Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**РАБОТА С УТИЛИТОЙ PSQL**

Отчёт о лабораторной работе № 1

по дисциплине «Базы данных»

Выполнил: студент гр. 431-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гурулёв А. В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Проверил: ассистент каф. АСУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Яблонский Я. В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023

г.Томск 2023

Оглавление

[1 Цель лабораторной работы 3](#_Toc150714694)

[2 Описание таблиц БД из индивидуального задания 4](#_Toc150714695)

[3 Описание процесса реализации таблиц 6](#_Toc150714696)

[4 Выводы 14](#_Toc150714697)

# Цель лабораторной работы

Цели:

* ознакомиться с принципами работы консольной утилиты psql;
* научиться создавать макеты таблиц с использованием команд SQL.

Задачи:

* используя psql создать таблицы GOODS;
* используя psql создать таблицы CLIENT;
* ввести в таблицы тестовые данные (1-2 строки);
* убедиться в том, что поддерживаются ограничения целостности атрибута, сущности и ссылочные.

# Описание таблиц БД из индивидуального задания

Список таблиц и их описание можно увидеть на рисунке 2.1.

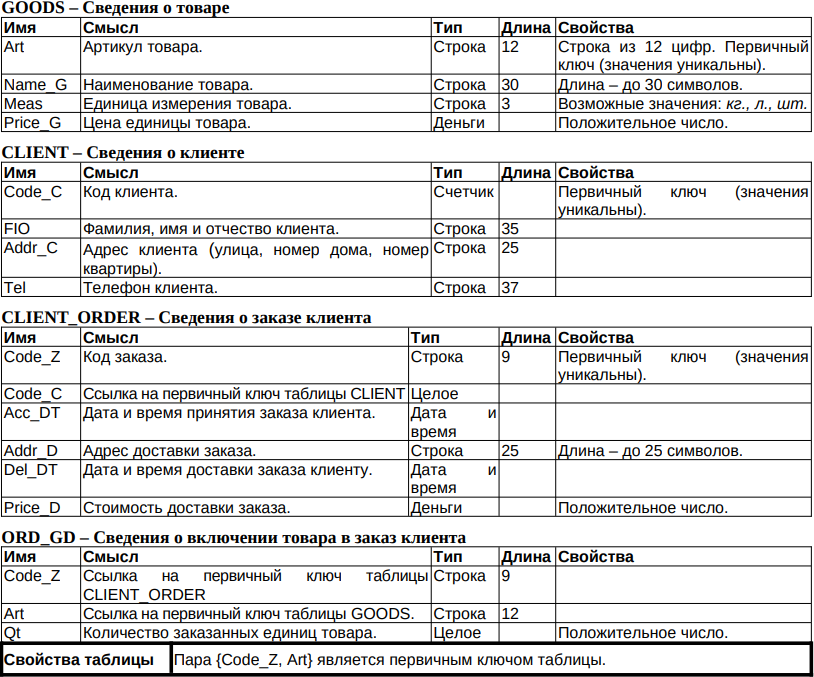


Рисунок . - Таблицы базы данных

Также действуют следующие ограничения целостности:

1. Порядковый номер записи о клиенте в книге регистрации считается кодом клиента.
2. При регистрации клиент может указать до трёх номеров контактных телефонов.
3. Число заказов клиента в течение дня не ограничено.
4. Моментом принятия заказа считается момент окончания оформления счёта.
5. Количество наименований товаров, включённых в один заказ, не ограничено.
6. Количество единиц товара конкретного наименования, включённых в один заказ, не ограничено.
7. Определённый товар может быть включён в определённый заказ только один раз.

# Описание процесса реализации таблиц

Для начала была создана база данных, используя следующий запрос: CREATE DATABASE labs;

Где:

* CREATE DATABASE – команда на создание базы данных с указанным именем;
* labs – имя таблицы.

Далее было проведено подключение к базе данных “labs” при помощи команды “\c”. Команда и ответ на неё изображен на рисунке 3.1.

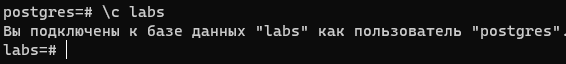


Рисунок . - Команда "\c" и ответ на неё

Также, для удобства, сменена кодировка, для корректного отображения кириллицы, при помощи команды “\! chcp 1251”.

