**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 «Системы обработки информации и управления»**

Лабораторная работа №3

Курс «Базы данных»

«Использование представлений в СУБД PostgreSQL»

Выполнил:

студент группы ИУ5-45Б

Шакиров Т.М.

Проверил:

Маслеников К.Ю.

2024 г.

**Цель лабораторной работы:**

Получить теоретические и практические навыки создания представлений

данных в СУБД PostgreSQL.

Простое представление и запрос с помощью него данных:

CREATE VIEW customer\_order\_details AS

SELECT

o.order\_id,

o.order\_date,

c.customer\_id,

c.first\_name AS customer\_first\_name,

c.last\_name AS customer\_last\_name,

p.product\_id,

p.product\_name AS product\_name,

p.product\_price,

i.quantity

FROM

orders o

JOIN

customers c ON o.customer\_id = c.customer\_id

JOIN

items i ON o.order\_id = i.order\_id

JOIN

products p ON i.product\_id = p.product\_id;

SELECT \* FROM customer\_order\_details



Проверка соответствия данных прямым запросом:

SELECT

o.order\_id,

o.order\_date,

c.customer\_id,

c.first\_name AS customer\_first\_name,

c.last\_name AS customer\_last\_name,

p.product\_id,

p.product\_name AS product\_name,

p.product\_price,

i.quantity

FROM

orders o

JOIN

customers c ON o.customer\_id = c.customer\_id

JOIN

items i ON o.order\_id = i.order\_id

JOIN

products p ON i.product\_id = p.product\_id;



Изменение созданного представления с помощью команды ALTER VIEW, с добавлением псевдонимов полям:

ALTER VIEW customer\_order\_details RENAME COLUMN order\_id TO order\_number;

ALTER VIEW customer\_order\_details RENAME COLUMN order\_date TO date\_of\_order;

ALTER VIEW customer\_order\_details RENAME COLUMN customer\_id TO customer\_number;

ALTER VIEW customer\_order\_details RENAME COLUMN customer\_first\_name TO customer\_name;

ALTER VIEW customer\_order\_details RENAME COLUMN customer\_last\_name TO customer\_surname;

ALTER VIEW customer\_order\_details RENAME COLUMN product\_id TO product\_number;

ALTER VIEW customer\_order\_details RENAME COLUMN product\_name TO name\_of\_product;

ALTER VIEW customer\_order\_details RENAME COLUMN product\_price TO price;

ALTER VIEW customer\_order\_details RENAME COLUMN quantity TO quantity\_ordered;

**

Изменение запроса созданного представления с помощью команды CREATE OR

REPLACE VIEW, с добавлением в него условия:

CREATE OR REPLACE VIEW customer\_order\_details (order\_number, date\_of\_order, customer\_number, customer\_name, customer\_surname, product\_number, name\_of\_product, price, quantity\_ordered) AS

SELECT

o.order\_id,

o.order\_date,

c.customer\_id,

c.first\_name,

c.last\_name,

p.product\_id,

p.product\_name,

p.product\_price,

i.quantity

FROM

orders o

JOIN

customers c ON o.customer\_id = c.customer\_id

JOIN

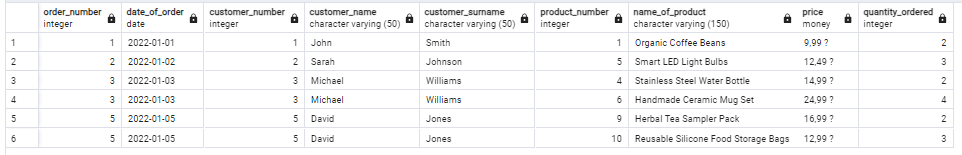
items i ON o.order\_id = i.order\_id

JOIN

products p ON i.product\_id = p.product\_id

WHERE

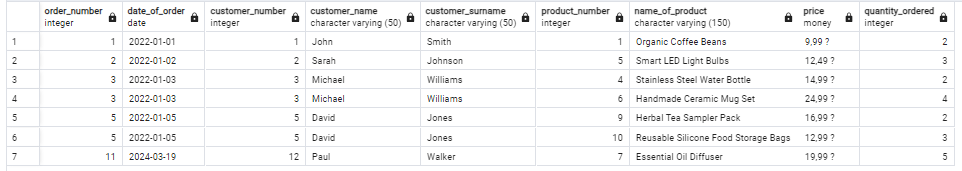
i.quantity >= 2



Вставка данных с помощью представления:

INSERT INTO orders (order\_id, customer\_id, order\_date, ship\_date, paid\_date, status)

VALUES (11, 12, '2024-03-19', '2024-03-20', '2024-03-22', 'P');



