



**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**МИРЭА – Российский технологический университет**  
**РТУ МИРЭА**  
**Колледж приборостроения и кибербезопасности**

**Практическая работа №9**  
по учебной дисциплине МДК.11.01  
**Технология разработки и защиты баз данных**  
специальность 09.02.07 Информационные системы и программирование  
**Тема: Создание и управление индексами**

выполнил  
студент группы ПКС-31  
Лопатин Л.В.  
преподаватель  
Понеделко Е. В.

Москва

2025

**Цель работы:** Формирование навыков работы в среде СУБД PostgreSQL по созданию индексов.

### Ход работы.

**Задание 1.** Рассмотрели примеры по созданию индексов.

**Задание 2.**

2.1 Определил индексы для таблиц БД “Сессия” (см. рисунок 1).

```
session=# \d students
      Table "public.students"
  Column |          Type          | Collation | Nullable |          Default
-----|-----|-----|-----|-----
id       | smallint               |           | not null | nextval('students_id_seq'::regclass)
name     | character varying(50)  |           |          |
surname  | character varying(50)  |           |          |
parentname | character varying(50) |           |          |
groupnum | character varying(15)  |           |          |
phonenum | character varying(15)  |           |          |
address  | text                   |           |          |
dateofbirth | date                  |           | not null |
Indexes:
    "students_pkey" PRIMARY KEY, btree (id)
    "name_inx" btree (name)
Check constraints:
    "name" CHECK (name IS NOT NULL)
Referenced by:
    TABLE "rating" CONSTRAINT "fk_reyting" FOREIGN KEY (idstudent) REFERENCES students(id)
    TABLE "summary" CONSTRAINT "summary_idstudent_fkey" FOREIGN KEY (idstudent) REFERENCES students(id)
Triggers:
    tr_cascade BEFORE DELETE ON students FOR EACH ROW EXECUTE FUNCTION trigger_cascade()

session=#
```

Рис. 1 Информация о таблице студентов

2.2 Включил секундомер(см. рисунок 2)

```
session=# \timing on
Timing is on.
session=#
```

Рис. 2 Включение секундомера

2.3 Выполнил запрос для проверки влияния индексов на заполненную таблицу(см. рисунок 3 и 4)

Data Output Messages Notifications								
	id [PK] smallint	name character varying (50)	surname character varying (50)	parentname character varying (50)	groupnum character varying (15)	phonenum character varying (15)	address text	dateofbirth date
4161	10778	Jayne	Hand	Jamie	4	906-56-86-551	458 Norbert Unions	1947-02-08
4162	10779	Kurt	Schimmel	River	5	991-10-69-491	887 April Views	2006-11-26
4163	10780	Vinnie	Powlowski	Taylor	1	901-87-67-839	007 Durgan Loop	1974-05-03
4164	10781	Xavier	Bode	Bailey	4	992-03-00-428	3363 Rasheed Hollow	1994-03-30
4165	10782	Dorothea	Abshire	Rowan	5	995-05-88-349	085 Jean Centets	2004-03-24
4166	10783	Bessie	Bailey	Emerson	5	979-27-36-557	29369 Macl Point	1953-12-02
4167	10784	Celestine	Koepp	Shawn	1	998-07-08-418	1863 Leonor Camp	1948-09-16
4168	10785	Blair	Wintheiser	Drew	3	984-83-89-440	6488 Marvin Ferry	1950-12-12
4169	10786	Eddie	Bode	Kennedy	4	994-23-20-166	814 Zieme Trail	1978-10-18
4170	10787	Gladys	Stiedemann	Kendall	4	908-99-64-367	92101 Wilza Shore	
4171	10788	Marlee	Morar	Parker	3	976-96-72-624	8291 Runolfsdottir S	
4172	10789	Juvenal	Carter	Logan	4	929-06-84-372	310 Oberbrunner M	

Total rows: 5000 of 10000 Query complete 00:00:00.150 Ln 1, Col 1

Рис. 3 Таблица студентов

```

session=# SELECT count(*) FROM students WHERE name = 'Tamara';
count
-----
      5
(1 row)

Time: 1,236 ms
session=#

```

Рис. 4 Время выполнения запроса без индекса

## 2.4 Создал индекс по столбцу name(см. рисунок 5 и 6)

```

Query  Query History
1      CREATE INDEX name_inx ON students(name);

Data Output  Messages  Notifications
CREATE INDEX

Query returned successfully in 156 msec.

```

Рис. 5 Создание индекса

```

Indexes:
  "students_pkey" PRIMARY KEY, btree (id)
  "name_inx" btree (name)

```

Рис. 6 Описание индексов

## 2.5 Выполнил повторную выборку

```
session=# SELECT count(*) FROM students WHERE name = 'Tamara';
count
-----
      5
(1 row)

Time: 0,678 ms
session=#
```

Рис. 7 Результат влияния индекса

### Задание 3.

3.1 Создал простой индекс для наименований дисциплин. Результат представлен на рисунках 8, 9, 10.

```
session=# SELECT count(*) FROM subject WHERE name = 'Физика';
count
-----
      0
(1 row)

Time: 0,655 ms
```

Рис. 8 Результат до создания индекса

Query	Query History
1	<b>CREATE INDEX</b> subj_inx <b>ON</b> subject(name);

Data Output	Messages	Notifications
CREATE INDEX		
Query returned successfully in 117 msec.		

