

Факультет программной инженерии и компьютерной техники

Информационные системы и базы данных

Лабораторная работа №1

Вариант №948

Преподаватель: Гаврилов Антон Валерьевич

Выполнили: Кульбако Артемий Юрьевич Р33113

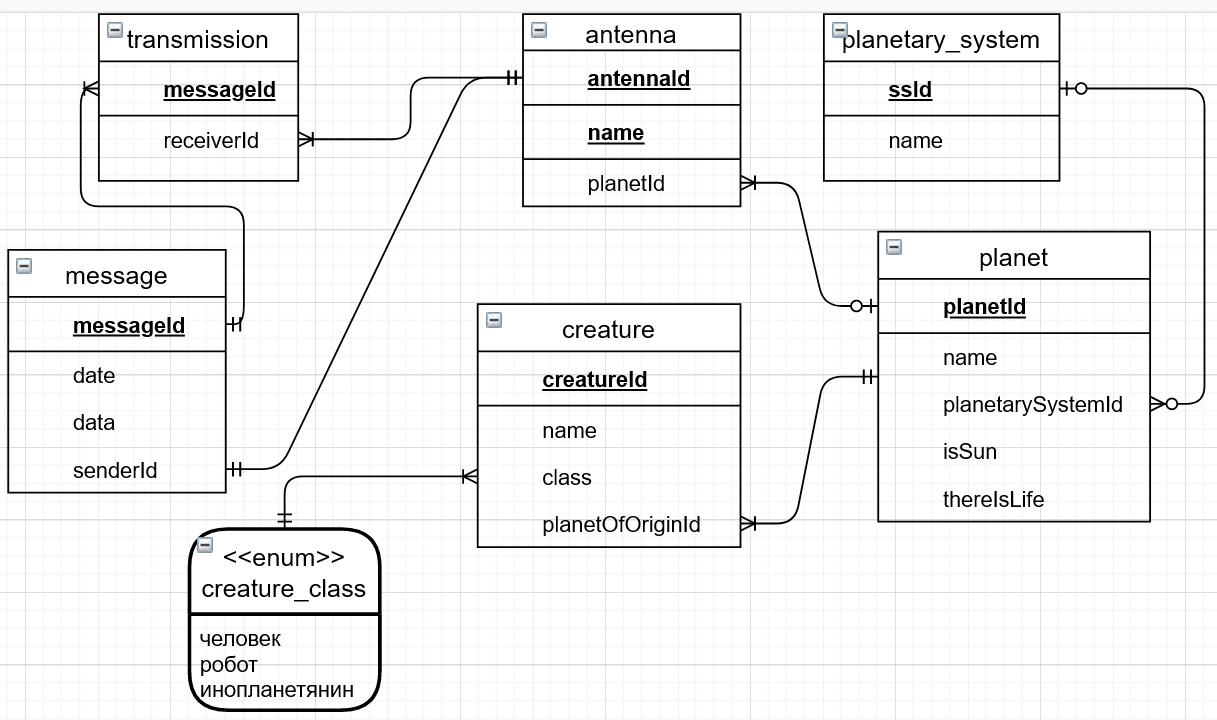
# Текст задания

С полминуты Боумен и Пул молча изучали экран. Это изображение передавала на пульт управления длиннофокусная телевизионная камера, укрепленная на ободе большой параболической антенны. Перекрестье, наложенное центром на изображение Земли, указывало, что антенна ориентирована точно. Ведь если узкий карандашик луча не был бы наведен точно на Землю, они не смогли бы ни передавать, ни принимать передач. Сигналы, посланные в обоих направлениях, не попадали бы на антенны и улетали бы, унося с собой нераскрытыми слова и образы сквозь всю Солнечную систему в бескрайнюю пустоту, простирающуюся за ней. Если бы они и были когда-нибудь приняты, то лишь через столетия - и, конечно, не землянами...

