

<b>Предмет</b>	МДК07.01. Управление и автоматизация баз данных
<b>Курс</b>	4
<b>Семестр</b>	2
<b>Работа</b>	29
<b>Группа</b>	494
<b>Фамилия</b>	Зубкова
<b>Имя</b>	Валерия
<b>Отчество</b>	Геннадьевна

## Отчет

### Создание базы данных

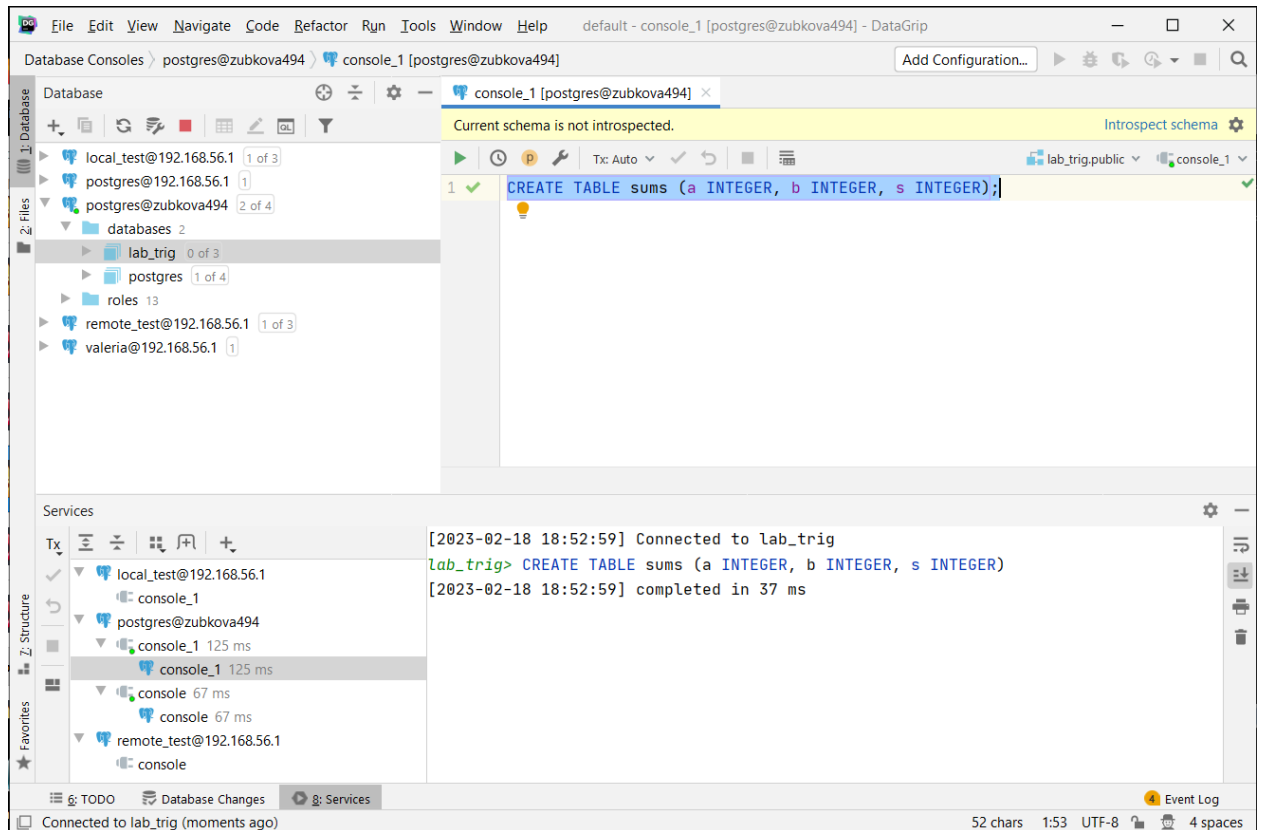
```

192.168.56.1 - PuTTY
zubkova494-m3:~# docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
b9bc3382eab1   postgres:alpine "docker-entrypoint.s... 6 hours ago   Up 6 hou
rs      0.0.0.0:5432->5432/tcp, :::5432->5432/tcp postgres_container
04818da237de   dpape/pgadmin4  "/entrypoint.sh"        6 hours ago   Up 6 hou
rs      443/tcp, 0.0.0.0:5050->80/tcp, :::5050->80/tcp pgadmin_container
zubkova494-m3:~# docker exec -it b9bc3382eab1 sh
/ # psql -U postgres
psql (15.1)
Type "help" for help.

postgres=#
postgres=# \c lab_trig
You are now connected to database "lab_trig" as user "postgres".
lab_trig=#

```

## Создание таблицы



## Просмотр таблицы после создания

```
lab_trig-# \dt
          List of relations
Schema | Name | Type  | Owner
-----+-----+-----+-----
public | sums | table | postgres
(1 row)
```

## Просмотр таблицы после создания

```
lab_trig-# \d sums
          Table "public.sums"
Column | Type   | Collation | Nullable | Default
-----+-----+-----+-----+-----
a      | integer |           |          |
b      | integer |           |          |
s      | integer |           |          |
```

## Заполнение таблицы

The screenshot shows the DataGrip interface with the following components:

- Database Consoles:** The console for 'console\_1 [postgres@zubkova494]' shows the execution of the following SQL commands:

```
lab_trig> CREATE TABLE sums (a INTEGER, b INTEGER, s INTEGER)
[2023-02-18 18:52:59] completed in 37 ms
lab_trig.public> INSERT INTO sums (a, b) VALUES (1,2), (3,4)
[2023-02-18 18:56:43] 2 rows affected in 7 ms
```
- Database Structure:** The left sidebar shows the database structure for 'lab\_trig', including 'postgres', 'roles', and 'remote\_test@192.168.56.1'.
- Services:** The bottom panel shows the 'Tx' (Transaction) list, including 'console\_1' and 'remote\_test@192.168.56.1'.
- Event Log:** The bottom right corner shows the event log with the message 'Connected to lab\_trig (4 minutes ago)'.

