**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Информационная безопасность баз данных»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

«Реализация БД в рамках СУБД»

**Выполнил:**

Студент гр. N3347 Нгуен Тхе Вьет



*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Салихов Максим Русланович, аспирант ФБИТ 

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Содержание

[Содержание 2](#_Toc182298212)

[ВВЕДЕНИЕ 3](#_Toc182298213)

[ХОД РАБОТЫ 4](#_Toc182298214)

[1.1 Выбор СУБД 4](#_Toc182298215)

[1.2 Создание БД 4](#_Toc182298216)

[1.2.1 Код для создания всех таблиц; 6](#_Toc182298217)

[1.2.2 Код для внесения данных в созданные таблицы 6](#_Toc182298218)

[1.2.3 Код одной SQL-команды для модифицирования структуры таблицы 8](#_Toc182298219)

[1.3 Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных. 8](#_Toc182298220)

[1.4 Установить взаимосвязи между таблицами 8](#_Toc182298221)

[1.5 Создать представления, составленные в пункте 5 лабораторной 1. 8](#_Toc182298222)

[1.5.1 Представление для клиентов 8](#_Toc182298223)

[1.5.2 Представление для сотрудников 9](#_Toc182298224)

[1.5.3 Представление для менеджер 10](#_Toc182298225)

[1.6 Дополнительно. Тестовых запросов к вашей БД 11](#_Toc182298226)

[1.6.1 Получить все заказы с информацией о клиентах и сотрудниках. 11](#_Toc182298227)

[1.6.2 Перечислить продукты в конкретной категории. 12](#_Toc182298228)

[1.6.3 Найти заказы, размещенные конкретным клиентом. 12](#_Toc182298229)

[1.6.4 Найти the Most Popular Product 12](#_Toc182298230)

[1.6.5 Расчет общего дохода за определенный период 13](#_Toc182298231)

ВВЕДЕНИЕ

**Цель работы:** Получение навыков по работе с современными системами управления базами данных

**Задание**

1. Выбрать систему управления базами данных (СУБД), которая будет использована в рамках лабораторной работы. Кратко обосновать свой выбор.
2. Создать БД в выбранной в вами СУБД на основе итоговой разработанной схемы отношений из ЛР 1. Заполните созданную вами БД информацией, сгенерируйте как минимум 7-8 кортежей с данными для каждой из ваших основных таблиц. В отчете по лабораторной работе укажите следующий SQL-код (написанный вами или сгенерированный средствами администрирования СУБД):

* код для создания всех таблиц;
* код для внесения данных в созданные таблицы;
* код хотя бы одной SQL-команды для модифицирования структуры таблицы;

1. Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.
2. Установить взаимосвязи между таблицами.
3. Дополнительно. Тестовых запросов к вашей БД
4. Создать представления, составленные в пункте 5 лабораторной 1.

ХОД РАБОТЫ

## Выбор СУБД

В качестве инструментария для создания БД использована СУБД postgresSQL, а также графический интерфейс pgAdmin4.

## Создание БД

CREATE DATABASE cafeteria\_db;

A screen shot of a computer

Description automatically generated

1. создать базу данных

A screen shot of a computer

Description automatically generated

1. получить доступ к базе данных

Для начала создадим сущности в соответствии со схемой отношений на рисунке 1

A screenshot of a diagram

Description automatically generated

1. Схема предварительных отношений БД кафетерия

### Код для создания всех таблиц;

-- Creating Employee table

CREATE TABLE Employee (

ID\_employee SERIAL PRIMARY KEY,

Name VARCHAR(255),

Position VARCHAR(255),

Phone\_Number VARCHAR(20),

Email VARCHAR(255)

);

-- Creating Product table

CREATE TABLE Product (

ID\_product SERIAL PRIMARY KEY,

Name VARCHAR(255),

Price DECIMAL(10,2),

Category VARCHAR(255)

);

-- Creating Customer Table

CREATE TABLE Customer (

Id\_customer SERIAL PRIMARY KEY,

Name VARCHAR(255),

Phone\_Number VARCHAR(20),

Email VARCHAR(255)

);

-- Creating Order table

CREATE TABLE "order" (

ID\_order SERIAL PRIMARY KEY,

"Date" DATE,

Total\_Amount DECIMAL(10,2),

Status VARCHAR(255),

Customer\_ID INT,

Employee\_ID INT

);

-- Creating Order\_Product table

CREATE TABLE Order\_Product (

Order\_ID INT,

Product\_ID INT,

PRIMARY KEY (Order\_ID, Product\_ID)

);

