**Задание SQL**

**Часть 2. Соединения таблиц**

1. **При создании БД использована СУБД PostgreSQL и графическая утилита pgAdmin**

-- Database: inventory\_db

-- DROP DATABASE IF EXISTS inventory\_db;

CREATE DATABASE inventory\_db

WITH

OWNER = postgres

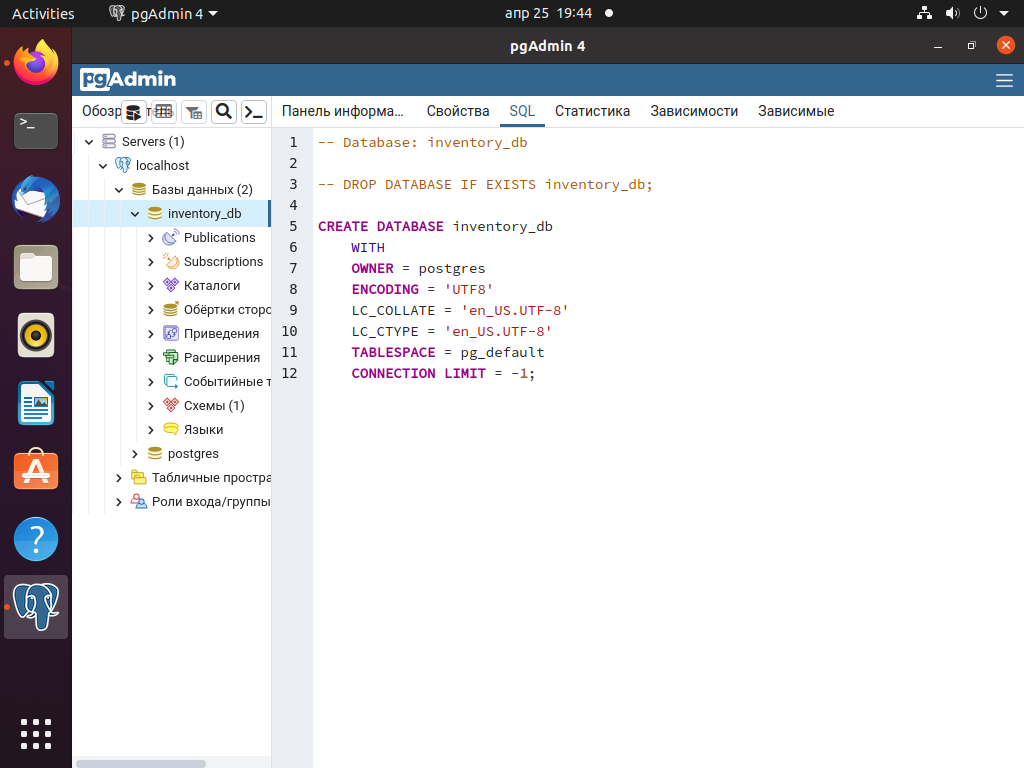
ENCODING = 'UTF8'

LC\_COLLATE = 'en\_US.UTF-8'

LC\_CTYPE = 'en\_US.UTF-8'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;



1. **Созданы таблицы Users и Computers. Между таблицами задано отношение 1:M (Users.id -> Computers.id\_user(FK))**

**2.1 Таблица Users**

-- Table: public.Users

-- DROP TABLE IF EXISTS public."Users";

