

## **Вариант 8**

### **Цель:**

Данная лабораторная работа призвана сформировать у студента понимание назначения пользовательских функций и процедур, их написание и использование.

### **Задачи:**

- Получить теоретические знания о назначении функций и процедур БД.
- Изучить синтаксис функций и процедур.
- Научиться добавлять функции и процедуры в БД.
- Научиться удалять и изменять функции и процедуры.
- Научиться использовать функции и процедуры.

### **Практическое задание ЛР**

Написать хранимые функции и процедуры, выполняющие следующие операции для своей предметной области.

- Добавление одной записи, только через вызов процедуры, для таблиц Проекты, Поручение, Работник.  
Проверять там, где надо, на дублирование записей с выдачей сообщения об ошибке.  
Проверять на корректность ввода данных, например, даты выдачи позже плановой даты окончания.
- Изменение одной записи, только через вызов процедуры, для таблиц Проекты, Поручение, Работник.  
Проверять там, где надо, на дублирование записей с выдачей сообщения об ошибке.
- Удаление одной записи только, через вызов процедуры, для таблиц Проекты, Поручение, Работник.  
Проверять на возможность удаления. Например, удаляем книгу, которая находится на руках у читателя (не допускать формирование исключения о нарушении целостности БД).
- Запрос читателей по первичному ключу, по адресу проживания, по выданной книге.

- Запрос списка книг, выданных заданному читателю.

