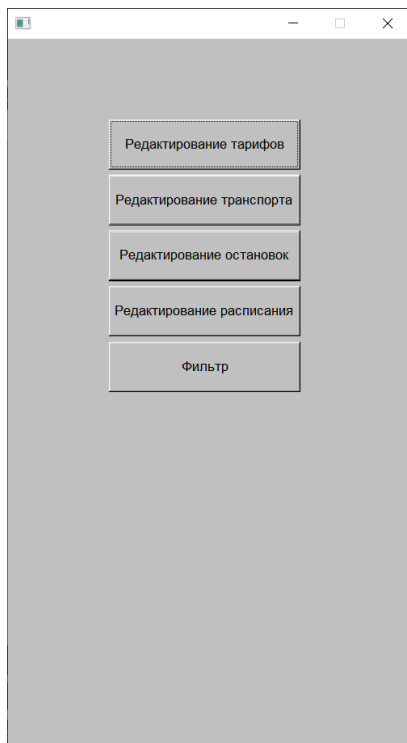


- Использование большого кол-ва параллелизма
- Быстрая работа с большими данными



- Использование большого кол-ва параллелизма
- Быстрая работа с большими данными

Интерфейс программы



Интерфейс программы

Фильтр

Номер маршрута: =

ID начальной остановки: =

ID конечной остановки: =

Тип транспорта:

Дата введения маршрута (YYYY-MM-DD): =

Найти

Номер маршрута: =

ID начальной остановки: =

ID конечной остановки: =

Цена билета: =

Время суток:

Найти

ID остановки: =

Название:

Адрес:

По требованию: ☐

Дата установки остановки (YYYY-MM-DD): =

Рельсы: ☐

Контактный провод: ☐

Найти

Номер маршрута: =

Время прибытия (ЧЧ:ММ): =

ID остановки: =

Максимальная цена за проезд: =

Ходит по выходным: ☐

Найти

Полный список тарифов

номер маршрута	Цена билета	начальной остановки	конечной остановки	Время
1	15	20	7	2 утра до 6 веч
2	26	10	1	5 тра до 10 ве
3	3	7	2	10 утра до 9 ве
4	1	17	7	6 удия до 10 в
5	2	16	2	8 утра до 6 веч
6	17	15	1	10 24 часа