A screenshot of a computer

Description automatically generated

### Код для внесения данных в созданные таблицы

-- Sample data for Employee table

INSERT INTO Employee (Name, Position, Phone\_Number, Email) VALUES

('Alice Johnson', 'Waiter', '555-123-4567', 'alice.johnson@example.com'),

('Bob Williams', 'Cook', '555-987-6543', 'bob.williams@example.com'),

('Charlie Brown', 'Manager', '555-567-8901', 'charlie.brown@example.com'),

('Diana Davis', 'Waiter', '555-234-5678', 'diana.davis@example.com'),

('Eve Edwards', 'Cook', '555-876-5432', 'eve.edwards@example.com'),

('Frank Foster', 'Cashier', '555-345-6789', 'frank.foster@example.com'),

('Grace Garcia', 'Waiter', '555-765-4321', 'grace.garcia@example.com'),

('Henry Harris', 'Cook', '555-456-7890', 'henry.harris@example.com');

-- Sample data for Product table

INSERT INTO Product (Name, Price, Category) VALUES

('Coffee', 2.50, 'Beverage'),

('Tea', 2.00, 'Beverage'),

('Pizza', 12.00, 'Food'),

('Burger', 8.50, 'Food'),

('Salad', 7.00, 'Food'),

('Cake', 5.00, 'Dessert'),

('Ice Cream', 4.00, 'Dessert'),

('Soda', 1.50, 'Beverage');

-- Sample data for Customer table

INSERT INTO Customer (Name, Phone\_Number, Email)

VALUES

('Ivan Ivanov', '555-111-2222', 'ian.ivanov@example.com'),

('Jane Jones', '555-333-4444', 'jane.jones@example.com'),

('Kevin Kim', '555-555-6666', 'kevin.kim@example.com'),

('Laura Lee', '555-777-8888', 'laura.lee@example.com'),

('Mike Miller', '555-999-0000', 'mike.miller@example.com'),

('Nancy Nelson', '555-222-3333', 'nancy.nelson@example.com'),

('Oscar Olsen', '555-444-5555', 'oscar.olsen@example.com'),

('Patricia Perez', '555-666-7777', 'patricia.perez@example.com');

-- Sample data for Order table

INSERT INTO "order" ("Date", Total\_Amount, Status, Customer\_ID, Employee\_ID)

VALUES

('2024-03-01', 14.50, 'Completed', 1, 1),

('2024-03-01', 10.50, 'Completed', 2, 2),

('2024-03-03', 7.00, 'Completed', 3, 3),

('2024-03-04', 7.50, 'Completed', 4, 4),

('2024-03-05', 19.00, 'Completed', 5, 5),

('2024-03-05', 10.00, 'Completed', 6, 6),

('2024-03-07', 11.00, 'Completed', 7, 7),

('2024-03-08', 12.00, 'Completed', 8, 8),

('2024-03-09', 19.00, 'Open', 3, 1),

('2024-03-10', 2.50, 'Open', 6, 4),

('2024-03-11', 8.50, 'Cancelled', 8, 2),

('2024-03-12', 4.00, 'Open', 1, 7),

('2024-03-12', 12.00, 'Open', 5, 8);

-- Sample data for Order\_Product table

INSERT INTO Order\_Product (Order\_ID, Product\_ID)

VALUES

(1, 1),

(1, 3),

(2, 2),

(2, 4),

(3, 5),

(4, 1),

(4, 6),

(5, 3),

(5, 7),

(6, 4),

(6, 8),

(7, 1),

(7, 5),

(8, 3),

(9, 2),

(9, 5),

(10, 1),

(11, 4),

(12, 7),

(13, 3);

### Код одной SQL-команды для модифицирования структуры таблицы

-- Adding a 'Hire Date' column to the Employee table

ALTER TABLE Employee

ADD "Hire Date" DATE;

-- Removing the 'Hire Date' column from the Employee table

ALTER TABLE Employee

DROP COLUMN "Hire Date";

## Индексировать таблицы. Добавить индексы для атрибутов, по которым происходит объединение таблиц, а также атрибуты по которым выполняется поиск/фильтрация данных.

