

Федеральное государственное автономное образовательное учреждение высшего  
образования  
«Санкт-Петербургский национальный исследовательский университет  
ИТМО»

**Институт ПИИКТ**

**Дисциплина: Информационные системы и базы данных**

## **Лабораторная работа №3**

Выполнил: Сиразетдинов Азат Ниязович

Группа: Р3116

Преподаватель: Горбунов

Михаил Витальевич

2023г.

# Оглавление

Задание.....	3
Выполнение .....	4
Исходная модель .....	4
Функциональные зависимости .....	4
Анализ зависимостей .....	5
1NF .....	5
2NF .....	5
3NF .....	6
НФБК .....	6
Нормализованная модель:.....	6
Денормализации .....	7
Объединение связанных таблиц .....	7
Добавление избыточных атрибутов .....	7
Функция .....	7
Триггер.....	8
Вывод.....	9

## Задание

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

Опишите функциональные зависимости для отношений полученной схемы (минимальное множество);

Приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум).

Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF;

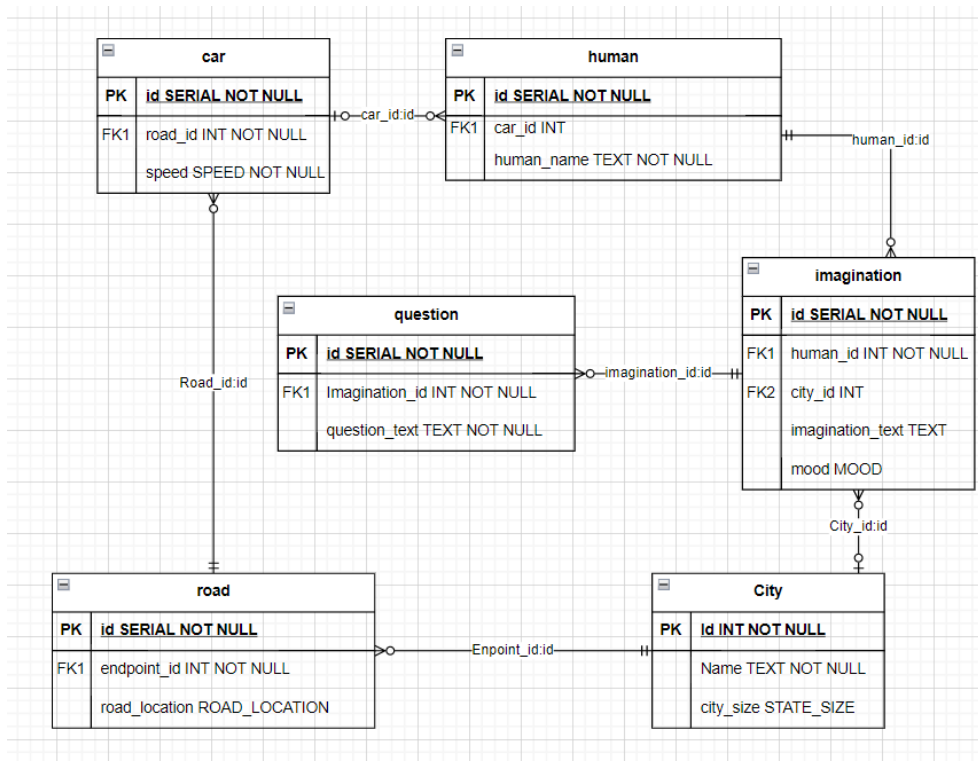
Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;

Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

# Выполнение

## Исходная модель



## Функциональные зависимости

**car:**  $id \rightarrow (speed, road\_id)$

**human:**  $id \rightarrow (car\_id, human\_name)$

**imagination:**  $id \rightarrow (human\_id, city\_id, imagination\_text),$   
 $(id, imagination\_text) \rightarrow mood$

