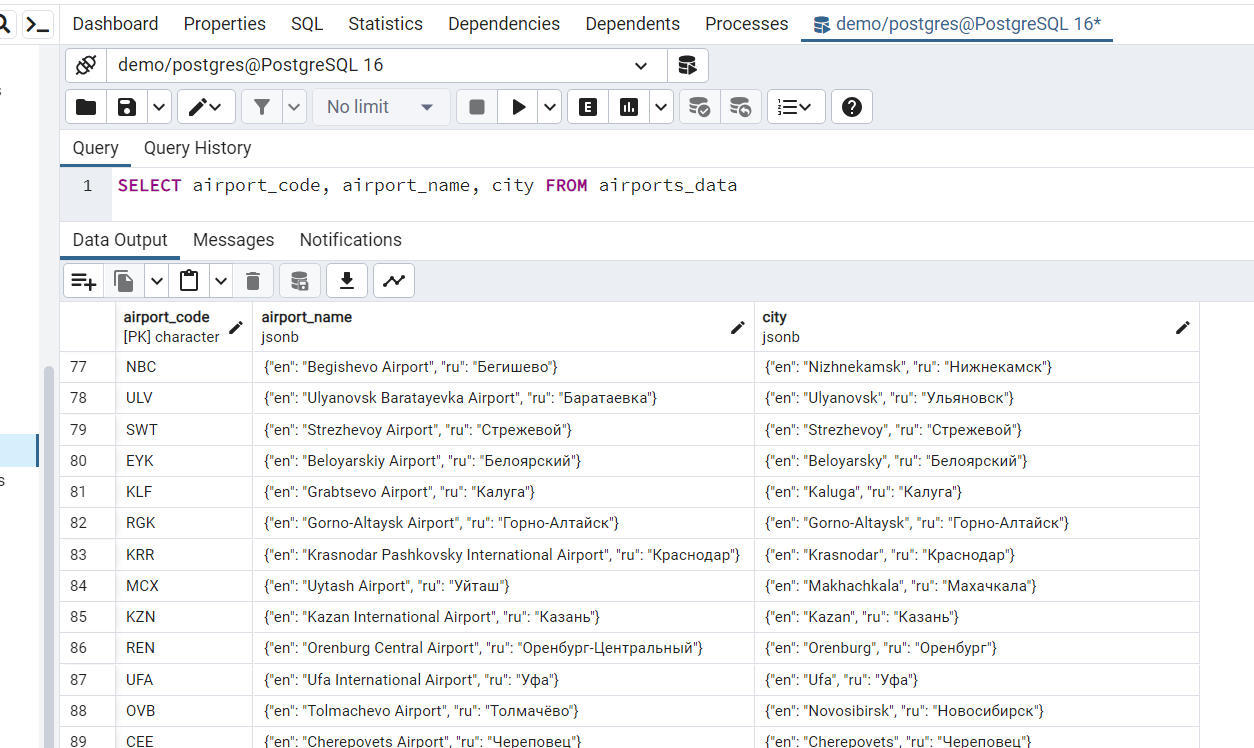
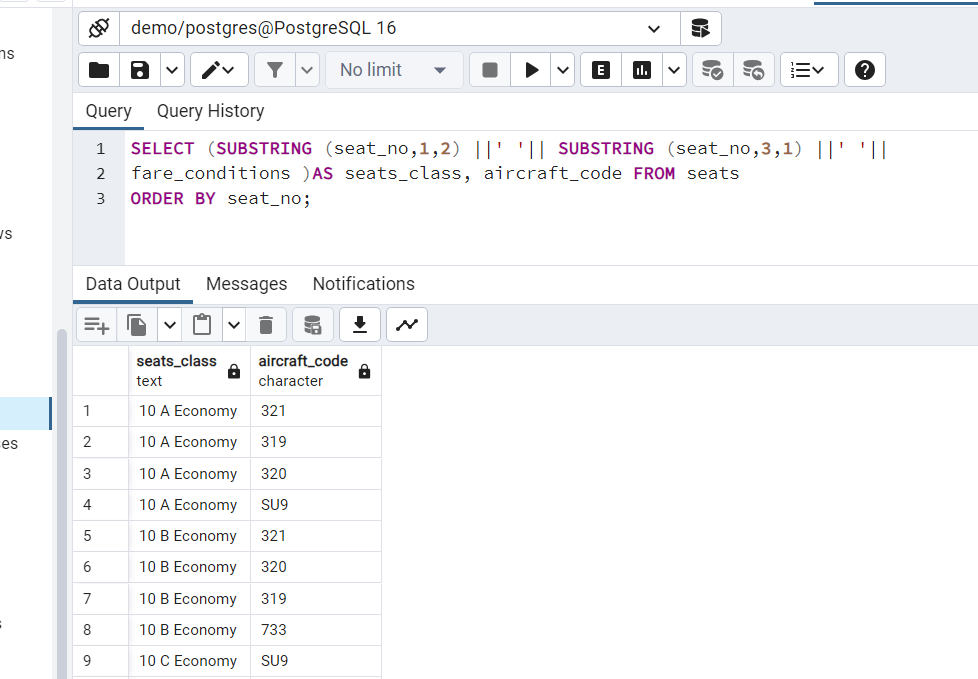
# Задание 1

