

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
(повна назва інституту/факультету)

КАФЕДРА інформатики та програмної інженерії
(повна назва кафедри)

КУРСОВА РОБОТА
з дисципліни «Бази даних»
(назва дисципліни)

на тему: »База даних для підтримки діяльності рекламного агентства»

Студента (ки) 2 курсу ІІІ-11 групи
спеціальності 121 «Інженерія програмного
забезпечення»

Прищеп В.С.

(прізвище та ініціали)

Керівник _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____

Кількість балів: _____ Оцінка ECTS _____

Члени комісії

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

(вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2022 рік

**Національний технічний університет України «Київський
політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)

Дисципліна Бази даних

Курс 2 Група ІІІ-11 Семестр 1

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Прищепя Владислав Станіславович
(прізвище, ім'я, по батькові)

1. Тема роботи База даних для підтримки діяльності рекламного агентства

керівник роботи

_____ (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____

—

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1.	Вступ	06.01.2023	
2.	Опис предметного середовища	06.01.2023	
3.	Постановка задачі	06.01.2023	
4.	Побудова ER-моделі	06.01.2023	
5.	Побудова реляційної схеми	06.01.2023	
6.	Створення бази даних з допомогою обраної СУБД	07.01.2023	
7.	Імпортування даних	07.01.2023	
8.	Створення користувачів та реалізація їх функціоналу	07.01.2023	
9.	Створення SQL запитів	08.01.2023	
10.	Висновок	09.01.2023	
11.	Перелік посилань	09.01.2023	

Студент

(підпис)

(прізвище та ініціали)

Керівник роботи

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 55 сторінок, 36 рисунків, 1 таблиця, 2 посилання.

Об'єкт дослідження: база даних для підтримки діяльності рекламного агентства.

Мета роботи: закріплення навичок з проектування, реалізації реляційних баз даних та їх використання для практичних задач.

Проведено аналіз предметного середовища, визначено сутності, атрибути та зв'язки між об'єктами. Побудована ER-модель предметного середовища, реляційну схему бази даних, виділено первинні та зовнішні ключі, визначено обмеження для підтримки цілісності даних. Розроблено скрипти для побудови спроектованої бази даних, імпортовано дані в неї, виконано різноманітні запити до неї.

Виконана програмна реалізація бази даних для підтримки діяльності рекламного агентства (варіант 25).

ЗМІСТ

Вступ	8
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	9
2 ПОСТАНОВКА ЗАВДАННЯ	10
3 ER-ДІАГРАМА	11
3.1 Бізнес правила.....	11
3.2 Вибір сутностей.....	11
3.3 Набори атрибутів сутностей.....	12
4 РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ.....	16
4.1 Побудова необхідних відношень та визначення первинних та зовнішніх ключів	16
4.2 Визначення обмежень цілісності для спроектованих відношень	16
5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ	17
5.1 Створення бази даних у форматі системи управління базою даних PostgreSQL	17
5.2 Імпортування даних в таблицю.....	21
6 СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ.....	23
6.1 Створення кадрового менеджера	23
6.2 Створення бухгалтера	23
6.3 Створення агента рекламного агентства	23
7 SQL ЗАПИТИ	24
7.1 Тригери.....	24
7.1.1 checkdate.....	24
7.1.2 countmat.....	25
7.1.3 checksum	26
7.1.4 ifsmallcom	27
7.1.5 checkatemp.....	29
7.1.6 ispayed	31
7.2 Функції та процедури	32
7.2.1 outcome.....	33
7.2.2 pop_ord_type	34
7.2.3 pop_dis.....	36
7.2.4 pop_distrib	37
7.2.5 freedep	38

7.2.6 unprayed	40
7.3 Запити	41
7.3.1 Список усіх замовлень	41
7.3.2 Список усіх матеріалів, які були замовлені у дистриб'юторів для виконання замовлень	42
7.3.3 Список усіх угод, які були укладені із замовниками на замовлення	44
7.3.4 Список усіх матеріалів, які треба доставити у відповідні відділи на вказані замовлення	45
7.3.5 Список усіх виданих зарплат робітникам	47
7.3.6 Список відділів та загальної кількості виданих замовлень кожному відділу	49
7.4 Представлення	50
7.4.1 stuff	50
7.4.2 transaction.....	51
7.5 Індекси	52
ВИСНОВОК	54
ПЕРЕЛІК ПОСИЛАНЬ.....	55

ВСТУП

Бази даних є важливою частиною будь-якого бізнесу, адже за допомогою них можна зберігати важливі дані про клієнтів, замовлення, робітників та прослідковувати залежності між явищами. На відміну від минулого, де все зберігалось на папері, бази даних дають можливість швидко отримувати інформацію, що нас цікавить.

Обрана мною тема є актуальною, тому що реклама є важливою частиною нашого життя, а рекламні агентства інформують пересічних громадян про акції та новинки споживчого ринку. Для створення великих обсягів реклами необхідно мати правильну організацію, з чим дуже допомагають бази даних.

Дана робота присвячена вивченню розробки програмного забезпечення з SQL, і стосується проектування бази даних для підтримки діяльності рекламного агентства, що допоможе організувати роботу та облік замовлень. Задача полягає в аналізі предметної області, побудови ER-діаграми та датологічної схеми реляційної бази даних, написанні SQL скриптів та запитів до неї. Виконання роботи проведено з допомогою PostgreSQL, оскільки вона є безкоштовним і легким в освоєнні засобом побудови баз даних.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

Основним об'єктом дослідження предметної області є створення реклами та процеси пов'язані з ними. Реклама створюється за замовленням клієнта.

Зазвичай замовлення створюється за договором між агентством та клієнтом і оплачується на місці переведенням коштів з банківського рахунку. Кожне замовлення має свій вид реклами, як от розклеювання листівок, реклама на рекламних щитах тощо. Замовлення осягає певний набір районів у місті (чи міст, якщо агентство працює по всій області чи країні). Замовлення віддають певному відділу агентства, що буде займатись цим замовленням, йому визначають термін виконання роботи.

Для замовлення необхідно у поставників замовити необхідні матеріали для виготовлення реклами та її розміщення.

У кожному відділку працює колектив, кожний учасник якого має свою посаду. Працівникам необхідно нараховувати зарплатню.

2 ПОСТАНОВКА ЗАВДАННЯ

Метою даної роботи є розробка бази даних для збереження результатів організації виконання замовлень на рекламу та організації робочого процесу. Тобто організація даних має бути такою в базі, щоб було максимально зручно вибудовувати план виконання замовлень. Виділимо основні задачі:

Агент рекламного агентства має мати змогу:

- Ввести облік замовлень, клієнтів, договорів.
- Ввести облік замовлень матеріалів у поставників
- Переглядати перелік усіх транзакцій з банків, якими були оплачені замовлення
- Знати про найпопулярніші види реклами, райони для їх розповсюдження та поставників матеріалів для надання рекомендацій клієнтам

Кадровий менеджер має мати змогу:

- Наймати та звільняти працівників та прив'язувати їх до відділів
- Вести облік діючих відділів агентства
- Змінювати дані про працівників

Бухгалтер має мати змогу:

- Переглядати перелік відділів та їх працівників
- Ввести облік нарахування та видачі заробітних плат працівникам
- Ввести облік усіх транзакцій з банків, якими були оплачені замовлення

3 ER-ДІАГРАМА

Після аналізу було виділено такі сутності та зв'язки між ними:

3.1 Бізнес правила

- 1) У відділі може працювати до 20 працівників, при кількості працівників менше 10 відділ вважається з малою кількістю працівників, при назначені такого відділку на замовлення повідомляти про його малу комплектацію.
- 2) Зарплатня для працівника складає від 5 до 20 тисяч
- 3) Один відділ може бути назначений на декілька замовлень одночасно
- 4) Для кожного району міста формується окреме замовлення і може бути назначені різні відділи; кожне таке замовлення посилається на договір загального замовлення, до договору додається платіжка, що покриває усі замовлення, що стосуються договору; допускається можливість однією платіжкою сплачувати декілька замовлень (замовлення з різними договорами)
- 5) Для запобігання марної витрати матеріалів, на кожне замовлення необхідно виділяти до 5 різних замовлень матеріалів у поставника
- 6) На замовлення виділяти не менше тижня
- 7) Строго вести облік нарахування та видачі зарплатні, не видавати зарплатню більше одного разу

3.2 Вибір сутностей

- Замовлення
- Тип замовлення
- Договір
- Клієнт
- Чек
- Банк

- Матеріал
- Поставник
- Район
- Відділ
- Працівник
- Зарплатня

3.3 Набори атрибутів сутностей

Таблиця 3.1 — Сутності та їхні атрибути

Сутність	Атрибути
orders	id department_id order_type_id deal_id district_id amount date_of_beginning date_of_end
order_type	id name
client	id name phone_number
deal	id number_of_document date client_id receipt_id
receipt	id

	bank_id sum date
bank	id name
district	id name
material	id order_id distributor_id name amount
distributor	id name
department	id name
employee	id name phone_number passport_id department_id position
salary	id employee_id sum date is_payed

Сутність **distributor** пов'язана *один до багатьох* із сутністю **material**, адже у одного поставника може бути замовлено декілька матеріалів.

