

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Кафедра «Информатика»
кафедра

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ

Разработка архитектуры БД
Тема

Преподаватель

подпись, дата

А. К. Погребников

инициалы, фамилия

Студент

КИ19-16/16 031939175

номер группы, зачетной
книжки

подпись, дата

А. Д. Непомнящий

инициалы, фамилия

Красноярск 2021

1 Цель работы

Цель работы состоит в получении навыков проектирования баз данных.

2 Задачи

Выполнение работы сводится к следующим задачам.

1. Придумать и оформить архитектуру БД.
2. Создать базу данных в PostgreSQL.
3. Заполнить созданные таблицы не менее десятью записями каждую.

3 Ход работы

3.1 Архитектура БД

Для работы в рамках настоящей и последующих практических работ взяты сущности из игры Elite: Dangerous. Мир видеоигры разбит на системы, в которых могут находиться станции. Каждая станция предлагает покупку и продажу по некоторой цене некоторого количества каких-то товаров (полный перечень которых известен).

В основу архитектуры заложена эта система, однако у каждой из сущностей есть еще ряд немаловажных в контексте игры параметров, все они отражены в реализации архитектуры БД ниже.

Далее приведена схема архитектуры БД.

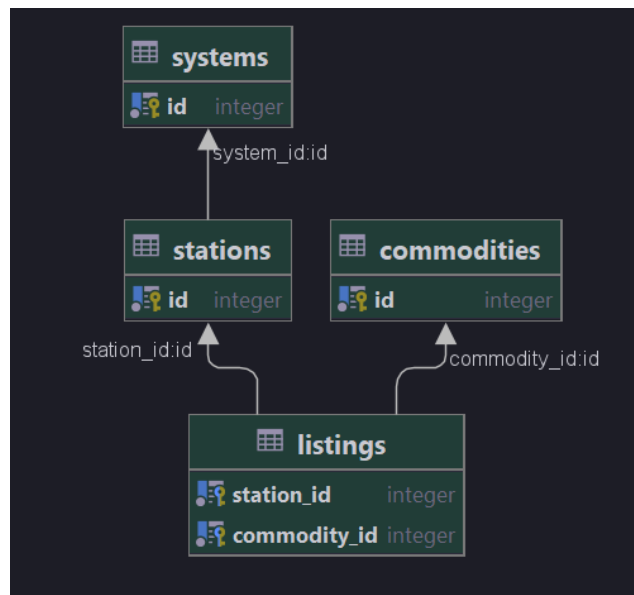


Рисунок 1 – Основная архитектура БД

3.2 Реализация БД

Далее приведен код PostgreSQL для создания таблиц.

Листинг 1 – Код для создания таблицы станций

```

create table stations
(
    id integer not null
        constraint stations_pk
            primary key,
    name text,
    system_id integer
        constraint stations_fk_system
            references systems
            on delete cascade,
    type_id integer
        constraint stations_fk_type
            references station_types,
    ed_market_id numeric,
    allegiance_id integer
        constraint stations_fk_allegiance
            references allegiances,
    controlling_minor_faction_id integer
        constraint stations_fk_faction
            references factions
            on delete cascade,
    government_id integer
        constraint stations_fk_government
            references governments,
    distance_to_star integer,
    has_blackmarket boolean,
    has_carrier_administration boolean,
    has_carrier_vendor boolean,
    has_commodities boolean,

```

Окончание листинга 1

```
has_interstellar_factors    boolean,  
  has_market                boolean,  
  has_rearm                 boolean,  
  has_refuel                boolean,  
  has_repair                boolean,  
  has_shipyard              boolean,  
  has_technology_broker     boolean,  
  has_universal_cartographics boolean,  
  has_docking               boolean,  
  is_planetary              boolean,  
  max_landing_pad_size      text,  
  has_material_trader       boolean,  
  has_outfitting            boolean,  
  body_id                   integer  
);
```

Листинг 2 – Код для создания таблицы систем

```
create table systems  
(  
  allegiance_id            integer  
    constraint systems_allegiances_allegiance_id_fk  
    references allegiances,  
  controlling_minor_faction_id integer  
    constraint systems_factions_id_fk  
    references factions,  
  ed_system_address        numeric,  
  edsm_id                  integer,  
  government_id            integer  
    constraint systems_governments_government_id_fk  
    references governments,  
  id                       integer not null  
    constraint systems_pk  
    primary key,  
  is_populated             boolean,  
  name                     text,  
  needs_permit             boolean not null,  
  population               numeric,  
  power                    text,  
  power_state_id           integer  
    constraint systems_power_states_power_state_id_fk  
    references power_states,  
  primary_economy_id       integer  
    constraint systems_economies_primary_economy_id_fk  
    references economies,  
  reserve_type_id          integer  
    constraint systems_reserve_types_reserve_type_id_fk  
    references reserve_types,  
  security_id              integer  
    constraint systems_securities_security_id_fk  
    references securities,  
  x                        numeric,  
  y                        numeric,  
  z                        numeric  
);
```

Листинг 3 – Код для создания таблицы товаров

```
create table commodities
(
    id            integer not null
        constraint commodities_pk
            primary key,
    ed_id         integer,
    name          text,
    category      text
);

alter table commodities
owner to eddb_admin;
ServerTask(socketFileDescriptor);
close(socketFileDescriptor);
}
```

Листинг 4 – Код для создания таблицы листингов станций

```
create table listings
(
    station_id    integer not null
        constraint listings_fk_station
            references stations,
    commodity_id  integer not null
        constraint listings_fk_commodity
            references commodities,
    sell_price    integer,
    supply        integer,
    buy_price     integer,
    demand        integer,
    constraint listings_pk
        primary key (station_id, commodity_id)
);
```

3.3 Записи в таблице

В соответствии с заданием таблицы были заполнены. Т.к. в таблицах уже созданы ключи и ограничения, достаточным для демонстрации факта заполненности таблиц будет результат запросов `SELECT count(*)` для каждой таблицы.

count
20585

count
140095

count
374

count
6258132


```

SELECT count(*)
FROM systems;
SELECT count(*)
FROM stations;
SELECT count(*)
FROM commodities;
SELECT count(*)
FROM listings

```

Рисунок 2 – Результат запросов SELECT count(*) к каждой из таблиц

4 Вывод

В ходе работы была спроектирована БД в соответствии с выбранной предметной областью будущих практических работ.