-- Index on Customer\_ID in the Order table

CREATE INDEX idx\_order\_customer\_id ON "order" (Customer\_ID);

-- Index on Name and Category in the Product table

CREATE INDEX idx\_product\_name\_category ON product (Name, Category);

## Установить взаимосвязи между таблицами

-- Add foreign keys to order\_product table

ALTER TABLE Order\_Product

ADD CONSTRAINT FK\_Order\_Product\_Order

FOREIGN KEY (Order\_ID) REFERENCES "order"(ID\_order),

ADD CONSTRAINT FK\_Order\_Product\_Product

FOREIGN KEY (Product\_ID) REFERENCES Product(ID\_product);

-- Add foreign keys to order table

ALTER TABLE "order"

ADD CONSTRAINT FK\_Order\_Customer

FOREIGN KEY (Customer\_ID) REFERENCES Customer(Id\_customer),

ADD CONSTRAINT FK\_Order\_Employee

FOREIGN KEY (Employee\_ID) REFERENCES Employee(ID\_employee);

## Создать представления, составленные в пункте 5 лабораторной 1.

**Представления для базы данных кафетерия**

### Представление для клиентов

* Это представление позволяет клиентам видеть свою историю заказов, включая идентификатор заказа, дату, общую сумму, статус и названия заказанных продуктов.

**Объяснение:**

CREATE VIEW customer\_order\_history AS

SELECT

o.id\_order,

o."Date",

o.total\_amount,

o.status,

STRING\_AGG(p.name, ', ') AS product\_names

FROM

"order" o

JOIN

order\_product op ON o.id\_order = op.order\_id

JOIN

product p ON op.product\_id = p.id\_product

JOIN

customer c ON o.customer\_id = c.id\_customer

GROUP BY

o.id\_order, o."Date", o.total\_amount, o.status;

A screenshot of a computer

Description automatically generated

1. Представление для клиентов

### Представление для сотрудников

* Это представление показывает сотрудникам список открытых заказов, включая идентификатор заказа, имя клиента, название продукта и назначенного сотрудника.

**Объяснение:**

CREATE VIEW employee\_open\_orders AS

SELECT

o.id\_order,

c.Name AS customer\_name,

p.Name AS product\_name,

e.Name AS employee\_name

FROM

"order" o

JOIN

order\_product op ON o.id\_order = op.order\_id

JOIN

product p ON op.product\_id = p.id\_product

JOIN

customer c ON o.customer\_id = c.id\_customer

JOIN

employee e ON o.employee\_id = e.id\_employee

WHERE

o.status = 'Open';

A screenshot of a computer

Description automatically generated

1. Представление для сотрудников

### Представление для менеджер

* Это представление предоставляет информацию о продажах продуктов, отображая название каждого продукта и его общую выручку.

**Объяснение:**

CREATE VIEW product\_sales AS

SELECT

p.Name AS product\_name,

SUM(p.Price) AS total\_revenue

FROM

"order" o

JOIN

order\_product op ON o.id\_order = op.order\_id

JOIN

product p ON op.product\_id = p.id\_product

WHERE

o.status = 'Completed'

GROUP BY

p.Name;

A screenshot of a computer

Description automatically generated

1. Представление для менеджер

## Дополнительно. Тестовых запросов к вашей БД

### Получить все заказы с информацией о клиентах и сотрудниках.

SELECT

o.ID\_order,

o."Date",

o.Total\_Amount,

o.Status,

c.Name AS CustomerName,

e.Name AS EmployeeName

FROM

"order" o

JOIN

Customer c ON o.Customer\_ID = c.Id\_customer

JOIN

Employee e ON o.Employee\_ID = e.ID\_employee;

Этот запрос извлекает все заказы из таблицы "order" и объединяет их с таблицами Customer и Employee, чтобы отобразить имя клиента и сотрудника, который обработал заказ.

A screenshot of a computer screen

Description automatically generated

### Перечислить продукты в конкретной категории.

SELECT

p.Name,

p.Price

FROM

Product p

WHERE

p.Category = 'Food';

--Этот запрос выбирает названия и цены всех продуктов, принадлежащих к категории 'Food'.

A black and white screen with white text

Description automatically generated

### Найти заказы, размещенные конкретным клиентом.

SELECT

o.ID\_order,

o."Date",

o.Total\_Amount,

o.Status

FROM

"order" o

JOIN

Customer c ON o.Customer\_ID = c.Id\_customer

WHERE

c.Name = 'Ivan Ivanov';

--Этот запрос извлекает сведения о заказах, размещенных клиентом по имени «Ivan Ivanov»

A screen shot of a computer

Description automatically generated

### Найти the Most Popular Product

SELECT

p.Name,

COUNT(op.Product\_ID) AS OrderCount

FROM

Order\_Product op

JOIN

Product p ON op.Product\_ID = p.ID\_product

GROUP BY

p.Name

ORDER BY

OrderCount DESC

LIMIT 1;

A black background with white text

Description automatically generated

### Расчет общего дохода за определенный период

SELECT

SUM(o.Total\_Amount) AS TotalRevenue

FROM

"order" o

WHERE

o."Date" BETWEEN '2024-03-01' AND '2024-03-12';

A black and white screen with numbers

Description automatically generated