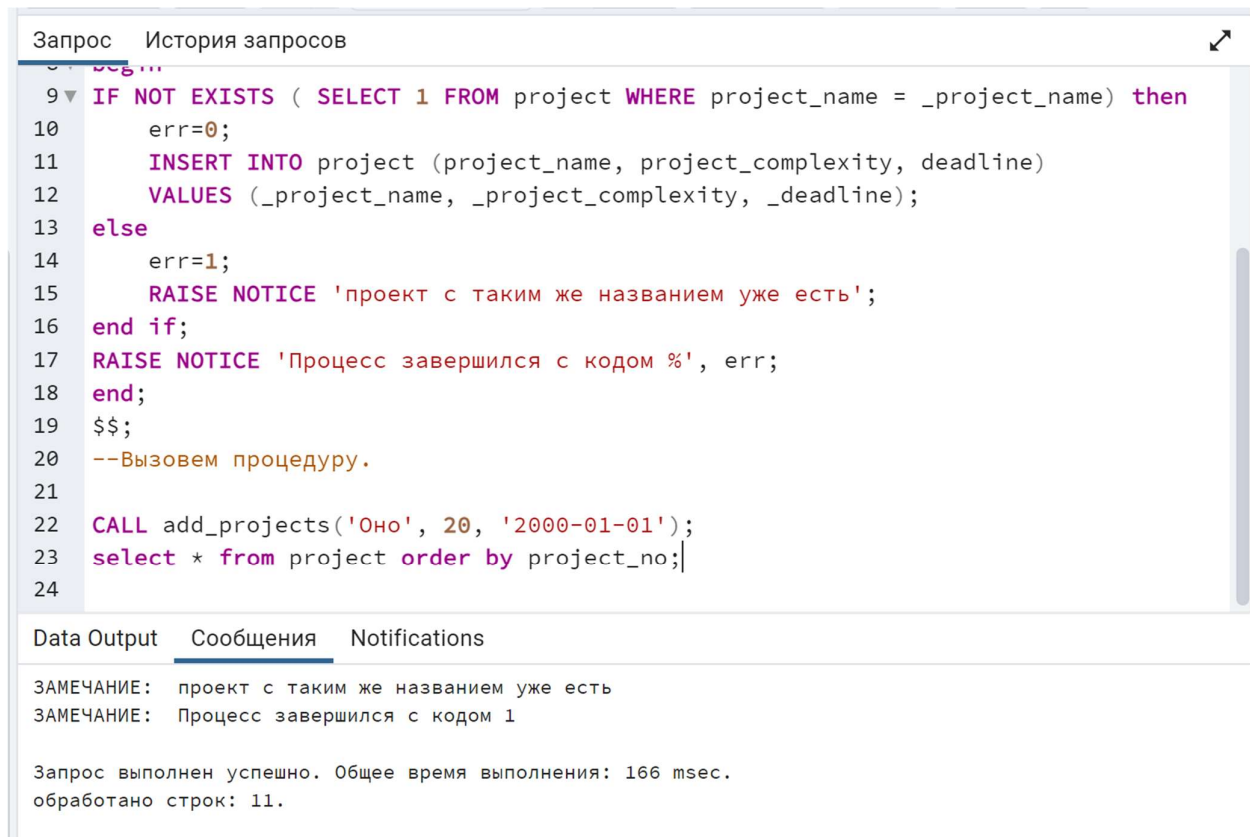
### **Задание 1**

*Добавление одной записи в таблицу Проекты:*

```
CREATE OR REPLACE PROCEDURE add_projects(_project_name char(10),
                                         _project_complexity integer,
                                         _deadline date)
LANGUAGE plpgsql
```

```
AS $$
declare
err integer;
begin
IF NOT EXISTS ( SELECT 1 FROM project WHERE project_name =
_project_name) then
    err=0;
    INSERT INTO project (project_name, project_complexity, deadline)
    VALUES (_project_name, _project_complexity, _deadline);
else
    err=1;
    RAISE NOTICE 'проект с таким же названием уже есть';
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.
```

```
CALL add_projects('Оно', 20, '2000-01-01');
select * from project order by project_no;
```



The screenshot shows a database query editor with a tab titled "Запрос" (Query) and a sub-tab "История запросов" (Query History). The main area displays a PL/SQL script with line numbers 9 through 24. The script checks for the existence of a project in a table named "project". If it doesn't exist, it inserts a new record. If it does exist, it raises a notice. The script ends with a call to a procedure "add\_projects" and a select statement. Below the script, there are three tabs: "Data Output", "Сообщения" (Messages), and "Notifications". The "Сообщения" tab is active, showing two messages: "ЗАМЕЧАНИЕ: проект с таким же названием уже есть" (WARNING: project with the same name already exists) and "ЗАМЕЧАНИЕ: Процесс завершился с кодом 1" (WARNING: Process finished with code 1). At the bottom, a status message indicates the query was executed successfully in 166 ms, processing 11 rows.

```
9 IF NOT EXISTS ( SELECT 1 FROM project WHERE project_name = _project_name) then
10     err=0;
11     INSERT INTO project (project_name, project_complexity, deadline)
12     VALUES (_project_name, _project_complexity, _deadline);
13 else
14     err=1;
15     RAISE NOTICE 'проект с таким же названием уже есть';
16 end if;
17 RAISE NOTICE 'Процесс завершился с кодом %', err;
18 end;
19 $$;
20 --Вызовем процедуру.
21
22 CALL add_projects('Оно', 20, '2000-01-01');
23 select * from project order by project_no;|
24
```

Сообщения

ЗАМЕЧАНИЕ: проект с таким же названием уже есть  
ЗАМЕЧАНИЕ: Процесс завершился с кодом 1

Запрос выполнен успешно. Общее время выполнения: 166 мсек.  
обработано строк: 11.

Рисунок 1 – добавление записи через процедуру с проверкой дублирования

*Добавление одной записи в таблицу Поручения:*

```
CREATE OR REPLACE PROCEDURE add_assign(_project_no integer,
                                         _worker_id integer,
                                         _handout date,
                                         _planned date,
                                         _real date,
                                         _complexity double precision)
LANGUAGE plpgsql
```

```
AS $$
declare
err integer;
begin
IF _handout>_planned then
    err=1;
    RAISE NOTICE 'дата выдачи больше плановой даты окончания';
elseif _planned>_real then
    err=2;
    raise notice 'дата плановой больше даты реальной сдачи';
else
    err=0;
    INSERT INTO assignment(project_no, worker_id, handout_date,
planned_end_date, real_end_date, assignment_complexity)
```

```
VALUES (_project_no, _worker_id, _handout, _planned, _real, _complexity);
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.
```

```
CALL add_assign(3,3, '2004-11-11', '2011-11-11', '2019-03-03', 65);
select * from assignment order by assignment_id desc;
```