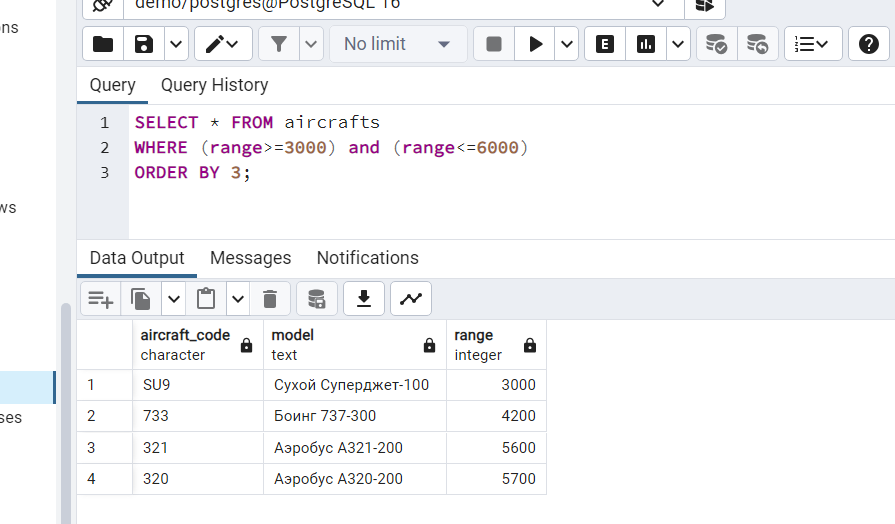
1



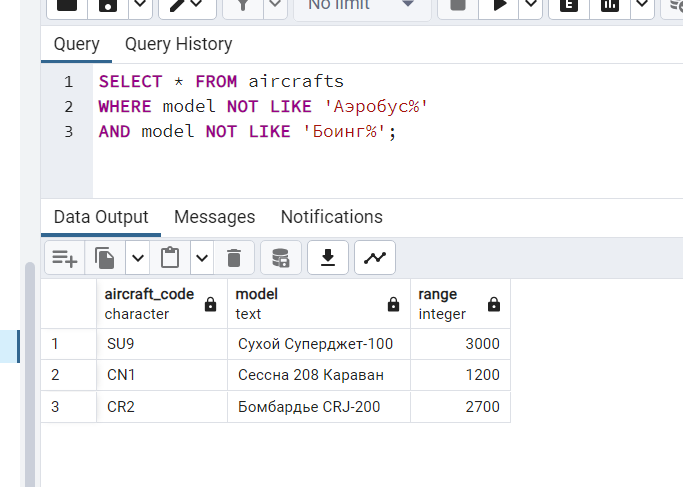
2



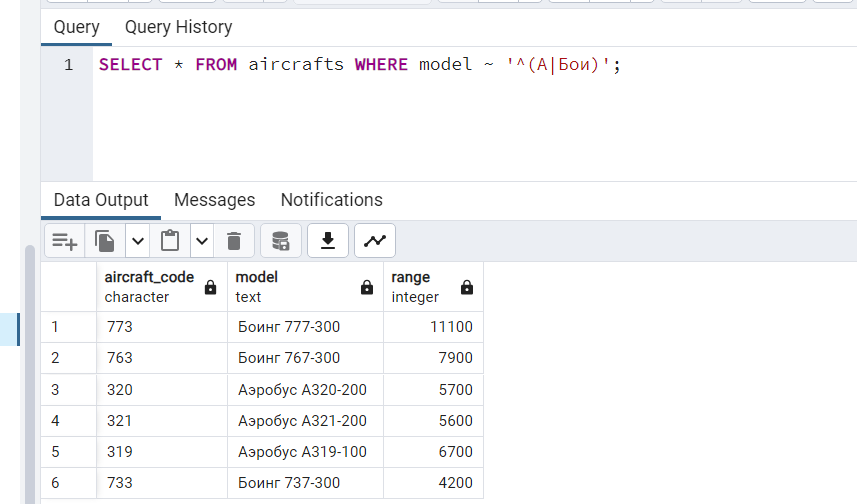
3



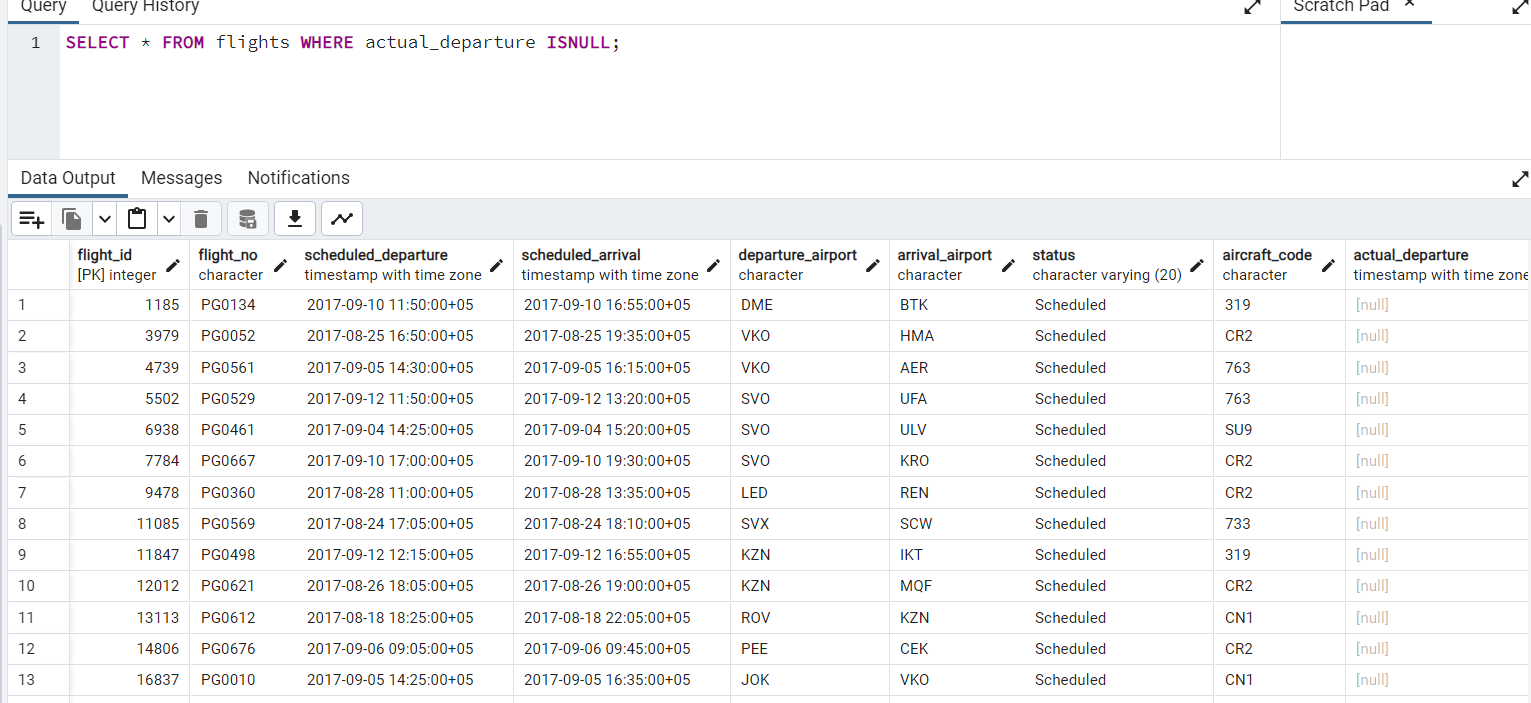
4



5



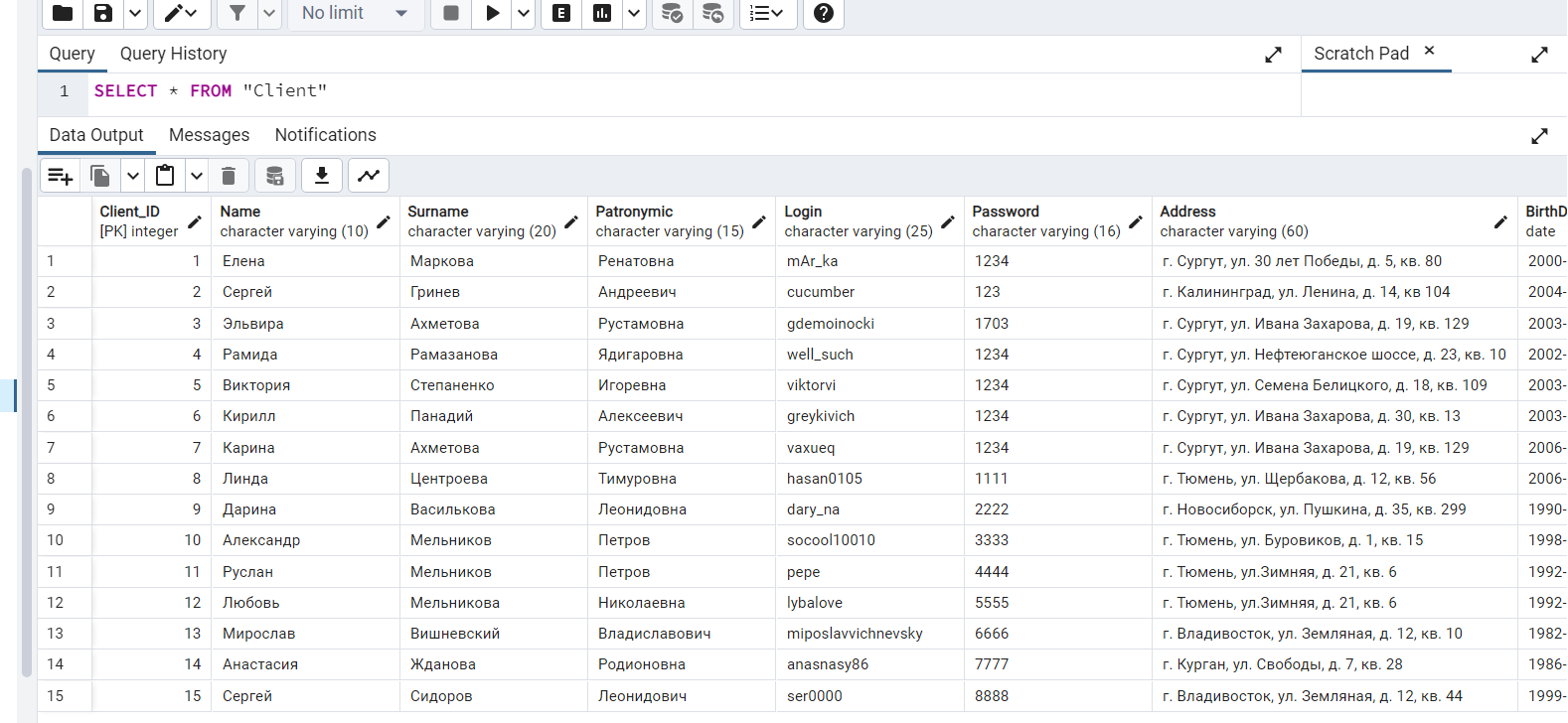
6



# Самостоятельная работа

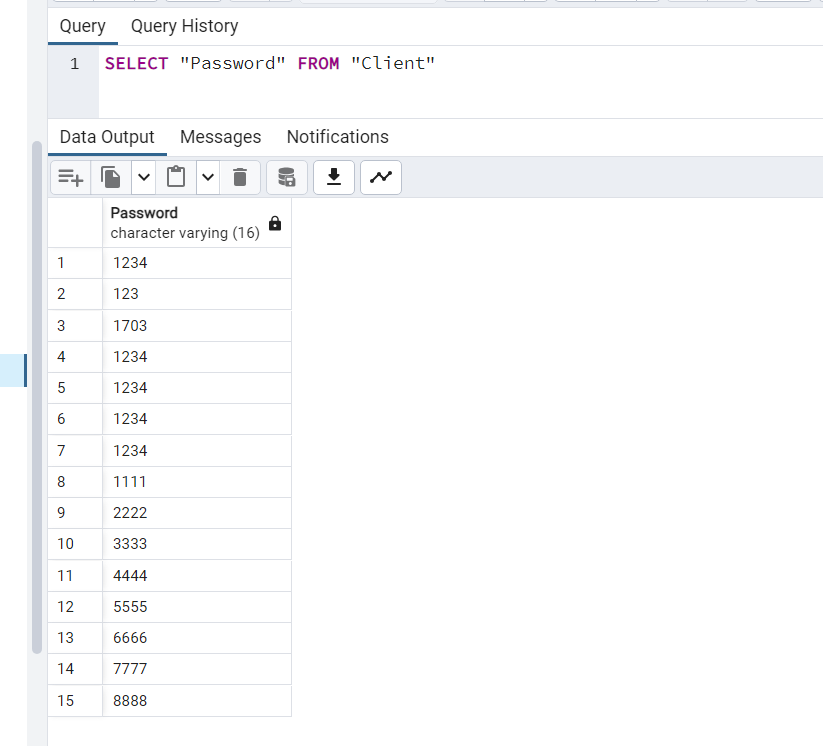
