

Projektowanie aplikacji z bazami danych

lista zadań nr 1

Poniższa lista ma za zadanie wypracować podstawowe umiejętności tworzenia zapytań SQL. Zadania należy zrealizować korzystając z przykładowej bazy danych AdventureWorksLT.

1. Utworzyć zapytanie, które na podstawie SalesOrderHeader.ShipToAddressID zwróci listę miast, do których towary zostały już dostarczone. Lista ma być posortowana i bez powtarzających się wartości.
[1p]
2. Utworzyć zapytanie, które w wyniku zwróci dwie kolumny: nazwę modelu produktu (ProductModel.Name) oraz liczbę produktów tego modelu, przy czym w wyniku chcemy widzieć tylko te, dla których ta liczba jest większa niż 1. Zastanowić się, jakie konsekwencje rodzi fakt wyboru nazwy jako wartości grupującej.
[1p]
3. Utworzyć zapytanie, które w wyniku zwróci trzy kolumny: nazwę miasta (z tabeli Address), liczbą klientów z tego miasta, liczbą SalesPerson obsługujących klientów z tego miasta.
[1p]
4. Kategorie produktów są w strukturze drzewa. Możemy oczekiwać, że wszystkie produkty będą przypisane tylko do kategorii będących w liściach tego drzewa. Utworzyć zapytanie, które zwróci dwie kolumny: nazwę kategorii i nazwę produktu dla produktów będących przypisanych do kategorii nie będących w liściach.
[1p]
5. Utworzyć zapytanie, które w wyniku zwróci trzy kolumny: nazwisko i imię klienta (Customer) oraz kwotę, którą ten klient zaoszczędził dzięki udzielonym rabatom (SalesOrderDetail.UnitPriceDiscount).
[1p]
6. Dla każdego klienta możemy obliczyć kwotę, na jaką złożył zamówienia i niech a będzie średnią z tych kwot. Utworzyć zapytanie, które w wyniku zwróci trzy kolumny: imię i nazwisko klienta oraz kwotę, na jaką ten klient złożył zamówienia, ale liście mają być tylko klienci, dla których ta kwota jest powyżej średniej a .
[1p]
7. Utworzyć tabelę OrdersToProcess(SalesOrderID INT, Delayed BIT), w której będą przechowywane zamówienia jeszcze nie dostarczone, a flaga Delayed określa czy DueDate nie został przekroczony. Przygotować zapytanie, które zaktualizuje tę tabelę w oparciu o tabelę SalesOrderHeader i należy skorzystać z konstrukcji MERGE. Aby przetestować działanie tej metody, warto dogenerować trochę danych (w bazie testowej jest ich niewiele).
[1p]
8. Do tabeli Customer dodaj kolumnę CreditCardNumber, w której nie są dozwolone wartości NULL. Sprawdź jaki poziom walidacji można zapewnić stosując klauzulę CHECK.
[1p]
9. W tabeli SalesOrderHeader dla 3 wybranych pozycji dowolną wartość dla kolumny CreditCardApprovalCode. Następnie, klientom (Customer), którzy mają zamówienia (SalesOrderHeader) z ustawionym CreditCardApprovalCode, zmień pole CreditCardNumber na wartość 'X'.
[1p]
10. Utwórz tabelę Test z kolumną IDENTITY, gdzie identyfikatory mają się zaczynać od 1000 i przesuwać o 10. Zademonstruj różnicę pomiędzy @@IDENTITY i IDENTITY_CURRENT.
[1p]

Uwaga: Jeśli ktoś nie będzie miał pomysłu na zrobienie zadania jednym zapytaniem, może je zrobić kilkoma zapytaniami, gdzie częściowe wyniki będą przechowywane w dodatkowo utworzonych tabelach.

Paweł Rajba