**question:**  $id \rightarrow (imagination\_id, question\_text)$

**city:**  $id \rightarrow (name, city\_size)$

# Анализ зависимостей

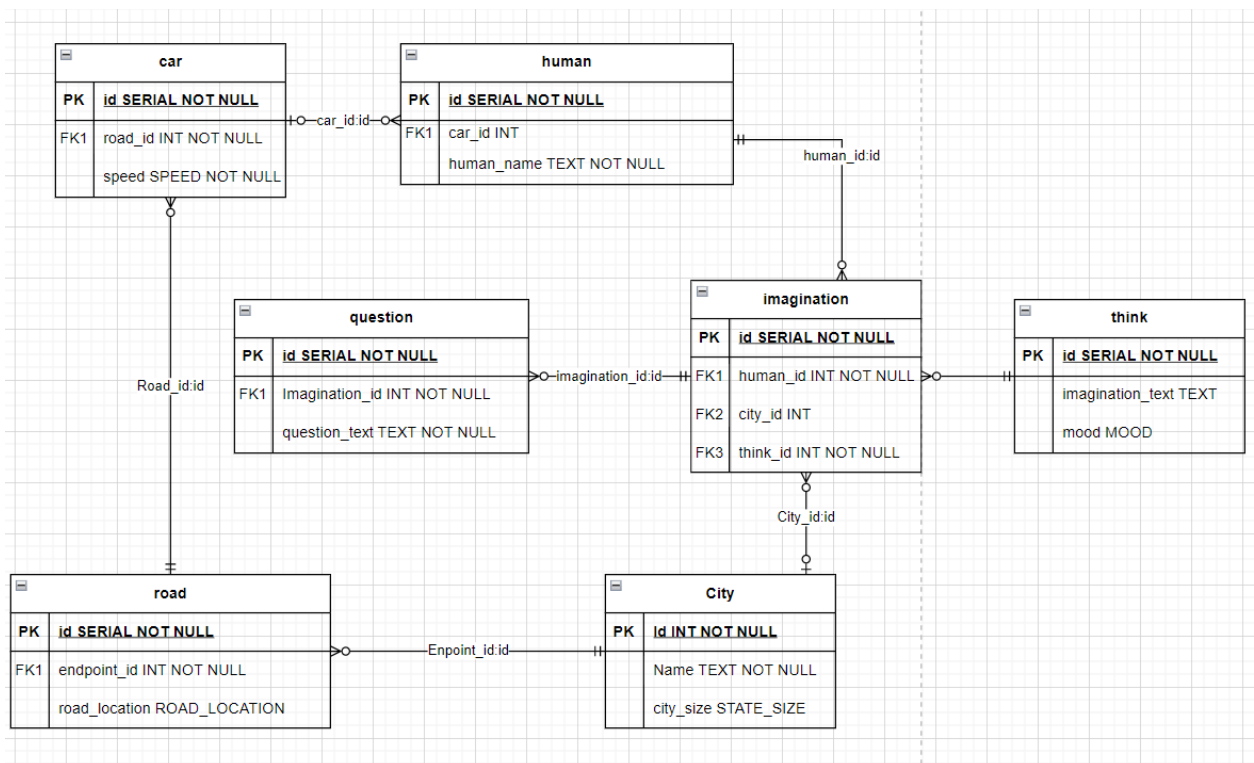
## 1NF

Отношение, на пересечении каждой строки и столбца — одно значение.  
Представленная модель удовлетворяет первой нормальной форме

## 2NF

- 1) отношение в 1НФ и
  - 2) атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения.
- Второй пункт не выполняется в таблице imagination.

Новая модель:



**car:** id → (speed, road\_id)

**human:** id → (car\_id, human\_name)

**imagination:** id → (human\_id, city\_id, think\_id)

**think:** id → (imagination\_text, mood)

**question:** id → (imagination\_id, question\_text)

**city:** id → (name, city\_size)

## 3NF

Отношение в

1) 1НФ и 2НФ и

2) все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа.

Рассматриваемая модель находится в третьей нормальной форме.

