|  |  |
| --- | --- |
|  | **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**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 2 |

**Название:** Оптимизация запросов. Основы EXPLAIN в PostgreSQL. Индексация

**Дисциплина:** Технология параллельных систем баз данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-12М |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | |  |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2023

**Введение**

**1. Цель работы:** формирование следующей компетенции: студент должен получить навыки работы с командой EXPLAIN. Он также должен познакомиться с эффективными методами индексации в PostgreSQL.

**Ход выполнения**

**2. Создание виртуальной машины**

Для выполнения курса лабораторных работ с помощью ПО “Virtula Box” была создана виртуальная машина на базе ОС Ubuntu 22.04 (рисунок 1).

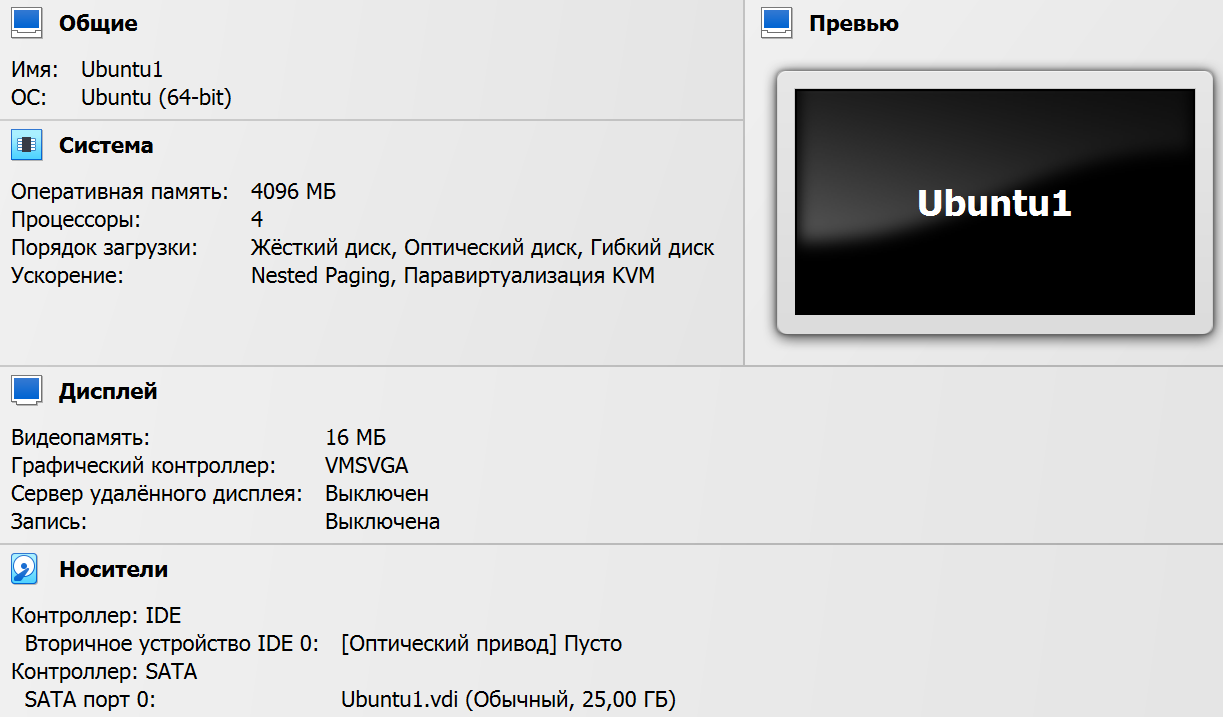


Рисунок 1 — параметры виртуальной машины

**3. Инсталляция СУБД PostgreSQL**

На рисунке 2 показано добавление ссылки на deb-пакет, получение соответствующего ключа и обновление индекса пакетов. Затем была проведена установка пакета командой “sudo apt -y install postgresql-9.6”.

На рисунке 3 показана проверка установки psql (запрос пути к исполняемому файлу psql), процесс установки пароля, переключение пользователя на postgres, создание и подключение к базе данных “iu6”.



