Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2 по дисциплине «База данных»

Реализация SQL-запросов для создания схемы базы данных

Студента-заочника 3 курса

группы №250541

Власов Р. Е.

Проверила: магистр технических наук

Куприянова Д. В.

МИНСК 2024

# Цель работы

Целью работы является создание запросов на выборку данных на языке SQL с использованием предложений SELECT, FROM (JOINS), WHERE, GROUP BY и ORDER BY оператора SELECT, а также подзапросов, скалярных и агрегатных функций.

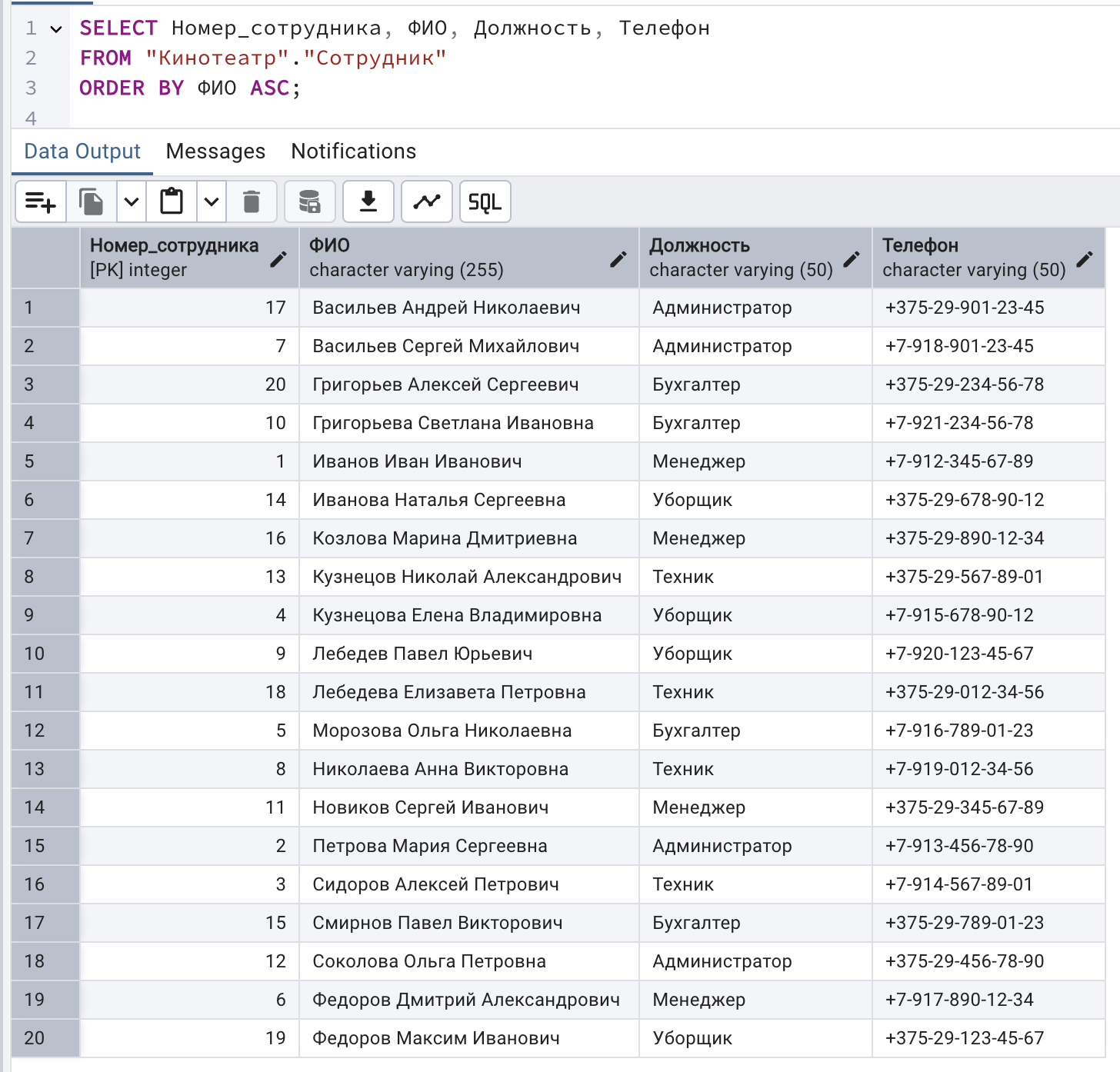
Результаты работы

1. Вывести список всех сотрудников и их должностей вместе с номерами телефонов, отсортированный по ФИО сотрудника.

