Київський національний університет імені
Тараса Шевченка
Факультету комп'ютерних наук та кібернетики

Лабараторна робота №1

З дисципліни «Сучасні технології баз даних»

Виконав студент 3-го курсу
Групи ТТП-32 Пелих Олександр

Київ-2024

Тема лабараторної роботи

Розробка реляційної бази даних для управління електронною комерцією

Опис проекту

Метою проекту є створення реляційної бази даних для підтримки роботи електронного магазину. Система забезпечує управління товарами, категоріями, клієнтами, замовленнями, постачальниками, купонами та іншими елементами, що характерні для платформи e-commerce. База даних включає мінімум 15 сутностей, кожна з яких реалізує функціонал, необхідний для ефективного управління магазином.

Проект орієнтований на демонстрацію сучасних підходів до моделювання та реалізації баз даних, таких як використання "м'якого видалення" (Soft Delete), збереження історії змін (дата оновлення та користувач, що змінив дані), тригерів, збережених процедур, а також створення користувацьких функцій та віртуальних таблиць (View).

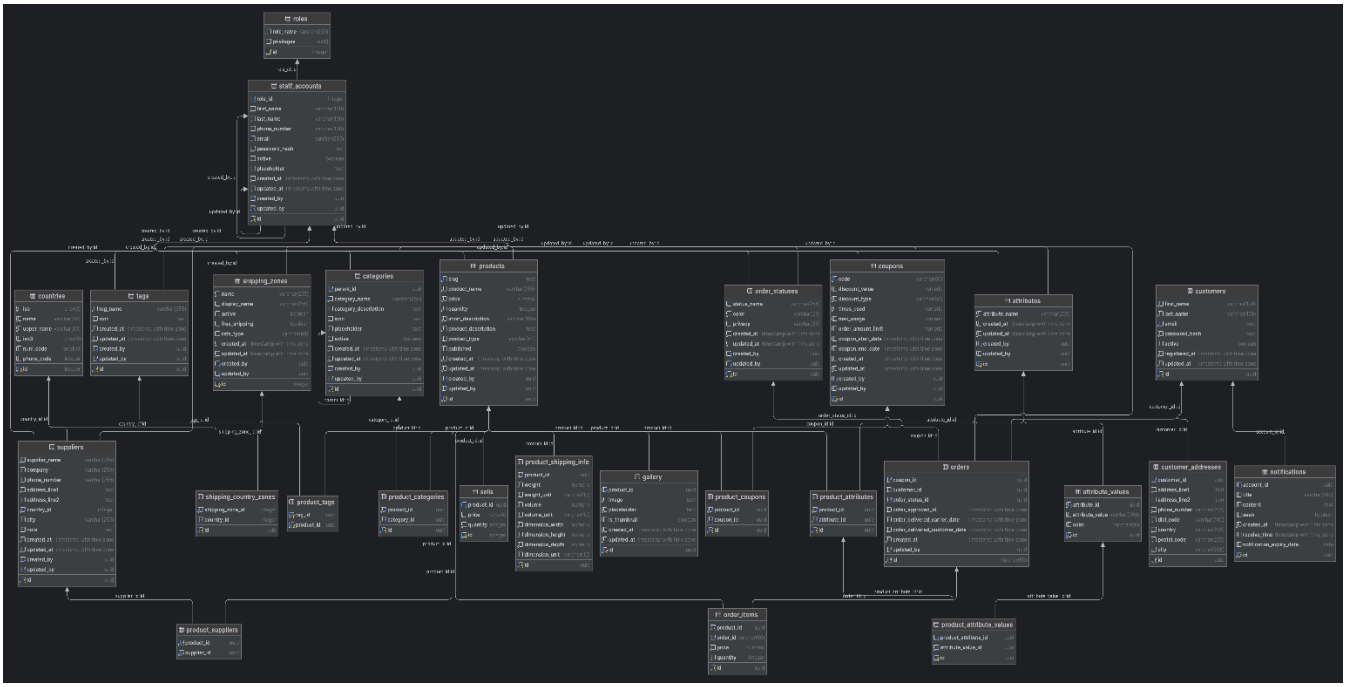
Реалізація

Проект було реалізовано в СУБД PostgreSQL. SQL-скрипти структуровані за категоріями:

- Init.sql: створення таблиць.
- Functions.sql: збережені процедури та функції.
- Triggers.sql: тригери.
- DefaultData.sql: початкові дані.
- Permissions.sql: права доступу.
- Testing.sql: сценарії тестування.

Цей проект демонструє глибоке розуміння принципів проєктування реляційних баз даних та їх реалізації у сучасних СУБД.

Схема реалізованої БД



Опис сутностей реалізованої БД

1) roles

Зберігає інформацію про ролі користувачів та їхні привілеї.

2) staff_accounts

Інформація про працівників магазину. Поля включають ідентифікатор ролі, ім'я, електронну пошту, пароль, активність.

3) categories

Ієрархічна структура категорій товарів. Поля: назва категорії, опис, іконка, активність.

4) products

Основна сутність для зберігання інформації про товари. Поля: назва товару, опис, ціна, кількість, статус публікації.

5) product_categories

Зв'язок між товарами та категоріями.

6) product_shipping_info

Інформація про вагу, об'єм, розміри товару.

