**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №4 «Запросы на выборку и модификацию данных. Представления. Работа с индексами**»**

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Бархатова Н.А.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

**Оглавление**

[Цель работы 3](#_Toc153123739)

[Практическое задание 3](#_Toc153123740)

[Схема базы данных: 3](#_Toc153123741)

[Выполнение 3](#_Toc153123742)

[Запросы к базе данных 3](#_Toc153123743)

[1. Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц. 4](#_Toc153123744)

[2. Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа). 4](#_Toc153123745)

[3. Подсчитать, сколько ингредиентов содержит каждое блюдо. 5](#_Toc153123746)

[4. Вывести название блюда, содержащее максимальное число ингредиентов. 5](#_Toc153123747)

[5. Какой повар может приготовить максимальное число видов блюд? 7](#_Toc153123748)

[6. Сколько закреплено столов за каждым из официантов? 8](#_Toc153123749)

[7. Какой из ингредиентов используется в максимальном количестве блюд? 9](#_Toc153123750)

[Представления 10](#_Toc153123751)

[1. Для расчета стоимости ингредиентов для заданного блюда 10](#_Toc153123752)

[2. Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату. 11](#_Toc153123753)

[Запросы на модификацию данных 14](#_Toc153123754)

[1. INSERT (добавление новых данных в таблицу order) 14](#_Toc153123755)

[2. UPDATE (изменение данных в таблице order) 15](#_Toc153123756)

[3. DELETE (удаление данных из таблицы order) 18](#_Toc153123757)

[Создание индексов 20](#_Toc153123758)

[1. Простой индекс 20](#_Toc153123759)

[2. Составной индекс 21](#_Toc153123760)

[Вывод 23](#_Toc153123761)

# Цель работы

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

# Практическое задание

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

# Схема базы данных:

Изображение выглядит как текст, диаграмма, число, линия

Автоматически созданное описание

# Выполнение

# Запросы к базе данных

## Вывести данные официанта, принявшего заказы на максимальную сумму за истекший месяц.

**SELECT**

e.full\_name **AS** waiter\_name,

**SUM** (mip.price) **AS** total\_order\_amount

**FROM**

**schema**.**order** o

**JOIN**

**schema**.employee e **ON** o.employee\_id = e.personnel\_number

**JOIN**

**schema**.order\_composition oc **ON** o.id = oc.order\_id

**JOIN**

**schema**.menu\_item\_price mip **ON** oc.menu\_item\_id = mip.menu\_item\_id

**WHERE**

o.date\_time >= date\_trunc('month', **CURRENT\_DATE**) - interval '1 month'

**AND** o.date\_time < date\_trunc('month', **CURRENT\_DATE**) **AND** o.payment\_status = 'Paid'

**GROUP** **BY**

e.full\_name

**HAVING**

**SUM** (mip.price) = (**SELECT** **MAX**(total\_order\_amount) **FROM** (**SELECT** e.full\_name **AS** waiter\_name, **SUM** (mip.price) **AS** total\_order\_amount

**FROM** **schema**.**order** o

**JOIN** **schema**.employee e **ON** o.employee\_id = e.personnel\_number

**JOIN** **schema**.order\_composition oc **ON** o.id = oc.order\_id

**JOIN** **schema**.menu\_item\_price mip **ON** oc.menu\_item\_id = mip.menu\_item\_id

**WHERE** o.date\_time >= date\_trunc('month', **CURRENT\_DATE**) - interval '1 month'

**AND** o.date\_time < date\_trunc('month', **CURRENT\_DATE**) **AND** o.payment\_status = 'Paid'

**GROUP** **BY** e.full\_name) **AS** subquery)

**ORDER** **BY**

total\_order\_amount **DESC**;

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

## Рассчитать премию каждого официанта за последние 10 дней (5% от стоимости каждого заказа).

**SELECT**

e.full\_name **AS** waiter\_name,

**SUM** (mip.price \* **0**.**05**) **AS** total\_bonus

**FROM**

**schema**.**order** o

**JOIN**

**schema**.employee e **ON** o.employee\_id = e.personnel\_number

**JOIN**

**schema**.order\_composition oc **ON** o.id = oc.order\_id

**JOIN**

**schema**.menu\_item\_price mip **ON** oc.menu\_item\_id = mip.menu\_item\_id

**WHERE**

DATE\_TRUNC ('day', o.date\_time) >= (**CURRENT\_DATE** - interval '39 day') **AND** o.payment\_status = 'Paid'