Рисунок 2 — загрузка deb-пакета

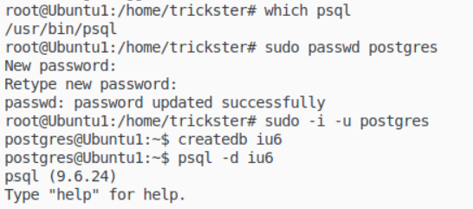


Рисунок 3 — проверка установки и создание базы данных

Создадим пользователя “admin” (рисунок 4). Проверим его существование, подключимся от его имени, создадим последовательность “user\_ids” и таблицу “user”. Заполним таблицу и проверим ее содержимое (рисунок 5).

Отберем из таблицы только пользователей с фамилией “Ivanov” (рисунок 6).

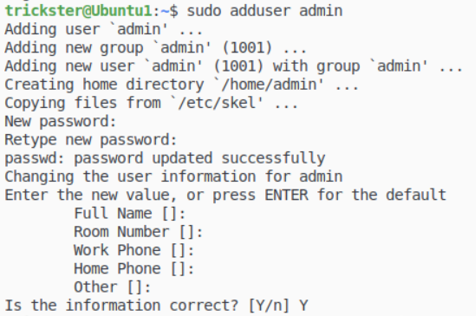


Рисунок 4 — создание пользователя admin

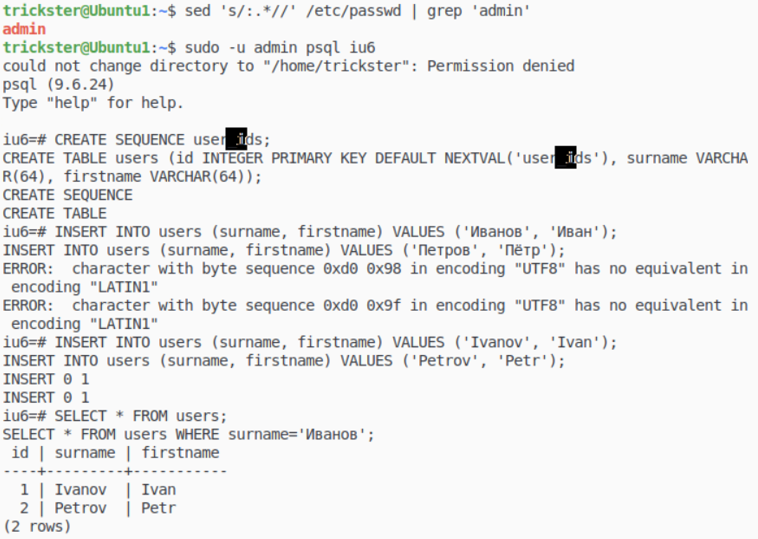


Рисунок 5 — создание таблицы “users”



Рисунок 6 — пример select-запроса

Запросим из БД информацию о таблицах, расширенную информацию о таблицах, информацию о таблице “users”, переключимся между пользователями (рисунок 7).

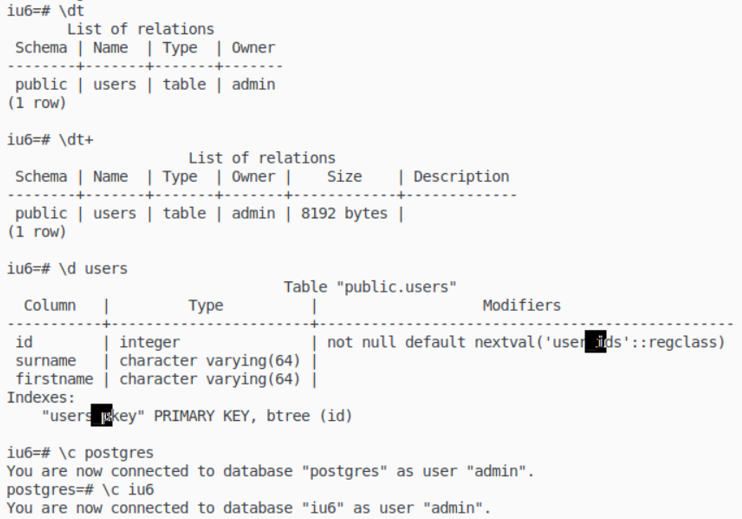


Рисунок 7 — выполнение полезных команд

**4. Работа с PostgreSQL в Python**

Установим библиотеку psycopg2 для Python3 (рисунок 8).