## Просмотр таблицы после заполнения

The screenshot shows the DataGrip interface with the following components:

- Database Consoles:** The console for 'console\_1 [postgres@zubkova494]' shows the execution of the SQL command:

```
SELECT * FROM sums;
```
- Database Structure:** The left sidebar shows the database structure for 'lab\_trig', including 'postgres', 'roles', and 'remote\_test@192.168.56.1'.
- Services:** The bottom panel shows the 'Tx' (Transaction) list, including 'console\_1' and 'remote\_test@192.168.56.1'.
- Output:** The bottom right panel shows the output of the query, displaying 2 rows of data in a table format:

	a	b	s
1	1	2	<null>
2	3	4	<null>

The status bar at the bottom indicates '2 rows retrieved starting from 1 in 242 ms (execution: 8 ms, fetching: 234 ms)'.

## Обновление таблицы

The screenshot shows the DataGrip interface with a PostgreSQL database console. The current schema is not introspected. The SQL editor contains the following queries:

```
1 UPDATE sums SET s = a + b;  
2 SELECT * FROM sums;
```

The output window shows the results of the SELECT query:

a	b	s
1	1	2
2	3	4

The status bar indicates that 2 rows were retrieved starting from 1 in 84 ms (execution: 5 ms, fetching: 79 ms).

## Добавление новых данных в таблицу

The screenshot shows the DataGrip interface with a PostgreSQL database console. The current schema is not introspected. The SQL editor contains the following query:

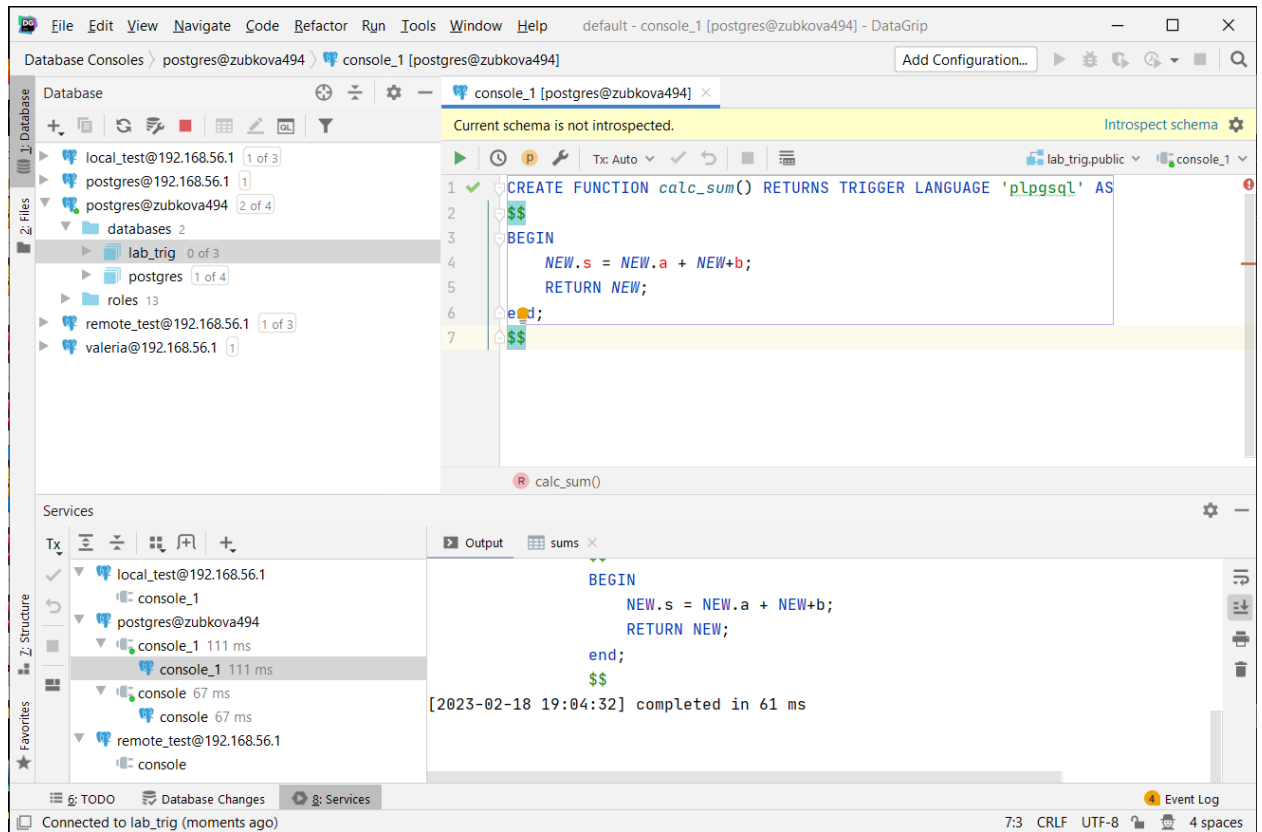
```
1 INSERT INTO sums (a,b) VALUES (5,6);
```

The output window shows the results of the INSERT query:

a	b	s
1	1	2
2	3	4
3	5	6

The status bar indicates that 3 rows were retrieved starting from 1 in 70 ms (execution: 56 ms, fetching: 14 ms).

