Федеральное государственное автономное образовательное учреждение

высшего образования «Национальный исследовательский университет ИТМО»

*Факультет программной инженерии и компьютерной техники*

**Лабораторная работа №3**

**По дисциплине**

**“Информационные системы и базы данных ”**

Вариант: лучший

Выполнил:

Левченко ЯА

Группа: Р3118

Преподаватель:

Инячина Диана

Санкт-Петербург, 2023г

## Задание

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

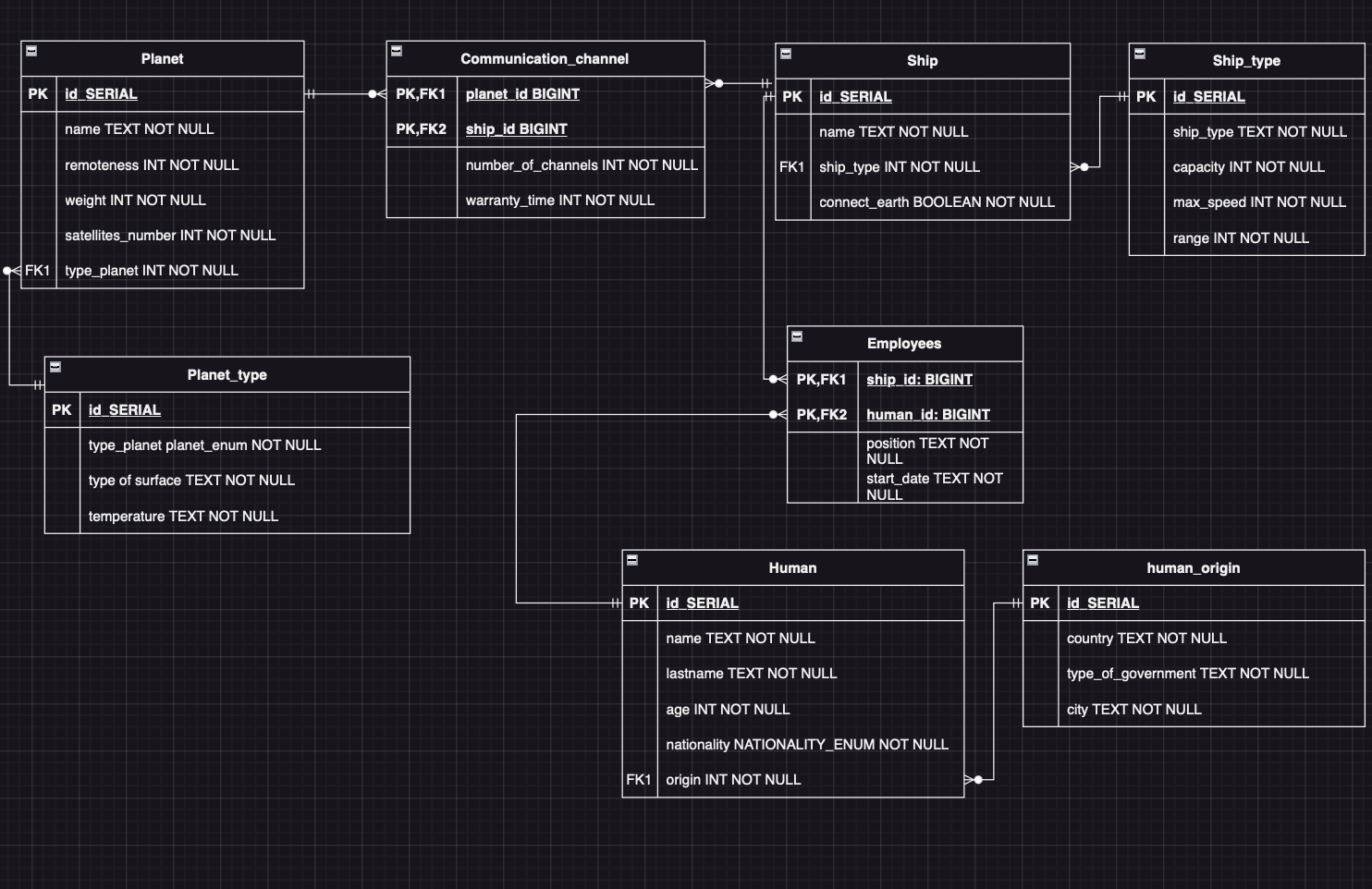
* опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
* приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум). Постройте схему на основе полученных отношений;
* опишите изменения в функциональных зависимостях, произошедшие после преобразования 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 функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL.

