

XML w DB2

Zad. 1. Odczytywanie danych XML za pomocą XQuery i XPath

1. uruchom narzędzie linii komend
2. uruchom menadżera bazy danych (db2start)
3. uruchom DataStudio
4. połącz się z bazą sample
5. w tabeli CUSTOMER, sprawdź, jakiego typu są kolumny. Jak przechowywane są dane XML?
6. napisz zapytanie XQuery, aby:
 - a. wybrać wszystkie dane z kolumny INFO
(wskazówka: wykorzystaj funkcję XQUERY xmlcolumn:
xquery db2-fn:xmlcolumn("CUSTOMER.INFO");)
 - b. wybrać wszystkie elementy name
 - c. wybrać tylko zawartość tekstową elementów name
 - d. wybrać tylko klienta o id 1003
 - e. wybrać tylko nazwiska klientów o id 1003 lub 1005 (tylko zawartość tekstowa)
 - f. wybrać numery telefonów klienta o id 1005
 - g. wybrać numery telefonów domowe klienta o id 1005
 - h. wybrać tylko tych klientów, którzy mają asystenta
 - i. wybrać nazwiska wszystkich asystentów (tylko sama zawartość tekstowa)
 - j. wybrać wszystkie nazwiska i numery telefonów asystentów (tylko sama zawartość tekstowa) (użyj * aby wybrać wszystkie dzieci i potem text())
 - k. wybrać nazwisko klienta asystentów (tylko sama zawartość tekstowa), którego kraj to Canada i który ma numer telefonu domowego
 - l. wybrać numery telefonów klientów z Toronto
 - m. wybrać nazwiska i numery telefonów asystentów klientów z Toronto
7. użyj wyrażenia FLWOR XQuery, aby
 - a. zwrócić sekwencję zawierającą nazwiska i numery telefonów klientów
 - b. zwrócić sekwencję zawierającą nazwiska i numery telefonów klientów, ale wypisać tylko numery telefonów do pracy
 - c. zwrócić sekwencję zawierającą nazwiska klientów (tylko tekst) oraz dla tych klientów, którzy mają asystenta, informację asystent: nazwisko asystenta, dla pozostałych taka informacja ma się nie pojawiać
(użyj wyrażenia XPath: if(warunek) then (wyrażenie) else (wyrażenie))
 - d. wybrać nazwiska klientów, ale tylko tych, którzy mają asystenta (wykorzystać where i let) oraz nazwiska i telefony tych asystentów
 - e. wybrać nazwiska wszystkich klientów i liczbę telefonów tych klientów (użyj funkcji COUNT())
8. użyj funkcji sqlquery, aby zamiast wskazywać całą kolumnę XML, ograniczyć się do wybranych wierszy (wskazówka: wykorzystaj funkcję XQUERY sqlquery:
xquery db2-fn:sqlquery("select INFO from customer");)
i wykonaj zadania:
 - a. wybrać tylko klienta o id 1003
 - b. wybrać tylko nazwiska klientów o id 1003 lub 1005 (tylko zawartość tekstowa)
 - c. wybrać numery telefonów klienta o id 1005
 - d. wybrać numery telefonów domowe każdego klienta o id mniejszym od 1005

9. Połącz się z bazą TESTDB
Utwórz nowy schemat LIB. W tym schemacie utwórz tabele biblioteki (za pomocą skryptu skrypt_biblioteka.sql). Przejrzyj zawartość tych tabel.
10. Dodaj nowego autora do tabeli AUTHOR o nazwisku „Jan Kowalski”; uzupełnij kolumnę INFO odpowiednim dokumentem XML, zawierającym email, kraj i datę urodzenia.
11. Za pomocą polecenia TRANSFORM w SQL/XML wykonaj zadania dla autora Jan Kowalski (odwołaj się do id autora).
 - a. Dodaj kolejny email
 - b. Usuń wszystkie węzły email
 - c. Dodaj nowy węzeł contact, zawierający atrybut type='phone', z wartością tekstową 12345
 - d. Dodaj kolejny węzeł contact, z atrybutem type='mail', z wartością jk@gg.com
 - e. Zmień nazwę pierwszego węzła contact na phone
 - f. Usuń atrybut type z węzła contact
 - g. Zmień wartość atrybutu type w węźle phone na work
 - h. Dodaj nowy element addr, który będzie zawierał dwa elementy potomne: city i street (z wartościami odpowiednio Szczecin, Nowa)
 - i. Dodaj do elementu addr kolejne dziecko country, jako pierwsze z dzieci, z wartością Polska
 - j. Dodaj kolejny węzeł phone (z atrybutem type=home), po już istniejącym węźle phone
 - k. Zmień nazwy wszystkich węzłów phone na contact_phone
 - l. Usuń węzeł addr (wraz z zawartością)
 - m. Dodaj do wszystkich węzłów phone nowy atrybut public=yes