SELECT Номер\_сотрудника, ФИО, Должность, Телефон

FROM "Кинотеатр"."Сотрудник"

ORDER BY ФИО ASC;

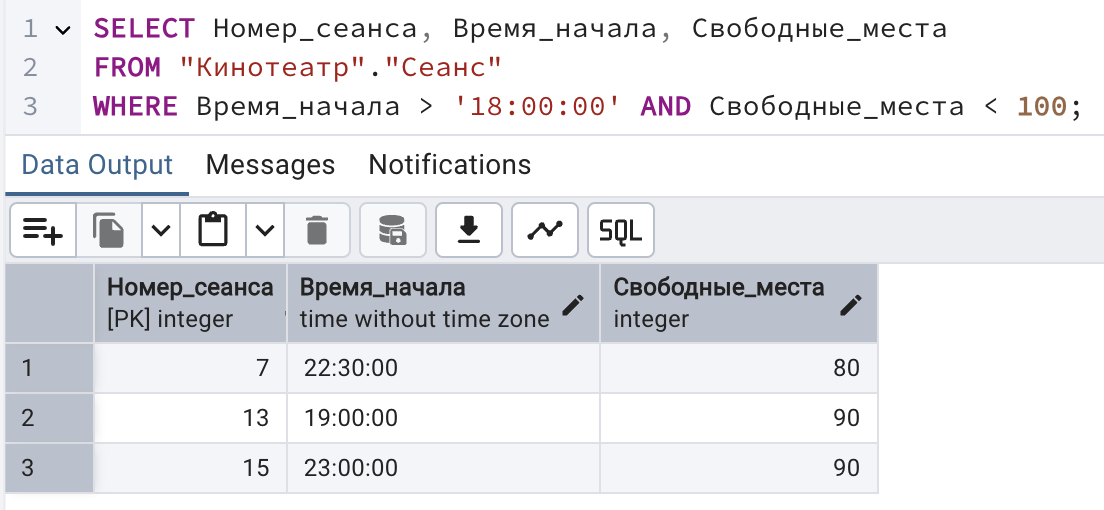


1. Найти сеансы, у которых время начала после 18:00:00 и количество свободных мест меньше 100.

SELECT Номер\_сеанса, Время\_начала, Свободные\_места

FROM "Кинотеатр"."Сеанс"

WHERE Время\_начала > '18:00:00' AND Свободные\_места < 100;