# Запрос на полную выборку данных.

SELECT \* FROM "Client"

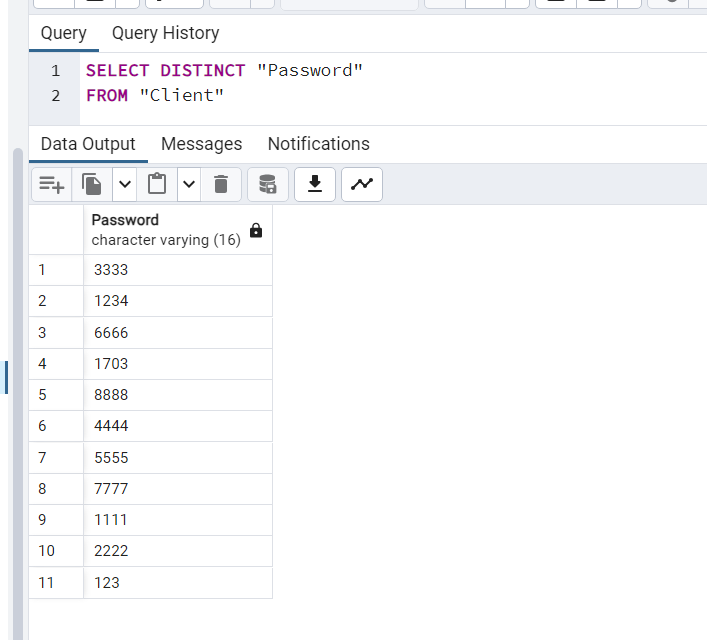


# Запрос на выборку данных без повторений.

Без

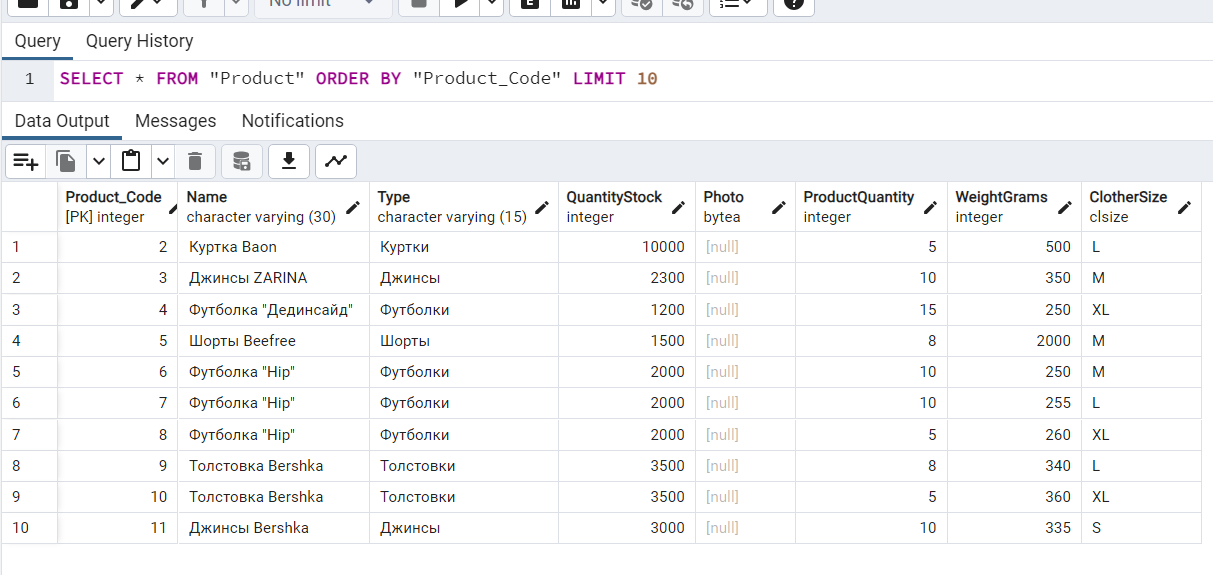


SELECT DISTINCT "Password" FROM "Client"



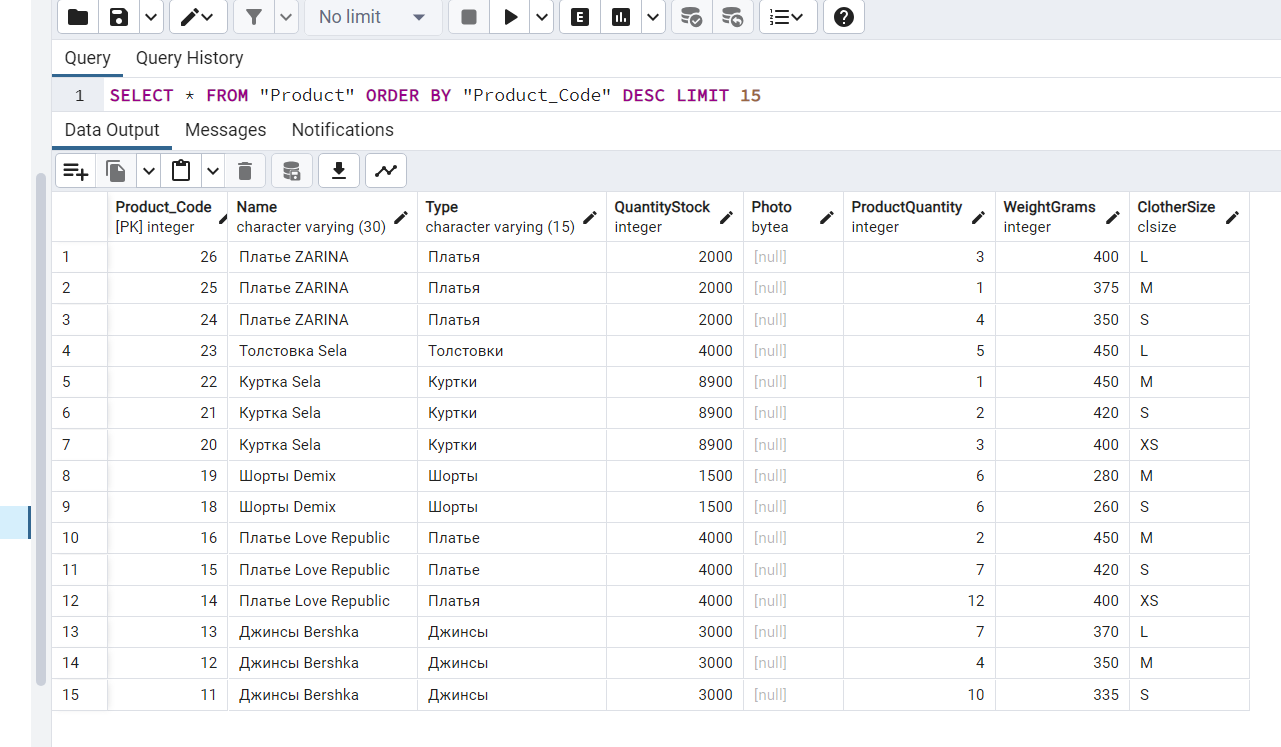
# Запрос на выборку первых 10 записей.