Сутність **orders** пов'язана *один до багатьох* із сутністю **material**, адже

на одне замовлення може знадобитися декілька матеріалів.

Сутність **order_type** пов'язана *один до багатьох* із сутністю **order**, адже декілька замовлень можуть бути одного типу.

Сутність **district** пов'язана *один до багатьох* із сутністю **order**, адже в одному районі можуть бути виконані декілька замовлень.

Сутність **department** пов'язана *один до багатьох* із сутністю **order**, адже відділ може виконувати декілька замовлень.

Сутність **department** пов'язана *один до багатьох* із сутністю **employee**, адже у відділку працює декілька працівників.

Сутність **employee** пов'язана *один до багатьох* із сутністю **salary**, адже працівнику можуть нарахувати декілька заробітних плат.

Сутність **receipt** пов'язана *один до багатьох* із сутністю **deal**, адже одним платежем можна покрити декілька замовлень (мова йде про загальні замовлення, для яких складається договір, до якого прикріплюється чек).

Сутність **bank** пов'язана *один до багатьох* із сутністю **receipt**, адже одним банком можуть користуватися декілька клієнтів, через що вони можуть надіслати свої платіжки з одного банку.

Сутність **deal** пов'язана *один до багатьох* із сутністю **order**, адже договір може покривати декілька районів міста, через що на кожен район, для зручності організації роботи, створюється окреме замовлення.

Сутність **client** пов'язана *один до багатьох* із сутністю **deal**, адже клієнт може декілька раз звернутися до агентства й зробити декілька замовлень.

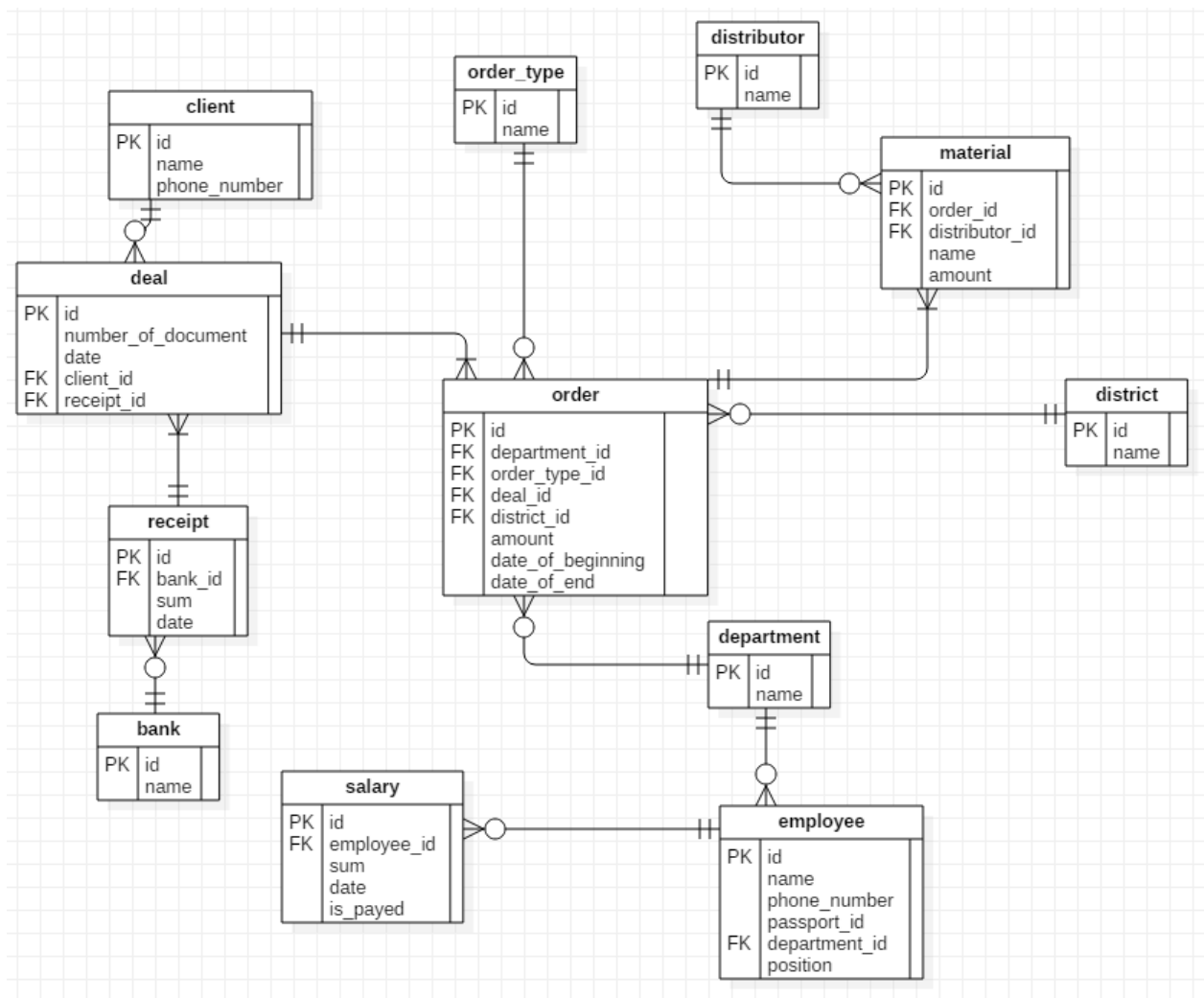


Рисунок 3.1 - ER-діаграма

4 РЕЛЯЦІЙНА МОДЕЛЬ БАЗИ ДАНИХ

4.1 Побудова необхідних відношень та визначення первинних та зовнішніх ключів

На рисунку можна побачити, що база даних знаходить у 3 нормальній формі, оскільки: поля таблиці декомпозовані, атрибути функціонально повно залежать від первинного ключа, кожен неключовий атрибут не є транзитивно залежним від первинного ключа.

4.2 Визначення обмежень цілісності для спроектованих відношень

1) Якщо не існує зовнішніх ключів, що посиляються на значення

головного ключа батьківського рядка, то тільки у цьому випадку його можна видалити. Для цього треба додати до таблиці обмеження з відповідними параметрами: `ALTER TABLE child ADD CONSTRAINT fk_parent_child FOREIGN KEY (parent_id) REFERENCES parent (id) ON UPDATE CASCADE ON DELETE RESTRICT`

2) Обов'язкові атрибути таблиць мають обмеження NOT NULL для запобігання помилок при роботі з даними

