

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики”

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

“Базы данных”

вариант №173378

Выполнил:

студент группы Р3119

Бардин Петр Алексеевич

Преподаватель:

Байрамова Хумай Бахруз Кызы

Содержание

1	Задание	2
2	Анализ	3
2.1	Описание предметной области	3
2.2	Описание сущностной	3
2.3	Инфологическая модель	4
3	Функциональные зависимости	5
3.1	Описание зависимостей	5
4	Приведение к нормальным формам	6
4.1	1НФ	6
4.2	2НФ	6
4.3	3НФ	6
4.4	НФБК	6
5	Изменения	7
6	Денормализации	7
7	Итоговая даталогическая модель	8
8	Функция	9
9	SQL	9
10	Вывод	9

1 Задание

Введите вариант:

Описание предметной области, по которой должна быть построена доменная модель:

Он проникал сквозь слои облаков все ниже и ниже, пока, наконец, не достиг точки, из которой даже обычный человек мог окинуть взглядом площадь в тысячу квадратных километров. Эта ничтожная по величине часть Большого Красного Пятна хранила тайну, о которой люди могли лишь догадываться.

Задачи:

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

1. описать функциональные зависимости для отношений полученной схемы (минимальное множество);
2. привести отношения в 3NF (как минимум). Постройте схему на основе полученных отношений;
3. описать изменения в функциональных зависимостях, произошедшие после преобразования в 3NF;
4. преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
5. какие денормализации будут полезны для вашей схемы? Приведите подробное описание;
6. Придумайте функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL;

2 Анализ

2.1 Описание предметной области

Исследовательский космический корабль, который имеет возможность перемещаться, для перемещения будем хранить телеметрию с корабля с данными о высоте в конкретный момент времени и прохождении препятствий (облаков).

Территориально определяем планету (Юпитер), объект на планете (Большое Красное Пятно) имеющий площадь, составные части объекта (также объекты). Над планетой определим слои облаков на определенной высоте.

Объект может содержать тайну, которая определяется содержимым, а также способностями отдельных типов сущностей к ее познанию.

2.2 Описание сущностной

1. Стержневые

- Strong
 - Живое существо
 - Вид живого существа
 - Космический корабль
 - Экипаж
 - Планета
 - Объект
 - Тайна
- Weak
 - Космонавт
 - Полет
 - Запись в полетном журнале

