|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 Прикладная информатика**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 3 |

**Название:**

Пользовательские функции и процедуры

**Дисциплина:** Базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-35 Б |  |  | Дулина И.А. |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Кудрявцев А.П. |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2023

**Вариант 8**

**Цель:**

Данная лабораторная работа призвана сформировать у студента понимание назначения пользовательских функции 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.
* Научится 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;

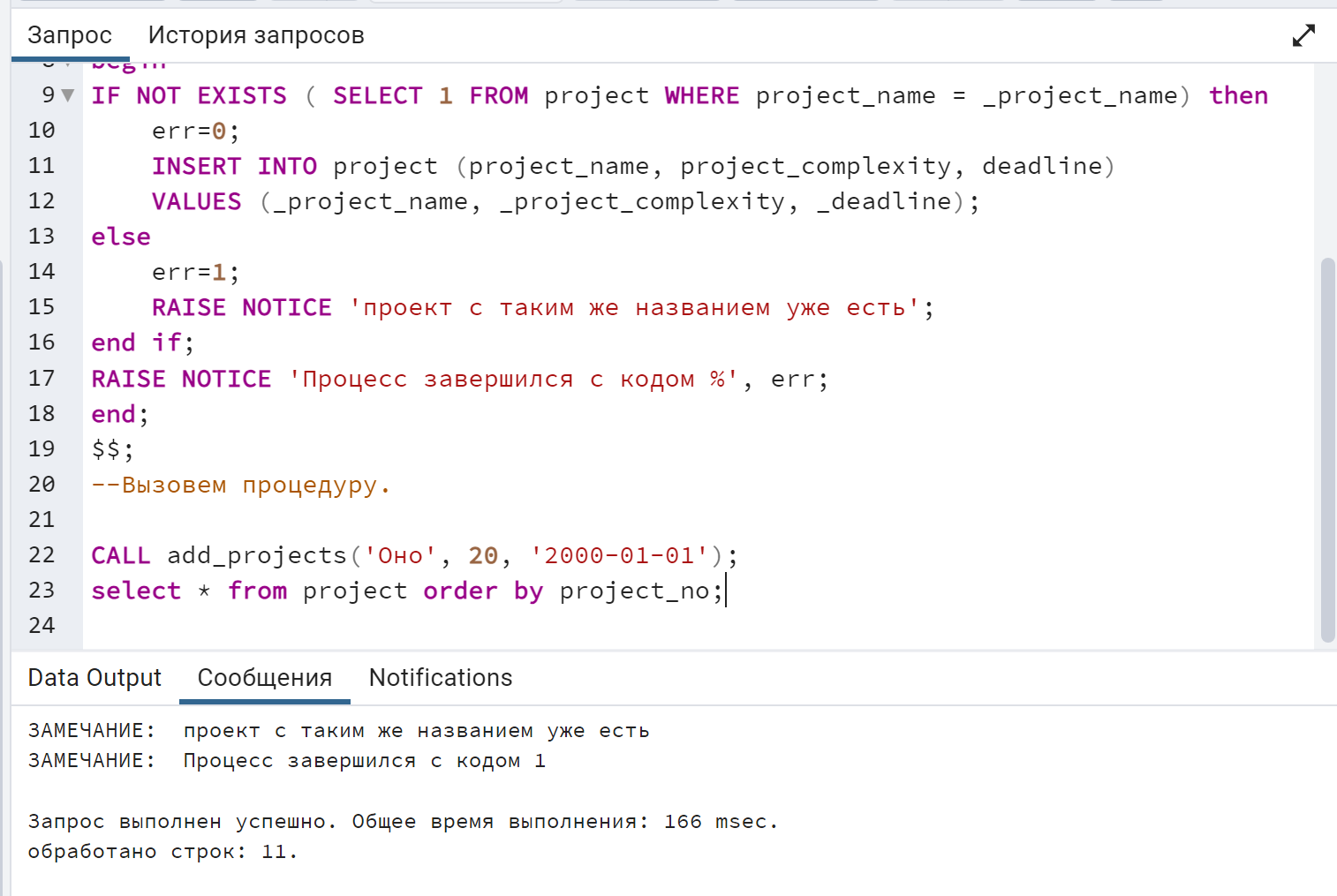


Рисунок 1 – добавление записи через процедуру с проверкой дублирования

*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 планновой 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;

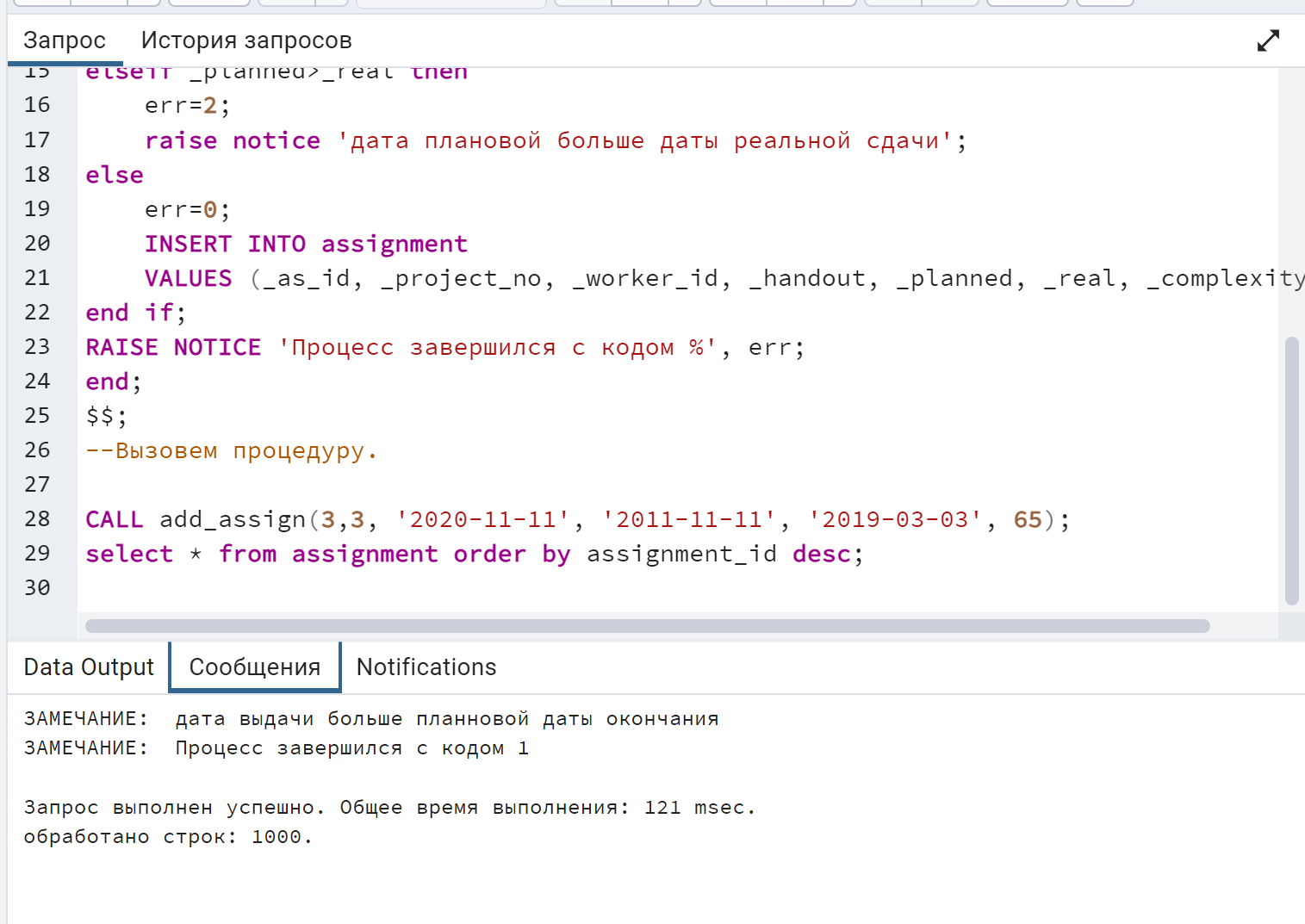


Рисунок 2 – добавление записи через процедуру с проверкой на корректность ввода данных

