Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОННИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Базы данных

Тема «Столовая Лидо»

Лабораторная работа №5

Реализация SQL-запросов на выборку данных с использованием подзапросов, агрегатных функций, группировки и операций над множествами

Студент: М.С. Патюпин

Преподаватель: Д.В. Куприянова

МИНСК 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc194440870)

[1 SQL-ЗАПРОСЫ 4](#_Toc194440871)

[1.1 Dish 4](#_Toc194440872)

[1.2 Order 6](#_Toc194440873)

[1.3 Ingredient 7](#_Toc194440874)

[1.4 Employee и Position 9](#_Toc194440875)

[1.5 Visitor 11](#_Toc194440876)

[1.5 Supplier 13](#_Toc194440877)

[ЗАКЛЮЧЕНИЕ 14](#_Toc194440878)

# ВВЕДЕНИЕ

В данной лабораторной работе будет получен опыт по создание SQL-запросы для выборки данных из реляционных баз данных, используя подзапросы, агрегатные функции, а также группировку данных с помощью оператора GROUP BY и операции над множествами (UNION, INTERSECT, MINUS). Работа включает несколько этапов, начиная с получения заданий от преподавателя на основе схемы данных, созданной в лабораторной работе №2, и реализованной в виде таблиц в СУБД в лабораторной работе №3.

Основная цель данной лабораторной работы — формирование практических навыков написания SQL-запросов, которые эффективно взаимодействуют с данными. При выполнении заданий важно учитывать правила написания запросов, включая ограничения на использование скалярных подзапросов и необходимость предварительного изучения данных в таблицах. Каждый запрос будет реализован в виде одного оператора SQL SELECT, который может включать подзапросы и группировку данных.

Эта лабораторная работа поможет углубить понимание работы с SQL и подготовит нас к более сложным задачам анализа и обработки данных в реляционных базах данных.

# SQL-ЗАПРОСЫ

## Dish

Задание: *Вывести список блюд дешевле среднего.*

Скрипт для выведения списка блюд дешевле среднего:

SELECT name, cost

FROM dish

WHERE cost < (SELECT AVG(cost) FROM dish)

ORDER BY name;

Часть данных таблицы Dish до выполнения скрипта представлена на рисунке 1.1.



Рисунок 1.1 – Таблица Dish до скрипта

Таблица Dish после скрипта представлена на рисунке 1.2.



Рисунок 1.2 – Результат cкрипта

Задание: *Вывести категории имеющие более 3х блюд.*

Скрипт для выведения категорий с более 3 блюд:

SELECT category, COUNT(\*)

FROM dish

GROUP BY category

HAVING COUNT(\*) > 3;

Таблица Dish после скрипта представлена на рисунке 1.3.

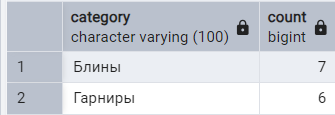


Рисунок 1.3 – Результат cкрипта

Задание: *Вывести блюда которые никогда не заказывали.*

Скрипт для выведения блюд которые никогда не заказывали:

SELECT name

FROM dish

WHERE dishid NOT IN (SELECT DISTINCT dishid FROM orderdish);

Таблица Dish после скрипта представлена на рисунке 1.4.

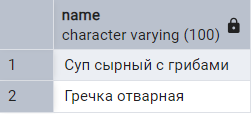


Рисунок 1.4 – Результат cкрипта

## 1.2 Order

Задание: *Вывести количество заказов по статусам.*

Скрипт для выведения количество заказов разного статуса:

SELECT status, COUNT(\*)

FROM "Order"

GROUP BY status;