CREATE TABLE IF NOT EXISTS public."Users"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

name character varying(25) COLLATE pg\_catalog."default",

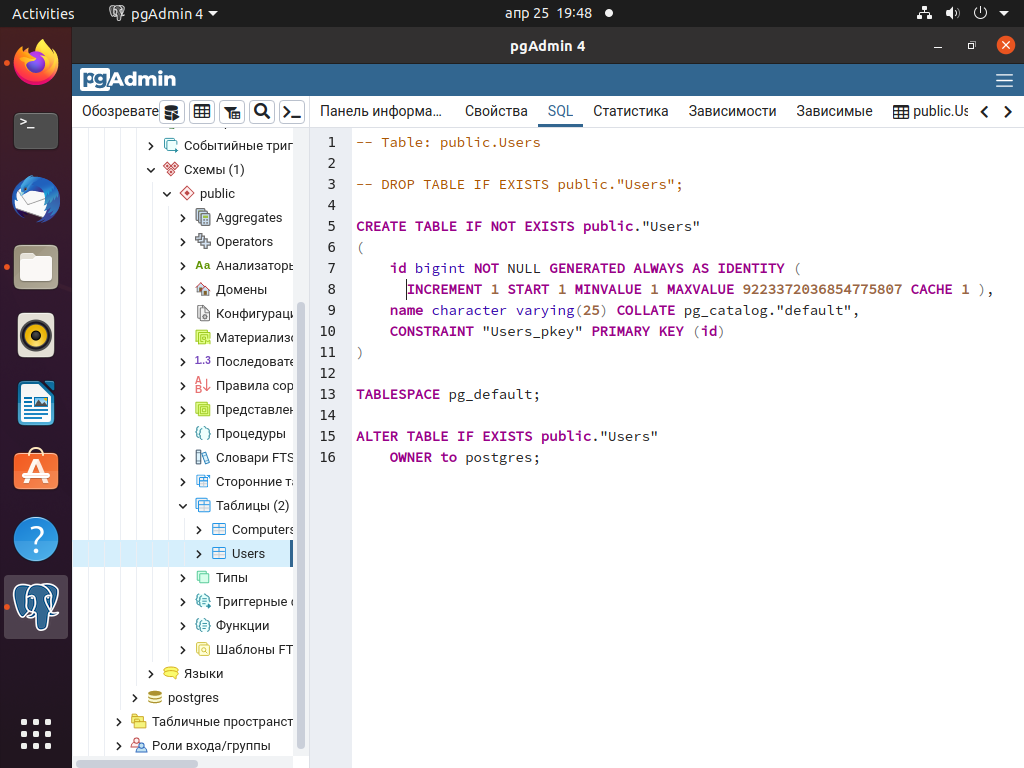
CONSTRAINT "Users\_pkey" PRIMARY KEY (id)

)

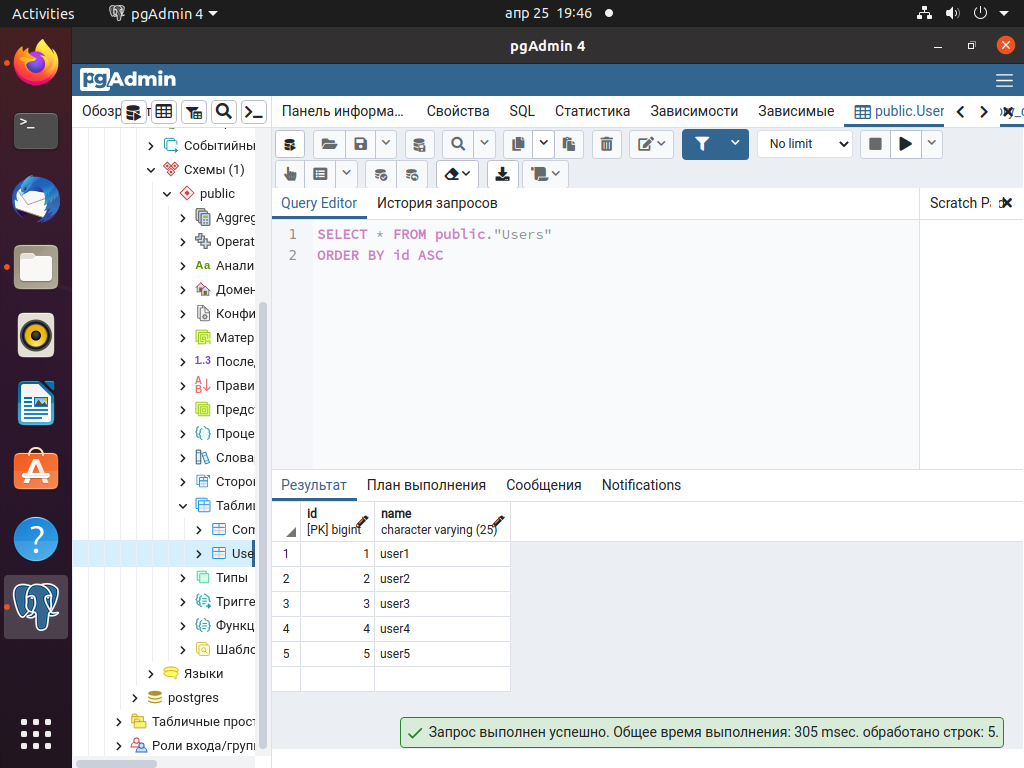
TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Users"