*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;

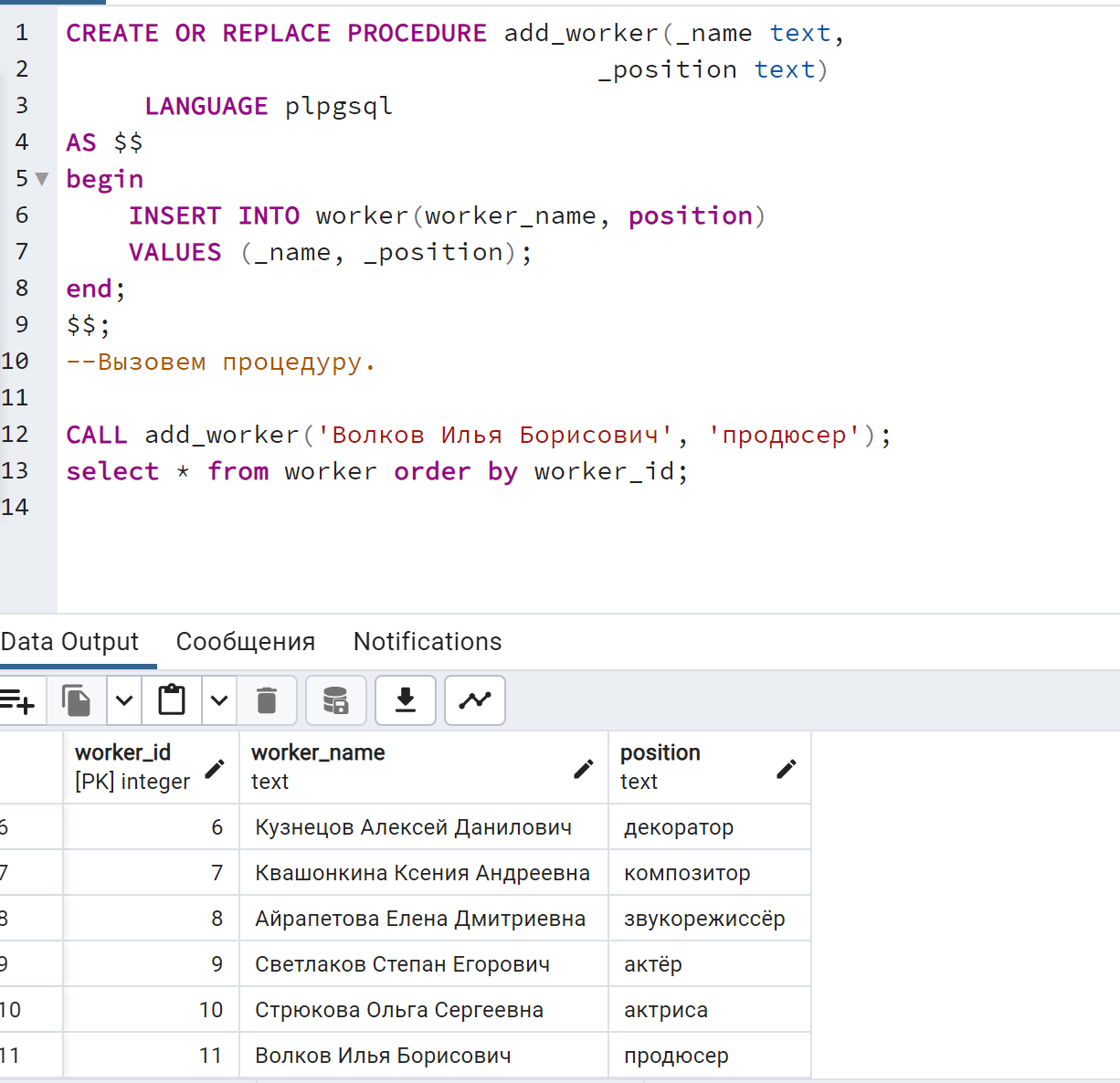


Рисунок 3 – добавление записи

**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, 'Тор', 20, '2000-01-01');

select \* from project order by project\_no;

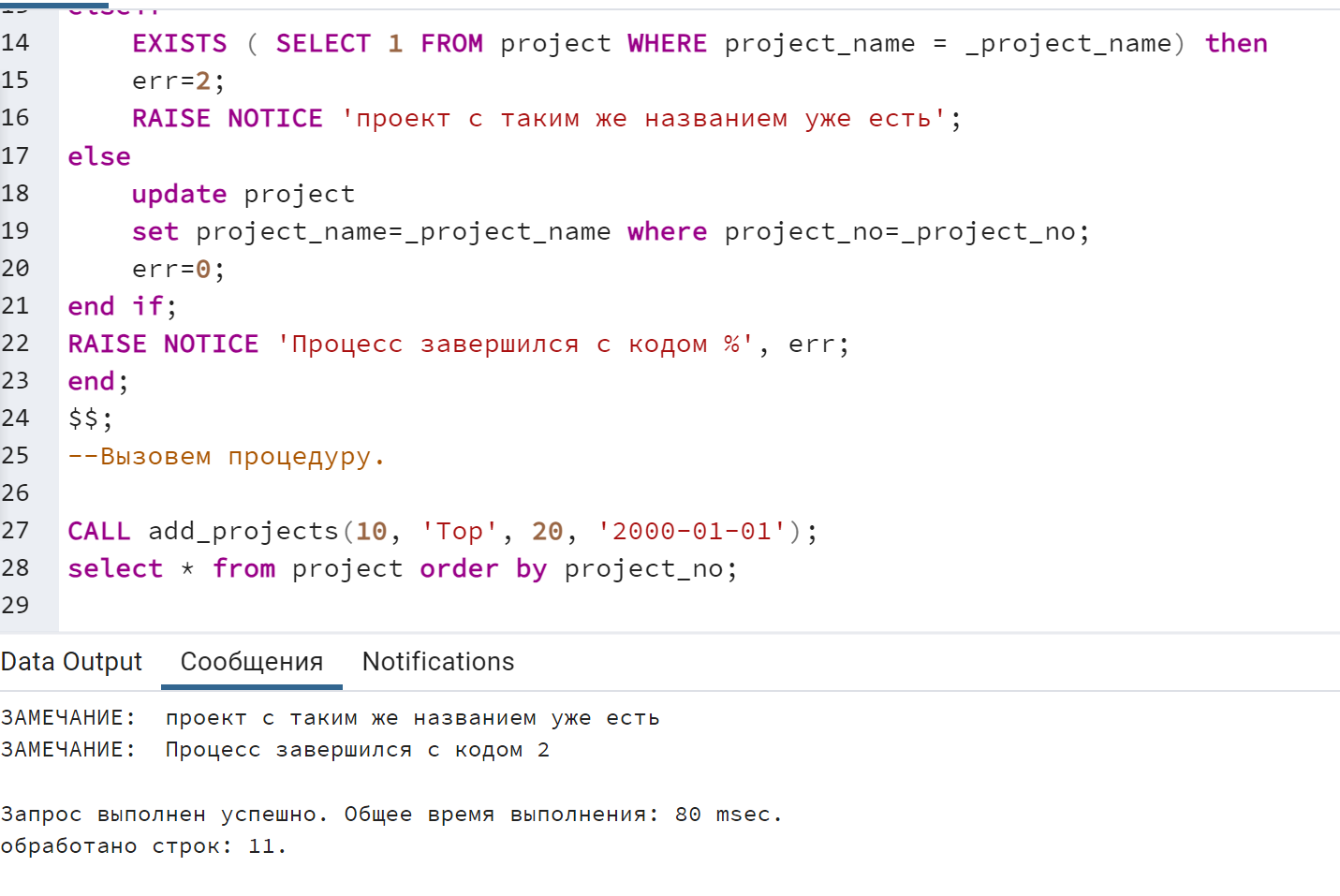


Рисунок 4 – изменение записи с проверкой на дублирование

*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;