Таблица Order до скрипта представлена на рисунке 1.5

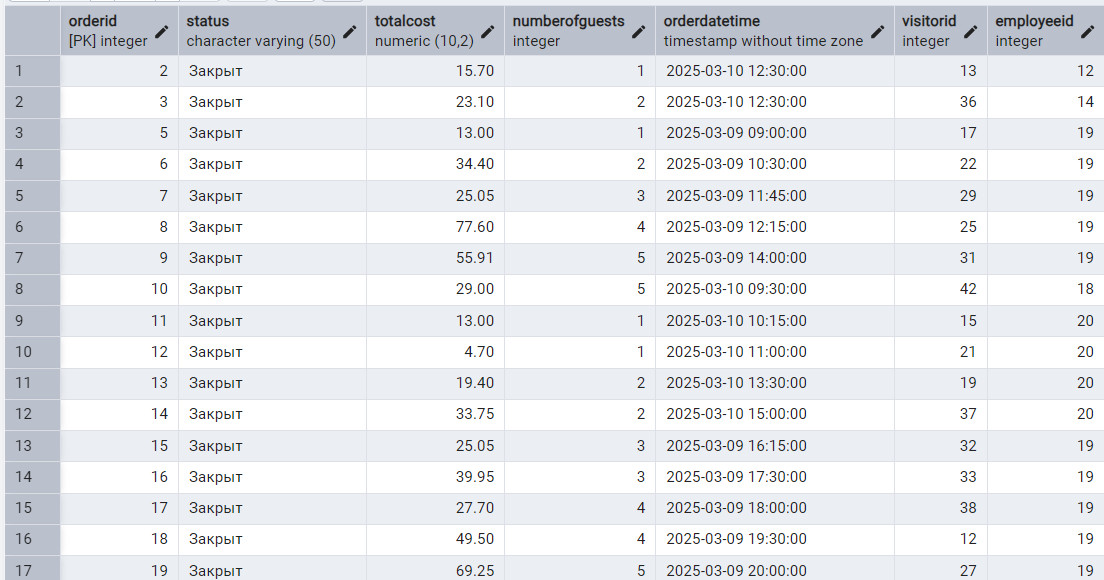


Рисунок 1.5 – Таблица Order до скрипта

Таблица Order после скрипта представлена на рисунке 1.6.

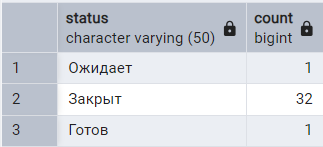


Рисунок 1.6 – Результат cкрипта

Задание: *Подсчитать средний чек по дням недели.*

Скрипт для подсчета среднего чека:

SELECT

EXTRACT(DOW FROM orderdatetime) AS day\_of\_week,

TO\_CHAR(orderdatetime, 'Day') AS day\_name,

AVG(totalcost) AS avg\_check

FROM "Order"

GROUP BY day\_of\_week, day\_name

ORDER BY day\_of\_week;

Таблица Order до скрипта представлена на рисунке 1.5.

Таблица Order после выполнения скрипта представлена на рисунке 1.7.

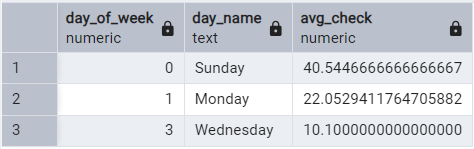


Рисунок 1.7 – Результат cкрипта

## 1.3 Ingredient

Задание: *Вывести 5 самых популярных ингредиентов.*

Скрипт для выведения популярных ингредиентов:

SELECT i.name, COUNT(di.dishid) AS dish\_count

FROM ingredient i

JOIN dishingredient di ON i.ingredientid = di.ingredientid

GROUP BY i.name

ORDER BY dish\_count DESC

LIMIT 5;

Таблица Ingredient до скрипта представлена на рисунке 1.8.

