Министерство высшего образования и науки Российской Федерации
Национальный научно-исследовательский университет ИТМО
Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

«БАЗЫ ДАННЫХ».

Вариант №696.

Выполнил:
Петров Вячеслав Маркович,
Студент группы P3108.
Преподаватель:
Афанасьев Дмитрий Борисович

Санкт-Петербург, 2024

Оглавление

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 NF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
- 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.

Даталогическая модель (исходная)

Можно сравнить начальную (Рисунок 1) и полученную (Рисунок 2) модели.

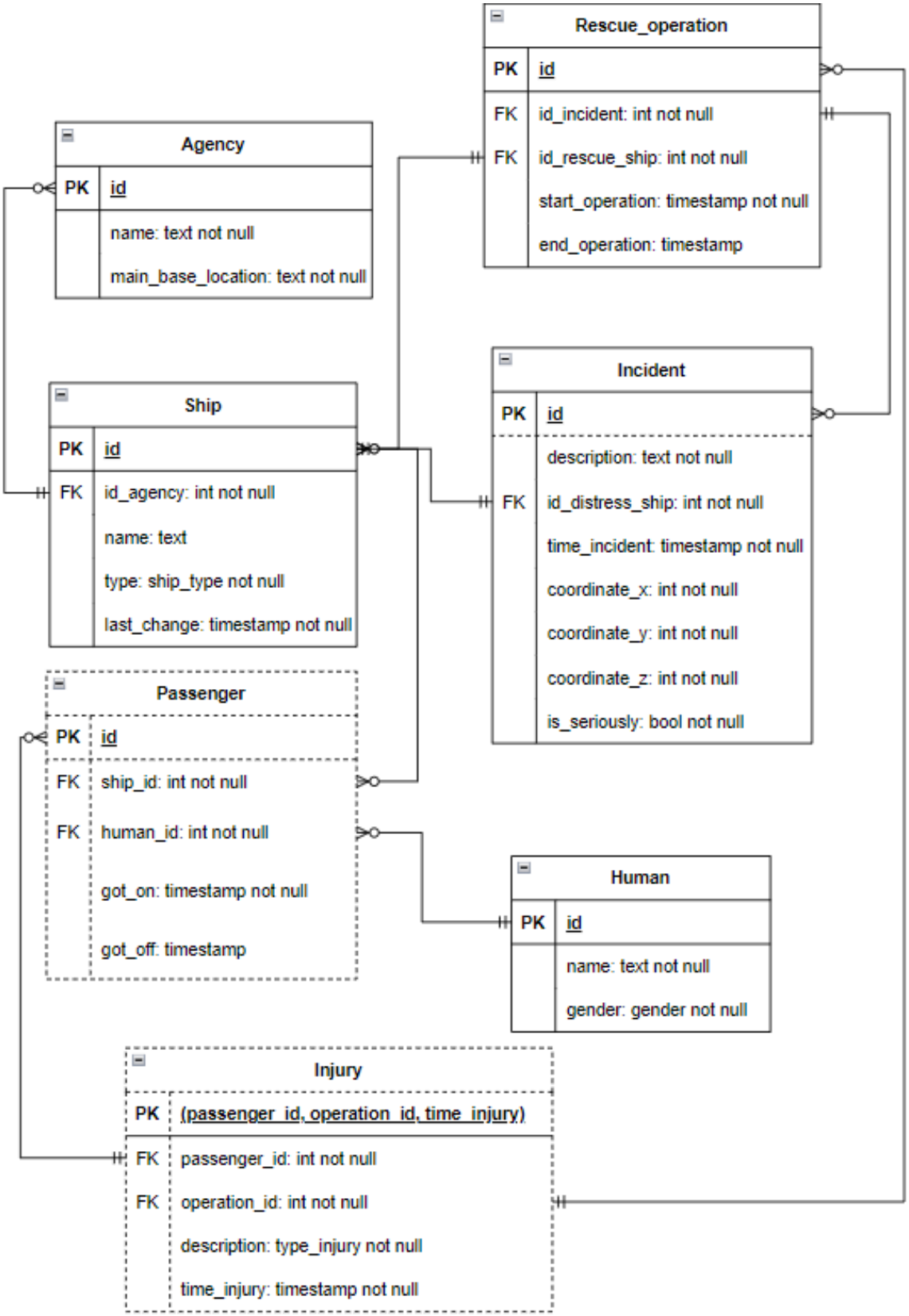

Рисунок 1 - Исходная модель

Функциональные зависимости (изначальные)

