

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 торгового майданчика для проведення електронних
закупівель

Студента 2 курсу ІП-22 групи
спеціальності 121 «Інженерія програмного
забезпечення»

Бадиги Б. В.

(прізвище та ініціали)

Керівник старший викладач Марченко О.І.
(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 – 2023 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-22 Семестр 3

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

Бадизі Богдану Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи База даних торгового майданчика для проведення
електронних закупівель

керівник роботи старший викладач Марченко Олена Іванівна

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

2. Строк подання студентом роботи 31.12.2023

3. Вихідні дані до роботи завдання на розробку бази даних торгового
майданчика для проведення електронних закупівель

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

1) Аналіз предметного середовища

2) Побудова ER-моделі

3) Побудова реляційної схеми з ER-моделі

4) Створення бази даних, у форматі обраної системи управління базою даних

5) Створення користувачів бази даних

6) Імпорт даних з використанням засобів СУБД в створену базу даних

7) Створення мовою SQL запитів

8) Оптимізація роботи запитів

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

6. Дата видачі завдання 31.11.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Аналіз предметного середовища	03.12.2023	
2	Побудова ER-моделі	03.12.2023	
3	Побудова реляційної схеми з ER-моделі	03.12.2023	
4	Створення бази даних, у форматі обраної системи управління базою даних	15.12.2023	
5	Створення користувачів бази даних	15.12.2023	
6	Імпорт даних з використанням засобів СУБД в створену базу даних	15.12.2023	
7	Створення мовою SQL запитів	21.12.2023	
8	Оптимізація роботи запитів	28.12.2023	
9	Оформлення пояснювальної записки	29.12.2023	
10	Захист курсової роботи	30.12.2023	

Студент

_____ Бадига Б. В.
(підпис) (прізвище та ініціали)

Керівник роботи

_____ Марченко О.І.
(підпис) (прізвище та ініціали)

ЗМІСТ

1. ВСТУП	2
2. ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	3
3. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ	6
4. ПОСТАНОВКА ЗАВДАННЯ	8
5. ПОБУДОВА ERD ТА РЕЛЯЦІЙНОЇ МОДЕЛІ	9
6. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ.....	22
7. РЕАЛІЗАЦІЯ РОЛЕЙ ТА КОРИСТУВАЧІВ.....	27
7.1. Створення ролей.....	27
7.2. Надання привілеїв для ролі аналітика даних	27
7.3. Надання привілеїв для ролі менеджера постачальників	27
7.4. Надання привілеїв для ролі менеджера клієнтів.....	27
7.5. Надання привілеїв для ролі бухгалтера	28
7.6. Створення користувачів	28
8. РЕАЛІЗАЦІЯ ТРИГЕРІВ.....	30
9. РЕАЛІЗАЦІЯ ПРЕДСТАВЛЕНЬ.....	42
10. РЕАЛІЗАЦІЯ ЗАПИТІВ	50
11. РЕАЛІЗАЦІЯ ФУНКЦІЙ ТА ПРОЦЕДУР.....	77
12. ОПТИМІЗАЦІЯ.....	92
13. ВИСНОВКИ	100
14. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	101
ДОДАТОК А.....	102

1. ВСТУП

У сучасному світі, де цифрові технології стають все більш важливими в кожній сфері діяльності, електронні закупівлі перетворюються на невід’ємну частину торгових відносин. В цьому контексті, база даних торгового майданчика для проведення електронних закупівель є ключовим елементом, що забезпечує ефективність, прозорість та безпеку усієї торговельної системи. Вона не просто зберігає інформацію про товари, постачальників та транзакції, але й гарантує гнучкість управління процесами, від пошуку товарів до заключення контрактів.

Важливість такої бази даних у сучасному економічному просторі не можна переоцінити. Вона дозволяє автоматизувати процеси закупівель, знижуючи ризики людських помилок, прискорює обробку даних та забезпечує ефективність у прийнятті рішень. Більше того, вона сприяє прозорості угод, що є критично важливим для запобігання корупції та забезпечення справедливої конкуренції між постачальниками.

Система повинна бути досконало налаштована для відповідності вимогам торгового майданчика, що включає в себе не тільки зберігання даних про товари та учасників, але й інтеграцію з іншими платформами та системами для забезпечення широких можливостей аналітики та звітності. Це створює умови для більш ефективної взаємодії між всіма учасниками ринку, відкриваючи нові горизонти для розвитку електронних торгів та посилення їхньої ролі в економіці.

Мета цієї курсової роботи – сформулювати вимоги до бази даних, проаналізувати їх, проаналізувати предметне середовище, існуючі програмні продукти, розробити ER-модель, розробити базу даних та забезпечити її правильну роботу.

Всі файли, створені в ході виконання курсової роботи, зберігаються на репозиторії в GitHub (<https://github.com/BogBad/DB-Coursework>).

2. ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА

Предметна область бази даних торгового майданчика для проведення електронних закупівель охоплює широкий спектр понять і механізмів, вимагає глибокого розуміння принципів роботи. Створення та управління базою даних для такого майданчика стає стратегічно важливим завданням, спрямованим на забезпечення прозорості, ефективності та надійності електронних закупівель.

Охоплюється широкий спектр аспектів, від управління та аналізу даних про товари та послуги до забезпечення безпеки та конфіденційності інформації. Важливо зазначити, що торгові майданчики відіграють ключову роль у підтримці відкритості ринків та конкурентоспроможності бізнесу. Така база даних повинна ефективно враховувати потреби різноманітних учасників процесу – від державних органів та комерційних підприємств до постачальників та покупців.

Завдання перед цією базою даних включають в себе не лише збір та зберігання інформації про торгові процедури, але й реалізацію механізмів контролю, аналізу та забезпечення взаємодії між усіма учасниками системи. Ця система повинна не лише реєструвати замовлення, але й аналізувати потоки даних, прогнозувати тенденції закупівель, оптимізувати процеси пошуку товарів та послуг, а також надавати інструменти для ефективного управління відносинами з клієнтами та постачальниками. Аналіз цієї предметної області дозволить визначити необхідність розробки та впровадження нових інструментів для автоматизації та оптимізації процесів закупівель, що, в свою чергу, сприятиме підвищенню ефективності використання державних та комерційних ресурсів.

Поговоримо конкретно як це працює. Спочатку людина реєструє себе у базі даних. Для цього їй потрібно ввести трохи інформації про себе, а саме: ім'я, прізвище, адресу, за якою вона зараз проживає, свій номер телефону та email, які необхідні для того, щоб за потреби з нею могли зв'язатися та

уточнити певні деталі, або ж щось обговорити. Варто також зазначити, що електронну пошту вводити не обов'язково. Кожен новий користувач проходить процедуру ретельної верифікації. Цей процес включає в себе не лише стандартну перевірку введених даних, але й оцінку потенційного ризику зловживань. Такий підхід забезпечує додаткову безпеку для всіх учасників електронних закупівель. Після цього вона стає повноправним клієнтом і уже може розпочинати шукати для себе постачальника, яких доволі велика кількість.

Кожен потенційний постачальник має у своєму розпорядженні певний список товарів, а також може надавати додаткові послуги. Для прикладу, безкоштовна доставка товару у зазначене місце чи може знижка на першу покупку. Кожен постачальник має право взяти участь у розіграші різноманітних тендерів, котрі надають певні переваги, або ж бонуси. Для цього вони пропонують свою суму і той, чия пропозиція зацікавить найбільше, власне, і отримує свою винагороду.

Кожен постачальник також реєструє себе спочатку у базі даних, тому потенційні клієнти матимуть потрібні ресурси для того, щоб зв'язатись із кожним із них. Після того, як людина обрала того, чиї послуги вважає для себе найкращими у даний момент часу, вони виходять на контакт. У випадку, коли перемовини пройшли успішно і всі задоволені запропонованими умовами, формується контракт між клієнтом та постачальником як підтвердження їх співпраці.

Тепер клієнт може вільно робити замовлення у свого постачальника та домовлятися з ним уже “на місці”. Вся інформація про кожне із замовлень також буде зберігатись у базі даних, як і історія всіх транзакцій, тому користувач може вільно переглядати свої старі замовлення, або ж оплачувати вже нові, це залежить від нього.

Крім того, важливою є інтеграція бази даних з різними платіжними системами. Система повинна підтримувати різноманітні способи оплати, включаючи банківські картки, електронні гаманці та банківські перекази, а також забезпечувати захист від шахрайства та відмивання коштів.

По закінченню контракту кожен клієнт має право написати відгук на свого постачальника, у якому стисло чи розгорнуто пояснити за його сильні сторони або мінуси. Також одночасно потрібно поставити йому оцінку, яка вплине на загальний рейтинг продавця. А рейтинг це одна з найголовніших ознак, по якій оцінюється перспектива працювати разом.

Нарешті, не можна забувати про важливість забезпечення конфіденційності даних користувачів. База даних має відповідати всім сучасним вимогам захисту персональних даних, включаючи регулювання згідно з GDPR та іншими національними та міжнародними нормами.

3. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ

Аналізуючи існуючі програмні продукти в контексті бази даних для торгового майданчика, призначеного для проведення електронних закупівель, можна відзначити кілька ключових аспектів, які важливі для її ефективного функціонування.

Для налагодження ділових відносин між клієнтом та постачальником потрібно в першу чергу укласти між ними контракт. Дані кожного укладеного контракту зберігаються у базі даних та наділені правами люди можуть переглянути їх у будь-який момент.

Щоб оформити замовлення клієнту потрібно спочатку зареєструватись, після чого він стає повноцінним учасником торгового майданчика. Всі замовлення, що логічно, формуються у електронному варіанті.

Перевагою такої системи є її надійність та безпека. Дані користувачів надійно збережені у базі даних, захищені від сторонніх очей. Окрім того, Торговий майданчик має вільний доступ до цих даних.

Недоліком є те, що деякі системи можуть не відповідати всім потребам користувачів, особливо в контексті індивідуальних вимог та інтеграції зі специфічними зовнішніми додатками.

Тепер поговоримо про уже існуючі справжні торгові майданчики.

Платформа Tender.UUB.com.ua, яка є частиною Української Універсальної Біржі, діє з 1997 року та була акредитована в системі Прозорро в 2016 році. Вона відзначається зручним і інтуїтивно зрозумілим інтерфейсом, що робить її привабливою для новачків. Платформа забезпечує повний спектр послуг з закупівель, охоплюючи як допорогові, так і надпорогові процедури. Особлива увага приділяється безпеці ІТ-систем та професійній підтримці користувачів. Основні послуги, що надаються

платформою, включають безкоштовну реєстрацію, обмін інформацією та оголошення тендерів.

Zakupivli.Pro, в свою чергу, фокусується на забезпеченні прозорості та відкритості процесів закупівель. Ця платформа є місцем для проведення як державних, так і комерційних торгів. Вона пропонує навчальний портал, безкоштовні онлайн-курси, вебінари, юридичний супровід та службу підтримки клієнтів. Завдяки зручному особистому кабінету та мобільному додатку, користувачі можуть ефективно управляти закупівлями у будь-якому місці. Платформа також підтримує комерційні торги та аукціони з продажу або оренди майна через систему Prozorro.Продажі.

E-Tender.ua є однією з найбільших платформ системи Prozorro та активно співпрацює як з державними підприємствами, так і з представниками бізнесу. Платформа орієнтована на індивідуальний підхід, пропонуючи компетентних професіоналів, освітні програми та комплексну консультаційну підтримку. E-Tender.ua регулярно оновлюється та вдосконалюється для забезпечення успішного розвитку бізнесу клієнтів, пропонуючи сучасні IT-послуги та підтримку повного циклу закупівель.

Кожна з цих платформ вносить свій вклад у забезпечення прозорих, ефективних та доступних процесів публічних закупівель в Україні, пропонуючи унікальні функції та послуги, які відповідають різним потребам користувачів.

4. ПОСТАНОВКА ЗАВДАННЯ

Задача цієї курсової роботи полягає в аналізі предметного середовища, детальному описі його характеристик, аналізі наявних продуктів, розкритті бізнес-правил та їхньому деталізованому описі. Також потрібно створити ER-модель, розробити базу даних з використанням SQL, а також створити всі необхідні тригери, функції та процедури для ефективного функціонування бази даних. Мета полягає в тому, щоб база даних повністю задовольняла вимоги торгового майданчика і забезпечувала його ефективну роботу. Крім того, в рамках завдання передбачено проведення дослідження щодо оптимізації запитів в створеній базі даних. Завершальна частина роботи включає в себе висновки на основі виконаного аналізу та розробки бази даних.

5. ПОБУДОВА ERD ТА РЕЛЯЦІЙНОЇ МОДЕЛІ

Для початку слід сформулювати бізнес-правила майбутньої системи:

- 1) Не можна записати 2 і більше клієнтів із однаковими електронною поштою та/або номером телефону
- 2) Не можна записати 2 і більше постачальників у базу даних із однаковими електронною поштою та/або номером телефону
- 3) При написанні нового відгуку на певного провайдера його рейтинг повинен автоматично оновитись на основі нових даних
- 4) Допоки постачальник сумлінно виконує свої обов'язки(його середній рейтинг від клієнтів не нижчий 80 зі 100) та не має інших проблем його не можна безпідставно звільнити(видалити з бази даних)
- 5) Після видалення певного відгуку, рейтинг провайдера, на котрого він був написаний також має оновитись

На основі опису предметного середовища можна сформулювати такі необхідні сутності:

- 1) client – сутність, що представляє всіх зареєстрованих на даний момент клієнтів.
- 2) provider – сутність, що представляє всіх доступних провайдерів.
- 3) contract – сутність, що представляє укладені контракти між клієнтом та постачальником.
- 4) service – сутність, що представляє доступні додаткові послуги.
- 5) product – сутність, що представляє товари, доступні у кожного постачальника.
- 6) tender – сутність, що представляє розіграні тендери.
- 7) bid – сутність, що представляє інформацію про ставки по конкретному тендеру.
- 8) review – сутність, що представляє написані відгуки.
- 9) payment – сутність, що містить інформацію про здійснені оплати.

10) order – сутність, що представляє інформацію про всі замовлення.

Окрім того, можна сформулювати такі зв'язки:

1) client - order – зв'язок «один до багатьох», адже кожне замовлення формується на основі одного клієнта, проте один клієнт може робити безліч замовлень.

2) client - review – зв'язок «один до багатьох», адже кожен відгук пише один конкретний клієнт, а один клієнт може написати багато відгуків.

3) client - contract – зв'язок «один до багатьох», тому що один клієнт може укласти багато контрактів, проте контракт може бути тільки для одного клієнта.

4) provider - contract – зв'язок «один до багатьох», тому що один постачальник може укласти багато контрактів, проте контракт може бути тільки для одного постачальника.

5) provider - review – зв'язок «один до багатьох», адже кожен відгук пишеться на конкретного провайдера, а один провайдер може отримати багато відгуків.

6) provider - service – зв'язок «один до багатьох», адже кожна послуга надається певним постачальником, але один постачальник може надавати багато послуг.

7) provider - product – зв'язок «один до багатьох», адже кожен продукт належить певному постачальнику, але один постачальник може мати багато товарів.

8) provider - bid – зв'язок «один до багатьох», тому що ставку робить певний провайдер, але провайдер може робити багато ставок.

9) tender - bid – зв'язок «один до багатьох», адже ставка робиться для тендеру, а на один певний тендер може бути багато ставок.

10) payment – order – зв'язок «один до багатьох», тому що ордер оплачується одним платежем, проте одним платежем можна оформити кілька ордерів відразу, якщо клієнт один і той самий.

Вкажемо користувачів, яких треба реалізувати, і їхні права.

1) data_analyst – аналітик даних. Відповідає за аналіз дани бази даних, тому має змогу читати дані з усіх таблиць. Також у нього є можливість створювати та видаляти тимчасові таблиці для аналізу та можливість виконувати запити SELECT на всіх таблицях.

2) provider_manager – керує роботою провайдерів. Він має права читання та оновлення даних у таблицях provider, contract, service та product, а також може додавати нові дані до даних таблиць. Має право лише читати інформацію про клієнтів.

3) client_manager – керує даними, зв'язаними із клієнтами. Він має права читання та оновлення даних у таблицях client, contract та order, а також може додавати нові дані до даних таблиць. Має право лише читати інформацію про провайдерів.

4) accountant – виконує роль бухгалтера. Має права читання даних у таблицях contract, order та payment. Це дасть йому можливість створювати звіти на основі даних про платежі та контракти.

Скрипти створення ролей наведено в розділі 7. Окрім створення ролей, для роботи із базою даних в цьому розділі створено користувачів із паролями.

Перед реалізацією бази даних засобами мови SQL, слід створити ER-діаграму бази даних. Побудована діаграма у лапковій моделі (Crow's Foot) і зображена на рисунку 5.1.

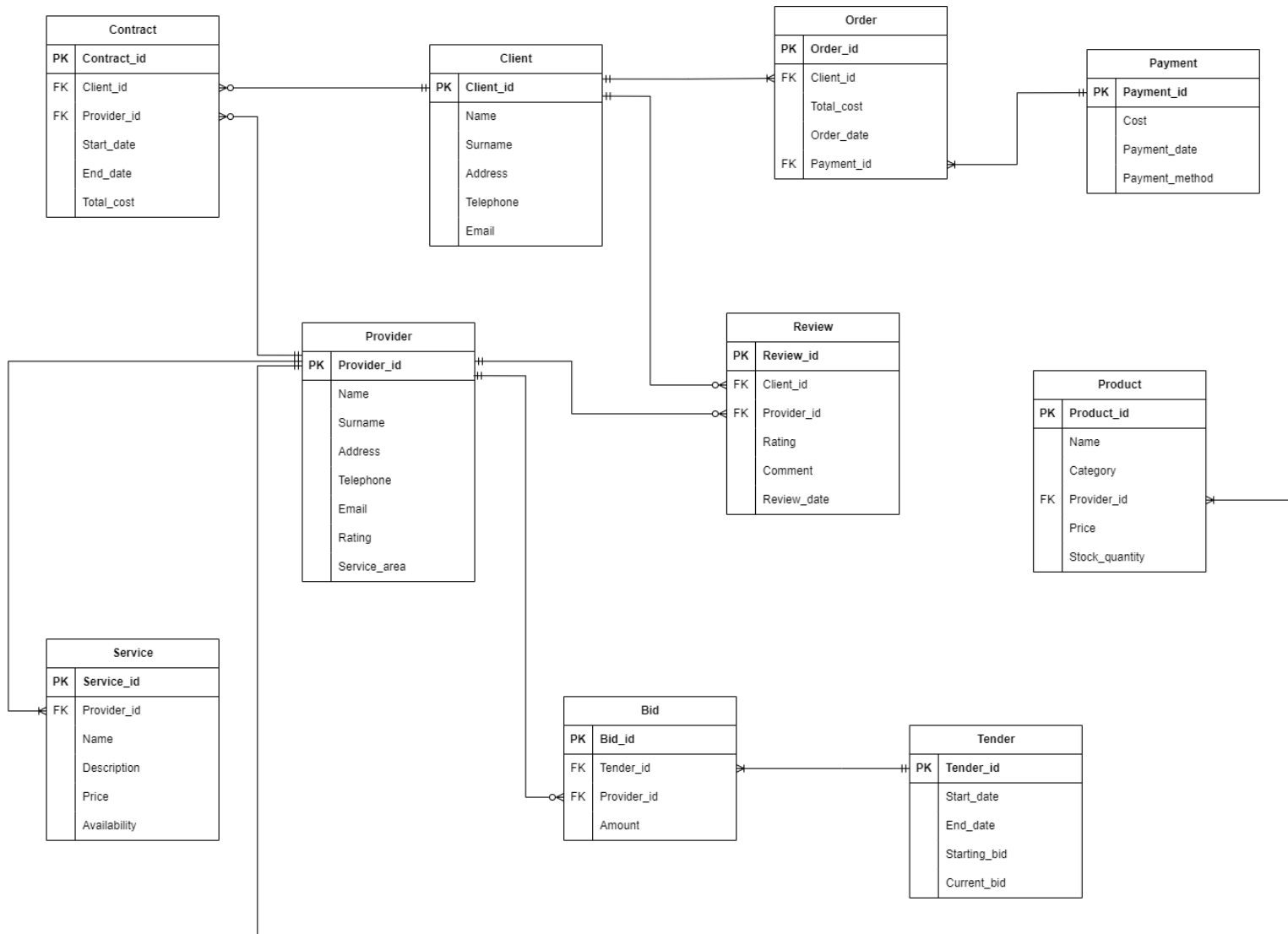


Рисунок 5.1 – ER діаграма до бази даних торгового майданчика

Нижче наведено опис необхідних таблиць.

