

Тестове завдання

**** Використовувати бажано Transact- SQL або PostgreSQL**

У вас є структура БД для інтернет-магазину:

Таблиця **customers** (Клієнти):

- `id`: INT (унікальний ідентифікатор)
- `name`: VARCHAR (ім'я клієнта)
- `email`: VARCHAR (email клієнта)
- `registration_date`: DATE (дата реєстрації клієнта)

Таблиця **orders** (Замовлення):

- `id`: INT (унікальний ідентифікатор)
- `customer_id`: INT (зовнішній ключ на `customers.id`)
- `order_date`: DATE (дата замовлення)
- `total_amount`: DECIMAL (сума замовлення)

Таблиця **order_items** (Позиції в замовленні):

- `id`: INT (унікальний ідентифікатор)
- `order_id`: INT (зовнішній ключ на `orders.id`)
- `product_name`: VARCHAR (назва продукту)
- `quantity`: INT (кількість)
- `price`: DECIMAL (ціна за одиницю товару)

Таблиця **products** (Продукти):

- `id`: INT (унікальний ідентифікатор)
- `name`: VARCHAR (назва продукту)
- `price`: DECIMAL (ціна товару)
- `stock`: INT (кількість на складі)

Завдання 1: Написання аналітичного запиту

Використовуючи віконні функції, напишіть запит, який для кожного замовлення виводить номер замовлення, клієнта, загальну суму замовлення та рангове місце цього замовлення за сумою серед усіх замовлень.

Завдання 2: Оптимізація зберігання та агрегування даних

Уявіть, що ви керуєте великим магазином із мільйонами товарів та замовлень. Напишіть запит, який повертає топ-10 товарів за прибутком за місяць ($\text{profit} = \text{кількість проданого} * \text{ціна за одиницю}$). Оптимізуйте запит швидкої обробки великого обсягу даних.

Завдання 3: Зберігання та управління історичними даними

Вам необхідно створити механізм зберігання історичних даних про замовлення. Розробіть SQL-скрипт, який:

1. Створить нову таблицю для зберігання історичних даних про замовлення.
2. Напишіть тригер, який при зміні або видаленні замовлення автоматично зберігатиме інформацію про старе значення в таблицю історії.

Завдання 4: Написання процедури

Напишіть процедуру, що зберігається, яка приймає ідентифікатор клієнта і повертає список усіх його замовлень, включаючи загальну суму кожного замовлення, а також кількість унікальних продуктів у кожному замовленні.

Завдання 5: Обробка складних даних

Уявіть, що в базі даних з'явилася помилка, через яку в замовленні вказано негативні суми. Напишіть SQL-запит для пошуку та виправлення всіх таких замовлень, при цьому скоригуйте дані таким чином, щоб вони відповідали реальним суммам товарів у замовленні.

Завдання 6: Проектування схеми

Запропонуйте зміну структури бази даних для покращення продуктивності та масштабованості в умовах швидкого зростання кількості замовлень та клієнтів (за умови отримання 2млн. замовлень на день). А також запропонуйте розширення схеми для уможливлення логувань змін.

Завдання 7: Робота з тимчасовими даними

Напишіть SQL запит для обчислення помісячного приросту нових клієнтів за останній рік. Крім того, визначте середній відсоток приросту клієнтів за місяцями.

Завдання 8 (На вибір виконайте) * :

1. Напишіть Python скрипт для вивантаження даних з таблиці замовлень до Excel у форматі *.xlsx
2. Напишіть Python скрипт для вивантаження даних з GoogleDocs (Excel файл) до таблиці Продукти, автоматизувавши розширення продуктової лінійки.