Департамент образования города Москвы

Государственное автономное образовательное учреждение

высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ОТЧЕТ

по дисциплине «Проектный практикум по разработке ETL-решений»

Направление подготовки 38.03.05 – бизнес-информатика

Профиль подготовки «Аналитика данных и эффективное управление»

(очная форма обучения)

Вебинар 14-03-2025

Выполнил:

st-89

Руководитель:

Москва  
2025

**Цель работы:** Заполнить таблицу person в базе данных PostgreSQL фейковыми данными не менее 100 записей и провести анализ этих данных с использованием SQL. Также создать визуализации для полученных результатов.

**Задание 1. Заполнение таблицы и анализ данных в PostgreSQL с визуализацией**

**Ход работы:**

1. Меняем время выполнения дагов на выполнения каждую минуту (*Рис. 1*)

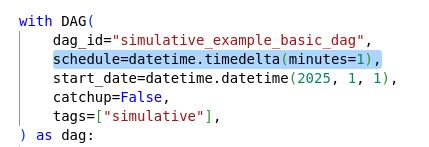


Рис. 1 – Изменения времени

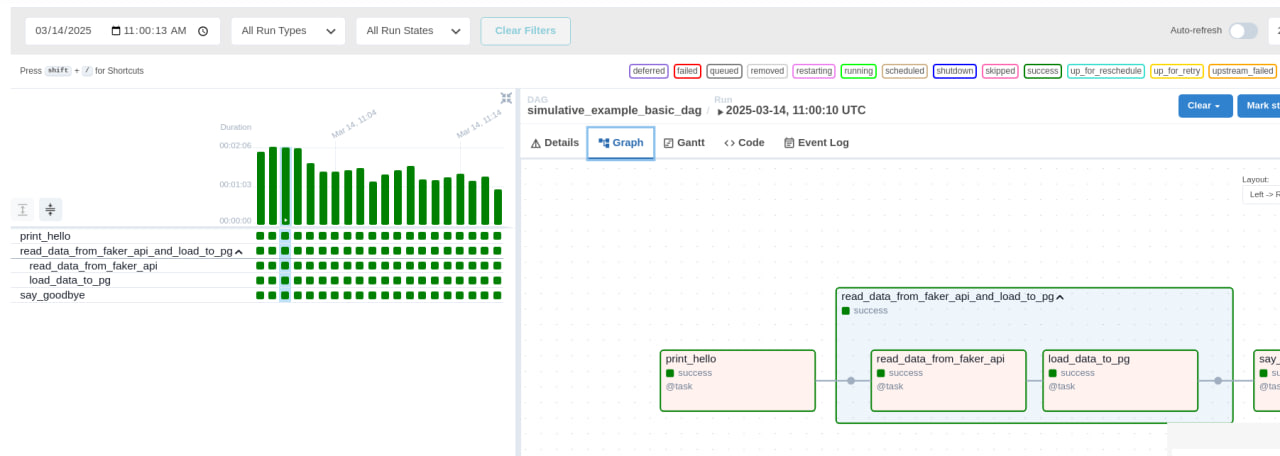


Рис. 2 - Генерация и загрзука данных

Как итог – загрузка тольк 30 записей в связи с высокой нагрузкой на компьютер и время выполнения для 100 записей.

