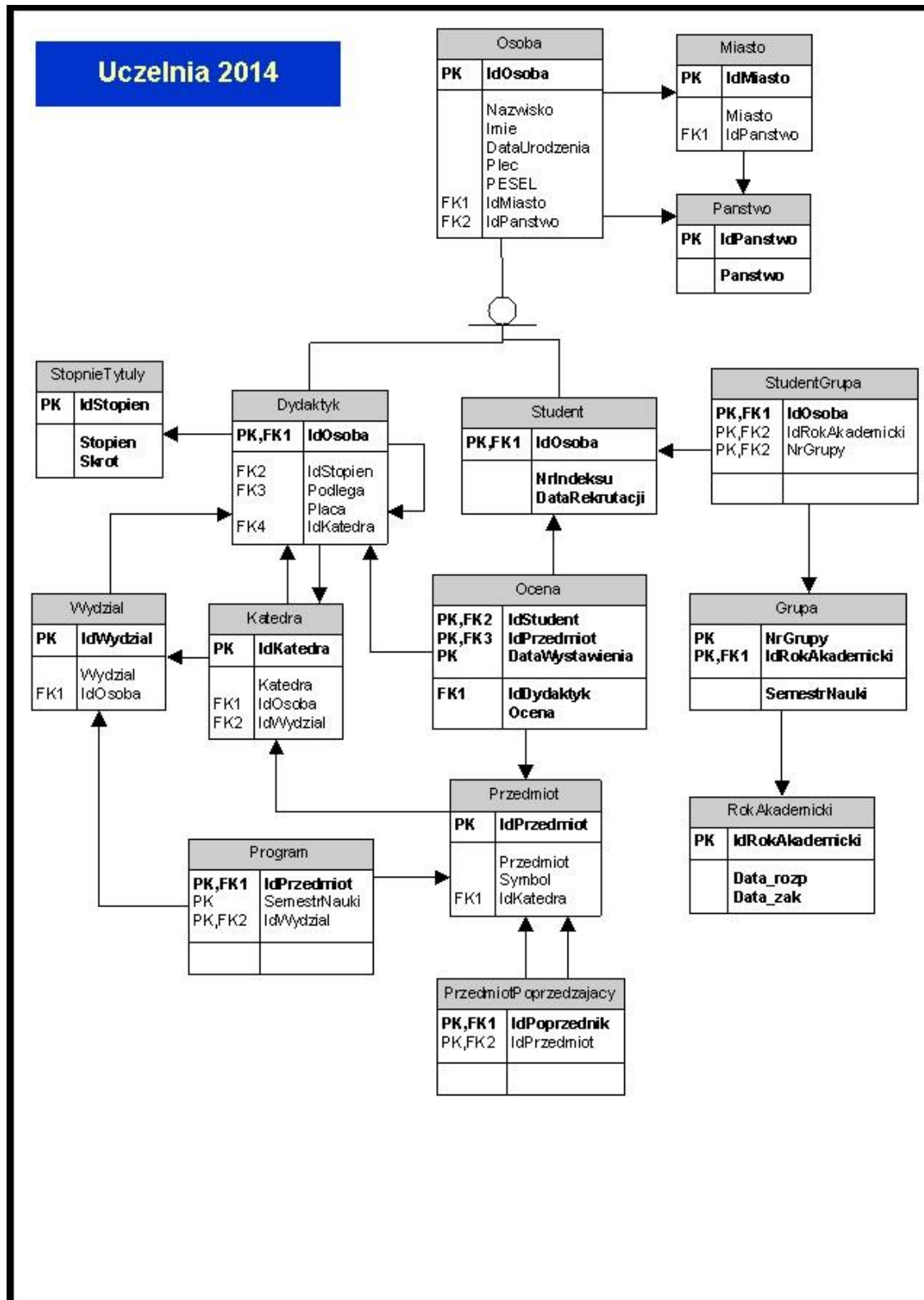


Zadania SELECT do schematu UCZELNIA



Uwagi wstępne

1. Skrypty zakładające tabele i wypełniające je danymi (ORACLE i SQL Server) nie tworzą wszystkich tabel pokazanych na powyższym diagramie. Utworzenie tabel i dopisanie do nich danych jest przedmiotem kilku niżej opisanych zadań.
2. Tabele **GRUPA** i **STUDENTGRUPA** założone przez skrypty mają inną strukturę niż pokazano na diagramie. W tabeli **GRUPA** dodano atrybut **IdGrupa** pełniący rolę klucza głównego (sztucznego), w tabeli **STUDENTGRUPA** kluczem głównym jest para kluczy obcych **IdGrupa** i **IdOsoba**.