**GROUP** **BY**

e.personnel\_number

**ORDER** **BY**

total\_bonus **DESC**;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

## Подсчитать, сколько ингредиентов содержит каждое блюдо.

**SELECT**

mic.menu\_item\_id,

mi.name **AS** menu\_item\_name,

**COUNT**(mic.ingredient\_id) **AS** num\_ingredients

**FROM**

**schema**.menu\_item\_composition mic

**JOIN**

**schema**.menu\_item mi **ON** mic.menu\_item\_id = mi.id

**GROUP** **BY**

mic.menu\_item\_id, mi.name

**ORDER** **BY**

mic.menu\_item\_id;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

## Вывести название блюда, содержащее максимальное число ингредиентов.

**SELECT**

mi.name **AS** menu\_item\_name,

**COUNT**(mic.ingredient\_id) **AS** num\_ingredients

**FROM**

**schema**.menu\_item\_composition mic

**JOIN**

**schema**.menu\_item mi **ON** mic.menu\_item\_id = mi.id

**GROUP** **BY**

mi.name

**HAVING**

**COUNT**(mic.ingredient\_id) = (**SELECT** **MAX**(num\_ingredients) **FROM** (**SELECT** mi.name **AS** menu\_item\_name, **COUNT**(mic.ingredient\_id) **AS** num\_ingredients

**FROM** **schema**.menu\_item\_composition mic

**JOIN** **schema**.menu\_item mi **ON** mic.menu\_item\_id = mi.id

**GROUP** **BY** mi.id, mi.name) **AS** subquery)

**ORDER** **BY**

num\_ingredients **DESC**;

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описаниеИзображение выглядит как текст, Шрифт, программное обеспечение, снимок экрана

Автоматически созданное описание

## Какой повар может приготовить максимальное число видов блюд?

**SELECT** e.full\_name, **COUNT**(cs.id\_menu\_item) **as** dish\_count

**FROM** **schema**.chef\_skills cs

**JOIN** **schema**.employee e **ON** e.personnel\_number = cs.id\_employee

**GROUP** **BY** e.personnel\_number, e.full\_name

**HAVING** **COUNT**(cs.id\_menu\_item) = (

**SELECT** **MAX**(dish\_count)

**FROM** (

**SELECT** **COUNT**(id\_menu\_item) **as** dish\_count

**FROM** **schema**.chef\_skills

**GROUP** **BY** id\_employee

) max\_dish\_count

);

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

## Сколько закреплено столов за каждым из официантов?

**SELECT**

e.full\_name **AS** waiter\_name,

**COUNT**(**DISTINCT** o.place\_id) **AS** num\_tables\_assigned

**FROM**

**schema**.**order** o

**JOIN**

**schema**.employee e **ON** o.employee\_id = e.personnel\_number

**GROUP** **BY**

e.personnel\_number, e.full\_name

**ORDER** **BY**

num\_tables\_assigned **DESC**;

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

## Какой из ингредиентов используется в максимальном количестве блюд?

**SELECT**

i.name,

**COUNT**(**DISTINCT** mic.menu\_item\_id) **AS** num\_dishes

**FROM**

**schema**.menu\_item\_composition mic

**JOIN**

**schema**.ingredient i **ON** i.id = mic.ingredient\_id

**GROUP** **BY**

i.id, i.name

**HAVING**

**COUNT**(**DISTINCT** mic.menu\_item\_id) = (**SELECT** **MAX**(num\_dishes) **FROM** (**SELECT** i.name, **COUNT**(**DISTINCT** mic.menu\_item\_id) **AS** num\_dishes

**FROM** **schema**.menu\_item\_composition mic

**JOIN** **schema**.ingredient i **ON** i.id = mic.ingredient\_id

**GROUP** **BY** i.id, i.name) **AS** subquery)

**ORDER** **BY**

num\_dishes **DESC**;

Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описание

## Представления

## Для расчета стоимости ингредиентов для заданного блюда

CREATE VIEW schema.view\_menu\_item\_cost AS

SELECT

mic.menu\_item\_id,

mi.name AS menu\_item\_name,

SUM (pc.ingredient\_price \* mic.ingredient\_value) AS total\_cost

FROM

schema.menu\_item\_composition mic

JOIN

schema.menu\_item mi ON mic.menu\_item\_id = mi.id

JOIN

schema.purchase\_composition pc ON mic.ingredient\_id = pc.ingredient\_id

GROUP BY

mic.menu\_item\_id, mi.name;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

## Для всех поваров количество приготовленных блюд по каждому блюду за определенную дату.

CREATE VIEW schema.view\_chef\_dishes\_count AS

SELECT

e.full\_name AS chef\_name,

mi.name AS menu\_item\_name,

COUNT(o.id) AS dishes\_count

FROM

schema.order o

JOIN

schema.order\_composition oc ON o.id = oc.order\_id

JOIN

schema.employee e ON oc.employee\_id = e.personnel\_number

JOIN

schema.menu\_item mi ON oc.menu\_item\_id = mi.id

WHERE

DATE\_TRUNC('day', o.date\_time) = '2023-11-01'::date

GROUP BY

e.full\_name, mi.name;

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

## Запросы на модификацию данных

## INSERT (добавление новых данных в таблицу order)

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

## UPDATE (изменение данных в таблице order)

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

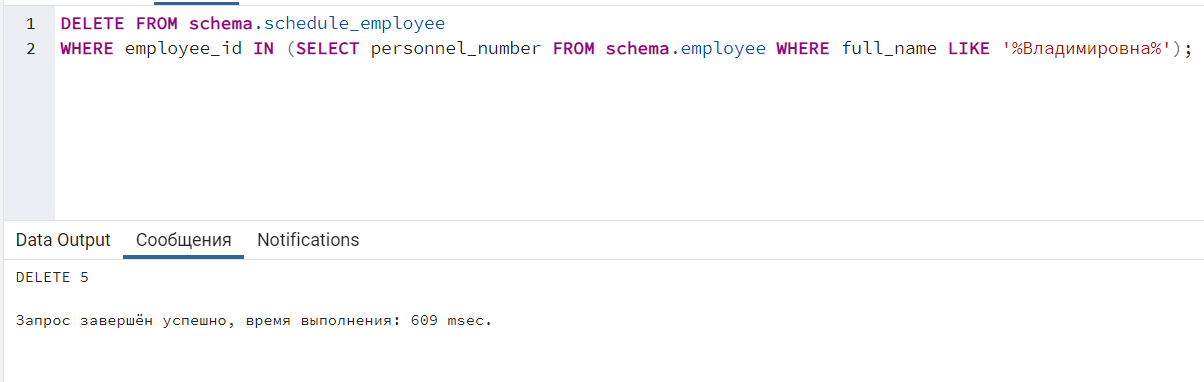
Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

## DELETE (удаление данных из таблицы order)

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

# Создание индексов

## Простой индекс

До создания индекса:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Создадим индекс:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

После создания индекса:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

## Составной индекс

До создания индекса:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Создадим индекс:

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

После создания индекса:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Изображение выглядит как снимок экрана, линия, диаграмма, График

Автоматически созданное описание

# Вывод

В ходе выполнения данной лабораторной работы я успешно взаимодействовала с базой данных PostgreSQL, создавая разнообразные запросы на выборку данных. Особое внимание уделялось работе с подзапросами, что позволило применять их в запросах на запись, удаление и обновление данных. Это открывает новые возможности для эффективного управления информацией в базе данных, сокращая необходимость в дополнительных запросах. В процессе работы над запросами на модификацию данных я убедились в гибкости подхода с использованием подзапросов, что позволяет осуществлять сложные манипуляции с данными, учитывая их взаимосвязи и контекст. Я выяснила, что применение индексов ускоряет операции чтения запросов. Я провели сравнение времени выполнения запросов до и после создания индексов, выявив повышение производительности. Важно отметить, что при этом следует балансировать между эффективностью чтения и возможными издержками при изменении данных в базе.