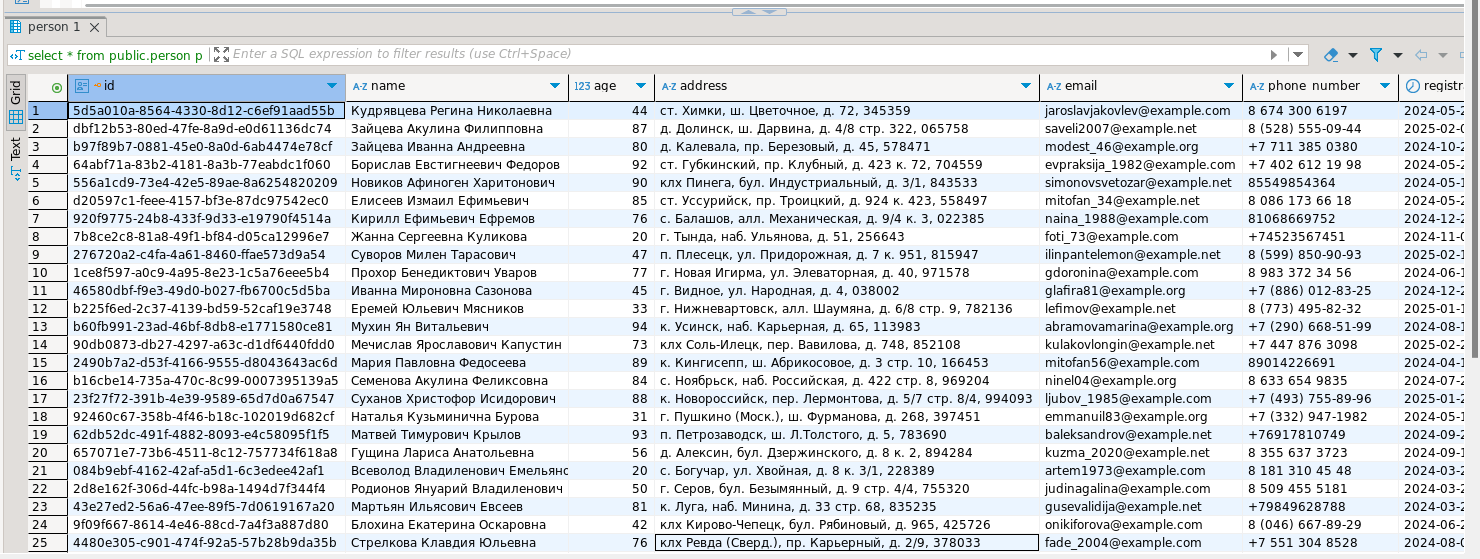


Рис. 3 – Записи в PostgreSQL

С помощью следующих скриптов SQL был проведен анализ данных и выгружен в Yandex DataLens

SELECT

AVG(age) AS avg\_age,

MIN(age) AS min\_age,

MAX(age) AS max\_age

FROM person;

SELECT

TO\_CHAR(registration\_date, 'YYYY-MM') AS month,

COUNT(\*) AS count

FROM person

GROUP BY month

ORDER BY month;