Polecenia SELECT

I Część. Proste instrukcje SELECT.

1. Wypisz nazwy i symbole wszystkich przedmiotów z tabeli **PRZEDMIOT**
2. Wypisz wszystkie dane stopni naukowych z tabeli **STOPNIETYTULY**. używając znaku „gwiazdki” zamiast wypisywania nazw kolumn.
3. Wypisz wszystkie wartości wszystkich kolumn z tabeli **GRUPA**.
4. Wypisz imiona, nazwiska i rok urodzenia osób z tabeli **OSOBA**. Posortuj malejąco po roku urodzenia i rosnąco po nazwisku. Użyj funkcji **YEAR** (SQL Server) lub **EXTRACT** (ORACLE).
5. Wypisz w jednej kolumnie imiona i nazwiska wszystkich osób z tabeli **OSOBA**. Kolumnę nazwij „Pracownicy i studenci”. Wynikowe rekordy posortuj rosnąco wg nazwisk.
6. Do kolumny z połączonymi imionami i nazwiskami (zad. 4) dołącz komentarz „student lub dydaktyk”.
7. Wypisz bez powtórzeń wszystkie imiona osób z tabeli **OSOBA**, sortując je alfabetycznie.
8. Używając funkcji **YEAR** (SQL Server) lub **EXTRACT** (ORACLE) wypisz bez powtórzeń i posortuj malejąco lata rekrutacji z tabeli **STUDENT**. Wynikową kolumnę zatytułuj „Lata rekrutacji”.
9. Sprawdź, ile miesięcy, dni i lat upłynęło od daty rekrutacji każdego studenta (podaj tylko numer indeksu). W MS SQL Server użyj funkcji **Datediff()** i **Getdate()**, w ORACLE funkcji **EXTRACT**.

II Część. Ograniczenie zwracanych rekordów przez użycie klauzuli WHERE w instrukcji SELECT.

1. Wypisz imiona osób z tabeli **OSOBA**, których nazwisko zaczyna się na literę ‘K’.
2. Wypisz nazwiska osób z tabeli **OSOBA**, których imiona zawierają literę ‘A’, a nie zawierają litery ‘B’.
3. Wypisz imiona i nazwiska osób, których imiona są pięcioliterowe.
4. Znajdź wszystkie dane osób, dla których znana jest data urodzenia.
5. Znajdź wszystkie dane osób, dla których nieznana jest data urodzenia.
6. Znajdź wszystkie dane osób, dla których nie jest znana płeć.

III Część. Złączenia tabel.

1. Wypisz imiona i nazwiska wszystkich studentów.
2. Wypisz imiona i nazwiska wszystkich dydaktyków.
3. Wypisz imiona i nazwiska osób, które są jednocześnie dydaktykami i studentami.
4. Znajdź imiona, nazwiska i datę rekrutacji studentów zarekrutowanych pomiędzy *1 lipca* i *30 września 2012* roku.
5. Wypisz imiona i nazwiska dydaktyków nie posiadających żadnego stopnia naukowego.
6. Wypisz imiona, nazwiska i stopnie naukowe dydaktyków nie posiadających szefa.
7. Wypisz imiona i nazwiska wszystkich dydaktyków posiadających stopień *doktora*.
8. Wypisz Imiona i nazwiska studentów, którzy otrzymali ocenę 2 z przedmiotu *Analiza matematyczna I* w roku 2013.
9. Wypisz imiona, nazwiska i stopnie naukowe wszystkich dydaktyków. Uwzględnij dydaktyków, którzy nie mają stopnia.
10. W wyniku zadania III.9 spowoduj, żeby w przypadku braku stopnia, w kolumnie **Stopien** pojawił się napis „Brak”.
11. Wypisz bez powtórzeń nazwy przedmiotów, z których wystawiono oceny.
12. Wypisz Imiona i nazwiska studentów, nazwę przedmiotu oraz ocenę, jaką z tego przedmiotu otrzymał. Wynik posortuj po nazwie przedmiotu i ocenie malejąco.
13. Wypisz imiona i nazwiska dydaktyków wraz z imionami i nazwiskami ich przełożonych.
14. Wypisz imiona i nazwiska studentów, uzyskane przez nich oceny z przedmiotów (wypisz ich nazwy) oraz imiona i nazwiska dydaktyków, którzy te oceny wystawili.
15. Wypisz w jednej kolumnie imiona i nazwiska wszystkich studentów z dopiskiem „student” oraz wszystkich dydaktyków z dopiskiem „dydaktyk”. Kolumnę nazwij „Rola w uczelni.
16. Wypisz imiona i nazwiska wszystkich osób będących jednocześnie studentami i dydaktykami.
17. Wypisz w jednej kolumnie imiona i nazwiska studentów, poprzedzając je słowem „Pani” w przypadku kobiet i „Pan” w przypadku mężczyzn.
18. Znajdź imię, nazwisko i daty rekrutacji studentów o numerach indeksów s3045, s3162, s3177.
19. Znajdź imiona, nazwiska, daty rekrutacji i numery indeksów studentek (pań) zarekrutowanych w 2012 roku lub mających nazwisko rozpoczynające się na literę ‘B’ .
20. Wypisz bez powtórzeń imiona i nazwiska studentów (panów), którzy mają wystawioną ocenę zarówno z przedmiotu „Administracja systemów operacyjnych” jak i „Relacyjne bazy danych”.
21. Wypisz nazwiska wszystkich studentów, którzy mają wystawioną ocenę z przedmiotu „Relacyjne bazy danych”, ale nie mają oceny z przedmiotu „Administracja systemów operacyjnych”.