## Создание триггера на складывание значений



## Проверка созданного триггера

```
lab_trig=# \df
              List of functions
Schema |   Name   | Result data type | Argument data types | Type
-----+-----+-----+-----+-----
public | calc_sum | trigger          |                     | func
(1 row)
```

## Создание триггера tg\_sum

The screenshot shows the DataGrip interface with a PostgreSQL database console. The left sidebar displays the database structure, including a schema named 'lab\_trig'. The main editor area contains the following SQL code:

```
1 CREATE TRIGGER tg_sum BEFORE INSERT ON sums FOR EACH ROW
2 EXECUTE PROCEDURE calc_sum();
```

The 'Services' panel at the bottom shows the execution history. The first entry shows the trigger creation command being executed successfully, completing in 61 ms. The second entry shows the trigger being executed, completing in 13 ms.

## Проверка таблицы после записи новых данных и срабатывания триггера

The screenshot shows the DataGrip interface with a PostgreSQL database console. The main editor area contains the following SQL code:

```
1 INSERT INTO sums (a, b) VALUES (7, 8);
```

The 'Services' panel at the bottom shows the execution history. The first entry shows the insert statement being executed successfully, completing in 64 ms. The second entry shows the trigger being executed, completing in 67 ms.

The 'Output' panel displays the results of the insert statement, showing a table with 4 rows and 3 columns (a, b, s):

	a	b	s
1	1	2	3
2	3	4	7
3	5	6	<null>
4	7	8	15

## Отключение триггера

The screenshot shows the DataGrip interface with the following components:

- Database Consoles:** postgres@zubkova494, console\_1 [postgres@zubkova494]
- Database Structure:** lab\_trig (0 of 3) containing schemas, access methods (7), extensions (1), languages (4), and postgres (1 of 4).
- Services:** console\_1 (82 ms)
- Output:** sums x

The SQL editor shows the command: `ALTER TABLE sums DISABLE TRIGGER tg_sum;`

The output log shows the following sequence of events:

- [2023-02-18 19:11:59] 3 rows retrieved starting from 1 in 20 ms (execution: 8 ms, fetching: 12 ms)
- lab\_trig.public> INSERT INTO sums (a, b) VALUES (7,8)
- [2023-02-18 19:15:47] 1 row affected in 7 ms
- lab\_trig.public> SELECT \* FROM sums
- [2023-02-18 19:15:50] 4 rows retrieved starting from 1 in 22 ms (execution: 8 ms, fetching: 14 ms)
- lab\_trig.public> ALTER TABLE sums DISABLE TRIGGER tg\_sum
- [2023-02-18 19:17:21] completed in 48 ms

## Проверка ввода данных

The screenshot shows the DataGrip interface with the following components:

- Database Consoles:** postgres@zubkova494, console\_1 [postgres@zubkova494]
- Database Structure:** lab\_trig (0 of 3) containing schemas, access methods (7), extensions (1), languages (4), and postgres (1 of 4).
- Services:** console\_1 (76 ms)
- Output:** sums x

The SQL editor shows the command: `INSERT INTO sums (a, b) VALUES (7,8);`

The output log shows the following sequence of events:

- [2023-02-18 19:11:59] 3 rows retrieved starting from 1 in 20 ms (execution: 8 ms, fetching: 12 ms)
- lab\_trig.public> INSERT INTO sums (a, b) VALUES (7,8)
- [2023-02-18 19:15:47] 1 row affected in 7 ms
- lab\_trig.public> SELECT \* FROM sums
- [2023-02-18 19:15:50] 4 rows retrieved starting from 1 in 22 ms (execution: 8 ms, fetching: 14 ms)
- lab\_trig.public> ALTER TABLE sums DISABLE TRIGGER tg\_sum
- [2023-02-18 19:17:21] completed in 48 ms

The resulting data in the sums table is as follows:

	a	b	s
1	1	2	3
2	3	4	7
3	5	6	<null>
4	7	8	15
5	7	8	<null>

## Включение триггера

The screenshot shows the DataGrip interface with the SQL editor containing the command: `ALTER TABLE sums ENABLE TRIGGER tg_sum;`. The left sidebar displays the database structure, including the `lab_trig` schema. The bottom right pane shows the output of the command, displaying a table with 5 rows and 3 columns: `a`, `b`, and `s`.

	a	b	s
1	1	2	3
2	3	4	7
3	5	6	<null>
4	7	8	15
5	7	8	<null>

## Проверка работы триггера

The screenshot shows the DataGrip interface with the SQL editor containing the command: `INSERT INTO sums (a, b) VALUES (10,15);`. The bottom right pane shows the output of the command, displaying a table with 6 rows and 3 columns: `a`, `b`, and `s`. The new row (6) has values `a=10`, `b=15`, and `s=25`.