The screenshot shows a database IDE with a SQL editor and a results pane. The SQL editor contains the following code:

```
15  elsif _planned > _real then
16      err=2;
17      raise notice 'дата плановой больше даты реальной сдачи';
18  else
19      err=0;
20      INSERT INTO assignment
21      VALUES (_as_id, _project_no, _worker_id, _handout, _planned, _real, _complexity)
22  end if;
23  RAISE NOTICE 'Процесс завершился с кодом %', err;
24  end;
25  $$;
26  --Вызовем процедуру.
27
28  CALL add_assign(3,3, '2020-11-11', '2011-11-11', '2019-03-03', 65);
29  select * from assignment order by assignment_id desc;
30
```

The results pane shows the following output:

Заметание: дата выдачи больше планновой даты окончания  
Заметание: Процесс завершился с кодом 1

Запрос выполнен успешно. Общее время выполнения: 121 мсек.  
обработано строк: 1000.

Рисунок 2 – добавление записи через процедуру с проверкой на корректность ввода данных

*Добавление одной записи в таблицу Работник:*

```
CREATE OR REPLACE PROCEDURE add_worker(_name text,
                                         _position text)
LANGUAGE plpgsql

AS $$
begin
    INSERT INTO worker(worker_name, position)
    VALUES (_name, _position);
end;
$$;
```

--Вызовем процедуру.

```
CALL add_worker('Волков Илья Борисович', 'продюсер');  
select * from worker order by worker_id;
```

```
1 CREATE OR REPLACE PROCEDURE add_worker(_name text,  
2                                     _position text)  
3     LANGUAGE plpgsql  
4 AS $$  
5 begin  
6     INSERT INTO worker(worker_name, position)  
7     VALUES (_name, _position);  
8 end;  
9 $$;  
10 --Вызовем процедуру.  
11  
12 CALL add_worker('Волков Илья Борисович', 'продюсер');  
13 select * from worker order by worker_id;  
14
```

Data Output   Сообщения   Notifications

	worker_id [PK] integer	worker_name text	position text
5	6	Кузнецов Алексей Данилович	декоратор
7	7	Квашонкина Ксения Андреевна	композитор
3	8	Айрапетова Елена Дмитриевна	звукорежиссёр
9	9	Светлаков Степан Егорович	актёр
10	10	Стрюкова Ольга Сергеевна	актриса
11	11	Волков Илья Борисович	продюсер

Рисунок 3 – добавление записи

## Задание 2

*Изменение записи в таблице Проекты:*

```
CREATE OR REPLACE PROCEDURE add_projects(_project_no integer,  
                                         _project_name  
                                         char(10),  
                                         _project_complexity integer,  
                                         _deadline date)  
    LANGUAGE plpgsql
```

```
AS $$  
declare
```

```

err integer;
begin
IF not exists(select 1 from project where project_no=_project_no) then
    err=1;
    RAISE NOTICE 'проекта с таким идентификатором не существует';
elseif
    EXISTS ( SELECT 1 FROM project WHERE project_name = _project_name)
then
    err=2;
    RAISE NOTICE 'проект с таким же названием уже есть';
else
    update project
    set project_name=_project_name where project_no=_project_no;
    err=0;
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.

```

```

CALL add_projects(10, 'Top', 20, '2000-01-01');
select * from project order by project_no;

```

```

14     EXISTS ( SELECT 1 FROM project WHERE project_name = _project_name) then
15     err=2;
16     RAISE NOTICE 'проект с таким же названием уже есть';
17 else
18     update project
19     set project_name=_project_name where project_no=_project_no;
20     err=0;
21 end if;
22 RAISE NOTICE 'Процесс завершился с кодом %', err;
23 end;
24 $$;
25 --Вызовем процедуру.
26
27 CALL add_projects(10, 'Top', 20, '2000-01-01');
28 select * from project order by project_no;
29

```

Data Output   Сообщения   Notifications

ЗАМЕЧАНИЕ: проект с таким же названием уже есть  
 ЗАМЕЧАНИЕ: Процесс завершился с кодом 2

Запрос выполнен успешно. Общее время выполнения: 80 msec.  
 обработано строк: 11.

