

FashionRetailDB 1

Tworzenie i administrowanie bazami danych

Zadanie 1 - CREATE / ALTER TABLE

1. Utwórz bazę danych fashion_retail_db
2. Przygotuj tabele customers oraz products według poniższego przykładu



fashion_retail_db customers	
customer_id	int(11)
name	varchar(100)
email	varchar(100)
telephone	varchar(30)
gender	varchar(10)
date_of_birth	date
city_id	int(11)
country_id	int(11)



fashion_retail_db products	
product_id	int(11)
category_id	int(11)
subcategory_id	int(11)
description	text
color_id	int(11)
production_cost	decimal(10,2)

3. Dodaj klucze główne do tabel customers oraz products
4. Zaimportuj pozostałe tabele wykorzystując plik „fashion_retail_db_tables.sql”
5. Dodaj klucze obce:
 - Do tabel: customers oraz products;
 - Klucz obcy customer_id do tabeli transactions
 - Klucz obcy product_id do tabeli transactions
6. Dodaj inserty wykorzystując plik „fashion_retail_db_insert.sql”

Zadanie 2 - ALTER / DROP

Tabela positions zawierająca tylko jeden rekord.

1. W tabeli employees dodaj kolumnę default_position typu VARCHAR(50), z domyślną wartością pobraną (np. SELECTem) z tabeli positions
2. Po udanej modyfikacji usuń tabelę positions.

Zadanie 3 - UPDATE

W tabeli products (produkty) dokonaj aktualizacji ceny. Napisz zapytanie UPDATE aktualizujące ceny produktów w tabeli products, podnosząc cenę o 10% tylko dla tych produktów, których cena jest obecnie niższa niż 100.

Zadanie 4 - DELETE

Usuń z tabeli transactions wszystkie rekordy, które dotyczą zamówień starszych niż 1 luty 2023.

Zadanie 5 - SELECT

Napisz kwerendę, która:

1. wybiera wszystkie rekordy z tabeli transactions, gdzie discount jest większy niż 0.5, sortuje wyniki według daty malejąco i ogranicza wynik do 10 rekordów;
2. wyszukuje w tabeli colors unikalne wartości kolumny color i wybiera tylko te kolory, które zaczynają się na literę „b”.
3. z tabeli stores wybiera rekordy, gdzie zip_code zawiera tylko cyfry, sortuje wyniki alfabetycznie według store_name i ogranicza wynik do 5 rekordów (użyj REGEXP).
4. wybiera rekordy z tabeli customers, gdzie kolumna name zawiera frazę „Mark”, a następnie grupuje wyniki według country_id i wyświetla liczbę klientów w każdej grupie.
5. z tabeli transactions oblicza łączną wartość kolumny invoice_total dla rekordów z określonym zakresem dat - od '2024-01-01' do '2024-04-30'.
6. z tabeli transactions oblicza średnią wartość unit_price dla transakcji, w których quantity jest większe niż 2
7. wybiera rekordy z tabeli cities, gdzie nazwa miasta zawiera literę „a”, sortuje wyniki według city rosnąco i ogranicza wynik do 7 rekordów
8. z tabeli subcategories wybiera rekordy, w których kolumna subcategory zaczyna się od litery „s”, sortuje wyniki według subcategory malejąco i ogranicza wynik do 3 rekordów.

Zadanie 6 - JOIN

Napisz kwerendę, która:

1. łączy tabele stores i cities za pomocą INNER JOIN, aby wyświetlić nazwę sklepu (store_name) oraz nazwę miasta (city) odpowiadającej sklepowi.
2. łączy tabele transactions i payment_methods wykorzystując INNER JOIN, aby wyświetlić identyfikator faktury (invoice_id) oraz nazwę metody płatności (payment_method), ogranicz liczbę wyników do 10.
3. łączy tabele transactions i currencies poprzez LEFT JOIN, aby wyświetlić identyfikator faktury (invoice_id) oraz nazwę waluty (currency).
4. łączy tabele stores, cities oraz countries przy użyciu INNER JOIN, aby wyświetlić nazwę sklepu (store_name), nazwę miasta (city) oraz nazwę kraju (country)