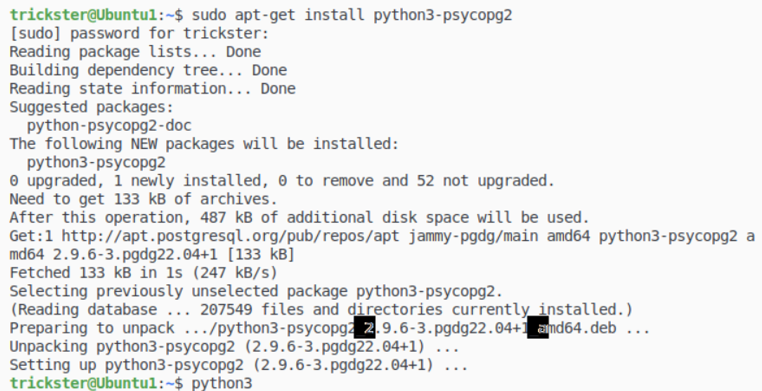


Рисунок 7 — установка psycopg2

Создадим файл ex1.py, запрашивающий 10 записей из таблицы “users” и выполним его (рисунки 8-9).



Рисунок 8 — текст программы для выполнения запроса к БД



Рисунок 9 — результат выполнения программы

**5. Реализация предметной области**

**Задание (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 iu6.

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».

Разработанная схема БД представлена на рисунке 10. В ней представлены следующие таблицы:

* “promotion\_blocks” — рекламные блоки;
* “promotion\_companies” — компании;
* “promotion\_sites” — сайты для размещения рекламы;
* “promotion\_bindings” — служебная таблица для реализации связи “многие-ко-многим” между сайтами и компаниями.

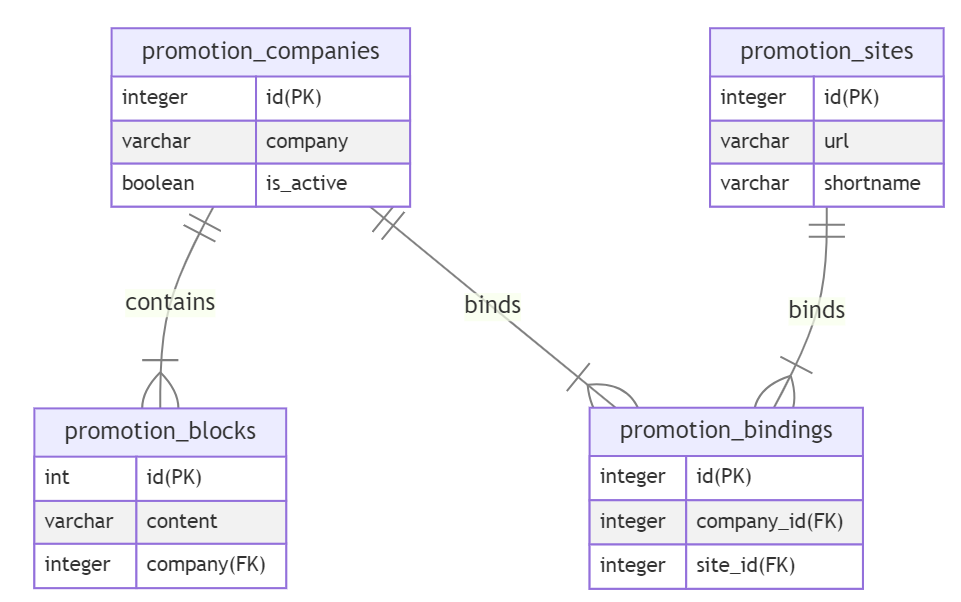


Рисунок 10 — схема БД

Операторы SQL, создающие описанные таблицы и заполняющие их приведены в листинге 1.