IV Część. Zapytania z funkcjami agregującymi

1. Znajdź liczbę przedmiotów zapisanych w bazie.
2. Policz studentów, którzy zapisali się na studia w 2012 r.
3. Znajdź średnią, najlepszą i najgorszą z ocen wystawionych z przedmiotu o skrócie AM1.
4. Znajdź liczbę wystawionych ocen oraz średnią ocenę z każdego przedmiotu. Podaj nazwę przedmiotu.
5. Znajdź liczbę wystawionych ocen oraz średnią ocenę każdego studenta z każdego przedmiotu. Podaj nazwę przedmiotu, imię i nazwisko studenta.
6. Z wyniku zadania poprzedniego wyeliminuj przypadki wystawienia jednemu studentowi więcej niż jednej oceny z jednego przedmiotu. Wynik posortuj po nazwie przedmiotu rosnąco i średniej ocen malejąco.
7. Nie stosując podapytania znajdź nazwy przedmiotów, z których wystawiono więcej niż 5 ocen pozytywnych (≥ 3).

V Podzapytania

1. Stosując podzapytanie znajdź imiona i nazwiska studentów.
2. Stosując podzapytanie znajdź nazwy przedmiotów, z których wystawiono oceny.
3. Znajdź imiona i nazwiska studentów, którzy rozpoczęli studia w tym samym roku co Alberta Ananas.
4. Znajdź imię, nazwisko i numer indeksu ostatnio przyjętego na uczelnię studenta.
5. Znajdź studentów, którzy mają średnią ocen wyższą niż średnia ocen Hieronima Kapusty.
6. Znajdź imiona i nazwiska studentów, którzy rozpoczęli studia w tym samym roku co Gryzelda Gruszka i mają średnią ocen wyższą niż średnia ocen Seweryna Selera.
7. Znajdź studenta z najwyższą średnią ocen.

VI Podzapytania skorelowane

1. Dla każdego rocznika rekrutacji znajdź pierwszego zarekrutowanego studenta.
2. Znajdź studentów (imię i nazwisko) z najwyższą oceną z każdego przedmiotu (podaj jego nazwę).
3. Znajdź imiona i nazwiska studentów, którzy nie mają jeszcze żadnej oceny.
4. Znajdź imiona i nazwiska studentów, którzy otrzymali co najmniej jedną oceną niedostateczną.
5. Wypisz imiona, nazwiska i stopnie dydaktyków, którzy nie mają podwładnych.
6. Dla każdego rocznika rekrutacji znajdź studenta z najwyższą średnią ocen.

Polecenia DML i DDL

VII DML - wstawianie i modyfikacja danych.

1. Dopisz do bazy dwa rekordy – jednego nowego studenta i jednego dydaktyka. Pamiętaj, że w MS SQL Server kolumna **IdOsoba_** w tabeli **OSOBA** jest wyposażona we właściwość **Identity**, a w ORACLE takiego automatyzmu nie ma. Dydaktykowi nadaj stopień magister. Ostatnie polecenie wykonaj jedną instrukcją SELECT.
2. Nowego dydaktyka zrób podwładnym pana Kajetana Kalafiora. Zadanie należy wykonać przy użyciu JEDNEGO polecenia SQL (można założyć unikalność pary wartości imię i nazwisko).
3. Zmodyfikuj numery indeksów w tabeli **STUDENT**, które omyłkowo nie posiadają litery s przed właściwym numerem.