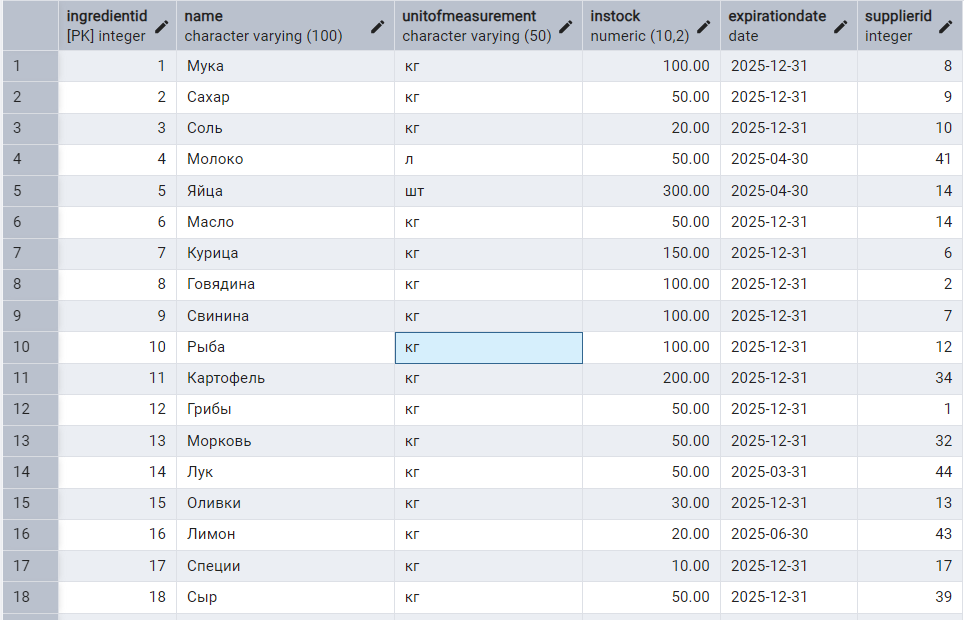


Рисунок 1.8 – Таблица Ingredient до скрипта

Таблица Ingredient после скрипта представлена на рисунке 1.9.

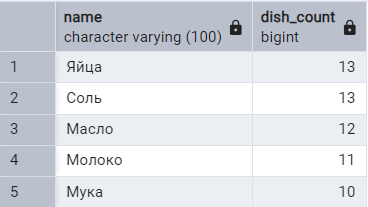


Рисунок 1.9 – Результат cкрипта

Задание: *Вывести ингредиенты не используемые ни в одном блюде.*

Скрипт для выведения не использующихся ингридиентов в текущих технологических картах блюд:

SELECT name

FROM ingredient

WHERE ingredientid NOT IN (SELECT DISTINCT ingredientid FROM dishingredient);

Таблица Ingredient до скрипта представлена на рисунке 1.8.

Таблица Ingredient после скрипта представлена на рисунке 1.10.

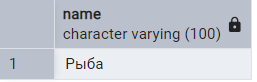


Рисунок 1.10 – Результат cкрипта

## 1.4 Employee и Position

Задание: *Вывести среднюю зарплату по должностям.*

Скрипт для выведения средней запрплаты:

SELECT p.name, p.workschedule, AVG(p.salary)

FROM "position" p

JOIN employeeposition ep ON p.positionid = ep.positionid

GROUP BY p.name, p.workschedule;

Таблица Position до скрипта представлена на рисунке 1.11.

Таблица Employee до скрипта представлена на рисунке 1.12.