OWNER to postgres;



**2.2 Внесены данные из задания в таблицу Users**



**2.3 Таблица Computers**

-- Table: public.Computers

-- DROP TABLE IF EXISTS public."Computers";

CREATE TABLE IF NOT EXISTS public."Computers"

(

id bigint NOT NULL GENERATED ALWAYS AS IDENTITY ( INCREMENT 1 START 1 MINVALUE 1 MAXVALUE 9223372036854775807 CACHE 1 ),

name character varying(25) COLLATE pg\_catalog."default" NOT NULL,

id\_user bigint,

CONSTRAINT "Computers\_pkey" PRIMARY KEY (id),

CONSTRAINT "Computers\_fkey" FOREIGN KEY (id\_user)

REFERENCES public."Users" (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

NOT VALID

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public."Computers"

OWNER to postgres;

-- Index: fki\_Computers\_fkey

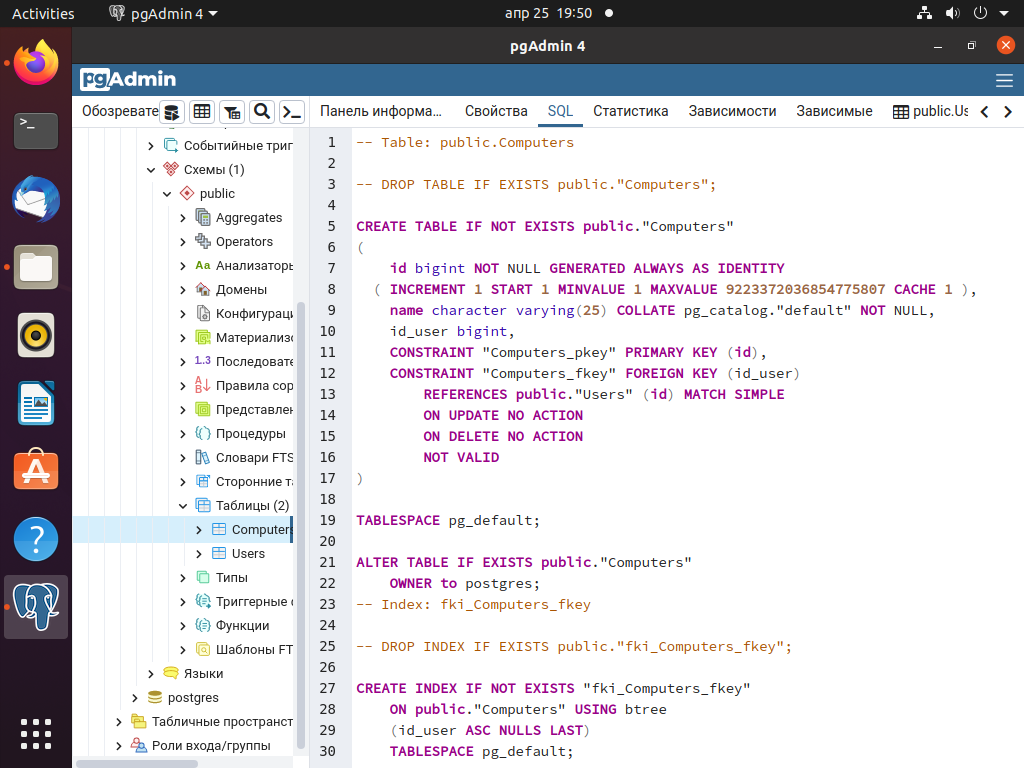
-- DROP INDEX IF EXISTS public."fki\_Computers\_fkey";