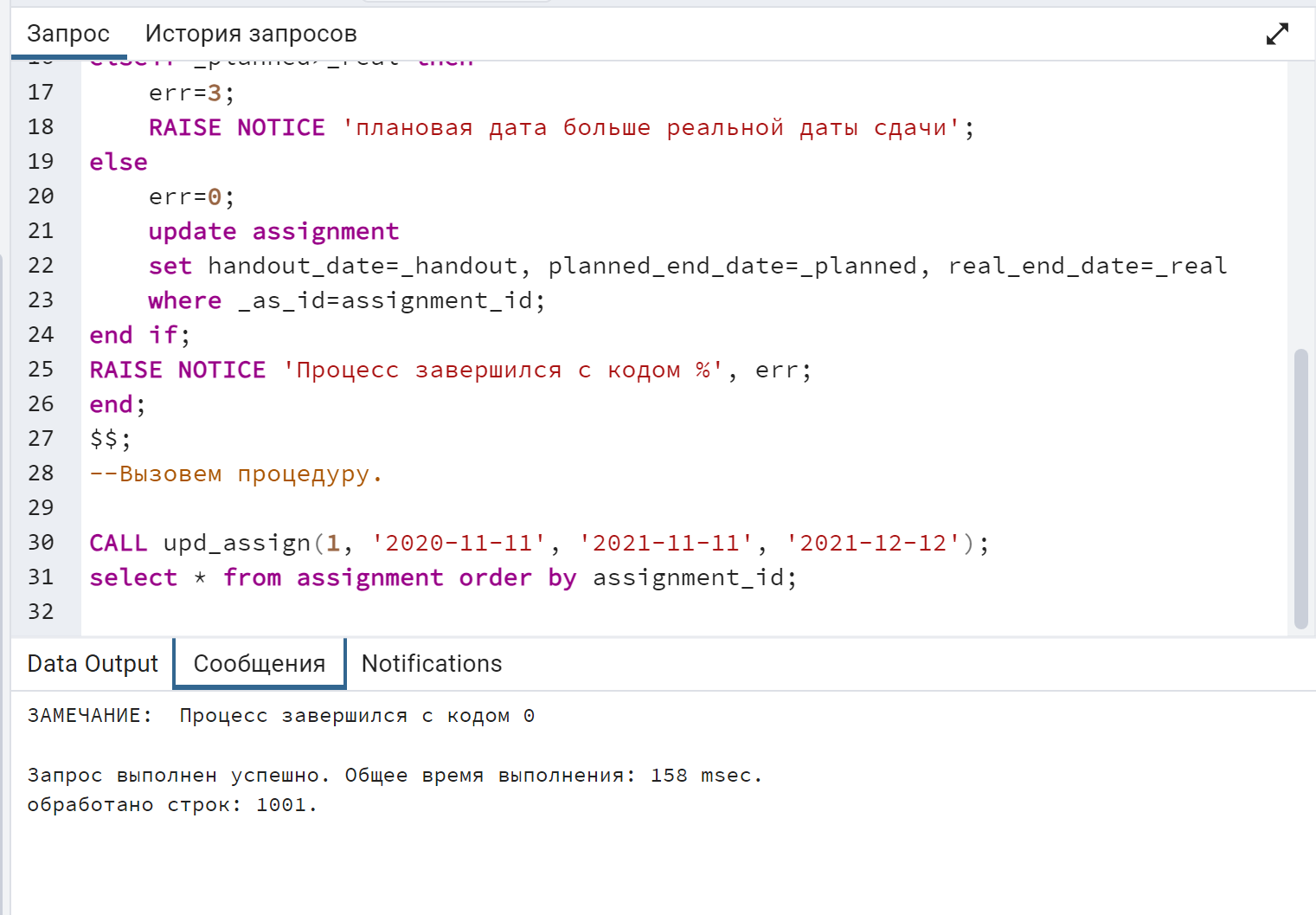


Рисунок 5 – изменение записи

*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;