1) У таблиці “Client” зберігаються дані про клієнтів

Таблиця 5.1 – Клієнт (client)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Client_id	BIGSERIAL		РК	Унікальний номер клієнта в базі даних
Name	VARCHAR	30		Ім'я клієнта
Surname	VARCHAR	30		Прізвище клієнта
Address	TEXT			Адреса проживання клієнта
Telephone	VARCHAR	20		Номер телефону клієнта
Email	VARCHAR	50		Адреса електронної пошти клієнта

2) У таблиці “Provider” зберігаються дані про постачальників

Таблиця 5.2 – Постачальник (provider)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Provider_id	BIGSERIAL		РК	Унікальний номер постачальника в базі даних
Name	VARCHAR	30		Ім'я постачальника
Surname	VARCHAR	30		Прізвище постачальника
Address	TEXT			Адреса проживання постачальника
Telephone	VARCHAR	20		Номер телефону постачальника
Email	VARCHAR	50		Адреса електронної пошти постачальника
Rating	SMALLINT			Рейтинг постачальника
Service_area	TEXT			Регіон, в якому постачальник працює

3) У таблиці “Contract” зберігається інформація про всі угоди між клієнтом та постачальником.

Таблиця 5.3 – Контракт (contract)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Contract_id	BIGSERIAL		PK	Унікальний номер контракту в базі даних
Client_id	BIGINT		FK	Унікальний номер клієнта в базі даних
Provider_id	BIGINT		FK	Унікальний номер постачальника в базі даних
Start_date	DATE			Дата укладання угоди
End_date	DATE			Дата закінчення дії угоди
Total_cost	NUMERIC			Загальна вартість угоди

- 4) У таблиці “Service” зберігаються послуги, які може надати кожен із постачальників

Таблиця 5.4 – Сервіс (service)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Service_id	BIGSERIAL		PK	Унікальний номер послуги в базі даних
Provider_id	BIGINT		FK	Унікальний номер постачальника в базі даних
Name	VARCHAR	50		Назва послуги
Description	TEXT			Опис даної послуги
Price	NUMERIC			Ціна послуги(може бути безкоштовною)
Availability	VARCHAR	20		Доступність послуги на даний момент

- 5) У таблиці “Bid” зберігаються дані про пропозиції, які роблять постачальники в рамках тендерів

Таблиця 5.5 – Ставки (bid)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Bid_id	BIGSERIAL		PK	Унікальний номер пропозиції в базі даних
Provider_id	BIGINT		FK	Унікальний номер постачальника в базі даних
Tender_id	BIGINT		FK	Унікальний номер тендеру
Amount	INT			Кількість пропозицій в рамках тендеру

- 6) У таблиці “Product” зберігаються дані про товар, який може запропонувати кожен із постачальників

Таблиця 5.6 – Товари (product)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Product_id	BIGSERIAL		PK	Унікальний номер товару в базі даних
Provider_id	BIGINT		FK	Унікальний номер постачальника в базі даних
Name	VARCHAR	50		Назва товару
Price	NUMERIC			Ціна товару
Category	VARCHAR	50		Категорія товару
Stock_quantity	INT			Кількість товару на складі

- 7) У таблиці “Tender” зберігаються дані про тендери, в яких можуть брати участь постачальники

Таблиця 5.7 – Тендери (tender)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Tender_id	BIGSERIAL		PK	Унікальний номер тендеру в базі даних
Start_date	DATE			Дата отримання тендеру
End_date	DATE			Дата кінця дії тендеру
Starting_bid	INT			Початкова пропозиція
Current_bid	INT			Поточна пропозиція

- 8) У таблиці “Review” зберігаються дані про відгуки та оцінки, які залишають користувачі щодо постачальників

Таблиця 5.8 – Відгуки (review)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Review_id	BIGSERIAL		PK	Унікальний номер відгуку в базі даних
Client_id	BIGINT		FK	Унікальний номер клієнта в базі даних
Provider_id	BIGINT		FK	Унікальний номер постачальника в базі даних
Rating	SMALLINT			Оцінка певного постачальника
Comment	TEXT			Коментар, в якому можна написати про плюси та мінуси роботи з постачальником
Review_date	DATE			Дата, коли був залишений відгук

- 9) У таблиці “Order” зберігаються дані про замовлення, які роблять покупці

Таблиця 5.9 – Замовлення (order)

Ім'я колонки	Тип	Розмір	Ключ	Опис
Order_id	BIGSERIAL		PK	Унікальний номер замовлення в базі даних
Client_id	BIGINT		FK	Унікальний номер клієнта в базі даних
Payment_id	BIGINT		FK	Унікальний номер платежу в базі даних
Total_cost	NUMERIC			Загальна вартість замовлення
Order_date	DATE			Дата замовлення

- 10) У таблиці “Payment” зберігаються дані про платіжні транзакції клієнтів

Таблиця 5.10 – Платежі (payment)

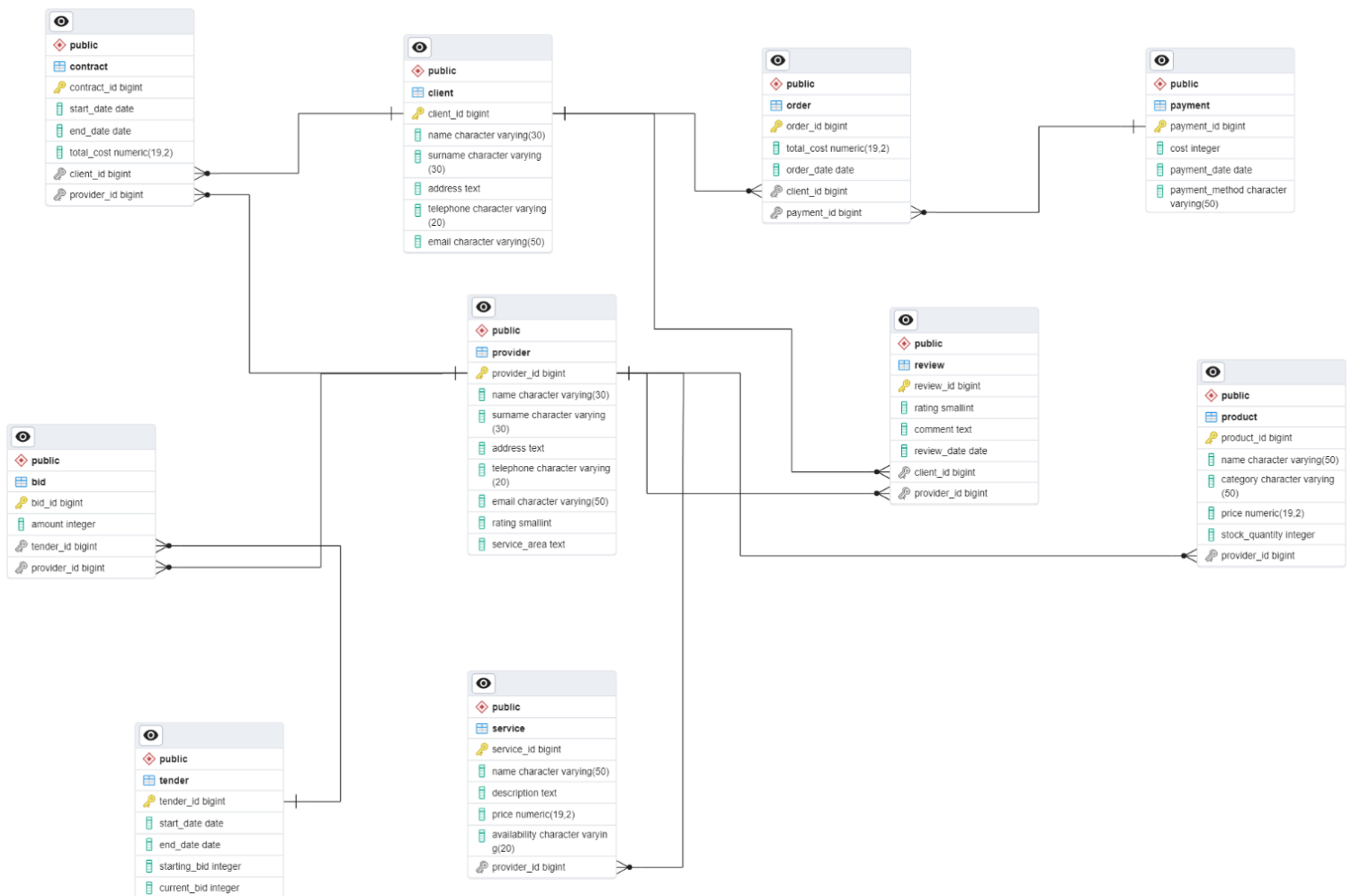
Ім'я колонки	Тип	Розмір	Ключ	Опис
Payment_id	BIGSERIAL		PK	Унікальний номер платежу в базі даних
Cost	INT			Сума транзакцій
Payment_date	DATE			Дата здійснення платежу
Payment_method	VARCHAR	50		Метод платежу(кредитна картка, банківський переказ, електронний гаманець)

Для реалізації бази даних було обрано PostgreSQL, один з діалектів мови програмування SQL. Більш детально про реалізацію бази даних наведено в наступному розділі.

PostgreSQL є потужною та розширюваною системою управління базами даних і має ряд переваг над іншими діалектами СУБД, такими як MySQL чи, наприклад, Microsoft SQL Server.

Postgre підтримує багато розширень. До них відносяться запити, функції та процедури. Діалект має велику і дуже активну спільноту розробників, що дозволяє швидко виявляти і виправляти помилки та розробляти нові можливості при написанні коду. Крім того, це значить, що написані цим діалектом програми будуть мати непоганий попит.

Код для створення бази даних для підтримки торгового майданчику для проведення електронних закупівель наведено в наступному розділі. На рисунку 5.2 показано ERD, створену засобами СУБД PostgreSQL.



6. РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

Результатом проєктування бази даних є її реалізація засобами мови програмування SQL. Нижче наведено скрипти, що забезпечують створення бази даних trading_platform та реалізацію її таблиць.

Окрім того, в створених таблицях використовуються генератори (поля з типом даних BIGSERIAL).

```
CREATE DATABASE trading_platform;
```

```
CREATE TABLE client(  
    client_id BIGSERIAL NOT NULL PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    surname VARCHAR(30) NOT NULL,  
    address TEXT NOT NULL,  
    telephone VARCHAR(20) NOT NULL,  
    email VARCHAR(50)  
);
```

```
CREATE TABLE provider(  
    provider_id BIGSERIAL NOT NULL PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    surname VARCHAR(30) NOT NULL,  
    address TEXT NOT NULL,  
    telephone VARCHAR(20) NOT NULL,  
    email VARCHAR(50),  
    rating SMALLINT NOT NULL,  
    service_area TEXT NOT NULL  
);
```

```
CREATE TABLE contract(  
    contract_id BIGSERIAL NOT NULL PRIMARY KEY,  
    start_date DATE NOT NULL,  
    end_date DATE NOT NULL,  
    total_cost NUMERIC(19, 2) NOT NULL,  
    client_id BIGINT NOT NULL REFERENCES client(client_id),  
    provider_id BIGINT NOT NULL REFERENCES provider(provider_id)  
);
```

```
CREATE TABLE service(  
    service_id BIGSERIAL NOT NULL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    description TEXT NOT NULL,  
    price NUMERIC(19, 2) NOT NULL,  
    availability VARCHAR(20) NOT NULL,  
    provider_id BIGINT NOT NULL REFERENCES provider(provider_id)  
);
```

```
CREATE TABLE product(  
    product_id BIGSERIAL NOT NULL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    price NUMERIC(19, 2) NOT NULL,  
    stock_quantity INT NOT NULL,  
    provider_id BIGINT NOT NULL REFERENCES provider(provider_id)  
);
```

```
CREATE TABLE tender(  
    tender_id BIGSERIAL NOT NULL PRIMARY KEY,  
    start_date DATE NOT NULL,  
    end_date DATE NOT NULL,  
    starting_bid INT NOT NULL,  
    current_bid INT NOT NULL  
);
```

```
CREATE TABLE bid(  
    bid_id BIGSERIAL NOT NULL PRIMARY KEY,  
    amount INT NOT NULL,  
    tender_id BIGINT NOT NULL REFERENCES tender(tender_id),  
    provider_id BIGINT NOT NULL REFERENCES provider(provider_id)  
);
```

```
CREATE TABLE review(  
    review_id BIGSERIAL NOT NULL PRIMARY KEY,  
    rating SMALLINT NOT NULL,  
    comment TEXT,  
    review_date DATE NOT NULL,  
    client_id BIGINT NOT NULL REFERENCES client(client_id),  
    provider_id BIGINT NOT NULL REFERENCES provider(provider_id)  
);
```

```
CREATE TABLE payment(  
    payment_id BIGSERIAL NOT NULL PRIMARY KEY,  
    cost INT NOT NULL,  
    payment_date DATE NOT NULL,  
    payment_method VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE "order"(  
    order_id BIGSERIAL NOT NULL PRIMARY KEY,  
    total_cost NUMERIC(19, 2) NOT NULL,  
    order_date DATE NOT NULL,  
    client_id BIGINT NOT NULL REFERENCES client(client_id),  
    payment_id BIGINT NOT NULL REFERENCES payment(payment_id)  
);
```

Код для заповнення бази даних даними наведено у Додатку А.

Для перевірки правильності створення бази даних `service_center_mia`, можна скористатися командою `\l` у SQL Shell. Результат на рисунку 6.1.

Для перевірки правильності створення таблиць БД `service_center_mia`, слід скористатися командою `\dt` у SQL Shell. Зауваження: спочатку потрібно буде підключитись до цієї бази даних за допомогою команди `\c trading_platform`; Результат на рисунку 6.2.

```
postgres=# \l
```

Name	Owner	Encoding	Locale Provider	Collate	Ctype	ICU Locale	ICU Rules	Access privileges
postgres	postgres	UTF8	libc	Ukrainian_Ukraine.1251	Ukrainian_Ukraine.1251			
template0	postgres	UTF8	libc	Ukrainian_Ukraine.1251	Ukrainian_Ukraine.1251			=c/postgres postgres=Ctc/postgres +
template1	postgres	UTF8	libc	Ukrainian_Ukraine.1251	Ukrainian_Ukraine.1251			=c/postgres postgres=Ctc/postgres +
trading_platform	postgres	UTF8	libc	Ukrainian_Ukraine.1251	Ukrainian_Ukraine.1251			

(4 rows)

```
postgres=#
```

Рисунок 6.1 – Створена база даних у SQL Shell

```
trading_platform=# \dt
```

Schema	Name	Type	Owner
public	bid	table	postgres
public	client	table	postgres
public	contract	table	postgres
public	order	table	postgres
public	payment	table	postgres
public	product	table	postgres
public	provider	table	postgres
public	review	table	postgres
public	service	table	postgres
public	tender	table	postgres

(10 rows)

Рисунок 6.2 – Створені таблиці БД `trading_platform`

7. РЕАЛІЗАЦІЯ РОЛЕЙ ТА КОРИСТУВАЧІВ

Для забезпечення безпеки бази даних, встановлення обмежень на доступ до даних та ідентифікації користувачів створюються ролі та користувачі. Опис ролей та їх прав наведено у 5 розділі пояснювальної записки.

Нижче наведено код реалізації ролей та користувачів для бази даних trading_platform та надання їм необхідних привілеїв.

7.1. Створення ролей

```
CREATE ROLE data_analyst_role;  
CREATE ROLE provider_manager_role;  
CREATE ROLE client_manager_role;  
CREATE ROLE accountant_role;
```

7.2. Надання привілеїв для ролі аналітика даних

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO data_analyst_role;
```

7.3. Надання привілеїв для ролі менеджера постачальників

```
GRANT SELECT, INSERT, UPDATE ON TABLE provider, contract, service,  
product TO provider_manager_role;  
  
GRANT SELECT ON TABLE client TO provider_manager_role;
```

7.4. Надання привілеїв для ролі менеджера клієнтів

```
GRANT SELECT, INSERT, UPDATE ON TABLE client, contract, "order" TO  
client_manager_role;  
  
GRANT SELECT ON TABLE provider TO client_manager_role;
```

7.5. Надання привілеїв для ролі бухгалтера

```
GRANT SELECT ON TABLE contract, "order", payment TO accountant_role;
```

7.6. Створення користувачів

```
CREATE USER data_analyst WITH PASSWORD 'analys';
```

```
CREATE USER provider_manager WITH PASSWORD 'manager777';
```

```
CREATE USER client_manager WITH PASSWORD 'manager123';
```

```
CREATE USER accountant WITH PASSWORD '00877';
```

```
GRANT data_analyst_role TO data_analyst;
```

```
GRANT provider_manager_role TO provider_manager;
```

```
GRANT client_manager_role TO client_manager;
```

```
GRANT accountant_role TO accountant;
```

Успішність створення ролей та користувачів перевіримо за допомогою запитів. Перевірка ролей показана на рисунку 7.1, перевірка користувачів – на рисунку 7.2.

```
trading_platform=# SELECT rolname FROM pg_roles;
rolname
-----
pg_database_owner
pg_read_all_data
pg_write_all_data
pg_monitor
pg_read_all_settings
pg_read_all_stats
pg_stat_scan_tables
pg_read_server_files
pg_write_server_files
pg_execute_server_program
pg_signal_backend
pg_checkpoint
pg_use_reserved_connections
pg_create_subscription
postgres
data_analyst_role
provider_manager_role
client_manager_role
accountant_role
data_analyst
provider_manager
client_manager
accountant
(23 rows)
```

Рисунок 7.1 – Створені ролі бази даних service_center_mia

```
trading_platform=# SELECT username FROM pg_user;
username
-----
postgres
data_analyst
provider_manager
client_manager
accountant
(5 rows)
```

Рисунок 7.2 – Створені користувачі бази даних service_center_mia

8. РЕАЛІЗАЦІЯ ТРИГЕРІВ

У базах даних тригери потрібні для автоматизації бізнес-логіки, забезпечення цілісності даних, а також виконання деяких дій перед або після операцій з даними.

Нижче наведено реалізовані тригери, їхній словесний опис та суть, а також приклади роботи.

8.1. Тригер, що реагує на вставку нового клієнта.

Цей тригер реалізує таке обмеження, що у базі даних не може одночасно знаходитись 2 або більше клієнти, у яких однаковий номер телефону або ж електронна пошта, тому що це зруйнує всю логіку бази даних та зробить користування нею незручним. Таку логіку виконують тригер `before_insert_client` і тригерна функція `check_client_uniqueness`. Нижче наведено їхній код.

```

CREATE OR REPLACE FUNCTION check_client_uniqueness()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT 1 FROM client WHERE email = NEW.email) THEN
        RAISE EXCEPTION 'Email % already exists for a client', NEW.email;
    END IF;
    IF EXISTS (SELECT 1 FROM client WHERE telephone = NEW.telephone)
    THEN
        RAISE EXCEPTION 'Telephone number % already exists for a client',
NEW.telephone;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER before_insert_client
BEFORE INSERT ON client
FOR EACH ROW
EXECUTE FUNCTION check_client_uniqueness();

```

На рисунку 8.1. показано роботу тригера при вставці нового клієнта, у якого номер телефону співпадає із уже існуючим клієнтом. На рисунку 8.2. показано вставку клієнта, котрий не порушив ніяких умов.

```

trading_platform=# INSERT INTO client (name, surname, address, telephone, email)
trading_platform=# VALUES ('John', 'Doe', '123 Main Street, Cityville', '107-859-2115', 'lwle0@biloe.ne.jp');
ERROR:  Telephone number 107-859-2115 already exists for a client
CONTEXT:  PL/pgSQL function check_client_uniqueness() line 8 at RAISE
trading_platform=#

```

Рисунок 8.1 – Вставка клієнта, що не підходить

```

trading_platform=# INSERT INTO client (name, surname, address, telephone, email)
trading_platform=# VALUES ('John', 'Doe', '123 Main Street, Cityville', '555-1234', 'lwle0@biloe.ne.jp');
INSERT 0 1
trading_platform=#

```

Рисунок 8.2 – Вставка клієнта, що підходить

8.2. Тригер, що реагує на вставку нового постачальника.

Ситуація ідентична, як і з клієнтом. Не може одночасно існувати два і більше постачальника із однаковим номером телефону та/або електронною поштою. Таку логіку виконують тригер `before_insert_provider` і тригерна функція `check_provider_uniqueness`. Нижче наведено їхній код.

