

# sem6\_Nemkov

## Task 1

Написать функцию, которая выводит первое слово из строки

```
CREATE OR REPLACE FUNCTION first_task(input_text text)
RETURNS text AS $$
BEGIN
    -- если на входе null, возвращаем null
    IF input_text IS NULL THEN
        RETURN NULL;
    END IF;
    -- иначе удаляем лишние пробелы и разбиваем строку на слова
    -- и возвращаем первое слово
    RETURN split_part(trim(input_text), ' ', 1);
END;
$$ LANGUAGE plpgsql;
-- пример 1 (выводит only_word)
SELECT first_task('only_word');
-- пример 2 (выводит 'first_word')
SELECT first_task('first_word second_word');
-- пример с null (выводит null)
SELECT first_task(null);
```

---

## Task 2

Написать функцию для вывода количества строк, содержащих искомое слово, в таблице.

```
CREATE OR REPLACE FUNCTION second_task(search_word text)
RETURNS integer AS $$
DECLARE
    count_result integer;
BEGIN
    EXECUTE format('SELECT COUNT(*) FROM tasks WHERE description ILIKE
    %L',
        '%' || search_word || '%')
```

```

        INTO count_result;

    RETURN count_result;
END;
$$ LANGUAGE plpgsql;
-- пример (выводит 2)
SELECT second_task('dashboard');

```

## Task 3

Написать функцию для проставления статуса “ready” для всех задач в работе “in\_work” со временем последнего обновления больше 3 дней назад

*До изменений*

```

SELECT * FROM tasks WHERE last_update < NOW() - INTERVAL '3 days'
AND task_status = 'in_work';

```

	id [PK] integer	description text	task_status status	last_update timestamp without time zone	responsible character varying (50)	is_deleted boolean
1	4	Optimize the dashboard for faster loading times.	in_work	2024-09-15 10:23:54	Vera	false
2	5	Create a database schema to store user information.	in_work	2024-10-13 16:17:47	Igor	false
3	6	Write scripts to migrate data to the new database.	in_work	2025-02-01 15:39:25	Tatiana	false
4	7	Test the authentication system for security vulnerabilities.	in_work	2024-12-25 14:15:54	Leonid	false
5	8	Perform load testing on the authentication service.	in_work	2024-01-22 09:33:42	Aleksander	false
6	9	Debug the payment gateway integration to fix errors.	in_work	2025-02-07 17:54:48	Michael	false
7	12	Ensure the API documentation is up-to-date and accurate.	in_work	2025-03-17 14:22:43	Igor	false
8	13	Design a responsive layout for the mobile version of the app.	in_work	2025-03-17 14:23:35	Anton	false
9	14	Test the responsive design on different screen sizes.	in_work	2025-03-17 14:25:54	Anton	false
10	15	Create a notification system to alert users about updates.	in_work	2025-03-17 14:20:20	Anton	false
11	16	Improve the notification system to reduce unnecessary aler...	in_work	2025-03-17 14:23:15	Elena	false

*image-153.png*

*Запускаем функцию*

```

CREATE OR REPLACE FUNCTION third_task()
RETURNS void AS $$
BEGIN
    UPDATE tasks
    SET task_status = 'ready'
    WHERE task_status = 'in_work'
    AND last_update < NOW() - INTERVAL '3 days';
END;
$$ LANGUAGE plpgsql;

```

```
SELECT third_task();
```

## После изменений

```
SELECT * FROM tasks WHERE last_update < NOW() - INTERVAL '3 days'
AND task_status = 'in_work';
```

Возвращает пустой список

```
SELECT * FROM tasks WHERE last_update < NOW() - INTERVAL '3 days'
AND task_status = 'ready';
```

	id [PK] integer ↗	description text ↗	task_status status ↗	last_update timestamp without time zone ↗	responsible character varying (50) ↗	is_deleted boolean ↗
1	4	Optimize the dashboard for faster loading times.	ready	2024-09-15 10:23:54	Vera	false
2	5	Create a database schema to store user information.	ready	2024-10-13 16:17:47	Igor	false
3	6	Write scripts to migrate data to the new database.	ready	2025-02-01 15:39:25	Tatiana	false
4	7	Test the authentication system for security vulnerabilities.	ready	2024-12-25 14:15:54	Leonid	false
5	8	Perform load testing on the authentication service.	ready	2024-01-22 09:33:42	Aleksander	false
6	9	Debug the payment gateway integration to fix errors.	ready	2025-02-07 17:54:48	Michael	false
7	12	Ensure the API documentation is up-to-date and accurate.	ready	2025-03-17 14:22:43	Igor	false
8	13	Design a responsive layout for the mobile version of the app.	ready	2025-03-17 14:23:35	Anton	false
9	14	Test the responsive design on different screen sizes.	ready	2025-03-17 14:25:54	Anton	false
10	15	Create a notification system to alert users about updates.	ready	2025-03-17 14:20:20	Anton	false
11	16	Improve the notification system to reduce unnecessary aler...	ready	2025-03-17 14:23:15	Elena	false
Total rows: 11 of 11    Query complete 00:00:00.139    Ln 12, Col 1						