7) gallery

Зображення товарів, включаючи мініатюри (thumbnail).

8) attributes та attribute_values

Зберігання атрибутів товарів (наприклад, колір, розмір) та їх значень.

9) product_attributes та product_attribute_values

Визначають зв’язки між товарами, атрибутами та їх значеннями.

10) customers

Дані про клієнтів магазину: ім’я, електронна пошта, пароль.

11) customer_addresses

Адреси клієнтів для доставки.

12) coupons

Управління знижками та промокодами.

13) orders та order_items

Замовлення клієнтів і їхні позиції.

14) suppliers

Інформація про постачальників товарів.

15) notifications

Повідомлення для клієнтів.

Опис функцій у реалізованій БД

1) update_at_timestamp()

- Опис: Оновлює поле updated_at на поточний час перед оновленням рядка.
- Викликається через тригери для 12 таблиць.

2) add_product()

- Опис: Додає новий продукт до таблиці products.
- Параметри:
  - p_slug, p_product_name, p_price, p_quantity, p_short_description, p_product_description, p_product_type, p_published, staff_email.
- Додатково перевіряє, чи існує співробітник із зазначеним email. Якщо ні — викликає помилку.

3) update_product()

- Опис: Оновлює інформацію про продукт за його slug.
- Параметри:
  - p_slug, p_product_name, p_price, p_quantity, p_short_description, p_product_description, staff_email.
- Аналогічно перевіряє існування співробітника перед оновленням.

4) get_product_counts_by_category()

- Опис: Повертає кількість продуктів у кожній категорії (product_type).
- Результат: Таблиця з полями:
  - category_name (назва категорії),
  - product_count (кількість продуктів).

5) get_product_counts_by_user()

- Опис: Повертає кількість продуктів, створених кожним користувачем (співробітником).
- Результат: Таблиця з полями:
  - user_email (email співробітника),
  - product_count (кількість продуктів).

6) notify_customers_about_coupon()

- Опис: Додає сповіщення про новий купон для всіх активних клієнтів (customers).
- Використовується як тригерна функція.

Опис тригерів у реалізованій БД

1) Тригери для автоматичного оновлення поля updated_at

```
CREATE TRIGGER category_set_update BEFORE UPDATE ON categories FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER gallery_set_update BEFORE UPDATE ON gallery FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER attribute_set_update BEFORE UPDATE ON attributes FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER product_set_update BEFORE UPDATE ON products FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER staff_set_update BEFORE UPDATE ON staff_accounts FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER coupon_set_update BEFORE UPDATE ON coupons FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER customer_set_update BEFORE UPDATE ON customers FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER order_set_update BEFORE UPDATE ON orders FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER notification_set_update BEFORE UPDATE ON notifications FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER tag_set_update BEFORE UPDATE ON tags FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER order_status_set_update BEFORE UPDATE ON order_statuses FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
CREATE TRIGGER suppliers_set_update BEFORE UPDATE ON suppliers FOR EACH ROW EXECUTE PROCEDURE update_at_timestamp();
```

- Ці тригери викликають функцію update_at_timestamp() перед оновленням даних у відповідних таблицях. Це забезпечує автоматичне оновлення часу останньої модифікації запису.

2) Тригер для сповіщення клієнтів про новий купон notify_customers

```
CREATE TRIGGER notify_customers
AFTER INSERT ON coupons
FOR EACH ROW
EXECUTE FUNCTION notify_customers_about_coupon();
```

- Цей тригер спрацьовує після додавання нового запису в таблицю coupons. Він викликає функцію notify_customers_about_coupon(), яка створює сповіщення для всіх активних клієнтів (customers) про новий купон.

Висновок

У межах лабораторної роботи було реалізовано низку функцій, тригерів та індексів для ефективного управління даними в реляційній базі даних. Виконані завдання демонструють використання механізмів PL/pgSQL для автоматизації бізнес-логіки на рівні бази даних.

Основні результати:

1. Створено функції для роботи з даними:
    - Функції для додавання та оновлення продуктів із використанням авторизації через електронну пошту працівника.
    - Функції для отримання статистики, такі як кількість продуктів за категоріями та кількість продуктів, створених окремими користувачами.
    - Функція для автоматичного створення сповіщень клієнтам про нові купони.
2. Реалізовано тригери:
    - Тригери для автоматичного оновлення часу модифікації (updated_at) у 12 основних таблицях.
    - Тригер для автоматичного створення записів у таблиці сповіщень після додавання нового купона.
3. Оптимізовано роботу бази даних за допомогою індексів:
    - Створено індекси для ключових таблиць і полів, що часто використовуються в запитах (наприклад, індексація поля email у таблиці customers та зв'язків між таблицями через зовнішні ключі).
4. Використано механізм захисту від помилок:
    - У функціях передбачено перевірку на існування відповідних записів у пов'язаних таблицях (наприклад, перевірка існування працівника за його email перед додаванням чи оновленням продукту).

Дана лабораторна робота продемонструвала практичне застосування розширених можливостей PostgreSQL, таких як тригери, користувацькі функції та індекси, для вирішення завдань, пов'язаних з автоматизацією процесів у базі даних. Реалізовані механізми покращують ефективність роботи з даними та забезпечують гнучкість і надійність системи.