```
CREATE OR REPLACE FUNCTION check_provider_uniqueness()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT 1 FROM provider WHERE email = NEW.email) THEN
        RAISE EXCEPTION 'Email % already exists for a provider', NEW.email;
    END IF;

    IF EXISTS (SELECT 1 FROM provider WHERE telephone = NEW.telephone)
    THEN
        RAISE EXCEPTION 'Telephone number % already exists for a provider',
        NEW.telephone;
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER before_insert_provider
BEFORE INSERT ON provider
FOR EACH ROW
EXECUTE FUNCTION check_provider_uniqueness();
```

На рисунку 8.3. показано роботу тригера при вставці нового клієнта, у якого номер телефону співпадає із уже існуючим клієнтом. На рисунку 8.4. показано вставку клієнта, котрий не порушив ніяких умов.

```
trading_platform=# INSERT INTO provider (name, surname, address, telephone, email, rating, service_area)
trading_platform=# VALUES ('ProviderFirstName', 'ProviderLastName', '456 Oak Street, Townsville', '555-5678', 'slindsay0@lycos.com', 4, 'Service Area Details');
ERROR:  Email slindsay0@lycos.com already exists for a provider
CONTEXT:  PL/pgSQL function check_provider_uniqueness() line 4 at RAISE
trading_platform=#
```

Рисунок 8.3 – Вставка постачальника, що не підходить

```
trading_platform=# INSERT INTO provider (name, surname, address, telephone, email, rating, service_area)
trading_platform=# VALUES ('ProviderFirstName', 'ProviderLastName', '456 Oak Street, Townsville', '555-5678', 'slinda0@lycos.com', 4, 'Service Area Details');
INSERT 0 1
trading_platform=#
```

Рисунок 8.4 – Вставка постачальника, що підходить

8.3. Тригер, що реагує на написання нового відгуку.

Цей тригер реагує на написання нового відгуку клієнтом та автоматично оновлює рейтинг провайдера, використовуючи нові дані. Цю логіку виконують тригер `after_insert_review` і тригерна функція `update_provider_rating`. Нижче наведено їхній код.

```
CREATE OR REPLACE FUNCTION update_provider_rating()
RETURNS TRIGGER AS $$
DECLARE
    average_rating NUMERIC;
BEGIN

    SELECT AVG(rating) INTO average_rating
    FROM review
    WHERE provider_id = NEW.provider_id;

    UPDATE provider
    SET rating = ROUND(average_rating)
    WHERE provider_id = NEW.provider_id;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER after_insert_review
AFTER INSERT ON review
FOR EACH ROW
EXECUTE FUNCTION update_provider_rating();
```

Візьмемо за приклад першого провайдера. На рисунку 8.5 показано його рейтинг на даний момент. На рисунку 8.6 показано роботу тригера. На рисунку 8.7 показано його рейтинг, обрахований з урахуванням нового відгуку

```
trading_platform=# SELECT * FROM provider;
```

provider_id	name	surname	address	telephone	email	rating	service_area
1	Sloan	Lindsay	60 Claremont Lane	353-112-2952	slindsay0@lycos.com	95	Asia
2	Pascale	Keelinge	48809 Lien Lane	547-628-0200	pkeelingel@elegantthemes.com	34	South America
3	Noak	Sibbs	7914 Maryland Alley	750-476-7998	nsibbs2@bloglines.com	69	North America
4	Hartley	Olyff	4782 Arizona Drive	264-637-0651	hollyff3@wix.com	58	Oceania
5	Lib	Kynnd	666 Manufacturers Pass	823-452-2342	lkynnd4@yahoo.co.jp	63	Africa
6	Philipa	Loukes	3 Nova Circle	168-821-5719		60	Europe
7	Junia	Stitcher	5771 Twin Pines Place	170-706-6112		29	Asia
8	Thedric	Doumic	8 Cordelia Junction	765-805-8693	tdoumic7@acquirethisname.com	51	South America
9	Dianemarie	Baudts	50015 Fisk Alley	379-921-2503		32	North America
10	Devondra	Mussetтини	3894 Carpenter Street	628-943-5090	dmussetтини9@angelfire.com	74	Oceania
11	Maura	Labdon	882 Huxley Drive	263-271-5666		59	Africa
12	Kyle	MacIlwrick	2 Lakewood Gardens Center	116-481-6936	kmacilwrickb@constantcontact.com	71	Europe
13	Kristoforo	Starsmeare	59 Macpherson Pass	432-443-1552	kstarsmearec@blogtalkradio.com	43	Asia

Рисунок 8.5 – Демонстрація рейтингу перед вставкою запису

```
trading_platform=# INSERT INTO review (rating, comment, review_date, client_id, provider_id)
trading_platform=# VALUES (10, 'Excellent service!', '2023-12-29', 1, 1);
INSERT 0 1
```

Рисунок 8.6 – Демонстрація роботи тригера

46	Justinian	Filippyev	5315 Muir Way	774-433-0247		67	Oceania
47	Durand	Bohden	20 Banding Parkway	328-537-1538		86	Africa
48	Vivyanne	Freak	7170 Amoth Terrace	894-843-9338	vfreak1b@washington.edu	66	Europe
49	Sandor	Wilber	05841 Rutledge Way	661-741-4389	swilber1c@mitbeian.gov.cn	83	Asia
50	Dayna	Maureen	3336 Linden Parkway	455-563-3539	dmaureen1d@cam.ac.uk	33	South America
52	ProviderFirstName	ProviderLastName	456 Oak Street, Townsville	555-5678	slinda0@lycos.com	4	Service Area Details
1	Sloan	Lindsay	60 Claremont Lane	353-112-2952	slindsay0@lycos.com	87	Asia

(51 rows)

Рисунок 8.7 – Демонстрація рейтингу після вставки запису

8.4. Тригер, що реагує на зміну рейтингу провайдера.

Даний тригер нічого не обраховує, а виконує більш показову роль, він був створений для зручності. Кожного разу, коли після роботи тригера на вставку відгуку чи їх видалення(про цей поговоримо трохи пізніше) Він виведе повідомлення з оновленим рейтингом. Цю логіку виконують тригер `before_update_provider` і тригерна функція `check_rating_change`. Нижче наведено їхній код.

```
CREATE OR REPLACE FUNCTION check_rating_change()
RETURNS TRIGGER AS $$
BEGIN
    IF OLD.rating <> NEW.rating THEN
        RAISE NOTICE 'Rating for provider % has changed from % to %',
NEW.name, OLD.rating, NEW.rating;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER before_update_provider
BEFORE UPDATE OF rating ON provider
FOR EACH ROW EXECUTE FUNCTION check_rating_change();
```

На рисунку 8.8 продемонстровано роботу тригера при вставці нового відгуку, або ж видалення вже існуючого. Результат буде ідентичним.

```
trading_platform=# INSERT INTO review (rating, comment, review_date, client_id, provider_id)
trading_platform=# VALUES (100, 'Excellent service!', '2023-12-29', 1, 1);
NOTICE: Rating for provider Sloan has changed from 87 to 88
INSERT 0 1
trading_platform=#
```

Рисунок 8.8 – Демонстрація роботи тригера

8.5. Тригер, що реагує на безпричинне видалення сумлінного постачальника, з котрим клієнтам подобається співпрацювати.

Цей тригер реалізує обмеження видалення хорошого постачальника. Коли хтось без підстави хоче видалити провайдера, у якого середній рейтинг не менший ніж 80 (зі ста) тригер якраз і не дозволяє цього зробити. Цю логіку виконують тригер `prevent_delete_high_rated_provider` і тригерна функція `prevent_delete_high_rated_provider`. Нижче наведено їхній код.

```
CREATE OR REPLACE FUNCTION prevent_delete_high_rated_provider()
RETURNS TRIGGER AS $$
BEGIN
    IF OLD.rating >= 80 THEN
        RAISE EXCEPTION 'Cannot delete provider with high rating (%).',
        OLD.rating;
    END IF;

    DELETE FROM contract WHERE provider_id = OLD.provider_id;

    DELETE FROM service WHERE provider_id = OLD.provider_id;

    DELETE FROM product WHERE provider_id = OLD.provider_id;

    DELETE FROM bid WHERE provider_id = OLD.provider_id;

    DELETE FROM review WHERE provider_id = OLD.provider_id;

    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER prevent_delete_high_rated_provider
BEFORE DELETE ON provider
FOR EACH ROW
EXECUTE FUNCTION prevent_delete_high_rated_provider();
```

На рисунку 8.9 продемонстровано роботу тригера при спробі видалення хорошого постачальника. На рисунку 8.10 продемонстровано роботу тригера при спробі видалення не дуже хорошого постачальника

```
trading_platform=# DELETE FROM provider WHERE provider_id = 1;
ERROR:  Cannot delete provider with high rating (88).
CONTEXT:  PL/pgSQL function prevent_delete_high_rated_provider() line 4 at RAISE
trading_platform=#
```

Рисунок 8.9 – Демонстрація роботи тригера при спробі видалення сумлінного постачальника

```
trading_platform=# DELETE FROM provider WHERE provider_id = 52;
DELETE 1
trading_platform=#
```

Рисунок 8.10 – Демонстрація роботи тригера при спробі видалення не дуже хорошого постачальника

8.6. Тригер, що реагує на оновлення рейтингу провайдера після видалення відгуку.

Ми вже згадували про нього вище. Цей тригер виконує таку ж логіку, як і тригер із додаванням нового відгуку, тільки на цей раз він спрацьовує при його видаленні. Цю логіку виконують тригер `after_delete_review` і тригерна функція `update_provider_rating_after_delete`. Нижче наведено їхній код.

```
CREATE OR REPLACE FUNCTION update_provider_rating_after_delete()  
RETURNS TRIGGER AS $$
```

```
DECLARE
```

```
    average_rating NUMERIC;
```

```
BEGIN
```

```
    SELECT COALESCE(AVG(rating), 0) INTO average_rating
```

```
    FROM review
```

```
    WHERE provider_id = OLD.provider_id;
```

```
    UPDATE provider
```

```
    SET rating = ROUND(average_rating)
```

```
    WHERE provider_id = OLD.provider_id;
```

```
    RETURN OLD;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER after_delete_review
```

```
AFTER DELETE ON review
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION update_provider_rating_after_delete();
```

На рисунку 8.10 продемонстровано видалення певного існуючого відгуку.

```
trading_platform=# DELETE FROM review WHERE review_id = 129;  
NOTICE: Rating for provider Dayna has changed from 33 to 35  
DELETE 1  
trading_platform=#
```

Рисунок 8.10 – Реакція тригера на видалення відгуку

9. РЕАЛІЗАЦІЯ ПРЕДСТАВЛЕНЬ

Важливим елементом бази даних є механізм створення представлень. Представлення - це об'єкти бази даних, які представляють віртуальні таблиці на основі результатів запиту. Представлення можна використовувати для спрощення складних запитів, об'єднання таблиць та інших операцій з даними, а також для повторного використання коду.

Нижче наведено і описано представлення, що були реалізовані в базі даних trading_platform, а також наведено результат їх використання.

9.1. Представлення, що містить у собі детальну інформацію про всі контракти, присутні у базі даних.

В цьому представлення виводяться основні стовпці, необхідні для інформативного виведення укладних контрактів. Нижче наведений код і приклад використання.

```
CREATE VIEW contract_details AS
SELECT
    c.contract_id,
    cl.name AS client_name,
    pr.name AS provider_name,
    c.start_date,
    c.end_date,
    c.total_cost
FROM contract c
JOIN client cl ON c.client_id = cl.client_id
JOIN provider pr ON c.provider_id = pr.provider_id;
```

На рисунку 9.1. показано приклад виведення такого представлення.

Для цього можна використати запит `SELECT * FROM contract_details;`

```
trading_platform=# SELECT * FROM contract_details;
```

contract_id	client_name	provider_name	start_date	end_date	total_cost
1	Charlot	Marji	2022-03-16	2024-03-25	136665.00
2	Conn	Jedd	2022-06-28	2024-08-16	130229.00
3	Blane	Golda	2022-06-24	2024-07-14	915633.00
4	Merola	Alfie	2022-04-21	2024-05-13	505675.00
5	Ora	Dicky	2022-11-11	2024-07-10	554144.00
6	Amos	Gunar	2022-07-26	2023-10-31	562966.00
7	Tore	Kristian	2022-06-23	2023-12-21	554149.00
8	Graeme	Ivette	2022-01-17	2023-11-17	259430.00
9	Ree	Cathrine	2022-04-21	2024-04-27	529774.00
10	Blair	Justinian	2022-06-07	2024-02-22	249851.00
11	Kermie	Estelle	2022-06-09	2024-01-26	956384.00
12	Aurelie	Justinian	2022-03-09	2024-02-21	793054.00
13	Kirk	Kyle	2022-08-28	2024-05-19	219624.00
14	Aloysia	Marji	2022-07-28	2023-11-04	289386.00
15	Kurtis	Gunar	2022-06-14	2024-06-17	907728.00
16	Shellie	Dianemarie	2022-01-18	2024-01-10	192856.00
17	Hettie	Hersh	2022-07-27	2023-11-09	482966.00
18	Keith	Kellyann	2022-06-28	2023-12-22	468326.00
19	Ginni	Norine	2022-12-07	2024-06-29	618293.00
20	Amber	Vaughn	2022-03-13	2024-12-19	365257.00
21	Vally	Hersh	2022-08-14	2024-11-08	540118.00
22	Leeann	Hartley	2022-12-01	2024-12-01	239195.00
23	Cly	Celene	2022-04-25	2024-08-27	322130.00
24	Bartholomeo	Junia	2022-12-14	2024-09-20	32180.00
25	Giles	Kristoforo	2022-04-11	2024-08-29	803931.00
26	Lamont	Hartley	2022-12-06	2023-10-21	181238.00
27	Joyce	Kristoforo	2022-09-09	2024-01-26	300694.00
28	Karita	Ted	2022-09-09	2024-10-14	55051.00
29	Virginia	Dicky	2022-04-29	2024-04-25	459151.00
30	Joey	Dicky	2022-04-08	2024-09-10	78004.00
31	Shawn	Sandor	2022-07-05	2024-10-03	606878.00
32	Sharl	Imogene	2022-04-02	2024-06-03	832158.00
33	Dennis	Vaughn	2022-01-20	2023-10-23	705222.00
34	Morey	Noak	2022-01-17	2024-07-01	458042.00
35	Scarface	Jedd	2022-02-14	2024-10-12	100110.00
36	Howard	Hartley	2022-08-10	2024-01-23	222513.00
37	Dalton	Sloan	2022-02-20	2024-09-22	645227.00
38	Paxon	Noak	2022-12-19	2024-11-15	290738.00
39	Kandy	Marji	2022-09-03	2024-08-22	616037.00
40	Kelli	Ivette	2022-08-15	2024-01-28	694877.00
41	Currey	Kristoforo	2022-08-06	2024-12-29	231520.00
42	Vivien	Sloan	2022-07-29	2024-07-16	853328.00
43	Dalila	Germana	2022-09-15	2024-12-03	895614.00
44	Augustine	Noak	2022-02-25	2024-11-19	114911.00
45	Maiga	Golda	2022-01-06	2023-12-12	180457.00

Рисунок 9.1 – Основна інформація про контракти

9.2. Представлення для відслідковування тендерів з їх поточним статусом.

В цьому представленні виводяться дані про кожен тендер, одним зі стовпців є якраз інформація, чи не закінчився термін дії тендеру станом на поточний день. Нижче наведено код цього представлення.

```
CREATE VIEW tender_status AS
SELECT
    t.tender_id,
    t.start_date,
    t.end_date,
    t.starting_bid,
    t.current_bid,
    CASE
        WHEN CURRENT_DATE > t.end_date THEN 'Closed'
        ELSE 'Open'
    END AS status
FROM tender t;
```

На рисунку 9.2. показано приклад виведення створеного представлення.

Для цього можна використати запит `SELECT * FROM tender_status;`

```
trading_platform=# SELECT * FROM tender_status;
```

tender_id	start_date	end_date	starting_bid	current_bid	status
1	2023-12-29	2023-12-30	4734	41196	Closed
2	2023-11-01	2023-11-02	3740	8010	Closed
3	2023-11-07	2024-01-08	4232	43862	Open
4	2023-11-24	2023-11-25	4141	38094	Closed
5	2023-12-29	2023-12-30	4301	33598	Closed
6	2023-12-14	2024-01-06	3310	25867	Open
7	2023-12-02	2023-12-03	3238	32105	Closed
8	2023-11-20	2024-01-12	4190	47433	Open
9	2023-12-04	2024-01-27	3010	29799	Open
10	2023-12-15	2023-12-16	4425	14481	Closed
11	2023-11-29	2023-11-30	4177	8910	Closed
12	2023-11-01	2024-01-08	4596	47211	Open
13	2023-12-08	2023-12-09	3543	33176	Closed
14	2023-11-18	2023-11-19	4179	15553	Closed
15	2023-11-11	2024-01-09	3273	32705	Open
16	2023-11-24	2024-01-17	4887	26622	Open
17	2023-12-18	2023-12-19	4066	38266	Closed
18	2023-12-24	2023-12-25	3267	26957	Closed
19	2023-11-09	2023-11-10	4263	40742	Closed
20	2023-12-11	2023-12-12	3106	24508	Closed
21	2023-12-14	2024-01-18	4920	30898	Open
22	2023-12-07	2023-12-08	3732	45779	Closed
23	2023-12-14	2023-12-15	3593	22118	Closed
24	2023-12-02	2024-01-28	3295	49942	Open
25	2023-12-14	2024-01-11	3769	39176	Open
26	2023-11-11	2023-11-12	4475	17345	Closed
27	2023-12-05	2023-12-06	4286	11014	Closed
28	2023-12-10	2024-01-15	4662	43121	Open
29	2023-12-03	2023-12-04	4454	35827	Closed
30	2023-12-13	2024-01-06	3457	44942	Open

(30 rows)

Рисунок 9.2 – Поточний статус тендерів

9.3. Представлення, що відображає детальну інформацію про кожен написаний відгук.

В цьому представленні виводиться найбільш корисна інформація щодо кожного відгуку, а саме: на якого провайдера від написаний, поставлена йому оцінка, за бажанням написаний коментар, а також дата, коли цей відгук був зроблений. Нижче наведено код цього представлення.

```
CREATE VIEW provider_reviews AS  
  
SELECT  
  
    pr.name AS provider_name,  
  
    r.rating,  
  
    r.comment,  
  
    r.review_date  
  
FROM provider pr  
  
JOIN review r ON pr.provider_id = r.provider_id;
```

На рисунку 9.3. показано приклад виведення всіх стовпчиків такого представлення. Для цього можна використати запит `SELECT * FROM provider_reviews;`

```
trading_platform=# SELECT * FROM provider_reviews;
```

provider_name	rating	comment	review_date
Marve	5	Unreliable service! Products received were of poor quality, and delivery was extremely delayed.	2023-11-29
Guy	85	I am impressed with the professionalism and attention to detail. A reliable supplier indeed!	2022-09-20
Madelyn	49		2023-09-03
Sloan	100	Exceptional service! The quality of the products and the speed of delivery exceeded my expectations.	2022-12-11
Madelyn	70	Average service. The products received were neither exceptional nor disappointing.	2022-10-05
Estelle	71		2022-11-14
Guy	89	Outstanding customer support! They went above and beyond to address my concerns promptly.	2023-10-09
Dianemarie	28		2022-05-02
Cathrine	25		2022-07-28
Cathrine	4	Very disappointed with the lack of professionalism. Numerous errors and a total disregard for customer concerns.	2023-11-21
Estelle	83	Reliable and consistent quality. It's a pleasure doing business with such a trustworthy supplier.	2022-11-02
Estelle	30		2022-10-28
Sloan	99		2022-01-06
Norine	52		2022-07-25
Norine	85		2023-10-31
Madelyn	63	Neutral experience overall. The supplier met basic expectations without any standout qualities.	2022-02-06
Guy	97	Prompt delivery and top-notch products. Highly recommended supplier in the industry!	2023-08-22
Justinian	75		2023-02-17
Estelle	84		2022-05-03
Marve	36		2022-02-16
Marve	6	Horrible customer support! Issues were ignored, and there was no effort to resolve the problems.	2022-11-07
Norine	37		2022-10-24
Madelyn	68	Adequate customer support. They addressed concerns, but the process was not particularly smooth.	2023-04-18
Madelyn	12		2023-07-31
Dianemarie	27		2022-01-14
Guy	89		2023-02-24
Guy	80		2022-07-21
Maura	64		2023-01-17
Maura	79		2022-11-26
Estelle	61	Average quality products delivered within the expected timeframe.	2023-09-03
Maura	18		2023-01-29
Ted	90		2023-12-03

Рисунок 9.3 – Детальна інформація про відгуки

9.4. Представлення, що відображає найкращих провайдерів за сумою прибутку від укладених контрактів.

За допомогою даного представлення можна подивитись топ 10 постачальників за сумою прибутку від усіх контрактів разом. Нижче наведено код цього представлення.