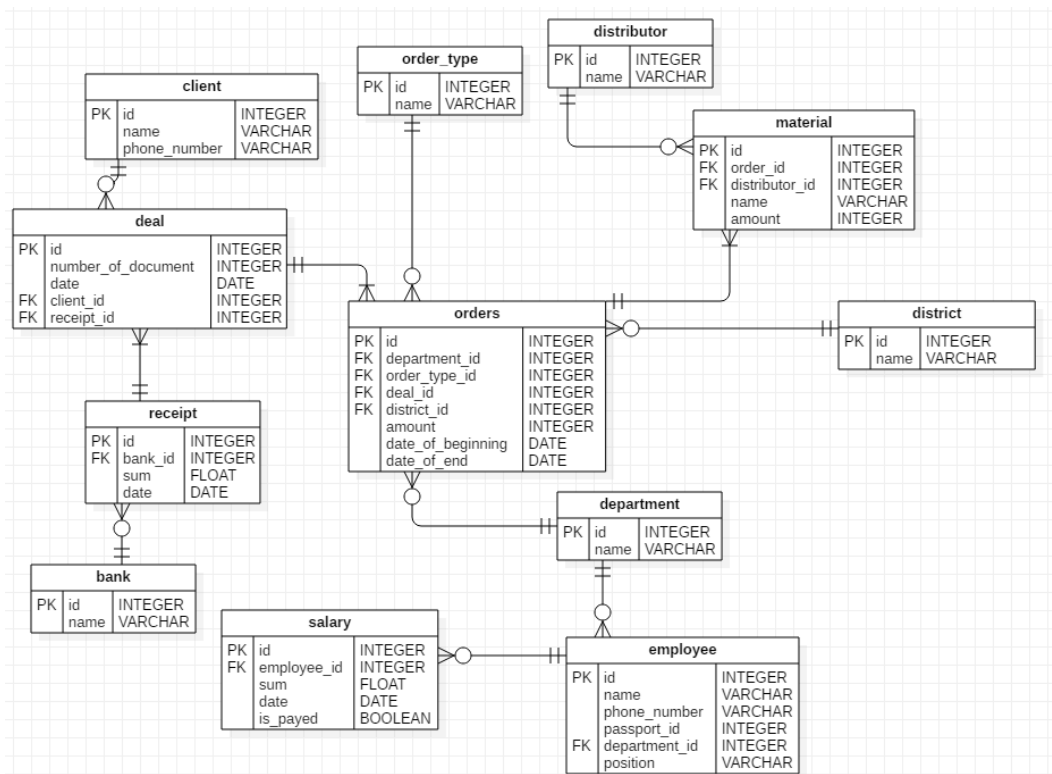


Рисунок 4.1 — Реляційна схема бази даних

5 РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

5.1 Створення бази даних у форматі системи управління базою даних PostgreSQL

```
CREATE TABLE IF NOT EXISTS client (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL,  
phone_number VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS bank (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS order_type (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS distributor (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS district (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS department (  

```

```
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS receipt (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
bank_id INT NOT NULL,  
sum NUMERIC(9, 2) NOT NULL,  
date DATE NOT NULL  
);
```

```
ALTER TABLE receipt ADD CONSTRAINT fk_bank_receipt  
FOREIGN KEY(bank_id) REFERENCES bank(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE IF NOT EXISTS deal (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
number_of_document INT NOT NULL,  
date DATE NOT NULL,  
client_id INT NOT NULL,  
receipt_id INT NOT NULL  
);
```

```
ALTER TABLE deal ADD CONSTRAINT fk_client_deal  
FOREIGN KEY(client_id) REFERENCES client(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
ALTER TABLE deal ADD CONSTRAINT fk_receipt_deal  
FOREIGN KEY(receipt_id) REFERENCES receipt(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE IF NOT EXISTS employee (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
name VARCHAR(40) NOT NULL,  
phone_number VARCHAR(20) NOT NULL,  
passport_id INT NOT NULL,  
department_id INT NOT NULL,  
position VARCHAR(20) NOT NULL  
);
```

```
ALTER TABLE employee ADD CONSTRAINT fk_department_employee  
FOREIGN KEY(department_id) REFERENCES department(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE IF NOT EXISTS salary (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
employee_id INT NOT NULL,  
sum NUMERIC(7, 2) NOT NULL,  
date DATE NOT NULL,  
is_payed boolean NOT NULL  
);
```

```
ALTER TABLE salary ADD CONSTRAINT fk_employee_salary  
FOREIGN KEY(employee_id) REFERENCES employee(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE IF NOT EXISTS orders (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
department_id INT NOT NULL,  
order_type_id INT NOT NULL,
```

```
deal_id INT NOT NULL,  
district_id INT NOT NULL,  
amount INT NOT NULL,  
date_of_beginning DATE NOT NULL,  
date_of_end DATE NOT NULL  
);
```

```
ALTER TABLE orders ADD CONSTRAINT fk_department_orders  
FOREIGN KEY(department_id) REFERENCES department(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
ALTER TABLE orders ADD CONSTRAINT fk_order_type_orders  
FOREIGN KEY(order_type_id) REFERENCES order_type(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
ALTER TABLE orders ADD CONSTRAINT fk_deal_orders  
FOREIGN KEY(deal_id) REFERENCES deal(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
ALTER TABLE orders ADD CONSTRAINT fk_district_orders  
FOREIGN KEY(district_id) REFERENCES district(id)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE IF NOT EXISTS material (  
id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
order_id INT NOT NULL,  
distributor_id INT NOT NULL,  
name VARCHAR(40) NOT NULL,  
amount INT NOT NULL  
);
```

```
ALTER TABLE material ADD CONSTRAINT fk_orders_material
FOREIGN KEY(order_id) REFERENCES orders(id)
ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
ALTER TABLE material ADD CONSTRAINT fk_distributor_material
FOREIGN KEY(distributor_id) REFERENCES distributor(id)
ON UPDATE CASCADE ON DELETE RESTRICT;
```

5.2 Імпортування даних в таблицю

Імпортуємо CSV-файли в БД:

```
coursework=# COPY client(name, phone_number) FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\client.csv' DELIMITER ',' CSV;
COPY 16
coursework=# SELECT * FROM client;
 id |      name      | phone_number
-----+-----+-----
  1 | name           | phone_number
  2 | Martena Juarez | (222) 776-7289
  3 | April Nieves   | 1-724-455-7127
  4 | Laura Hutchinson | (384) 472-1031
  5 | Gil Burgess    | 1-247-647-6243
  6 | Galvin Hester  | 1-342-522-1597
  7 | Kiana Benton   | 1-362-815-8858
  8 | Maryam Mcmillan | 1-552-896-6574
  9 | Lamar Cooper   | (972) 683-3158
 10 | Gabriel Carroll | 1-267-537-2311
 11 | Alma Petersen  | (696) 325-6954
 12 | Kameko Guzman  | 1-688-674-6546
 13 | Chester Lang    | 1-270-811-3778
 14 | Quon Neal      | 1-476-162-8332
 15 | Mallory Sweet   | 1-712-623-0457
 16 | Scarlett Dejesus | 1-449-691-2343
(16 rows)
```

Рисунок 5.2.1 - Імпортування даних в таблиці client за допомогою CSV-файлу

```
coursework=# COPY bank FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\bank.csv' DELIMITER ',' CSV;
COPY 11
coursework=# SELECT * FROM bank;
 id | name
-----+-----
  1 | Curabitur Egestas Nunc Associates
  2 | Sed Inc.
  3 | Molestie Foundation
  4 | Est Nunc Ullamcorper Foundation
  5 | Amet Industries
  6 | Nibh Institute
  7 | Vulputate Dui Nec Ltd
  8 | Quis Arcu Limited
  9 | Sed Sem Corporation
 10 | Et Netus LLC
 11 | Sem Nulla Associates
(11 rows)
```

Рисунок 5.2.2 - Імпортування даних в таблиці bank за допомогою CSV-файлу

```

coursework=# COPY receipt FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\receipt.csv' DELIMITER ',' CSV;
COPY 20
coursework=# COPY deal FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\deal.csv' DELIMITER ',' CSV;
COPY 17
coursework=# COPY order_type FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\order_type.csv' DELIMITER ',' CSV;
COPY 14

```

Рисунок 5.2.3 - Імпортування даних в таблиць reseipt, deal за допомогою CSV-файлів

```

coursework=# COPY order_type FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\order_type.csv' DELIMITER ',' CSV;
COPY 14
coursework=# COPY distributor FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\distributor.csv' DELIMITER ',' CSV;
COPY 15
coursework=# COPY district FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\district.csv' DELIMITER ',' CSV;
COPY 17
coursework=# COPY department FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\department.csv' DELIMITER ',' CSV;
COPY 10
coursework=# COPY employee FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\employee.csv' DELIMITER ',' CSV;
COPY 120
coursework=# COPY salary FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\salary.csv' DELIMITER ',' CSV;
COPY 100
coursework=# COPY orders FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\orders.csv' DELIMITER ',' CSV;
COPY 19
coursework=# COPY material FROM 'E:\Install\Projects\BASA_DANNYKH\Course work\CSV\material.csv' DELIMITER ',' CSV;
COPY 45

```

Рисунок 5.2.4 - Імпортування даних в таблиць order_type, distributor, district, department, employee, salary, orders, material за допомогою CSV-файлів