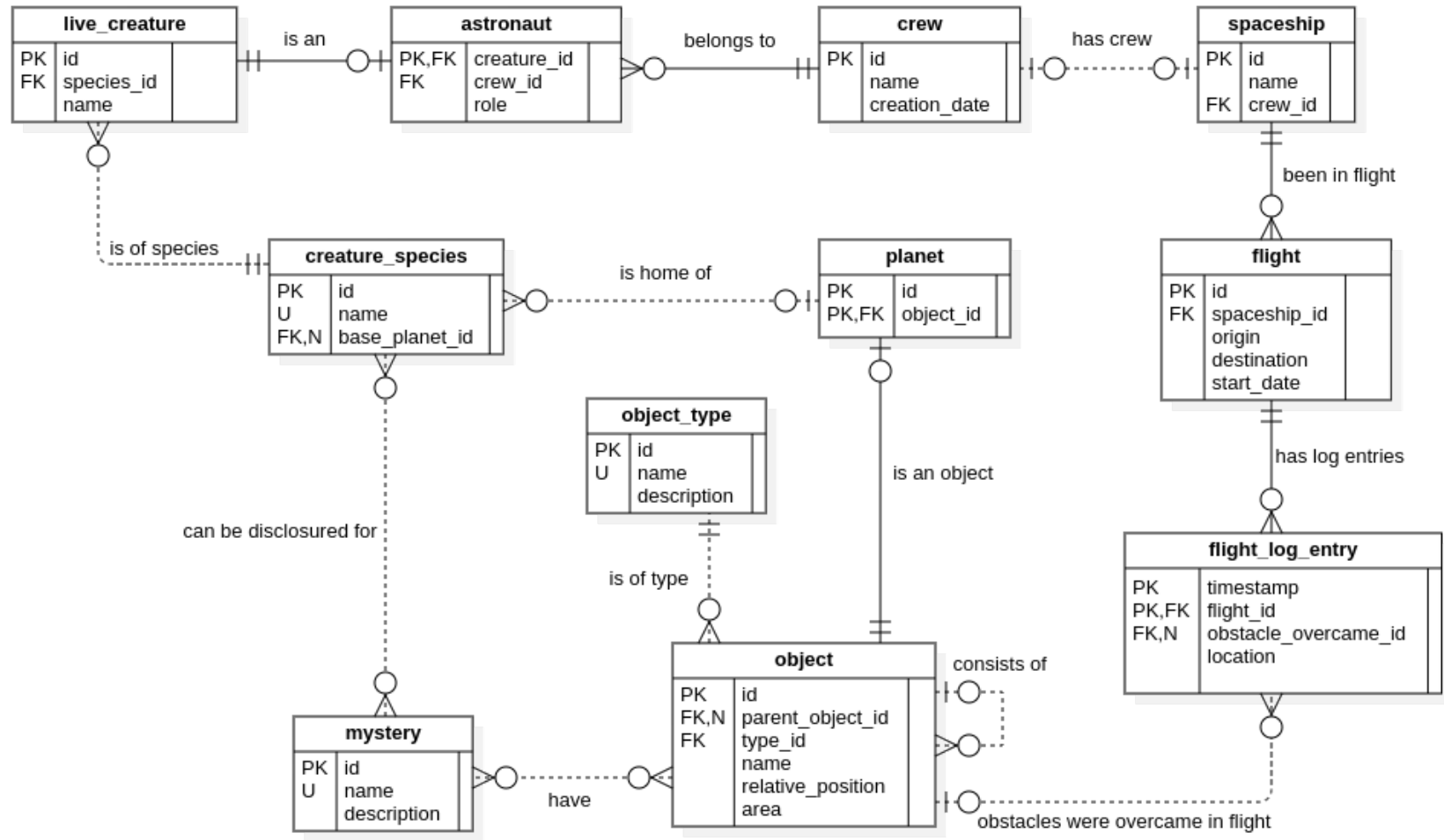
2. Ассоциативные

- Обладание тайной
- Возможность раскрытия тайны для вида

3. Характеристические

- Тип объекта
- Тип раскрытия тайны

2.3 Инфологическая модель



3 Функциональные зависимости

3.1 Описание зависимостей

Минимальное множество.

Тривиальные ФЗ опущены.

- live_creature
 - id \rightarrow species_id
 - id \rightarrow name
- astronaut
 - creature_id \rightarrow crew_id
 - creature_id \rightarrow role
- crew
 - id \rightarrow name
 - id \rightarrow creation_date
- spaceship
 - id \rightarrow name
 - id \rightarrow crew_id
- flight
 - id \rightarrow spaceship_id
 - id \rightarrow origin
 - id \rightarrow destination
 - id \rightarrow start_date
- flight_log_entry
 - timestamp, flight_id \rightarrow id
 - id, flight_id \rightarrow timestamp
 - id, flight_id \rightarrow location
 - id, flight_id \rightarrow obstacle_overcame_id
- object
 - id \rightarrow type_id
 - id \rightarrow name
 - id \rightarrow parent_object_id
 - id \rightarrow planet_id
 - id \rightarrow relative_position
 - id \rightarrow area
 - parent_object_id \rightarrow planet_id
- planet
 - id \rightarrow name
 - id \rightarrow location
- mystery
 - id \rightarrow name
 - id \rightarrow description
- mystery_disclosure
 - mystery_id, creature_species_id \rightarrow available_type_id
- object_type
 - id \rightarrow name
 - id \rightarrow description
- creature_species
 - id \rightarrow name
 - id \rightarrow base_planet_id
- disclosure_type
 - id \rightarrow description
 - id \rightarrow disclosure_percent
- objects_mystery

4 Приведение к нормальным формам

4.1 1НФ

Каждый кортеж во всех отношениях содержит только одно значение для каждого атрибута. Изменений не требовалось.

