

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего  
образования

Санкт-Петербургский национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

**Лабораторная работа 3. Создание представления и индексов в PostgreSQL**

По дисциплине «Проектирование баз данных»

Выполнил:

студент группы №М3212

Тимофеев Вячеслав

Проверила:

Чеботарева



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург  
2025

## Задача:

Создать представление и индексы к нескольким таблицам (минимум к двум).

## Требования к представлению:

1. Должны быть задействованы минимум 2 таблицы.
2. Должно быть задействовано минимум 4 поля.

## Требования к индексу:

1. Должны использоваться минимум 2 поля для создания индекса.
2. Должно быть создано минимум 2 индекса к разным таблицам.

## Ход работы:

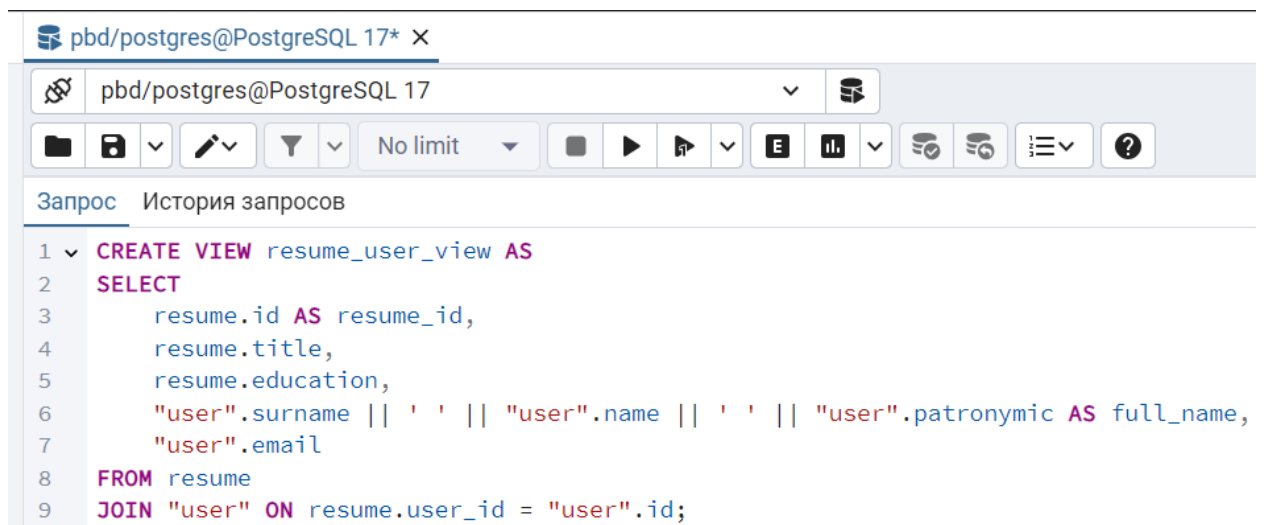
### Представление

Создадим представление для таблиц:

- user
- resume

задействуя атрибуты:

- ✓ resume.id
- ✓ resume.title
- ✓ resume.education
- ✓ user.name
- ✓ user.surname
- ✓ user.patronymic
- ✓ user.email