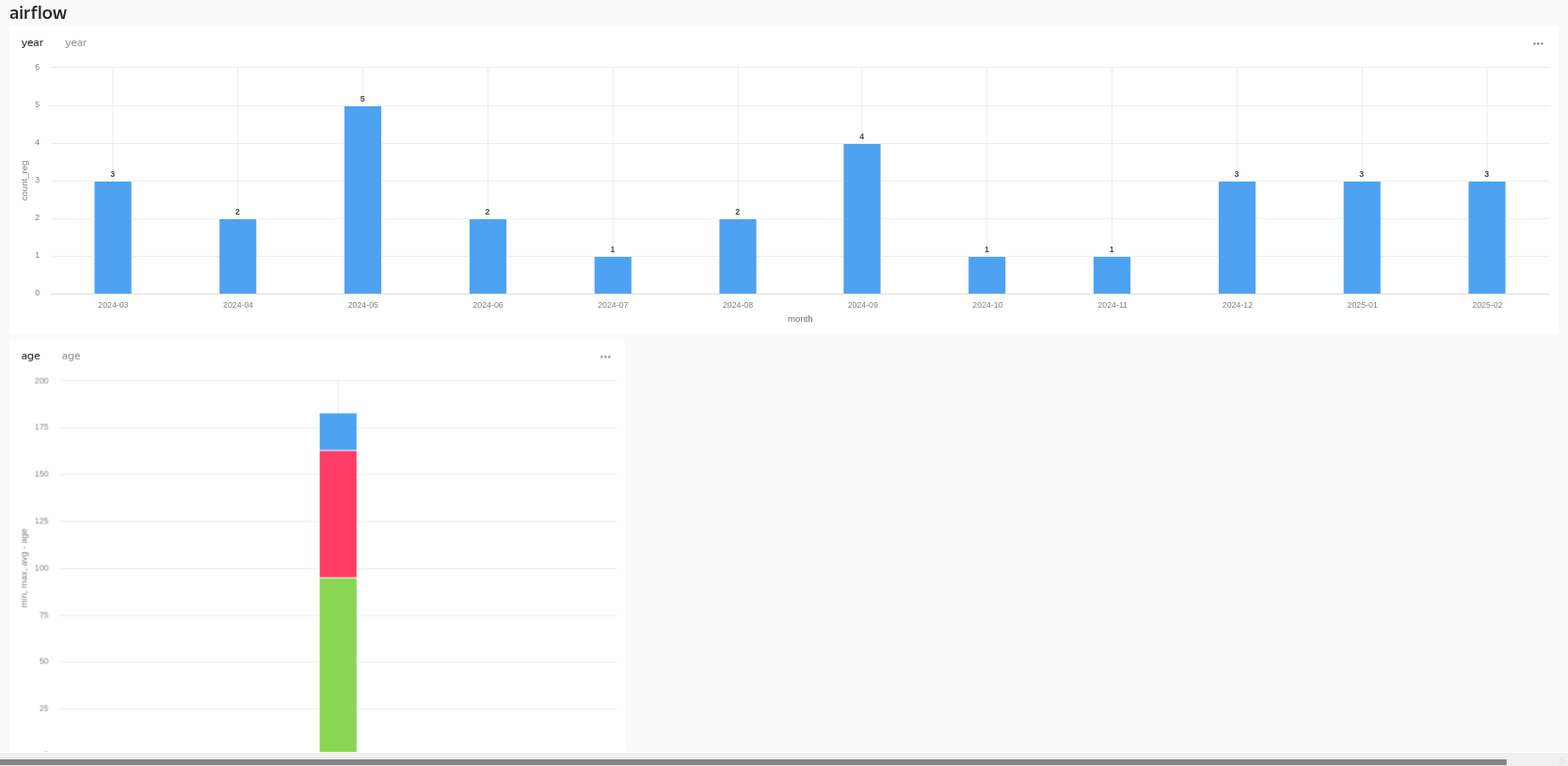


Рис. 4 – Визуализация данных

Ознакомится можно по ссылке <https://datalens.yandex/8b4wq97pqjrgu>

**Вывод:** Была проведена загрузка данных и их визуализация.

**ПРИЛОЖЕНИЕ**

SQL код для создания таблиц

-- Таблица заказов (основная информация о продажах)

CREATE TABLE orders (

row\_id INT PRIMARY KEY,

order\_date DATE,

ship\_date DATE,

ship\_mode VARCHAR(50),

sales DECIMAL(10,2),

quantity INT,

discount DECIMAL(4,2),

profit DECIMAL(10,2),

returned TINYINT(1) DEFAULT 0 -- 1 = Yes, 0 = No

);

-- Таблица клиентов

-- Пересоздаем таблицу customers

DROP TABLE IF EXISTS customers;

CREATE TABLE customers (

id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id VARCHAR(20) NOT NULL,

customer\_name VARCHAR(100),

segment VARCHAR(50),

country VARCHAR(100),

city VARCHAR(100),

state VARCHAR(100),

postal\_code VARCHAR(20),

region VARCHAR(50),

INDEX idx\_customer\_id (customer\_id),

INDEX idx\_region (region)

);

-- Пересоздаем таблицу products

DROP TABLE IF EXISTS products;

CREATE TABLE products (

id INT AUTO\_INCREMENT PRIMARY KEY,

product\_id VARCHAR(20) NOT NULL,

category VARCHAR(50),

sub\_category VARCHAR(50),

product\_name VARCHAR(255),

person VARCHAR(100),

INDEX idx\_product\_id (product\_id), -- Обычный индекс вместо UNIQUE

INDEX idx\_category (category),

INDEX idx\_subcategory (sub\_category)

);