CREATE INDEX IF NOT EXISTS "fki\_Computers\_fkey"

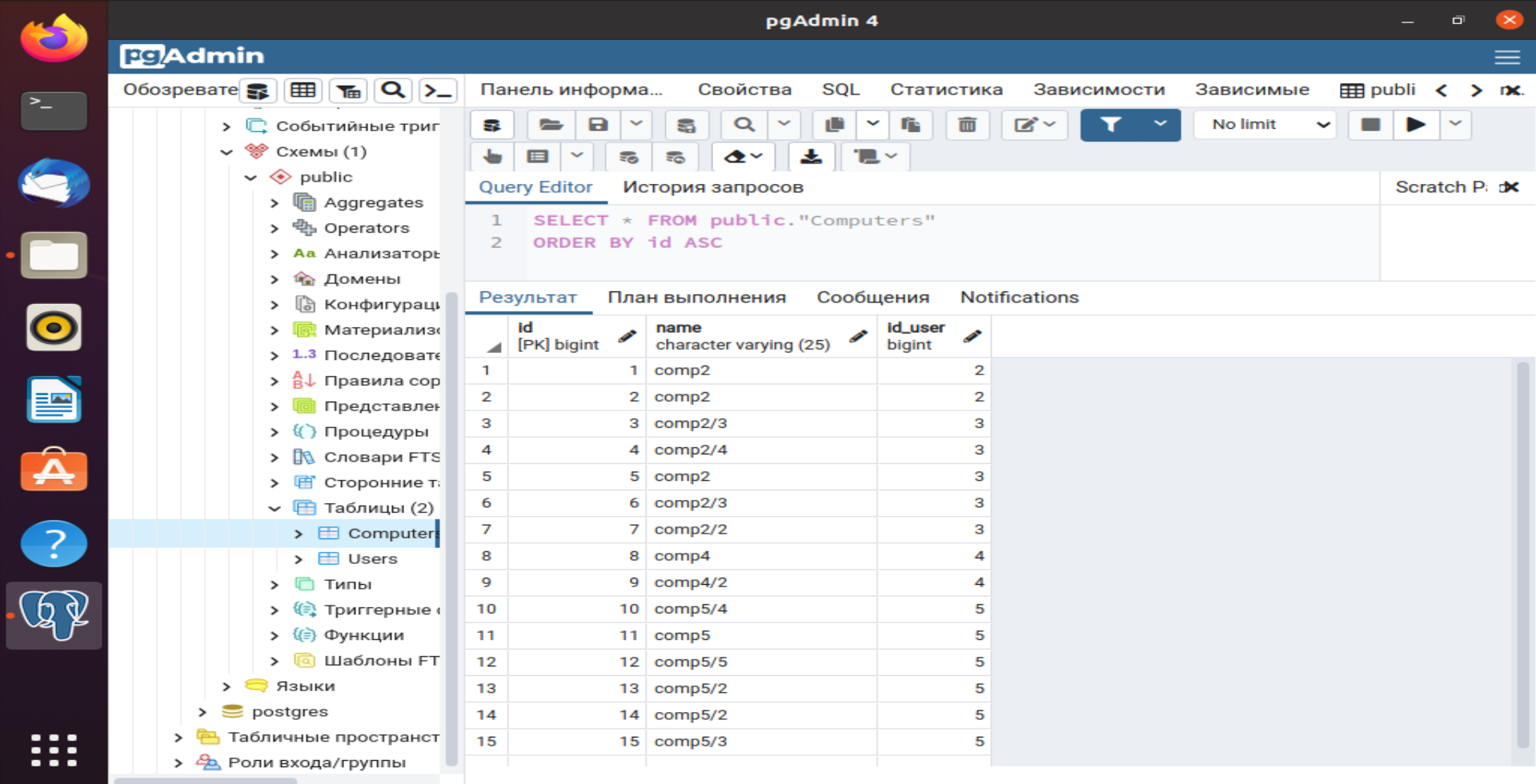
ON public."Computers" USING btree

(id\_user ASC NULLS LAST)

TABLESPACE pg\_default;



**2.4 Внесены данные из задания в таблицу Computers**



1. **Составление запросов**

Из данных к заданию можно сделать вывод, что отношение Computers не приведено ко 2-НФ, вследствие чего возникла избыточность, например, comp2 закреплен 2 раза за пользователем с id=2.

