



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных

О Т Ч Е Т

по лабораторной работе № 1

Название: Инсталляция СУБД PostgreSQL 9.6 на виртуальной машине,
создание базы данных и работа с ней

Дисциплина: Технология параллельных систем баз данных

Студент

ИУ6-12М

(Группа)

(Подпись, дата)

Д.С. Каткова

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.Д. Пономарев

(И.О. Фамилия)

Москва, 2023

Цель лабораторной работы

Целью лабораторной работы является формирование следующей компетенции: студент должен получить навыки инсталляции СУБД PostgreSQL в среде ОС Ubuntu. Он должен протестировать её работу, разработать небольшую базу данных и выполнить запросы к ней.

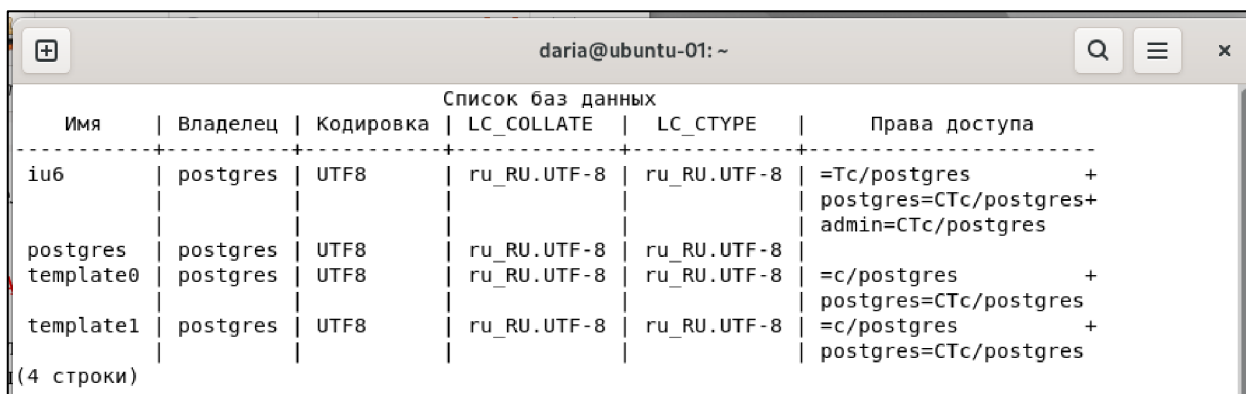
Инсталляция СУБД PostgreSQL 9.6

После установки виртуальной машины на нее была установлена СУБД PostgreSQL 9.6. Результат установки изображен на рисунке 1.

```
daria@ubuntu-01:~$ psql --version
psql (PostgreSQL) 9.6.24
```

Рисунок 1 – Результат установки СУБД PostgreSQL 9.6

Далее был создан экземпляр базы данных, что изображено на рисунке 2.

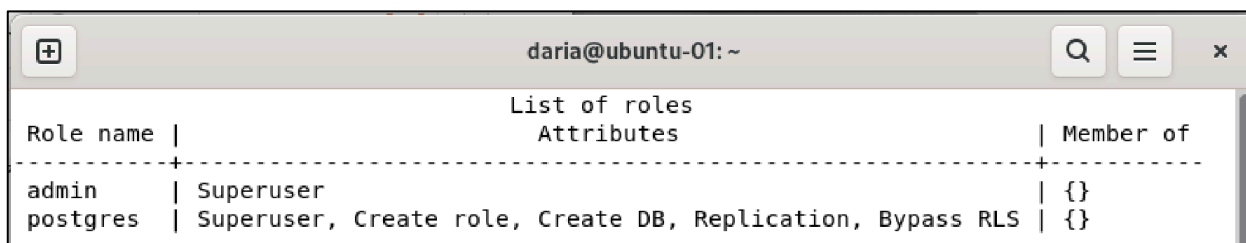


Список баз данных					
Имя	Владелец	Кодировка	LC_COLLATE	LC_CTYPE	Права доступа
iu6	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	=Tc/postgres + postgres=CTc/postgres+ admin=CTc/postgres
postgres	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	=c/postgres + postgres=CTc/postgres
template0	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	=c/postgres + postgres=CTc/postgres

(4 строки)

Рисунок 2 – Создание базы данных

Далее был создан новый пользователь с именем “admin” с правами супервизора, что изображено на рисунке 3.



List of roles		
Role name	Attributes	Member of
admin	Superuser	{}
postgres	Superuser, Create role, Create DB, Replication, Bypass RLS	{}

Рисунок 3 – Создание нового пользователя

Создание нового пользователя, ввод и поиск данных

В Ubuntu был создан новый пользователь с именем “admin”, что изображено на рисунке 4.