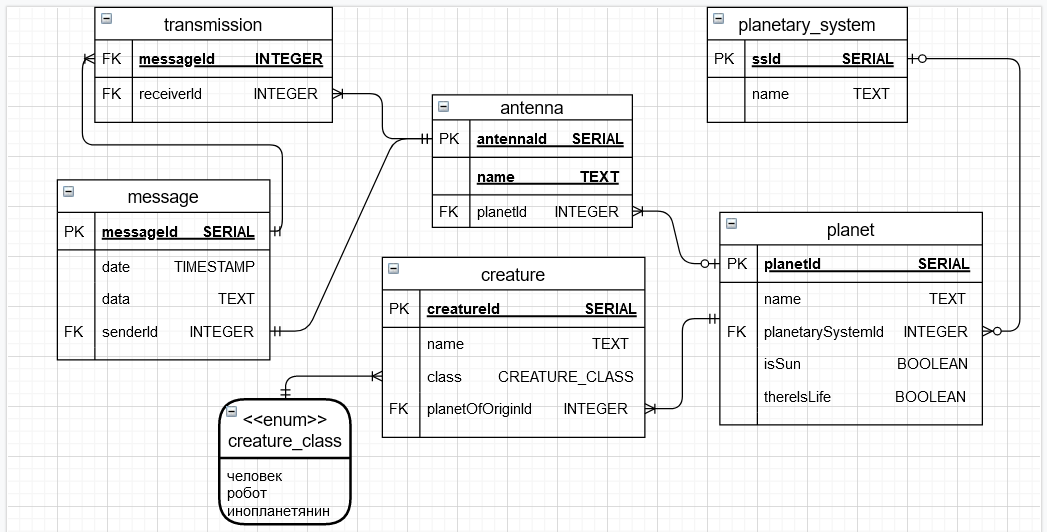
# Предметная область

Первым встречаемым субъектом в тексте являются действующие лица Боумен и Пул, а в конце текста «не земляне», что наводит на мысль существования СУЩЕСТВ разных типов. Действующие лица передают СООБЩЕНИЯ (передавать и принимать сообщения будут АНТЕННЫ). Так-как одно и то же сообщение может иметь несколько адресатов (sos, к примеру, рассылается всем), то есть смысл в создании ассоциативной сущности ПЕРЕДАЧА, с информацией о сообщении и получателях (отправителя будет хранить непосредственно в сообщении, во избежание избыточного дублирования информации). Антенны должны располагаться где-то, например на ПЛАНЕТАХ (в тексте упоминается передача сообщения на Землю). Естественно, планеты могут являться частью ПЛАНЕТНОЙ\_СИСТЕМЫ.

# Инфологическая модель



# Даталогическая модель



# Создание схемы

--стержневая

CREATE TABLE planetary\_system

(

ssId SERIAL PRIMARY KEY,

name TEXT NOT NULL

);

--стержневая

CREATE TABLE planet

(

planetId SERIAL PRIMARY KEY,

name TEXT NOT NULL,

planetarySystemId INTEGER REFERENCES planetary\_system ON DELETE CASCADE,

isSun BOOLEAN,

thereIsLife BOOLEAN

);

--характеристическая

CREATE TYPE creature\_class AS ENUM ('человек', 'робот', 'инопланетянин');

--стержневая

CREATE TABLE creature

(

crId SERIAL PRIMARY KEY,

name TEXT NOT NULL,

class CREATURE\_CLASS NOT NULL,

planetOfOrigin INTEGER REFERENCES planet ON DELETE RESTRICT

);

--стержневая

CREATE TABLE antenna

(

antennaId SERIAL PRIMARY KEY,

name TEXT,

planetId INTEGER NOT NULL REFERENCES planet

);

--стержневая

CREATE TABLE message

(

messageId SERIAL PRIMARY KEY,

date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

data TEXT,

senderId INTEGER NOT NULL REFERENCES antenna

);

--ассоциативная

CREATE TABLE transmission

(

messageId INTEGER NOT NULL REFERENCES message,

reciverId INTEGER NOT NULL REFERENCES antenna

);

INSERT INTO planetary\_system (name) VALUES

('Солнечная система'),

('TRAPPIST-1'),

('UX Tau A');

INSERT INTO planet (name, planetarySystemId, isSun, thereIsLife) VALUES

('Земля', 1, false, true),

('Солнце', 1, true, false);

INSERT INTO planet (name, planetarySystemId, isSun) VALUES ('TRAPPIST-1b', 2, false);

INSERT INTO creature (name, class, planetOfOrigin) VALUES

('Боумен', 'человек', 1),

('Пул', 'человек', 1),

('HAL9000', 'робот', 1);

INSERT INTO creature (name, class) VALUES ('Чёрный монолит', 'инопланетянин');

INSERT INTO antenna (name, planetId) VALUES

('Discovery', 1),

('Beyond', 3);

INSERT INTO message (data, senderId) VALUES

('За бесконечность', 1),

('Поехали!', 1),

('Земля в иллюминатарееее', 1),

('Небрежность в космосе - простейший способ самоубийства', 2),

('Чем совершеннее техника передачи информации, тем более заурядным, пошлым, серым становится её содержание', 2),

('Миссия выполнена', 2);

INSERT INTO transmission VALUES

(1, 1),

(1, 2),

(2, 1),

(3, 1),

(3, 2);

# Вывод

SQL стандартизировал подход для хранения данных, что позволило программистом отказаться от траты времени на изобретение собственных файловых форматов хранения данных, а в прикладном ПО использовать реляционное отображение, как на серверах, так и локальных устройствах (молодые ОС активно используют этот подход).

В процессе выполнения, большой сложностью было переключить образ мышления с объектно-ориентированного на реляционное (концепция формирования связей на основе промежуточных таблиц порой сильно путает). Учитывая повсеместное использование ООП в реальных проектах, я думаю, если бы не огромное количество легаси-кода, использование ООБД было бы куда продуктивнее с точки зрения сокращения времени разработки.