Создать представление с опцией WITH CHECK OPTION:

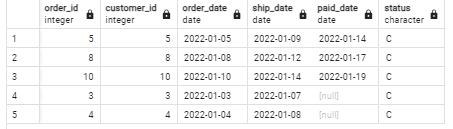
CREATE VIEW completed\_orders AS

SELECT \*

FROM orders

WHERE status = 'С'

WITH CHECK OPTION;

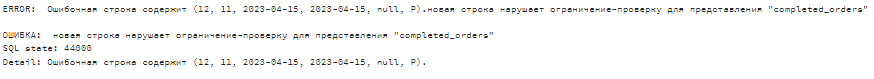
**

Проверка работы:

INSERT INTO completed\_orders (order\_id, customer\_id, order\_date, ship\_date, paid\_date, status)

VALUES

(12, 11, '2023-04-15', '2023-04-15', NULL, 'P')

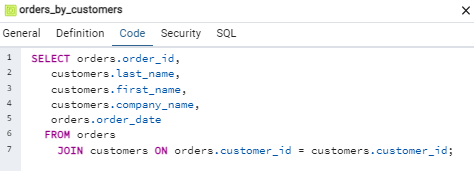


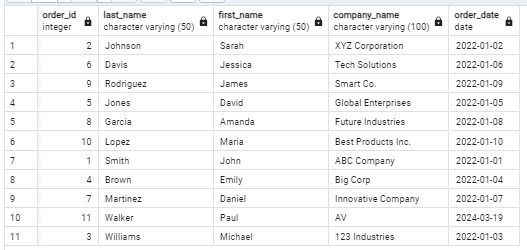
Удаление представление:

DROP VIEW completed\_orders

**

Создание представление на выборку из двух таблиц с помощью редактора:

**

**

Создание роли Test\_creator без права входа в систему, но с правом создания БД и ролей:

CREATE ROLE Test\_creator WITH

NOLOGIN

NOSUPERUSER

CREATEDB

CREATEROLE

INHERIT

NOREPLICATION

NOBYPASSRLS

CONNECTION LIMIT -1

PASSWORD 'xxxxxx';

Создание пользователя user1 с правом входа в систему:

CREATE ROLE user1 WITH

LOGIN

NOSUPERUSER

NOCREATEDB

NOCREATEROLE

INHERIT

NOREPLICATION

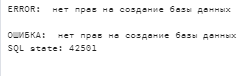
NOBYPASSRLS

CONNECTION LIMIT -1

PASSWORD 'xxxxxx';

Проверка, что user1 не может создать БД:

CREATE DATABASE testdb

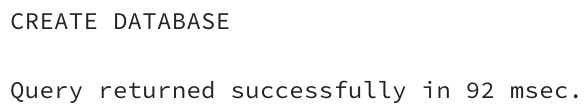
**

Включение пользователя user1 в группу Test\_creator:

GRANT test\_creator TO user1;

Создать БД под пользователем user1, для проверки создать новое подключение для пользователя user1 с ролью Test\_creator.

CREATE DATABASE testdb;



Создание роли без права создания таблицы:

CREATE ROLE role\_without\_create\_table

LOGIN

PASSWORD '123';



С правом создания таблицы:

CREATE ROLE role\_with\_create\_table WITH

CREATEDB

LOGIN

PASSWORD '321';



GRANT CREATE ON SCHEMA PUBLIC TO role\_with\_create\_table



Последовательная проверка работы ролей:

CREATE TABLE cars(

car\_id int PRIMARY KEY,

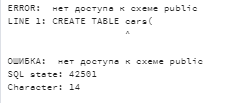
car\_name text

)

role\_with\_create\_table



role\_without\_create\_table



Добавить к роли право на любые действия с таблицей:

CREATE ROLE all\_privileges

GRANT ALL PRIVILEGES ON TABLE person TO all\_role



Проверка работы прав:

* Вставка новой строки в таблицу:

INSERT INTO cars (car\_id, car\_name)

VALUES

(1, 'Мерседес'),

(2, 'БМВ'),

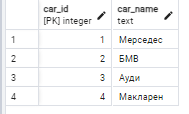
(3, 'Ауди'),

(4, 'Макларен')



* Просмотр данных из таблицы:

SELECT \* FROM cars;

**

* Изменение строки в таблице:

UPDATE cars

SET car\_name = 'Ферари'

WHERE car\_id = 4



* Удаление строки из таблицы:

DELETE FROM cars

WHERE car\_id = 3



Удалить право вставки в таблицу, проверить работу прав:

REVOKE INSERT ON TABLE cars FROM all\_privileges



INSERT INTO cars (car\_id, car\_name)

VALUES

(5, 'Ламборгини')