```
CREATE VIEW top_10_providers_by_contract_value AS
SELECT
    pr.provider_id,
    pr.name AS provider_name,
    SUM(co.total_cost) AS total_contract_value
FROM provider pr
JOIN contract co ON pr.provider_id = co.provider_id
GROUP BY pr.provider_id, pr.name
ORDER BY total_contract_value DESC
LIMIT 10;
```

На рисунку 9.4. показано приклад виведення такого представлення. Для його виклику потрібно використати запит `SELECT * FROM top_10_providers_by_contract_value;`

```
trading_platform=# SELECT * FROM top_10_providers_by_contract_value;
 provider_id | provider_name | total_contract_value
-----+-----+-----
          13 | Kristoforo    | 15743304.00
          22 | Cinda         | 14522385.00
           4 | Hartley       | 14407748.00
          31 | Cathrine      | 14329605.00
          18 | Gunar         | 14259773.00
          37 | Godfry        | 13643627.00
          11 | Maura         | 13522735.00
          20 | Golda         | 12673799.00
          17 | Jedd          | 12576556.00
          48 | Vivyanne      | 12441575.00
(10 rows)

trading_platform=#
```

Рисунок 9.4 – Постачальники із найбільшим прибутком

10. РЕАЛІЗАЦІЯ ЗАПИТІВ

Запити можна використовувати в мові програмування SQL для взаємодії з базами даних. Вони використовуються для виконання різних операцій над даними, таких як пошук, оновлення, вставка і видалення інформації в базі даних, і є дуже важливими і корисними. Мова обробки даних DML включає такі запити, як SELECT, INSERT, UPDATE і DELETE.

Нижче наведено приклади корисних запитів типу SELECT до бази даних trading_platform, а також їхній словесний опис та приклад використання.

10.1. Запит для отримання списку клієнтів та інформацію про їх контракти.

Використовується запит, що отримує дані із таблиці клієнтів, а потім підзапит, який витягує дані про контракти, що потім будуть об'єднані з основною таблицею клієнтів. На рисунку 10.1. показано виведення після запиту до бази даних.

SELECT

```
c.name AS client_name,  
c.surname AS client_surname,  
c.address AS client_address,  
c.telephone AS client_telephone,  
c.email AS client_email,  
contract_data.contract_id,  
contract_data.start_date,  
contract_data.end_date,  
contract_data.total_cost
```

FROM client c

JOIN (

SELECT

co.contract_id,

co.start_date,

co.end_date,

co.total_cost,

co.client_id

FROM contract co

) AS contract_data ON c.client_id = contract_data.client_id;

```
trading_platform=# SELECT
trading_platform=# c.name AS client_name,
trading_platform=# c.surname AS client_surname,
trading_platform=# c.address AS client_address,
trading_platform=# c.telephone AS client_telephone,
trading_platform=# c.email AS client_email,
trading_platform=# contract_data.contract_id,
trading_platform=# contract_data.start_date,
trading_platform=# contract_data.end_date,
trading_platform=# contract_data.total_cost
trading_platform=# FROM client c
trading_platform=# JOIN (
trading_platform=# SELECT
trading_platform=# co.contract_id,
trading_platform=# co.start_date,
trading_platform=# co.end_date,
trading_platform=# co.total_cost,
trading_platform=# co.client_id
trading_platform=# FROM contract co
trading_platform=# ) AS contract_data ON c.client_id = contract_data.client_id;
```

client_name	client_surname	client_address	client_telephone	client_email	contract_id	start_date	end_date	total_cost
Charlot	Lowle	96378 Evergreen Trail	355-230-4922	clowle0@biglobe.ne.jp	1	2022-03-16	2024-03-25	136665.00
Conn	Duggon	41 Toban Pass	238-742-6075	cduggon1@about.me	2	2022-06-28	2024-08-16	130229.00
Blane	Moffatt	4 Coolidge Court	340-654-7696	bmoftatt2@imgur.com	3	2022-06-24	2024-07-14	915633.00
Merola	Bonicelli	45113 Hawk Point	130-476-4741	mbonicelli3@com.com	4	2022-04-21	2024-05-13	505675.00
Ora	Streathley	789 Dovetail Avenue	712-554-2784		5	2022-11-11	2024-07-10	554144.00
Amos	Canner	1280 Coolidge Road	980-303-0350	acanner5@amazon.de	6	2022-07-26	2023-10-31	562966.00
Tore	Thick	14 Sundown Hill	237-716-2921	tthick6@furl.net	7	2022-06-23	2023-12-21	554149.00
Graeme	Stonier	66709 Prairie Rose Park	367-178-0954	gstonier7@discuz.net	8	2022-01-17	2023-11-17	259430.00
Ree	Mars	42826 Mandrake Place	665-472-6709	rmars8@businesswire.com	9	2022-04-21	2024-04-27	529774.00
Blair	Yanshonok	48784 Oriole Plaza	258-233-5905		10	2022-06-07	2024-02-22	249851.00
Kermie	Goldthorp	830 Meadow Valley Street	112-381-7579	kgoldthorpa@bloglines.com	11	2022-06-09	2024-01-26	956384.00
Aurelie	Marfe	3957 Sheridan Parkway	207-860-1068	amarfeb@digg.com	12	2022-03-09	2024-02-21	793054.00
Kirk	Toppes	973 Truax Alley	547-179-1491		13	2022-08-28	2024-05-19	219624.00
Aloysia	Pennell	883 Farwell Pass	795-649-4022		14	2022-07-28	2023-11-04	289386.00
Kurtis	Flement	4707 2nd Place	374-910-5361	kflemente@blogger.com	15	2022-06-14	2024-06-17	907728.00
Shellie	Reddin	6 Westerfield Place	710-783-7166	sreddinf@goo.gl	16	2022-01-18	2024-01-10	192856.00
Hettie	Sebring	3 Dwight Court	565-120-8856	hsebringg@blogtalkradio.com	17	2022-07-27	2023-11-09	482966.00
Keith	O'Gormally	031 Algoma Road	827-178-0944	kogormallyh@fotki.com	18	2022-06-28	2023-12-22	468326.00
Ginni	Torres	9110 Schmedeman Point	459-532-7854	gtorresi@devhub.com	19	2022-12-07	2024-06-29	618293.00
Amber	Ridger	45 Dottie Plaza	632-801-9803	aridgerj@wikia.com	20	2022-03-13	2024-12-19	365257.00
Vally	Campes	35522 Village Point	680-698-3149	vcampesk@linkedin.com	21	2022-08-14	2024-11-08	540118.00
Leeann	Halnkin	62847 Marquette Circle	655-376-9739		22	2022-12-01	2024-12-01	239195.00
Cly	Maffei	99 Arapahoe Lane	600-505-9826	cmaffei@bigcartel.com	23	2022-04-25	2024-08-27	322130.00
Bartholomeo	Woffenden	8247 Canary Lane	165-310-0784	bwoffendenn@google.es	24	2022-12-14	2024-09-20	32180.00
Giles	Argue	4 Straubel Way	303-456-6911	gargueo@blog.com	25	2022-04-11	2024-08-29	803931.00
Lamont	Archbold	1 Burning Wood Alley	233-929-1923	larchboldp@intel.com	26	2022-12-06	2023-10-21	181238.00
Joyce	Coulter	06 Forster Trail	818-649-6697	jcoulterq@hc360.com	27	2022-09-09	2024-01-26	300694.00

Рисунок 10.1 – Всі контракти в базі даних

10.2. Запит на визначення середньої кількості товарів на складі для кожної категорії.

Запит виводить всі три категорії товарів, а також середню к-сть товарів для кожних категорії. Для додаткової зручності виводиться також к-сть постачальників, які володіють товарами певної категорії. На рисунку 10.2. показано виведення після запиту до бази даних.

```
SELECT
    p.category,
    ROUND(AVG(p.stock_quantity)) AS average_stock_quantity,
    COUNT(DISTINCT pr.provider_id) AS number_of_providers
FROM product p
JOIN provider pr ON p.provider_id = pr.provider_id
GROUP BY p.category;
```

```
trading_platform=# SELECT
trading_platform=#     p.category,
trading_platform=#     ROUND(AVG(p.stock_quantity)) AS average_stock_quantity,
trading_platform=#     COUNT(DISTINCT pr.provider_id) AS number_of_providers
trading_platform=# FROM product p
trading_platform=# JOIN provider pr ON p.provider_id = pr.provider_id
trading_platform=# GROUP BY p.category;
   category   | average_stock_quantity | number_of_providers
-----+-----+-----
aid           |          5178         |          21
electronic    |           56          |          21
food          |          4576         |          30
(3 rows)
```

Рисунок 10.2 – Середня к-сть товарів на складі

10.3. Запит на виведення інформації про контракти з найбільшою та найменшою вартістю.

Запит виводить прізвище та ім'я клієнта, з яким було укладено даний контракт, а також його вартість. На рисунку 10.3 показано результат такого запиту.

```
SELECT c.client_id, c.name, c.surname,  
       co.total_cost  
FROM contract co  
JOIN client c ON co.client_id = c.client_id  
WHERE co.total_cost = (SELECT MAX(total_cost) FROM contract)  
       OR co.total_cost = (SELECT MIN(total_cost) FROM contract);
```

```
trading_platform=# SELECT c.client_id, c.name, c.surname,  
trading_platform-#         co.total_cost  
trading_platform-# FROM contract co  
trading_platform-# JOIN client c ON co.client_id = c.client_id  
trading_platform-# WHERE co.total_cost = (SELECT MAX(total_cost) FROM contract)  
trading_platform-#         OR co.total_cost = (SELECT MIN(total_cost) FROM contract);  
 client_id | name  | surname | total_cost  
-----+-----+-----+-----  
      290 | Tadeas | Derrett | 999845.00  
      962 | Elga   | Diglin  | 10524.00  
(2 rows)
```

Рисунок 10.3 – Контракти з найбільшою та найменшою вартістю

10.4. Запит на визначення кількості клієнтів, які зробили замовлення та середньої вартості 1 замовлення за кожен місяць.

Запит виводить інтервал часу(у форматі місяць - місяць), власне, середню вартість одного замовлення, а також кількість клієнтів, які робили замовлення у даний місяць. Приклад використання запиту показано на рисунку 10.4.

SELECT

```
TO_CHAR(MIN("order".order_date)::DATE, 'Mon YYYY') || ' - ' ||  
TO_CHAR(MAX("order".order_date)::DATE, 'Mon YYYY') AS month_interval,  
  
COUNT(DISTINCT client.client_id) AS unique_client_count,  
  
ROUND(AVG("order".total_cost), 2) AS average_order_cost  
  
FROM "order"  
  
JOIN client ON "order".client_id = client.client_id  
  
GROUP BY TO_CHAR("order".order_date, 'YYYY-MM')  
  
ORDER BY MIN("order".order_date);
```

```
trading_platform=# SELECT  
trading_platform=# TO_CHAR(MIN("order".order_date)::DATE, 'Mon YYYY') || ' - ' || TO_CHAR(MAX("order".order_date)::DATE, 'Mon YYYY') AS month_interval,  
trading_platform=# COUNT(DISTINCT client.client_id) AS unique_client_count,  
trading_platform=# ROUND(AVG("order".total_cost), 2) AS average_order_cost  
trading_platform=# FROM "order"  
trading_platform=# JOIN client ON "order".client_id = client.client_id  
trading_platform=# GROUP BY TO_CHAR("order".order_date, 'YYYY-MM')  
trading_platform=# ORDER BY MIN("order".order_date);  
 month_interval | unique_client_count | average_order_cost  
-----  
Jan 2022 - Jan 2022 | 81 | 522003.62  
Feb 2022 - Feb 2022 | 72 | 476956.96  
Mar 2022 - Mar 2022 | 84 | 514116.46  
Apr 2022 - Apr 2022 | 80 | 514803.98  
May 2022 - May 2022 | 99 | 531286.75  
Jun 2022 - Jun 2022 | 69 | 545369.10  
Jul 2022 - Jul 2022 | 84 | 485513.15  
Aug 2022 - Aug 2022 | 70 | 460653.07  
Sep 2022 - Sep 2022 | 96 | 503084.01  
Oct 2022 - Oct 2022 | 85 | 527863.53  
Nov 2022 - Nov 2022 | 81 | 503596.27  
Dec 2022 - Dec 2022 | 99 | 463450.57  
(12 rows)  
  
trading_platform=#
```

Рисунок 10.4 – Кількість клієнтів, які зробили замовлення за кожен місяць

10.5. Запит про виведення усіх клієнтів, котрі з запізненням оплатили контракт.

Не всі клієнти встигають оплатити свої укладені контракти, тому даний запит допомагає відслідковувати таких людей та виводить про них інформацію. Результат запиту на рисунку 10.5.

```
SELECT cl.client_id, cl.name, cl.surname,  
       co.contract_id, co.start_date AS contract_start_date,  
       pa.payment_date  
FROM client cl  
JOIN contract co ON cl.client_id = co.client_id  
JOIN payment pa ON co.contract_id = pa.payment_id  
WHERE pa.payment_date > co.start_date;
```

```
trading_platform=# SELECT cl.client_id, cl.name, cl.surname,  
trading_platform-#       co.contract_id, co.start_date AS contract_start_date,  
trading_platform-#       pa.payment_date  
trading_platform-# FROM client cl  
trading_platform-# JOIN contract co ON cl.client_id = co.client_id  
trading_platform-# JOIN payment pa ON co.contract_id = pa.payment_id  
trading_platform-# WHERE pa.payment_date > co.start_date;  
 client_id | name   | surname | contract_id | contract_start_date | payment_date  
-----  
      23 | Cly    | Maffei  |      23 | 2022-04-25 | 2022-04-29  
      34 | Morey  | Titterrell |      34 | 2022-01-17 | 2022-01-18  
      44 | Augustine | Durnin |      44 | 2022-02-25 | 2022-02-27  
      88 | Leland | McCombe |      88 | 2022-03-29 | 2022-03-30  
     122 | Red    | Littlefield |     122 | 2022-08-19 | 2022-08-29  
     128 | Doe    | Rowdell  |     128 | 2022-05-16 | 2022-05-18  
     253 | Jase   | Bambara  |     253 | 2022-12-09 | 2022-12-10  
     391 | Nara   | Irons    |     391 | 2022-04-18 | 2022-04-19  
     399 | Lana   | Beteriss |     399 | 2022-01-07 | 2022-01-08  
     439 | Reinold | Lowrance |     439 | 2022-10-13 | 2022-10-14  
     444 | Berny  | Stacey   |     444 | 2022-10-20 | 2022-10-21  
     478 | Goldi  | Haggerwood |     478 | 2022-12-16 | 2022-12-19  
     526 | Rutledge | Brunone |     526 | 2022-01-21 | 2022-01-23  
     543 | Cristin | Greenhough |     543 | 2022-09-20 | 2022-09-22  
     574 | Maddy  | Lionel   |     574 | 2022-03-28 | 2022-03-29  
     585 | Audrye | Sidebottom |     585 | 2022-06-04 | 2022-06-06  
     586 | Helena | Tonkes   |     586 | 2022-01-26 | 2022-01-27  
     641 | Sigvard | Tuffell  |     641 | 2022-06-29 | 2022-06-30  
     651 | Merl   | Beri     |     651 | 2022-05-09 | 2022-05-17  
     705 | Ronnie | Broadberry |     705 | 2022-12-15 | 2022-12-17  
     745 | Shaylah | Whales   |     745 | 2022-11-10 | 2022-11-11  
     746 | Riccardo | Frankiss |     746 | 2022-01-13 | 2022-01-16  
     784 | Madelle | Bourdice |     784 | 2022-12-18 | 2022-12-19  
     803 | Aldon  | Whitewood |     803 | 2022-02-25 | 2022-02-26  
     880 | Gillian | Mackerel |     880 | 2022-12-25 | 2022-12-29  
     886 | Sarena | Pavy     |     886 | 2022-05-15 | 2022-05-17  
     953 | Sib    | MacGuigan |     953 | 2022-06-09 | 2022-06-11  
     964 | Palm   | Sollam   |     964 | 2022-02-04 | 2022-02-08  
     986 | Dorene | Krollman |     986 | 2022-07-19 | 2022-07-22  
     992 | Iain   | Fowell   |     992 | 2022-12-20 | 2022-12-21  
     997 | Zorina | Luckwell |     997 | 2022-09-29 | 2022-09-30  
(31 rows)
```

Рисунок 10.5 – Клієнти, які оплатили із запізненням

10.6. Запит на визначення кількості та загальної вартості укладених контрактів для кожного постачальника.

Запит виводить постачальника, кількість контрактів, яку він уклав та загальну суму за всі контракти. Для наглядності також додано поле із найбільш дорогим контрактом. Результат запити відображено на Рисунок 10.6.

```
SELECT  
  
    pr.provider_id,  
  
    pr.name AS provider_name,  
  
    COUNT(co.contract_id) AS contract_count,  
  
    SUM(co.total_cost) AS total_contract_cost,  
  
    MAX(co.total_cost) AS most_expensive_contract  
  
FROM provider pr  
  
LEFT JOIN contract co ON pr.provider_id = co.provider_id  
  
GROUP BY pr.provider_id, pr.name  
  
ORDER BY pr.provider_id;
```

```
trading_platform=# SELECT
trading_platform=#     pr.provider_id,
trading_platform=#     pr.name AS provider_name,
trading_platform=#     COUNT(co.contract_id) AS contract_count,
trading_platform=#     SUM(co.total_cost) AS total_contract_cost,
trading_platform=#     MAX(co.total_cost) AS most_expensive_contract
trading_platform=# FROM provider pr
trading_platform=# LEFT JOIN contract co ON pr.provider_id = co.provider_id
trading_platform=# GROUP BY pr.provider_id, pr.name
trading_platform=# ORDER BY pr.provider_id;
```

provider_id	provider_name	contract_count	total_contract_cost	most_expensive_contract
1	Sloan	19	9565269.00	969612.00
2	Pascale	18	8380647.00	865414.00
3	Noak	21	11056548.00	972858.00
4	Hartley	23	14407748.00	994272.00
5	Lib	16	7245634.00	998664.00
6	Philipa	20	9419754.00	998325.00
7	Junia	17	9040137.00	942485.00
8	Thedric	22	8714518.00	947980.00
9	Dianemarie	22	10967382.00	963144.00
10	Devondra	21	10317287.00	954826.00
11	Maura	22	13522735.00	990920.00
12	Kyle	23	10990088.00	990653.00
13	Kristoforo	29	15743304.00	988957.00
14	Oren	17	8793546.00	927403.00
15	Griffie	12	5195635.00	995101.00
16	Imogene	17	10495877.00	999845.00
17	Jedd	25	12576556.00	986597.00
18	Gunar	25	14259773.00	977420.00
19	Marji	28	11548516.00	932947.00
20	Golda	23	12673799.00	938399.00
21	Guy	20	9064091.00	923767.00
22	Cinda	26	14522385.00	975745.00
23	Giovanna	15	7919170.00	901428.00
24	Mano	13	6598443.00	911213.00
25	Germana	20	9293122.00	991295.00
26	Alfie	20	8774838.00	907226.00
27	Celene	21	9431195.00	994794.00
28	Madelyn	17	8917055.00	971578.00
29	Hersh	20	9493768.00	911679.00
30	Estelle	22	11106776.00	956384.00
31	Cathrine	23	14329605.00	983437.00
32	Rosita	16	8158168.00	973409.00
33	Kellyann	17	7454793.00	858669.00
34	Dicky	26	11833190.00	979764.00
35	Ivette	19	9594311.00	788643.00
36	Vaughn	21	11034938.00	949663.00

Рисунок 10.6 – Контракти постачальників

10.7. Запит на визначення кількості відгуків та середньої оцінки для кожного провайдера.

Запит виводить інформацію про відгуки, написані на кожного постачальника, а також середній рейтинг на основі цих відгуків. Результат запиту відображено на рисунку 10.7.

