

Zadania rozwiąż proszę w PL/SQL dla Oracle. Gotowe rozwiązanie wyślij na adres zadanie@datafuze.com z tytułem - Zadanie rekrutacyjne.
W przypadku pytań do zadań wysyłaj je na ten sam adres.

PROJEKT COMPARATOR

Mamy tabelę FACT_HISTORY która przechowuje wersje danych tabeli FACT taką że:

- klucz główny PK_FACT_HISTORY (ID_PACK INTEGER, ID_FACT NUMBER) składa się z numeru paczki wersji tj. ID_PACK oraz klucza głównego tabeli źródłowej tj. ID_FACT
- tabela zawiera oprócz kolumn klucza PK_FACT_HISTORY, kolumny danych (np. po 10 VARCHAR2, DATE, NUMBER)

Zad 1.

Napisz pakiet z procedurą COMPARE_FACT (ID_PACK_OLD, ID_PACK_NEW), która pozwala podać dwa różne ID_PACK by porównać dane z obu wersji. Wynik porównania powinien być dostępny w tabeli tymczasowej gdzie:

TMP_FACT_HISTORY_COMPARE_STATUS

- ID_FACT
- STATUS (1 - MODIFIED, 2 - NEW, 3 - DELETED)
- brak ID_FACT w tej tabeli oznacza że wiersz jest identyczny w obu pakietach ID_PACK_OLD oraz ID_PACK_NEW

Zad 2.

Procedura COMPARE_FACT (ID_PACK_OLD, ID_PACK_NEW, BUILD_DIFF default false) pozwala opcjonalnie wymusić uzupełnienie tabeli tymczasowej jn.

TMP_FACT_HISTORY_COMPARE_DIFF

- ID_FACT
- COL_NAME
- COL TYPE
- COL_OLD_VALUE
- COL_NEW_VALUE
- wartości COL_OLD_VALUE oraz COL_NEW_VALUE reprezentują oryginalne wartości kolumn które się różnią w postaci sformatowanej VARCHAR2 (format stały zdefiniowany w pakiecie)
- w tabeli tej znajdują się tylko wiersze STATUS = 1 (MODIFIED)

PROJEKT REZERWACJA MIEJSC

Zad. 3

Napisz procedurę, która umożliwia rezerwację biletu na koncert wg:

- parametr IN <- PESEL
- parametr OUT -> STATUS (0 - dokonana rezerwacja, 1 - wykorzystany limit całkowity, 2 ... inne jeżeli uznasz)
- parametr OUT -> IDR (unikalny id np. UUID rezerwacji gdy się powiodła)

przyjmij że:

- procedurę mogą wywoływać równoległe niezależne sesje np. 50
- limit rezerwacji np. 50k
- każda próba rejestracji powinna zostać odłożona w dedykowanej tabeli (zaproponuj jej strukturę)

* zaproponuj sposób testowania rozwiązania

** dla chętnych procedura do anulowania rezerwacji - rozważ ew. nadużycia

INNE

Zad 4

Masz tabelę ORDERS, która zawiera dane o zamówieniach w następujący sposób:

```
CREATE TABLE ORDERS (  
  ORDER_ID NUMBER,  
  CUSTOMER_ID NUMBER,  
  ORDER_DATE DATE,  
  AMOUNT NUMBER(10,2)  
);
```

Napisz metodę PL/SQL, która przyjmuje dwa argumenty typu DATE (StartDate i EndDate) oraz zwraca sumaryczną kwotę zamówień złożonych pomiędzy tymi datami dla każdego klienta.

Wynik powinien być posortowany malejąco według sumarycznej kwoty.

Zad 5

Napisz metodę, która za parametr przyjmuje json klient:zamówienia i rozkłada go na poszczególne tabele.

Przykład json:

```
{
  "klient": {
    "imie": "Jan",
    "nazwisko": "Kowalski",
    "adres": "ul. Testowa 123, 00-000 Warszawa"
  },
  "zamowienia": [
    {
      "produkt": "Laptop",
      "ilosc": 1,
      "cena": 3500.00
    },
    {
      "produkt": "Smartphone",
      "ilosc": 2,
      "cena": 1500.00
    },
    {
      "produkt": "Słuchawki bezprzewodowe",
      "ilosc": 1,
      "cena": 200.00
    }
  ]
}
```

TABELE:

- CUSTOMERS
 - customer_id NUMBER
 - name VARCHAR2(100)
 - surname VARCHAR2(100)
 - address varchar2(255)
- ORDERS
 - product VARCHAR2(100)
 - cnt NUMBER
 - price NUMBER
 - customer_id NUMBER