image-154.png

## Task 4

Написать функцию и добавить триггер в таблицу, в результате чего вместо удаления строки будет проставляться флаг, что строка удалена, если последнее обновление записи младше недели, а если больше – удалять запись.

```
CREATE OR REPLACE FUNCTION fourth_task()
RETURNS TRIGGER AS $$
BEGIN
    -- последнее обновление записи младше недели, проставляем флаг
    -- удаления
    IF OLD.last_update > NOW() - INTERVAL '1 week' THEN
        UPDATE tasks
        SET is_deleted = TRUE
        WHERE id = OLD.id;
```

```

        RETURN NULL; -- прерываем выполнение триггера, чтобы избежать
фактического удаления
    ELSE
        -- последнее обновление записи старше недели, удаляем запись
        DELETE FROM tasks
        WHERE id = OLD.id;
        RETURN OLD; -- возвращаем старое значение для фактического
удаления
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER my_trigger
BEFORE DELETE ON tasks
FOR EACH ROW
EXECUTE FUNCTION fourth_task();

-- удаление строки с id = 1 (последнее обновление младше недели)
DELETE FROM tasks WHERE id = 1;
SELECT * FROM tasks;

-- удаление строки с id = 5 (последнее обновление старше недели)
DELETE FROM tasks WHERE id = 5;
SELECT * FROM tasks;

```

Строка с `id = 1` была помечена как удаленная, но не фактически удалена, так как последнее обновление было менее недели назад. Строка с `id = 5` была фактически удалена, так как последнее обновление было более недели назад.

## Task 5

Написать функцию и триггер для логирования изменений в таблице.

```

CREATE TABLE task_history (
    id SERIAL PRIMARY KEY,                -- идентификатор записи
    change_time TIMESTAMP DEFAULT NOW(),  -- время изменения
    table_name TEXT,                       -- имя таблицы, в которой
произошли изменения
    operation_type TEXT,                  -- тип операции (INSERT,
UPDATE, DELETE)
    row_data JSON,                       -- данные строки (новые или

```

старые значения)

```
    row_count INT                                -- количество строк в
таблице после изменения
);

CREATE OR REPLACE FUNCTION fifth_task()
RETURNS TRIGGER AS $$
DECLARE
    current_row_count INT;
BEGIN
    -- ищем количество строк в таблице tasks
    EXECUTE format('SELECT COUNT(*) FROM %I', TG_TABLE_NAME) INTO
current_row_count;

    -- записываем логи в зависимости от типа операции
    IF TG_OP = 'INSERT' THEN
        INSERT INTO task_history (table_name, operation_type, row_data,
row_count)
        VALUES (TG_TABLE_NAME, TG_OP, row_to_json(NEW),
current_row_count);
    ELSIF TG_OP = 'UPDATE' THEN
        INSERT INTO task_history (table_name, operation_type, row_data,
row_count)
        VALUES (TG_TABLE_NAME, TG_OP, row_to_json(NEW),
current_row_count);
    ELSIF TG_OP = 'DELETE' THEN
        INSERT INTO task_history (table_name, operation_type, row_data,
row_count)
        VALUES (TG_TABLE_NAME, TG_OP, row_to_json(OLD),
current_row_count);
    END IF;

    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

-- триггер будет вызывать функцию fifth_task
-- после каждой операции INSERT, UPDATE или DELETE в таблице tasks
CREATE TRIGGER task_changes_trigger
AFTER INSERT OR UPDATE OR DELETE ON tasks
FOR EACH ROW
EXECUTE FUNCTION fifth_task();
```

-- пример

```
INSERT INTO tasks (description, task_status, last_update, responsible,
is_deleted)
VALUES ('New task for testing', 'open', NOW(), 'Test User', FALSE);
```

	id [PK] integer	change_time timestamp without time zone	table_name text	operation_type text	row_data json
1	1	2025-03-21 15:11:54.045715	tasks	INSERT	{"id":21,"description":"New task for testing","task_status":"open","last_update":"2025-03-21T15:11:54.045715"}

image-155.png