

Zadanie programistyczne 2

Julita Kulesza

Back-up utworzonej bazy danych (.bak) zawiera bazę danych dla prostego sklepu internetowego. Celem tejże bazy jest przechowywanie szczegółowych informacji o produktach takich jak ich przypisana kategoria, cena, stan w magazynach, powiązane promocje i okna czasowe kiedy produkty były dostępne w naszej ofercie. Ponadto, baza przechowuje szczegółowe informacje o klientach i historii ich zamówień, z rozbiem na poszczególne produkty dla danego zamówienia.

Baza zawiera tabele

- **discount**, przechowującą
 - identyfikatory zniżek (*bigint*),
 - nazwę każdej poszczególnej zniżki (*varchar(50)*)*,
 - szczegółowy opis (*text*)*,
 - procent zniżki (*decimal(3,0)*)
 - status informujący czy zniżka jest aktywna (*bit*);
- **order_details**, przechowującą
 - identyfikatory zamówień (*bigint*),
 - identyfikatory użytkowników, którzy dokonali tych zamówień (*bigint*),
 - łączne wartości zamówienia (*money*),
 - daty złożenia zamówień (*datetimeoffset(7)*);
- **order_items**, przechowującą
 - identyfikatory zamówionych produktów (*bigint*),
 - identyfikatory odpowiednich zamówień (*bigint*),
 - identyfikatory zamówionych produktów (*bigint*),
 - liczby sztuk zamówionych produktów (*int*);
- **product**, przechowującą
 - identyfikatory produktów (*bigint*),
 - nazwę produktu (*varchar(255)*)*,
 - opisy (*nvarchar(MAX)*)*,
 - identyfikatory kategorii do których należą produkty (*bigint*),
 - daty wprowadzenia produktów (*datetimeoffset(7)*);
 - daty wycofania produktów (*datetimeoffset(7)*)*;
 - identyfikatory zniżek przypisanych do produktu (*bigint*),
 - ceny za sztukę produktów (*money*),
 - identyfikatory do inwentarzy (*bigint*);
- **product_category**, przechowującą
 - identyfikatory kategorii (*bigint*),
 - nazwy (*varchar(50)*)*,
 - opisy (*text*)*;
- **product_inventory**, przechowującą
 - identyfikatory inwentarzy (*bigint*),
 - liczby produktów (*int*);
- **user**, przechowującą
 - identyfikatory klientów (*bigint*),
 - nazwy kont klientów (*varchar(50)*),
 - hasła klientów (*varchar(50)*),
 - pierwsze imiona klientów (*varchar(50)*)*,
 - nazwiska klientów (*varchar(50)*)*,
 - daty zarejestrowania konta (*datetimeoffset(7)*);

* dopuszczane są wartości NULL.

- o daty zawieszenia konta (*datetimeoffset(7)*)*;
- **user_address**, przechowującą
 - o identyfikatory adresów (*bigint*),
 - o identyfikatory klienta z którym powiązany jest adres (*bigint*),
 - o ulice (*varchar(50)*),
 - o nazwy budynków (*varchar(50)*),
 - o miasta (*varchar(50)*),
 - o kody pocztowe (*varchar(10)*),
 - o kraje (*varchar(50)*),

Klucze główne dla każdej z tabel zostały wyróżnione **podkreśleniem**.

W skrypcie *database.sql* tworzę bazę danych sklep_internetowy, następnie tworzę kolejne tabele opisane wcześniej. Na koniec definiuję klucze obce dla kolumn.

W skrypcie *fake_data.sql* dodaję po pięć rekordów do każdej z tabel, z zachowaniem integralności.

Skrypt *transaction.sql* zawiera kod transakcji, w której dla zadanego produktu sprawdza się, czy przechowywana w magazynie ilość tego produktu jest nie mniejsza niż zadana. Jeśli tak, ilość produktu w magazynie jest odpowiednio pomniejszana. Dzięki takiemu podejściu mamy pewność, że nie oferujemy klientowi towaru, którego nie mamy na stanie.

Ponieważ nie dzieje się to automatycznie, w skrypcie *indexes.sql* utworzyłam indeksy na kolumnach będących kluczami obcymi. Dodatkowo, utworzyłam nieklastrowy indeks dla pary column „nazwa” oraz „kategoria” w tabeli produktów. Motywacją był fakt, że ta para kolumn jest często wyszukiwana razem, jako że produkty o tych samych nazwach mogą pojawiać się w różnych kategoriach. Przykład: „opaska,, w kategorii „akcesoria do włosów” oraz „opaska” w kategorii „akcesoria budowlane”.

W skrypcie *stored_procedure.sql* zawarłam procedurę, która po przyjęciu identyfikatora klienta, indeksu produktu który zamierza on kupić oraz interesującej go ilości:

- Zweryfikuje, czy klient znajduje się obecnie w bazie.
- Zweryfikuje, czy produkt jest oferowany przez sklep.
- Następnie sprawdzi, czy ilość produktu jest wystarczająca by zrealizować zamówienie.
- Udostępni właściwą informację w konsoli.

* dopuszczane są wartości NULL.