	a	b	s
1	1	2	3
2	3	4	7
3	5	6	<null>
4	7	8	15
5	7	8	<null>
6	10	15	25



## Обновление данных

The screenshot shows the DataGrip interface with a PostgreSQL database. The left sidebar displays the database structure, including a table named `sums` under the `lab_trig` schema. The main editor contains the following SQL query:

```
UPDATE sums SET a = 13, b = 27 WHERE a = 3 AND b = 4;
```

The 'Output' tab shows the results of the query, which is an empty table with 6 rows and 3 columns: `a`, `b`, and `s`.

	a	b	s
1	1	2	3
2	5	6	<null>
3	7	8	15
4	7	8	<null>
5	10	15	25
6	13	27	7

## Удаление триггера

The screenshot shows the DataGrip interface with a PostgreSQL database. The left sidebar displays the database structure, including a table named `sums` under the `lab_trig` schema. The main editor contains the following SQL query:

```
DROP TRIGGER tg_sum ON sums;
```

The 'Output' tab shows the results of the query, which is an empty table with 6 rows and 3 columns: `a`, `b`, and `s`.

	a	b	s
1	1	2	3
2	5	6	<null>
3	7	8	15
4	7	8	<null>
5	10	15	25
6	13	27	7

## Запись данных после удаления

The screenshot shows the DataGrip interface with a PostgreSQL console. The console has executed the following SQL statement:

```
INSERT INTO sums (a, b) VALUES (10,40);
```

The 'Output' tab displays the results of the query in a table format:

	a	b	s
1	1	2	3
2	5	6	<null>
3	7	8	15
4	7	8	<null>
5	10	15	25
6	13	27	7
7	10	40	<null>

The interface also shows a database tree on the left with the 'lab\_trig' database selected. The status bar at the bottom indicates 'Connected to lab\_trig (a minute ago)' and '39 chars 1:40 UTF-8 4 spaces'.

## Создание нового триггера

The screenshot shows the DataGrip interface with a PostgreSQL console. The console has executed the following SQL statement:

```
CREATE TRIGGER tg_sum BEFORE INSERT OR UPDATE ON sums FOR EACH ROW EXECUTE PROCEDURE calc_sum();
```

The 'Output' tab displays the results of the query, showing the execution of the trigger and the insertion of data into the 'sums' table:

```
[2023-02-18 19:24:00] 6 rows retrieved starting from 1 in 22 ms (execution: 8 ms, fetching: 14 ms)
lab_trig.public> DROP TRIGGER tg_sum ON sums
[2023-02-18 19:25:07] completed in 10 ms
lab_trig.public> INSERT INTO sums (a, b) VALUES (10,40)
[2023-02-18 19:26:14] 1 row affected in 9 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:26:15] 7 rows retrieved starting from 1 in 19 ms (execution: 7 ms, fetching: 12 ms)
lab_trig.public> CREATE TRIGGER tg_sum BEFORE INSERT OR UPDATE ON sums FOR EACH ROW EXECUTE PROCEDURE calc_sum()
[2023-02-18 19:27:54] completed in 9 ms
```

The interface also shows a database tree on the left with the 'lab\_trig' database selected. The status bar at the bottom indicates 'Connected to lab\_trig (moments ago)' and '77:1 UTF-8 4 spaces'.

## Обновление данных с помощью запроса и триггера

The screenshot shows the DataGrip interface with a PostgreSQL console. The console is connected to a database named 'postgres@zubkova494'. The current schema is 'lab\_trig'. The console shows the following SQL query:

```
UPDATE sums SET a = 5, b = 6 WHERE a = 1 AND b = 2
```

The results of the query are displayed in a table with 7 rows:

	a	b	s
1	5	6	<null>
2	7	8	15
3	7	8	<null>
4	10	15	25
5	13	27	7
6	10	40	<null>
7	5	6	11

The bottom status bar indicates that 7 rows were retrieved starting from 1 in 21 ms (execution: 7 ms, fetching: 14 ms).

## Добавление колонки к таблице

The screenshot shows the DataGrip interface with a PostgreSQL console. The console is connected to a database named 'postgres@zubkova494'. The current schema is 'lab\_trig'. The console shows the following SQL query:

```
ALTER TABLE sums ADD COLUMN u INTEGER;
```

The results of the query are displayed in a table with 7 rows:

	a	b	s	u
1	5	6	<null>	
2	7	8	15	
3	7	8	<null>	
4	10	15	25	
5	13	27	7	
6	10	40	<null>	
7	5	6	11	

The bottom status bar indicates that 7 rows were retrieved starting from 1 in 21 ms (execution: 7 ms, fetching: 14 ms).

## Обновление данных для новой колонки

The screenshot shows the DataGrip interface with a PostgreSQL database. The left sidebar displays the database structure, including the 'lab\_trig' schema. The main editor shows the following SQL query:

```
UPDATE sums SET u = 0 WHERE s IS NULL
```

The 'Output' tab displays the results of the query, showing 7 rows affected. The table structure is as follows:

	a	b	s	u
1	5	6	<null>	<null>
2	7	8	15	<null>
3	7	8	<null>	<null>
4	10	15	25	<null>
5	13	27	7	<null>
6	10	40	<null>	<null>
7	5	6	11	<null>

The status bar at the bottom indicates '3 rows affected in 9 ms'.

