Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

По базам данных

Вариант №766

Выполнил:

Ступин Тимур Русланович

Группа № P3108

Преподаватель:

Афанасьев Дмитрий Борисович

Санкт-Петербург 2024

Содержание

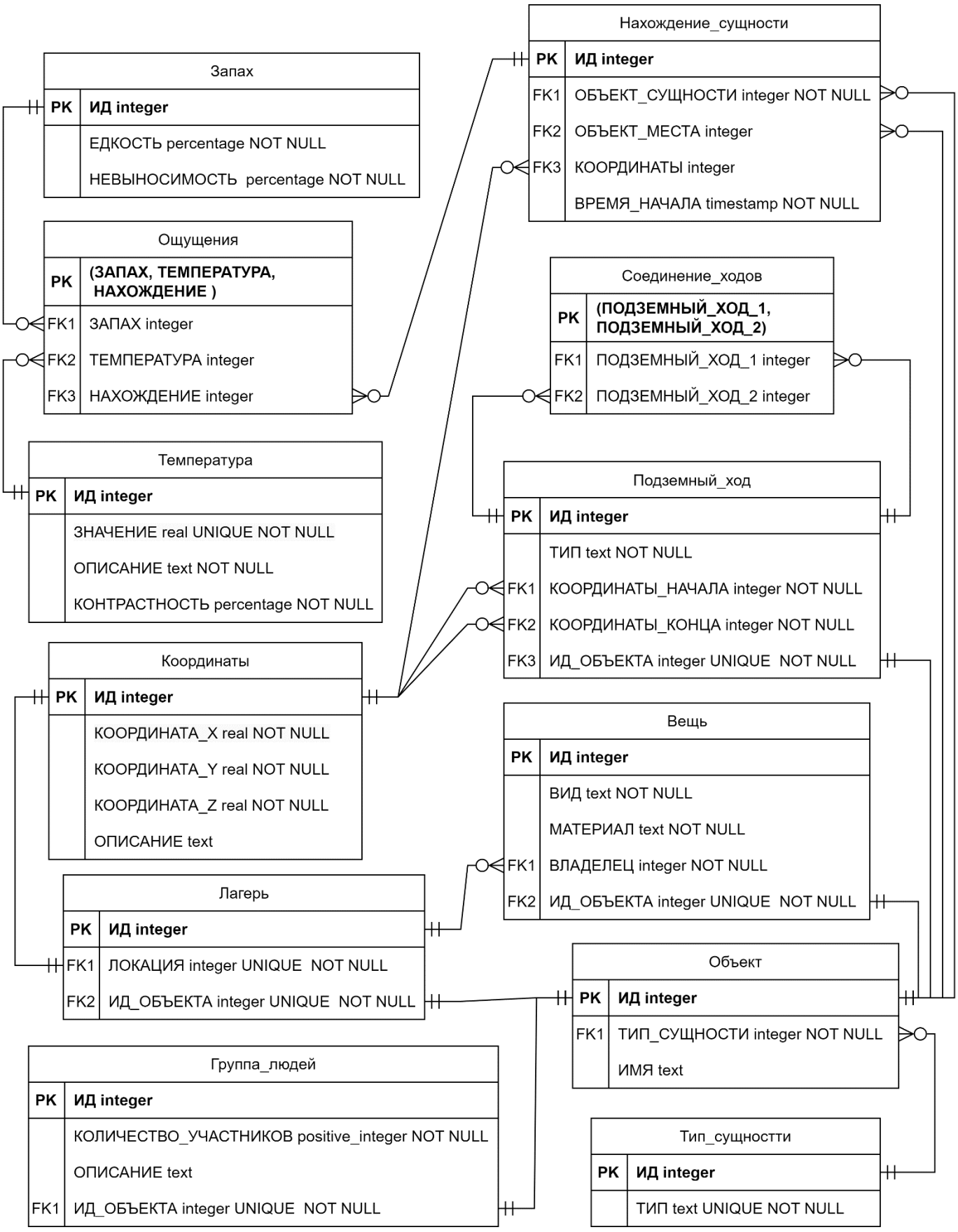
Текст задания

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

Исходная модель



Функциональные зависимости

- Группа_людей
  - ИД → (КОЛИЧЕСТВО_УЧАСТНИКОВ, ОПИСАНИЕ, ИД_ОБЪЕКТА)
- Вещь
  - ИД → (ВИД, МАТЕРИАЛ, ВЛАДЕЛЕЦ, ИД_ОБЪЕКТА)
- Подземный_ход
  - ИД → (ТИП, КООРДИНАТЫ_НАЧАЛА, КООРДИНАТЫ_КОНЦА, ИД_ОБЪЕКТА)
- Лагерь
  - ИД → (ЛОКАЦИЯ, ИД_ОБЪЕКТА)
- Запах
  - ИД → (ЕДКОСТЬ, НЕВЫНОСИМОСТЬ)
- Температура
  - ИД → ЗНАЧЕНИЕ, ОПИСАНИЕ, КОНТРАСТНОСТЬ
- Координаты
  - ИД → (КООРДИНАТА_X, КООРДИНАТА_Y, КООРДИНАТА_Z)
- Объект
  - ИД → (ТИП_СУЩНОСТИ, ИМЯ)
- Тип_сущности
  - ИД → (ТИП)
- Ощущения
  - ЗАПАХ, ТЕМПЕРАТУРА, НАХОЖДЕНИЕ → ()
- Нахождение_сущности
  - ИД → (ОБЪЕКТ_СУЩНОСТИ, ОБЪЕКТ_МЕСТА, КООРДИНАТЫ, ВРЕМЯ_НАЧАЛА)
- Соединение_ходов
  - ПОДЗЕМНЫЙ_ХОД_1, ПОДЗЕМНЫЙ_ХОД_2 → ()

Нормальные формы

1NF: Отношение находится в 1NF, если на пересечении каждой строки и столбца – одно значение и нет повторяющихся групп. Моя модель удовлетворяет условиям 1NF, так как на пересечении каждой строки и столбца находится ровно одно значение и нет повторяющихся строк

2NF: Отношение находится в 2NF, если оно находится в 1NF и атрибуты, не входящие в первичный ключ, находятся в полной функциональной зависимости от первичного ключа. Моя модель удовлетворяет условиям 2NF, так как она удовлетворяет 1NF и атрибуты, не входящие в первичный ключ, находятся в полной функциональной зависимости от первичного ключа.

3NF: Отношение находится в 3NF, если оно находиться в 1NF и 2NF и все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа. Моя модель удовлетворяет условиям 3NF, так как она удовлетворяет 1NF и 2NF и все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа.

BCNF

Отношение находится в BCNF, когда для всех функциональных зависимостей $A_1 \rightarrow A_2$ выполняется условие: A_1 — потенциальный ключ. Моя модель удовлетворяет условиям BCNF, так как для всех функциональных зависимостей A_1 является потенциальным ключом.