Также непонятно – уникальны ли имена компьютеров и может ли компьютер с одним именем выдаваться разным пользователям, например, comp2 выдан сразу 2 пользователям с id 2 и id=3.

Будем считать, что имя компьютера уникально, но один компьютер может выдаваться разным пользователям, если компьютер с определенным именем выдан несколько раз одному пользователю – избыточность, вследствие отсутствия нормализации отношения.

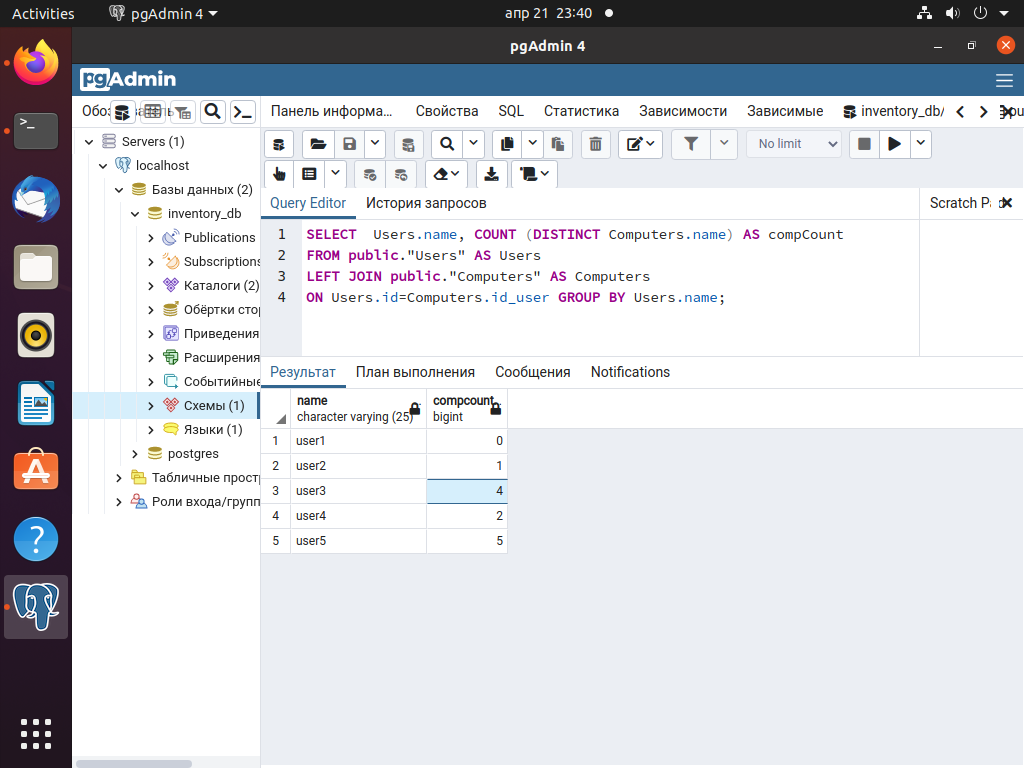
**3.1**  **Подсчет выданных каждому пользователю компьютеров**