6 СТВОРЕННЯ КОРИСТУВАЧІВ БАЗИ ДАНИХ

6.1 Створення кадрового менеджера

```
Type "help" for help.  
  
coursework=# CREATE USER hr_manager;  
CREATE ROLE  
coursework=# GRANT ALL PRIVILEGES ON department, employee TO hr_manager;  
GRANT  
coursework=# CREATE USER accountant;  
CREATE ROLE
```

Рисунок 6.1 – Створення користувача кадрового менеджера та надання йому відповідних прав

6.2 Створення бухгалтера

```
coursework=# CREATE USER accountant;  
CREATE ROLE  
coursework=# GRANT ALL PRIVILEGES ON bank, receipt, salary TO accountant;  
GRANT  
coursework=# GRANT SELECT ON department, employee TO accountant;  
GRANT
```

Рисунок 6.2 – Створення користувача бухгалтера та надання йому відповідних прав

6.3 Створення агента рекламного агентства

```
coursework=# CREATE USER agent;  
CREATE ROLE  
coursework=# GRANT ALL PRIVILEGES ON deal, client, order_type, orders, distributor, material, district TO agent;  
GRANT  
coursework=# GRANT SELECT ON receipt, department TO agent;  
GRANT  
coursework=#
```

Рисунок 6.3 – Створення користувача агента рекламного агентства та надання йому відповідних прав

7 SQL ЗАПИТИ

7.1 Тригери

7.1.1 checkdate

Створення тригера для таблиці orders, щоб не допускати проміжок часу менше тижня на виконання замовлення при виконанні INSERT

```
CREATE OR REPLACE FUNCTION chkdtln()
RETURNS trigger
AS $$
DECLARE
    a interval;
BEGIN
    a = AGE(NEW.date_of_end, NEW.date_of_beginning);
    IF a < '7 days'
    THEN
        RAISE EXCEPTION 'Not enough time for order!';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER checkdate
BEFORE INSERT ON orders
FOR EACH ROW EXECUTE PROCEDURE chkdtln();
```



```
coursework=# INSERT INTO orders(department_id, order_type_id, deal_id, district_id, amount, date_of_beginning, date_of_end) VALUES(1, 1, 1, 1, 1, '2023-01-11', '2023-01-14');
ОШИБКА: Not enough time for order!
CONTEXT:  функция PL/pgSQL chkdtln(), строка 1, оператор RAISE
coursework=#
```

Рисунок 7.1 – Повідомлення про помилку, що надано замало часу на виконання замовлення

7.1.2 countmat

Створення тригера на таблицю material, щоб не допускати кількість матеріалів на одне замовлення більше 5

```
CREATE OR REPLACE FUNCTION cntmat()
```

```
RETURNS trigger
```

```
AS $$
```

```
DECLARE
```

```
    a integer;
```

```
BEGIN
```

```
    SELECT
```

```
        COUNT(*)
```

```
    INTO
```

```
        a
```

```
    FROM
```

```
        material
```

```
    WHERE
```

```
        order_id = NEW.order_id;
```

```
    IF a > 4
```

```
    THEN
```

```
        RAISE EXCEPTION 'Limit of purchases on the order is reached!';
```

```
    END IF;
```

```
    RETURN NEW;
```

END;

\$\$ LANGUAGE plpgsql;

CREATE TRIGGER countmat

BEFORE INSERT ON material

FOR EACH ROW EXECUTE PROCEDURE cntmat();

```
coursework=# SELECT order_id, COUNT(order_id) FROM material GROUP BY order_id;
```

order_id	count
11	2
9	4
15	2
3	1
17	2
5	3
4	3
10	2
6	3
13	4
2	3
16	3
7	3
1	2
18	5
8	3

(16 rows)

Рисунок 7.2 – Кількість замовлених позицій із матеріалами на кожне замовлення

```
coursework=# INSERT INTO material(order_id, distributor_id, name, amount) VALUES(18, 1, 'test', 1);
ОШИБКА: Limit of purchases on the order is reached!
CONTEXT: функция PL/pgSQL cntmat(), строка 1, оператор RAISE
coursework=#
```

Рисунок 7.3 – Повідомлення про помилку, що на замовлення за вказаним id замовлено максимальну кількість позицій із матеріалами.

7.1.3 checksum

Створення тригера на таблицю salary, що не допускає нарахування зарплатні за межами допустимих значень (від 5 до 20 тисяч)

```

CREATE OR REPLACE FUNCTION chksl()
RETURNS trigger
AS $$
BEGIN
    IF NEW.sum < 5000
    THEN
        RAISE EXCEPTION 'The sum is too small!';
    END IF;
    IF NEW.sum > 20000
    THEN
        RAISE EXCEPTION 'The sum is too big!';
    END IF; RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER checksum
BEFORE INSERT ON salary
FOR EACH ROW EXECUTE PROCEDURE chksl();

```

```

coursework=# INSERT INTO salary(employee_id, sum, date, is_payed) VALUES(1, 25000, '2022-12-12', '1');
ОШИБКА: The sum is too big!
CONTEXT: функция PL/pgSQL chksl(), строка 1, оператор RAISE
coursework=# INSERT INTO salary(employee_id, sum, date, is_payed) VALUES(1, 4000, '2022-12-12', '1');
ОШИБКА: The sum is too small!
CONTEXT: функция PL/pgSQL chksl(), строка 1, оператор RAISE
coursework=#

```

Рисунок 7.4 – Повідомлення про помилки, що в першому випадку сума
завелика, в другому - замала

7.1.4 ifsmallcom

Створення тригера на таблицю orders, щоб повідомляти, коли на створене замовлення назначений відділ з малою кількістю працівників (менше 10)

```
CREATE OR REPLACE FUNCTION ifsmldp()
  RETURNS trigger
  AS $$
  DECLARE
    a integer;
    n VARCHAR;
  BEGIN
    SELECT
      COUNT(*)
    INTO
      a
    FROM
      employee
    WHERE
      department_id = NEW.department_id;
    IF a < 10
    THEN
      SELECT
        name
      INTO
        n
      FROM
        department
      WHERE
        id = NEW.department_id;
      RAISE NOTICE '% has small amount of employees!', n;
    END IF;
    RETURN NEW;
  END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER ifsmallcom
```

```
AFTER INSERT ON orders
```

```
FOR EACH ROW EXECUTE PROCEDURE ifsmldp());
```

```
coursework=# SELECT department_id, COUNT(department_id) FROM employee GROUP BY department_id;
```

department_id	count
8	15
9	9
10	8
7	8
1	6
5	16
4	20
2	13
6	10
3	15

```
(10 rows)
```

Рисунок 7.5 – Кількість працівників у кожному відділі

```
coursework=# INSERT INTO orders(id, department_id, order_type_id, deal_id, district_id, amount, date_of_beginning, date_of_end) VALUES(22, 7, 1, 1, 1, 1, '2022-12-03', '2023-01-05');
ЗАМЕЧАНИЕ: Sit Amet Industries has small amount of employees!
INSERT 0 1
coursework=#
```

Рисунок 7.6 – Повідомлення, що у відділі, на який було назначено це замовлення, працівників менше 10

7.1.5 checkamemp

Створення тригера на таблицю employee, щоб не допускати кількість працівників у одному відділі більше 20

```
CREATE OR REPLACE FUNCTION chkszdp()
```

```
RETURNS trigger
```

```
AS $$
```

```
DECLARE
```

```
a integer;
```

```
n VARCHAR;
```

```
BEGIN
  SELECT
    COUNT(*)
  INTO
    a
  FROM
    employee
  WHERE
    department_id = NEW.department_id;
  IF a > 19
  THEN
    SELECT
      name
    INTO
      n
    FROM
      department
    WHERE
      id = NEW.department_id;
    RAISE EXCEPTION '% is already full!', n;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER checkamemp
  BEFORE INSERT ON employee
  FOR EACH ROW EXECUTE PROCEDURE chkszdp();
```

```
coursework=# INSERT INTO employee(name, phone_number, passport_id, department_id, position) VALUES ('Jack', '25-34-2343', 234, 4, 'senior');
ОШИБКА: Temporal Bibendum Consulting is already full!
CONTEXT: функция PL/pgSQL chkszdp(), строка 1, оператор RAISE
coursework=#
```

Рисунок 7.7 – Повідомлення про помилку, що у відділі вже нема місць для нових працівників