Human:

id → (name, gender)

Agency:

id → (name, main_base_location)

Ship:

id → (id_agency, name, type, last_change)

Incident:

id → (description, id_distress_ship, time_incident, coordinate_x, coordinate_y, coordinate_z, is_seriously)

Rescue_operation:

id → (id_incident, id_rescue_ship, start_operation, end_operation)

Passenger:

id → (ship_id, human_id, got_on, got_off)

Injury:

(passenger_id, operation_id, time_injury) → (description)

Преобразование к 1НФ

Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения. Не потребовалось, условие “на пересечении каждой строки и столбца – 1 значение” и так выполнялось.

Преобразование к 2НФ

Отношение находится во 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Не потребовалось, поскольку у всех первичных ключей нет подмножеств, а значит атрибуты всех отношений – в полной функциональной зависимости от соответствующих первичных ключей.

Преобразование к 3НФ

В некоторых отношениях наблюдались транзитивные зависимости:

- agency: id -> name, name -> main_base_location, id -> main_base_location;

По сути, транзитивность здесь даёт то, что в этих отношениях как бы два первичных ключа, только один указан явно (id), а другой получается в силу того, что значения атрибута должно быть уникальными (name). Поэтому:

- в agency убираем id и делаем первичным ключом name.

Преобразование к BCNF

Отношение находится в BCNF, если для каждой функциональной зависимости X →Y, X является потенциальным ключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является потенциальным ключом.

Денормализация

Объединение таблиц может ускорить обработку запросов. Например, можно рассмотреть объединение таблиц incident и rescue_operation, если часто запрашиваются данные об инциденте и спасательной операции, прилагающейся к нему, одновременно.

Также можно добавить несколько избыточных атрибутов, что может улучшить производительность запросов. Например, если часто запрашивается количество погибших людей во время инцидента, можно добавить атрибут number_of_deaths в таблицу Incident. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении Injury с атрибутом description = «не совместимые с жизнью».

Функциональные зависимости (после преобразований)

Human:

id → (name, gender)

Agency:

name → (main_base_location)

Ship:

id → (id_agency, name, type, last_change)

Incident:

id → (description, id_distress_ship, time_incident, coordinate_x, coordinate_y, coordinate_z, is_seriously, id_rescue_ship, start_operation, end_operation)

Passenger:

id → (ship_id, human_id, got_on, got_off)

Injury:

(passenger_id, incident_id, time_injury) → (description)

Даталогическая модель (после преобразований)

Можно сравнить начальную (Рисунок 1) и полученную (Рисунок 2) модели.