## Отключение триггера

The screenshot shows the DataGrip interface with a PostgreSQL database. The left sidebar displays the database structure, including the 'lab\_trig' schema. The main editor shows the following SQL query:

```
DROP TRIGGER tg_sum ON sums;
```

The 'Output' tab displays the execution log, showing the following commands and their results:

```
[2023-02-18 19:33:12] completed in 9 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:33:55] 7 rows retrieved starting from 1 in 33 ms (execution: 10 ms, fetching: 23 ms)
lab_trig.public> UPDATE sums SET u = 0 WHERE s IS NULL
[2023-02-18 19:34:45] 3 rows affected in 9 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:35:14] 7 rows retrieved starting from 1 in 26 ms (execution: 8 ms, fetching: 18 ms)
lab_trig.public> DROP TRIGGER tg_sum ON sums
[2023-02-18 19:36:17] completed in 8 ms
```

The status bar at the bottom indicates 'Connected to lab\_trig (moments ago)'.

## Создание нового триггера

The screenshot shows the DataGrip interface with a PostgreSQL database. The left sidebar displays the database structure, including a schema named 'lab\_trig'. The main editor shows the SQL command to create a trigger:

```
CREATE TRIGGER tg_sum BEFORE INSERT OR UPDATE ON sums FOR EACH ROW WHEN ( NEW.a > 100) EXECUTE PROCEDURE calc_sum()
```

The 'Services' panel at the bottom shows the execution of several SQL commands:

```
lab_trig.public> UPDATE sums SET u = 0 WHERE s IS NULL
[2023-02-18 19:34:45] 3 rows affected in 9 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:35:14] 7 rows retrieved starting from 1 in 26 ms (execution: 8 ms, fetching: 18 ms)
lab_trig.public> DROP TRIGGER tg_sum ON sums
[2023-02-18 19:36:17] completed in 8 ms
lab_trig.public> CREATE TRIGGER tg_sum BEFORE INSERT OR UPDATE ON sums FOR EACH ROW WHEN ( NEW.a > 100) EXECUTE PROCEDURE ca
[2023-02-18 19:38:21] completed in 47 ms
```

## Добавление значений в таблицу

The screenshot shows the DataGrip interface with the same PostgreSQL database. The main editor shows the SQL command to insert data into the 'sums' table:

```
INSERT INTO sums (a,b) VALUES (17,18)
```

The 'Services' panel at the bottom shows the execution of the command. Below it, a table view displays the data in the 'sums' table:

	a	b	s	u
1	7	8	15	<null>
2	10	15	25	<null>
3	13	27	7	<null>
4	5	6	11	<null>
5	5	6	11	0
6	7	8	15	0
7	10	40	50	0
8	17	18	<null>	<null>

## Проверка сохраненных данных после добавления более крупного числа (триггер сработал)

The screenshot shows the DataGrip interface with a PostgreSQL database. The SQL editor contains the following query:

```
INSERT INTO sums (a,b) VALUES (170,180)
```

The output pane displays the results of the query, showing 9 rows of data. The columns are labeled 'a', 'b', 's', and 'u'. The data is as follows:

a	b	s	u
1	10	25	<null>
2	13	27	<null>
3	5	6	<null>
4	5	6	11
5	5	6	11
6	7	8	15
7	10	40	50
8	17	18	<null>
9	170	180	350

## Создание новой таблицы для пользователей

The screenshot shows the DataGrip interface with a PostgreSQL database. The SQL editor contains the following query:

```
CREATE TABLE acss (usr VARCHAR(10), pass VARCHAR(50))
```

The output pane displays the results of the query, showing the execution of the query and subsequent SELECT statements. The output is as follows:

```
[2023-02-18 19:39:30] 1 row affected in 46 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:39:32] 8 rows retrieved starting from 1 in 18 ms (execution: 7 ms, fetching: 11 ms)
lab_trig.public> INSERT INTO sums (a,b) VALUES (170,180)
[2023-02-18 19:40:15] 1 row affected in 8 ms
lab_trig.public> SELECT * FROM sums
[2023-02-18 19:40:18] 9 rows retrieved starting from 1 in 24 ms (execution: 10 ms, fetching: 14 ms)
lab_trig.public> CREATE TABLE acss (usr VARCHAR(10), pass VARCHAR(50))
[2023-02-18 19:42:58] completed in 51 ms
```

## Заполнение таблицы данными

The screenshot shows the DataGrip interface with the following details:

- Database:** postgres@zubkova494
- Schema:** lab\_trig (Current schema is not introspected.)
- SQL Editor:**

```
1 INSERT INTO acss (usr, pass) VALUES ('test', '123');
2 SELECT * FROM acss;
```
- Output:** A table with 1 row and 2 columns: 

usr	pass
test	123
- Services:** console\_1 (181 ms)
- Status:** 1 row retrieved starting from 1 in 140 ms (execution: 7 ms, fetching: 133 ms)

## Создание триггера для новой таблицы

The screenshot shows the DataGrip interface with the following details:

- Database:** postgres@zubkova494
- Schema:** lab\_trig (Current schema is not introspected.)
- SQL Editor:**

```
1 CREATE FUNCTION calc_hash() RETURNS TRIGGER LANGUAGE 'plpgsql' AS
2 $$
3 BEGIN
4     NEW.pass = md5(NEW.pass);
5     RETURN NEW;
6 END
7 $$
```
- Output:** A table with 1 row and 2 columns: 

usr	pass
test	123
- Services:** console\_1 (39 ms)
- Status:** Connected to lab\_trig (moments ago)

## Добавление новых данных в таблицу

The screenshot shows the DataGrip interface with a PostgreSQL database 'postgres@zubkova494'. The 'console\_1' schema is selected. The SQL editor contains the following statement:

```
INSERT INTO acss (usr, pass) VALUES ('test', 'pass');
```

The 'Output' tab shows the results of the query, displaying two rows:

	usr	pass
1	test	123
2	test	pass

The status bar at the bottom indicates '2 rows retrieved starting from 1 in 17 ms (execution: 6 ms, fetching: 11 ms)'.