7.1.6 ispayed

Створення тригера на таблицю salary, щоб не видавати двічі зарплатню, що вже була видана

```
CREATE OR REPLACE FUNCTION ispyd()
```

```
RETURNS trigger
```

```
AS $$
```

```
DECLARE
```

```
    a boolean;
```

```
    b VARCHAR;
```

```
BEGIN
```

```
    SELECT
```

```
        is_payed
```

```
    INTO
```

```
        a
```

```
    FROM
```

```
        salary
```

```
    WHERE
```

```
        id = NEW.id;
```

```
    IF a = '1' AND NEW.is_payed = '1'
```

```
    THEN
```

```
        SELECT
```

```
            name
```

```
        INTO
```

```

        b
    FROM
        employee
    WHERE
        id = NEW.employee_id;
    RAISE EXCEPTION '% has received this salary!', b;
END IF;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER ispayed
BEFORE UPDATE ON salary
FOR EACH ROW EXECUTE PROCEDURE ispyd();

```

```

coursework=# SELECT * FROM salary LIMIT 5;
 id | employee_id |  sum   |  date   | is_payed
-----+-----+-----+-----+-----
  1 |          76 | 5260.00 | 2022-12-27 | t
  2 |         105 | 18007.50 | 2023-01-03 | t
  3 |          42 | 10788.00 | 2022-12-25 | f
  4 |         102 | 17025.00 | 2022-12-11 | f
  5 |          70 | 18652.00 | 2022-12-18 | f
(5 rows)

```

Рисунок 7.8 – Перелік нарахованих зарплат працівникам

```

coursework=# UPDATE salary SET is_payed = '1' WHERE id = 1;
ОШИБКА: Caldwell Barlow has received this salary!
CONTEXT:  функция PL/pgSQL ispyd(), строка 1, оператор RAISE
coursework=#

```

Рисунок 7.9 – Повідомлення про помилку, що нарахована зарплата, яку намагались видати, вже була видана працівнику

7.2 Функції та процедури

7.2.1 outcome

Створення функції, що підрахує чистий прибуток агентства (з урахуванням усіх нарахованих зарплат) за вказаний проміжок часу

```
CREATE OR REPLACE FUNCTION outcome(dat1 DATE, dat2 DATE)
RETURNS NUMERIC
AS $$
DECLARE
    a NUMERIC;
    b NUMERIC;
BEGIN
    SELECT
        SUM(sum)
    INTO
        a
    FROM
        receipt
    WHERE
        date > dat1 AND date < dat2;
    SELECT
        SUM(sum)
    INTO
        b
    FROM
        salary
    WHERE
        date > dat1 AND date < dat2;
    a = a - b;
    RETURN a;
```

END;

\$\$ LANGUAGE plpgsql;

```
coursework=# SELECT SUM(sum) FROM receipt WHERE date > '2022-12-15' AND date < '2023-01-15';
sum
-----
6364649.30
(1 row)

coursework=# SELECT SUM(sum) FROM salary WHERE date > '2022-12-15' AND date < '2023-01-15';
sum
-----
683245.01
(1 row)
```

Рисунок 7.10 – Загальні суми грошей, які були зароблені та які були нараховані працівникам в якості зарплат в період з середини грудня 22 року до середини січня 23 року

```
coursework=# SELECT outcome('2022-12-15', '2023-01-15');
outcome
-----
5681404.29
(1 row)
```

Рисунок 7.11 – Чистий заробіток агентства в період з середини грудня 22 року до середини січня 23 року

7.2.2 pop_ord_type

Створення процедури для швидкого визначення найпопулярнішого типу замовлення серед замовників

```
CREATE OR REPLACE PROCEDURE pop_ord_type()
AS $$
DECLARE
    a INT;
    b VARCHAR;
BEGIN
```

```

SELECT
    order_type_id
INTO
    a
FROM
    orders
GROUP BY
    order_type_id
ORDER BY
    COUNT(order_type_id) DESC
LIMIT 1;
SELECT
    name
INTO
    b
FROM
    order_type
WHERE
    id = a;
RAISE NOTICE 'ID:%, Name:%', a, b;
END;
$$ LANGUAGE plpgsql;

```

```

coursework=# SELECT order_type_id, COUNT(order_type_id) FROM orders GROUP BY order_type_id ORDER BY COUNT(order_type_id) DESC;
 order_type_id | count
-----+-----
          11 |      5
           2 |      3
          12 |      3
           6 |      2
          10 |      2
           4 |      1
           8 |      1
           1 |      1
           7 |      1
          14 |      1
(10 rows)

```

Рисунок 7.12 – Перелік кількості замовлень із кожним типом замовлення

```
coursework=# CALL pop_ord_type();  
ЗАМЕЧАНИЕ: ID:11, Name:#e51975  
CALL  
coursework=# _
```

Рисунок 7.13 – Результат виконання процедури pop_ord_type для визначення найпопулярнішого типу замовлення

7.2.3 pop_dis

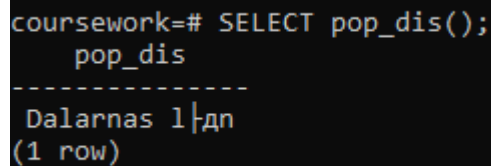
Створення функції для швидкого визначення найпопулярнішого району для розміщення своєї реклами серед замовників

```
CREATE OR REPLACE FUNCTION pop_dis()  
RETURNS VARCHAR  
AS $$  
DECLARE  
    a INT;  
    b VARCHAR;  
BEGIN  
    SELECT  
        district_id  
    INTO  
        a  
    FROM  
        orders  
    GROUP BY  
        district_id  
    ORDER BY  
        COUNT(district_id) DESC  
    LIMIT 1;
```

```

SELECT
    name
INTO
    b
FROM
    district
WHERE
    id = a;
RETURN b;
END;
$$ LANGUAGE plpgsql;

```



```

coursework=# SELECT pop_dis();
 pop_dis 
-----
Dalarnas 1 | 1
(1 row)

```

Рисунок 7.14 – Результат виконання функції pop_dis

7.2.4 pop_distrib

Створення процедури для швидкого визначення дистриб'ютора матеріалів, у якого агентство найчастіше робить замовлення

```

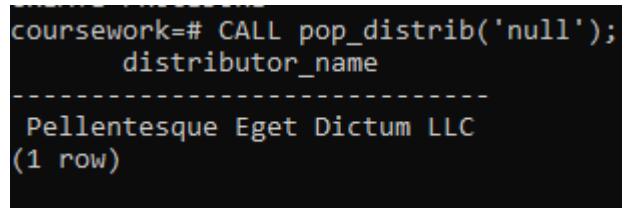
CREATE OR REPLACE PROCEDURE pop_distrib(Distributor_name INOUT
VARCHAR)
AS $$
DECLARE
    a INT;
BEGIN
    SELECT

```

```

        distributor_id
    INTO
        a
    FROM
        material
    GROUP BY
        distributor_id
    ORDER BY
        COUNT(distributor_id) DESC
    LIMIT 1;
    SELECT
        name
    INTO
        Distributor_name
    FROM
        distributor
    WHERE
        id = a;
END;
$$ LANGUAGE plpgsql;

```



```

coursework=# CALL pop_distrib('null');
 distributor_name
-----
Pellentesque Eget Dictum LLC
(1 row)

```

Рисунок 7.15 - Результат виконання процедури pop_distrib

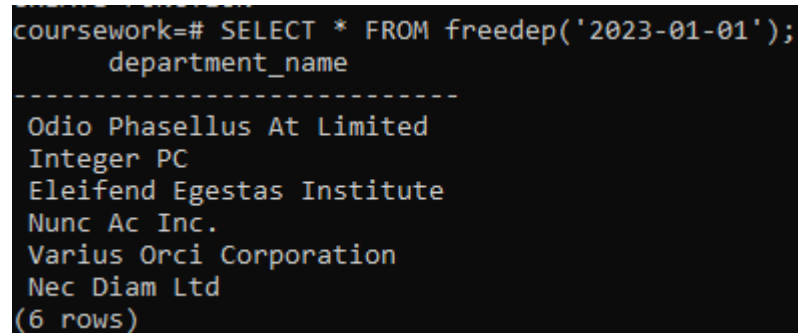
7.2.5 freedep

Створення функції для визначення переліку вільних відділів на вказану дату

```

CREATE OR REPLACE FUNCTION freedep(dat DATE)
  RETURNS TABLE(department_name VARCHAR)
  AS $$
  BEGIN
    RETURN QUERY
      SELECT
        name
      FROM
        department
      WHERE
        id NOT IN (
          SELECT
            department_id
          FROM
            orders
          WHERE
            date_of_beginning < dat AND date_of_end > dat
        );
  END;
$$ LANGUAGE plpgsql;