Рисунок 4 – изменение записи с проверкой на дублирование

*Изменение записи в таблице Поручения:*

```

CREATE OR REPLACE PROCEDURE upd_assign(_as_id integer,
                                         _handout date,
                                         _planned date,
                                         _real date)

    LANGUAGE plpgsql

AS $$
declare
err integer;
begin
if not exists(select 1 from assignment where _as_id=assignment_id) then
    err=1;
    RAISE NOTICE 'поручения с таким идентификатором не существует';
elseif _handout>_planned AND then
    err=2;
    RAISE NOTICE 'дата выдачи больше планновой даты окончания';
elseif _planned>_real then
    err=3;
    RAISE NOTICE 'плановая дата больше реальной даты сдачи';
else
    err=0;
    update assignment
        set handout_date=_handout, planned_end_date=_planned, real_end_date=_real
        where _as_id=assignment_id;
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.

CALL upd_assign(1, '2020-11-11', '2022-11-11', '2021-12-12');
select * from assignment order by assignment_id;

```



Рисунок 5 – изменение записи

*Изменение записи в таблице Проекты:*

```
CREATE OR REPLACE PROCEDURE add_projects(_id integer,
                                         _name text)
LANGUAGE plpgsql
```

```
AS $$
declare
err integer;
begin
if not exists (select 1 from worker where worker_id=_id) then
    err=1;
    RAISE NOTICE 'работника с таким идентификатором не существует';
elseif EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name) then
    err=2;
    RAISE NOTICE 'человек с таким же ФИО уже существует';
else
    err=0;
    update worker
    set worker_name=_name where worker_id=_id;
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
```



\$\$;

--Вызовем процедуру.

CALL add\_projects(11, 'Кузнецов Алексей Данилович');

select \* from worker order by worker\_id;

Запрос	История запросов
11	<pre>ELSEIF EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name) then err=2; RAISE NOTICE 'человек с таким же ФИО уже существует'; else err=0; update worker set worker_name=_name where worker_id=_id; end if; RAISE NOTICE 'Процесс завершился с кодом %', err; end; \$\$; --Вызовем процедуру.</pre>
24	<pre>CALL add_projects(11, 'Кузнецов Алексей Данилович');</pre>
25	<pre>select * from worker order by worker_id;</pre>
26	

Data Output	Сообщения	Notifications
ЗАМЕЧАНИЕ: человек с таким же ФИО уже существует		
ЗАМЕЧАНИЕ: Процесс завершился с кодом 2		
Запрос выполнен успешно. Общее время выполнения: 60 msec.		
обработано строк: 11.		

Рисунок 6 – изменение записи с проверкой на дублирование

```
Запрос  История запросов
20 RAISE NOTICE 'проекта с таким идентификатором не существует';
11 elseif EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name) then
12     err=2;
13     RAISE NOTICE 'человек с таким же ФИО уже существует';
14 else
15     err=0;
16     update worker
17     set worker_name=_name where worker_id=_id;
18 end if;
19 RAISE NOTICE 'Процесс завершился с кодом %', err;
20 end;
21 $$;
22 --Вызовем процедуру.
23
24 CALL add_projects(11, 'Кузнецов Дмитрий Данилович');
25 select * from worker order by worker_id;
26
```

Data Output Сообщения Notifications

ЗАМЕЧАНИЕ: Процесс завершился с кодом 0

Запрос выполнен успешно. Общее время выполнения: 54 msec.  
обработано строк: 11.

Рисунок 7 – изменение записи с успешным завершением

### Задание 3

*Удаление записи из таблицы Проекты:*

```
CREATE OR REPLACE PROCEDURE del_projects(_name text)
LANGUAGE plpgsql

AS $$
declare
err integer;
begin
IF NOT EXISTS ( SELECT 1 FROM project WHERE project_name = _name)
then
    err=1;
    RAISE NOTICE 'проекта с таким названием не существует';
else
    err=0;
    delete from project where project_name = _name;
end if;
RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.
```

```
CALL del_projects('Оно');
select * from project order by project_no;
```