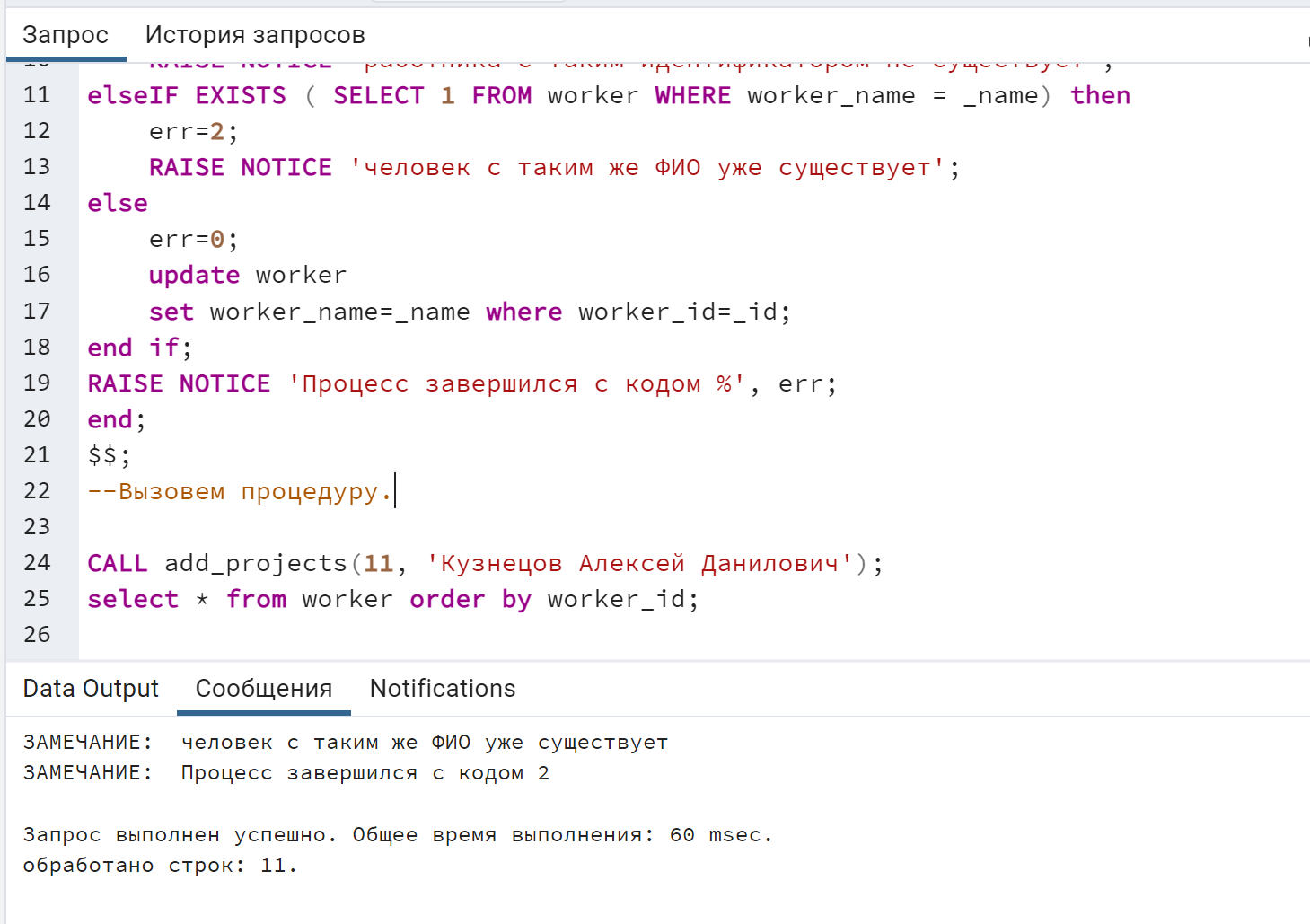


Рисунок 6 – изменение записи с проверкой на дублирование

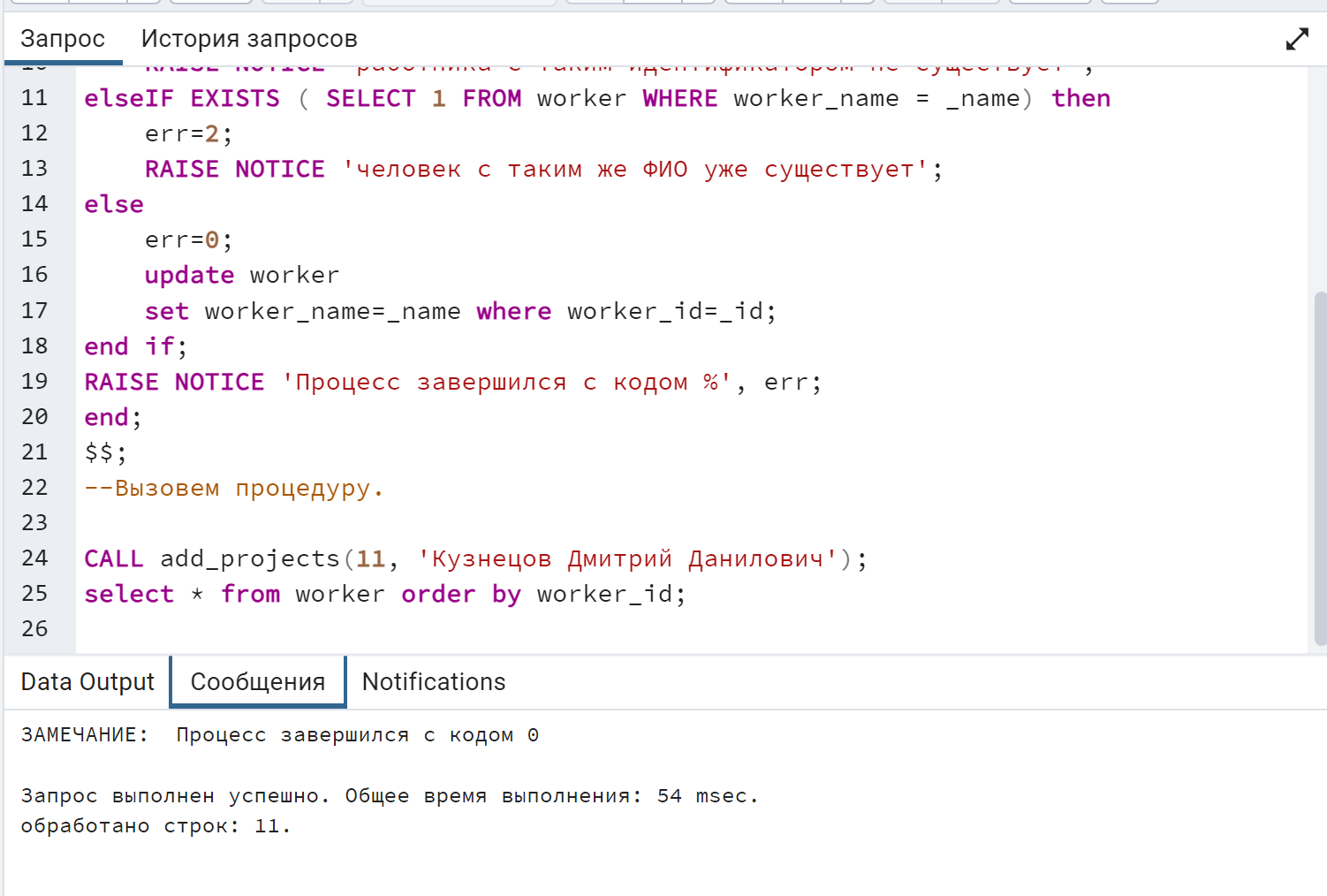


Рисунок 7 – изменение записи с успешным завершением

**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;

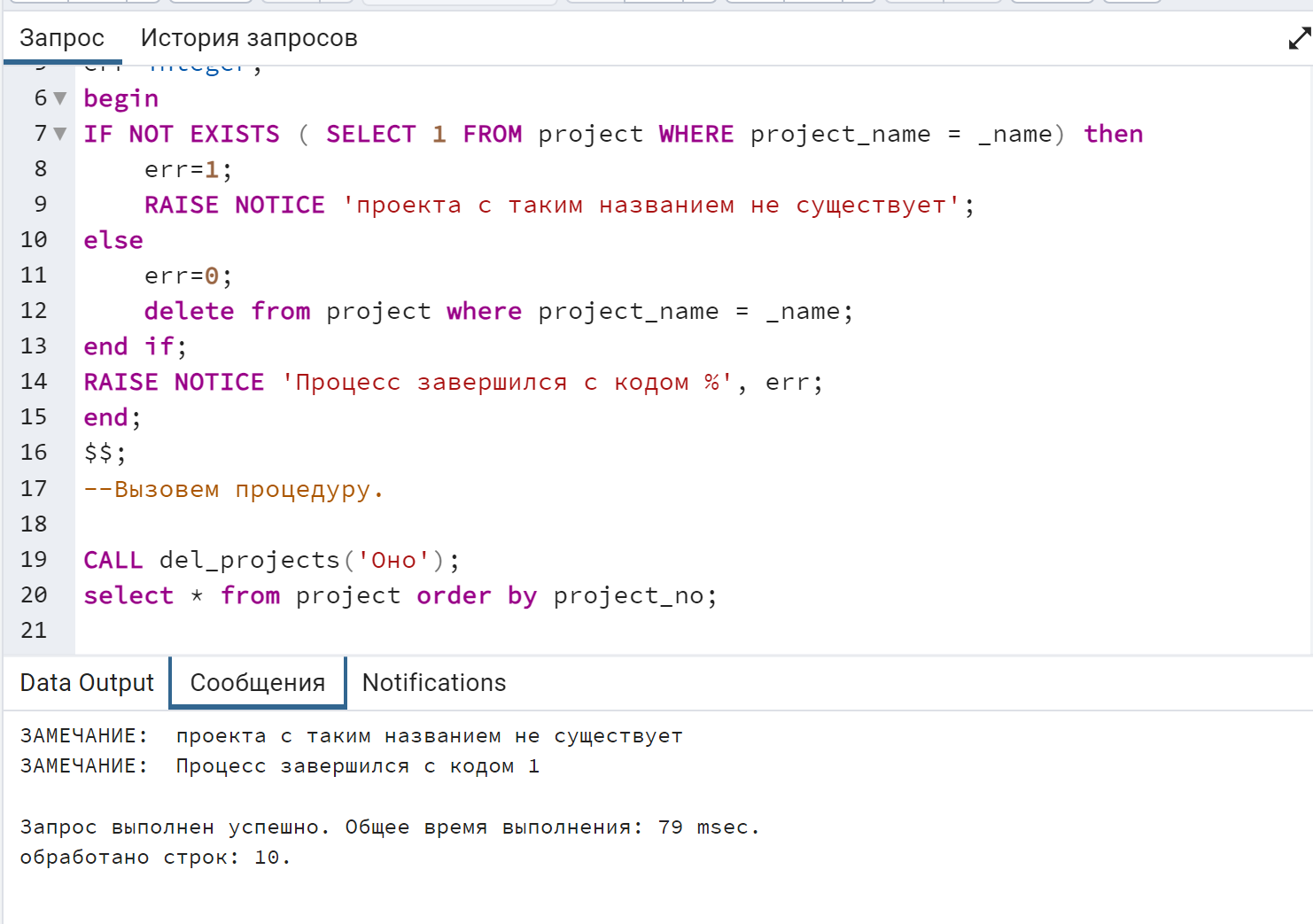


