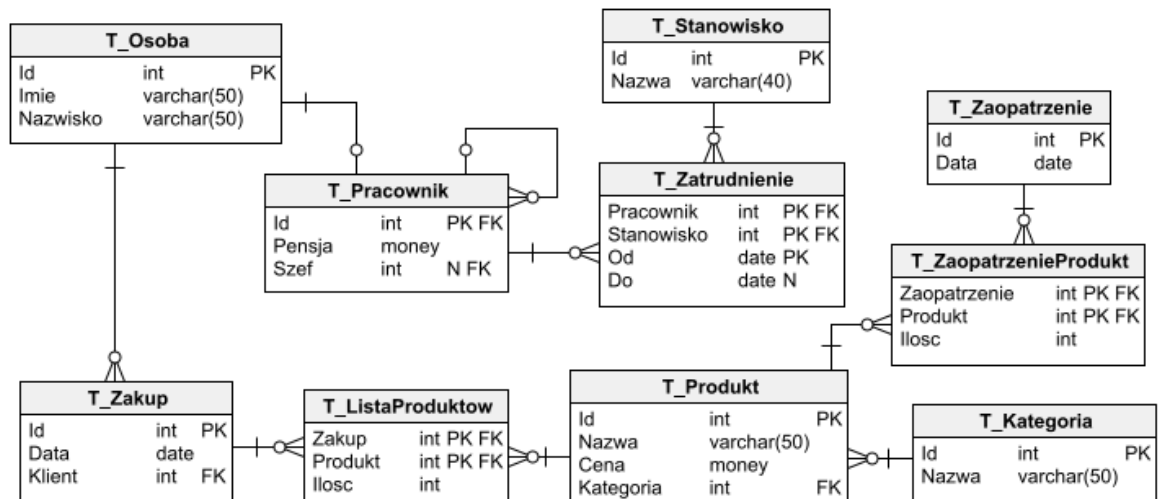


SBD Lab06

T-SQL Wyzwalacze

Link do wygenerowania poniższej bazy danych: [link](#).



Zad 1

Napisz prosty wyzwalacz *INSTEAD OF DELETE*, który nie pozwoli nam na usunięcie rekordów z tabeli *T_ListaProduktow*. Wypisz błąd „Nie można usuwać rekordów z tabeli *T_ListaProduktow*” poprzez *raiserror*.

Przed utworzeniem wyzwalacza przetestuj usuwanie rekordu, zadziała:

```
DELETE FROM T_ListaProduktow
WHERE zakup = 55 AND produkt = 10;
```

Następnie, po utworzeniu wyzwalacza, spróbuj usunąć inny rekord:

```
DELETE FROM T_ListaProduktow
WHERE zakup = 55 AND produkt = 5;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_ListaProduktow_Delete, Line 5 [Batch Start Line 9]
Nie można usuwać rekordów z tabeli T_ListaProduktow
```

Zad 2

Przekształć wyzwalacz z zadania 1 tak, aby dodatkowo wypisywał informację: „*Usuwanie rekordu dla zakupu= {IdZakupu} i produktu= {IdProduktu} nie powiodło się*”. Wykorzystaj w tym celu tabelę *Deleted*.

Po zaktualizowaniu wyzwalacza spróbuj jeszcze raz usunąć rekord z tabeli:

```
DELETE FROM T_ListaProduktow
WHERE zakup = 55 AND produkt = 5;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_ListaProduktow_Delete, Line 10 [Batch Start Line 16]
Nie można usuwać rekordów z tabeli T_ListaProduktow. Usuwanie rekordu dla zakupu= 55 i produktu= 5 nie powiodło się.
```

Zad 3

Napisz wyzwalacz *INSEAD OF INSERT* dla tabeli T_Produkt, który będzie wypisywać: „*Naruszono więzy klucza głównego, rekord nie został dodany*” jeśli podany klucz główny już istnieje. W przeciwnym razie dodajemy rekord do tabeli wypisując informację: „*{produkt} został dodany*”. Nie korzystaj z *IF-a*, zamiast tego użyj *TRY CATCH*.

Po utworzeniu wyzwalacza, spróbuj dodać produkt z id, które już istnieje:

```
INSERT INTO T_Produkt(Id, Nazwa, Cena, Kategoria) VALUES (10, 'Cucumber', 0.25, 2);
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Produkt_Insert, Line 14 [Batch Start Line 22]
Naruszono więzy klucza głównego, rekord nie został dodany
Msg 3616, Level 16, State 1, Line 23
An error was raised during trigger execution. The batch has been aborted and the user transaction, if any, has been rolled back.
```

Po utworzeniu wyzwalacza, spróbuj dodać produkt z id, które jeszcze nie istnieje:

```
INSERT INTO T_Produkt(Id, Nazwa, Cena, Kategoria) VALUES (11, 'Cucumber', 0.25, 2);
```

Oczekiwany wynik:

```
Cucumber został dodany
```

Zad 4

Napisz wyzwalacz *AFTER INSERT*, który nie pozwoli na dodanie nowej osoby do tabeli T_Osoba jeśli istnieje już osoba o takim samym nazwisku. Jeśli natomiast takie nazwisko jeszcze nie istnieje, wtedy dodajemy nową osobę wraz z informacją: „*{nazwisko} został pomyślnie dodany*”.

