Transact SQL (T-SQL) I PL/SQL

I. Proste zadania programistyczne

1. Napisz prosty program w Transact-SQL (PL/SQL). Zadeklaruj zmienną, przypisz na tą zmienną liczbę rekordów w tabeli **OSOBA** (lub jakiejkolwiek innej) i wypisz uzyskany wynik używając instrukcji Print (T-SQL) lub dbms_output (PL/SQL), w postaci napisu np. "W tabeli jest 10 rekordów".

2. Używając T-SQL (PL/SQL), policz dydaktyków z tabeli **DYDAKTYK**. Jeśli ich liczba jest mniejsza niż 16, wstaw dydaktyka: panią doktor *Celestynę Cykorię* i wypisz odpowiedni komunikat. Jeśli liczba pracowników jest większa niż 15, wypisz komunikat informujący o tym, że nie wstawiono danych z powodu braku etatów. Jeśli p. Cykoria została zatrudniona, zatrudnij ją w katedrze *Sztucznej inteligencji* i wygeneruj jej PESEL (p. zadanie VIII.18).

3. Do tabeli Osoba dodaj kolumnę Plec (Char1) jeżeli jej jeszcze nie ma. Przy użyciu kursora odczytaj imiona z tabeli i na ich podstawie wypełnij kolumnę Plec, zakładając że jest ona określona jest przez ostatnią literę imienia (‘a’ dla kobiet, z wyjątkiem imienia Barnaba).

4. Utwórz tabelę **REKRUTACJA**

```
CREATE TABLE Rekrutacja (
Imie Varchar(32), Nazwisko Varchar(32), DataUrodzenia Date, Obywatelstwo
Varchar(16)
);
```

Dopisz dane nowo rekrutowanych studentów

```
INSERT INTO Rekrutacja (Imie, Nazwisko, DataUrodzenia, Obywatelstwo)
VALUES ('Adela','Ananas','1997-02-15','Mołdawia'),
         ('Alojzy','Arbuz','1998-09-13','Ukraina'),
         ('Barnaba','Burak','1999-01-21','Polska'),
         ('Benita','Brukiew','1998-12-22','Niemcy'),
         ('Cyprian','Cząber','1996-08-30','Polska'),
         ('Celestyna','Ciecierzyca','1995-06-03','Słowacja'),
         ('Delfina','Durian','1996-10-30','Francja'),
         ('Dionizy','Daktyl','1997-09-21','Grecja');
```

Na podstawie danych z tabeli **REKRUTACJA** dopisz nowe rekordy do tabel **OSOBA**, **STUDENT** przyjmując następujące założenia:

- płeć określona jest przez ostatnią literę imienia (a dla kobiet, z wyjątkiem imienia Barnaba)
- na podstawie nazwy państwa z tabeli **REKRUTACJA** wpisujemy obywatelstwo studenta w tabeli **OSOBA**.`IdPanstwo`
- data rekrutacji jest datą dzisiejszą zwracaną przez funkcję `Getdate()`
- numery indeksów nadawane są jako kolejne liczby:

```
SELECT Max(RIGHT(rTrim(NrIndeksu), Len(NrIndeksu) - 1)) + 1
FROM  dbo.Student s;
```

poprzedzane literą ‘s’

Wskazówka:

Po zapisaniu danych w tabeli **REKRUTACJA** znajdź aktualnie największy numer indeksu i zapisz go na zmiennej;
Utwórz kursor odczytujący dane z tabeli **REKRUTACJA**. W każdym przejściu kursora:

- dopisz imię, nazwisko i datę urodzenia do tabeli **OSOBA**, odczytaj i podstaw na zmienną wygenerowaną automatycznie wartość `IdOsoba`,
- zaktualizuj w bieżącym wierszu pola `IdPanstwo` i `Plec` tabeli **OSOBA**,
- dopisz nowy rekord do tabeli **OSOBA** – `IdOsoba` i `NrIndeksu`