SELECT \* FROM "Product" ORDER BY "Product\_Code" LIMIT 10



# Запрос на выборку последних 15 записей.

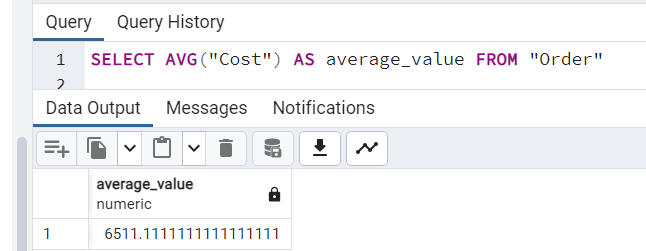
SELECT \* FROM "Product" ORDER BY "Product\_Code" DESC LIMIT 15



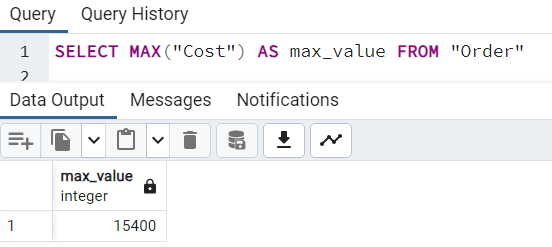
# Запросы на выполнение функций Average, Max, Min

AVG - Эта функция возвращает среднее арифметическое группы значений. Значения NULL она не учитывает.

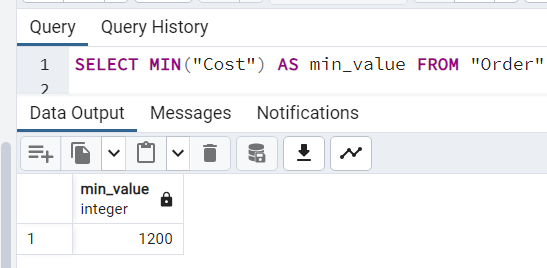
SELECT AVG("Cost") AS average\_value FROM "Order"



SELECT MAX("Cost") AS max\_value FROM "Order"



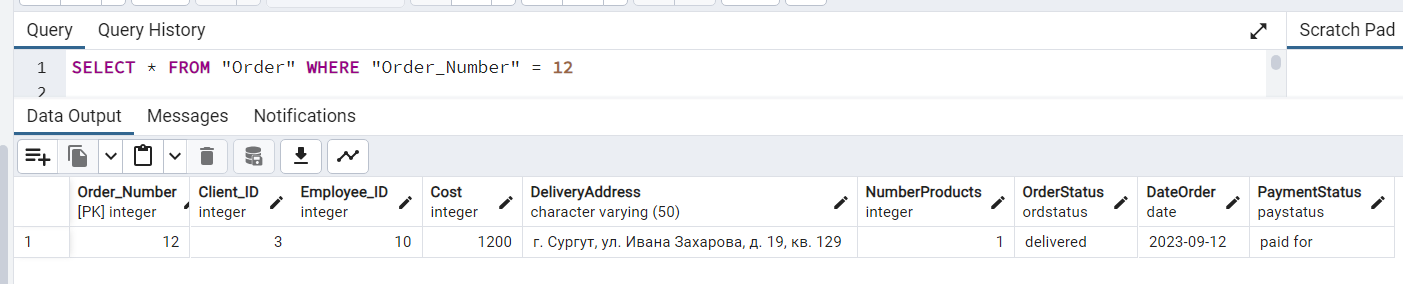
SELECT MIN("Cost") AS min\_value FROM "Order"



# Сконструируйте запросы с использованием оператора Where:

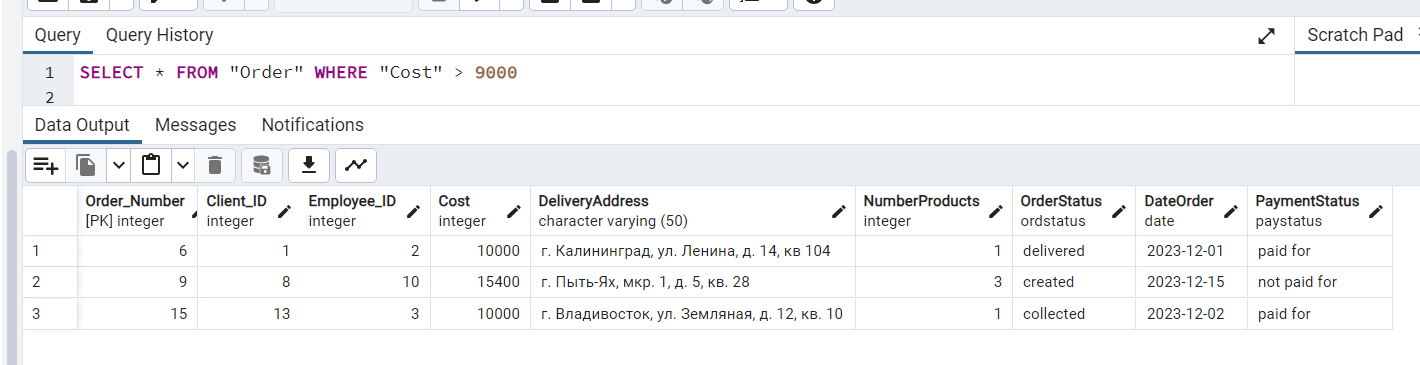
- запрос на возвращение определенного кортежа по первичному ключу;

SELECT \* FROM "Order" WHERE "Order\_Number" = 12

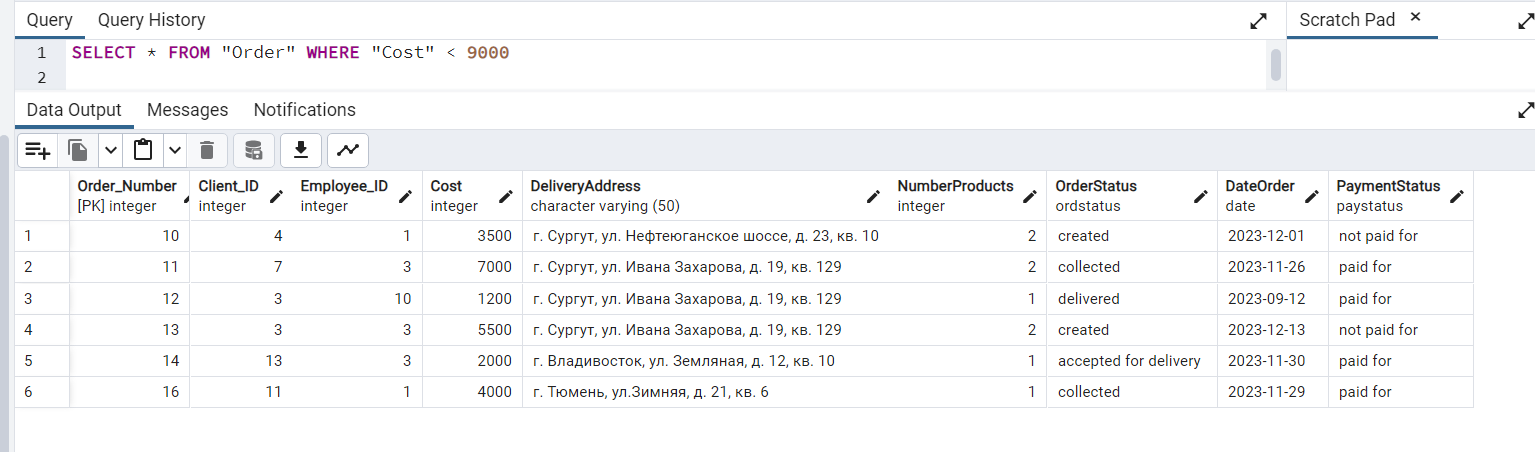


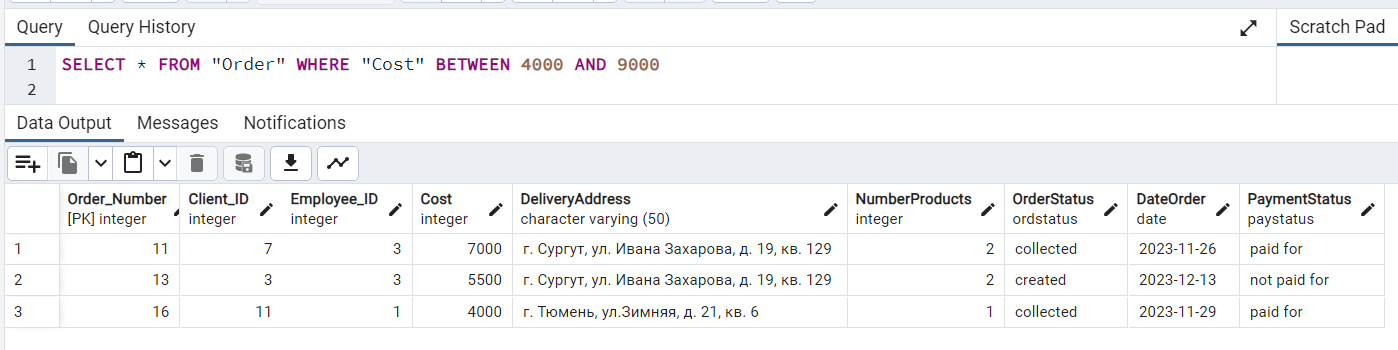
- запросы на возвращение значения по условиям больше, меньше и между;