## Добавление триггера

The screenshot shows the DataGrip interface with the same PostgreSQL database. The 'console\_1' schema is selected. The SQL editor contains the following statement:

```
CREATE TRIGGER tg_hash BEFORE INSERT ON acss FOR EACH ROW EXECUTE PROCEDURE calc_hash();
```

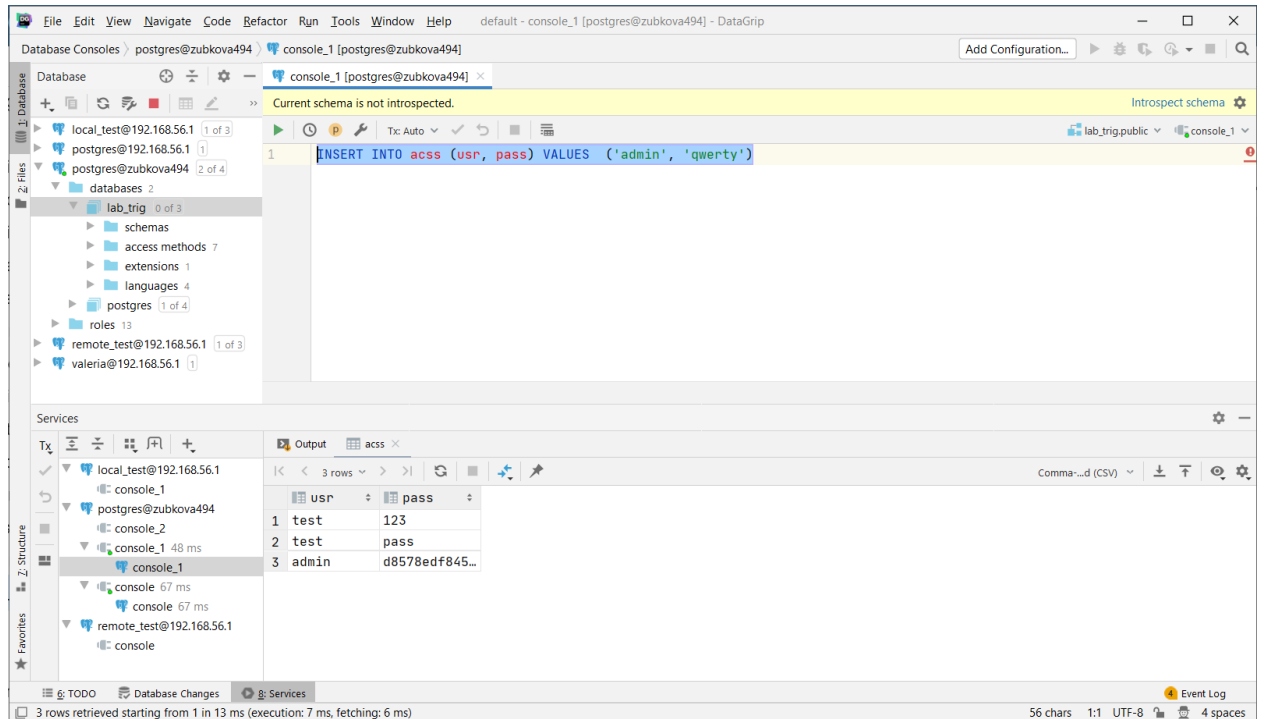
The 'Output' tab shows the results of the query, displaying two rows:

	usr	pass
1	test	123
2	test	pass

The status bar at the bottom indicates 'Connected to lab\_trig (moments ago)'.



## Проверка сохранения новых данных и срабатывание триггера для пароля



The screenshot shows the DataGrip interface with a PostgreSQL database named 'zubkova494'. The 'console\_1' schema is selected. The SQL editor contains the following statement:

