# Лабораторная работа 4

### Задачи:

- Замаскировать поля с конфиденциальными данными
- Провести анонимизацию данных 3 разными способами

В наше время тот, кто владеет информацией, владеет миром. Поэтому в работе было бы некорректно предоставлять сотрудникам полные данные о всех людях. А если человек попытается зайти в чужой аккаунт? А если отследит нужного человека и приедет к нему домой?

Именно тут и нужна анонимизация.

С помощью расширения Postgresql Anonymizer была выполнена лабораторная работа по анонимизации данных разными способами.

### Была создана таблица с пользователями, в неё была добавлена информация

| | id | name | email | phone | address | age |
|---|---|---|---|---|---|---|
| 1 | 1 | Komarov Niko… | komarov@gmail.… | +7777777… | Russia, Moscow, Red Square 1 | 19 |
| 2 | 2 | Druzhinin Eg… | druzh@gmail.com | +7819502… | Russia, Saint-Petersburg, Vyzem… | 19 |
| 3 | 3 | Viktoria Kus… | vikakuskus@niu… | +7952812… | Russia, Saint-Petersburg, Kronv… | 19 |
| 4 | 4 | Mike de Geof… | mikedegeofroy@… | +7393102… | Russia, Saint-Petersburg, Kronv… | 19 |

### Динамическая маскировка данных:

```
CREATE EXTENSION IF NOT EXISTS anon CASCADE;
SELECT anon.start_dynamic_masking();
CREATE ROLE basic_user LOGIN;
SECURITY LABEL FOR anon ON ROLE basic_user IS 'MASKED';

SECURITY LABEL FOR anon ON COLUMN users.phone
IS 'MASKED WITH FUNCTION anon.partial(phone,2,$$******$$,2)';
 // замаскировали номера телефона
```

При запросе SELECT * FROM users результат выглядит так:

```
postgres=> SELECT * FROM users;
 id |        name        |         email         |   phone    |                  address                  | age
----+--------------------+-----------------------+------------+-------------------------------------------+-----
  1 | Komarov Nikolay    | komarov@gmail.com     | +7******77 | Russia, Moscow, Red Square 1              |  19
  2 | Druzhinin Egor     | druzh@gmail.com       | +7******01 | Russia, Saint-Petersburg, Vyzemsky lane 5-7 |  19
  3 | Viktoria Kuskusova | vikakuskus@niuitmo.ru | +7******52 | Russia, Saint-Petersburg, Kronverksky 49    |  19
  4 | Mike de Geofroy    | mikedegeofroy@mail.ru | +7******92 | Russia, Saint-Petersburg, Kronverksky 49    |  19
(4 rows)
```

Номер телефона действительно скрыт

## Generalization

> 💡 Generalization - заменяет данные более широкими и менее точными значениями, диапазонами.

Предположим, что для аналитики не требуется знать точного возраста человека, а лишь то, в каком интервале находится его возраст. В таком случае можно использовать следующее:

```
CREATE MATERIALIZED VIEW users_generalized AS SELECT
 id,
 name,
 anon.generalize_int4range(age, 5) AS age
 FROM users;
```

Результат:

```
postgres=# select * from users_generalized;
 id |        name        |   age
----+--------------------+---------
  2 | Druzhinin Egor     | [15,20)
  4 | Mike de Geofroy    | [15,20)
  1 | Komarov Nikolay    | [20,25)
  3 | Viktoria Kuskusova | [20,25)
(4 rows)
```

## Faking

Предположим, что мы хотим подменить адреса электронных почт у пользователей. В такой ситуации применим функцию anon.fake_email();

```
CREATE MATERIALIZED VIEW users_faked AS SELECT
 id,
 name,
 anon.fake_email() as email FROM users;
```

Результат:

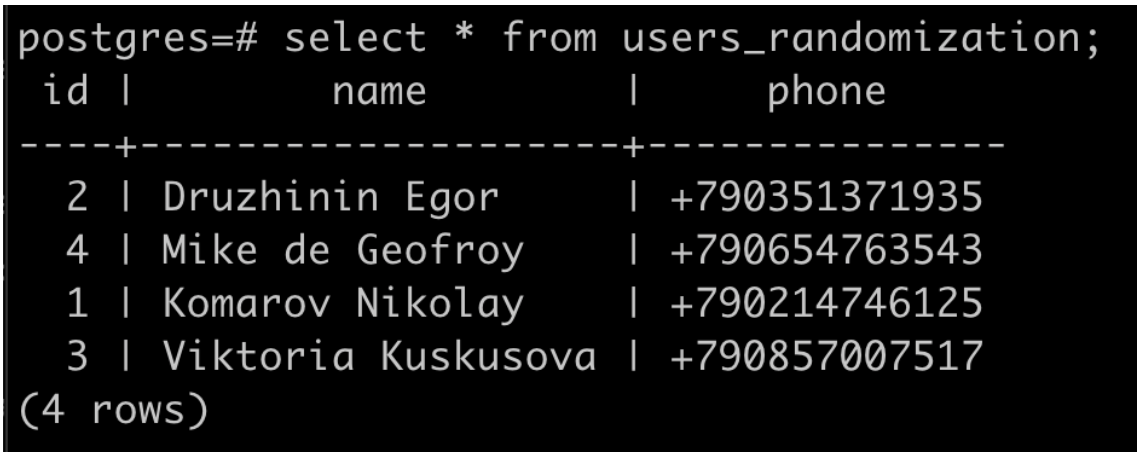
```
postgres=# select * from users_faked;
 id |        name        |           email
----+--------------------+---------------------------
  2 | Druzhinin Egor     | ymcbride@example.net
  4 | Mike de Geofroy    | susanferguson@example.com
  1 | Komarov Nikolay    | jessicalee@example.net
  3 | Viktoria Kuskusova | randy24@example.org
(4 rows)
```

## Randomization

Предположим, что нам нужно анонимизировать номер телефона, в таком случае можем использовать функцию anon.random_phone(p), p - префикс

```
CREATE MATERIALIZED VIEW users_randomization AS SELECT
id,
name,
anon.random_phone('+790') AS phone
FROM users;
```

Результат:




В ходе лабораторной работы применили динамическую маскировку данных, Generalization, Faking и Randomization.