SELECT \* FROM "Order" WHERE "Cost" > 9000



SELECT \* FROM "Order" WHERE "Cost" < 9000

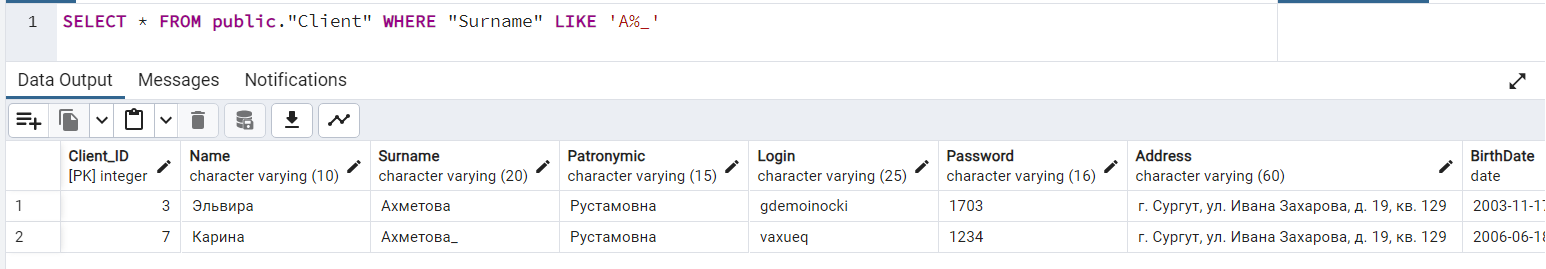


SELECT \* FROM "Order" WHERE "Cost" BETWEEN 4000 AND 9000

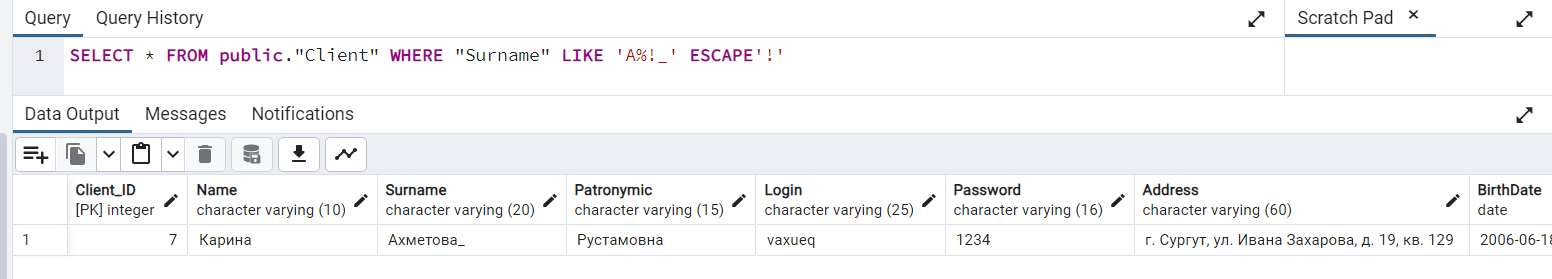
- запросы на возвращении всех кортежей по условию с использованием

оператора LIKE и ESCAPE;

SELECT \* FROM public."Client" WHERE "Surname" LIKE 'А%\_'



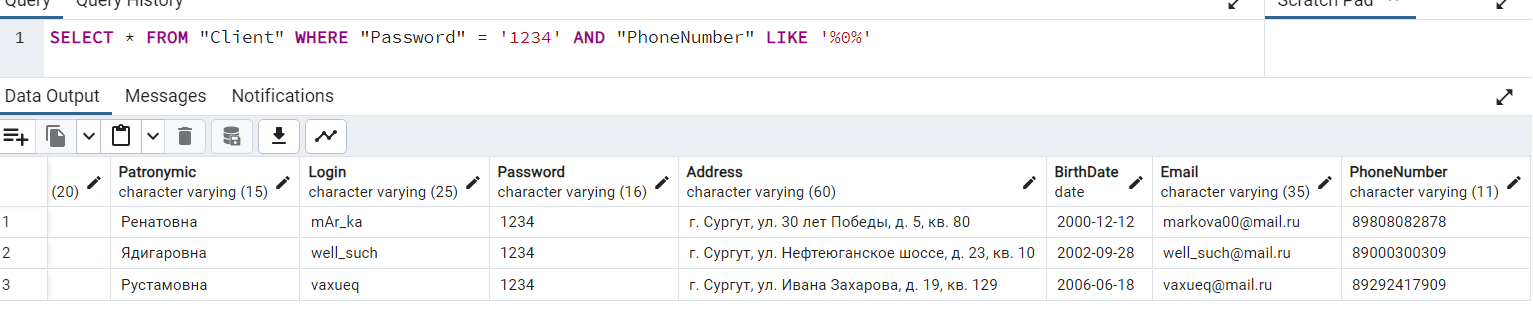
SELECT \* FROM public."Client" WHERE "Surname" LIKE 'А%!\_' ESCAPE'!'



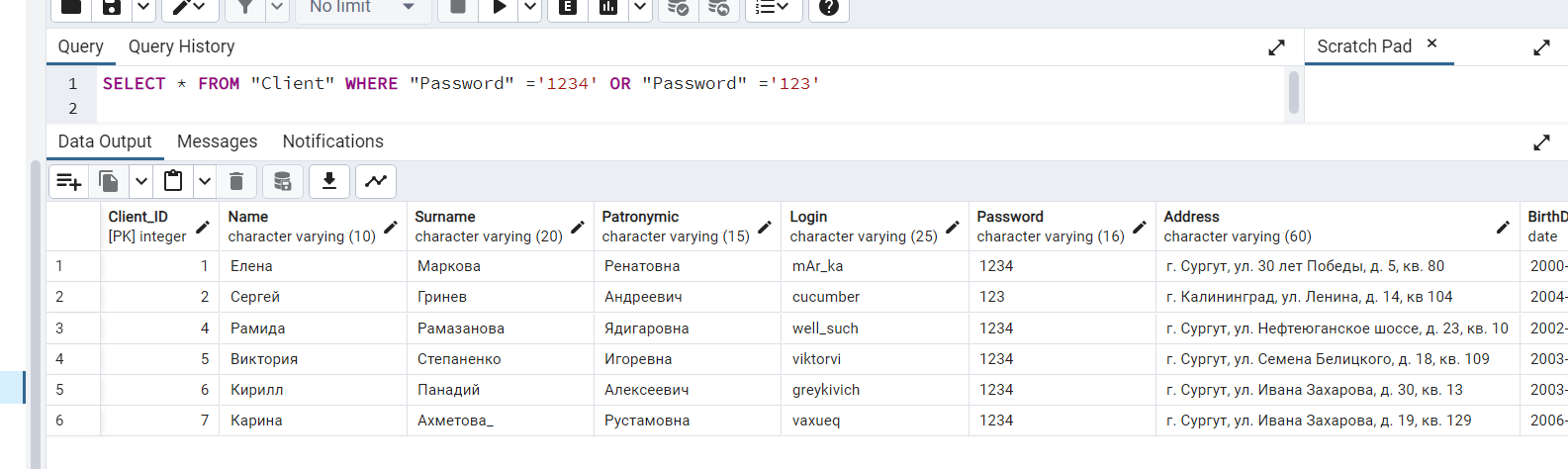
- запрос на возвращение кортежей со сложным условием на основе

логических операторов И, ИЛИ, НЕ, EXISTS;