После, для создания таблицы “goods” был сформирован следующий запрос:

CREATE TABLE IF NOT EXISTS goods(

art char(12) PRIMARY KEY CHECK(art similar to '\d{12}'),

name\_g varchar(30) CHECK(name\_g similar to '[[:alpha:]]+'),

meas varchar(3) CHECK(meas similar to '(кг|л|шт)\.'),

price\_g money CHECK(price\_g > '0')

);

Где:

* CREATE TABLE – команда на создание таблицы, с указанным названием и свойствами, описанными внутри круглых скобок, после названия;
* IF NOT EXISTS – условие, что команда выполняется только если таблица ещё не существует;
* goods – название таблицы;
* name\_g – имя атрибута 1;
* char(12) – тип данных, для атрибута 1, который соответствует строке, длинною до 12 символов;
* PRIMARY KEY – указывает, что атрибут 1 является первичным ключом таблицы;
* CHECK – проверка вводимых данных для атрибута 1, на соответствие условиям, указным внутри круглых скобок;
* similar to – указывает, что атрибут, указанный слева, должен соответствовать шаблону, составленному при помощи регулярных выражений, справа;
* ‘\d{12}’ – регулярное выражение, обозначающая строку из 12 цифр;

Все остальные атрибуты описываются схожим образом, поэтому рассмотрим только регулярные выражения, заданные в таблице:

* ‘[[:alpha:]]+’ - строка должна содержать только одно слово, состоящее из букв, длинной от 1 символа до лимита;
* ‘(кг|л|шт)\.’ – строка может содержать либо “кг”, либо “л”, либо “шт” с точкой в конце;
* CHECK(price\_g > ‘0’) – проверка, является ли атрибут “price\_g” положительным числом.

Отправку запроса и ответ на него можно увидеть на рисунке 3.2.

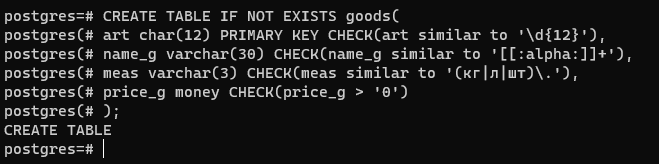


Рисунок . – Запрос на создание таблицы “goods” и ответ на него

Далее, схожим образом, была создана таблица “client”:

CREATE TABLE IF NOT EXISTS client(

code\_c int GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

fio varchar(35) CHECK(fio similar to '[А-ЯЁA-Z][а-яёa-z]+( [А-ЯЁA-Z][а-яёa-z]+){1,2}'),

addr\_c varchar(25) CHECK(addr\_c similar to '[[:alpha:]]+, \d{1,3}[[:alpha:]]?, \d{1,3}'),

tel varchar(38) CHECK(tel similar to '\+\d{11}( \+\d{11}){0,2}')

);

Ответ на запрос можно увидеть на рисунке 3.3.

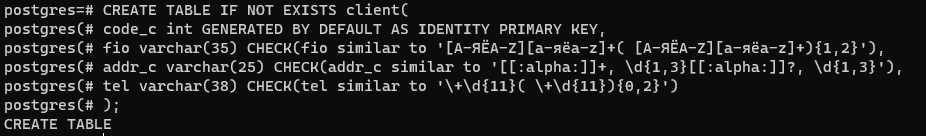


Рисунок . - Запрос на создание таблицы "client" и ответ на него

Далее были внесены тестовые данные:

INSERT INTO goods VALUES('000000000000', 'яйцо', 'шт.', 10),

('000000000001', 'молоко', 'л.', 100),

('000000000002', 'мука', 'кг.', 150),

('000000000003', 'хлеб', 'шт.', 20),

('000000000004', 'масло', 'л.', 98.9);

INSERT INTO client(fio, addr\_c, tel) VALUES('Александр Гурулёв Владиславович', 'Мокрушина, 1А, 51', '+79509977215'),

('Иван Иванов Иванович', 'Мокрушина, 1А, 52', '+79509778223'),

('Петя Петров Петрович', 'Мокрушина, 1А, 53', '+79764659369'),

('Морозова Екатерина Елисеевна', 'Мокрушина, 1А, 54', '+79088931985 +79361942060'),

('Пахомов Роман Дмитриевич', 'Мокрушина, 1А, 55', '+79066651624 +79058183870 +79478783155');

Где:

* INSERT INTO – команда на внесение данных в указанную таблицу;
* client или goods – имя таблицы, куда вносятся данные, после, внутри круглых скобок, вносится список интересующих атрибутов, если нужны все, можно их не указывать;
* VALUES – внутри круглы скобок, после этой команды, перечисляются данные, которые мы хотим внести.