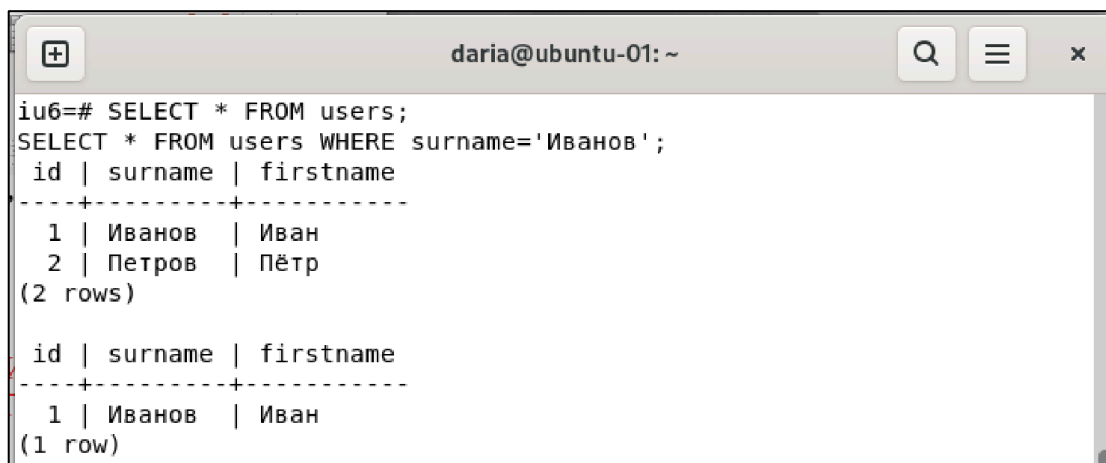
```
daria
lxd
rtkit
dnsmasq
whoopsie
usbmux
avahi
cups-pk-helper
speech-dispatcher
saned
colord
geoclue
pulse
gdm
postgres
admin
```

Рисунок 4 – Добавление в Ubuntu нового пользователя

На следующем этапе была создана сущность “users” в соответствии с текстом методическим пособием. После чего были выполнены запрос к созданной базе данных. Результаты изображены на рисунках 5-6.

```
iu6=# \dt
          I
      List of relations
 Schema | Name  | Type  | Owner
-----+-----+-----+-----
 public | users | table | admin
(1 row)
```

Рисунок 5 – Создание сущности users



The screenshot shows a terminal window titled "daria@ubuntu-01: ~". It contains the following SQL queries and their results:

```
iu6=# SELECT * FROM users;
SELECT * FROM users WHERE surname='Иванов';
id | surname | firstname
-----+-----+-----
 1 | Иванов  | Иван
 2 | Петров  | Пётр
(2 rows)

id | surname | firstname
-----+-----+-----
 1 | Иванов  | Иван
(1 row)
```

Рисунок 6 – Выполнение запросов

Работа с PostgreSQL в Python

Далее была установлена библиотека `psycopg2` для Python3, после чего была запущена программа, результат которой изображен на рисунке 7.

```
...  
(1, 'Иванов', 'Иван')  
(2, 'Петров', 'Пётр')  
>>>
```

Рисунок 7 – Результат работы программы, написанной на Python

Реализация предметной области (вариант 1)

Сервис размещения промоутирующих (продвигаемых, рекламируемых) текстов/изображений. Каждый промоутирующий блок может быть привязан к рекламной компании, которая может быть включена/выключена.

Задание:

1. Разработать схему базы данных, состоящую из 3-4 таблиц. Описать её в базе `iub`.
2. Включить в каждую таблицу несколько записей.
3. Написать и реализовать в виде одного оператора `Select` следующий запрос: «Найти имена сайтов и рекламных компаний, которые продвинули на эти сайты промоутирующие блоки, в текстовой части которых встречается основа слова «автомобиль»».

Выполнение

В соответствии с предметной областью задания была предложена структура базы данных, даталогическая модель которой изображена на рисунке 8.