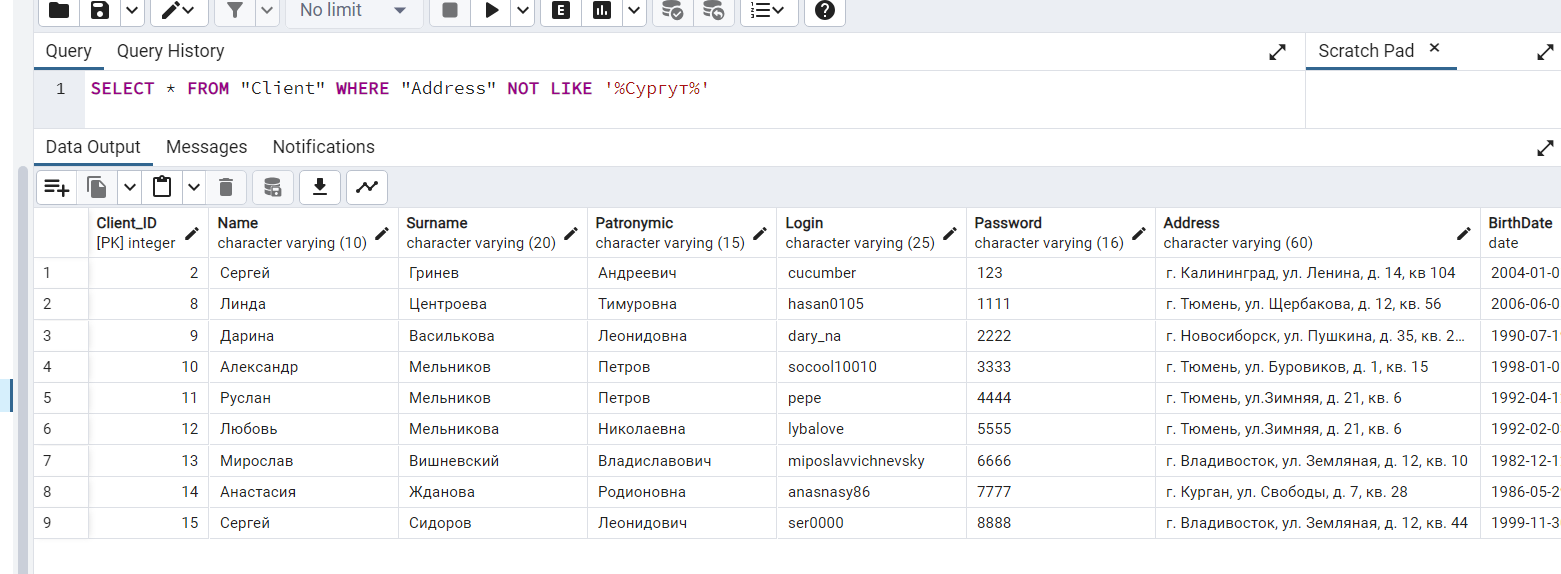
SELECT \* FROM "Client" WHERE "Password" = '1234' AND "PhoneNumber" LIKE '%0%'

****

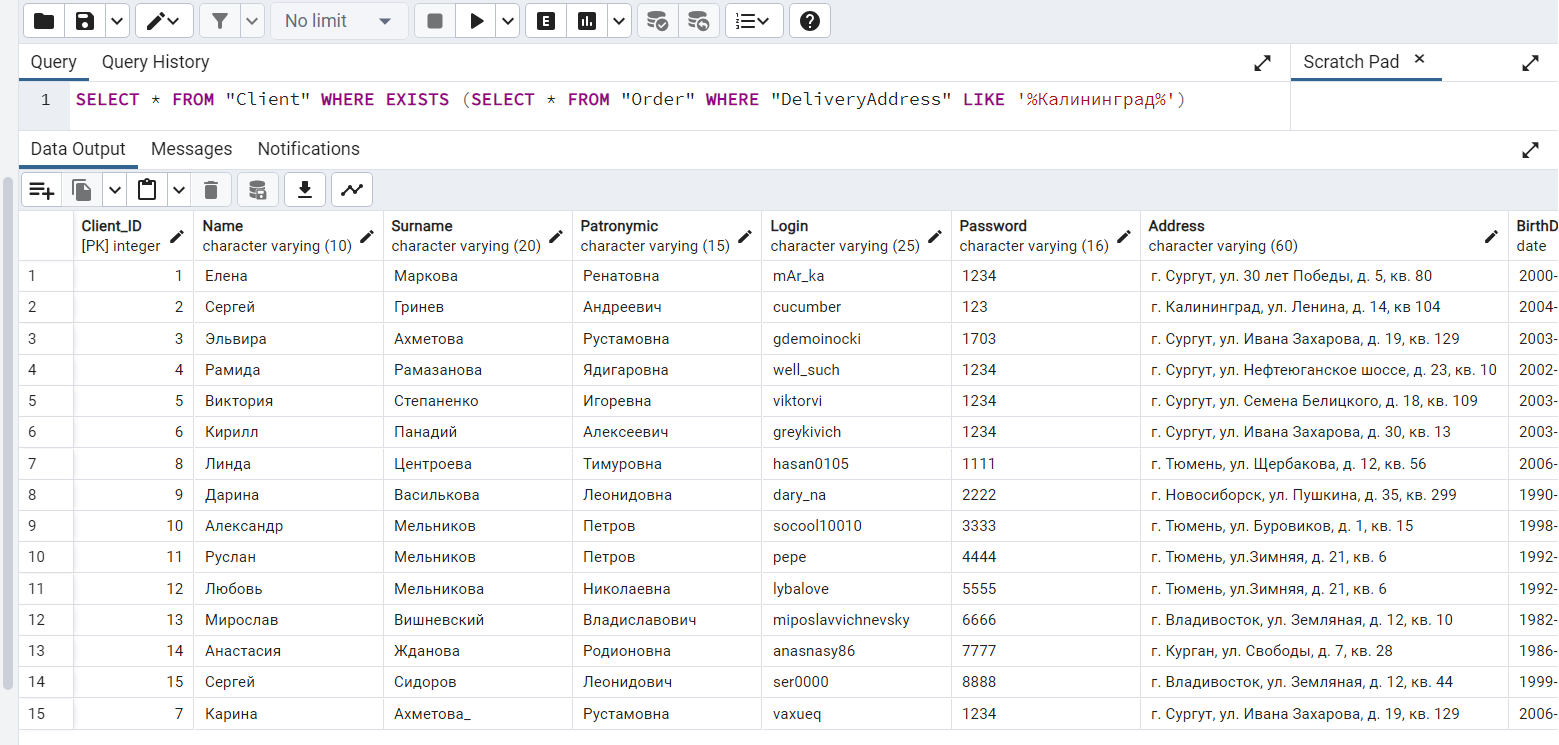
SELECT \* FROM "Client" WHERE "Password" ='1234' OR "Password" ='123'



SELECT \* FROM "Client" WHERE "Address" NOT LIKE '%Сургут%'