На рисунке 3.4 можно увидеть результаты этих запросов.

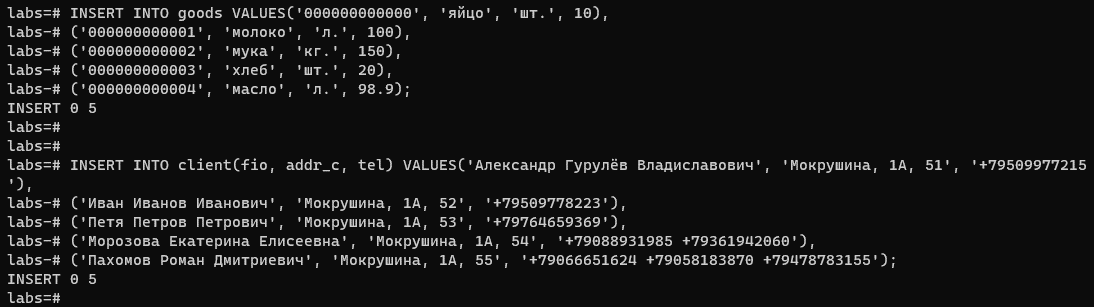


Рисунок . – Результаты запросов на внесение данных

Далее необходимо проверить, работают ли ограничения, которые были заданы. Для этого были сформированы следующие запросы для таблицы “goods”:

INSERT INTO goods(art) VALUES('000000000000');

INSERT INTO goods(art) VALUES('00000000000');

INSERT INTO goods(art) VALUES('0000000000000');

INSERT INTO goods(art) VALUES('kkkkkkkkkkkk');

INSERT INTO goods(art) VALUES('kkkkkkkkkkk');

INSERT INTO goods(art) VALUES('kkkkkkkkkkkkk');

UPDATE goods SET name\_g = 'na me' WHERE art = '000000000000';

UPDATE goods SET name\_g = 'na\_me' WHERE art = '000000000000';

UPDATE goods SET name\_g = 'na1me' WHERE art = '000000000000';

UPDATE goods SET name\_g = '' WHERE art = '000000000000';

UPDATE goods SET meas = '' WHERE art = '000000000000';

UPDATE goods SET meas = 'кг' WHERE art = '000000000000';

UPDATE goods SET meas = 'кг..' WHERE art = '000000000000';

UPDATE goods SET meas = 'гр.' WHERE art = '000000000000';

UPDATE goods SET price\_g = -1 WHERE art = '000000000000';

UPDATE goods SET price\_g = 0 WHERE art = '000000000000';

UPDATE goods SET price\_g = 'ggg' WHERE art = '000000000000';

И для “client”:

UPDATE client SET fio = '' WHERE code\_c = 1;

UPDATE client SET fio = 'Александр ' WHERE code\_c = 1;

UPDATE client SET fio = 'АлександрГурулёв' WHERE code\_c = 1;

UPDATE client SET fio = ' Александр Гурулёв Владиславович' WHERE code\_c = 1;

UPDATE client SET fio = 'Александр Гурулёв Владиславович ' WHERE code\_c = 1;

UPDATE client SET fio = '214 24 241' WHERE code\_c = 1;

UPDATE client SET fio = 'александр гурулёв владиславович' WHERE code\_c = 1;

UPDATE client SET fio = 'АлЕкСаНдР ГуРуЛёВ вЛаДиСлАвОвИч' WHERE code\_c = 1;

UPDATE client SET addr\_c = '' WHERE code\_c = 1;

UPDATE client SET addr\_c = ' ' WHERE code\_c = 1;

UPDATE client SET addr\_c = 'Мокрушина1а51' WHERE code\_c = 1;

UPDATE client SET addr\_c = 'Мокрушина 1а 1' WHERE code\_c = 1;

UPDATE client SET addr\_c = 'Мокрушина, 1а 51' WHERE code\_c = 1;

UPDATE client SET addr\_c = ' Мокрушина, 1А, 51' WHERE code\_c = 1;

UPDATE client SET addr\_c = 'Мокрушина, 1А, 51 ' WHERE code\_c = 1;

UPDATE client SET tel = '' WHERE code\_c = 1;

UPDATE client SET tel = '89509977215' WHERE code\_c = 1;

UPDATE client SET tel = '+79509977215 ' WHERE code\_c = 1;

UPDATE client SET tel = '+79509977215 +79509977215 ' WHERE code\_c = 1;

UPDATE client SET tel = '+79509977215 +79509977215 +79509977215 ' WHERE code\_c = 1;

UPDATE client SET tel = ' +79509977215 +79509977215 +79509977215' WHERE code\_c = 1;

UPDATE client SET tel = ' +79509977215 +79509977215+79509977215' WHERE code\_c = 1;

UPDATE client SET tel = ' +79509977215+79509977215 +79509977215' WHERE code\_c = 1;

UPDATE client SET tel = ' +79509977215+79509977215+79509977215' WHERE code\_c = 1;

Результаты этих запросов можно увидеть на рисунках 3.5 – 3.6.