VIII DDL - tworzenie tabel, zmiana schematu tabel, więzy spójności i związane z tym operacje DML.

1. Utwórz tabelę **MIASTO** {**IdMiasto** PK, **Miasto**}. Dobierz odpowiednie typy danych. Na kolumnie klucza głównego zrealizuj autonumerowanie, na kolumnie **Miasto** nie dopuść wstawienia wartości NULL.
2. Utwórz więzy referencyjne pomiędzy tabelami **MIASTO** i **PANSTWO** pozwalające przypisać każde miasto do jednego Państwa.
3. Przypisz kilka przykładowych miast do właściwych państw.
4. Do tabeli **OSOBA** dodaj kolumnę klucza obcego **IdMiasto**, wskazującą na miasto zamieszkania osoby. Utwórz więzy referencyjne do tabeli **MIASTO**.
5. Osoby o **IdOsoba** zakresu 1 – 8 zrób mieszkańcami Warszawy, o **IdOsoba** równym 10, 12, 14 zrób mieszkańcami Krakowa, a imionach rozpoczynających się od liter G, H, J mieszkańcami Poznania.
6. Do tabeli **OSOBA** dodaj kolumnę klucza obcego **IdPanstwo**, wskazującą na obywatelstwo osoby. Utwórz więzy referencyjne do tabeli **PANSTWO**.
7. Studenta *Salomona Selera* zrób obywatelem Stanów Zjednoczonych AP.
8. Utwórz tabelę **KATEDRA** (**IdKatedra** PK, **Katedra** Not Null). Dobierz odpowiednie typy danych dla kolumn. W MS SQL Server zrealizuj możliwość automatycznej generacji wartości klucza głównego.
9. Do tabeli **DYDAKTYK** dodaj kolumnę **IdKatedra**, utwórz więzy referencyjne do tabeli **KATEDRA**. Rolą utworzonego klucza obcego będzie przechowywanie informacji o przynależności dydaktyków do poszczególnych katedr. Więzy uzupełnij definicją akcji referencyjnej uzupełniającej wartością NULL wskaźnik do usuwanej katedry.
10. Do tabeli **KATEDRA** dodaj kolumnę **IdOsoba**, utwórz więzy referencyjne do tabeli **DYDAKTYK**. Rolą utworzonego klucza obcego będzie przechowywanie informacji o

dydaktykach będących kierownikami poszczególnych katedr. Zagwarantuj niemożność usunięcia dydaktyka będącego kierownikiem katedry.

11. Do tabeli **KATEDRA** wpisz katedry: *Baz danych, Inżynierii oprogramowania, Sztucznej inteligencji*.
12. Dydaktyków, których nazwiska zaczynają się na literę B zrób pracownikami katedry *Baz Danych*.
13. Szefem katedry *Baz Danych* zrób *Bazylego Brokuła*, szefem katedry *Sztucznej Inteligencji* zrób *Kunegundę Karp*.
14. Do tabeli **DYDAKTYK** dodaj kolumnę **Placa** (Money w MS SQL Server i NUMBER(6,2) w ORACLE) z więzami DEFAULT = 2000. Wypełnij kolumnę, przypisując inżynierom i magistrów płacę w wysokości 2500, pozostałym 5000. Sprawdź działanie więzów przez dodanie nowego dydaktyka.
15. Zaktualizuj płace używając składni CASE. Tym razem ustalamy następującą siatkę płac:
 - Profesor Doktor habilitowany 3500 zł
 - Doktor habilitowany 3000 zł
 - Doktor 2500 zł
 - Magister 2000 zł
 - Inżynier 1800 zł
16. Podnieś płacę panu profesorowi Apolinaremu Anyżkowi o 10%.
17. Utwórz tabelę **ListaImion**, zawierającą pojedyncze wystąpienia imion z tabeli **OSOBA**. Zadanie należy wykonać jednym poleceniem – tabela ma zostać utworzona wraz z wpisaniem do niej danych.
18. Do tabeli **OSOBA** dodaj kolumnę **PESEL** Char(11) z opcją UNIQUE. Sprawdź, wstawiając kilka rekordów, czy unikalność numerów PESEL będzie rzeczywiście sprawdzana.

