



**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 ИУ5 «Системы обработки информации и управления»**

Курс «Базы данных»

Отчет по лабораторной работе №3

«Использование представлений в базах данных PostgreSQL»

**Выполнил:
студент группы ИУ5-41Б
Кашима А.**

**Проверил:
преподаватель каф. ИУ5
Ковалева Н.А.**

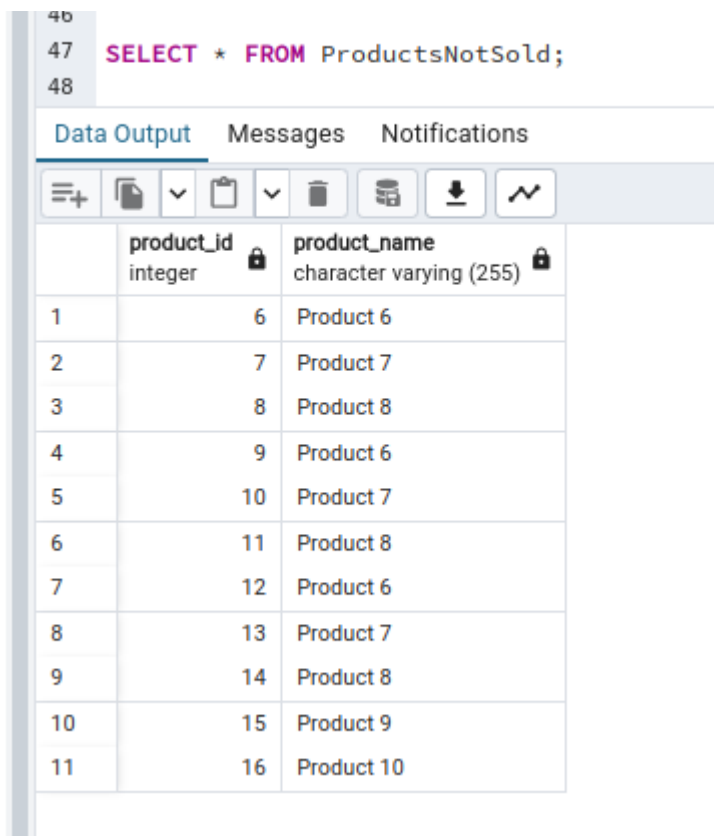
Москва, 2024 г.

Цель лабораторной работы

Изучить механизм использования представлений и предоставления прав в PostgreSQL, получить практически навыки создания представлений в среде PostgreSQL.

1. Создать любое простое представление и запросить с помощью него данные.

```
CREATE VIEW ProductsNotSold AS  
SELECT p.product_id, p.product_name  
FROM products p  
WHERE NOT EXISTS (  
    SELECT 1  
    FROM items oi  
    WHERE p.product_id = oi.product_id  
)  
ORDER BY p.product_id ASC;
```



The screenshot shows a PostgreSQL IDE interface. At the top, a SQL query is entered in a text editor. Below the editor, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with the results of the query. The table has two columns: 'product_id' (integer) and 'product_name' (character varying (255)). The results are ordered by product_id in ascending order.

	product_id integer	product_name character varying (255)
1	6	Product 6
2	7	Product 7
3	8	Product 8
4	9	Product 6
5	10	Product 7
6	11	Product 8
7	12	Product 6
8	13	Product 7
9	14	Product 8
10	15	Product 9
11	16	Product 10

2. Проверить соответствие данных прямым запросом.

```
49 SELECT p.product_id, p.product_name
50 FROM products p
51 WHERE NOT EXISTS (
52     SELECT 1
53     FROM items oi
54     WHERE p.product_id = oi.product_id
55 )
56 ORDER BY p.product_id ASC;
57
```

Data Output Messages Notifications

	product_id [PK] integer	product_name character varying (255)
1	6	Product 6
2	7	Product 7
3	8	Product 8
4	9	Product 6
5	10	Product 7
6	11	Product 8
7	12	Product 6
8	13	Product 7
9	14	Product 8
10	15	Product 9
11	16	Product 10

3. Изменить созданное представление с помощью команды ALTER VIEW, добавив псевдонимы полям.

```
57
58 alter view ProductsNotSold rename column product_id to id;
```

Data Output Messages Notifications

ALTER VIEW

Query returned successfully in 38 msec.

4. Изменить запрос созданного представления с помощью команды CREATE OR REPLACE VIEW. Создать представление с опцией WITH CHECK OPTION.

```
1 create or replace view dealdate as
2 select deal_id, product_id, deal_price, deal_quantity, deal_date from deal
3 where deal_date < '19/05/2022' order by deal_id desc with cascaded check option;
4
```

Data Output Сообщения Notifications

CREATE VIEW

Запрос завершён успешно, время выполнения: 68 msec.

6. Удалить представление.

```
76
77 drop view ProductsNotSold
```

Data Output Messages Notifications

DROP VIEW

Query returned successfully in 163 msec.

8. Создать роль Test_creator без права входа в систему, но с правом создания БД и ролей.

```
1 create role test_creator with nologin createdb createrole
2
```

Data Output Messages Notifications

CREATE ROLE

Query returned successfully in 48 msec.

9. Создать пользователя user1 с правом входа в систему. Убедиться, что user1 не может создать БД.

```
1 create role user1 with createdb login password '1'
```

Data Output Messages Notifications

CREATE ROLE

Query returned successfully in 47 msec.

10. Включить пользователя user1 в группу Test_creator.

```
3 create database try;
```

Data Output Messages Notifications

CREATE DATABASE

Query returned successfully in 109 msec.

```
5 alter group test_creator add user user1
```

Data Output Messages Notifications

ALTER ROLE

Query returned successfully in 62 msec.

12. Создать роли без права создания таблицы и с правом создания таблицы, последовательно проверить работу ролей.

Создадим роль без права создания таблиц:

```
11  
12 CREATE ROLE role_without_create;  
13
```

Data Output Messages Notifications

CREATE ROLE

Query returned successfully in 81 msec.

Создадим роль с правом создания таблиц:

```
13  
14 CREATE ROLE role_with_create WITH CREATEDB;  
15
```

Data Output Messages Notifications

CREATE ROLE

Query returned successfully in 95 msec.

CREATE TABLE test_table_role1 (id SERIAL PRIMARY KEY, name TEXT);

```
16 CREATE TABLE test_table_role1 (id SERIAL PRIMARY KEY, name TEXT);  
17  
18
```

Data Output Messages Notifications

ERROR: relation "test_table_role1" already exists

SQL state: 42P07

13. Добавить к роли право на любые действия с таблицей, проверить работу прав.

Добавление прав к ролям:

```
18 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO role_without_create;  
19
```

Data Output Messages Notifications

GRANT

Query returned successfully in 43 msec.

```
20 CREATE TABLE test_table_role1_1 (id SERIAL PRIMARY KEY, name TEXT);  
21
```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 67 msec.

проверить работу прав.

Data Output

Messages

Notifications

id

[PK] integer

name

text

```
25 REVOKE INSERT ON ALL TABLES IN SCHEMA public FROM role_without_create;
26 |
```

Data Output Messages Notifications

REVOKE

Query returned successfully in 108 msec.

Data Output Messages Notifications

ERROR: relation "role_without_create" does not exist
LINE 27: INSERT INTO role_without_create (column1) VALUES ('value1');
 ^

SQL state: 42P01
Character: 681

Изучен механизм использования представлений и предоставления прав в PostgreSQL, получены практические навыки создания представления в PostgreSQL.