

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
"Национальный исследовательский университет  
"Высшая школа экономики"  
Московский институт электроники и математики им. А.Н.Тихонова  
Департамент компьютерной инженерии**

**Лабораторная работа № 6**  
по курсу «Базы данных»  
Тема: Создание функций в СУБД

Студент: Молчанов Данил  
Андреевич

Группа: М\_БД\_8

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2023

## 1. Постановка задачи

### Вариант 2

1. Функция, определяющая стаж работы человека. Параметры: стаж на прежних местах работы и дата поступления на работу. Возвращает число полных лет стажа на текущую дату.
2. Функция начисления премий сотрудникам. Входной параметр – базовая ставка. Размер премии вычисляется как произведение базовой ставки на оклад и на коэффициент, который зависит от стажа работы на данном предприятии:
  - 0.1, если стаж от 1 до 5 лет;
  - 0.2, если стаж от 5 до 10 лет;
  - 0.3, если стаж от 10 до 20 лет;
  - 0.5, если стаж от 20 до 30 лет;
  - 1, если стаж свыше 30 лет.

Результат работы функции добавляется в таблицу «Премии» (поля ФИО сотрудника, стаж, размер премии), которая в начале работы очищается от старых данных. Людям, проработавшим менее 1 года, премия не выплачивается (в таблицу они не добавляются). Использовать ранее созданную функцию.

## 2. Решение

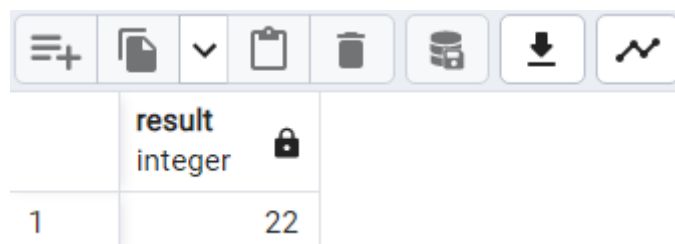
1. Функция, определяющая стаж работы человека. Параметры: стаж на прежних местах работы и дата поступления на работу. Возвращает число полных лет стажа на текущую дату:

```
CREATE
OR REPLACE FUNCTION get_experience(
    experience numeric(5, 2),
    date_taking date,
    out_result int
) AS $$ BEGIN result := experience + extract(
    year
    from
        age(date_taking)
) RETURN;
END;
$$ LANGUAGE plpgsql;
```

Тестирование:

```
select * from get_experience(12.00, '2012-11-22');
```

Результат:



The screenshot shows a database query result interface. At the top, there is a toolbar with icons for various actions like adding, deleting, and refreshing. Below the toolbar, there is a table with one column and one row. The column header is 'result' with a data type of 'integer' and a lock icon. The row contains the value '22'.

	result integer
1	22

Рисунок 1 – результат работы функции get\_experience

2. Функция начисления премий сотрудникам. Входной параметр – базовая ставка. Размер премии вычисляется как произведение базовой ставки на оклад и на коэффициент, который зависит от стажа работы на данном предприятии:

- 0.1, если стаж от 1 до 5 лет;
- 0.2, если стаж от 5 до 10 лет;
- 0.3, если стаж от 10 до 20 лет;
- 0.5, если стаж от 20 до 30 лет;
- 1, если стаж свыше 30 лет.

Результат работы функции добавляется в таблицу «Премии» (поля ФИО сотрудника, стаж, размер премии), которая в начале работы очищается от старых данных. Людям, проработавшим менее 1 года, премия не выплачивается (в таблицу они не добавляются). Использовать ранее созданную функцию.