Исходная модель

****

**Функциональные зависимости**

**В таблице Ship\_type:**

**id -> ship\_type, capacity, max\_speed, range**

**В таблице Ship:**

**id -> name, ship\_type, connect\_earth**

**В таблице Planet\_type:**

**id -> type\_planet, type\_of\_surface, temperature**

**В таблице Planet:**

**id -> name, remoteness, weight, satelittes\_number, type\_planet\_id**

**В таблице Communication\_channel:**

**(Ship\_id, Planet\_id) -> number\_of\_channels, warranty\_time**

**В таблице human\_origin:**

**id -> country, type\_of\_government, city**

**В таблице Human:**

**id -> name, lastname, age, nationality, human\_origin\_id**

**В таблице Employees:**

**(Ship\_id, Human\_id) -> position, start\_date**

2) Первая нормальная форма:

Каждая таблица имеет первичный ключ, который уникально идентифицирует каждую запись в таблице.

Все столбцы в таблицах содержат атомарные значения, то есть нет множественных значений или повторяющихся групп данных в одной ячейке.

1NF соблюдается

3) Вторая нормальная форма:

Таблица в 1NF

Таблицы связаны первичными ключами

Все атрибуты зависят от первичного ключа целиком

2NF соблюдается

4) Третья нормальная форма:

Таблица в 2NF

Все атрибуты зависят только от первичных ключей, а не от других атрибутов( отсутствуют транзитивные функциональные зависимости не ключевых атрибутов от ключевых)

3NF соблюдается

5) Нормальная форма Бойса-Кодда (НФБК):

Для отношений, имеющих один потенциальный ключ (первичный), НФБК является 3НФ.

Исходная модель уже соответствует модели НФБК, потому что в таблицах часть составного ключа не зависит от какого-то неключевого атрибута.

6. Денормализованная

По тексту на корабле всего один сотрудник, поэтому мы могли бы удалить таблицу Employees и Human, помести эти данные в таблицу Ship. Появятся новые функциональные зависимости

Ship\_ip -> human\_id

Ship\_ip -> human\_id -> name

Ship\_ip -> human\_id -> lastname и так далее

Эта денормализация нарушает третью нормальную форму, так как появляются транзитивные зависимости от первичного ключа

Вдобавок можно объединить таблицы Human и human\_origin, если мы отказываемся от первой предложенной нормализации. Это не создаст транзитивных зависимостей от первичного ключа, но может сильно увеличить дублирование данных. (Но по тексту чел один, поэтому вариантик рабочий)

Триггер:

CREATE OR REPLACE FUNCTION update\_ship\_connect\_earth()

RETURNS TRIGGER AS $$

BEGIN

IF EXISTS (

SELECT 1

FROM Communication\_channel cc

INNER JOIN Ship s ON cc.Ship\_id = s.id

WHERE cc.Ship\_id = NEW.id

) THEN

NEW.connect\_earth = TRUE;

ELSE

NEW.connect\_earth = FALSE;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

Применение триггера:

CREATE TRIGGER update\_ship\_connect\_earth\_trigger

BEFORE INSERT OR UPDATE ON Ship

FOR EACH ROW

EXECUTE FUNCTION update\_ship\_connect\_earth();

Описание

update\_ship\_connect\_earth вызывается каждый раз, когда происходит обновление данных в таблице Ship. Функция проверяет, существуют ли связанные записи в таблице Communication\_channel для данного корабля (Ship\_id = NEW.id). Если такие записи существуют, поле connect\_earth устанавливается в значение TRUE, в противном случае - в значение FALSE.

7. Выводы по работе  
ВО время выполнения лабораторной работы я научился создавать инфологическую и датологическую модель БД, реализовывать её на SQL, а так же узнал про классификицаию сущностей и виды связей в БД.