```
SELECT
```

```
    pr.provider_id,
```

```
    pr.name AS provider_name,
```

```
    COUNT(rv.review_id) AS review_count,
```

```
    ROUND(AVG(rv.rating), 2) AS average_rating
```

```
FROM provider pr
```

```
LEFT JOIN review rv ON pr.provider_id = rv.provider_id
```

```
GROUP BY pr.provider_id, pr.name
```

```
ORDER BY pr.provider_id;
```

```
trading_platform=# SELECT
trading_platform=#     pr.provider_id,
trading_platform=#     pr.name AS provider_name,
trading_platform=#     COUNT(rv.review_id) AS review_count,
trading_platform=#     ROUND(AVG(rv.rating), 2) AS average_rating
trading_platform=# FROM provider pr
trading_platform=# LEFT JOIN review rv ON pr.provider_id = rv.provider_id
trading_platform=# GROUP BY pr.provider_id, pr.name
trading_platform=# ORDER BY pr.provider_id;
```

provider_id	provider_name	review_count	average_rating
1	Sloan	10	94.90
2	Pascale	7	34.29
3	Noak	2	68.50
4	Hartley	5	58.20
5	Lib	5	62.60
6	Philipa	6	60.33
7	Junia	11	28.55
8	Theatric	9	50.67
9	Dianemarie	11	32.00
10	Devondra	3	74.33
11	Maura	6	58.50
12	Kyle	7	70.71
13	Kristoforo	3	43.33
14	Oren	4	31.50
15	Griffie	9	30.22
16	Imogene	4	77.25
17	Jedd	7	53.29
18	Gunar	13	22.85
19	Marji	3	56.33
20	Golda	3	66.00
21	Guy	10	90.00
22	Cinda	5	78.20
23	Giovanna	5	92.40
24	Mano	6	78.50
25	Germana	4	72.50
26	Alfie	10	26.50
27	Celene	1	47.00
28	Madelyn	13	34.15
29	Hersh	8	36.63
30	Estelle	6	61.83
31	Cathrine	3	12.67
32	Rosita	4	61.75
33	Kellyann	8	32.25
34	Dicky	6	36.17
35	Ivette	5	30.80
36	Vaughn	8	47.75
37	Godfry	6	48.17
38	Natassia	6	27.50
39	Melisenda	6	42.00

Рисунок 10.7 – Відгуки по провайдерах

10.8. Запит на виведення лише тих ревію, в котрих присутній коментар.

Запит відбере лише ті ревію, в котрих присутній коментар від клієнта, так як вони несуть більшу цінність. Результат запити відображено на рисунку 10.8.

SELECT

r.review_id, r.comment,
c.name AS client_name,
p.name AS provider_name

FROM review r

RIGHT JOIN client c ON r.client_id = c.client_id

RIGHT JOIN provider p ON r.provider_id = p.provider_id

WHERE r.comment IS NOT NULL;

```
trading_platform=# SELECT
trading_platform=#     r.review_id, r.comment,
trading_platform=#     c.name AS client_name,
trading_platform=#     p.name AS provider_name
trading_platform=# FROM review r
trading_platform=# RIGHT JOIN client c ON r.client_id = c.client_id
trading_platform=# RIGHT JOIN provider p ON r.provider_id = p.provider_id
trading_platform=# WHERE r.comment IS NOT NULL;
```

review_id	comment	client_name	provider_name
36	Extremely slow delivery and substandard products. A terrible experience overall.	Conn	Madelyn
186	Top-quality goods and excellent customer service. A supplier that stands out from the rest.	Graeme	Guy
102	Consistently subpar products. I would not recommend this supplier to anyone.	Vally	Devondra
30	Average quality products delivered within the expected timeframe.	Leeann	Estelle
277	The level of professionalism sets this supplier apart. A true asset to any business.	Karita	Kristian
134	Customer care is non-responsive. No effort made to address concerns or rectify issues.	Kelli	Gunar
212	Consistently exceeds expectations. I confidently recommend this supplier to others.	Gabbi	Imogene
274	Consistently fails to meet expectations. A supplier not worthy of trust or investment.	Alyosha	Kellyann
104	Poor quality products and unhelpful service. Definitely not worth the investment.	Gilligan	Kristoforo
1	Unreliable service! Products received were of poor quality, and delivery was extremely delayed.	Renee	Marve
281	The overall experience was average. The supplier delivered on the basics without exceeding expectations.	Lesley	Dianemarie
154	Subpar customer service. No effort made to rectify issues or address concerns.	Doe	Susanetta
11	Reliable and consistent quality. It's a pleasure doing business with such a trustworthy supplier.	Elva	Estelle
39	The product met basic standards, but there is room for improvement in terms of quality.	Gallagher	Hartley
203	Exceptional product knowledge. Their expertise adds tremendous value to our partnership.	Isidro	Giovanna
164	Reliable and trustworthy. I appreciate the transparency and integrity in every transaction.	Nanni	Guy
121	Consistently delivers on time. A supplier you can count on for reliability.	Andie	Guy
70	Unreliable delivery times and inconsistent quality. A regrettable choice for our business needs.	Kelbee	Pascale
47	Fairly standard service. The supplier fulfilled the order without exceeding expectations.	Sky	Justinian

Рисунок 10.8 – Рев'ю з присутнім коментарем

10.9. Запит на виведення переможця кожного тендеру.

Запит виводить інформацію про постачальників, котрі виграли кожен тендер. Результат запити відображено на рисунку 10.9.

SELECT

t.tender_id, t.start_date, t.end_date, t.current_bid AS final_price,

MAX(b.amount) AS bid_count,

p.name AS provider_name,

p.surname AS provider_surname

FROM tender t

LEFT JOIN bid b ON t.tender_id = b.tender_id

LEFT JOIN provider p ON b.provider_id = p.provider_id

GROUP BY t.tender_id, p.provider_id

ORDER BY t.tender_id;

```
trading_platform=# SELECT
trading_platform=#   t.tender_id, t.start_date, t.end_date, t.current_bid AS final_price,
trading_platform=#   MAX(b.amount) AS bid_count,
trading_platform=#   p.name AS provider_name,
trading_platform=#   p.surname AS provider_surname
trading_platform=# FROM tender t
trading_platform=# LEFT JOIN bid b ON t.tender_id = b.tender_id
trading_platform=# LEFT JOIN provider p ON b.provider_id = p.provider_id
trading_platform=# GROUP BY t.tender_id, p.provider_id
trading_platform=# ORDER BY t.tender_id;
```

tender_id	start_date	end_date	final_price	bid_count	provider_name	provider_surname
1	2023-12-29	2023-12-30	41196	16	Giovanna	Flawn
2	2023-11-01	2023-11-02	8010	6	Cinda	Raeburn
3	2023-11-07	2024-01-08	43862	5	Kristian	Rolfo
4	2023-11-24	2023-11-25	38094	11	Kyle	MacIlwrick
5	2023-12-29	2023-12-30	33598	14	Germana	Kosiada
6	2023-12-14	2024-01-06	25867	4	Philipa	Loukes
7	2023-12-02	2023-12-03	32105	22	Lib	Kynnd
8	2023-11-20	2024-01-12	47433	5	Mano	Castiello
9	2023-12-04	2024-01-27	29799	19	Rosita	Nund
10	2023-12-15	2023-12-16	14481	12	Maura	Labdon
11	2023-11-29	2023-11-30	8910	3	Sandor	Wilber
12	2023-11-01	2024-01-08	47211	3	Guy	Marwick
13	2023-12-08	2023-12-09	33176	27	Sloan	Lindsay
14	2023-11-18	2023-11-19	15553	26	Ted	Hobbema
15	2023-11-11	2024-01-09	32705	4	Justinian	Filippyev
16	2023-11-24	2024-01-17	26622	22	Jedd	Bamber
17	2023-12-18	2023-12-19	38266	11	Imogene	Gudgen
18	2023-12-24	2023-12-25	26957	28	Noak	Sibbs
19	2023-11-09	2023-11-10	40742	16	Estelle	Pond
20	2023-12-11	2023-12-12	24508	19	Vaughn	Cregg
21	2023-12-14	2024-01-18	30898	7	Durand	Bohden
22	2023-12-07	2023-12-08	45779	14	Marji	Lickorish
23	2023-12-14	2023-12-15	22118	27	Kristoforo	Starsmeare
24	2023-12-02	2024-01-28	49942	5	Sloan	Lindsay
25	2023-12-14	2024-01-11	39176	30	Golda	Mularkey
26	2023-11-11	2023-11-12	17345	23	Celene	Gypson
27	2023-12-05	2023-12-06	11014	15	Hartley	Olyff
28	2023-12-10	2024-01-15	43121	22	Devondra	Mussettini
29	2023-12-03	2023-12-04	35827	24	Godfry	Shingfield
30	2023-12-13	2024-01-06	44942	28	Vivyanne	Freak

(30 rows)

Рисунок 10.9 – Переможці тендерів

10.10. Запит на виведення списку послуг, доступних у певному районі обслуговування.

Запит дозволяє переглянути усі доступні послуги, які доступні лише у контретному заданому районі обслуговування. У нашому випадку це Азія. Результат запиту на рисунку 10.10.

```
SELECT
```

```
    s.service_id,
```

```
    s.name AS service_name,
```

```
    s.description,
```

```
    s.price,
```

```
    s.availability,
```

```
    p.name AS provider_name,
```

```
    p.service_area
```

```
FROM service s
```

```
JOIN provider p ON s.provider_id = p.provider_id
```

```
WHERE p.service_area = 'Asia'
```

```
ORDER BY provider_name;
```

```
trading_platform=# SELECT
trading_platform=#   s.service_id,
trading_platform=#   s.name AS service_name,
trading_platform=#   s.description,
trading_platform=#   s.price,
trading_platform=#   s.availability,
trading_platform=#   p.name AS provider_name,
trading_platform=#   p.service_area
trading_platform=# FROM service s
trading_platform=# JOIN provider p ON s.provider_id = p.provider_id
trading_platform=# WHERE p.service_area = 'Asia'
trading_platform=# ORDER BY provider_name;
```

service_id	service_name	description	price	availability	provider_name	service_area
15	Product Customization Hub	Tailor your products to your exact specifications with our customization service.	350.00	unavailable	Cathrine	Asia
51	Personal Shopper Assistant	Expert advice and personalized shopping assistance to help you choose the perfect product.	152.00	available	Germana	Asia
57	Priority Restock Alert	An exclusive notification service for clients about the restocking of high-demand products.	312.00	available	Germana	Asia
63	Instant Refund Processing	Accelerated refund service for quick and hassle-free returns.	48.00	available	Germana	Asia
8	Home Trial Experience	Service allowing customers to try products at home before finalizing the purchase.	223.00	available	Godfry	Asia
17	Health and Safety Certification	Products come with a verified health and safety certification for peace of mind.	347.00	available	Godfry	Asia
84	Extended Warranty Guard	Additional warranty service that extends the standard coverage period for your purchased goods.	469.00	available	Godfry	Asia
80	One-to-One Training Sessions	Personalized training and tutorials for effective and optimal use of your products.	98.00	available	Junia	Asia
86	Custom Product Tailoring	Personalization service for products, allowing customization to fit individual preferences.	232.00	available	Kristoforo	Asia
81	Family & Friends Referral Rewards	Earn rewards and discounts by referring our products to your loved ones.	486.00	available	Kristoforo	Asia
100	24/7 Product Support Line	Around-the-clock customer support line for immediate assistance with your products.	441.00	available	Kristoforo	Asia
92	VIP Access Pass	Exclusive membership providing early access to new releases and special discounts on products.	192.00	available	Marji	Asia
96	Smart Compatibility Check	Ensure your new purchase seamlessly integrates with your existing smart devices.	87.00	available	Marji	Asia
58	Tech Setup Assistance	Specialized service for setting up and configuring new technological devices.	265.00	available	Norine	Asia
59	Product Upcycle Program	A sustainable service offering to upgrade or recycle your old products when purchasing new ones.	29.00	available	Sandor	Asia
82	Long-Term Storage Option	Secure storage services for your products, ideal for space-saving or temporary needs.	110.00	available	Sandor	Asia
69	Health and Safety Certification	Products come with a verified health and safety certification for peace of mind.	2.00	available	Sandor	Asia
16	Early Bird Delivery	Get your products delivered first thing in the morning with our early delivery option.	0.00	available	Sloan	Asia
10	VIP Access Pass	Exclusive membership providing early access to new releases and special discounts on products.	0.00	available	Sloan	Asia
3	24/7 Product Support Line	Around-the-clock customer support line for immediate assistance with your products.	0.00	available	Sloan	Asia
2	Extended Warranty Guard	Additional warranty service that extends the standard coverage period for your purchased goods.	0.00	available	Sloan	Asia
1	Express Shipping Plus	Premium fast shipping service ensuring expedited delivery of your products.	0.00	available	Sloan	Asia
30	Long-Term Storage Option	Secure storage services for your products, ideal for space-saving or temporary needs.	428.00	available	Sloan	Asia

(23 rows)

Рисунок 10.10 – Послуги у певному районі(Азія)

10.11. Запит для виведення контрактів, котрі в найближчий місяць стануть не дійсні.

Наступний запит виводить усі запити, у котрих завершиться термін дії у найближчий місяць. Результат запиту відображено на рисунку 10.11.

```
SELECT
    c.contract_id,
    c.end_date,
    c.total_cost,
    cl.name AS client_name,
    cl.surname AS client_surname,
    p.name AS provider_name,
    p.surname AS provider_surname
FROM contract c
JOIN client cl ON c.client_id = cl.client_id
JOIN provider p ON c.provider_id = p.provider_id
WHERE c.end_date > CURRENT_DATE AND c.end_date <= CURRENT_DATE
+ INTERVAL '1 month';
```

```
trading_platform=# SELECT
trading_platform=#     c.contract_id,
trading_platform=#     c.end_date,
trading_platform=#     c.total_cost,
trading_platform=#     cl.name AS client_name,
trading_platform=#     cl.surname AS client_surname,
trading_platform=#     p.name AS provider_name,
trading_platform=#     p.surname AS provider_surname
trading_platform=# FROM contract c
trading_platform=# JOIN client cl ON c.client_id = cl.client_id
trading_platform=# JOIN provider p ON c.provider_id = p.provider_id
trading_platform=# WHERE c.end_date > CURRENT_DATE AND c.end_date <= CURRENT_DATE + INTERVAL '1 month';
 contract_id | end_date | total_cost | client_name | client_surname | provider_name | provider_surname
-----
11 | 2024-01-26 | 956384.00 | Kermie | Goldthorp | Estelle | Pond
16 | 2024-01-10 | 192856.00 | Shellie | Reddin | Dianemarie | Baudts
27 | 2024-01-26 | 300694.00 | Joyce | Coulter | Kristoforo | Starsmeare
36 | 2024-01-23 | 222513.00 | Howard | Baake | Hartley | Olyff
40 | 2024-01-28 | 694877.00 | Kelli | Korting | Ivette | Silberschatz
48 | 2024-01-25 | 513438.00 | Marie-ann | Offin | Mano | Castiello
50 | 2024-01-27 | 497758.00 | Ethelred | Ible | Justinian | Filippyev
52 | 2024-01-02 | 714406.00 | Dolly | Hansen | Hartley | Olyff
58 | 2024-01-10 | 109242.00 | Gaynor | Cohrs | Dicky | Liepins
63 | 2024-01-17 | 280295.00 | Scottie | Torrijos | Kellyann | McGaugie
69 | 2024-01-15 | 64504.00 | Harv | Whawell | Germana | Kosiada
87 | 2024-01-06 | 489441.00 | Agosto | Laise | Sandor | Wilber
104 | 2024-01-13 | 110326.00 | Wynne | Culp | Kristian | Rolfo
118 | 2024-01-07 | 228365.00 | Anett | Reeme | Godfry | Shingfield
130 | 2024-01-19 | 197376.00 | Patricia | Ashlee | Susanetta | Froom
159 | 2024-01-09 | 536076.00 | Cherilyn | Devil | Imogene | Gudgen
162 | 2024-01-26 | 36424.00 | Kelwin | McCrorie | Vaughn | Cregg
166 | 2024-01-23 | 689636.00 | Royal | Rawles | Rosita | Nund
173 | 2024-01-05 | 913461.00 | Winifield | Ferrotti | Susanetta | Froom
177 | 2024-01-08 | 325740.00 | Dulcea | Lowen | Guy | Marwick
212 | 2024-01-29 | 525522.00 | Fair | Monnoyer | Thedric | Doumic
245 | 2024-01-29 | 280449.00 | Flynn | Edwinson | Vivyanne | Freak
251 | 2024-01-28 | 945472.00 | Alicia | O'Sesnane | Rosita | Nund
316 | 2024-01-26 | 755129.00 | Rice | Danielsohn | Imogene | Gudgen
319 | 2024-01-28 | 236741.00 | Oralee | Krol | Gunar | McGowing
344 | 2024-01-30 | 554025.00 | Kathlin | Elflain | Kristian | Rolfo
350 | 2024-01-15 | 184193.00 | Alia | Wannop | Dicky | Liepins
354 | 2024-01-21 | 131433.00 | Mechelle | Weatherburn | Dianemarie | Baudts
359 | 2024-01-16 | 305593.00 | Marcy | Besnard | Philipa | Loukes
372 | 2024-01-13 | 849675.00 | Burnard | Gasking | Jedd | Bamber
379 | 2024-01-22 | 473165.00 | Wendeline | Skrzynski | Ted | Hobbema
380 | 2024-01-17 | 472998.00 | Margarethe | Coultous | Giovanna | Flawn
383 | 2024-01-08 | 701230.00 | Theodoric | Bonellie | Vaughn | Cregg
396 | 2024-01-03 | 736880.00 | Phyllys | Hunday | Kyle | MacIlwrick
402 | 2024-01-21 | 395861.00 | Jeffry | Balmer | Sloan | Lindsay
```

Рисунок 10.11 – Контракти, що скоро стануть недійсними

10.12. Запит для виведення найбільш активних клієнтів за сумою, витраченою на замовлення.

Запит виводить 10 клієнтів у вигляді топу. За критерій береться сума, яку вони витратили на всі свої замовлення. Результат запити відображено на рисунку 10.12.

```
SELECT
    cl.name AS client_name,
    cl.surname AS client_surname,
    MAX(o.total_cost) AS highest_order_cost
FROM client cl
JOIN "order" o ON cl.client_id = o.client_id
GROUP BY cl.client_id
ORDER BY highest_order_cost DESC
LIMIT 10;
```

```
trading_platform=# SELECT
trading_platform=#     cl.name AS client_name,
trading_platform=#     cl.surname AS client_surname,
trading_platform=#     MAX(o.total_cost) AS highest_order_cost
trading_platform=# FROM client cl
trading_platform=# JOIN "order" o ON cl.client_id = o.client_id
trading_platform=# GROUP BY cl.client_id
trading_platform=# ORDER BY highest_order_cost DESC
trading_platform=# LIMIT 10;
 client_name | client_surname | highest_order_cost
-----+-----+-----
Tadeas      | Derrett        | 999845.00
Glen        | Wyre           | 998664.00
Cy          | Paskin         | 998325.00
Emlynne     | Landreth       | 995101.00
David       | McQuillin      | 994794.00
Rutger      | Summersett     | 994272.00
Rutledge    | Brunone        | 991295.00
Cookie      | Twinbourne     | 990920.00
Arman       | Skentelbury    | 990653.00
Cyndie      | Sandes         | 989743.00
(10 rows)
```

Рисунок 10.12 – Найактивніші клієнти

10.13. Запит для визначення середньої вартості контрактів по кожному постачальнику.

Запит виводить інформацію про найдешевший та найдорожчий контракти для постачальників, а також їх середню вартість. Результат запиту відображено на рисунку 10.13.

SELECT

```
pr.name AS provider_name,  
pr.telephone AS phone_number,  
ROUND(AVG(co.total_cost), 2) AS average_contract_cost,  
MIN(co.total_cost) AS least_expensive_contract,  
MAX(co.total_cost) AS most_expensive_contract
```

FROM provider pr

JOIN contract co ON pr.provider_id = co.provider_id

GROUP BY pr.provider_id;

```
trading_platform=# SELECT  
trading_platform=# pr.name AS provider_name,  
trading_platform=# pr.telephone AS phone_number,  
trading_platform=# ROUND(AVG(co.total_cost), 2) AS average_contract_cost,  
trading_platform=# MIN(co.total_cost) AS least_expensive_contract,  
trading_platform=# MAX(co.total_cost) AS most_expensive_contract  
trading_platform=# FROM provider pr  
trading_platform=# JOIN contract co ON pr.provider_id = co.provider_id  
trading_platform=# GROUP BY pr.provider_id;
```

provider_name	phone_number	average_contract_cost	least_expensive_contract	most_expensive_contract
Cinda	191-778-5332	558553.27	36810.00	975745.00
Maura	263-271-5666	614669.77	211794.00	990920.00
Susanetta	443-920-0485	481313.25	23527.00	989743.00
Kristian	763-454-8835	461913.06	64449.00	916908.00
Kalli	491-635-4452	466243.67	48079.00	888862.00
Norine	548-977-5599	521076.44	15138.00	926323.00
Dianemarie	379-921-2503	498517.36	96586.00	963144.00
Griffie	183-156-4135	432969.58	31159.00	995101.00
Vivyanne	894-843-9338	460799.07	91445.00	989402.00
Alfie	427-248-9601	438741.90	66016.00	907226.00
Marji	676-113-7462	412447.00	73678.00	932947.00
Estelle	530-543-5968	504853.45	13218.00	956384.00
Guy	657-589-4244	453204.55	13134.00	923767.00
Noak	750-476-7998	526502.29	28525.00	972858.00
Jedd	765-956-0447	503062.24	47149.00	986597.00
Godfry	195-668-5662	593201.17	90956.00	985853.00
Madelyn	471-228-8134	524532.65	90774.00	971578.00
Lib	823-452-2342	452852.13	45481.00	998664.00
Hersh	625-273-6992	474688.40	66981.00	911679.00
Hartley	264-637-0651	626423.83	105570.00	994272.00
Dicky	250-817-9833	455122.69	58680.00	979764.00
Devondra	628-943-5090	491299.38	30379.00	954826.00
Ivette	182-345-1037	504963.74	63275.00	788643.00

Рисунок 10.13 – Середня вартість контрактів для постачальника

10.14. Запит для отримання інформації про всі сервіси, які надаються вибраними провайдерами.

Запит виводить основну інформацію про всі послуги, котрі надає один обраний за іменем постачальник. Результат запиту відображено на рисунку 10.14.