Денормализация

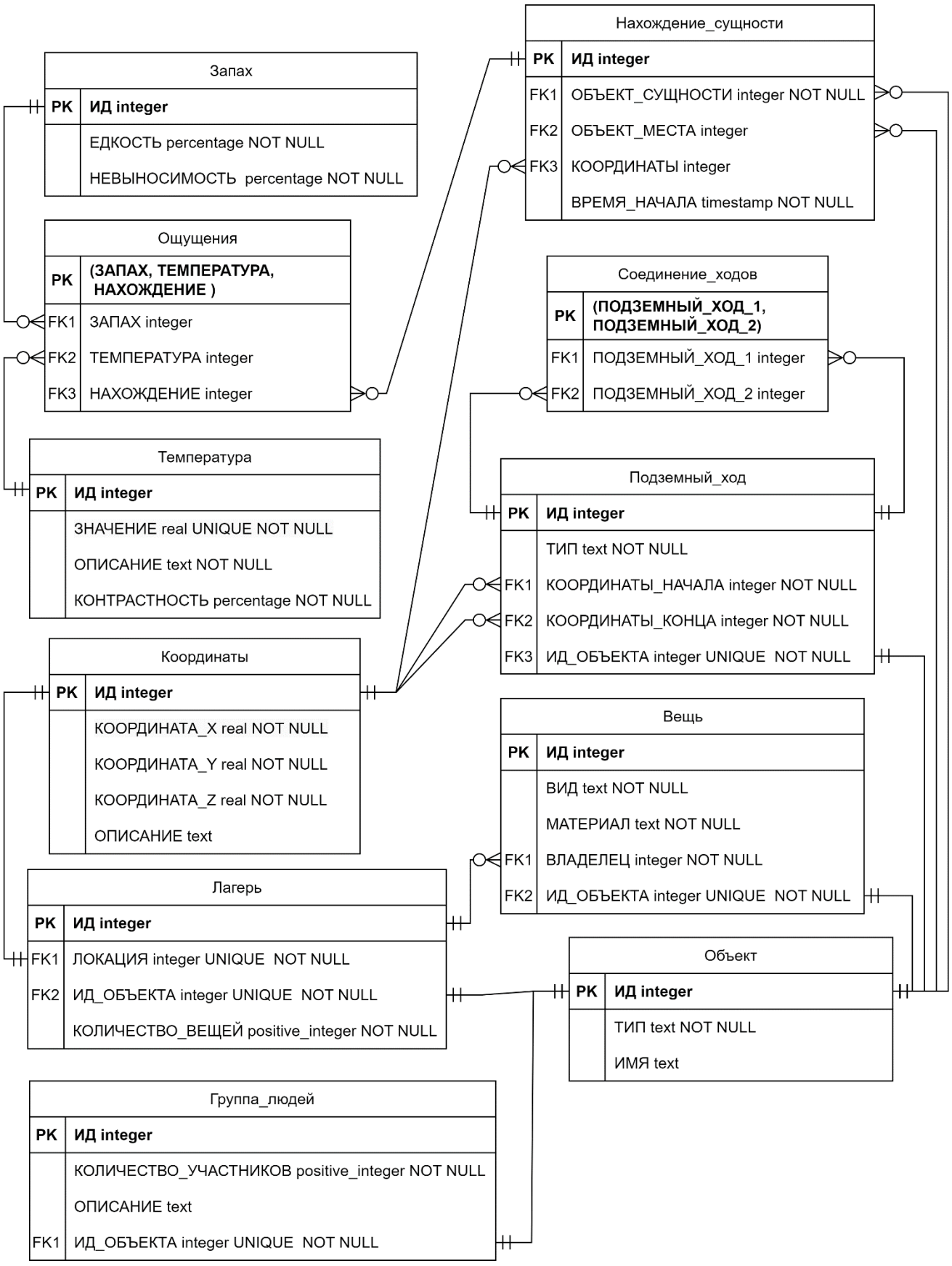
Объединение нескольких таблиц

Можно объеденить таблицы Объект и Тип_сущности, если запросы на выборку объектов заданного типа выполняются достаточно часто. Это позволит устранить одну из операций JOIN в запросе.

Добавление атрибутов

Если необходимо часто подсчитывать количество вещей, принадлежащих лагерю, то можно добавить в таблицу Лагерь поле КОЛИЧЕСТВО_ВЕЩЕЙ, что ускорит выполнение запросов, но при этом приведёт к необходимости постоянно обновлять данное значение при добавлении новых вещей.

Денормализованная модель



Функция и триггер

Данный триггер необходим для проверки ограничения целостности, а именно того факта что ощущения испытывает именно группа людей, а не некоторый другой объект.

Триггер вызывается перед операциями UPDATE или INSERT и вызывает исключение, в случае если ограничения целостности нарушены.

```
CREATE OR REPLACE FUNCTION проверка_ощущения() RETURNS trigger
LANGUAGE plpgsql AS
$$

DECLARE тип_объекта_сущности text;

BEGIN
  --Выбираем тип объекта сущности, которая испытывает ощущения
  SELECT INTO тип_объекта_сущности ТИП
  FROM Нахождение_сущности
  JOIN Объект ON Нахождение_сущности.ОБЪЕКТ_СУЩНОСТИ = Объект.ИД
  JOIN Тип_сущности ON Объект.ТИП_СУЩНОСТИ = Тип_сущности.ИД
  WHERE Нахождение_сущности.ИД = NEW.НАХОЖДЕНИЕ;

  IF тип_объекта_сущности <> 'Группа_людей' THEN
    RAISE EXCEPTION 'Объекты принадлежащие типу % не могут испытывать
ощущения', тип_объекта_сущности;
  END IF;
  RETURN NEW;
END;
$$;

CREATE TRIGGER триггер_ощущений
BEFORE UPDATE OR INSERT
ON Ощущения
FOR EACH ROW
     EXECUTE PROCEDURE проверка_ощущения();
```