Листинг 1 — создание и заполнение таблиц

|  |
| --- |
| CREATE SEQUENCE promotion\_blocks\_ids;  CREATE TABLE promotion\_blocks (  id INTEGER PRIMARY KEY DEFAULT NEXTVAL('promotion\_blocks\_ids'),  content VARCHAR(250),  company INTEGER  );  INSERT INTO promotion\_blocks (content, company) VALUES ('Toyta - drive your dream', 1);  INSERT INTO promotion\_blocks (content, company) VALUES ('Toyta - japaniese auto qulity', 1);  INSERT INTO promotion\_blocks (content, company) VALUES ('Mercedes - das auto', 2);  INSERT INTO promotion\_blocks (content, company) VALUES ('Mercedes - deutch auto #1', 2);  INSERT INTO promotion\_blocks (content, company) VALUES ('Lada - family automobile', 3);  INSERT INTO promotion\_blocks (content, company) VALUES ('Lada - new horizons', 3);  INSERT INTO promotion\_blocks (content, company) VALUES ('Ocuvue - clear vision', 4);  CREATE SEQUENCE promotion\_companies\_ids;  CREATE TABLE promotion\_companies (  id INTEGER PRIMARY KEY DEFAULT NEXTVAL('promotion\_companies\_ids'),  company VARCHAR(30),  is\_active boolean  );  INSERT INTO promotion\_companies (company, is\_active) VALUES ('Toyota', TRUE);  INSERT INTO promotion\_companies (company, is\_active) VALUES ('Mercedez Benz', FALSE);  INSERT INTO promotion\_companies (company, is\_active) VALUES ('LADA', TRUE);  INSERT INTO promotion\_companies (company, is\_active) VALUES ('Ocuvue', FALSE);  CREATE SEQUENCE promotion\_sites\_ids;  CREATE TABLE promotion\_sites (  id INTEGER PRIMARY KEY DEFAULT NEXTVAL('promotion\_sites\_ids'),  url VARCHAR(50),  shortname VARCHAR(15)  );  INSERT INTO promotion\_sites (url, shortname) VALUES ('ya.ru', 'yandex');  INSERT INTO promotion\_sites (url, shortname) VALUES ('youtube.com', 'youtube');  INSERT INTO promotion\_sites (url, shortname) VALUES ('vk.com', 'vKontake');  CREATE SEQUENCE promotion\_bindings\_ids;  CREATE TABLE promotion\_bindings (  id INTEGER PRIMARY KEY DEFAULT NEXTVAL('promotion\_bindings\_ids'),  company\_id INTEGER,  site\_id INTEGER  );  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (1, 1);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (1, 2);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (2, 2);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (2, 3);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (3, 1);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (3, 3);  INSERT INTO promotion\_bindings (site\_id, company\_id) VALUES (4, 2); |

Оператор select, реализующий требуемый запрос, состоит из объединения всех таблиц по ключам и применения условия для текста рекламного блока. Текст и результат выполнения запроса приведены на рисунке 10.

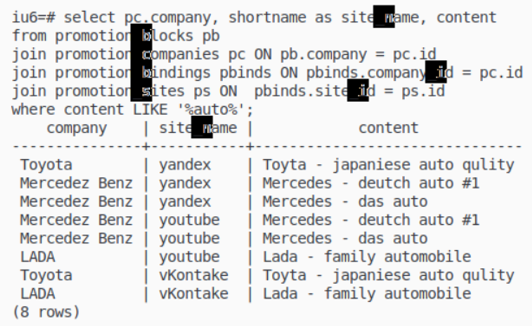


Рисунок 10 — запрос и результат его выполнения

**Вывод:** В ходе выполнения данной лабораторной работы были получены навыки инсталляции СУБД PostgreSQL в ОС Ubuntu. Была протестирована ее работа, а также создана небольшая база данных и разработан запрос к ней в соответствии с вариантом, выданным преподавателем. Задание было успешно выполнено в соответствии с выданным вариантом.