Решение:

```
create table prize as
select
    surname,
    name_and_patronymic,
    salary * 0 as prize
FROM
    employee
    JOIN posts ON employee.curr_post = posts.post_name;
CREATE
OR REPLACE FUNCTION get_prize(
    base_rate numeric(1, 3)
```

```

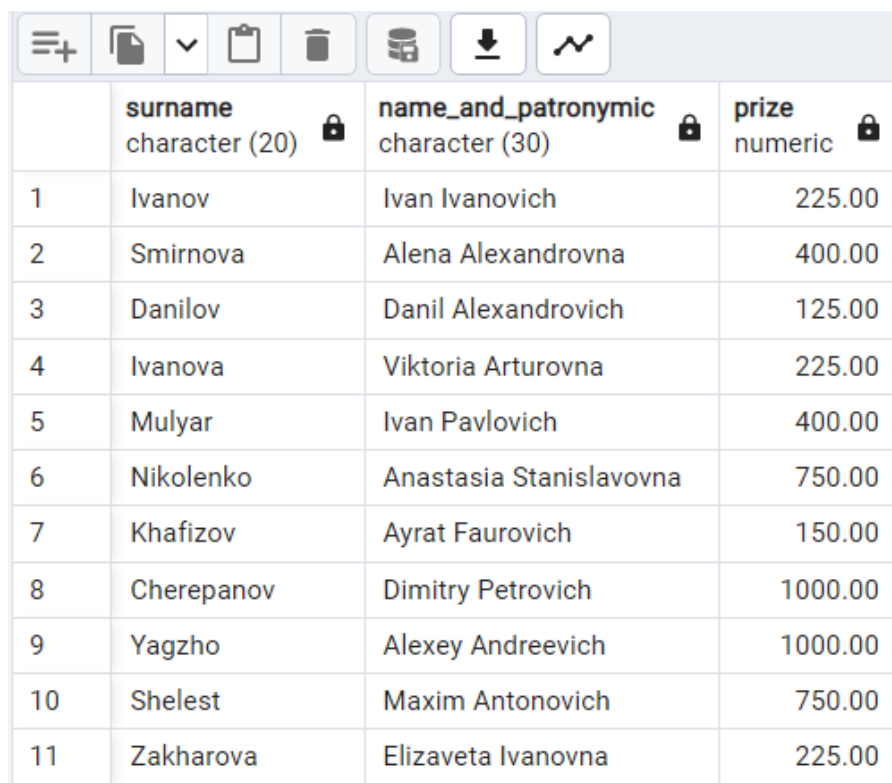
) RETURNS BOOLEAN AS $$ BEGIN
UPDATE
  prize
SET
  prize = (
    SELECT
      t.prize
    from
      (
        SELECT
          surname,
          name_and_patronymic,
          (
            base_rate * salary * CASE WHEN get_experience(
              work_experience, date_of_taking_office
            ) BETWEEN 1
            AND 5 THEN 0.1 WHEN get_experience(
              work_experience, date_of_taking_office
            ) BETWEEN 5
            AND 10 THEN 0.2 WHEN get_experience(
              work_experience, date_of_taking_office
            ) BETWEEN 10
            AND 20 THEN 0.3 WHEN get_experience(
              work_experience, date_of_taking_office
            ) BETWEEN 20
            AND 30 THEN 0.5 WHEN get_experience(
              work_experience, date_of_taking_office
            ) > 30 THEN 1 ELSE 0 END
          ) AS prize
        FROM
          employee
          JOIN posts ON employee.curr_post = posts.post_name
      ) as t
    WHERE
      prize.surname = t.surname
  );
RETURN FOUND;
END;
$$ LANGUAGE plpgsql;

```

Тестирование:

```
select get_prize(0.1);
```

Результат:



	<b>surname</b> character (20)	<b>name_and_patronymic</b> character (30)	<b>prize</b> numeric
1	Ivanov	Ivan Ivanovich	225.00
2	Smirnova	Alena Alexandrovna	400.00
3	Danilov	Danil Alexandrovich	125.00
4	Ivanova	Viktoria Arturovna	225.00
5	Mulyar	Ivan Pavlovich	400.00
6	Nikolenko	Anastasia Stanislavovna	750.00
7	Khafizov	Ayrat Faurovich	150.00
8	Cherepanov	Dimitry Petrovich	1000.00
9	Yagzho	Alexey Andreevich	1000.00
10	Shelest	Maxim Antonovich	750.00
11	Zakharova	Elizaveta Ivanovna	225.00

Рисунок 2 – результат работы функции get\_prize

Список литературы:

1. PostgreSQL Documentation - The PostgreSQL Global Development Group (1996 - 2023) [Электронный ресурс]. URL: <https://www.postgresql.org/docs/>