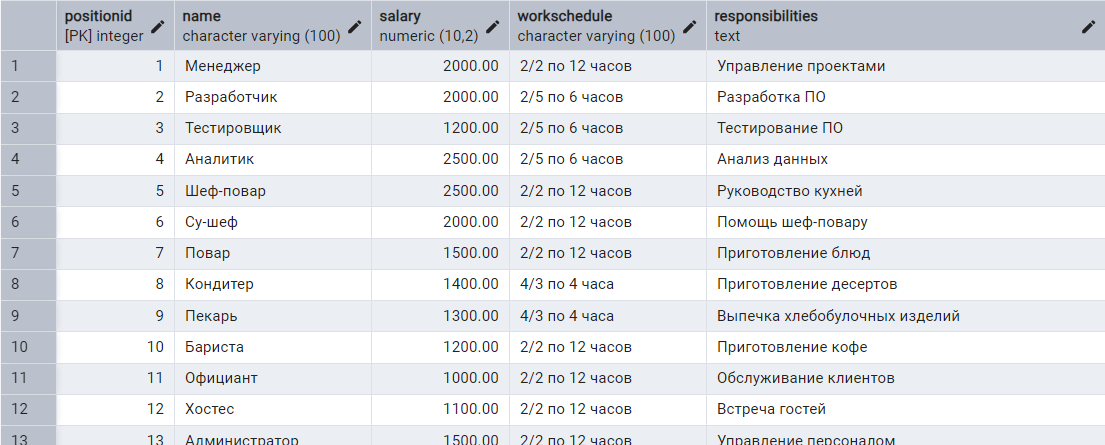


Рисунок 1.11 – Таблица Position до скрипта

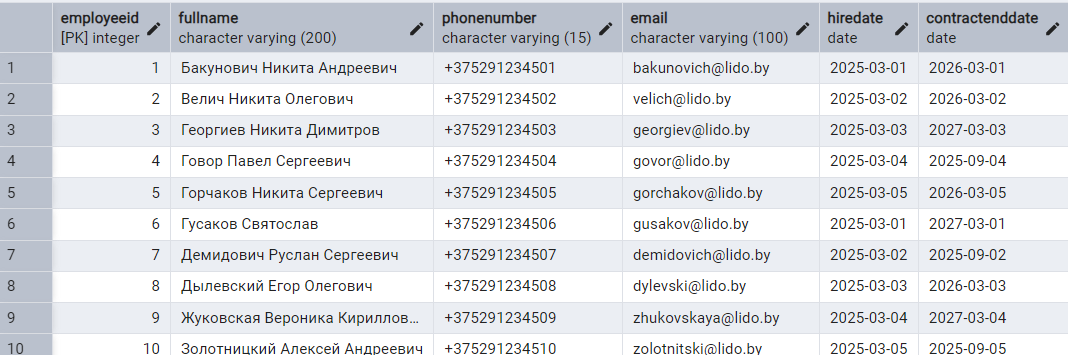


Рисунок 1.12 – Таблица Employee до скрипта

Таблица Position после скрипта представлена на рисунке 1.13.



Рисунок 1.13 – Результат cкрипта

Задание: *Вывести загруженности сотрудников принимающих заказы.*

Скрипт для выведения загруженности:

SELECT e.fullname, COUNT(o.orderid) AS orders\_served

FROM employee e

LEFT JOIN "Order" o ON e.employeeid = o.employeeid

GROUP BY e.employeeid, e.fullname

HAVING COUNT(o.orderid) > 0

ORDER BY orders\_served DESC;

Таблица Employee до скрипта представлена на рисунке 1.11.

Таблица Employee после скрипта представлена на рисунке 1.14.

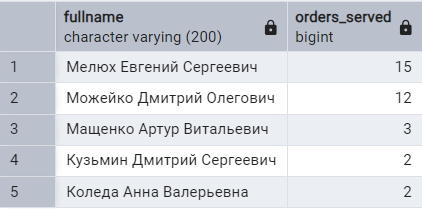


Рисунок 1.14 – Результат cкрипта

Задание: *Вывести должности занимаемые сотрудником, если таковых несколько.*

Скрипт для совместных должностей сотрудников:

SELECT

e.fullname AS "Сотрудник",

STRING\_AGG(p.name, ', ') AS "Должности"

FROM employee e

JOIN employeeposition ep ON e.employeeid = ep.employeeid

JOIN "position" p ON ep.positionid = p.positionid

GROUP BY e.employeeid, e.fullname

HAVING COUNT(ep.positionid) > 1;

Таблица Employee до скрипта представлена на рисунке 1.12.

Таблица Employee после скрипта представлена на рисунке 1.15.



Рисунок 1.15 – Результат cкрипта

## 1.5 Visitor

Задание: *Вывести 5 клиентов с наибольшей суммой заказов.*

Скрипт для выведения клиентов с наибольшой суммой заказов:

SELECT v.name, SUM(o.totalcost) AS totalcost

FROM visitor v

JOIN "Order" o ON v.visitorid = o.visitorid

GROUP BY v.visitorid, v.name

ORDER BY totalcost DESC

LIMIT 5;

Таблица Visitor до скрипта представлена на рисунке 1.16.

Таблица Visitor после скрипта представлена на рисунке 1.17.