Дополнительное задание

Триггер для проверки перемещения по туннелям

Проверяет корректность перемещения с учётом таблицы соединения туннелей

Триггер вызывается перед операциями UPDATE или INSERT и вызывает исключения, в случае если ограничения целостности нарушены.

```
CREATE OR REPLACE FUNCTION проверка_перемещения_по_туннелю() RETURNS
trigger
LANGUAGE plpgsql AS
$$

DECLARE
  предыдущее_место integer;
  предыдущий_туннель integer;
  текущий_туннель integer;
```

```
BEGIN
  --Если перемещение не в новый объект то выходим
  IF NEW.ОБЪЕКТ_МЕСТА IS NULL THEN
    RETURN NEW;
  END IF;

  --Проверяем что перемещение происходило именно в новый туннель
  IF (SELECT ТИП
    FROM Объект
    JOIN Тип_сущности ON Объект.ТИП_СУЩНОСТИ = Тип_сущности.ИД
    WHERE Объект.ИД = NEW.ОБЪЕКТ_МЕСТА) <> 'Подземный_ход' THEN
    RETURN NEW;
  END IF;

  --Получаем последнее указанное место в котором был объект
  SELECT INTO предыдущее_место ОБЪЕКТ_МЕСТА
  FROM Нахождение_сущности
  WHERE ОБЪЕКТ_МЕСТА IS NOT NULL
  ORDER BY ВРЕМЯ_НАЧАЛА DESC
  LIMIT 1;

  --Если это первое указанное место то выходим
  IF предыдущее_место IS NULL THEN
    RETURN NEW;
  END IF;

  --Проверяем что последнее указанное место в котором находился объект
было туннелем
  --Если это не так то выходим
  IF (SELECT ТИП
    FROM Объект
    JOIN Тип_сущности ON Объект.ТИП_СУЩНОСТИ = Тип_сущности.ИД
    WHERE Объект.ИД = предыдущее_место) <> 'Подземный_ход' THEN
    RETURN NEW;
  END IF;

  --Определяем текущий туннель
  SELECT INTO текущий_туннель Подземный_ход.ИД
  FROM Подземный_ход
  WHERE ИД_ОБЪЕКТА = NEW.ОБЪЕКТ_МЕСТА;

  --Определяем предыдущий туннель
  SELECT INTO предыдущий_туннель Подземный_ход.ИД
  FROM Подземный_ход
  WHERE ИД_ОБЪЕКТА = предыдущее_место;

  --Если текущий туннель совпал с предыдущим, то выходим
  IF текущий_туннель = предыдущий_туннель THEN
    RETURN NEW;
  END IF;

  --Если текущий и предыдущие туннели не соединены, то выбрасываем
исключение
  IF
    NOT EXISTS (
      SELECT 1
      FROM Соединение_ходов
      WHERE
        (ПОДЗЕМНЫЙ_ХОД_1 = предыдущий_туннель AND ПОДЗЕМНЫЙ_ХОД_2 =
текущий_туннель) OR
```

```
        (ПОДЗЕМНЫЙ_ХОД_2 = предыдущий_туннель AND ПОДЗЕМНЫЙ_ХОД_1 =
текущий_туннель)
    ) THEN
    RAISE EXCEPTION 'Переход между туннелями с id=% и id=% невозможен,
так как они не связаны',
    предыдущий_туннель, текущий_туннель;
  END IF;
  RETURN NEW;
END;
$$;

CREATE TRIGGER триггер_премещения_по_туннелям
BEFORE UPDATE OR INSERT
ON Нахождение_сущности
FOR EACH ROW
     EXECUTE PROCEDURE проверка_перемещения_по_туннелю();
```

Триггер проверки корректности добавляемых записей о нахождении объектов

Выполняет проверку на строгое возрастание времени начала нахождения для каждой сущности

Триггер вызывается перед операциями UPDATE или INSERT и вызывает исключения, в случае если ограничения целостности нарушены.

```
CREATE OR REPLACE FUNCTION проверка_порядка_перемещений() RETURNS trigger
LANGUAGE plpgsql AS
$$

DECLARE предыдущее_время_начала timestamp;

BEGIN
  --Определяем время последней записи для данного объекта
  SELECT INTO предыдущее_время_начала Нахождение_сущности.ВРЕМЯ_НАЧАЛА
      FROM Нахождение_сущности
      WHERE Нахождение_сущности.ОБЪЕКТ_СУЩНОСТИ = NEW.ОБЪЕКТ_СУЩНОСТИ
      ORDER BY Нахождение_сущности.ВРЕМЯ_НАЧАЛА DESC
      LIMIT 1;
  --Если новое время начала меньше или равно предыдущего то выбрасываем
исключение
  IF NEW.ВРЕМЯ_НАЧАЛА <= предыдущее_время_начала THEN
    RAISE EXCEPTION 'Попытка добавить новое нахождение для сущности %,
время начала которого (%) не больше времени начала её последнего
местанахождения (%)',
    NEW.ОБЪЕКТ_СУЩНОСТИ, NEW.ВРЕМЯ_НАЧАЛА, предыдущее_время_начала;
  END IF;
  RETURN NEW;
END;
$$;

CREATE TRIGGER триггер_нахождения_сущности
BEFORE UPDATE OR INSERT
ON Нахождение_сущности
FOR EACH ROW
     EXECUTE PROCEDURE проверка_порядка_перемещений();
```

Вывод

В ходе работы я изучил различные способы нормализации и типы нормальных форм, а также привёл свою модель в нормальной форме Бойса-Кодда. Также я ознакомился с понятием денормализации и продумал возможные изменения, которые можно внести в мою модель для увеличения производительности запросов. В процессе выполнения работы я также придумал триггер и функцию для него, которые затем реализовал на языке pgSQL.