Рис. 9 Создание простого индекса

```
session=# SELECT count(*) FROM subject WHERE name = 'физика';
count
-----
      0
(1 row)

Time: 0,523 ms
session=#
```

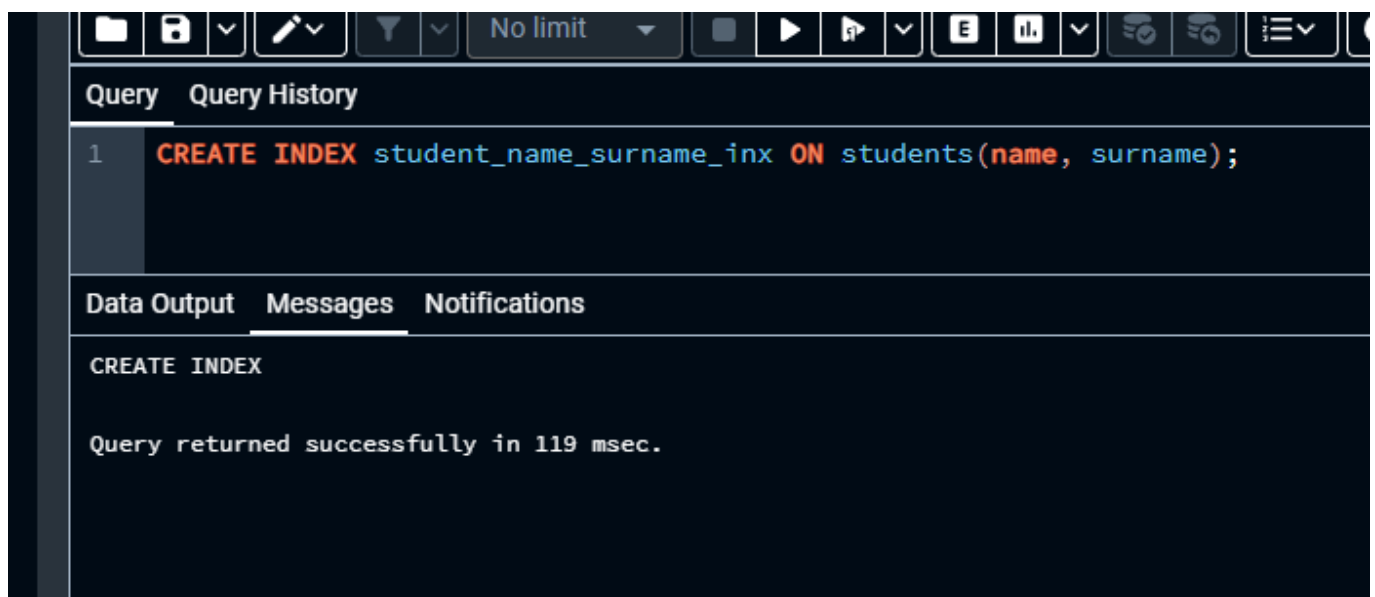
Рис. 10 Результат после создания индекса

3.2 Создал составной индекс для фамилии и имени студента. Результат представлен на рисунках 11, 12, 13.

```
session=# SELECT COUNT(*) FROM students WHERE name = 'Guy' AND surname = 'McGlynn';
count
-----
      1
(1 row)

Time: 0,668 ms
```

Рис. 11 Результат до создания индекса



The screenshot shows a database client window with a toolbar at the top. The 'Query' tab is active, displaying the command: `CREATE INDEX student_name_surname_inx ON students(name, surname);`. Below the query editor, the 'Data Output' tab is selected, showing the message: `CREATE INDEX` and `Query returned successfully in 119 msec.`