```



```

coursework=# SELECT * FROM freedep('2023-01-01');
 department_name
-----
Odio Phasellus At Limited
Integer PC
Eleifend Egestas Institute
Nunc Ac Inc.
Varius Orci Corporation
Nec Diam Ltd
(6 rows)

```

Рисунок 7.16 – Результат виконання функції freedep

7.2.6 unpaid

Створення функції для визначення усіх зарплат, що не були видані працівникам вказаного відділу

```
CREATE OR REPLACE FUNCTION unpaid(depid INT)
RETURNS TABLE(employee_name VARCHAR, sum NUMERIC, date DATE)
AS $$
BEGIN
RETURN QUERY
SELECT
    employee.name AS nam,
    salary.sum AS summoney,
    salary.date AS dat
FROM
    salary,
    employee
WHERE
    employee.id = salary.employee_id AND salary.is_payed = '0' AND
employee.department_id = depid;
END;
$$ LANGUAGE plpgsql;
```

```
coursework=# SELECT * FROM unpaid(2);
 employee_name |      sum      |      date
-----+-----+-----
 Nevada Nixon  | 17025.00      | 2022-12-11
 Tatiana Reyes | 18071.00      | 2022-12-11
 Kenyon Hunter | 16441.00      | 2023-01-30
 Chaney Palmer | 17729.00      | 2023-01-10
 Caldwell Ray  | 13798.00      | 2023-01-21
 Quentin Skinner | 14517.56     | 2023-01-15
 Chloe Woods   | 18987.00      | 2023-01-22
 Holly Mcleod  | 8389.00       | 2023-01-21
(8 rows)
```

Рисунок 7.17 – Результат виконання функції unpaid

7.3 Запити

7.3.1 Список усіх замовлень

Створення списку усіх замовлень із вказанням району, типу замовлення, розміру та термінів виконання

```
SELECT
    id,
    (
        SELECT
            name AS type
        FROM
            order_type
        WHERE
            order_type.id = orders.order_type_id
    ),
    amount,
    (
        SELECT
            name AS district
        FROM
            district
        WHERE
            district.id = orders.district_id
    ),
    date_of_beginning,
    date_of_end
FROM
    orders;
```

id	type	amount	district	date_of_beginning	date_of_end
1	#bdc3f9	23	Valparaíso	2023-01-11	2023-01-28
2	#a9e062	41	East Region	2022-12-29	2023-01-20
3	#052770	29	FATA	2022-12-02	2022-12-25
4	#ff916d	26	Sachsen	2022-12-20	2022-12-30
5	#63e296	42	Warmińsko-mazurskie	2022-12-17	2022-12-25
6	#bdc3f9	49	Zuid Holland	2023-01-05	2023-01-19
7	#d1f49f	41	Dalarnas län	2022-12-07	2022-12-24
8	#e51975	8	Limburg	2022-12-14	2022-12-24
9	#052770	40	Davao Region	2022-12-24	2023-01-09
10	#e51975	35	Davao Region	2023-01-04	2023-01-13
11	#ff916d	27	Warmińsko-mazurskie	2023-01-01	2023-01-28
12	#e51975	4	Dalarnas län	2023-01-09	2023-01-23
13	#a9e062	11	Dalarnas län	2023-01-01	2023-02-01
14	#b51955	19	Zuid Holland	2022-12-12	2022-12-27
15	#e51975	48	Limburg	2023-01-04	2023-02-01
16	#052770	7	South Gyeongsang	2022-12-04	2023-01-08
17	#e51975	12	Sachsen	2023-01-05	2023-01-22
18	#a9e062	49	South Gyeongsang	2023-01-13	2023-01-29
19	#264ee0	6	Zuid Holland	2022-12-13	2022-12-28
20	#7c7cf9	1	Santa Catarina	2022-12-03	2023-01-05

(20 rows)

Рисунок 7.18 – Список усіх замовлень

7.3.2 Список усіх матеріалів, які були замовлені у дистриб'юторів для виконання замовлень

Створення списку усіх матеріалів, що були замовленні у дистриб'юторів для виконання замовлень

```

SELECT
  id,
  (
    SELECT
      id AS order_id
    FROM
      orders
    WHERE
      orders.id = material.order_id
  ),

```

```

name AS material,
amount,
(
    SELECT
        name AS distributor
    FROM
        distributor
    WHERE
        distributor.id = material.distributor_id
)
FROM
    material;

```

id	order_id	material	amount	distributor
1	18	#b20200	8	Accumsan Sed LLC
2	5	#f28c60	15	Molestie Tortor Nibh Corp.
3	4	#edb85c	7	Hendrerit Consectetuer Foundation
4	7	#1614b7	9	Pellentesque Eget Dictum LLC
5	17	#21117c	15	Pellentesque Eget Dictum LLC
6	8	#a985d3	3	Hendrerit Consectetuer Foundation
7	3	#119162	25	Dolor LLC
8	18	#8251ba	25	Ornare Placerat Orci Foundation
9	9	#9dffb9b	20	Hendrerit Consectetuer Foundation
10	9	#cbef56	16	Accumsan Sed LLC
11	13	#703ce0	7	Dui In Sodales PC
12	18	#fc5fd5	23	Pellentesque Eget Dictum LLC
13	18	#d18ee5	7	Erat In Consulting
14	2	#c7b8f9	12	Dolor LLC
15	15	#86f751	21	Ornare Egestas Corp.
16	9	#e0c600	2	Erat In Consulting
17	17	#5d22cc	10	Pellentesque Eget Dictum LLC
18	7	#316eaf	12	Molestie Tortor Nibh Corp.
19	6	#7035ce	10	Erat Vivamus Corporation
20	6	#599fe0	22	Pellentesque Eget Dictum LLC
21	9	#2a56a3	5	Sem Corp.
22	16	#fc6dfff	5	Erat In Consulting
23	10	#083068	1	Phasellus Corp.
24	4	#c3e27f	8	Phasellus Corp.
25	8	#a1ad1a	12	Erat In Consulting
26	13	#3ec148	20	Molestie Tortor Nibh Corp.
27	5	#378ad3	15	Dui In Sodales PC
28	11	#eda8e7	2	Phasellus Corp.
29	4	#14ba3d	22	Accumsan Sed LLC
30	2	#d838a5	9	Molestie Tortor Nibh Corp.
31	8	#b7e2f7	25	Phasellus Corp.
32	1	#2b8c0b	14	Ornare Placerat Orci Foundation
33	15	#eb3af4	10	Accumsan Sed LLC
34	16	#81f4a4	23	Erat In Consulting
35	18	#ce02c4	15	Phasellus Corp.
36	7	#699ace	4	Pellentesque Eget Dictum LLC
37	11	#446ddd	14	Erat Vivamus Corporation
38	5	#0c0cff	7	Tortor Integer Aliquam Corporation
39	13	#e8dc3a	21	Dui In Sodales PC
40	13	#e8b468	13	Sem Corp.
41	10	#30e8d8	3	Erat Vivamus Corporation
42	6	#8946db	19	Ornare Placerat Orci Foundation
43	16	#59db85	13	Ornare Egestas Corp.
44	2	#efa5b6	24	Pellentesque Eget Dictum LLC
45	1	#bf8601	2	Tortor Integer Aliquam Corporation

(45 rows)