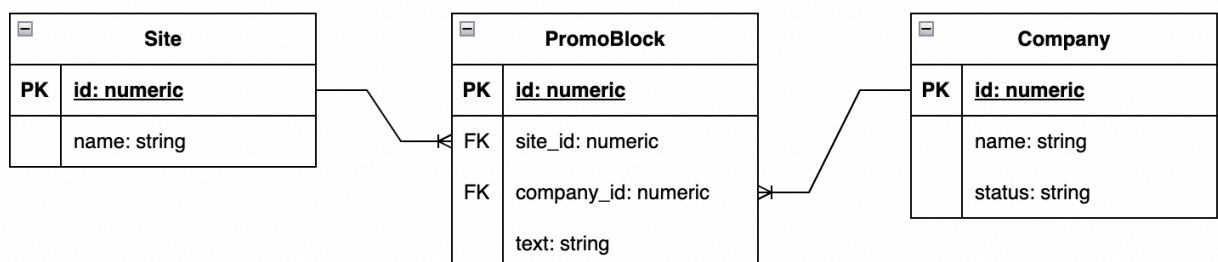


Рисунок 8 – Даталогическая модель базы данных

Далее были созданы соответствующие таблицы и выполнено их заполнение, что изображено на рисунках 9-11.

```

iu6=# SELECT * FROM "SITE";
 id |      name
-----+-----
  1 | autospot.ru
  2 | musicsell.com
  3 | climatcontrol.ru
(3 строки)

```

Рисунок 9 – Сущность SITE

```

iu6=# SELECT * FROM "COMPANY";
 id |      name      | status
-----+-----+-----
  1 | Mitsubishi     | включена
  2 | Yamaha         | включена
  3 | Haier          | выключена
(3 строки)

```

Рисунок 10 – Сущность COMPANY

id	text	site_id	company_id
5	Приобретите автомобиль Mitsubishi по выгодной цене только сегодня	1	1
6	Кондиционер от официального дилера Mitsubishi	3	1
7	Самые качественные гитары Yamaha	2	2
8	Кондиционер Haier со скидкой 50%	3	3

(4 строки)

Рисунок 11 – Сущность PROMOBLOCK

В конце был реализован оператор SELECT в соответствии с заданием. Результат его работы изображен на рисунке 12.

```

iu6=# SELECT "SITE".name, "COMPANY".name
FROM "SITE"
INNER JOIN "PROMOBLOCK" ON "PROMOBLOCK".site_id = "SITE".id
INNER JOIN "COMPANY" ON "COMPANY".id = "PROMOBLOCK".company_id
WHERE "PROMOBLOCK".text LIKE '%автомобиль%'
OR "PROMOBLOCK".text LIKE '%Автомобиль%';
 name |      name
-----+-----
 autospot.ru | Mitsubishi
(1 строка)

```

Рисунок 12 – Результат работы запроса

Исходный код задания

```
CREATE SEQUENCE site_ids;  
CREATE TABLE "SITE" (id INTEGER PRIMARY KEY DEFAULT  
NEXTVAL('site_ids'), name VARCHAR(64));
```

```
CREATE SEQUENCE company_ids;  
CREATE TABLE "COMPANY" (id INTEGER PRIMARY KEY DEFAULT  
NEXTVAL('company_ids'), name VARCHAR(64), status VARCHAR(64));
```

```
CREATE SEQUENCE promoblock_ids;  
CREATE TABLE "PROMOBLOCK" (id INTEGER PRIMARY KEY DEFAULT  
NEXTVAL('promoblock_ids'), text VARCHAR(120), site_id INTEGER REFERENCES  
"SITE"), company_id INTEGER REFERENCES "COMPANY");
```

```
INSERT INTO "SITE" (name) VALUES ('autospot.ru');  
INSERT INTO "SITE" (name) VALUES ('music sell.com');  
INSERT INTO "SITE" (name) VALUES ('climatcontrol.ru');
```

```
INSERT INTO "COMPANY" (name, status) VALUES ('Mitsubishi', 'включена');  
INSERT INTO "COMPANY" (name, status) VALUES ('Yamaha', 'включена');  
INSERT INTO "COMPANY" (name, status) VALUES ('Haier', 'выключена');
```

```
INSERT INTO "PROMOBLOCK" (text, site_id, company_id) VALUES ('Приобретите  
автомобиль Mitsubishi по выгодной цене только сегодня', 1, 1);  
INSERT INTO "PROMOBLOCK" (text, site_id, company_id) VALUES ('Кондиционер  
от официального дилера Mitsubishi', 3, 1);  
INSERT INTO "PROMOBLOCK" (text, site_id, company_id) VALUES ('Самые  
качественные гитары Yamaha', 2, 2);  
INSERT INTO "PROMOBLOCK" (text, site_id, company_id) VALUES ('Кондиционер  
Haier со скидкой 50%', 3, 3);
```

```
SELECT "SITE".name, "COMPANY".name  
FROM "SITE"  
INNER JOIN "PROMOBLOCK" ON "PROMOBLOCK".site_id = "SITE".id  
INNER JOIN "COMPANY" ON "COMPANY".id = "PROMOBLOCK".company_id  
WHERE "PROMOBLOCK".text LIKE '%автомобиль%'  
OR "PROMOBLOCK".text LIKE '%Автомобиль%';
```