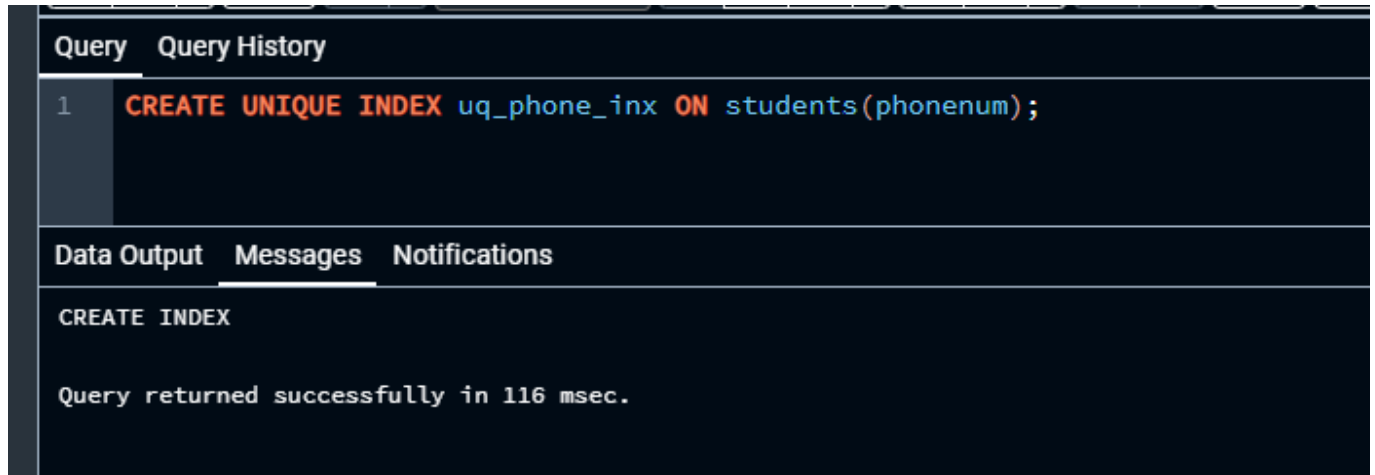
Рис. 12 Создание составного индекса

```
session=# SELECT COUNT(*) FROM students WHERE name = 'Guy' AND surname = 'McGlynn';
count
-----
      1
(1 row)

Time: 0,567 ms
session=#
```

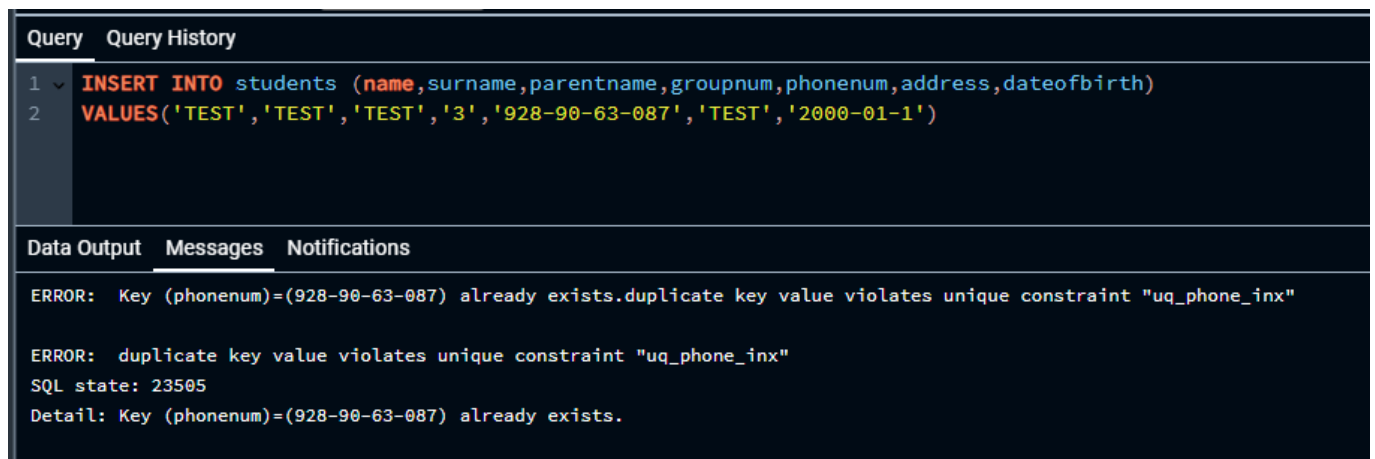
Рис. 13 Результат после создания индекса

3.3 Создал уникальный индекс для телефона студента Результат представлен на рисунках 14, 15.



The screenshot shows a database query window with two tabs: "Query" and "Query History". The "Query" tab is active, displaying a single SQL statement: `CREATE UNIQUE INDEX uq_phone_inx ON students(phonenum);`. Below the query, there are three tabs: "Data Output", "Messages", and "Notifications". The "Messages" tab is active, showing the output: `CREATE INDEX` and `Query returned successfully in 116 msec.`

Рис. 14 Создание уникального индекса



The screenshot shows a database query window with two tabs: "Query" and "Query History". The "Query" tab is active, displaying two SQL statements: `INSERT INTO students (name,surname,parentname,grouppnum,phonenum,address,dateofbirth)` and `VALUES('TEST','TEST','TEST','3','928-90-63-087','TEST','2000-01-1')`. Below the query, there are three tabs: "Data Output", "Messages", and "Notifications". The "Messages" tab is active, showing an error message: `ERROR: Key (phonenum)=(928-90-63-087) already exists.duplicate key value violates unique constraint "uq_phone_inx"`. Below this, it says `ERROR: duplicate key value violates unique constraint "uq_phone_inx"`, `SQL state: 23505`, and `Detail: Key (phonenum)=(928-90-63-087) already exists.`

Рис. 15 Проверка уникального индекса

Вывод: в ходе выполнения практической работы сформировали навыки работы в среде СУБД PostgreSQL по созданию индексов.