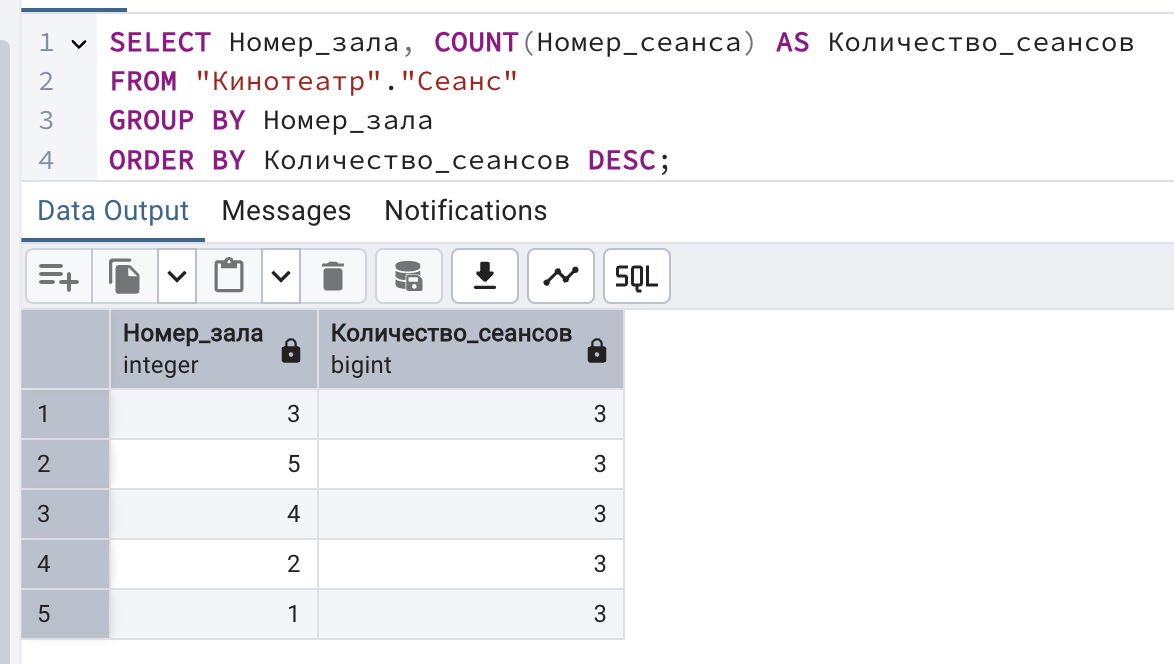


1. Подсчитать количество сеансов в каждом зале, сгруппировать по Номер\_зала и отсортировать по количеству сеансов по убыванию.

SELECT Номер\_сеанса, Время\_начала, Свободные\_места

FROM "Кинотеатр"."Сеанс"

WHERE Время\_начала > '18:00:00' AND Свободные\_места < 100;



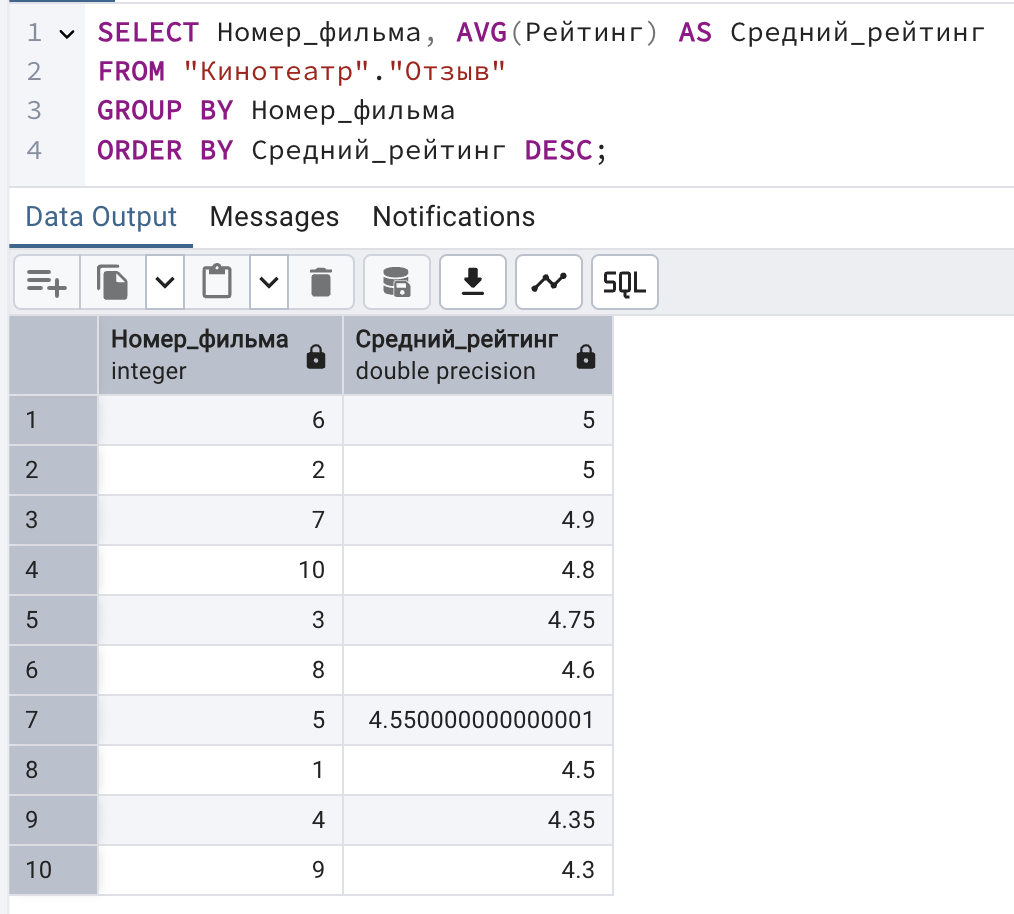
1. Вывести средний рейтинг фильмов, сгруппировать по Номер\_фильма и отсортировать по убыванию среднего рейтинга.