4.2 2НФ

1НФ + каждый атрибут, не входящий в первичный ключ, находится в полной неприводимой ФЗ от ПК.

Исправлена таблица `flight_log_entry` - множество ФЗ не было минимально, не оптимально выбран первичный ключ, что создает избыточность. Исходные зависимости:

- $\{id, flight_id\} \longrightarrow \{id, flight_id, timestamp\}$
- $id \longrightarrow \{id, flight_id, timestamp\}$
- $\{flight_id, timestamp\} \longrightarrow id$
- $id \longrightarrow \{location, obstacle_overcame_id\}$
- $\{flight_id, timestamp\} \longrightarrow \{location, obstacle_overcame_id\}$

При этом поле `id` не несло смысловой нагрузки вовсе (не используется в модели), записи журнала идентифицируются однозначно по номеру полета и временной отметке, т.е `id` могло быть выведено из $\{flight_id, timestamp\}$. Таким образом, убирая поле `id` и заменяя первичный ключ на указанное ранее множество, получаем единственную ФЗ: $\{flight_id, timestamp\} \longrightarrow \{location, obstacle_overcame_id\}$.

4.3 3НФ

2НФ + каждый атрибут, не входящий в первичный ключ, находится в нетранзитивной ФЗ от ПК.

Убрана транзитивная зависимость в `object`: $id \longrightarrow parent_object_id \longrightarrow planet_id$. Решено через разделение планеты на объект-планету и описание планеты, тогда в указание на принадлежность планете встраивается в существующую иерархическую модель на основе `parent_object_id`.

4.4 НФБК

Детерминант ФЗ - потенциальный ключ

Изменений не требовалось.

В во всех отношениях детерминантом выступает первичный ключ, который является потенциальным.