```
SELECT
    s.name AS service_name,
    s.description,
    p.name AS provider_name
FROM service s
JOIN provider p ON s.provider_id = p.provider_id
WHERE p.provider_id IN (SELECT provider_id FROM provider WHERE name
IN ('Sloan'));
```

```
trading_platform=# SELECT
trading_platform=#     s.name AS service_name,
trading_platform=#     s.description,
trading_platform=#     p.name AS provider_name
trading_platform=# FROM service s
trading_platform=# JOIN provider p ON s.provider_id = p.provider_id
trading_platform=# WHERE p.provider_id IN (SELECT provider_id FROM provider WHERE name IN ('Sloan'));
 service_name | description | provider_name
-----|-----|-----
Express Shipping Plus | Premium fast shipping service ensuring expedited delivery of your products. | Sloan
Extended Warranty Guard | Additional warranty service that extends the standard coverage period for your purchased goods. | Sloan
24/7 Product Support Line | Around-the-clock customer support line for immediate assistance with your products. | Sloan
VIP Access Pass | Exclusive membership providing early access to new releases and special discounts on products. | Sloan
Early Bird Delivery | Get your products delivered first thing in the morning with our early delivery option. | Sloan
Long-Term Storage Option | Secure storage services for your products, ideal for space-saving or temporary needs. | Sloan
(6 rows)
```

Рисунок 10.14 – Послуги обраного провайдера

10.15. Запит на виведення контрактів, вартість яких становить більше 950 тисяч.

Запит виводить усі контракти, вартість яких більше 950 тисяч та інформацію про них. Результат запиту відображено на рисунку 10.15.

SELECT

c.name AS client_name,
pr.name AS provider_name,
co.contract_id,
pa.cost AS cost,
co.start_date AS contract_start_date,
co.end_date AS contract_end_date

FROM client c

JOIN contract co ON c.client_id = co.client_id

JOIN provider pr ON co.provider_id = pr.provider_id

JOIN payment pa ON co.contract_id = pa.payment_id

WHERE pa.cost > 950000;


```
trading_platform=# SELECT
trading_platform=#     c.name AS client_name,
trading_platform=#     pr.name AS provider_name,
trading_platform=#     co.contract_id,
trading_platform=#     pa.cost AS cost,
trading_platform=#     co.start_date AS contract_start_date,
trading_platform=#     co.end_date AS contract_end_date
trading_platform=# FROM client c
trading_platform=# JOIN contract co ON c.client_id = co.client_id
trading_platform=# JOIN provider pr ON co.provider_id = pr.provider_id
trading_platform=# JOIN payment pa ON co.contract_id = pa.payment_id
trading_platform=# WHERE pa.cost > 950000;
 client_name | provider_name | contract_id | cost | contract_start_date | contract_end_date
-----+-----+-----+-----+-----+-----
Kermie      | Estelle      | 11          | 956384 | 2022-06-09         | 2024-01-26
Rena        | Rosita       | 129         | 973409 | 2022-06-04         | 2024-09-16
Arman       | Kyle         | 134         | 990653 | 2022-02-28         | 2023-10-16
Linda       | Dicky        | 136         | 959621 | 2022-06-05         | 2024-07-04
Korie       | Godfry       | 169         | 985853 | 2022-02-19         | 2024-12-30
Rutger      | Hartley      | 226         | 994272 | 2022-06-09         | 2024-03-11
Hertha      | Imogene      | 266         | 964516 | 2022-06-26         | 2024-03-13
Tadeas      | Imogene      | 290         | 999845 | 2022-07-19         | 2024-10-13
Kakalina    | Noak         | 322         | 972858 | 2022-11-16         | 2024-12-29
Benjamin    | Dicky        | 360         | 974876 | 2022-03-01         | 2024-06-24
Caren       | Philipa      | 387         | 953276 | 2022-05-11         | 2024-07-16
Harold      | Kyle         | 411         | 957831 | 2022-03-17         | 2024-09-02
Teodora     | Sloan        | 428         | 969612 | 2022-01-02         | 2024-12-18
Lorilyn     | Gunar        | 462         | 957161 | 2022-05-07         | 2023-12-31
Shelbi      | Dianemarie   | 470         | 963144 | 2022-06-13         | 2024-09-07
Cy          | Philipa      | 475         | 998325 | 2022-09-02         | 2024-11-04
Antonin     | Gunar        | 492         | 957893 | 2022-07-02         | 2024-01-06
Rutledge    | Germana      | 526         | 991295 | 2022-01-21         | 2023-12-08
Alick       | Cinda        | 539         | 950113 | 2022-09-20         | 2024-03-18
Alejandro   | Cathrine     | 552         | 983437 | 2022-01-23         | 2024-11-23
Sanford     | Hartley      | 569         | 955559 | 2022-01-02         | 2023-10-17
Zia         | Sandor       | 621         | 989154 | 2022-12-22         | 2024-12-14
Emlynne     | Griffie      | 628         | 995101 | 2022-05-28         | 2024-08-30
Beniamino   | Susanetta    | 635         | 962253 | 2022-08-16         | 2024-10-29
Brinn       | Devondra     | 656         | 954826 | 2022-05-20         | 2023-10-15
Brinn       | Justinian    | 677         | 973586 | 2022-12-03         | 2024-11-18
Kale        | Marve        | 708         | 973056 | 2022-04-14         | 2023-11-17
Cyndie      | Susanetta    | 713         | 989743 | 2022-08-14         | 2024-03-29
Cornelia    | Natassia     | 740         | 958704 | 2022-12-06         | 2024-06-12
Bond        | Kristoforo   | 755         | 988957 | 2022-02-08         | 2024-01-03
Dinnie     | Dicky        | 758         | 979764 | 2022-11-02         | 2024-02-20
Aubrette    | Jedd         | 765         | 986597 | 2022-09-10         | 2024-08-26
Glen        | Lib          | 781         | 998664 | 2022-10-21         | 2024-04-22
David       | Celene       | 801         | 994794 | 2022-04-22         | 2023-10-31
Erena       | Vivyanne     | 811         | 989402 | 2022-04-26         | 2024-07-27
Carlynn     | Gunar        | 845         | 977420 | 2022-10-14         | 2024-02-10
```

Рисунок 10.15 – Контракти, дорожчі за 950000

10.16. Запит для визначення провайдерів, які надають товари в певній категорії.

Запит виводить постачальників, які володіють товарами у певній категорії(у нашому випадку, в категорії “Електроніка”). На рисунку 10.16 показано результат такого запиту.

SELECT

```
pr.name AS provider_name,
pr.address AS provider_address,
pr.telephone AS provider_telephone,
p.name AS product_name,
p.price,
p.category
```

FROM provider pr

JOIN product p ON pr.provider_id = p.provider_id

WHERE p.category = 'electronic'

ORDER BY pr.name;

```
trading platform=# SELECT
trading platform=# pr.name AS provider_name,
trading platform=# pr.address AS provider_address,
trading platform=# pr.telephone AS provider_telephone,
trading platform=# p.name AS product_name,
trading platform=# p.price,
trading platform=# p.category
trading platform=# FROM provider pr
trading platform=# JOIN product p ON pr.provider_id = p.provider_id
trading platform=# WHERE p.category = 'electronic'
trading platform=# ORDER BY pr.name;
```

provider_name	provider_address	provider_telephone	product_name	price	category
Celene	12 Cody Center	888-383-4022	Virtual Reality Kit "VR Supreme"	800.00	electronic
Dicky	98113 Dennis Parkway	250-817-9833	Automatic Coffee Maker "CoffeeMaster Deluxe"	250.00	electronic
Durand	20 Banding Parkway	328-537-1538	Solar Power Bank "SolarPower Max"	300.00	electronic
Estelle	696 Meadow Vale Trail	530-543-5968	Premium Sound System "SoundWave Deluxe"	879.00	electronic
Estelle	696 Meadow Vale Trail	530-543-5968	Modern Robot Vacuum "CleanBot Turbo"	300.00	electronic
Estelle	696 Meadow Vale Trail	530-543-5968	Gaming Laptop "GameMaster X9"	2500.00	electronic
Germana	8 Darwin Circle	957-624-6853	3D Printer "PrintCraft Elite"	3500.00	electronic
Giovanna	223 Hansons Trail	769-531-2915	Waterproof Backpack "AquaPack Adventure"	150.00	electronic
Giovanna	223 Hansons Trail	769-531-2915	Interactive Drawing Board "ArtCanvas Pro"	400.00	electronic
Godfry	4623 Logan Crossing	195-668-5662	Drone "SkyExplorer Pro"	12002.00	electronic
Golda	5281 Blue Bill Park Court	545-937-7930	Electric Skateboard "E-Skate Blaze"	700.00	electronic
Hartley	4782 Arizona Drive	264-637-0651	Electric Bike "E-Bike Thunder"	420.00	electronic
Imogene	48 Cardinal Court	718-452-8548	Smart Glass Table "TechTable SmartGlass"	15.00	electronic
Jedd	5219 Melvin Terrace	765-956-0447	High-Precision GPS Tracker "NavTech Precision"	150.00	electronic
Kalli	14989 West Drive	491-635-4452	Virtual Assistant "AI Helper Pro"	500.00	electronic
Kyle	2 Lakewood Gardens Center	116-481-6936	Smart Watch "TechTime Master"	400.00	electronic
Kyle	2 Lakewood Gardens Center	116-481-6936	Electric Scooter "SpeedyScoot X1"	800.00	electronic
Mano	7902 Cardinal Terrace	731-279-4697	Professional Microphone "AudioMaster Pro"	400.00	electronic
Mano	7902 Cardinal Terrace	731-279-4697	High-Definition Action Camera "ActionPro 4K"	300.00	electronic
Mano	7902 Cardinal Terrace	731-279-4697	Laser Engraver "LaserCraft 5000"	600.00	electronic
Marji	63992 Maryland Pass	676-113-7462	Electric Toothbrush "DentaCare Smart"	80.00	electronic
Maura	882 Huxley Drive	263-271-5666	Home Theater Projector "CineMagic HD"	99.00	electronic
Maura	882 Huxley Drive	263-271-5666	Designer Laptop "DesignCraft Studio"	2899.00	electronic
Noak	7914 Maryland Alley	750-476-7998	Smart Refrigerator "SmartCool Pro"	2.00	electronic
Noak	7914 Maryland Alley	750-476-7998	Smartphone "TechGuru X20"	1500.00	electronic
Sloan	60 Claremont Lane	353-112-2952	Smart Thermostat "EcoHeat Smart"	250.00	electronic
Sloan	60 Claremont Lane	353-112-2952	Supercomputer "Quantum Beast"	10000000.00	electronic
Susanetta	145 Haas Circle	443-920-0485	Professional Camera "Pixel Master 9000"	5500.00	electronic
Vaughn	7317 Sachs Circle	557-300-1851	Kitchen Assistant Robot "ChefBot Deluxe"	700.00	electronic
Vivyanne	7170 Amoth Terrace	894-843-9338	Satellite Lamp "CosmicLight Satellite"	120.00	electronic

(30 rows)

Рисунок 10.16 – Провайдери з товарами у певній категорії

10.17. Запит на виведення всіх ревізій для конкретного провайдера.

Запит дуже корисний у випадках, коли нам треба подивитись, що про конкретного провайдера думають клієнти, котрі з ним співпрацювали. Результат виводу на рисунку 10.17.

SELECT

c.name AS client_name,
c.surname AS client_surname,
r.rating,
r.comment

FROM client c

JOIN review r ON c.client_id = r.client_id

WHERE r.provider_id = (SELECT provider_id FROM provider WHERE name = 'Sloan');

```
trading_platform=# SELECT
trading_platform=#   c.name AS client_name,
trading_platform=#   c.surname AS client_surname,
trading_platform=#   r.rating,
trading_platform=#   r.comment
trading_platform=# FROM client c
trading_platform=# JOIN review r ON c.client_id = r.client_id
trading_platform=# WHERE r.provider_id = (SELECT provider_id FROM provider WHERE name = 'Sloan');
 client_name | client_surname | rating |
-----+-----+-----+
Gaultiero   | Pratley        | 95     |
Gerick       | Kiln           | 99     |
Tandi       | Frood          | 95     |
Livy         | Merring        | 97     |
Danit        | Dominy         | 99     |
Annaliese   | Dodworth       | 100    |
Valma       | Hehl           | 87     |
Margaux     | Starten        | 95     |
Rab         | Fearn          | 93     |
Yuma        | Seyfart        | 89     |
(10 rows)
```

Рисунок 10.17 – Відгуки про певного постачальника

10.18. Запит на виведення к-сті оплат кожним способ та найціннішого клієнта з його оплатою.

Запит підраховує, скільки оплат здійснили клієнти кожним із доступних способів оплати(всього їх 18). Також буде виведено найціннішого клієнта для кожного методу оплати. Результат виводу результату на рисунку 10.18.

```
WITH PaymentInfo AS (
```

```
    SELECT
```

```
        p.payment_method,
```

```
        MAX(o.total_cost) AS max_payment,
```

```
        COUNT(o.order_id) AS usage_count
```

```
    FROM payment p
```

```
    JOIN "order" o ON p.payment_id = o.payment_id
```

```
    GROUP BY p.payment_method
```

```
)
```

```
SELECT
```

```
    pi.payment_method,
```

```
    c.name AS most_valuable_client,
```

```
    pi.max_payment,
```

```
    pi.usage_count
```

```
FROM PaymentInfo pi
```

```
JOIN client c ON c.client_id IN (
```

```
    SELECT
```

```
        o.client_id
```

```
    FROM "order" o
```

```
    JOIN payment p ON o.payment_id = p.payment_id AND o.total_cost =  
    pi.max_payment
```

```
)
```

```
GROUP BY pi.payment_method, c.name, pi.max_payment, pi.usage_count;
```

```

trading_platform=# WITH PaymentInfo AS (
trading_platform=# SELECT
trading_platform=#     p.payment_method,
trading_platform=#     MAX(o.total_cost) AS max_payment,
trading_platform=#     COUNT(o.order_id) AS usage_count
trading_platform=# FROM payment p
trading_platform=# JOIN "order" o ON p.payment_id = o.payment_id
trading_platform=# GROUP BY p.payment_method
trading_platform=# )
trading_platform=# SELECT
trading_platform=#     pi.payment_method,
trading_platform=#     c.name AS most_valuable_client,
trading_platform=#     pi.max_payment,
trading_platform=#     pi.usage_count
trading_platform=# FROM PaymentInfo pi
trading_platform=# JOIN client c ON c.client_id IN (
trading_platform=#     SELECT
trading_platform=#         o.client_id
trading_platform=#     FROM "order" o
trading_platform=#     JOIN payment p ON o.payment_id = p.payment_id AND o.total_cost = pi.max_payment
trading_platform=# )
trading_platform=# GROUP BY pi.payment_method, c.name, pi.max_payment, pi.usage_count;

```

payment_method	most_valuable_client	max_payment	usage_count
Apple Pay	Cyndie	989743.00	36
Bank Transfer	Kale	973056.00	36
Bill Payment	Lauritz	975745.00	36
Cash	Glen	998664.00	142
Check	Zia	989154.00	36
Credit Card	Rutger	994272.00	142
Cryptocurrency	Alick	950113.00	36
Debit Card	Vonny	926573.00	36
E-money	Dorene	959156.00	36
Google Pay	Emlynne	995101.00	71
Mobile Payment	Aubrette	986597.00	71
Non-cash Payment	David	994794.00	36
Online Payment	Tadeas	999845.00	71
Partial Payment	Nichol	977706.00	36
Payment on Delivery (COD)	Antonin	957893.00	36
PayPal	Kermie	956384.00	71
Subscription Payment	Kurtis	907728.00	36
Wire Transfer	Jarret	866156.00	36

(18 rows)

Рисунок 10.18 – Інформація про методи оплати

10.19. Запит на виведення інформації про певний спосіб оплати.

Запит виводить потрібну інформацію про всі транзакції конкретним методом оплати на вибір. Результат запиту на рисунку 10.19.

```
SELECT
    p.payment_id,
    p.payment_method,
    o.total_cost AS cost,
    c.name AS client_name
FROM payment p
JOIN "order" o ON p.payment_id = o.payment_id
JOIN client c ON o.client_id = c.client_id
WHERE p.payment_method = 'Cash';
```

```
trading_platform=# SELECT
trading_platform=#   p.payment_id,
trading_platform=#   p.payment_method,
trading_platform=#   o.total_cost AS cost,
trading_platform=#   c.name AS client_name
trading_platform=# FROM payment p
trading_platform=# JOIN "order" o ON p.payment_id = o.payment_id
trading_platform=# JOIN client c ON o.client_id = c.client_id
trading_platform=# WHERE p.payment_method = 'Cash';
 payment_id | payment_method |   cost   | client_name
-----+-----+-----+-----
          1 | Cash           | 136665.00 | Charlot
         19 | Cash           | 618293.00 | Ginni
         25 | Cash           | 803931.00 | Giles
         27 | Cash           | 300694.00 | Joyce
         29 | Cash           | 459151.00 | Virginia
         47 | Cash           | 127604.00 | Vidovic
         53 | Cash           | 264921.00 | Renae
         55 | Cash           | 361423.00 | Gregory
         57 | Cash           | 748932.00 | Sky
         75 | Cash           | 610879.00 | Bronnie
         81 | Cash           | 122774.00 | Kip
         83 | Cash           | 612051.00 | Fred
         85 | Cash           | 224037.00 | Ruthe
        103 | Cash           | 710025.00 | Seumas
        109 | Cash           | 111494.00 | Faber
        111 | Cash           | 434220.00 | Torrance
        113 | Cash           | 125540.00 | Puff
        131 | Cash           | 335495.00 | Jamill
        137 | Cash           | 841047.00 | Aharon
```

Рисунок 10.19 – Інформація про транзакції певним способом

10.20. Запит на виведення лише доступних на даний момент послуг.

Запит виводить всі послуги, які на даний момент часу є доступними(availability = 'available'). Приклад використання запиту показано на рисунку 10.23.

SELECT

s.service_id,

s.name AS service_name,

s.description,

s.price,

p.name AS provider_name

FROM service s

JOIN provider p ON s.provider_id = p.provider_id

WHERE s.availability = 'available'

ORDER BY provider_name;