The screenshot shows a database query window with a tab labeled "Запрос" and "История запросов". The query is a PL/SQL block. The execution result shows a message in the "Сообщения" tab: "ЗАМЕЧАНИЕ: проекта с таким названием не существует".

```
1  err integer;
2
3  begin
4
5
6  begin
7  IF NOT EXISTS ( SELECT 1 FROM project WHERE project_name = _name) then
8      err=1;
9      RAISE NOTICE 'проекта с таким названием не существует';
10 else
11     err=0;
12     delete from project where project_name = _name;
13 end if;
14 RAISE NOTICE 'Процесс завершился с кодом %', err;
15 end;
16 $$;
17 --Вызовем процедуру.
18
19 CALL del_projects('Оно');
20 select * from project order by project_no;
21
```

Запрос История запросов

Data Output Сообщения Notifications

ЗАМЕЧАНИЕ: проекта с таким названием не существует  
ЗАМЕЧАНИЕ: Процесс завершился с кодом 1

Запрос выполнен успешно. Общее время выполнения: 79 мсес.  
обработано строк: 10.

Рисунок 8 – удаление записи с проверкой существования

The screenshot shows the same database query window as Figure 8, but the execution result shows a message in the "Сообщения" tab: "ЗАМЕЧАНИЕ: Процесс завершился с кодом 0".

```
1  err integer;
2
3  begin
4
5
6  begin
7  IF NOT EXISTS ( SELECT 1 FROM project WHERE project_name = _name) then
8      err=1;
9      RAISE NOTICE 'проекта с таким названием не существует';
10 else
11     err=0;
12     delete from project where project_name = _name;
13 end if;
14 RAISE NOTICE 'Процесс завершился с кодом %', err;
15 end;
16 $$;
17 --Вызовем процедуру.
18
19 CALL del_projects('Оно');
20 select * from project order by project_no;
21
```

Запрос История запросов

Data Output Сообщения Notifications

ЗАМЕЧАНИЕ: Процесс завершился с кодом 0

Запрос выполнен успешно. Общее время выполнения: 185 мсес.  
обработано строк: 10.

Рисунок 9 – удачное завершение удаления

*Удаление записи из таблицы Поручения:*

```
CREATE OR REPLACE PROCEDURE del_assign(_as_id integer)
  LANGUAGE plpgsql

AS $$
declare
  err integer;
begin
  IF NOT EXISTS ( SELECT 1 FROM assignment WHERE assignment_id =
_as_id) then
    err=1;
    RAISE NOTICE 'поручения с таким ключом не существует';
  else
    err=0;
    delete from assignment where assignment_id = _as_id;
  end if;
  RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.

CALL del_assign(1001);
select * from assignment order by assignment_id desc;
```

The screenshot displays a database IDE interface. The top section, titled 'Запрос' (Query), shows the SQL code being executed, which is a PL/pgSQL procedure named 'del\_assign' and its subsequent calls. The code is color-coded: keywords in purple, strings in red, and comments in orange. The bottom section, titled 'Сообщения' (Messages), shows the output of the execution, including two notices and a final status message.

Запрос	История запросов
6 begin	
7 IF NOT EXISTS ( SELECT 1 FROM assignment WHERE assignment_id = _as_id) then	
8 err=1;	
9 RAISE NOTICE 'поручения с таким ключом не существует';	
10 else	
11 err=0;	
12 delete from assignment where assignment_id = _as_id;	
13 end if;	
14 RAISE NOTICE 'Процесс завершился с кодом %', err;	
15 end;	
16 \$\$;	
17 --Вызовем процедуру.	
18	
19 CALL del_assign(1001);	
20 select * from assignment order by assignment_id desc;	
21	

Data Output	Сообщения	Notifications
	ЗАМЕЧАНИЕ: поручения с таким ключом не существует	
	ЗАМЕЧАНИЕ: Процесс завершился с кодом 1	
	Запрос выполнен успешно. Общее время выполнения: 142 msec.	
	обработано строк: 1001.	