SELECT Users.name, COUNT (DISTINCT Computers.name) AS compCount

FROM public."Users" AS Users

LEFT JOIN public."Computers" AS Computers

ON Users.id=Computers.id\_user GROUP BY Users.name;



**3.2 Выборка пользователей с количеством выданных компьютеров больше 3**

SELECT \* FROM

(SELECT Users.name, COUNT (DISTINCT Computers.name) AS compCount

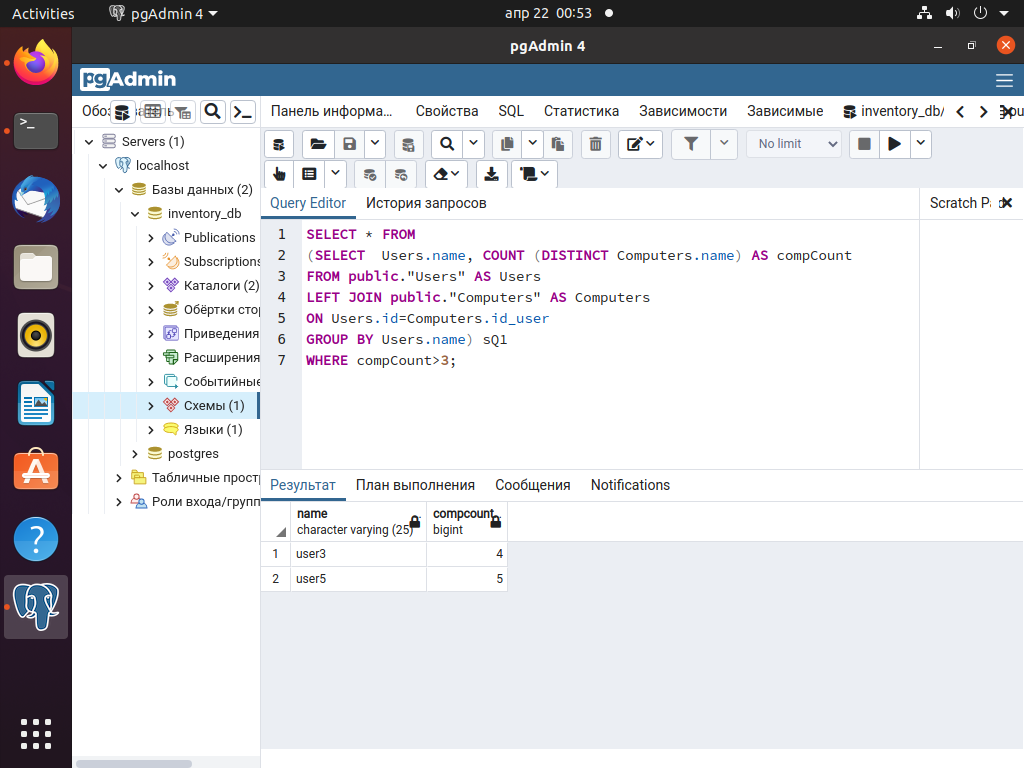
FROM public."Users" AS Users

LEFT JOIN public."Computers" AS Computers

ON Users.id=Computers.id\_user

GROUP BY Users.name) sQ1

WHERE compCount>3;



**3.3 Выборка пользователей, которым не выдано компьютеров**

SELECT \* FROM

(SELECT Users.name, COUNT (DISTINCT Computers.name) AS compCount

FROM public."Users" AS Users

LEFT JOIN public."Computers" AS Computers

ON Users.id=Computers.id\_user

GROUP BY Users.name) sQ1

WHERE compCount=0;