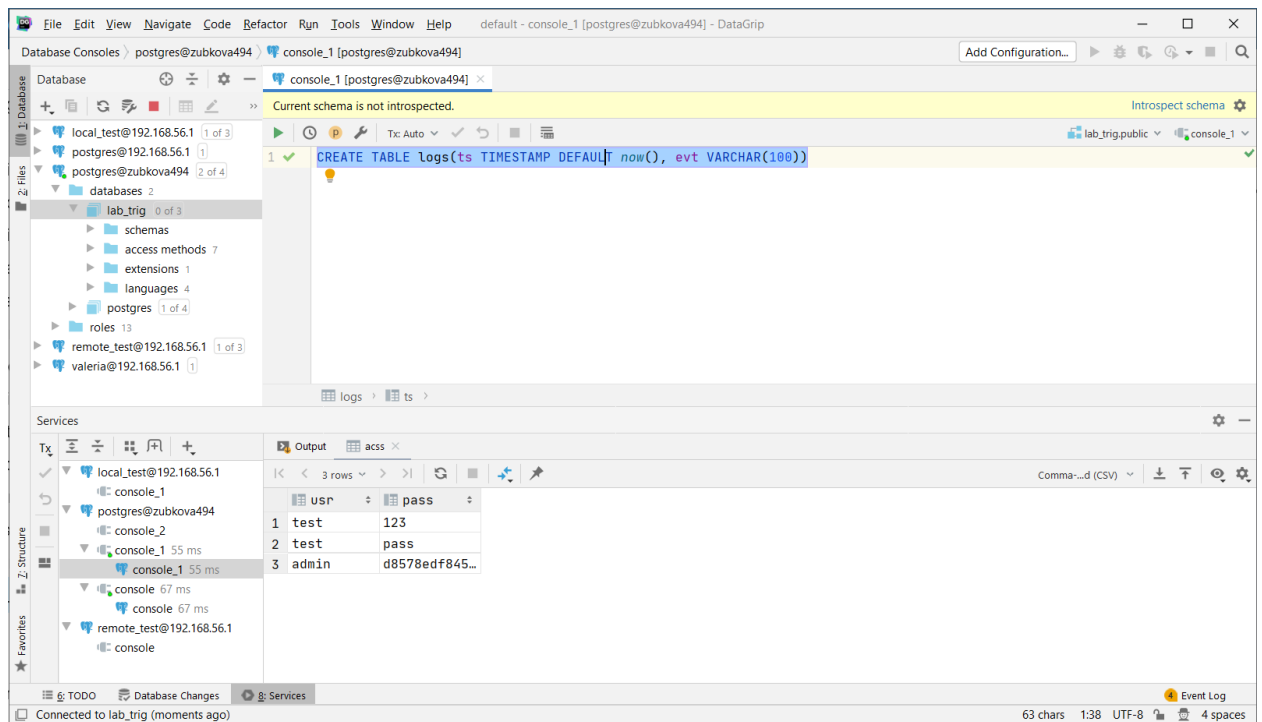
```
INSERT INTO acss (usr, pass) VALUES ('admin', 'qwerty')
```

The 'Output' tab shows the results of the query, displaying 3 rows:

usr	pass
1 test	123
2 test	pass
3 admin	d8578edf845...

The status bar at the bottom indicates '3 rows retrieved starting from 1 in 13 ms (execution: 7 ms, fetching: 6 ms)'.

## Создание новой таблицы



The screenshot shows the DataGrip interface with the same PostgreSQL database. The 'console\_1' schema is selected. The SQL editor contains the following statement:

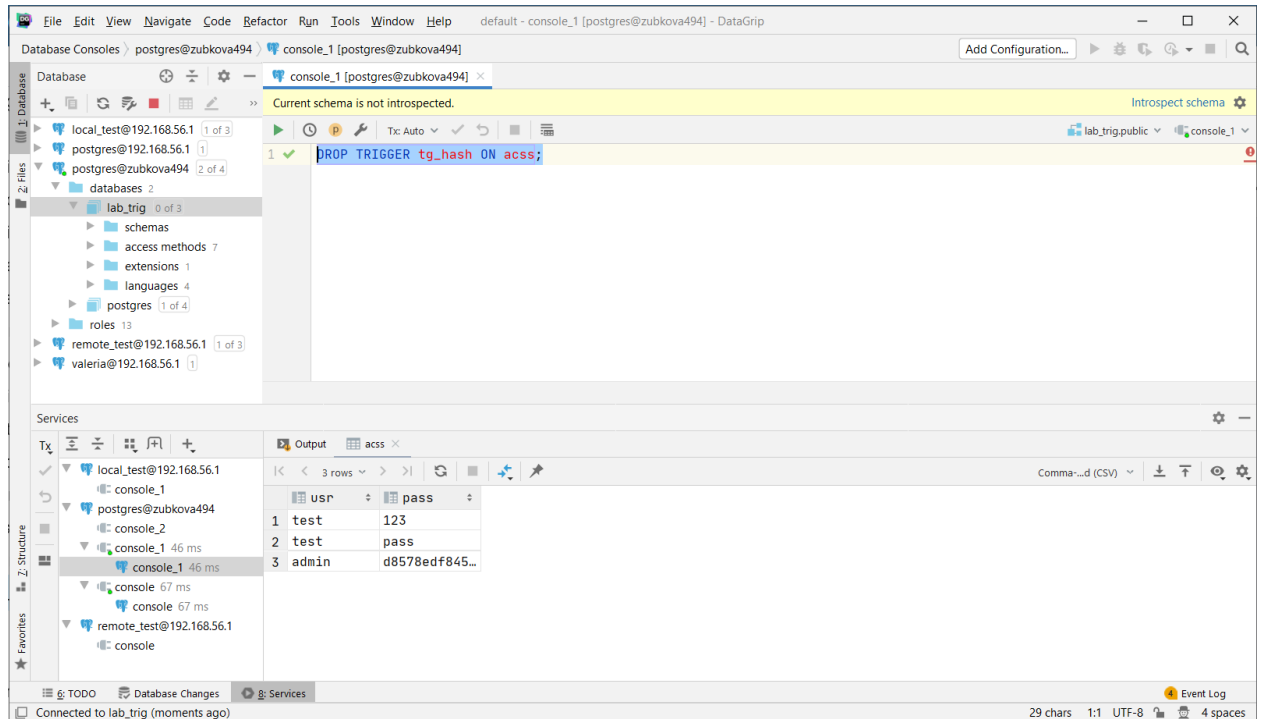
```
CREATE TABLE logs(ts TIMESTAMP DEFAULT now(), evt VARCHAR(100))
```

The 'Output' tab shows the results of the query, displaying 3 rows:

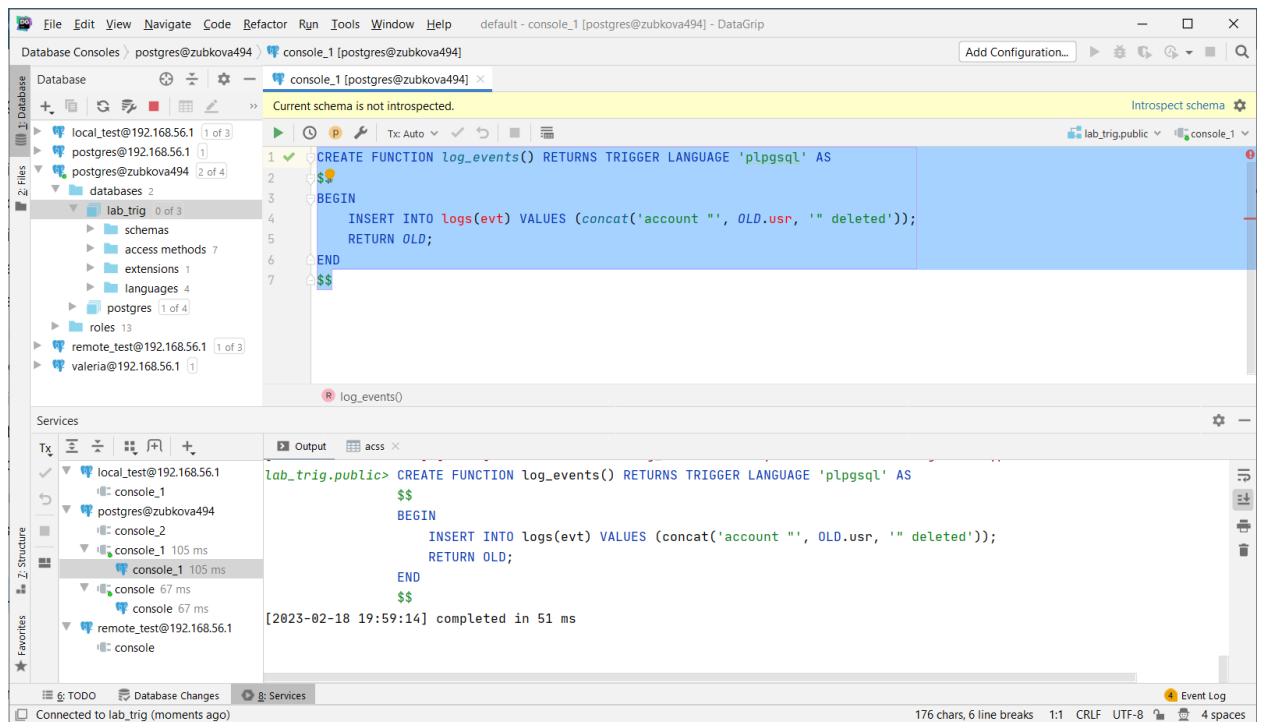
usr	pass
1 test	123
2 test	pass
3 admin	d8578edf845...

The status bar at the bottom indicates 'Connected to lab\_trig (moments ago)'.

## Убираем триггер с прошлой таблицы



## Создание триггера log\_events()



## Создание триггера для функции log\_events

The screenshot shows the DataGrip interface with a PostgreSQL database console. The SQL editor contains the following code:

```
CREATE TRIGGER tg_log BEFORE DELETE ON acss FOR EACH ROW EXECUTE PROCEDURE log_events();
```

The output pane shows the execution results:

```
[2023-02-18 19:59:14] completed in 51 ms
lab_trig.public> CREATE TRIGGER tg_log BEFORE DELETE ON acss FOR EACH ROW EXECUTE PROCEDURE log_events()
[2023-02-18 20:00:43] completed in 7 ms
```

The left sidebar shows the database structure, including the 'lab\_trig' schema and the 'acss' table.

## Удаление строк из таблицы с пользователями

The screenshot shows the DataGrip interface with a PostgreSQL database console. The SQL editor contains the following code:

```
DELETE FROM acss WHERE usr='test';
```

The output pane shows the execution results:

```
1 row
```

The left sidebar shows the database structure, including the 'lab\_trig' schema and the 'acss' table.

usr	pass
admin	d8578edf845...

# Проверка работы логирования

Database Consoles

postgres@zubkova494

console\_1 [postgres@zubkova494]

Add Configuration...

Database

Current schema is not introspected.

Introspect schema

local\_test@192.168.56.1

postgres@192.168.56.1

postgres@zubkova494

databases 2

lab\_trig 0 of 3

schemas

access methods 7

extensions 1

languages 4

postgres 1 of 4

roles 13

remote\_test@192.168.56.1

valeria@192.168.56.1

Services

Output logs

Comma-separated (CSV)

2 rows

ts

evt

1 2022-12-18 22:17:51.082270 account "test" deleted

2 2022-12-18 22:17:51.082270 account "test" deleted

local\_test@192.168.56.1

postgres@zubkova494

console\_1

console\_2

console\_1 215 ms

console\_1 215 ms

console 67 ms

console 67 ms

remote\_test@192.168.56.1

console

TODO

Database Changes

Services

Event Log

2 rows retrieved starting from 1 in 180 ms (execution: 7 ms, fetching: 173 ms)

19 chars 1:1 UTF-8 4 spaces