II. Procedury Transact_SQL i PL/SQL

1. Utwórz w procedurę zwracająca dane studentów (Imię, Nazwisko, Miasto, Numer indeksu), których rok rekrutacji będzie podawany w parametrze procedury.
2. Utwórz procedurę zwracającą liczbę studentów, których rok rekrutacji zostanie podany w parametrze procedury. W T_SQL przećwicz 3 sposoby zwracania danych przez procedurę (parametr typu OUTPUT, RETURN i ResultSet).
3. Utwórz procedurę, która będzie „przenosiła” zapisane w bazie danych osoby z miasta, którego nazwa jest podana w parametrze procedury do innego miasta, też podanego (nazwa) w parametrze procedury. W wyniku działania procedury proszę też wyświetlić komunikat z informacją, ile osób zostało przeniesionych i pomiędzy jakimi miastami.
4. Utwórz procedurę służącą do dopisywania nowego przedmiotu do bazy. Procedura będzie otrzymywała w parametrach nazwę i symbol przedmiotu. W procedurze należy sprawdzić, czy przedmiot o danej nazwie lub symbolu istnieje. Jeżeli nie, należy go dopisać. Na zakończenie należy wypisać komunikat z informacją o wykonaniu (lub nie) operacji.
5. Utwórz procedurę dopisującą nowego dydaktyka do bazy. Imię, nazwisko, płeć i nazwa stopnia naukowego będą podawane w parametrach procedury. W procedurze sprawdź istnienie w bazie stopnia naukowego o podanej w parametrze nazwie, oraz rekordu zawierającego dane kandydata (imię, nazwisko, stopień). Jeżeli taki dydaktyk już jest w bazie odnotowany, nie dopisuj go. Jeżeli stopień o podanej nazwie nie jest w bazie odnotowany, wstaw NULL w rekordzie dydaktyka. Nowo dopisanego dydaktyka zrób podwładnym profesora Cezarego Czosnka. Procedurę należy zakończyć komunikatem informującym o wykonanej operacji. W PL/SQL utwórz i wykorzystaj sekwencję do realizacji wartości klucza głównego w tabelach **OSOBA** i **DYDAKTYK**.

III. Procedury z wykorzystaniem kursora

1. Przy pomocy kursora przejrzyj wszystkich dydaktyków i zmodyfikuj wynagrodzenia tak, aby osoby zarabiające poniżej granicznej wartości miały zwiększone wynagrodzenie o 10%, natomiast osoby zarabiające powyżej kolejnej granicznej wartości miały zmniejszone wynagrodzenie o 10%. Wartości graniczne będą podawane w parametrach procedury. Wypisz informacje o wszystkich wprowadzanych zmianach – imiona i nazwiska osób, którym zmieniono płacę, oraz nowe wartości płac.
2. Utwórz tabelę **SIATKAPLAC** { IdStopien Int FK, Stawka Money}. Utwórz procedurę, która otrzyma w parametrze stawkę minimalną. Wykorzystując kursor, wypełnij tabelę danymi, stosując następujące reguły: stawka minimalna przysługuje inżynierom, a każdy kolejny stopień ma stawkę większą o 20%. Zakładamy, że IdStopien inżyniera ma największą wartość, a stopnie są zapisane według malejącej wartości IdStopien, odwrotnie do ich „ważności". Poza procedurą przypisz stawki wszystkim dydaktykom, zgodnie z ich stopniami. Wykorzystaj skorelowany **UPDATE**.

IV. Wyzwalacze na tabelach bazy danych

W niniejszym zestawie zadań znajduje się znaczna liczba zadań związana z wyzwalaczami. Nie jest spowodowane lansowaniem przez autora przykładów użycia wyzwalacza, jako najlepszego możliwego rozwiązania w SZBD, lecz czysto dydaktycznymi względami. Wyzwalacz jest de facto procedurą, w dodatku na ogół wymaga operowania poleceniami SQL na złączeniach tabel, często wymaga użycia kursora (oby jak najrzadziej!), uświadamia także fakt wykonania wszystkich jego poleceń w jednej transakcji.

1. Utwórz wyzwalacz, który nie pozwoli usunąć rekordu z tabeli **OCENA**.
2. Utwórz wyzwalacz niepozwalający usunąć osoby (dydaktyka), która ma podwładnych. Zakładamy, że może być usuwany tylko jeden rekord i nie jest to zrealizowane przez więzy referencyjne.

   UWAGA: Jeżeli wymuszone zostały więzy referencyjne, mają one wyższy priorytet, niż wykonanie wyzwalacza, zatem pierwszy pojawi się komunikat o próbie ich naruszenia, a wyzwalacz nie zostanie uruchomiony.
3. Utwórz wyzwalacz, który przy wpisywaniu nowego studenta do bazy wygeneruje mu numer indeksu, jeśli nie był on podany w instrukcji **INSERT**.

   UWAGA: Jeżeli wymuszone zostały więzy NOT NULL na kolumnie NrIndeksu, mają one wyższy priorytet, niż wykonanie wyzwalacza, zatem pierwszy pojawi się komunikat o próbie ich naruszenia, a wyzwalacz nie zostanie uruchomiony.

   ORACLE

```
create or replace Trigger tr_NrIndeksu
BEFORE INSERT ON Student
FOR EACH ROW
DECLARE
v_nrIndeksu Varchar2(12) := '';
```

```
BEGIN
IF :new.nrIndeksu IS NULL THEN
  SELECT 's' || Cast(Max(Substr(NrIndeksu, 2)) + 1 As Varchar(12))
  INTO v_nrIndeksu FROM Student;
  :new.nrindeksu := v_nrIndeksu;
    dbms_output.put_line('Wygenerowany nowy nr indeksu ' ||
_nrIndeksu);
END IF;
END;
```

4. Wariant rozszerzony zadania XV-3: do tabeli **STUDENT** dodaj kolumnę KontoWplat Char(22). Zakładamy, że pierwsze 16 znaków jest stałą wartością kodującą numer konta bankowego (może być zapisana „na sztywno" w wyzwalaczu), natomiast ostatnie 6 znaków koduje konto wirtualne każdego studenta. W wyzwalaczu, oprócz wygenerowania nowego numeru indeksu, wygeneruj każdemu nowo dopisywanemu studentowi jego indywidualne konto bankowe. Załóż, że może być dopisywany więcej niż jeden rekord w jednej operacji.