```
trading_platform=# SELECT
trading_platform=#   s.service_id,
trading_platform=#   s.name AS service_name,
trading_platform=#   s.description,
trading_platform=#   s.price,
trading_platform=#   p.name AS provider_name
trading_platform=# FROM service s
trading_platform=# JOIN provider p ON s.provider_id = p.provider_id
trading_platform=# WHERE s.availability = 'available'
trading_platform=# ORDER BY provider_name;
```

service_id	service_name	description	price	provider_name
60	Home Trial Experience	Service allowing customers to try products at home before finalizing the purchase.	494.00	Celene
76	Exclusive Event Invitations	Receive invitations to product launches and exclusive brand events.	322.00	Celene
38	Extended Warranty Guard	Additional warranty service that extends the standard coverage period for your purchased goods.	112.00	Cinda
29	Family & Friends Referral Rewards	Earn rewards and discounts by referring our products to your loved ones.	406.00	Cinda
45	Early Bird Delivery	Get your products delivered first thing in the morning with our early delivery option.	219.00	Cinda
75	Seasonal Product Refresh	Regular updates and refreshes for products to keep them in line with the latest trends.	300.00	Dayna
98	Express Shipping Plus	Premium fast shipping service ensuring expedited delivery of your products.	130.00	Devondra
91	Annual Maintenance Checkup	Regular maintenance service for products to ensure longevity and optimal performance.	172.00	Devondra
28	One-to-One Training Sessions	Personalized training and tutorials for effective and optimal use of your products.	220.00	Devondra
11	Instant Refund Processing	Accelerated refund service for quick and hassle-free returns.	305.00	Dianemarie
70	Interactive Product Tutorial	Access to exclusive online tutorials to maximize your product's potential.	14.00	Dicky
90	Home Trial Experience	Service allowing customers to try products at home before finalizing the purchase.	133.00	Durand
50	Trade-In Upgrade Program	Exchange your old models for discounts on the latest versions of our products.	9.00	Durand
95	Gift Wrapping Boutique	Make your purchase special with our bespoke gift wrapping service.	359.00	Durand
37	Express Shipping Plus	Premium fast shipping service ensuring expedited delivery of your products.	400.00	Estelle
34	Custom Product Tailoring	Personalization service for products, allowing customization to fit individual preferences.	471.00	Estelle
35	Priority Restock Alert	An exclusive notification service for clients about the restocking of high-demand products.	334.00	Estelle
57	Priority Restock Alert	An exclusive notification service for clients about the restocking of high-demand products.	312.00	Germana
63	Instant Refund Processing	Accelerated refund service for quick and hassle-free returns.	48.00	Germana
51	Personal Shopper Assistant	Expert advice and personalized shopping assistance to help you choose the perfect product.	152.00	Germana
40	Custom Product Tailoring	Personalization service for products, allowing customization to fit individual preferences.	64.00	Giovanna
5	Priority Restock Alert	An exclusive notification service for clients about the restocking of high-demand products.	277.00	Giovanna
13	Exclusive Members Club	Join our members-only club for special deals, offers, and product previews.	145.00	Giovanna

Рисунок 10.20 – Доступні послуги

11. РЕАЛІЗАЦІЯ ФУНКЦІЙ ТА ПРОЦЕДУР

Збережені функції та процедури - це програмні об'єкти, які зберігають і викликають певний набір SQL-операцій у базі даних. Ці об'єкти можуть приймати параметри, обробляти параметри та виконувати інші дії на вимогу розробника. Функції і процедури можна використовувати, визначаючи і використовуючи локальні змінні. Крім того, функції можуть повертати значення, рядки і таблиці за необхідності.

Нижче наведено реалізовані функції, їхній словесний опис та суть, а також приклади використання.

11.1. Функція для виведення відгуків до заданого провайдера.

Функція `Provider_feedback` використовується для отримання списку відгуків про конкретного провайдера. Коли ви викликаєте цю функцію, передаючи їй ID провайдера, вона виконує запит до таблиці `review`, фільтруючи відгуки за цим ID. Відгуки, які відносяться до цього провайдера, будуть виведені, включаючи інформацію про ID відгуку, рейтинг, текст коментаря, дату відгуку, ID клієнта, який залишив відгук, та ID самого провайдера. Сортування відгуків за спаданням рейтингу допомагає швидко знайти найвище оцінені відгуки. Для виклику функції слід використати запит

```
CREATE OR REPLACE FUNCTION Provider_feedback(given_provider_id  
BIGINT)
```

```
RETURNS TABLE (
```

```
    review_id BIGINT,
```

```
    rating SMALLINT,
```

```
    comment TEXT,
```

```
    review_date DATE,
```

```
    client_id BIGINT,
```

```
    provider_id BIGINT )
```



```

AS $$
BEGIN
    RETURN QUERY
    SELECT *
    FROM review
    WHERE review.provider_id = given_provider_id
    ORDER BY rating DESC;
END;
$$ LANGUAGE PLPGSQL;

```

```
trading_platform=# SELECT * FROM Provider_feedback(1);
```

review_id	rating	comment	review_date	client_id	provider_id
4	100	Exceptional service! The quality of the products and the speed of delivery exceeded my expectations.	2022-12-11	502	1
13	99		2022-01-06	268	1
91	99	Impeccable service! I couldn't be happier with the efficiency and product quality.	2023-06-14	409	1
162	97		2022-09-03	351	1
138	95		2022-07-18	195	1
52	95		2023-04-27	291	1
143	95	Outstanding quality and service. I am grateful for the positive impact on my business.	2022-06-03	937	1
128	93	Impressed with the level of customer care. Quick response and problem resolution.	2023-12-24	943	1
63	89	The product exceeded my expectations. I appreciate the commitment to excellence.	2023-10-19	970	1
85	87	Highly professional team. They make the entire purchasing process seamless and enjoyable.	2023-07-05	579	1

(10 rows)

Рисунок 11.1 – Отримані всі відгуки постачальника

11.2. Функція для обрахунку загальної вартості контрактів для заданого провайдера.

Функція `Total_contracts_cost` використовується для визначення сумарної вартості контрактів, які провайдер уклав. Після виклику функції з ID провайдера, вона звертається до таблиці `contract`, обчислює загальну суму вартості контрактів цього провайдера і повертає це значення. Якщо провайдер не має жодного контракту, функція поверне 0, завдяки використанню `COALESCE`. Нижче наведено код процедури, а на рисунках 11.2-11.3 – результат виклику цієї функції.

```
CREATE OR REPLACE FUNCTION Total_contracts_cost(given_provider_id
BIGINT)
RETURNS NUMERIC
AS $$
DECLARE
    total_cost NUMERIC;
BEGIN
    SELECT SUM(contract.total_cost) INTO total_cost
    FROM contract
    WHERE contract.provider_id = given_provider_id;

    RETURN COALESCE(total_cost, 0);
END;
$$ LANGUAGE PLPGSQL;
```

```
trading_platform=# SELECT Total_contracts_cost(1);
total_contracts_cost
-----
          9565269.00
(1 row)
```

Рисунок 11.2 – Результат обрахунку провайдера із контрактами

```
trading_platform=# SELECT Total_contracts_cost(51);
total_contracts_cost
-----
              0
(1 row)
```

Рисунок 11.3 – Результат обрахунку провайдера без контрактів

11.3. Функція для порівняння двох заданих типів оплат.

Функція `Compare_payment_methods` використовується для аналізу та порівняння популярності двох різних методів оплати. Після виклику функції з двома методами оплати в якості параметрів, вона здійснює запит до таблиці `payment`, відфільтровує оплати, що відповідають цим методам, та підраховує кількість транзакцій для кожного з них. Результатом є таблиця, яка показує кількість транзакцій, здійснених кожним із цих методів оплати. Код функції наведено нижче, приклад використання – на рисунку 11.4.

```
CREATE OR REPLACE FUNCTION Compare_payment_methods(method1
VARCHAR(50), method2 VARCHAR(50))
```

```
RETURNS TABLE (
```

```
    method VARCHAR(50),
```

```
    payment_count BIGINT
```

```
)
```

```
AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT payment_method AS method,
```

```
           COUNT(*) AS payment_count
```

```
    FROM payment
```

```
    WHERE payment_method IN (method1, method2)
```

```
    GROUP BY payment_method;
```

```
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
trading_platform=# SELECT * FROM Compare_payment_methods('PayPal', 'Cash');
 method | payment_count
-----+-----
 PayPal |             71
  Cash  |             142
(2 rows)
```

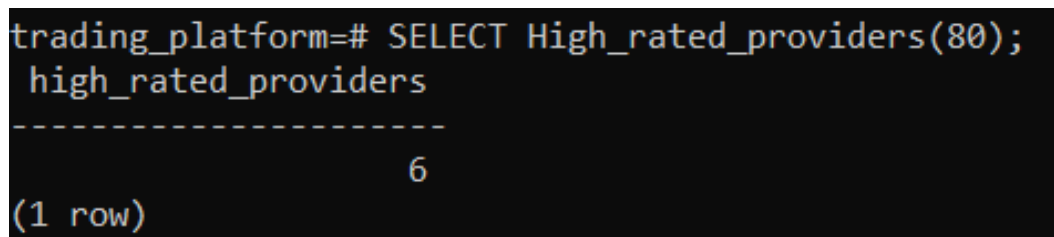
Рисунок 11.4 – Порівняння методів оплати готівкою та PayPal

11.4. Функція для визначення кількості провайдерів з рейтингом більше заданного.

Функція `High_rated_providers` використовується для визначення кількості провайдерів у базі даних, чий рейтинг вищий за вказане мінімальне значення. Це може бути корисним для швидкого аналізу якості провайдерів, наявних у системі. Код функції наведено нижче, результат виклику – на рисунку 11.5.

```
CREATE OR REPLACE FUNCTION High_rated_providers(min_rating
INTEGER)
RETURNS INTEGER
AS $$
DECLARE
    driver_count INTEGER;
BEGIN
    SELECT
        COUNT(*) INTO driver_count
    FROM provider
    WHERE rating > min_rating;

    RETURN driver_count;
END;
$$ LANGUAGE PLPGSQL;
```



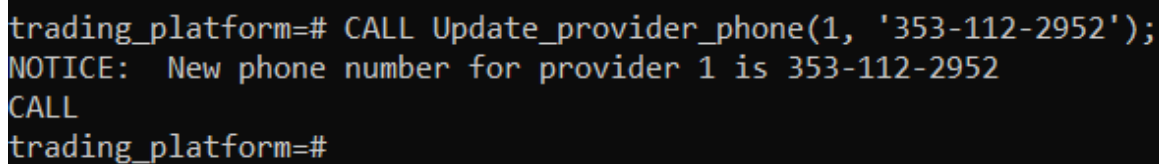
```
trading_platform=# SELECT High_rated_providers(80);
 high_rated_providers
-----
                    6
(1 row)
```

Рисунок 11.5 – Кількість провайдерів із рейтингом більше 80

11.5. Процедура для зміни номеру телефону провайдера.

Ця процедура використовується для оновлення контактної інформації провайдерів у вашій системі. Коли ми викликаємо цю процедуру, передаючи її ID провайдера і новий номер телефону, вона оновлює відповідний запис у таблиці Provider. Після оновлення, вона виводить повідомлення з підтвердженням, що номер телефону було змінено. Нижче наведено код процедури, а на рисунку 11.6 показано результат її виклику.

```
CREATE OR REPLACE PROCEDURE Update_provider_phone(p_provider_id
INTEGER, p_new_phone_number VARCHAR(20))
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE Provider
    SET telephone = p_new_phone_number
    WHERE provider_id = p_provider_id;
    RAISE NOTICE 'New phone number for provider % is %', p_provider_id,
p_new_phone_number;
END;
$$;
```



```
trading_platform=# CALL Update_provider_phone(1, '353-112-2952');
NOTICE:  New phone number for provider 1 is 353-112-2952
CALL
trading_platform=#
```

Рисунок 11.6 – Зміна номеру телефону провайдера

11.6. Процедура для видалення певної послуги.

Ця процедура є безпечним способом видалення послуг з бази даних. Вона забезпечує, що видалення відбувається лише у випадку, коли послуга існує, тим самим уникнувши помилок видалення невідомих або неіснуючих записів.. Нижче наведено код функції, а на рисунку 11.7 та 11.8 показано результат виклику цієї процедури.

```
CREATE OR REPLACE PROCEDURE Delete_service(input_service_id
BIGINT)
```

```
AS $$
```

```
BEGIN
```

```
    IF NOT EXISTS (SELECT 1 FROM service WHERE service_id =
input_service_id) THEN
```

```
        RAISE EXCEPTION 'Service with specified service_id does not exist.';
```

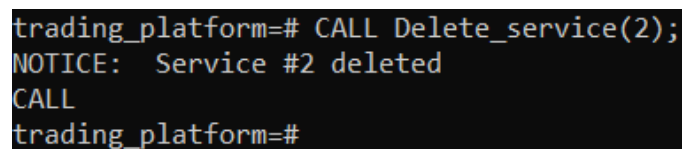
```
    END IF;
```

```
    DELETE FROM service WHERE service_id = input_service_id;
```

```
    RAISE NOTICE 'Service #%% deleted', input_service_id;
```

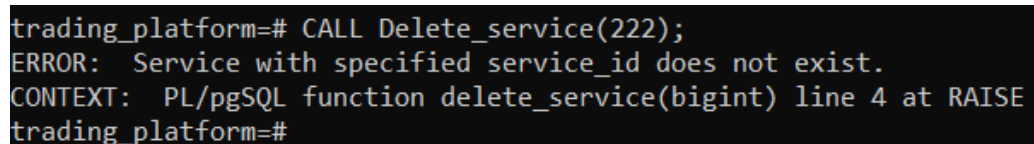
```
END;
```

```
$$ LANGUAGE PLPGSQL;
```



```
trading_platform=# CALL Delete_service(2);
NOTICE: Service #2 deleted
CALL
```

Рисунок 11.7 – Видалення існуючої послуги



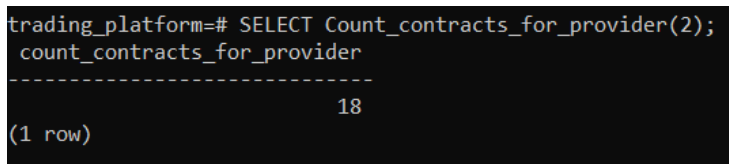
```
trading_platform=# CALL Delete_service(222);
ERROR: Service with specified service_id does not exist.
CONTEXT: PL/pgSQL function delete_service(bigint) line 4 at RAISE
```

Рисунок 11.8 – Видалення неіснуючої послуги

11.7. Функція для підрахунку кількості контрактів для заданого провайдера.

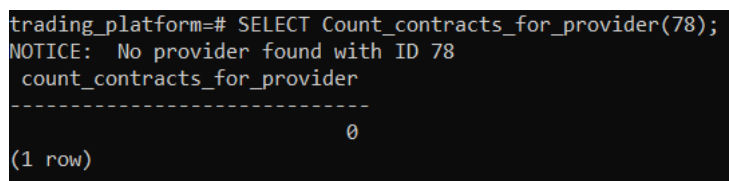
Функція `Count_contracts_for_provider` використовується для визначення загальної кількості контрактів, укладених певним провайдером. Це може бути корисно для оцінки активності та популярності окремого провайдера в рамках бізнесу. Код процедури наведено нижче, на рисунках 11.9 та 11.10 показано результати виклику цієї процедури.

```
CREATE OR REPLACE FUNCTION
Count_contracts_for_provider(given_provider_id BIGINT)
RETURNS INTEGER
AS $$
DECLARE
    contract_count INTEGER;
BEGIN
    SELECT COUNT(*) INTO contract_count
    FROM contract
    WHERE provider_id = given_provider_id;
    RETURN contract_count;
END;
$$ LANGUAGE PLPGSQL;
```



```
trading_platform=# SELECT Count_contracts_for_provider(2);
count_contracts_for_provider
-----
18
(1 row)
```

Рисунок 11.9 – Підрахунок к-сті контрактів для існуючого постачальника



```
trading_platform=# SELECT Count_contracts_for_provider(78);
NOTICE: No provider found with ID 78
count_contracts_for_provider
-----
0
(1 row)
```

Рисунок 11.10 – Підрахунок к-сті контрактів для неіснуючого постачальника

11.8. Функція для виведення деталей обраного тендеру.

Функція `Tender_details_and_bids` надає детальний огляд конкретного тендеру, включаючи інформацію про початкову та поточну ставку, дати початку та завершення тендеру, а також інформацію про к-сть ставок, зроблених на цей тендер. Це корисно для аналізу активності учасників тендеру та визначення переможця. Код процедури наведено нижче, на рисунку 11.11 показано результати виклику цієї процедури.

```
CREATE OR REPLACE FUNCTION Tender_details_and_bids(given_tender_id
BIGINT)
```

```
RETURNS TABLE (
```

```
    tender_id BIGINT,
```

```
    start_date DATE,
```

```
    end_date DATE,
```

```
    starting_stake INT,
```

```
    end_stake INT,
```

```
    winner_provider_name VARCHAR(60),
```

```
    amount_of_stakes INT
```

```
)
```

```
AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT t.tender_id, t.start_date, t.end_date, t.starting_bid, t.current_bid,
p.name AS provider_name, b.amount
```

```
    FROM tender t
```

```
    LEFT JOIN bid b ON t.tender_id = b.tender_id
```

```
    LEFT JOIN provider p ON b.provider_id = p.provider_id
```

```
    WHERE t.tender_id = given_tender_id;
```

```
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
trading_platform=# SELECT * FROM Tender_details_and_bids(1);
 tender_id | start_date | end_date | starting_stake | end_stake | winner_provider_name | amount_of_stakes
-----+-----+-----+-----+-----+-----+-----
          1 | 2023-12-29 | 2023-12-30 |          4734 |        41196 | Giovanni              |              16
(1 row)
```

Рисунок 11.13 – Виведення інформації про обраний тендер

11.9. Функція для обрахування кількості провайдерів у заданому регіоні.

Функція `Providers_in_region` призначена для отримання списку провайдерів, які працюють у певному регіоні, та їхніх рейтингів. Під час виконання функції також підраховується загальна кількість провайдерів у заданому регіоні та виводиться для інформаційних цілей. Код процедури наведено нижче, результат виклику – на рисунку 11.14.

```
CREATE OR REPLACE FUNCTION Providers_in_region(region TEXT)
```

```
RETURNS TABLE (
```

```
    provider_id BIGINT,
```

```
    name VARCHAR(30),
```

```
    surname VARCHAR(30),
```

```
    rating SMALLINT )
```

```
AS $$
```

```
DECLARE
```

```
    provider_count INT;
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT provider.provider_id, provider.name, provider.surname,  
    provider.rating
```

```
    FROM provider
```

```
    WHERE provider.service_area = region
```

```
    ORDER BY provider.rating DESC;
```

```
    SELECT COUNT(*) INTO provider_count
```

```
    FROM provider
```

```
    WHERE provider.service_area = region;
```

```
    RAISE NOTICE 'Total providers in region %: %', region, provider_count;
```

```
END;
```

```
$$ LANGUAGE PLPGSQL;
```

```
trading_platform=# SELECT * FROM Providers_in_region('Asia');
NOTICE: Total providers in region Asia: 9
 provider_id | name | surname | rating
-----+-----+-----+-----
          1 | Sloan | Lindsay |    95
         49 | Sandor | Wilber |    83
         25 | Germana | Kosiada |    73
         43 | Norine | Seilmann |    58
         19 | Marji | Lickorish |    56
         37 | Godfry | Shingfield |    48
         13 | Kristoforo | Starsmeare |    43
          7 | Junia | Stitcher |    29
         31 | Cathrine | Dey |    13
(9 rows)
```

Рисунок 11.14 – Провайдери у заданому регіоні

11.10. Функція для обрахування потенційної виручки від певного продукту на складі.

Функція CalculateInventoryValueForProduct призначена для обчислення вартості інвентарю для певного продукту та надає інформацію про провайдера(ів), який цим продуктом володіє(ють). В результаті виклику функції отримуємо таблицю з інформацією про провайдерів, їхні контактні дані та загальну вартість інвентарю для цього продукту у кожного провайдера. Код процедури наведено нижче, результат роботи – на рисунку 11.15.

```
CREATE OR REPLACE FUNCTION
```

```
CalculateInventoryValueForProduct(product_name VARCHAR(50))
```

```
RETURNS TABLE (
```

```
    owner_provider_name VARCHAR(60),
```

```
    provider_telephone VARCHAR(20),
```

```
    provider_region TEXT,
```

```
    total_product_value NUMERIC
```

```
)
```

```
AS $$
```

```
BEGIN
```

```
    RETURN QUERY
```

```
    SELECT
```

```
        CAST(pr.name || ' ' || pr.surname AS VARCHAR(60)) AS  
owner_provider_name,
```

```
        pr.telephone AS provider_telephone,
```

```
        pr.service_area AS provider_region,
```

```

SUM(p.price * p.stock_quantity) AS total_product_value

FROM provider pr

JOIN product p ON pr.provider_id = p.provider_id

WHERE p.name = product_name

GROUP BY pr.provider_id, pr.name, pr.surname, pr.telephone, pr.service_area;

END;

$$ LANGUAGE PLPGSQL;

```

```

trading_platform=# SELECT * FROM CalculateInventoryValueForProduct('Drone "SkyExplorer Pro"');
 owner_provider_name | provider_telephone | provider_region | total_product_value
-----+-----+-----+-----
 Godfry Shingfield   | 195-668-5662       | Asia            | 684114.00
(1 row)

```

Рисунок 11.17 – Потенційна виручка від певного продукту

12. ОПТИМІЗАЦІЯ

Процес підвищення продуктивності та ефективності бази даних є ключовим елементом реалізації. Це особливо важливо в сучасних реаліях, коли великі обсяги даних і велика кількість одночасних запитів можуть призвести до затримок у реакції системи. Оптимізація спрямована на скорочення часу виконання запитів, використання ресурсів і підвищення ефективності.

Одним із способів оптимізації продуктивності бази даних є використання індексів. Нижче наведено код для створення індексу в базі даних `trading_platform`.

Що ще важливіше, індекси не завжди підвищують продуктивність, а іноді навіть знижують її. Це може статися, якщо індекси встановлені для полів з великою кількістю значень, що повторюються, або полів, які часто оновлюються. Занадто велика кількість індексів також може негативно вплинути на оптимізацію.