UWAGA: MS SQL Server traktuje NULL, jako wartość znaczącą, mogącą pojawić się tylko jeden raz w kolumnie z więzami UNIQUE!!! Problem ten nie występuje w ORACLE. W MS SQL Server rozwiązaniem jest dodanie kolumny, wypełnienie jej unikalnymi danymi, a następnie dodanie więzów UNIQUE na tej kolumnie. Poniższy przykład pokazuje metodę uzyskania unikalnych wartości, niemających jednak nic wspólnego z rzeczywistymi wartościami PESEL.
19. Do tabeli **OSOBA** dodaj więzy CHECK na kolumnie **DataUrodzenia** pilnując, aby w bazie nie pojawiły się osoby urodzone po 1999-01-01 ani przed 1900-01-01. Sprawdź działanie tych więzów.
20. Do tabeli **OSOBA** dodaj kolumnę **Plec** Char(1) z opcją NOT NULL. Kolumna będzie dopuszczała tylko dwie wartości: K lub M co należy zrealizować przy pomocy więzów CHECK.

UWAGA: Serwer bazy danych nie dopuści do dodania do tabeli kolumny z więzami NOT NULL, jeżeli w tej tabeli znajdują się rekordy. Rozwiązaniem jest dodanie kolumny, uzupełnienie jej znaczącymi danymi a następnie utworzenie na niej więzów NOT NULL.

Obejściem problemu może być dodanie kolumny z więzami DEFAULT i wypełnienie jej w trakcie tworzenia zadeklarowaną wartością domyślną. W ORACLE będzie to operacja automatyczna, w MS SQL Server dyrektywę DEFAULT należy uzupełnić klauzulą WITH VALUES.

21. Kolumnę **Plac** wypełnij danymi, przyjmując założenie, że imiona żeńskie kończą się literą „a”.
22. Sprawdź działanie wszystkich utworzonych więzów.

Sposób rozwiązania tego zadania pozostawiamy studentom do samodzielnego wykonania.

IX Widoki (perspektywy)

1. Utwórz widok **V_DYDAKTYK** prezentujący dane dydaktyków – *Imię, Nazwisko, Stopień* (tekst), *Miasto* (tekst). Perspektywa powinna uwzględniać dydaktyków nieposiadających żadnego stopnia naukowego, a także tych, którzy nie podali swojego miejsca zamieszkania. Sprawdź działanie widoku.
2. Wykonaj widok **V_OcenaIns** służący do wpisywania ocen. Zagwarantuj niemożliwość wstawienia poprzez ten widok oceny spoza zbioru {2, 3.0, 3.5, 4.0, 4.5, 5.0}.

UWAGA: oczywiście nadal istnieje możliwość wstawienia oceny spoza tego zbioru, instrukcją odwołującą się bezpośrednio do tabeli, lub do innej perspektywy.

X Usuwanie danych z tabel

1. Usuń jednego dopisanego przez siebie dydaktyka i jednego studenta, odwołując się do ich nazwisk.

W rozwiązaniu podany jest przykład usunięcia tylko jednego studenta:

Ale jeżeli student został już przypisany do grupy studenckiej i/lub miał wystawioną oceną, należy te wpisy usunąć z tabel **OCENA** i **STUDENTGRUPA** w sposób analogiczny do usuwania rekordu z tabeli **STUDENT**.

2. Sprawdź, czy uda się usunąć miasto z tabeli **MIASTO**, do którego odwołują się rekordy z tabeli **OSOBA**. A może uda się usunąć tabelę **MIASTO**?

Nie uda się – nie dopuszczają do tego więzy referencyjne pomiędzy tabelami.

XI Usuwanie obiektów bazy danych

1. Usuń tabelę **LISTAIMION** oraz perspektywę **V_DYDAKTYK**.

Perspektywę możemy usunąć “bezkarne”. Tabelę, do której nie odwołują się rekordy z innych tabel również. W przypadku istnienia takich odwołań zadziałają więzy referencyjne i (ewentualnie, jeśli istnieją) akcje referencyjne.

Transact SQL (T-SQL) I PL/SQL

XII Proste zadania programistyczne

1. Napisz prosty program w Transact-SQL (PL/SQL). Zadeklaruj zmienną, przypisz na tą zmienną liczbę rekordów w tabeli **OSOBA** (lub jakiegokolwiek 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 Cykorie* 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', 'Ciecierzycza', '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**

XIII Procedury Transact_SQL i PL/SQL

1. Utwórz w procedurę zwracającą 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**.

XIV 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](#).

XV 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.
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).

XVI 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 sprawdzi 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 studenckiej obliczy średnią ocen studentów tej grupy. Instrukcję tę wykorzystaj do wykonania skorelowanej instrukcji UPDATE zmodyfikowanej tabeli Grupa. Użyj CTE.