   UWAGI do rozwiązań zadań XI-3 i XI-4. Kod obu zadań powinien być umieszczony w jednym wyzwalaczu, który powinien być jedynym miejscem, w którym tworzone są numery kont i indeksów. Kod zadania XI-4, po niewielkiej modyfikacji, może zostać wykorzystany do wypełnienia kolumny KontoWplat, po jej utworzeniu na tabeli zawierającej rekordy. Po uzupełnieniu danymi, będzie można wymusić na tej kolumnie więzy UNIQUE i NOT NULL (to samo dotyczy kolumny NrIndeksu).

5. Utwórz wyzwalacz, który przy wstawianiu lub modyfikowaniu danych w tabeli **DYDAKTYK** sprawdzi, czy nowe zarobki (wstawiane lub modyfikowane) są większe niż 2000. W przeciwnym wypadku wyzwalacz powinien zmienić na 2000 wartość w kolumnie Placa w modyfikowanym lub wstawianym rekordzie (sprawdzenie można oczywiście osiągnąć używając więzów CHECK na kolumnie Placa; korekty jednak już tą metoda nie da się zrealizować).

6. Utwórz tabelę **BUDZET** (Wartosc INT NOT NULL, DataAktualizacji). W tabeli tej będzie przechowywana łączna wartość wynagrodzeń wszystkich dydaktyków. Tabela będzie zawsze zawierała jeden wiersz. Oblicz początkową sumę zarobków i uzupełnij tabelę **BUDZET**. Należy to zrealizować jednym poleceniem! Utwórz wyzwalacz, który będzie pilnował, aby wartość w tabeli **BUDZET** była zawsze aktualna, a więc przy wszystkich operacjach aktualizujących tabelę **DYDAKTYK** (INSERT, UPDATE, DELETE), wyzwalacz będzie aktualizował wpis w tabeli **BUDZET**.

7. Utwórz tabelę **ROCZNIK** {Rok Int UNIQUE, Liczba Int, DataAktualizacji Date}. Na tabeli **STUDENT** utwórz wyzwalacz, który po każdej zmianie w tabeli (Insert, Update, Delete) uaktualni tabelę **ROCZNIK** tak, aby zawsze zawierała aktualne liczby studentów każdego rocznika (według dat rekrutacji).

V. Zadania dodatkowe

1. Dopisz do bazy UCZELNIA tabelę **WYDZIAL**{(IdWydzial PK, Wydzial NOT NULL}. Na kolumnie klucza głównego zrealizuj autoinkrementację.
2. Dopisz do tabeli **WYDZIAL** 3 przykładowe rekordy (nazwy wydziałów): Baz Danych, Inżynierii Oprogramowania, Sztucznej Inteligencji.
3. Zmodyfikuj tabelę **WYDZIAL** tworząc więzy referencyjne do tabeli **DYDAKTYK** tak, aby każdemu wydziałowi można było przypisać dziekana.
4. Napisz procedurę T-SQL która w parametrach otrzyma imię i nazwisko osoby oraz nazwę wydziału. Procedura sprawdzi, czy dana osoba jest dydaktykiem posiadającym stopień doktora. Jeżeli tak - zapisze daną osobę jako dziekana wskazanego wydziału. Jeżeli nie, wypisze stosowny komunikat.
5. Napisz procedurę, która w parametrze otrzyma rok rekrutacji. Procedura przy użyciu kursora sprawdź braki studentów (niezaliczone przedmioty) z danego roku rekrutacji. Brakiem jest brak oceny z przedmiotu lub ocena niedostateczna. Należy wypisać imię i nazwisko studenta, a poniżej listę braków (nazw przedmiotów). Pod uwagę bierzemy wszystkie przedmioty zapisane w bazie. Raport o brakach powinien zostać zapisany w tabeli tymczasowej.
6. Na tabeli **STUDENTGRUPA** utwórz wyzwalacz, który nie dopuści do dopisania więcej niż 5 studentów do jednej grupy studenckiej.
7. Zmodyfikuj tabelę **GRUPA** dodając kolumnę SredniaOcen. Napisz instrukcję SELECT która dla każdej grupy studenckie obliczy średnią ocen studentów tej grupy. Instrukcję tę wykorzystaj do wykonania skorelowanej instrukcji UPDATE zmodyfikowanej tabeli Grupa. Użyj CTE.
    -