Рисунок 8 – удаление записи с проверкой существования

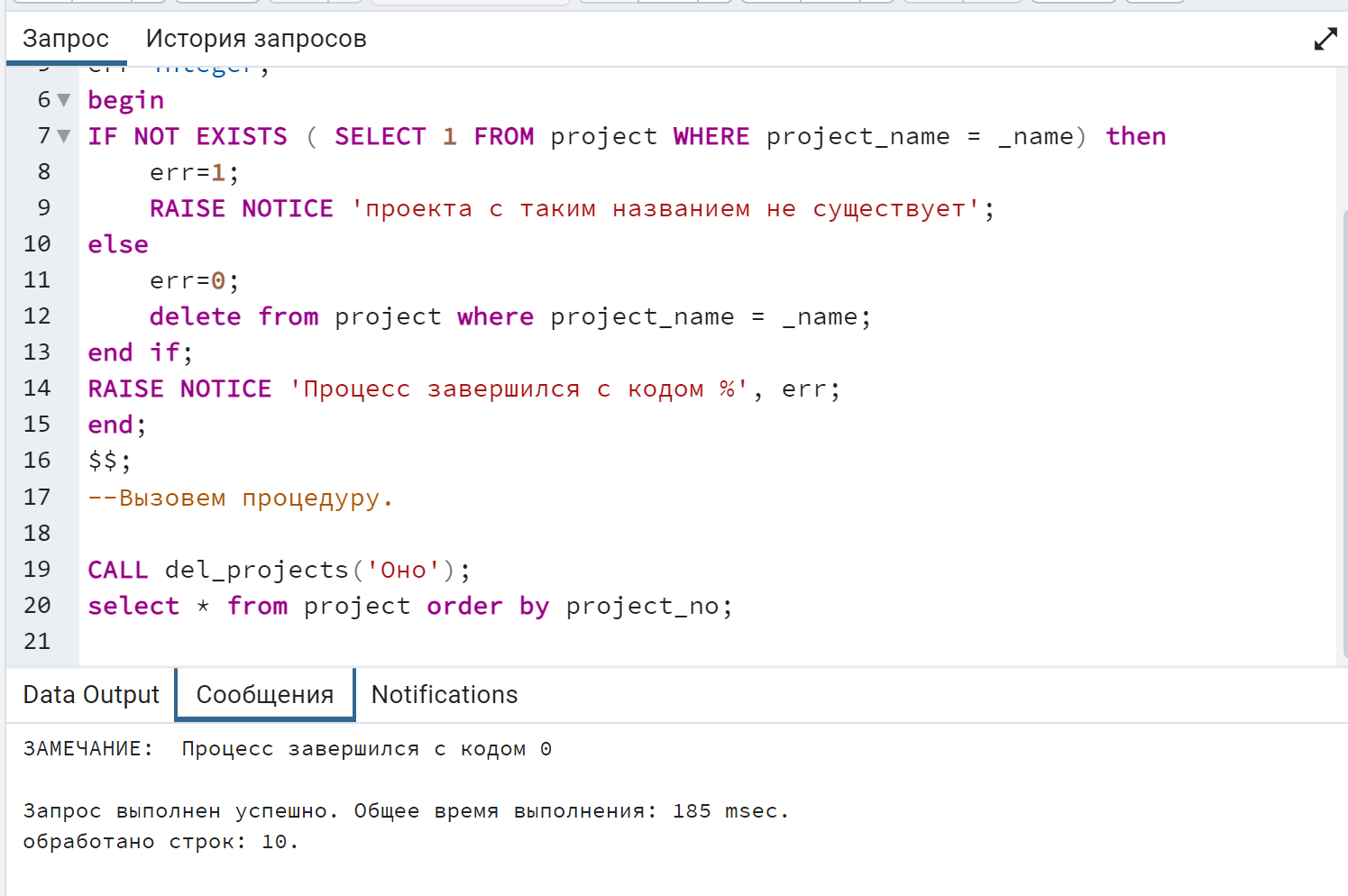


Рисунок 9 – удачное завершение удаления

*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;

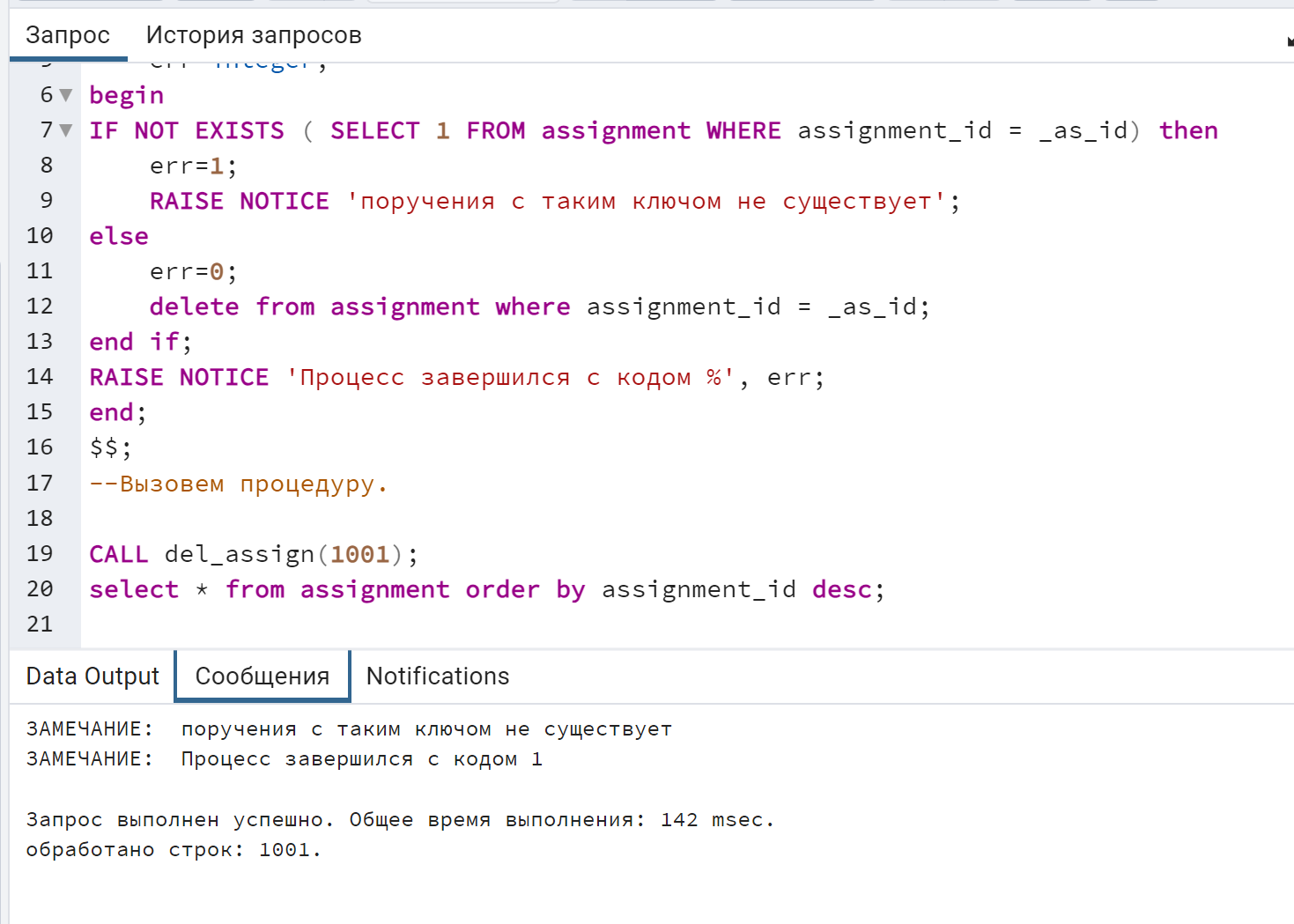
end;

$$;

--Вызовем процедуру.