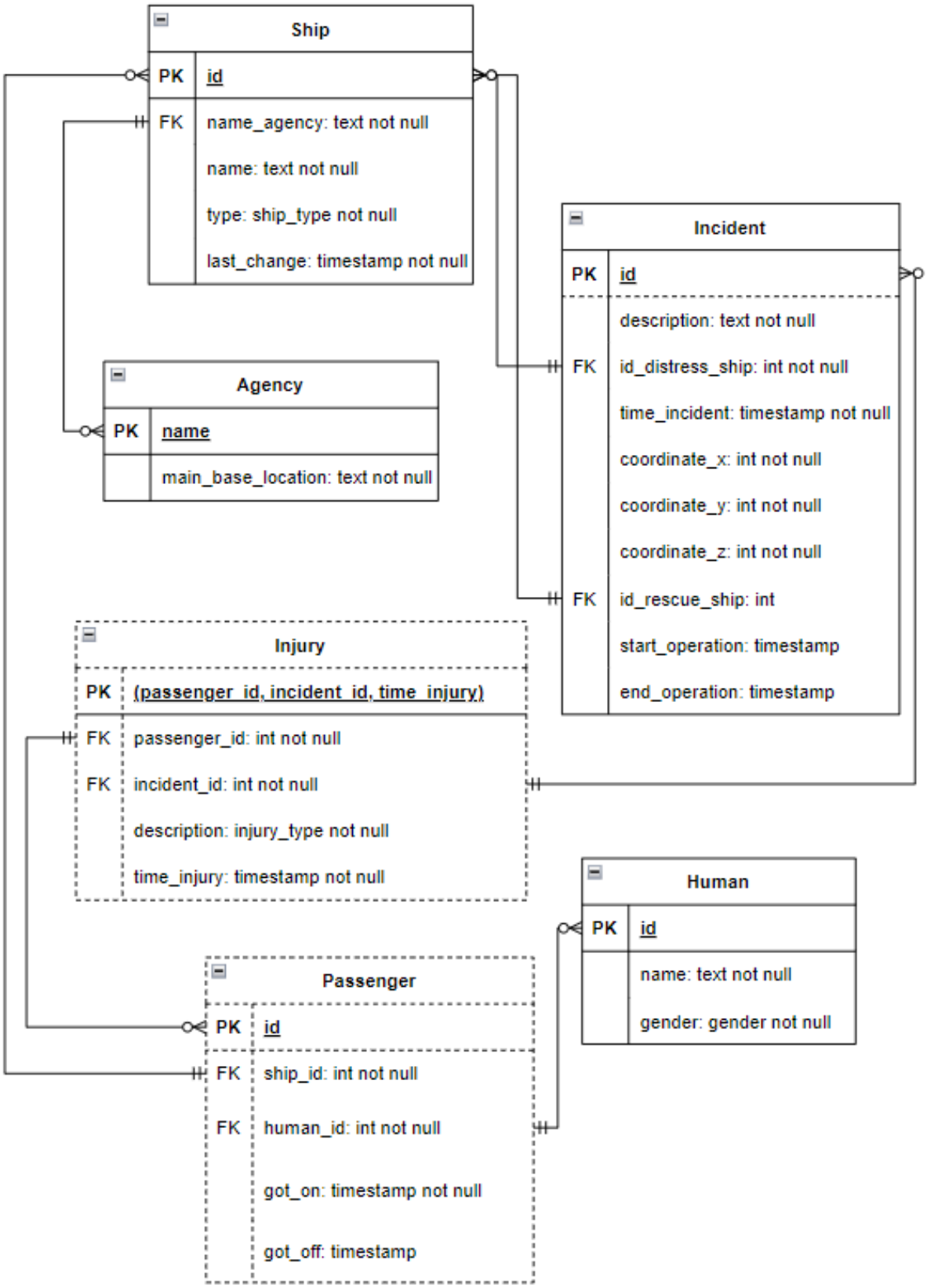

Рисунок 2 - получившаяся модель

Триггер и функция

Триггер вызывает функцию, которая проверяет тип спасательного корабля, и если он «distress», то не добавляем запись, выкидывая исключение. Также, если ошибки нет, то обновляем поля спасательного и терпящего бедствие кораблей

```
CREATE OR REPLACE FUNCTION check_and_update_type_ship() RETURNS trigger AS
$$
declare
   type_of_rescue_ship ship_type;
BEGIN
   SELECT ship.type INTO type_of_rescue_ship FROM ship WHERE ship.id =
NEW.id_rescue_ship;
   IF (type_of_rescue_ship = 'distress') then
      RAISE EXCEPTION 'Терпящий бедствие корабль не может прийти на помощь';
   end if;
   UPDATE ship
   SET type = 'distress',
      last_change = now()
   WHERE id = NEW.id_distress_ship;
   UPDATE ship
   SET type = 'rescue',
      last_change = now()
   WHERE id = NEW.id_rescue_ship;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE TRIGGER check_and_update_type
   BEFORE INSERT OR UPDATE
   ON incident
EXECUTE FUNCTION check_and_update_type_ship();
```

Выводы по работе

В ходе выполнения данной лабораторной работы я познакомился с различными нормальными формами и процессом приведения к ним, а также с процессом денормализации модели и написанием параметризированных функций на языке plpgsql.