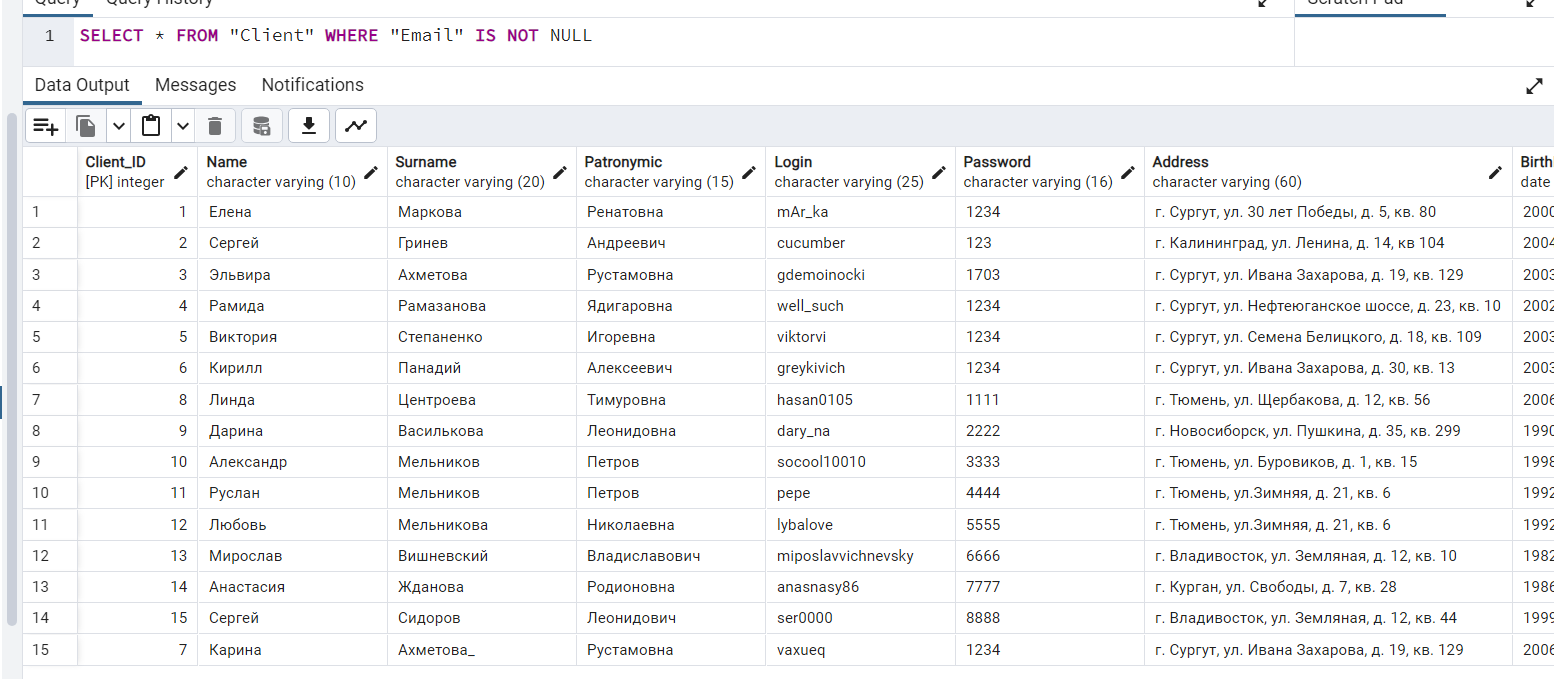


SELECT \* FROM "Client" WHERE EXISTS (SELECT \* FROM "Order" WHERE "DeliveryAddress" LIKE '%Калининград%')



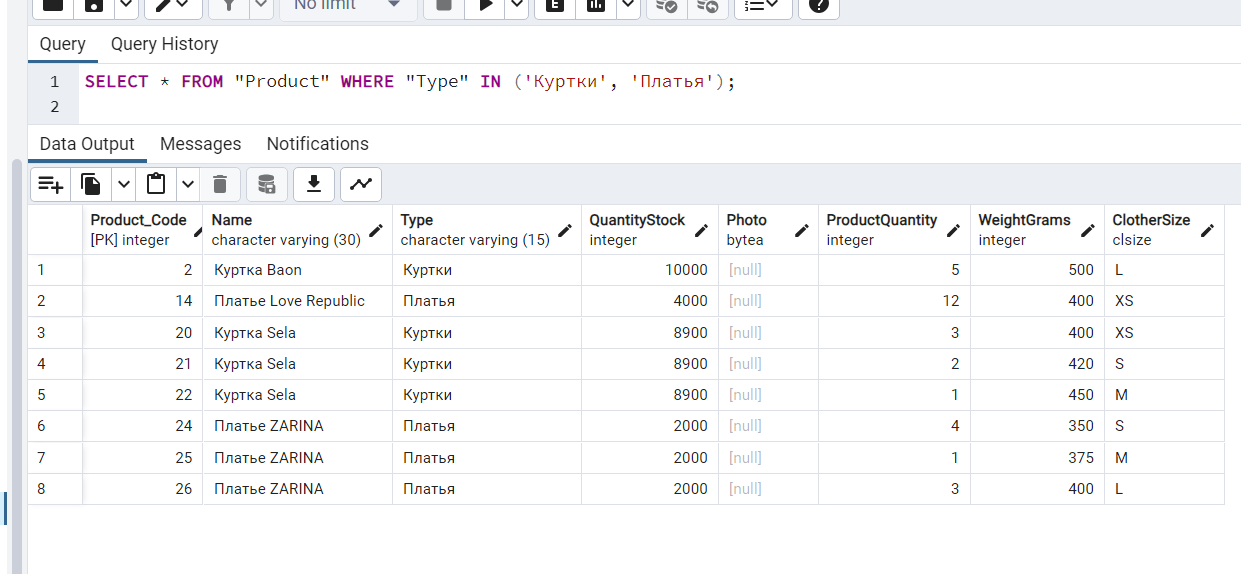
- запрос с использованием оператора NOT NULL в условии отбора.

SELECT \* FROM "Client" WHERE "Email" IS NOT NULL



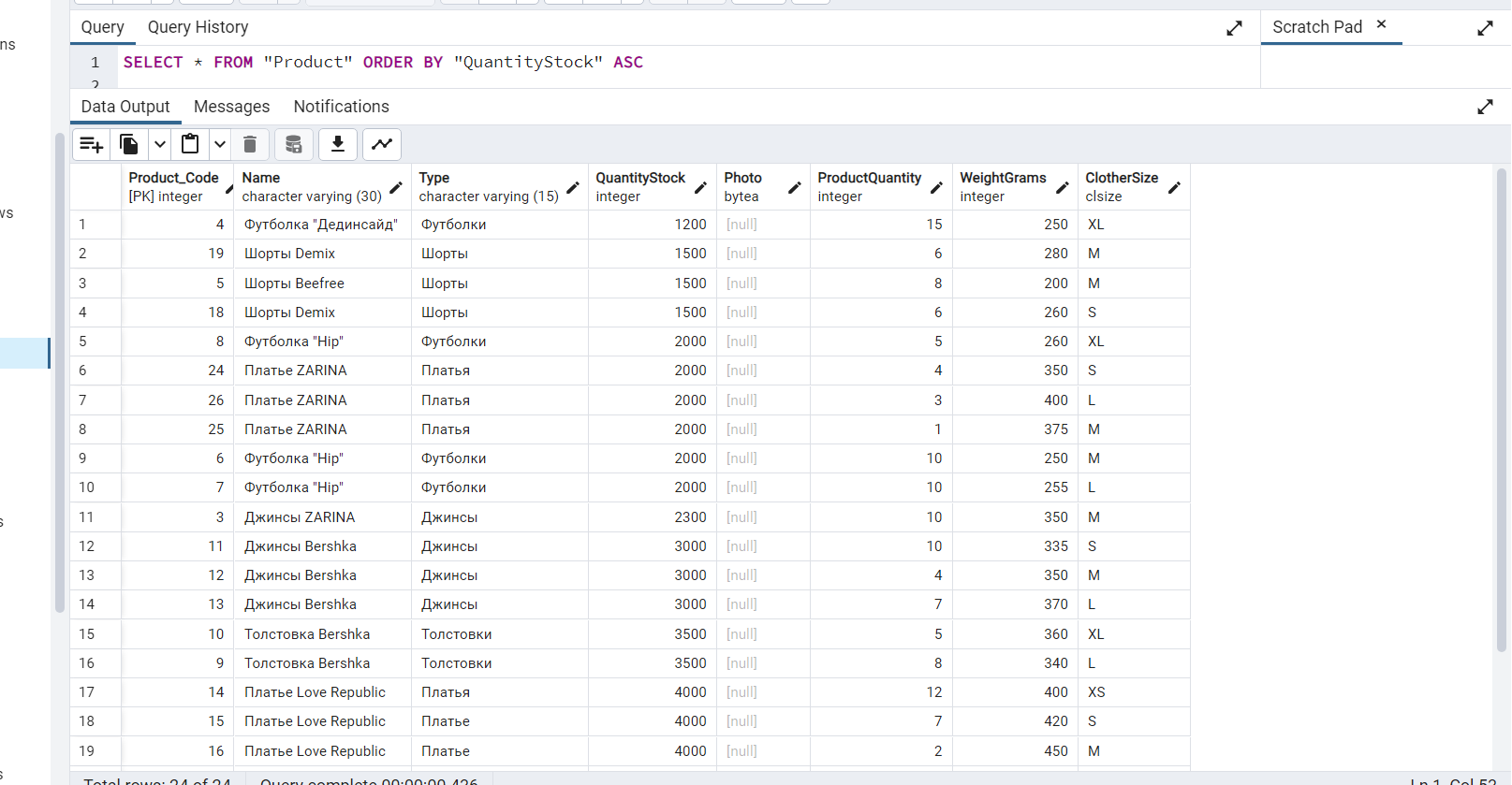
# Запрос с простыми условиями, условиями, содержащими IN или BETWEEN.