CALL del\_assign(1001);

select \* from assignment order by assignment\_id desc;

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;

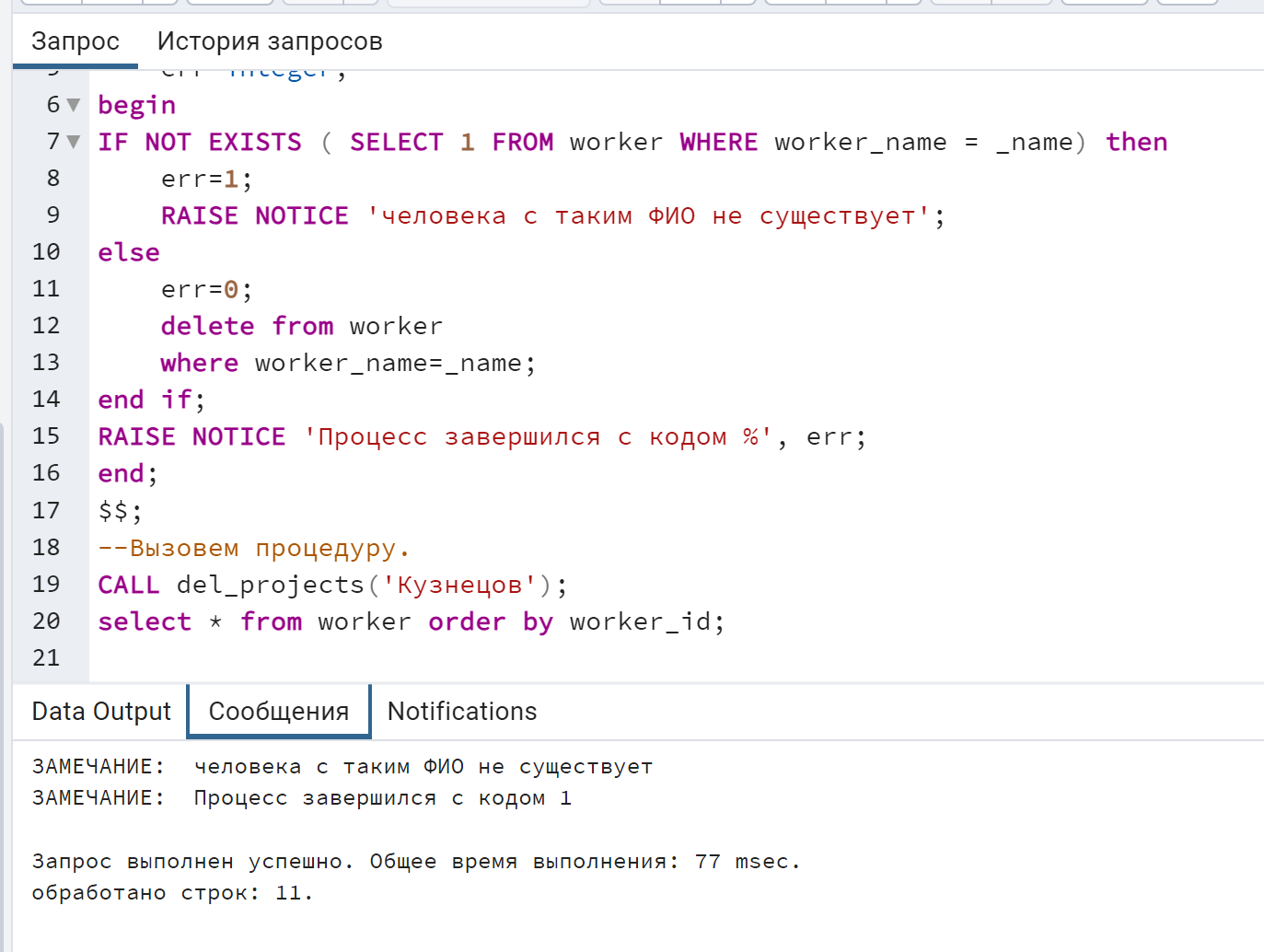


Рисунок 11 – удаление записи с проверкой на существование работника

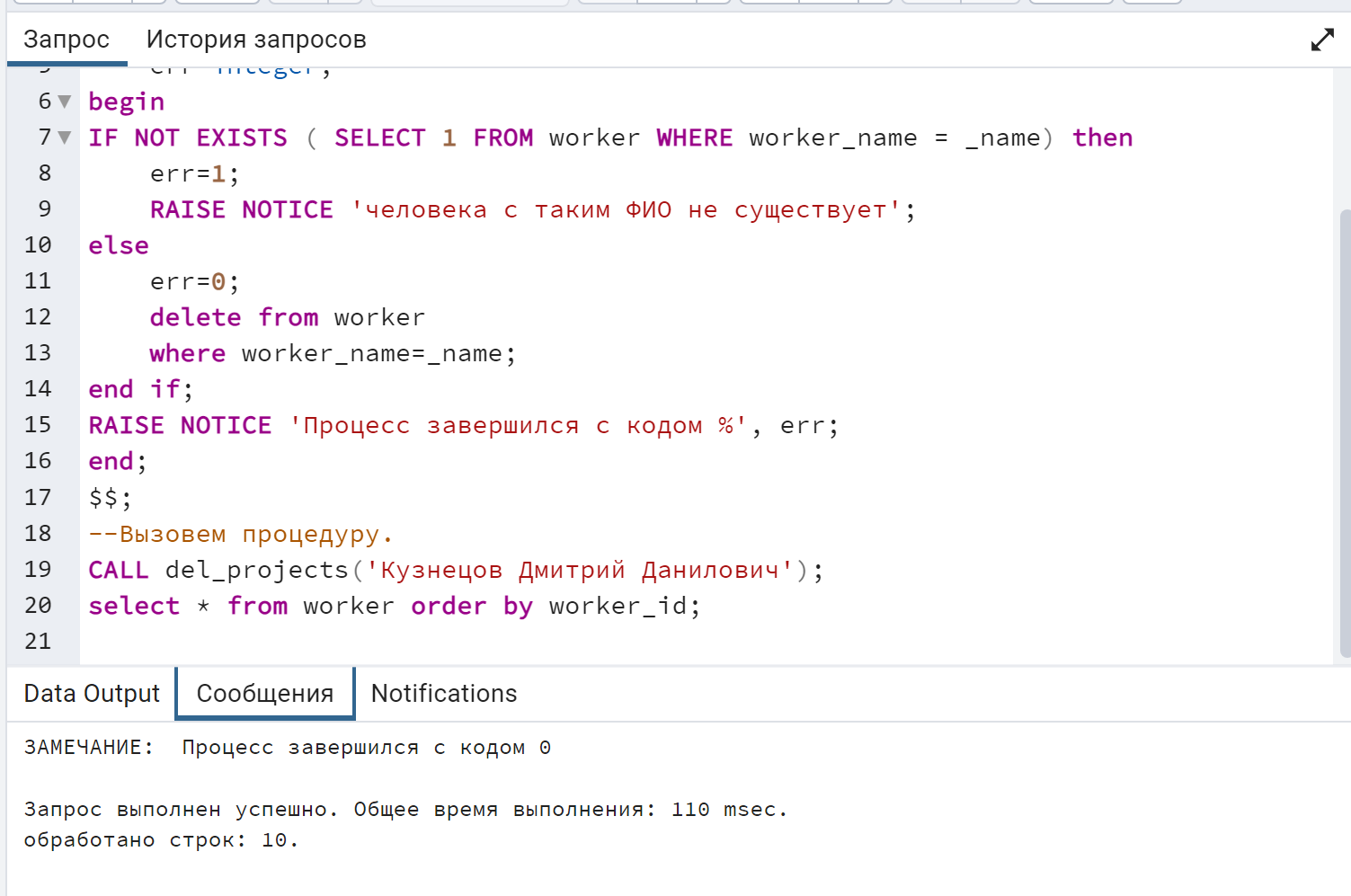


Рисунок 12 – успешное завершение процедуры

**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);

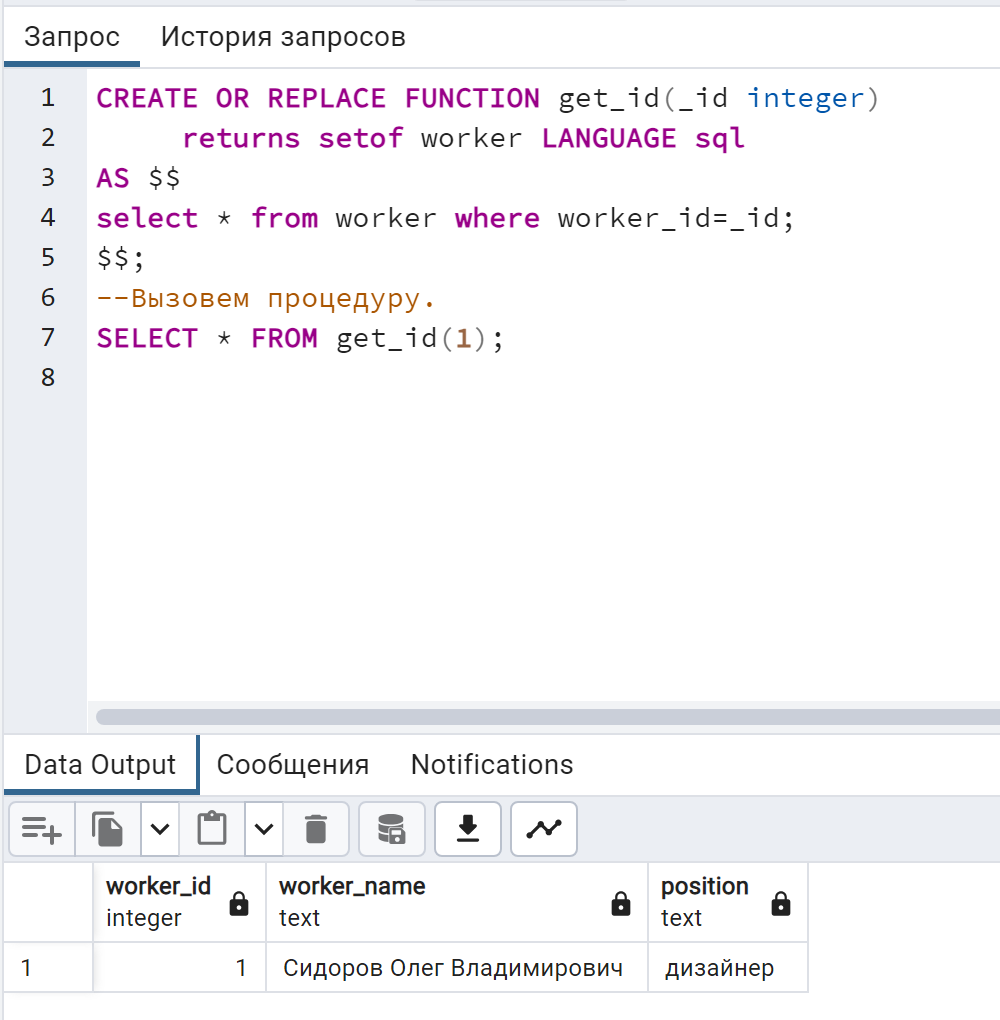