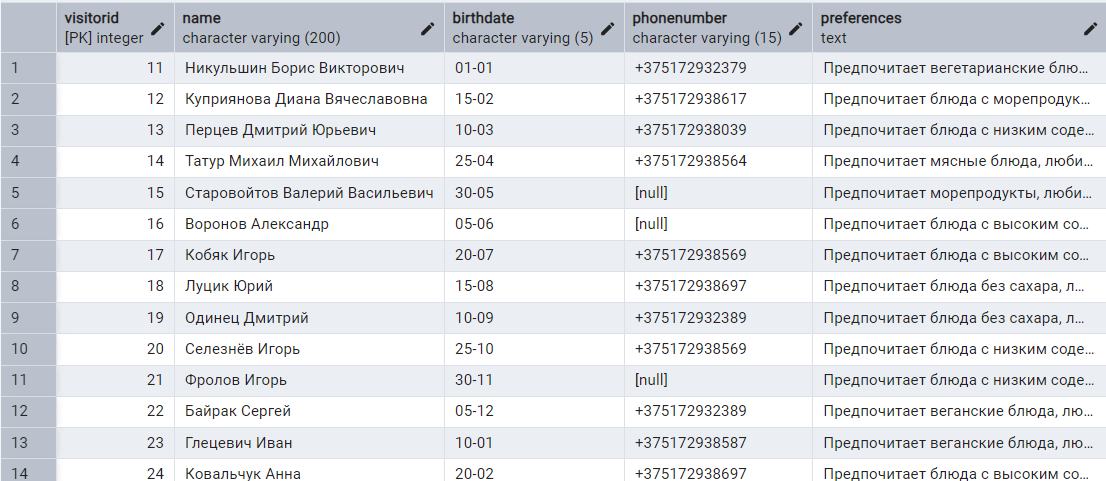


Рисунок 1.16 – Таблица Visitor до скрипта

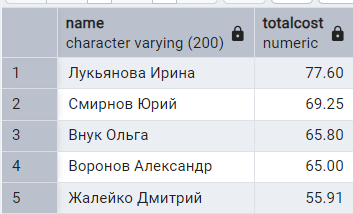


Рисунок 1.17 – Результат cкрипта

Задание: *Вывести постоянных клиентов с днем рождения в текущем месяце.*

Скрипт для выведения клиентов с днем рождения в текущем месяце:

SELECT name, birthdate

FROM visitor

WHERE

birthdate ~ '^\d{2}-\d{2}$' AND

EXTRACT(MONTH FROM TO\_DATE(birthdate || '-2000', 'DD-MM-YYYY')) = EXTRACT(MONTH FROM CURRENT\_DATE);

Таблица Visitor до скрипта представлена на рисунке 1.16.

Таблица Visitor после скрипта представлена на рисунке 1.18.

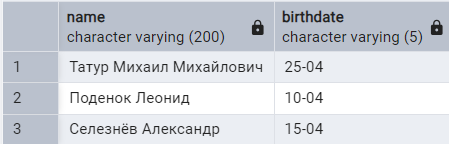


Рисунок 1.18 – Результат cкрипта

## 1.5 Supplier

Задание: *Вывести топ 3 поставщиков по количеству поставляемой продукции.*

Скрипт для выведения топа поставщиков:

SELECT s.companyname, COUNT(i.ingredientid) AS ingredients\_count

FROM supplier s

LEFT JOIN ingredient i ON s.supplierid = i.supplierid

GROUP BY s.supplierid, s.companyname

ORDER BY ingredients\_count DESC

LIMIT 3;

Таблица Supplier до скрипта представлена на рисунке 1.19.



Рисунок 1.19 – Таблица Supplier до скрипта

Таблица Supplier после скрипта представлена на рисунке 1.20.

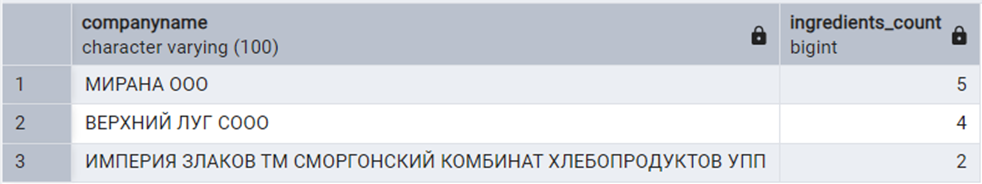


Рисунок 1.20 – Результат cкрипта

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были успешно реализованы SQL-запросы для выборки данных из реляционной базы данных столовой Лидо с использованием подзапросов, агрегатных функций, группировки и операций над множествами. Работа позволила закрепить теоретические знания и развить практические навыки взаимодействия с реляционными базами данных.

Основные результаты работы:

1. Освоение сложных SQL-конструкций.  
2. Работа с агрегатными функциями и группировкой.  
3. Применение операций над множествами.

4. Анализ структуры данных.

Работа подтвердила важность корректного проектирования типов данных на этапе создания таблиц. Навыки, полученные в ходе выполнения заданий, позволяют эффективно решать задачи анализа данных, включая формирование отчётов и статистики. Дальнейшее развитие может быть направлено на изучение оконных функций, индексов для оптимизации запросов и методов работы с большими объёмами данных.

Лабораторная работа стала важным этапом в освоении языка SQL и подготовила базу для решения более сложных задач в области проектирования и управления базами данных.