**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №4

по дисциплине «**Базы данных**»

Автор: Парфенова Ольга

Факультет: ПИиКТ

Группа: P33201

Преподаватель: Райла Мартин

Санкт-Петербург, 2023

# **Задание**

Для выполнения лабораторной работы №4 необходимо:

* Реализовать разработанную в рамках лабораторной работы №3 даталогическую модель в реляционной СУБД PostgreSQL.
* Заполнить созданные таблицы данными.
* Обеспечить целостность данных при помощи средств языка DDL.
* В рамках лабораторной работы должны быть разработаны скрипты для создания/удаления требуемых объектов базы данных, заполнения/удаления содержимого созданных таблиц.

# **Описание предметной области**

Софт для операторов техподдержки интернет-провайдера, предоставляющая работу с оборудованием, линиями связи, учёт абонентов, персонала, взаимодействие с биллингом, автоматизацию технического обслуживания сетей и эффективно выстраивающий связь абонент-оператор.

Пример: UserSide.

Примеры таблиц:

Адреса (id, город, улица)

Дома (id, улица, номер, список предоставляемых услуг, провайдер, подрядчик)

Квартиры (id, дом, номер абонент, провайдер, свитч)

Абоненты (id, ФИО, логин, номер лицевого счета, провайдер, тариф, свитч, порт, кто подключал, фамилия монтажника, количество заявок, количество открытых заявок)

Свитч (id, порт, статус порта, линк, мак абонента, абонент в порту)

Провайдеры (id, название, тип подключения, количество абонентов)

Заявка (номер, абонент, автор, тип заявки, дата, время, статус заявки, дата изменения, дата завершения, исполнители)

Комментарии к заявке (номер заявки, ФИО автора комментария, дата время, текст)

Сотрудники (ФИО, должность, дата рождения, дата принятия на работу)

Заявки службы технической поддержки

Заявки монтажного отдела

Заявки клиентского отдела

# **DDL-скрипты и DML-скрипты**

| create table arp\_diagnostics (  id serial primary key,  abonent\_id serial,  has\_arp boolean not null,  arp\_id serial,  time timestamp );  create table district (  id serial primary key,  town text,  district text );  create table employee (  id serial primary key,  name text not null,  surname text not null,  role text not null,  employment\_date date );  create table cable\_diagnostics (  id serial primary key,  employee\_id serial not null,  port\_id serial not null,  link\_state text,  diagnostics text,  constraint fk\_cable\_diagnostics\_employee\_id  foreign key (employee\_id)  references employee(id) );  create table executor (  request\_id serial primary key,  department text );  create table provider (  id serial primary key,  name text not null,  type\_of\_connection text );  create table abonent (  id serial primary key,  login text not null unique,  account\_number integer not null unique,  name text,  surname text,  phone\_number text unique,  email text unique,  status text,  address\_id serial not null,  note text,  provider\_id serial not null,  constraint fk\_abonent\_provider\_id  foreign key (provider\_id)  references provider(id) );  create table vlan (  id serial primary key,  name text not null );  create table switch (  id serial primary key,  ip varchar(16) not null unique,  model text,  port\_count integer,  house\_id serial,  provider\_id serial,  uplink\_port integer );  create table commutation (  abonent\_id serial,  switch\_id serial,  port\_id serial,  cable\_type text,  primary key (abonent\_id, switch\_id, port\_id) );  create table equipment\_connections (  equipment1\_id serial,  equipment2\_id serial,  speed integer not null check (speed = 100 or speed = 1000 or speed = 10000),  primary key (equipment1\_id, equipment2\_id) );  create table payment\_history (  abonent\_id serial primary key,  date timestamp not null,  balance\_change integer not null );  create table address (  id serial primary key,  house\_id serial not null,  porch integer,  floor integer,  flat integer not null );  create table request (  id serial primary key,  abonent\_id serial not null,  author\_id serial not null,  create\_date timestamp not null,  status text not null,  status\_change\_date timestamp,  close\_date timestamp,  type text not null,  text text,  constraint fk\_request\_author\_id  foreign key (author\_id)  references employee(id) );  create table node (  id serial primary key,  model text,  district\_id serial,  provider\_id serial,  constraint fk\_node\_district\_id  foreign key (district\_id)  references district(id)  constraint fk\_node\_provider\_id  foreign key (provider\_id)  references provider(id) );  create table mac\_address (  port\_id serial,  mac\_address text,  primary key (port\_id, mac\_address) );  create table connection\_diagnostics (  id serial primary key,  employee\_id serial not null,  switch text not null,  port text not null,  has\_link boolean,  link\_type text,  has\_mac boolean,  errors integer,  errors\_rise boolean,  correct\_vlan boolean,  switch\_loss integer );  create table billing (  abonent\_id serial primary key,  balance integer not null,  status text );  create table house (  id serial primary key,  district\_id serial not null,  street text not null,  number integer not null );  create table router\_settings (  abonent\_id serial primary key,  ip varchar(16) not null,  mask varchar(16) not null,  gateway varchar(16) not null );  create table arp (  id serial primary key,  create\_date timestamp not null,  mac\_address\_id serial not null,  node\_id serial not null,  vlan\_id serial not null );  create table comment (  id serial primary key,  request\_id serial not null,  date timestamp not null,  text text not null,  employee\_id serial not null,  constraint fk\_comment\_request\_id  foreign key (request\_id)  references request(id),  constraint fk\_comment\_employee\_id  foreign key (employee\_id)  references employee(id) );  create table port (  id serial primary key,  switch\_id serial not null,  port\_number integer not null,  address\_id serial,  vlan\_id serial,  port\_status text not null,  connection\_speed integer check (connection\_speed = 0 or connection\_speed = 10 or connection\_speed = 100 or connection\_speed = 1000),  description text,  connection\_status text,  constraint fk\_port\_switch\_id  foreign key (switch\_id)  references switch(id),  constraint fk\_port\_vlan\_id  foreign key (vlan\_id)  references vlan(id) ); |
| --- |

<https://github.com/emshen6/database-4>

# **Выводы по работе**

В рамках данной лабораторной работы были разработаны скрипты для создания таблиц, заполнения таблиц данными, удаления таблиц, удаления данных из таблиц, создания ограничений целостности для указанной предметной области.

За эту лабораторную работу я улучшила свой навык владение DML и DDL, научилась работать с СУБД psycopg2, создавать ssh tunnels c помощью python.