## НФБК

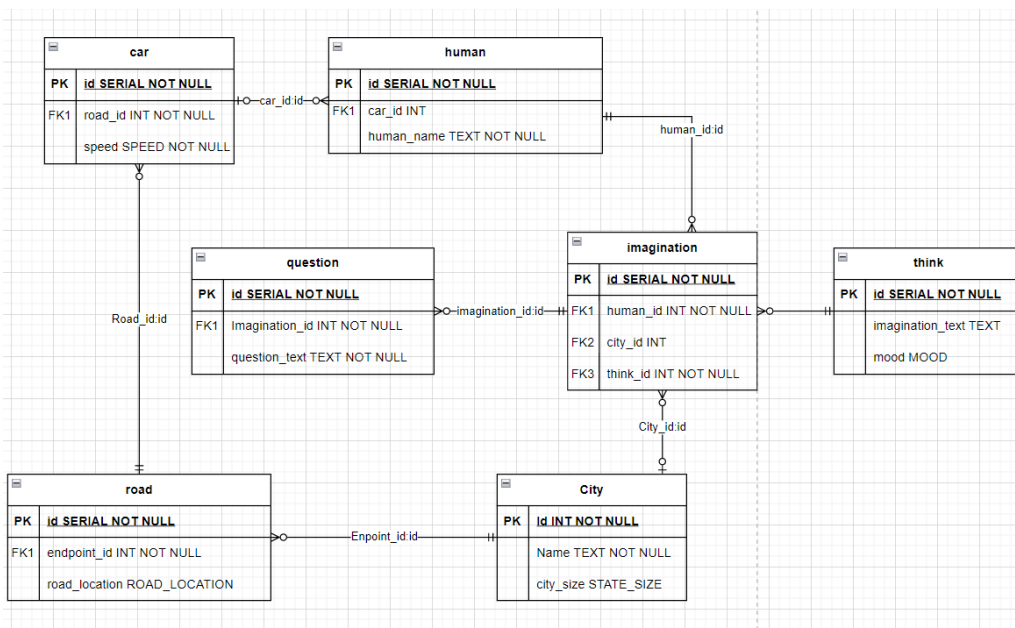
Отношение в НФБК, когда для всех

функциональных зависимостей отношения выполняется

условие: детерминант — потенциальный ключ.

Рассматриваемая модель находится в нормальной форме Бойса-Кодда

## Нормализованная модель:



**car:** id → (speed, road\_id)

**human:** id → (car\_id, human\_name)

**imagination:** id → (human\_id, city\_id, think\_id)

**think:** id → (imagination\_text, mood)

**question:** id → (imagination\_id, question\_text)

**city:** id → (name, city\_size)

# Денормализации

## *Объединение связанных таблиц*

Если в модели у каждой мысли будет ровно один вопрос (что вполне возможно, используя правило 5 вопросов «Почему?»), то стоит объединить таблицы think и question.

## *Добавление избыточных атрибутов*

Вероятно при использовании базы данных для прокладки маршрута придется искать дорогу с наименьшей загруженностью, тогда было бы удобно иметь поле car\_count в таблице road отображающее количество автомобилей на дороге, которое мы бы обновляли при добавлении машины на дорогу.

## Функция

Функция на языке PL/pgSQL для поиска дороги с минимальной загруженностью

```
CREATE OR REPLACE FUNCTION low_traffic_road_id() RETURNS integer AS $$
DECLARE
    min_road integer;
BEGIN
    IF (EXISTS (
        SELECT *
        FROM road
        LEFT JOIN car
        ON car.road_id = road.id
        WHERE car.id IS NULL
    )) THEN
        SELECT road.id INTO min_road
        FROM road
        LEFT JOIN car
        ON car.road_id = road.id
        WHERE car.id IS NULL
        LIMIT 1;
        RETURN min_road;
    ELSE
        SELECT road_id INTO min_road
        FROM car
        GROUP BY car.road_id
        ORDER BY COUNT(*)
        LIMIT 1;
        RETURN min_road;
    END IF;
END;
$$
LANGUAGE plpgsql;
```

# Триггер

Я добавил поле traffic\_load в таблицу road, которое отвечает за количество машин на дороге и написал для автоматического обновления этого атрибута триггер

```
CREATE OR REPLACE FUNCTION update_road_traffic()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF (TG_OP = 'INSERT') THEN  
        UPDATE road  
        SET traffic_load = traffic_load + 1  
        WHERE id = NEW.road_id;  
    ELSIF (TG_OP = 'DELETE') THEN  
        UPDATE road  
        SET traffic_load = traffic_load - 1  
        WHERE id = OLD.road_id;  
    END IF;  
    RETURN NULL;  
END  
$$  
LANGUAGE plpgsql;  
  
CREATE OR REPLACE TRIGGER update_road_traffic_trigger  
AFTER INSERT OR DELETE ON car  
FOR EACH ROW  
EXECUTE FUNCTION update_road_traffic();
```



## Вывод

В процессе выполнения работы я узнал про нормализацию, привел мою модель к НФБК и доказал это, написал функцию и триггер.