Рисунок . – Результаты запросов некорректного заполнения таблицы “goods”

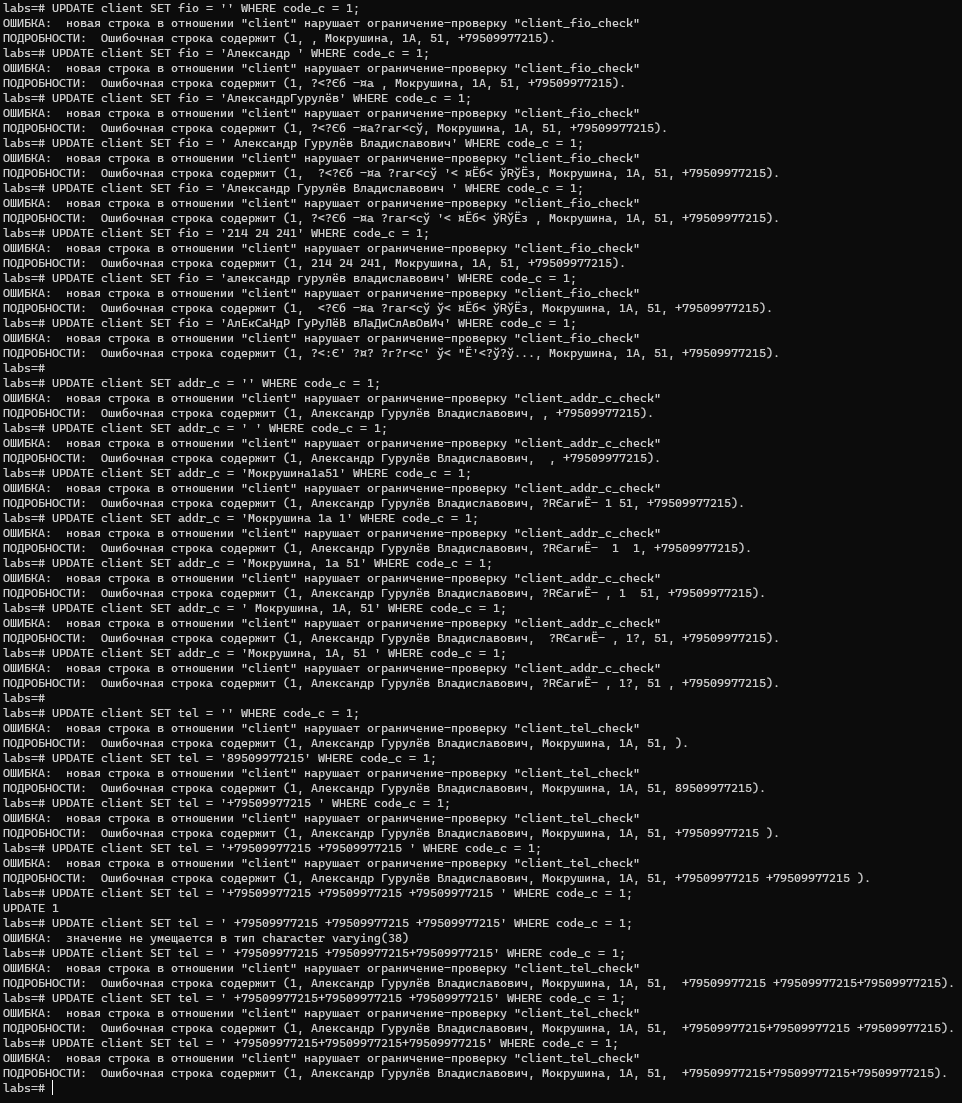


Рисунок . - Результаты запросов некорректного заполнения таблицы “client”

# Выводы

В ходе лабораторной работы:

* ознакомился с принципами работы консольной утилиты psql;
* научился создавать макеты таблиц с использованием команд SQL;
* научился устанавливать ограничение вводимых данных;
* научился вносить данные с использованием команд SQL.