```
CREATE INDEX idx_client_telephone ON client(telephone);  
CREATE INDEX idx_client_email ON client(email);  
CREATE INDEX idx_client_name ON client(name);
```

```
CREATE INDEX idx_provider_telephone ON provider(telephone);  
CREATE INDEX idx_provider_email ON provider(email);  
CREATE INDEX idx_provider_name ON provider(name);
```

```
CREATE INDEX idx_contract_start_date ON contract(start_date);  
CREATE INDEX idx_contract_end_date ON contract(end_date);
```

```
CREATE INDEX idx_review_comment ON review(comment);
```

```
CREATE INDEX idx_payment_payment_method ON  
payment(payment_method);
```

Вплив створених індексів слід протестувати. Нижче будуть наведені 3 різнотипні запити. Завдяки ним можна протестувати середню зміну у швидкості роботи бази даних. Для підрахунку часу виконання використаємо «EXPLAIN ANALYZE» перед запитом.

12.1. Аналіз впливу індексів на першому запиті.

Код першого запиту наведено нижче. В таблиці 12.1 наведено час виконання запиту для 5 випробувань до і після додання індексів. На рисунку 12.1 та 12.2 показано схеми запитів та час перших випробувань.

EXPLAIN ANALYZE

SELECT

c.name AS client_name,
c.surname AS client_surname,
c.address AS client_address,
c.telephone AS client_telephone,
c.email AS client_email,
contract_data.contract_id,
contract_data.start_date,
contract_data.end_date,
contract_data.total_cost

FROM client c

JOIN (

SELECT

co.contract_id,
co.start_date,
co.end_date,
co.total_cost,
co.client_id

FROM contract co

) AS contract_data ON c.client_id = contract_data.client_id;

Таблиця 12.1 – Аналіз впливу індексів на першому запиті

Номер спроби	1	2	3	4	5
Час виконання до додання індексу, мс	0.716	0.632	0.576	0.643	0.704
Час виконання після додання індексу, мс	0.566	0.566	0.560	0.620	0.564

Середнє значення перед доданням індексів = 0.654 мс.

Середнє значення після додання індексів = 0.575 мс.

Отже, з індексами час, витрачений на запит, зменшився на 12%.

```

trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#      c.name AS client_name,
trading_platform=#      c.surname AS client_surname,
trading_platform=#      c.address AS client_address,
trading_platform=#      c.telephone AS client_telephone,
trading_platform=#      c.email AS client_email,
trading_platform=#      contract_data.contract_id,
trading_platform=#      contract_data.start_date,
trading_platform=#      contract_data.end_date,
trading_platform=#      contract_data.total_cost
trading_platform=# FROM client c
trading_platform=# JOIN (
trading_platform=#      SELECT
trading_platform=#          co.contract_id,
trading_platform=#          co.start_date,
trading_platform=#          co.end_date,
trading_platform=#          co.total_cost,
trading_platform=#          co.client_id
trading_platform=#      FROM contract co
trading_platform=# ) AS contract_data ON c.client_id = contract_data.client_id;
                                QUERY PLAN
-----
Hash Join  (cost=35.50..57.14 rows=1000 width=89) (actual time=0.334..0.595 rows=1000 loops=1)
  Hash Cond: (co.client_id = c.client_id)
    -> Seq Scan on contract co  (cost=0.00..19.00 rows=1000 width=31) (actual time=0.012..0.072 rows=1000 loops=1)
    -> Hash  (cost=23.00..23.00 rows=1000 width=74) (actual time=0.304..0.305 rows=1000 loops=1)
          Buckets: 1024  Batches: 1  Memory Usage: 114kB
          -> Seq Scan on client c  (cost=0.00..23.00 rows=1000 width=74) (actual time=0.007..0.122 rows=1000 loops=1)
Planning Time: 0.661 ms
Execution Time: 0.716 ms
(8 rows)

```

Рисунок 12.1 – Схема запиту 1 до додання індексів

```

trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#      c.name AS client_name,
trading_platform=#      c.surname AS client_surname,
trading_platform=#      c.address AS client_address,
trading_platform=#      c.telephone AS client_telephone,
trading_platform=#      c.email AS client_email,
trading_platform=#      contract_data.contract_id,
trading_platform=#      contract_data.start_date,
trading_platform=#      contract_data.end_date,
trading_platform=#      contract_data.total_cost
trading_platform=# FROM client c
trading_platform=# JOIN (
trading_platform=#      SELECT
trading_platform=#          co.contract_id,
trading_platform=#          co.start_date,
trading_platform=#          co.end_date,
trading_platform=#          co.total_cost,
trading_platform=#          co.client_id
trading_platform=#      FROM contract co
trading_platform=# ) AS contract_data ON c.client_id = contract_data.client_id;
                                QUERY PLAN
-----
Hash Join  (cost=35.50..57.14 rows=1000 width=89) (actual time=0.258..0.518 rows=1000 loops=1)
  Hash Cond: (co.client_id = c.client_id)
    -> Seq Scan on contract co  (cost=0.00..19.00 rows=1000 width=31) (actual time=0.009..0.068 rows=1000 loops=1)
    -> Hash  (cost=23.00..23.00 rows=1000 width=74) (actual time=0.243..0.243 rows=1000 loops=1)
          Buckets: 1024  Batches: 1  Memory Usage: 114kB
          -> Seq Scan on client c  (cost=0.00..23.00 rows=1000 width=74) (actual time=0.005..0.114 rows=1000 loops=1)
Planning Time: 2.427 ms
Execution Time: 0.566 ms
(8 rows)

```

Рисунок 12.2 – Схема запиту 1 після додання індексів

12.2. Аналіз впливу на другому запиті.

Код другого запиту наведено нижче. В таблиці 12.2 наведено час виконання запиту для 5 випробувань до і після додання індексів. На рисунку 12.3 та 12.4 показано схеми запитів та час перших випробувань.

EXPLAIN ANALYZE

SELECT

r.review_id, r.comment,
c.name AS client_name,
p.name AS provider_name

FROM review r

RIGHT JOIN client c ON r.client_id = c.client_id

RIGHT JOIN provider p ON r.provider_id = p.provider_id

WHERE r.comment IS NOT NULL;

Таблиця 12.2 – Аналіз впливу на другому запиті

Номер спроби	1	2	3	4	5
Час виконання до додання індексу, мс	0.358	0.298	0.302	0.269	0.301
Час виконання після додання індексу, мс	0.243	0.305	0.285	0.250	0.270

Середнє значення перед доданням індексів = 0.306 мс.

Середнє значення після додання індексів = 0.271 мс.

Отже, з індексами час, витрачений на запит, зменшився на 11.5%.

```

trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#     r.review_id, r.comment,
trading_platform=#     c.name AS client_name,
trading_platform=#     p.name AS provider_name
trading_platform=# FROM review r
trading_platform=# RIGHT JOIN client c ON r.client_id = c.client_id
trading_platform=# RIGHT JOIN provider p ON r.provider_id = p.provider_id
trading_platform=# WHERE r.comment IS NOT NULL;

QUERY PLAN
-----
Hash Join (cost=10.25..40.66 rows=90 width=107) (actual time=0.104..0.259 rows=90 loops=1)
  Hash Cond: (r.provider_id = p.provider_id)
    -> Hash Join (cost=8.12..38.27 rows=90 width=108) (actual time=0.060..0.202 rows=90 loops=1)
      Hash Cond: (c.client_id = r.client_id)
      -> Seq Scan on client c (cost=0.00..23.00 rows=1000 width=14) (actual time=0.005..0.065 rows=1000 loops=1)
      -> Hash (cost=7.00..7.00 rows=90 width=110) (actual time=0.043..0.043 rows=90 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 21kB
        -> Seq Scan on review r (cost=0.00..7.00 rows=90 width=110) (actual time=0.008..0.034 rows=90 loops=1)
          Filter: (comment IS NOT NULL)
          Rows Removed by Filter: 210
    -> Hash (cost=1.50..1.50 rows=50 width=15) (actual time=0.027..0.027 rows=50 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 11kB
      -> Seq Scan on provider p (cost=0.00..1.50 rows=50 width=15) (actual time=0.015..0.020 rows=50 loops=1)
Planning Time: 0.286 ms
Execution Time: 0.358 ms
(15 rows)

```

Рисунок 12.3 – Схема запиту 2 до додання індексів

```

trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#     r.review_id, r.comment,
trading_platform=#     c.name AS client_name,
trading_platform=#     p.name AS provider_name
trading_platform=# FROM review r
trading_platform=# RIGHT JOIN client c ON r.client_id = c.client_id
trading_platform=# RIGHT JOIN provider p ON r.provider_id = p.provider_id
trading_platform=# WHERE r.comment IS NOT NULL;

QUERY PLAN
-----
Hash Join (cost=10.25..40.66 rows=90 width=107) (actual time=0.070..0.222 rows=90 loops=1)
  Hash Cond: (r.provider_id = p.provider_id)
    -> Hash Join (cost=8.12..38.27 rows=90 width=108) (actual time=0.048..0.186 rows=90 loops=1)
      Hash Cond: (c.client_id = r.client_id)
      -> Seq Scan on client c (cost=0.00..23.00 rows=1000 width=14) (actual time=0.004..0.061 rows=1000 loops=1)
      -> Hash (cost=7.00..7.00 rows=90 width=110) (actual time=0.040..0.040 rows=90 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 21kB
        -> Seq Scan on review r (cost=0.00..7.00 rows=90 width=110) (actual time=0.006..0.031 rows=90 loops=1)
          Filter: (comment IS NOT NULL)
          Rows Removed by Filter: 210
    -> Hash (cost=1.50..1.50 rows=50 width=15) (actual time=0.018..0.019 rows=50 loops=1)
      Buckets: 1024 Batches: 1 Memory Usage: 11kB
      -> Seq Scan on provider p (cost=0.00..1.50 rows=50 width=15) (actual time=0.008..0.012 rows=50 loops=1)
Planning Time: 2.360 ms
Execution Time: 0.243 ms
(15 rows)

```

Рисунок 12.4 – Схема запиту 2 після додання індексів

12.3. Аналіз впливу на третьому запиті.

Код третього запиту наведено нижче. В таблиці 12.3 наведено час виконання запиту для 5 випробувань до і після додання індексів. На рисунку 12.5 та 12.6 показано схеми запитів та час перших випробувань.

EXPLAIN ANALYZE

SELECT

p.payment_id,
p.payment_method,
o.total_cost AS cost,
c.name AS client_name

FROM payment p

JOIN "order" o ON p.payment_id = o.payment_id

JOIN client c ON o.client_id = c.client_id

WHERE p.payment_method = 'Cash';

Таблиця 12.3 – Аналіз впливу на третьому запиті

Номер спроби	1	2	3	4	5
Час виконання до додання індексу, мс	0.462	0.460	0.566	0.513	0.453
Час виконання після додання індексу, мс	0.447	0.533	0.462	0.447	0.444

Середнє значення перед доданням індексів = 0.491 мс.

Середнє значення після додання індексів = 0.467 мс.

Отже, з індексами час, витрачений на запит, зменшився на 5%.

```
trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#   p.payment_id,
trading_platform=#   p.payment_method,
trading_platform=#   o.total_cost AS cost,
trading_platform=#   c.name AS client_name
trading_platform=# FROM payment p
trading_platform=# JOIN "order" o ON p.payment_id = o.payment_id
trading_platform=# JOIN client c ON o.client_id = c.client_id
trading_platform=# WHERE p.payment_method = 'Cash';

QUERY PLAN

-----
Hash Join  (cost=45.69..73.86 rows=142 width=32) (actual time=0.294..0.436 rows=142 loops=1)
  Hash Cond: (c.client_id = o.client_id)
    -> Seq Scan on client c  (cost=0.00..23.00 rows=1000 width=14) (actual time=0.009..0.070 rows=1000 loops=1)
    -> Hash  (cost=43.91..43.91 rows=142 width=34) (actual time=0.281..0.282 rows=142 loops=1)
        Buckets: 1024  Batches: 1  Memory Usage: 17kB
        -> Hash Join  (cost=22.27..43.91 rows=142 width=34) (actual time=0.115..0.266 rows=142 loops=1)
            Hash Cond: (o.payment_id = p.payment_id)
            -> Seq Scan on "order" o  (cost=0.00..19.00 rows=1000 width=23) (actual time=0.007..0.065 rows=1000 loops=1)
            -> Hash  (cost=20.50..20.50 rows=142 width=19) (actual time=0.103..0.103 rows=142 loops=1)
                Buckets: 1024  Batches: 1  Memory Usage: 15kB
                -> Seq Scan on payment p  (cost=0.00..20.50 rows=142 width=19) (actual time=0.007..0.090 rows=142 loops=1)
                    Filter: ((payment_method)::text = 'Cash'::text)
                    Rows Removed by Filter: 858
Planning Time: 0.438 ms
Execution Time: 0.462 ms
(15 rows)
```

Рисунок 12.5 – Схема запиту 3 до додання індексів

```
trading_platform=# EXPLAIN ANALYZE
trading_platform=# SELECT
trading_platform=#   p.payment_id,
trading_platform=#   p.payment_method,
trading_platform=#   o.total_cost AS cost,
trading_platform=#   c.name AS client_name
trading_platform=# FROM payment p
trading_platform=# JOIN "order" o ON p.payment_id = o.payment_id
trading_platform=# JOIN client c ON o.client_id = c.client_id
trading_platform=# WHERE p.payment_method = 'Cash';

QUERY PLAN

-----
Hash Join  (cost=40.21..68.38 rows=142 width=32) (actual time=0.258..0.401 rows=142 loops=1)
  Hash Cond: (c.client_id = o.client_id)
    -> Seq Scan on client c  (cost=0.00..23.00 rows=1000 width=14) (actual time=0.006..0.064 rows=1000 loops=1)
    -> Hash  (cost=38.44..38.44 rows=142 width=34) (actual time=0.249..0.249 rows=142 loops=1)
        Buckets: 1024  Batches: 1  Memory Usage: 17kB
        -> Hash Join  (cost=16.80..38.44 rows=142 width=34) (actual time=0.087..0.233 rows=142 loops=1)
            Hash Cond: (o.payment_id = p.payment_id)
            -> Seq Scan on "order" o  (cost=0.00..19.00 rows=1000 width=23) (actual time=0.006..0.062 rows=1000 loops=1)
            -> Hash  (cost=15.03..15.03 rows=142 width=19) (actual time=0.077..0.077 rows=142 loops=1)
                Buckets: 1024  Batches: 1  Memory Usage: 15kB
                -> Bitmap Heap Scan on payment p  (cost=5.25..15.03 rows=142 width=19) (actual time=0.034..0.062 rows=142 loops=1)
                    Recheck Cond: ((payment_method)::text = 'Cash'::text)
                    Heap Blocks: exact=8
                -> Bitmap Index Scan on idx_payment_payment_method  (cost=0.00..5.21 rows=142 width=0) (actual time=0.029..0.029 rows=142 loops=1)
                    Index Cond: ((payment_method)::text = 'Cash'::text)
Planning Time: 1.280 ms
Execution Time: 0.447 ms
(17 rows)
```

Рисунок 12.6 – Схема запиту 3 після додання індексів

Отже, введення індексації дещо покращило роботу з базою даних, зменшення часу виконання запитів становить до 12%.

13. ВИСНОВКИ

В рамках курсової роботи було проаналізовано предметну область, бізнес-процеси та існуючі програмні продукти, а також розроблено ER-модель бази даних. На основі цього аналізу була розроблена база даних з використанням діалекту PostgreSQL та проведені дослідження з цією базою даних.

Аналіз предметної області дозволив моделювання баз даних для конкретних умов і потреб, які використовуються для розуміння бізнес-процесів, визначення вимог користувачів і створення концептуальних моделей. Предметний аналіз необхідний для того, щоб забезпечити відповідність нормативним документам і стандартам у предметній області, в якій розробляється програмне забезпечення.

Розуміння бізнес-процесів допомагає визначити завдання, які база даних повинна вирішувати в конкретному бізнес-середовищі. Визначивши основні бізнес-процеси, можна точно визначити необхідні дані та їх структуру.

Вимоги користувачів визначаються реальними процесами в області, де розробляється програмне забезпечення. Обмеження доступу забезпечує цілісність даних, захищає конфіденційність і запобігає витоку даних.

Розроблена ER діаграма спрощує розробку бази даних, а також дозволяє розробити її правильно, згідно з усіма вимогами. Діаграма містить ключові сутності та їхню взаємодію.

Розробка та робота з базами даних передбачає створення програмних додатків для роботи з нею, тригерів, збережених функцій і процедур, а ще представлень і запитів. Запити забезпечують ефективну взаємодію користувача з базою даних, тригери гарантують правильну поведінку бази даних, а функції та процедури спрощують роботу з нею.

Розроблена база даних відповідає сформованим вимогам, забезпечує роботу торгового майданчика для проведення електронних закупівель. База даних здатна не тільки забезпечити ефективну роботу, а й спростити її для всіх учасників процесу, зокрема постачальників товарі та їх клієнтів.

14. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1) PostgreSQL 16.1 Documentation. The PostgreSQL Global Development Group. URL: <https://www.postgresql.org/docs/16/index.html> (дата звернення: 31.12.2023)
- 2) PostgreSQL Tutorial. Basic PostgreSQL Tutorial. URL: <https://www.postgresqltutorial.com/> (дата звернення: 31.12.2023)
- 3) 1. PostgreSQL: how to write a trigger. Cybertec. URL: https://www.cybertec-postgresql.com/en/postgresql-how-to-write-a-trigger/?gclid=CjwKCAiA-bmsBhAGEiwAoaQNmhaZCtF1KjIFP6hmvQLb17NYcr42-p3LqrYRVh6Gefs9uAFv_2UmNBoC2JcQAvD_BwE (дата звернення: 31.12.2023).
- 4) Everything you need to know about Postgres stored procedures and functions. EnterpriseDB. URL: <https://www.enterprisedb.com/postgres-tutorials/everything-you-need-know-about-postgres-stored-procedures-and-functions> (дата звернення: 31.12.2023).
- 5) Майданчики Прозоро URL: https://tender.uub.com.ua/pro-prozorro/maydanchyky-prozorro?gad_source=1&gclid=Cj0KCQiAv8SsBhC7ARIsALIkVT1VAqMFwPXbXTGPHtx0Xt15aTDAyC2G2e1cSDfNOPYCIT1K3-94DvIaAmGvEALw_wcB
- 6) E-tender - Вигідні Тендери | Сайт Тендерів України URL: https://e-tender.ua/?gad_source=1&gclid=Cj0KCQiAv8SsBhC7ARIsALIkVT3g5L1LGRZyK18a6QuooK2E8AHdrbnHvUVtoy9FM42QjeHiOEgoi9oaAhHwEALw_wcB
- 7) Zakupivli.Pro URL: <https://zakupivli.pro/>

ДОДАТОК А

Заповнення бази даних можна організувати за допомогою імпортування з файлів. Нижче наведені команди для заповнення тестовими даними бази даних trading_platform за допомогою імпортування з файлів, що розташовані на фізичному носії за адресою D:\data2\...

```
\COPY client(name, surname, address, telephone, email) FROM 'D:\data2\client.csv'  
DELIMITER ';' CSV HEADER;
```

```
\COPY provider(name, surname, address, telephone, email, rating, service_area) FROM  
'D:\data2\provider.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY contract(start_date, end_date, total_cost, client_id, provider_id) FROM  
'D:\data2\contract.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY service(name, description, price, availability, provider_id) FROM 'D:\data2\service.csv'  
DELIMITER ';' CSV HEADER;
```

```
\COPY product(name, category, price, stock_quantity, provider_id) FROM  
'D:\data2\product.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY tender(start_date, end_date, starting_bid, current_bid) FROM 'D:\data2\tender.csv'  
DELIMITER ';' CSV HEADER;
```

```
\COPY bid(amount, tender_id, provider_id) FROM 'D:\data2\bid.csv' DELIMITER ';' CSV  
HEADER;
```

```
\COPY review(rating, comment, review_date, client_id, provider_id) FROM  
'D:\data2\review.csv' DELIMITER ';' CSV HEADER;
```

```
\COPY payment(cost, payment_date, payment_method) FROM 'D:\data2\payment.csv'  
DELIMITER ';' CSV HEADER;
```

```
\COPY "order"(total_cost, order_date, client_id, payment_id) FROM 'D:\data2\order.csv'  
DELIMITER ';' CSV HEADER;
```