Рисунок 10 – удаление записи

*Удаление записи из таблицы Работник:*

```
CREATE OR REPLACE PROCEDURE del_worekr(_name text)
  LANGUAGE plpgsql

AS $$
declare
  err integer;
begin
  IF NOT EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name)
then
    err=1;
    RAISE NOTICE 'человека с таким ФИО не существует';
  else
    err=0;
    delete from worker
      where worker_name=_name;
  end if;
  RAISE NOTICE 'Процесс завершился с кодом %', err;
end;
$$;
--Вызовем процедуру.
CALL del_projects('Кузнецов');
select * from worker order by worker_id;
```

Запрос История запросов

```
5  err integer;  
6  begin  
7  IF NOT EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name) then  
8      err=1;  
9      RAISE NOTICE 'человека с таким ФИО не существует';  
10 else  
11     err=0;  
12     delete from worker  
13     where worker_name=_name;  
14 end if;  
15 RAISE NOTICE 'Процесс завершился с кодом %', err;  
16 end;  
17 $$;  
18 --Вызовем процедуру.  
19 CALL del_projects('Кузнецов');  
20 select * from worker order by worker_id;  
21
```

Data Output   **Сообщения**   Notifications

ЗАМЕЧАНИЕ: человека с таким ФИО не существует  
ЗАМЕЧАНИЕ: Процесс завершился с кодом 1

Запрос выполнен успешно. Общее время выполнения: 77 msec.  
обработано строк: 11.

Рисунок 11 – удаление записи с проверкой на существование работника

Запрос История запросов

```
5  err integer;  
6  begin  
7  IF NOT EXISTS ( SELECT 1 FROM worker WHERE worker_name = _name) then  
8      err=1;  
9      RAISE NOTICE 'человека с таким ФИО не существует';  
10 else  
11     err=0;  
12     delete from worker  
13     where worker_name=_name;  
14 end if;  
15 RAISE NOTICE 'Процесс завершился с кодом %', err;  
16 end;  
17 $$;  
18 --Вызовем процедуру.  
19 CALL del_projects('Кузнецов Дмитрий Данилович');  
20 select * from worker order by worker_id;  
21
```

Data Output   **Сообщения**   Notifications

ЗАМЕЧАНИЕ: Процесс завершился с кодом 0

Запрос выполнен успешно. Общее время выполнения: 110 msec.  
обработано строк: 10.

Рисунок 12 – успешное завершение процедуры

#### Задание 4

*Запрос работника по первичному ключу:*

```
CREATE OR REPLACE FUNCTION get_id(_id integer)
    returns setof worker LANGUAGE sql
```

```
AS $$
```

```
select * from worker where worker_id=_id;
```

```
$$;
```

```
--Вызовем процедуру.
```

```
SELECT * FROM get_id(1);
```