Рисунок 7.19 – Список усіх матеріалів, замовлених для замовлень

7.3.3 Список усіх угод, які були укладені із замовниками на замовлення

Створення списку усіх угод, що були укладені із замовниками, дат укладення та сум на оплату відповідних замовлень

```
SELECT
    orders.id AS order_id,
    deal.number_of_document,
    deal.date,
    receipt.sum,
    client.name AS name_of_client
FROM
    orders
INNER JOIN
    deal
ON
    orders.deal_id = deal.id
INNER JOIN
    receipt
ON
    deal.receipt_id = receipt.id
INNER JOIN
    client
ON
    deal.client_id = client.id;
```

order_id	number_of_document	date	sum	name_of_client
1	50	2023-01-19	637932.00	Galvin Hester
2	23	2022-12-15	457115.00	Martena Juarez
3	39	2022-12-05	502982.00	Chester Lang
4	10	2022-12-16	258506.00	Quon Neal
5	17	2023-01-09	457115.00	Lamar Cooper
6	34	2023-01-30	77990.50	Quon Neal
7	50	2023-01-19	637932.00	Galvin Hester
8	23	2022-12-15	457115.00	Martena Juarez
9	17	2023-01-09	457115.00	Lamar Cooper
10	18	2023-01-30	252448.00	Galvin Hester
11	35	2023-01-23	252448.00	Lamar Cooper
12	23	2022-12-15	457115.00	Martena Juarez
13	32	2023-01-17	457115.00	Gabriel Carroll
14	45	2023-01-22	238286.00	Galvin Hester
15	50	2023-01-19	637932.00	Galvin Hester
16	11	2022-12-23	953783.00	name
17	13	2022-12-25	530774.00	Laura Hutchinson
18	39	2022-12-05	502982.00	Chester Lang
19	11	2022-12-23	953783.00	name
20	20	2023-01-25	400049.00	Alma Petersen

(20 rows)

Рисунок 7.20 – Список всіх укладених із замовниками угод на замовлення

7.3.4 Список усіх матеріалів, які треба доставити у відповідні відділи на вказані замовлення

Створення списку усіх необхідних матеріалів, які треба надати відділам для виконання замовлень

SELECT

material.id AS material_id,
material.name AS material_name,
material.amount,
orders.id AS order_id,
department.name AS department

FROM

material

INNER JOIN

orders

ON

material.order_id = orders.id

INNER JOIN

department

ON

orders.department_id = department.id;

material_id	material_name	amount	order_id	department
1	#b20200	8	18	Varius Orci Corporation
2	#f28c60	15	5	Eleifend Egestas Institute
3	#edb85c	7	4	Varius Orci Corporation
4	#1614b7	9	7	Integer PC
5	#21117c	15	17	Odio Phasellus At Limited
6	#a985d3	3	8	Integer PC
7	#119162	25	3	Sit Amet Industries
8	#8251ba	25	18	Varius Orci Corporation
9	#9dff9b	20	9	Tempor Bibendum Consulting
10	#cbef56	16	9	Tempor Bibendum Consulting
11	#703ce0	7	13	Tempor Bibendum Consulting
12	#fc5fd5	23	18	Varius Orci Corporation
13	#d18ee5	7	18	Varius Orci Corporation
14	#c7b8f9	12	2	Erat PC
15	#86f751	21	15	Nec Diam Corporation
16	#e0c600	2	9	Tempor Bibendum Consulting
17	#5d22cc	10	17	Odio Phasellus At Limited
18	#316eaf	12	7	Integer PC
19	#7035ce	10	6	Tempor Bibendum Consulting
20	#599fe0	22	6	Tempor Bibendum Consulting
21	#2a56a3	5	9	Tempor Bibendum Consulting
22	#fc6dff	5	16	Nec Diam Corporation
23	#083068	1	10	Eleifend Egestas Institute
24	#c3e27f	8	4	Varius Orci Corporation
25	#a1ad1a	12	8	Integer PC
26	#3ec148	20	13	Tempor Bibendum Consulting
27	#378ad3	15	5	Eleifend Egestas Institute
28	#eda8e7	2	11	Integer PC
29	#14ba3d	22	4	Varius Orci Corporation
30	#d838a5	9	2	Erat PC
31	#b7e2f7	25	8	Integer PC
32	#2b8c0b	14	1	Odio Phasellus At Limited
33	#eb3af4	10	15	Nec Diam Corporation
34	#81f4a4	23	16	Nec Diam Corporation
35	#ce02c4	15	18	Varius Orci Corporation
36	#699ace	4	7	Integer PC
37	#446ddd	14	11	Integer PC
38	#0c0cff	7	5	Eleifend Egestas Institute
39	#e8dc3a	21	13	Tempor Bibendum Consulting
40	#e8b468	13	13	Tempor Bibendum Consulting
41	#30e8d8	3	10	Eleifend Egestas Institute
42	#8946db	19	6	Tempor Bibendum Consulting
43	#59db85	13	16	Nec Diam Corporation
44	#efa5b6	24	2	Erat PC
45	#bf8601	2	1	Odio Phasellus At Limited

(45 rows)

Рисунок 7.21 – Список матеріалів, які треба надати відділам для виконання відповідних замовлень

7.3.5 Список усіх виданих зарплат робітникам

Створення списку усіх зарплат, виданих робітникам відділів, по датам нарахування

```
SELECT
    employee.name AS employee,
    department.name AS department,
    salary.sum, salary.date
FROM
    employee
INNER JOIN
    department
ON
    employee.department_id = department.id
INNER JOIN
    salary
ON
    salary.employee_id = employee.id
WHERE
    salary.is_payed = '1';
```

employee	department	sum	date
Caldwell Barlow	Nunc Ac Inc.	5260.00	2022-12-27
Julian Curry	Sit Amet Industries	18007.50	2023-01-03
Chelsea Casey	Eleifend Egestas Institute	7396.00	2022-12-06
Mechelle Byers	Nec Diam Ltd	7073.00	2022-12-15
Galena Burke	Sit Amet Industries	15744.00	2022-12-28
Brooke Weber	Erat PC	19592.00	2023-01-04
Amir Higgins	Nunc Ac Inc.	16765.00	2022-12-16
Aquila Trevino	Tempor Bibendum Consulting	6616.00	2022-12-19
Elton Johnston	Nunc Ac Inc.	16781.00	2023-01-21
Calvin Crawford	Varius Orci Corporation	19895.00	2022-12-31
Kai Sears	Nec Diam Ltd	7898.00	2023-01-22
Wallace Marquez	Varius Orci Corporation	11928.00	2023-01-20
Shaine Coffey	Tempor Bibendum Consulting	18894.00	2022-12-12
Francis Melendez	Nec Diam Ltd	7402.00	2023-01-03
Levi Hooper	Odio Phasellus At Limited	12225.00	2022-12-15
Len Fields	Tempor Bibendum Consulting	8978.00	2022-12-16
Aileen Holder	Tempor Bibendum Consulting	6845.00	2022-12-25
Risa Solomon	Eleifend Egestas Institute	9288.00	2023-02-01
Nevada Nixon	Integer PC	8442.00	2022-12-18
Quin Larsen	Varius Orci Corporation	10983.00	2023-01-05
Melvin Mcfadden	Varius Orci Corporation	17051.00	2023-01-20
Len Fields	Tempor Bibendum Consulting	16006.00	2022-12-14
Shaeleigh Shaw	Varius Orci Corporation	14949.00	2022-12-17
Abigail Glover	Eleifend Egestas Institute	7578.00	2022-12-21
Lara Ramirez	Varius Orci Corporation	15759.00	2023-02-01
Callum Schmidt	Tempor Bibendum Consulting	6628.00	2022-12-29
Jin Salazar	Eleifend Egestas Institute	14626.00	2022-12-23
Clarke Hodges	Nunc Ac Inc.	17567.00	2023-01-18
Charles Duran	Sit Amet Industries	19258.00	2023-01-29
Quentin Skinner	Integer PC	10538.00	2023-01-10
Hector Anderson	Nunc Ac Inc.	18322.00	2023-01-13
Caldwell Barlow	Nunc Ac Inc.	18677.00	2023-01-04
Francesca Coffey	Eleifend Egestas Institute	14848.00	2023-01-23
Georgia Kaufman	Varius Orci Corporation	16054.00	2022-12-13
Jenette Velasquez	Odio Phasellus At Limited	13886.00	2023-01-07
Jenette Velasquez	Odio Phasellus At Limited	9328.00	2022-12-23
Wylie Walton	Erat PC	12464.00	2022-12-27
Xena Warren	Tempor Bibendum Consulting	15756.00	2022-12-03
Brooke Weber	Erat PC	18124.00	2023-01-12
Gisela Woods	Sit Amet Industries	12037.00	2023-01-05
Aileen Holder	Tempor Bibendum Consulting	9861.00	2022-12-03
Sean Combs	Nec Diam Corporation	15557.46	2023-01-23
Chastity Stark	Varius Orci Corporation	15182.00	2022-12-18
Wylie Walton	Erat PC	17420.00	2023-01-09
Julian Curry	Sit Amet Industries	16994.00	2023-01-14
Gwendolyn Gallagher	Tempor Bibendum Consulting	10889.00	2023-01-25
Travis Delaney	Erat PC	5834.00	2022-12-18
Ira Short	Erat PC	17576.00	2023-01-29
Ira Short	Erat PC	16788.00	2022-12-15
Troy Pickett	Nec Diam Corporation	7814.00	2023-01-29
Francis Melendez	Nec Diam Ltd	11047.00	2022-12-19
Gemma Terry	Tempor Bibendum Consulting	19649.00	2023-01-08
Aristotle Bush	Eleifend Egestas Institute	15615.00	2023-01-18