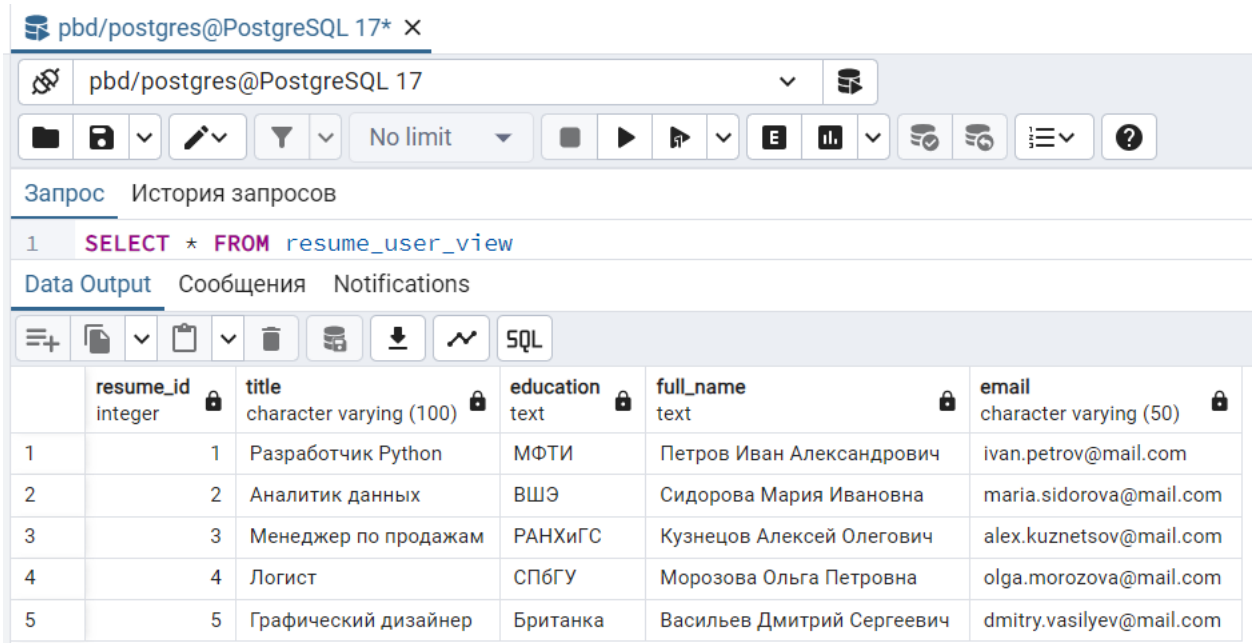
The screenshot shows a PostgreSQL client window titled 'pbd/postgres@PostgreSQL 17\* x'. The connection is 'pbd/postgres@PostgreSQL 17'. The interface includes a toolbar with icons for file operations, query execution, and view management. The 'Запрос' (Query) tab is active, displaying the following SQL code:

```
1 CREATE VIEW resume_user_view AS
2 SELECT
3     resume.id AS resume_id,
4     resume.title,
5     resume.education,
6     "user".surname || ' ' || "user".name || ' ' || "user".patronymic AS full_name,
7     "user".email
8 FROM resume
9 JOIN "user" ON resume.user_id = "user".id;
```

**Назначение представления:** доступ к информации о резюме и пользователе. С его помощью можно быстро находить кандидатов по

специальности и конкретному месту учебы; отображение контактных данных без дополнительных JOIN'ов.

### Данные из представления:



The screenshot shows a PostgreSQL client window titled 'pbd/postgres@PostgreSQL 17\*'. The query bar contains 'SELECT \* FROM resume\_user\_view'. Below the query bar, there are tabs for 'Запрос' (Query), 'История запросов' (Query History), 'Data Output', 'Сообщения' (Messages), and 'Notifications'. The 'Data Output' tab is active, showing a table with 6 columns: resume\_id, title, education, full\_name, and email. The table contains 5 rows of data.

	resume_id integer	title character varying (100)	education text	full_name text	email character varying (50)
1	1	Разработчик Python	МФТИ	Петров Иван Александрович	ivan.petrov@mail.com
2	2	Аналитик данных	ВШЭ	Сидорова Мария Ивановна	maria.sidorova@mail.com
3	3	Менеджер по продажам	РАНХиГС	Кузнецов Алексей Олегович	alex.kuznetsov@mail.com
4	4	Логист	СПбГУ	Морозова Ольга Петровна	olga.morozova@mail.com
5	5	Графический дизайнер	Британка	Васильев Дмитрий Сергеевич	dmitry.vasilyev@mail.com