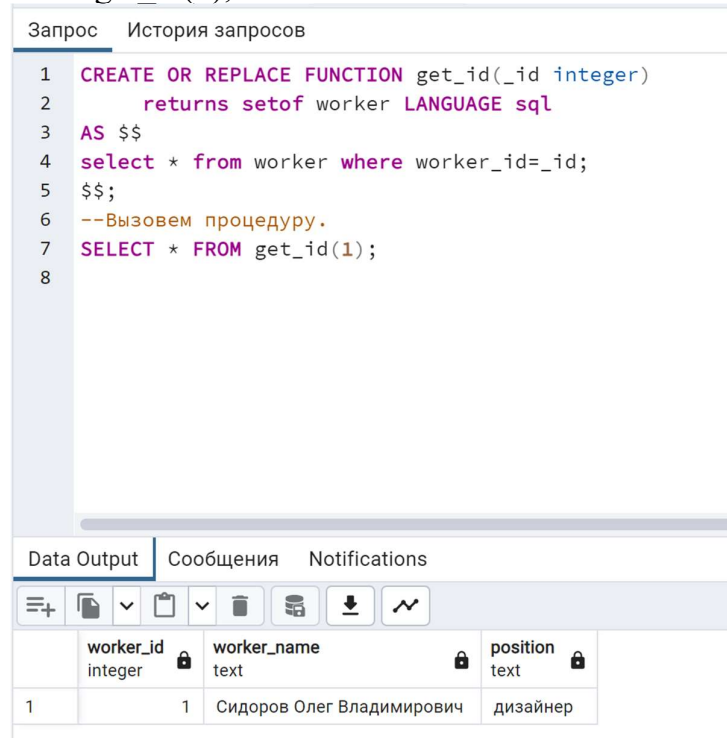


Рисунок 13 – вывод запроса

*Запрос работника по ФИО:*

```
CREATE OR REPLACE FUNCTION get_fio(_name text)
    returns setof worker LANGUAGE sql
```

```
AS $$
```

```
select * from worker where worker_name=_name;
```

```
$$;
```

```
--Вызовем процедуру.
```

```
SELECT * FROM get_fio('Повова Жанна Романовна');
```

```

1 CREATE OR REPLACE FUNCTION get_fio(_name text)
2     returns setof worker LANGUAGE sql
3 AS $$
4 select * from worker where worker_name=_name;
5 $$;
6 --Вызовем процедуры.
7 SELECT * FROM get_fio('Повова Жанна Романовна');
8

```

Data Output   Сообщения   Notifications			
<div> <div>⋮</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	worker_id integer	worker_name text	position text
1	4	Повова Жанна Романовна	сценарист

Рисунок 14 – вывод запроса

*Запрос работника по поручению:*

```

CREATE OR REPLACE FUNCTION get_worker(_id integer)
    returns table(assignment_id integer,
                  worker_id integer,
                  worker_name text,
                  position text) stable language sql

AS $$
select a.assignment_id, w.worker_id, w.worker_name, w.position
from assignment as a
inner join worker as w on a.worker_id=w.worker_id
where a.assignment_id=_id;
$$;
--Вызовем функцию.
SELECT * FROM get_worker(3);

```



Запрос
История запросов

```

1 CREATE OR REPLACE FUNCTION get_worker(_id integer)
2     returns table(assignment_id integer,
3                   worker_id integer,
4                   worker_name text,
5                   pposition text) stable language sql
6 AS $$
7 select a.assignment_id, w.worker_id, w.worker_name, w.position
8 from assignment as a
9 inner join worker as w on a.worker_id=w.worker_id
10 where a.assignment_id=_id;
11 $$;
12 --Вызовем функцию.
13 SELECT * FROM get_worker(3);
14

```

Data Output
Сообщения
Notifications

+

📄

▼

📋

▼

🗑️

🗑️

📥

⬇️

📈

	assignment_id integer	worker_id integer	worker_name text	pposition text
1	3	2	Фёдоров Иван Алексеевич	режиссёр

Рисунок 15 – вывод запроса

### Задание 5

Запрос списка поручений, выданных данному работнику

```

CREATE OR REPLACE FUNCTION get_list(_id integer)
    returns table(worker_id integer,
                  project_no integer,
                  project_name text,
                  project_complexity integer,
                  deadline date) stable language sql

AS $$
select distinct a.worker_id, p.project_no, p.project_name, p.project_complexity,
p.deadline
from assignment as a
inner join project as p on a.project_no=p.project_no
where worker_id=_id order by project_no;
$$;
--Вызовем функцию.
SELECT * FROM get_list(10);

```

ЗапросИстория запросов

```

6      project_complexity integer,
7      deadline date) stable language sql
8  AS $$
9  select distinct a.worker_id, p.project_no, p.project_name, p.project_complexity,
10 from assignment as a
11 inner join project as p on a.project_no=p.project_no
12 where worker_id=_id order by project_no;
13 $$;
14 --Вызовем функцию.
15 SELECT * FROM get_list(10);
16

```

Data OutputСообщенияNotifications

	worker_id integer	project_no integer	project_name text	project_complexity integer	deadline date
1	10	1	Барби	60	2023-06-01
2	10	2	Бойцовский клуб	80	2046-04-12
3	10	3	Большая игра	45	2050-03-13
4	10	4	Игра в имитацию	76	2012-11-07
5	10	5	Шерлок	23	2024-07-01
6	10	6	Тор	10	2025-05-25
7	10	7	Настоящий детектив	69	2011-10-30
8	10	8	Оппенгеймер	90	2015-09-09
9	10	9	Иллюзия обмана	52	2004-02-23
10	10	10	Престиж	46	2004-04-15

Рисунок 16 – вывод запроса

**Вывод:** мы поняли назначения пользовательских функции и процедур, их написание и использование.