(53 rows)

Рисунок 7.22 – Список усіх виданих зарплат робітникам

7.3.6 Список відділів та загальної кількості виданих замовлень кожному відділу

Створення списку відділів та загальної кількості назначених замовлень на кожний із них

```
SELECT
    name AS department,
    (
        SELECT
            COUNT(department_id) AS amount
        FROM
            orders
        WHERE
            orders.department_id = department.id
        GROUP BY
            department_id
    )
FROM
    department
ORDER BY
    amount DESC;
```

department	amount
Nec Diam Ltd	4
Integer PC	3
Eleifend Egestas Institute	3
Tempor Bibendum Consulting	2
Odio Phasellus At Limited	2
Varius Orci Corporation	2
Nec Diam Corporation	2
Sit Amet Industries	2
Nunc Ac Inc.	1
Erat PC	1
(10 rows)	

Рисунок 7.23 – Список відділів та кількості назначених замовлень на них

7.4 Представлення

7.4.1 stuff

Створення представлення для відображення списку всіх працівників із вказанням відділу, у якому вони працюють

```
CREATE VIEW stuff AS
```

```
SELECT
```

```
    name,
```

```
    passport_id,
```

```
    phone_number,
```

```
    (
```

```
        SELECT
```

```
            name AS department
```

```
        FROM
```

```
            department
```

```
        WHERE
```

```
            department.id = employee.department_id
```

```
    ),
```

```
    position
```

```
FROM
```

```
    employee;
```

```
coursework=# SELECT * FROM stuff ORDER BY name LIMIT 10;
```

name	passport_id	phone_number	department	position
Abbot Potts	330	(841) 429-0679	Eleifend Egestas Institute	massa. Suspendisse
Abigail Glover	970	(437) 497-5685	Eleifend Egestas Institute	dictum. Proin
Adam Stewart	903	1-414-687-6447	Erat PC	vitae velit
Aileen Holder	1423	(652) 757-8930	Tempor Bibendum Consulting	lacus. Quisque
Amir Higgins	296	(266) 741-8462	Nunc Ac Inc.	quis pede.
Anthony Mcgee	1108	(724) 218-3711	Erat PC	Donec est.
Aquila Trevino	694	(898) 495-4331	Tempor Bibendum Consulting	Ut tincidunt
Aristotle Bush	1026	(663) 534-3487	Eleifend Egestas Institute	et ultrices
Blossom Salazar	1459	1-645-994-7210	Erat PC	fermentum fermentum
Branden Colon	1356	1-135-555-3818	Varius Orci Corporation	sagittis placerat.

```
(10 rows)
```

Рисунок 7.24 – Вибірка 10 робітників із представлення stuff

7.4.2 transaction

Створення представлення для відображення всіх транзакцій з банків на рахунок рекламного агентства

```
CREATE VIEW transaction AS
```

```
SELECT
```

```
  id,
```

```
  (
```

```
    SELECT
```

```
      name AS bank
```

```
    FROM
```

```
      bank
```

```
    WHERE
```

```
      bank.id = receipt.bank_id
```

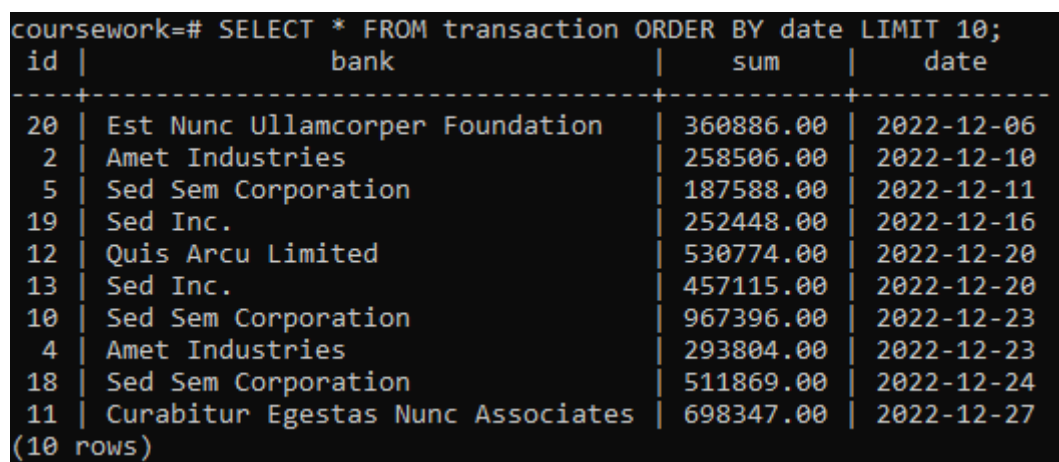
```
  ),
```

```
  sum,
```

```
  date
```

```
FROM
```

```
receipt;
```



```
coursework=# SELECT * FROM transaction ORDER BY date LIMIT 10;
```

id	bank	sum	date
20	Est Nunc Ullamcorper Foundation	360886.00	2022-12-06
2	Amet Industries	258506.00	2022-12-10
5	Sed Sem Corporation	187588.00	2022-12-11
19	Sed Inc.	252448.00	2022-12-16
12	Quis Arcu Limited	530774.00	2022-12-20
13	Sed Inc.	457115.00	2022-12-20
10	Sed Sem Corporation	967396.00	2022-12-23
4	Amet Industries	293804.00	2022-12-23
18	Sed Sem Corporation	511869.00	2022-12-24
11	Curabitur Egestas Nunc Associates	698347.00	2022-12-27

(10 rows)

Рисунок 7.25 – Вибірка 10 транзакцій із представлення transaction

7.5 Індокси

Виконаємо простий SQL-запит, де покажемо ім'я клієнта, номер телефону та номер документу угоди. Спочатку без використання індекса:

```
EXPLAIN ANALYZE
```

```
SELECT
```

```
    client.name AS client,
```

```
    client.phone_number,
```

```
    deal.number_of_document AS document
```

```
FROM
```

```
    client
```

```
INNER JOIN
```

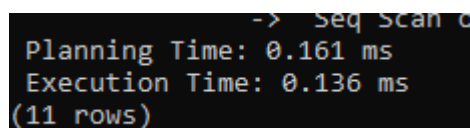
```
    deal
```

```
ON
```

```
    deal.client_id = client.id
```

```
ORDER BY
```

```
    client;
```

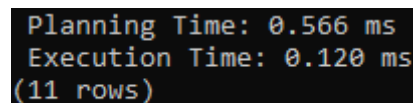


```
-> Seq Scan o  
Planning Time: 0.161 ms  
Execution Time: 0.136 ms  
(11 rows)
```

Рисунок 7.26 - Запит без використання індексів

Застосуємо індекси:

```
CREATE INDEX client_index ON client (name, phone_number);
```



```
Planning Time: 0.566 ms  
Execution Time: 0.120 ms  
(11 rows)
```

Рисунок 7.27 – Запит із використанням індексів

Бачимо, що ефективність виконання запиту із застосуванням індексів збільшилася у 1.13 разів. Ця різниця стане ще помітніше на великих базах даних. Отже, індекси є важливими у роботі з базами даних.

ВИСНОВОК

Бази даних — це важливий інструмент для ефективної роботи будь-якого сервісу. Вони дозволяють зручно та швидко зберігати інформацію та обробляти її, щоб виявити певні тренди споживчого ринку.

У цій роботі було спроектовано базу даних для підтримки діяльності рекламного агентства. У ході написання праці було вивчено предметне середовище, бізнес-правила, створено ER-діаграму, реляційну модель бази даних у третій нормальній формі. Також були написані SQL-скрипти, користувачі бази даних, тригери, представлення, функції та процедури.

ПЕРЕЛІК ПОСИЛАНЬ

1. <https://www.postgresql.org/docs/>
2. <https://www.postgresqltutorial.com/>