5 Изменения

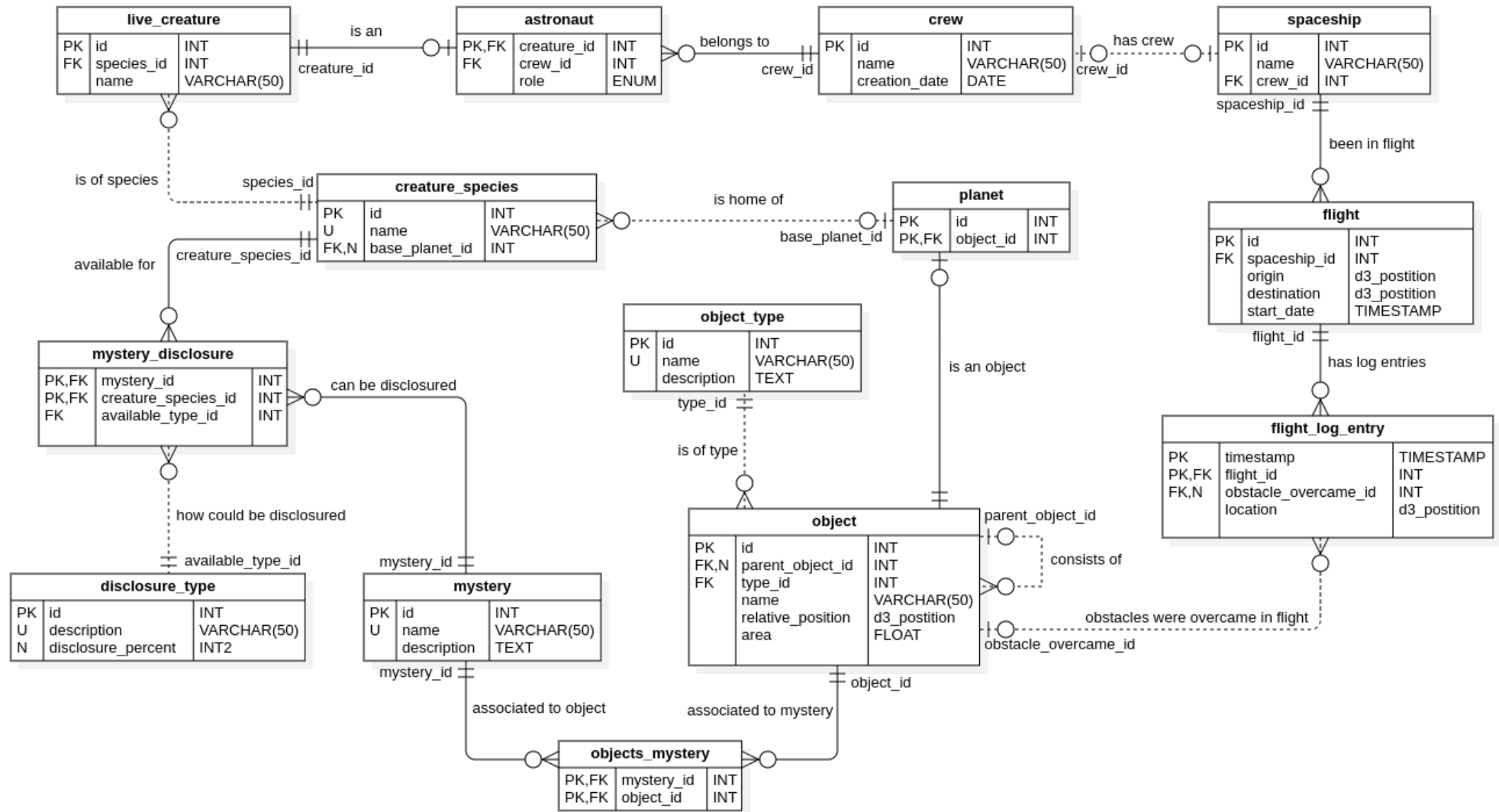
- planet
 - id \rightarrow object_id
- object
 - id \rightarrow type_id
 - id \rightarrow name
 - id \rightarrow parent_object_id
 - id \rightarrow relative_position
 - id \rightarrow area
- flight_log_entry
 - timestamp, flight_id \rightarrow obstacle_overcame_id
 - timestamp, flight_id \rightarrow location

6 Денормализации

Проведение денормализаций не имеет смысла, так как разделение таблиц продиктовано в первую очередь предметной областью, а не нормализациями, таким образом, объединение отношений приведет не только к увеличению избыточности и возможным аномалиям, но и к нарушениям ограничений целостности из модели или потере смысла.

Максимальное, что допустимо сделать, это объединить live_creature и astronaut, в общем случае делать этого нельзя, но в текущей модели, живых существ, не являющихся космонавтами (*сейчас*) нет (от live_creature зависит только astronaut), а значит можно их объединить, так как связь 1-к-1, и в итоговой таблице из ключей останется только id от live_creature, а creature_id не будет нужен.

7 Итоговая даталогическая модель



8 Функция

Будем считать, что если в иерархии физических объектов родительский объект имеет тайну, то эта тайна распространяется на все объекты-части исходного. Значит при добавлении тайны объекту, необходимо убедиться, что эта тайна не присвоена ему за счет наследования тайн с родительских объектов, дабы оптимизировать занимаемый объем базы данных.

Разработана функция-триггер, обеспечивающая рекурсивную проверку данного условия при выполнении *CREATE/UPDATE* над `objects_mystery`. А также функция обеспечивающая выдачу полного списка тайн для объекта, включая наследуемые.

9 SQL

Реализация даталогической модели на PostgreSQL для данной лабораторной размещена в системе контроля версий Git на сервисе Github:

<https://github.com/BardinPetr/itmo-labs/tree/main/db/year1/lab3>.

10 Вывод

В ходе выполнения работы были изучены основы реляционной алгебры, способы и принципы нормализаций/денормализаций, нормальные формы 1-3 и НФБК. На практике отработан анализ существующей модели относительно соответствия нормальным формам. Также изучено создание и использование функций в PostgreSQL.