Рисунок 13 – вывод запроса

*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');

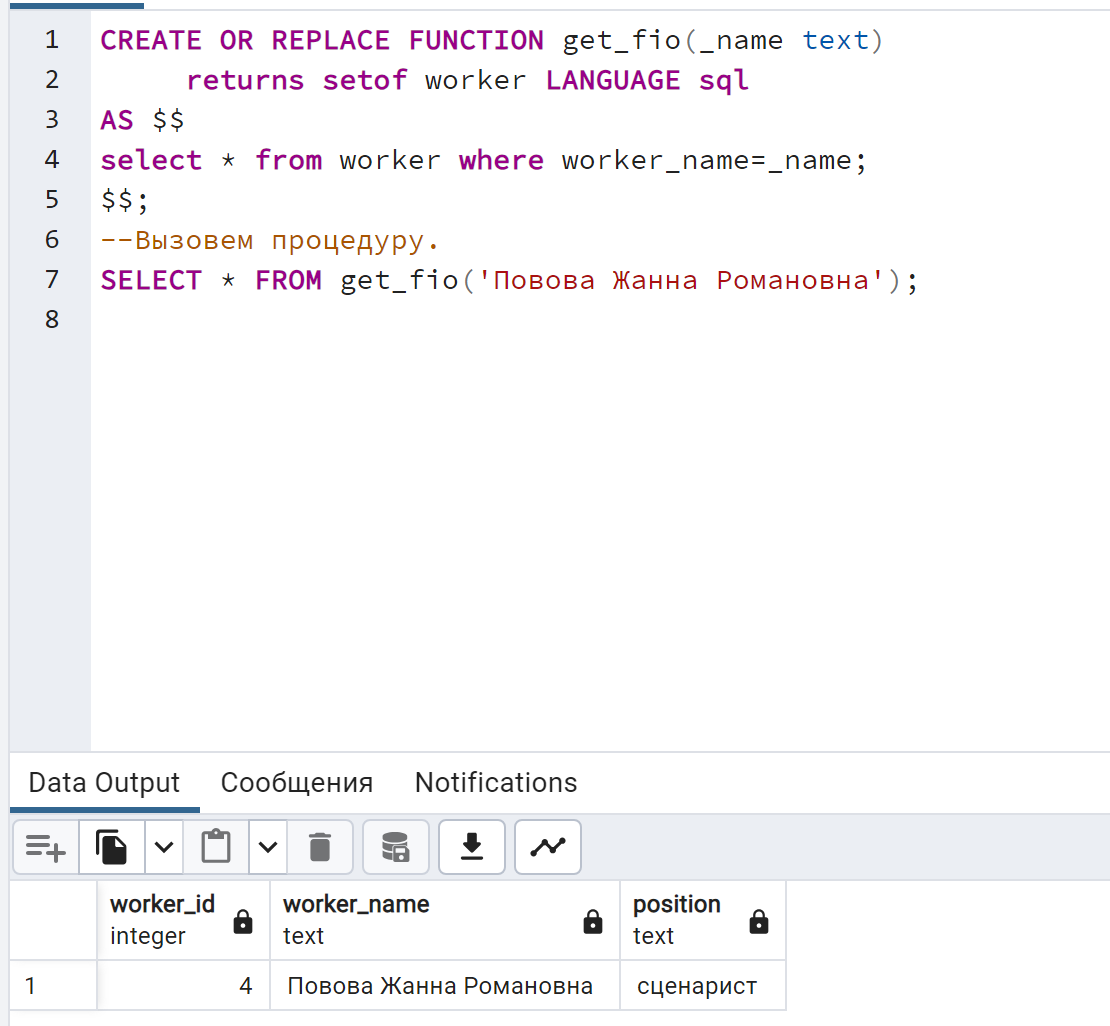


Рисунок 14 – вывод запроса

*Запрос работника по поручению:*

CREATE OR REPLACE FUNCTION get\_worker(\_id integer)

returns table(assignment\_id integer,

worker\_id integer,

worker\_name text,

pposition 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);