Номер маршрута: Новое время суток: Поменять

Номер маршрута: Новый ид-р нач. остановки: Поменять

Номер маршрута: Новый ид-р кон. остановки: Поменять

Номер маршрута: Новая цена билета: Поменять

Проведение эксперимента

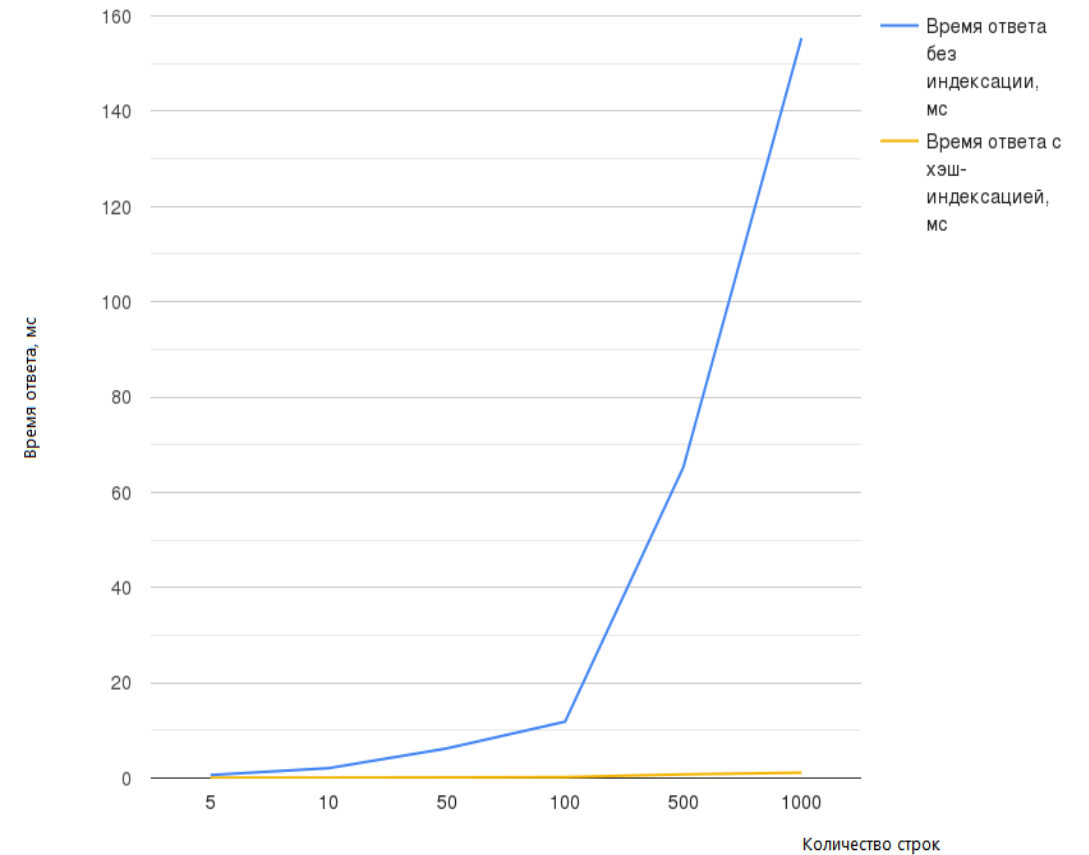
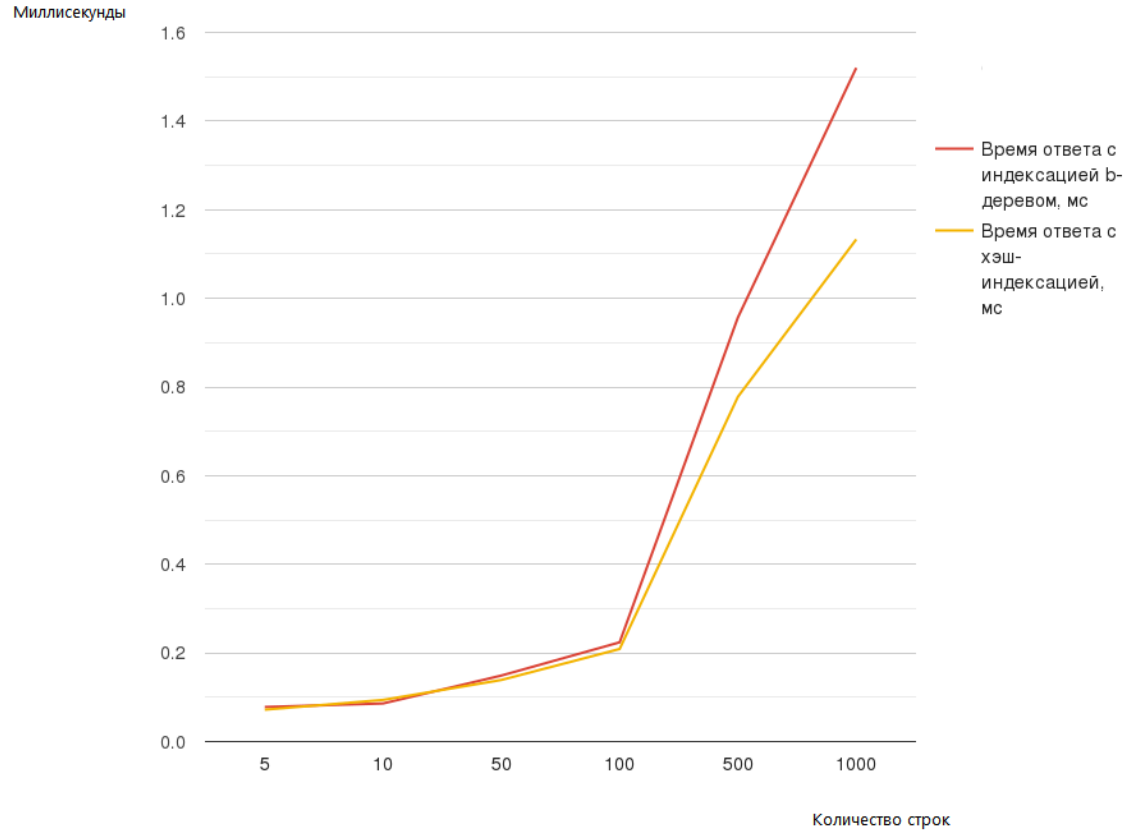
- Цель: выявление зависимости времени ответа базы данных от наличия индексации и количества строк в одной из таблиц БД.
- Типы индексов: b-дерево, хэш-индекс
- Запрос:

```
select root, max_price
from timetable
where max_price > all
(
    select max_price
    from fare
    where start_id = 123
);
```

Результаты эксперимента. Таблица

Кол-во строк	Время ответа без индексации, мс	Время ответа с индексацией деревом, мс	Время ответа с хэш- индексацией, мс
5	0.650	0.078	0.072
10	2.089	0.086	0.094
50	6.233	0.149	0.139
100	11.851	0.224	0.209
500	65.209	0.957	0.778
1000	155.330	1.520	1.330

Результаты эксперимента. График



Заключение

- Проведён анализ предметной области;
- Формализовано задание, определена необходимая функциональность;
- Проведён анализ СУБД;
- Описана структура базы данных, включая объекты, из которых она состоит;
- База данных создана и заполнена;
- Спроектировано и реализовано приложение для доступа к БД;
- Исследована производительность доступа к базе данных.

Дальнейшее развитие

- Разработка функций удаления и создания таблиц непосредственно из приложения.
- Улучшение и оптимизация графического интерфейса.