Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3

Выполнил:

Горин Семён Дмитриевич

Группа P3108

Проверила:

Заболотняя Ольга Михайловна

**Содержание**

[Задание 3](#_Toc197380340)

[Инфологическая модель (Логическая модель) 3](#_Toc197380341)

[Даталогическая модель (Физическая модель) 4](#_Toc197380342)

[Функциональные зависимости 4](#_Toc197380343)

[Приведение модели к 3NF 4](#_Toc197380344)

[Приведение модели к BCNF 4](#_Toc197380345)

[Денормализация 5](#_Toc197380346)

[Функция вызываемая триггером 5](#_Toc197380347)

[Выводы 5](#_Toc197380348)

# 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.

**Описание предметной области:**

|  |
| --- |
| Горы еще зябли в тени, когда корабль достиг Шалмирейна. С высоты, на которой они находились, огромная чаша крепости выглядела совсем крохотной. Казалось просто невероятным, что когда-то от этого вот черного как ночь кружка зависели судьбы Земли. |

# Инфологическая модель (Логическая модель)

Инфологическая модель представлена на рисунке 1.

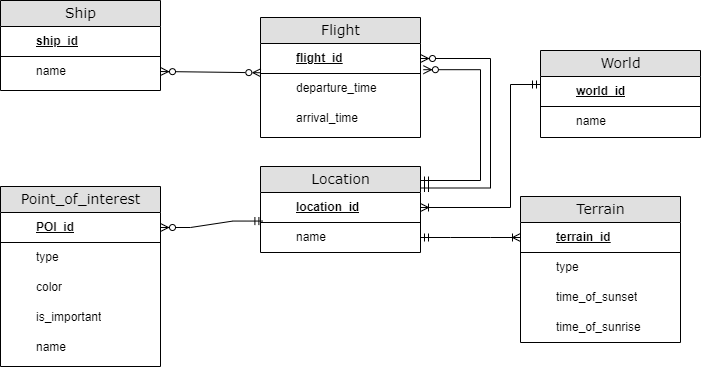


Рисунок 1. Инфологическая модель.

# Даталогическая модель (Физическая модель)

Даталогическая модель представлена на рисунке 2.

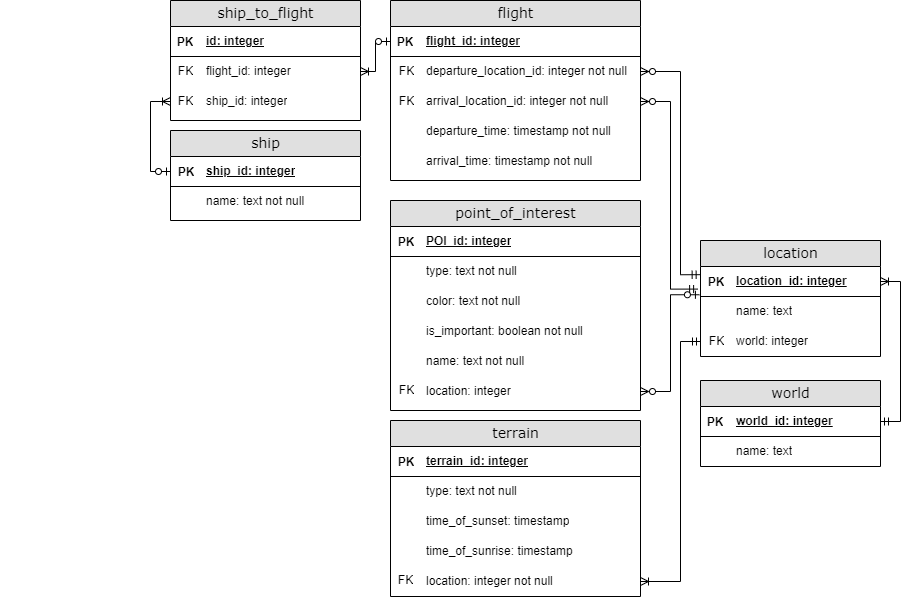


Рисунок 2. Даталогическая модель.

# Функциональные зависимости

Таблица 1

|  |  |
| --- | --- |
| Отношение | Минимальное множество ФЗ |
| **world** | world\_id → name |
| **location** | location\_id → name, world\_id |
| **point\_of\_interest** | poi\_id → type, color, is\_important, name, location\_id |
| **flight** | flight\_id → departure\_location\_id, arrival\_location\_id, departure\_time, arrival\_time |
| **ship** | ship\_id → name |
| **ship\_to\_flight** | id → flight\_id, ship\_id |
| **terrain** | terrain\_id → type, time\_of\_sunrise, time\_of\_sunset, location\_id |

# Приведение модели к 3NF

Поскольку исходная даталогическая модель уже была спроектирована с суррогатными ключами и без очевидных частичных или транзитивных зависимостей, схема на основе NF (здесь 3NF) совпадает с исходной даталогической моделью(представлена на рисунке 2).

# Приведение модели к BCNF

BCNF требует, чтобы для любой ФЗ X → A в отношении X было суперключом. Из таблицы 1 можем заметить, что во всех случаях единственные ФЗ исходят от первичных ключей, которые являются суперключами. Следовательно, вся схема уже находится в BCNF.

# Денормализация

Можно включить название мира(World.name) в таблицу локации(Location) в качестве аттрибута. Это позволит избежать излишних соединений при большом количестве операций чтения данных локации и мира. Для реализации данной денормализации следует:

1. Добавить новый столбец в location

**ALTER** location **ADD** **COLUMN** world\_name TEXT;

1. Популировать значения в столбце location.world\_name.

Вариант, работающий в PostgreSQL:

**UPDATE** location

**SET** world\_name **=** world**.**name

**FROM** world

**WHERE** location**.**world\_id **=** world**.**id**;**

СУБД-независимый вариант:

**UPDATE** location

**SET** world\_name **=** **(SELECT** world**.**name

**FROM** world

**WHERE** world**.**world\_id **=** location**.**world\_id**);**

1. Написать триггер, чтобы синхронизировать изменения из world.

**CREATE** **OR** **REPLACE** **TRIGGER** location\_update **AFTER** **UPDATE** **ON** world **OF** name **FOR** **EACH** **ROW** **EXECUTE** **PROCEDURE** location\_update\_func();

# Функция вызываемая триггером

**CREATE** **OR** **REPLACE** **FUNCTION** location\_update\_func() **RETURNS** **TRIGGER** **AS**

$$

BEGIN

**UPDATE** **location**

**SET** world\_name = NEW.**name**

**WHERE** **location**.world\_id = NEW.world\_id;

**RETURN** NEW;

END;

$$ **LANGUAGE** plpgsql;

# Выводы

В ходе лабораторной работы я составил функциональные зависимости для описанных в первой лабораторной работе отношений, исследовал полученную в ходе ЛР1 модель на соответствие 3НФ и НФБК, подумал как можно улучшить мою модель при помощи денормализации и написал триггер для её реализации.