МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет безопасности информационных технологий**

**Направление подготовки: 10.03.01 Информационная безопасность**

**Образовательная программа: "Информационная безопасность / Information security"**

**Дисциплина:**

**«*Информационная безопасность баз данных*»**

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

***«Шифрование в базах данных»***


**Выполнил студент:**

N3246 / ИББД N23 1.3

Суханкулиев Мухаммет /_____

*ФИО* *Подпись*

**Проверила:**

Карманова Наталия Андреевна /_____

*ФИО* *Подпись*


_____

*Отметка о выполнении (один из вариантов: отлично, хорошо, удовлетворительно, зачтено)*

_____

*Дата*

Санкт-Петербург

2025 г.

# ВВЕДЕНИЕ

**Цель работы** – изучить и практически применить методы хеширования паролей и шифрования данных в PostgreSQL с использованием расширения pgcrypto.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Создать таблицу, в которой два столбца содержат хешированные значения, где одно из них сгенерировано с помощью алгоритма SHA-1. Показать, как можно выполнить проверку, используя данные двух хешей.
2. Создать таблицу, в которой данные имеют байтовый тип. Зашифровать этот столбец и показать, как пользователь может расшифровать данные во время обычного select-запроса к зашифрованному столбцу.

# 1 ШИФРОВАНИЕ В БД

Подключение расширения `pgcrypto`.

```
ibbd=# CREATE EXTENSION IF NOT EXISTS pgcrypto;
CREATE EXTENSION
```

```
ibbd=# \dx
                Список установленных расширений
   Имя    | Версия |   Схема    |           Описание
----------+--------+------------+------------------------------
 pgcrypto | 1.3    | public     | cryptographic functions
 plpgsql  | 1.0    | pg_catalog | PL/pgSQL procedural language
(2 строки)
```

## 1.1 Таблица 1

```
CREATE TABLE user_hashes (
    username TEXT,
    password_crypt TEXT,
    password_sha1 BYTEA
);

CREATE TABLE Query returned successfully in 56 msec.
```

```
INSERT INTO user_hashes(username, password_crypt, password_sha1)
VALUES (
    'user1',
    crypt('mysecurepassword', gen_salt('bf')),
    digest('mysecurepassword', 'sha1')
);

INSERT 0 1 Query returned successfully in 53 msec.
```

Проверка пароля

```
SELECT username
FROM user_hashes
WHERE username = 'user1'
  AND password_crypt = crypt('mysecurepassword', password_crypt);

username
----------
 user1
(1 строка)
```

```
SELECT username
FROM user_hashes
WHERE username = 'user1'
  AND password_sha1 = digest('mysecurepassword', 'sha1');

username
----------
 user1
(1 строка)
```

**В боевых условиях crypt() безопаснее, так как она использует адаптивные алгоритмы и соль.*

### 1.2 Таблица 2

```
CREATE TABLE secure_data (
    id SERIAL PRIMARY KEY,
    secret_data BYTEA
);

CREATE TABLE Query returned successfully in 71 msec.
```

```
INSERT INTO secure_data(secret_data)
VALUES (
    pgp_sym_encrypt('Sensitive info', 'secretkey123')
);

INSERT 0 1 Query returned successfully in 72 msec.
```

Расшифровка:

```
SELECT id, pgp_sym_decrypt(secret_data, 'secretkey123') AS decrypted_data
FROM secure_data;

id | decrypted_data
----+----------------
  1 | Sensitive info
(1 строка)
```

## ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы были освоены механизмы хеширования и симметричного шифрования в PostgreSQL. Для хеширования паролей использовались функции crypt() и digest(), а для шифрования данных — функции pgp_sym_encrypt() и pgp_sym_decrypt() из расширения pgcrypto.

Полученные знания позволяют обеспечить базовую защиту пользовательских данных в СУБД PostgreSQL.