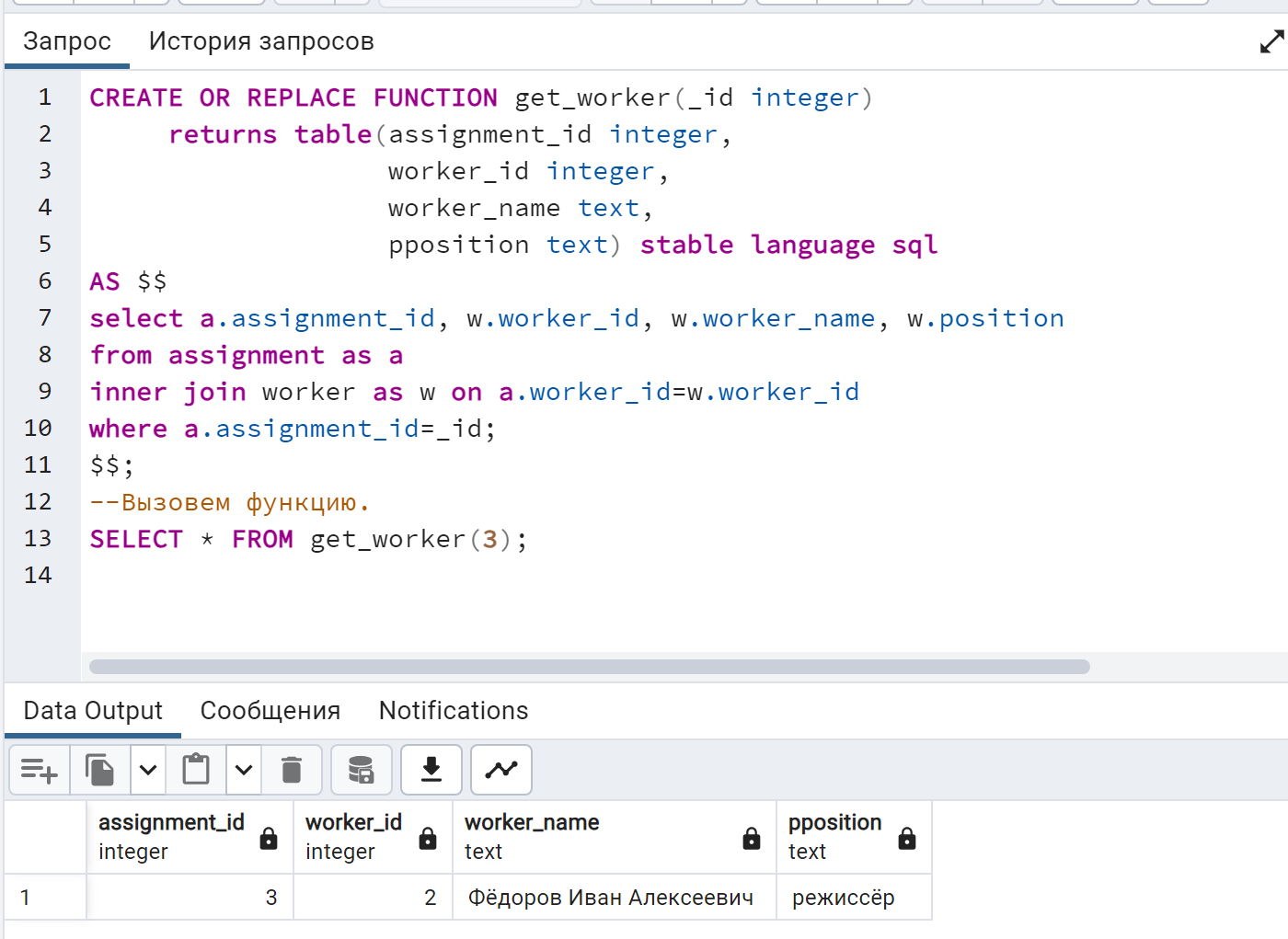


Рисунок 15 – вывод запроса

**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);

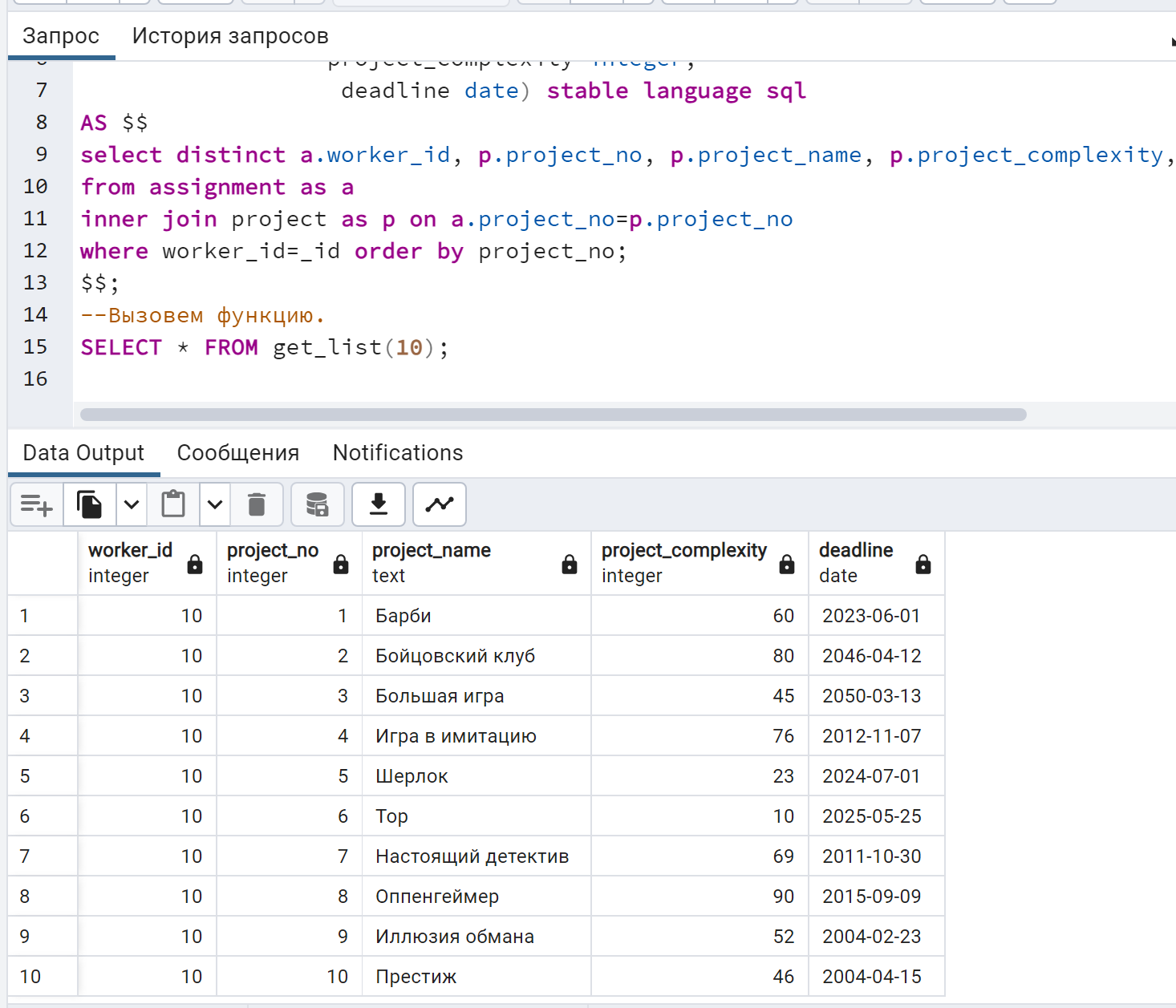


Рисунок 16 – вывод запроса

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