SELECT Номер\_фильма, AVG(Рейтинг) AS Средний\_рейтинг

FROM "Кинотеатр"."Отзыв"

GROUP BY Номер\_фильма

ORDER BY Средний\_рейтинг DESC;



1. Подсчитать количество отзывов с рейтингом выше 4.5 для каждого фильма, отсортировать по убыванию количества отзывов.

SELECT Номер\_фильма, COUNT(Номер\_отзыва) AS Количество\_высоких\_оценок

FROM "Кинотеатр"."Отзыв"

WHERE Рейтинг > 4.5

GROUP BY Номер\_фильма

ORDER BY Количество\_высоких\_оценок DESC;

# 

1. Подсчитать общее количество мест и количество электронных мест в каждом зале, отсортировав по возрастанию номера зала.

# SELECT Номер\_зала,

# COUNT(Номер\_сидения) AS Общее\_количество\_мест,

# SUM(CASE WHEN Является\_электронным = true THEN 1 ELSE 0 END) AS Электронные\_места

# FROM "Кинотеатр"."Место"

# GROUP BY Номер\_зала

# ORDER BY Номер\_зала;

# 

1. Определить ряды, в которых все места являются электронными, и вывести номера залов и рядов.

# SELECT Номер\_зала, Ряд

# FROM "Кинотеатр"."Место"

# GROUP BY Номер\_зала, Ряд

# HAVING COUNT(Номер\_сидения) = SUM(CASE WHEN Является\_электронным = true THEN 1 ELSE 0 END);

# 

8) Найти клиентов с максимальным количеством заказанных фильмов и вывести их ФИО, email и количество фильмов.

# WITH КоличествоФильмов AS (

# SELECT ФИО, email,

# LENGTH(История\_заказов) - LENGTH(REPLACE(История\_заказов, ',', '')) + 1 AS Количество\_фильмов

# FROM "Кинотеатр"."Клиент"

# )

# SELECT ФИО, email, Количество\_фильмов

# FROM КоличествоФильмов

# WHERE Количество\_фильмов = (SELECT MAX(Количество\_фильмов) FROM КоличествоФильмов);

# 

1. Найти клиентов, у которых в истории заказов есть фильм “Матрица”, и вывести их ФИО, email и полную историю заказов.

# SELECT ФИО, email, История\_заказов

# FROM "Кинотеатр"."Клиент"

# WHERE История\_заказов ILIKE '%Матрица%'

# ORDER BY ФИО;

# 

1. Подсчитать общее количество билетов, проданных в каждой категории, и общую выручку по каждой категории.

# SELECT Категория,

# COUNT(Номер\_билета) AS Количество\_билетов,

# SUM(Цена) AS Общая\_выручка

# FROM "Кинотеатр"."Билет"

# GROUP BY Категория

# ORDER BY Общая\_выручка DESC;

# Вывод

При выполнении лабораторной работы были созданы запросы на выборку данных на языке SQL с использованием предложений SELECT, FROM, WHERE, GROUP BY и ORDER BY оператора SELECT, а также подзапросов, скалярных и агрегатных функций.