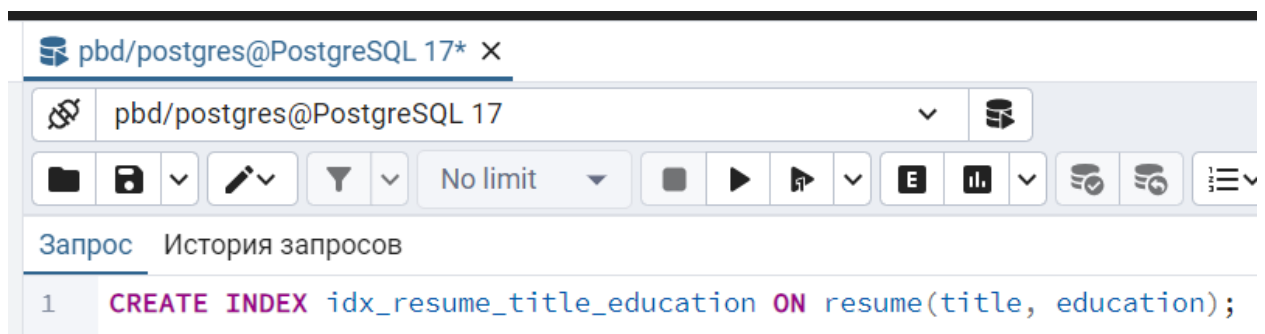
### Индексы

Создадим idx\_resume\_title\_education для таблицы:

- resume

по атрибутам:

- ✓ title
- ✓ education



The screenshot shows a PostgreSQL client window titled 'pbd/postgres@PostgreSQL 17\*'. The query bar contains 'CREATE INDEX idx\_resume\_title\_education ON resume(title, education);'. Below the query bar, there are tabs for 'Запрос' (Query), 'История запросов' (Query History), and 'Data Output'. The 'Запрос' tab is active, showing the SQL query.

	SQL
1	CREATE INDEX idx_resume_title_education ON resume(title, education);

**Назначение индекса:** ускорение поиска резюме по специальности и месту учебы

**Пример использования:**

kbd/postgres@PostgreSQL 17\* x

kbd/postgres@PostgreSQL 17

Запрос История запросов

```

1 SELECT *
2 FROM resume_user_view
3 WHERE title ILIKE '%Разработчик Python%' AND education ILIKE '%МФТИ%';

```

Data Output Сообщения Notifications

	resume_id integer	title character varying (100)	education text	full_name text	email character varying (50)
1	1	Разработчик Python	МФТИ	Петров Иван Александрович	ivan.petrov@mail.com

## Query plan:

```

EXPLAIN ANALYZE
SELECT *
FROM resume_user_view
WHERE title = 'Разработчик Python' AND education = 'МФТИ';

```

Output Сообщения Notifications

QUERY PLAN	
text	
Nested Loop (cost=0.28..16.55 rows=1 width=404) (actual time=0.070..0.071 rows=1 loops=1)	
-> Index Scan using idx_resume_title_education on resume (cost=0.13..8.15 rows=1 width=258) (actual time=0.061..0.062 rows=1 loop...	
Index Cond: (((title)::text = 'Разработчик Python'::text) AND (education = 'МФТИ'::text))	
-> Index Scan using user_pkey on "user" (cost=0.14..8.16 rows=1 width=476) (actual time=0.006..0.006 rows=1 loops=1)	
Index Cond: (id = resume.user_id)	
Planning Time: 0.153 ms	
Execution Time: 0.087 ms	

Создадим idx\_vacancy\_position\_salary для таблицы:

- vacancy

по атрибутам:

- ✓ title
- ✓ salary\_level

kbd/postgres@PostgreSQL 17\* x

kbd/postgres@PostgreSQL 17

Запрос История запросов

```

1 CREATE INDEX idx_vacancy_title_salarylevel ON vacancy(title, salary_level);

```

### Пример использования:

**Query plan:**

Output   Сообщения   Notifications

**Вывод:** в ходе лабораторной работы сделал представление для удобного доступа к контактным данным нужных кандидатов; создал индексы, позволяющие эффективно фильтровать данные по часто используемым пользователям полям; убедился в использовании индексов с помощью вывода Query Plan выборки.