SELECT \* FROM "Product" WHERE "Type" IN ('Куртки', 'Платья');



# Запросы с сортировкой по нескольким полям, направлениям.

SELECT \* FROM "Product" ORDER BY "QuantityStock" ASC

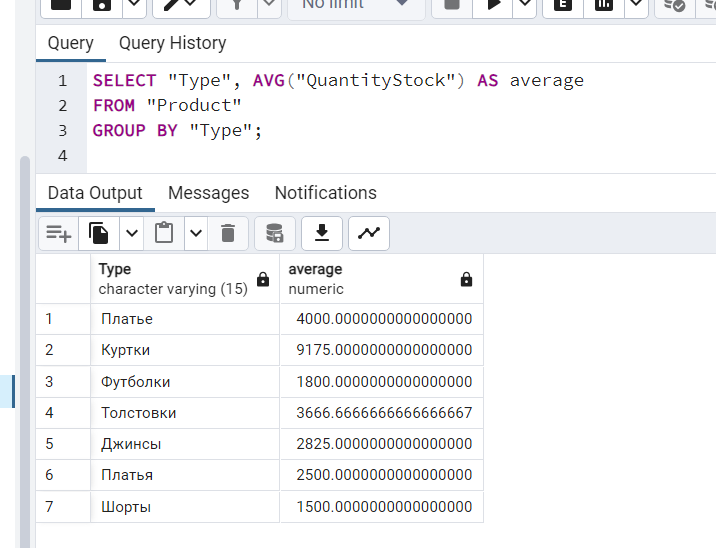


# Запросы с использованием групповых операций (группировка статистические функции, отбор по групповым функциям).

SELECT "Type", AVG("QuantityStock") AS average

FROM "Product"

GROUP BY "Type";



# Запросы с операцией над множествами (обязательно используя сортировку).

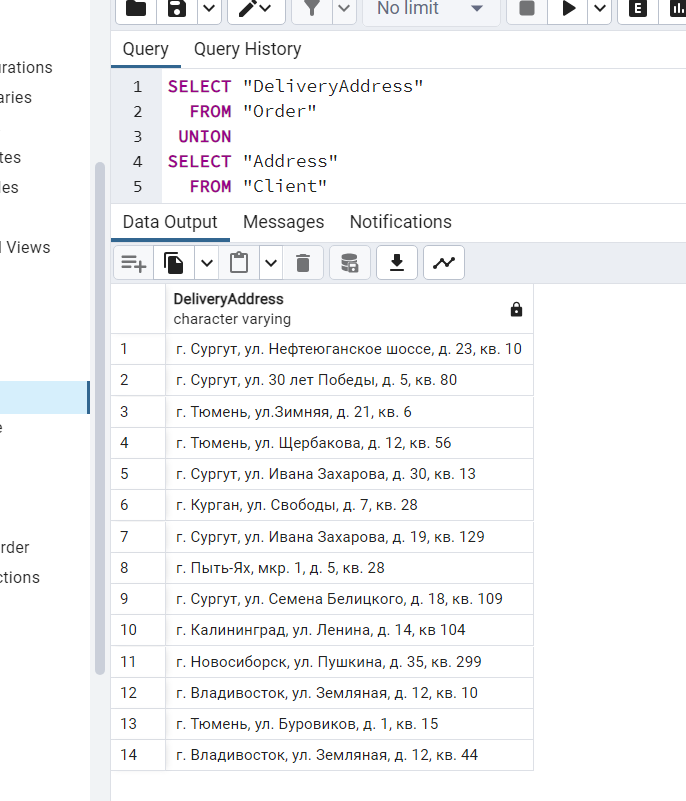
SELECT "DeliveryAddress"

FROM "Order"

UNION

SELECT "Address"

FROM "Client"

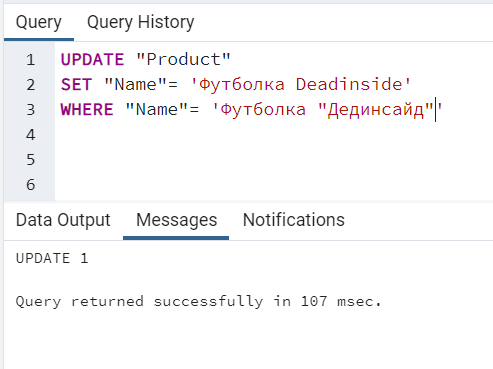


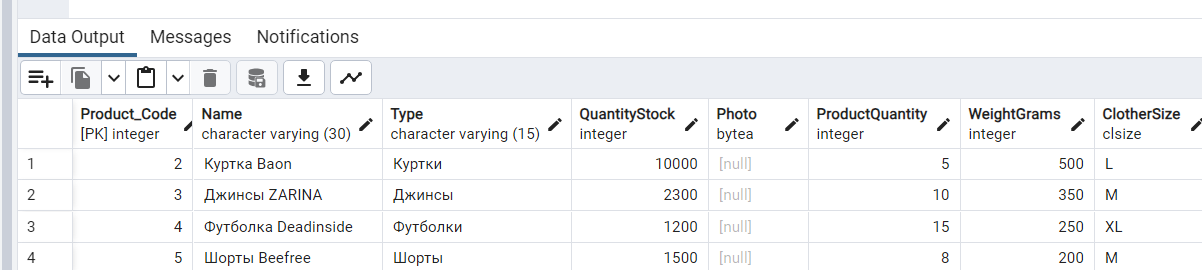
# Запросы на обновление.

UPDATE "Product"

SET "Name"= 'Футболка Deadinside'

WHERE "Name"= 'Футболка "Дединсайд"'

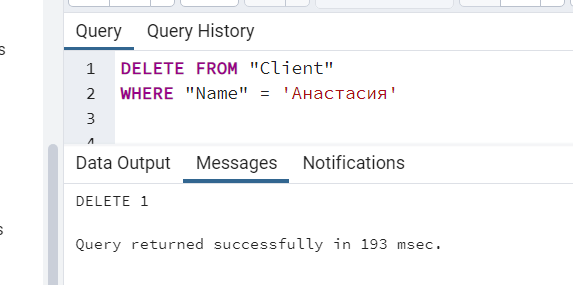




# Запросы на удаление.

DELETE FROM "Client"

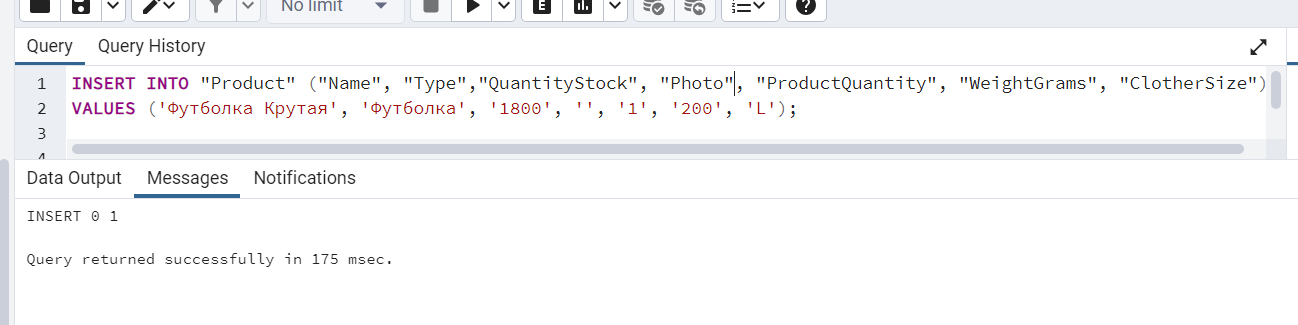
WHERE "Name" = 'Анастасия'



# Запросы на вставку.

INSERT INTO "Product" ("Name", "Type","QuantityStock", "Photo", "ProductQuantity", "WeightGrams", "ClotherSize")

VALUES ('Футболка Крутая', 'Футболка', '1800', '', '1', '200', 'L');





# Используя таблицу с персональными данными из своей БД или demo БД в PostgreSQL отобразите список сотрудников/персон (указав их Фамилию И. в одной колонке), которые в следующем месяце будут отмечать юбилей, с указанием возраста, даты рождения, даты юбилея. Заголовки должны соответствовать шаблону вывода данных.

SELECT

CONCAT("Surname", ' ', LEFT("Name", 1), '.') AS "FullName",

DATE\_PART('year', AGE(NOW(), "BirthDate")) AS "Age",

"BirthDate" AS "BirthDate",

"BirthDate" +INTERVAL '1 year' + INTERVAL '1 year' \* DATE\_PART('year', AGE(NOW(), "BirthDate")) AS "Anniversary"

FROM

"Client"

WHERE

EXTRACT(MONTH FROM "BirthDate") = EXTRACT(MONTH FROM NOW() + INTERVAL '1 month')

AND DATE\_PART('year', AGE(NOW(), "BirthDate"))::integer % 5 = 4