Po utworzeniu wyzwalacza spróbuj dodać osobę o nazwisku istniejącym już w tabeli:

```
INSERT INTO T_Osoba(Id, Imie, Nazwisko) VALUES (11, 'Keith', 'Theramenes')
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Osoba_Insert, Line 10 [Batch Start Line 21]
Osoba o takim nazwisku już istnieje, rekord nie został dodany
Msg 3609, Level 16, State 1, Line 22
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj dodać osobę o nazwisku nie istniejącym jeszcze w tabeli:

```
INSERT INTO T_Osoba(Id, Imie, Nazwisko) VALUES (11, 'Trevor', 'Telesphorus')
```

Oczekiwany wynik:

```
Telesphorus został pomyślnie dodany
```

Po dokonaniu wszystkich operacji poniższe zapytanie powinno zwracać wynik widoczny na obrazku:

```
SELECT imie, nazwisko FROM T_Osoba;
```

	imie	nazwisko
1	Thomas	Theramenes
2	Mark	Clearchus
3	John	Cheirisophus
4	Ethan	Ephialtes
5	Peter	Phrynichus
6	Liam	Letodorus
7	Richard	Sitalces
8	Paul	Prothytes
9	Chris	Charitimides
10	Paul	Prepelaus
11	Trevor	Telesphorus

Zad 5

Przerób wyzwalacz z zad 4 na *INSTEAD OF INSERT*, który będzie działać tak samo.

Po utworzeniu wyzwalacza spróbuj dodać osobę o nazwisku istniejącym już w tabeli:

```
INSERT INTO T_Osoba(Id, Imie, Nazwisko) VALUES (12, 'Keith', 'Theramenes')
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Osoba_Insert, Line 9 [Batch Start Line 20]
Taka osoba już istnieje, rekord nie został dodany
```

Po utworzeniu wyzwalacza spróbuj dodać osobę o nazwisku nie istniejącym jeszcze w tabeli:

```
INSERT INTO T_Osoba(Id, Imie, Nazwisko) VALUES (12, 'Mike', 'Mindarus')
```

Oczekiwany wynik:

```
Mindarus został pomyślnie dodany
```

Po dokonaniu wszystkich operacji poniższe zapytanie powinno zwracać wynik widoczny na obrazku:

```
SELECT imie, nazwisko FROM T_Osoba;
```

	imie	nazwisko
1	Thomas	Theramenes
2	Mark	Clearchus
3	John	Cheirisoph...
4	Ethan	Ephialtes
5	Peter	Phrynichus
6	Liam	Letodorus
7	Richard	Sitalces
8	Paul	Prothytes
9	Chris	Charitimides
10	Paul	Prepelaus
11	Trevor	Telesphorus
12	Mike	Mindarus

Zad 6

Utwórz wyzwalacz *AFTER*, który po wstawieniu nowego pracownika w tabeli *T_Pracownik* automatycznie przypisze mu szefa (o id= 2) jeśli nie podamy tej wartości w instrukcji *INSERT*. Wykorzystaj w tym celu tabelę *Inserted*.

Uwaga: W tym zadaniu używamy wyzwalacza jedynie w celach treningowych, ponieważ taką funkcjonalność najlepiej byłoby zrealizować zakładając ograniczenie *DEFAULT* na kolumnę „Szef” w następujący sposób:

```
ALTER TABLE T_Pracownik
ADD CONSTRAINT default_szef
DEFAULT 2 FOR Szef;
```

Po utworzeniu wyzwalacza wstaw nowego pracownika nie przypisując mu szefa:

```
INSERT INTO T_Pracownik(id, pensja) VALUES (10, 5555);
```

Następnie sprawdź dane tego pracownika:

```
SELECT id, pensja, szef FROM T_Pracownik WHERE id = 10;
```

Oczekiwany wynik:

	id	pensja	szef
1	10	5555.00	2

Zad 7

Utwórz wyzwalacz, który przy wstawianiu lub modyfikowaniu danych w tabeli *T_Pracownik* sprawdzi czy nowe zarobki (wstawiane lub modyfikowane) są mniejsze niż 10 000. Jeśli warunek nie zostanie spełniony wyzwalacz powinien zgłosić błąd poprzez *raiserror* i nie dopuścić do wstawienia rekordu.

Uwaga: W tym zadaniu używamy wyzwalacza jedynie w celach treningowych, ponieważ taką funkcjonalność najlepiej byłoby zrealizować poprzez założenie więzów spójności typu *CHECK* na kolumnę pensja w następujący sposób:

```
ALTER TABLE T_Pracownik
ADD CONSTRAINT CHK_Pensja CHECK (Pensja < 10000);
```

Po utworzeniu wyzwalacza spróbuj zmodyfikować pensję pracownika na większą niż 10 000:

```
UPDATE T_Pracownik
SET pensja = 40000
WHERE id =1;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Pracownik_Pensja, Line 10 [Batch Start Line 17]
Pensja za duża, dodawanie/modyfikowanie rekordu nie powiodło się
Msg 3609, Level 16, State 1, Line 18
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj dodać nowego pracownika z pensją większą niż 10 000:

```
INSERT INTO T_Pracownik(Id, pensja) VALUES (7, 40000)
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Pracownik_Pensja, Line 10 [Batch Start Line 17]
Pensja za duża, dodawanie/modyfikowanie rekordu nie powiodło się
Msg 3609, Level 16, State 1, Line 18
The transaction ended in the trigger. The batch has been aborted.
```

Zad 8

Napisz wyzwalacz *FOR UPDATE* dla tabeli *T_Zatrudnienie*, który nie pozwoli na modyfikowanie wartości żadnej z kolumn z wyjątkiem kolumny *Do*. Dodatkowo kolumnę *Do* można modyfikować tylko wtedy gdy jej wartość jest *NULLEM* i nie może mieć przypisanej daty *Do* wcześniejszej niż data *Od*.

Po utworzeniu wyzwalacza spróbuj zmodyfikować jakąś kolumnę inną niż *Do*:

```
UPDATE T_Zatrudnienie
SET pracownik = 3
WHERE pracownik = 1;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Update, Line 8 [Batch Start Line 29]
Można modyfikować jedynie kolumnę "Do"
Msg 3609, Level 16, State 1, Line 30
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj zmodyfikować wartość *Do*, która nie jest *NULLEM*:

```
UPDATE T_Zatrudnienie
SET Do = CONVERT(Date, '11-09-2021', 103)
WHERE pracownik = 1 AND stanowisko = 1;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Update, Line 13 [Batch Start Line 29]
Można modyfikować jedynie wartości "Do" które są NULLem
Msg 3609, Level 16, State 1, Line 30
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj przypisać datę *Do* wcześniejszą niż data *Od*:

```
UPDATE T_Zatrudnienie
SET Do = CONVERT(Date, '11-09-2016', 103)
WHERE pracownik = 5 AND stanowisko = 2;
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Update, Line 18 [Batch Start Line 30]
Data "Do" nie może być wcześniejsza niż data "Od"
Msg 3609, Level 16, State 1, Line 31
The transaction ended in the trigger. The batch has been aborted.
```

Zad 9

Napisz wyzwalacz *FOR INSERT* dla tabeli *T_Zatrudnienie*, który:

- Nie pozwoli na ręczne przypisywanie wartości dla kolumny *Do* (czyli przy dodawaniu rekordu wartość *Do* powinna zostać pominięta)
- Nie pozwoli na przypisanie pracownika na stanowisko na którym jest już aktualnie zatrudniony (czyli ma *NULL* w kolumnie *Do* dla tego samego stanowiska)
- Nie pozwoli na przypisanie daty *Od*, która jest wcześniejsza niż jakkolwiek inna data *Od* lub *Do* dla danego pracownika
- Wypisze pracownika ze starego stanowiska kiedy przypisujemy mu nowe (czyli zmieni *Do* dla starego stanowiska na datę przypisania nowego stanowiska)

Po utworzeniu wyzwalacza spróbuj ręcznie wpisać wartość dla kolumny *Do*:

```
INSERT INTO T_Zatrudnienie(stanowisko, pracownik, "Od", "Do") VALUES
(3,2,CONVERT(DATE, '19-03-2023', 103), GETDATE());
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Insert, Line 10 [Batch Start Line 28]
Nie można ręcznie wpisać wartości dla "Do", rekord nie został dodany
Msg 3609, Level 16, State 1, Line 29
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj dodać zatrudnienie na to samo stanowisko na którym dany pracownik jest już zatrudniony:

```
INSERT INTO T_Zatrudnienie(pracownik, stanowisko, Od)
VALUES(6,2, CONVERT(Date, '19-03-2023', 103));
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Insert, Line 15 [Batch Start Line 28]
Pracownik jest już aktualnie zatrudniony na dane stanowisko, rekord nie został dodany
Msg 3609, Level 16, State 1, Line 29
The transaction ended in the trigger. The batch has been aborted.
```

Po utworzeniu wyzwalacza spróbuj dodać zatrudnienie z datą *Od* wcześniejszą niż inna data *Od*/*Do* danego pracownika:

```
INSERT INTO T_Zatrudnienie(pracownik, stanowisko, Od)
VALUES(6,3, CONVERT(Date, '06-06-2006', 103));
```

Oczekiwany wynik:

```
Msg 50000, Level 16, State 1, Procedure TR_Zatrudnienie_Insert, Line 20 [Batch Start Line 28]
Data "Od" nie może być wcześniejsza niż inne daty "Od" i "Do" danego pracownika, rekord nie został dodany
Msg 3609, Level 16, State 1, Line 29
The transaction ended in the trigger. The batch has been aborted.
```

Zad 10

Utwórz tabelę *T_SprzedaneProdukty* z jedną kolumną „wartosc”, która będzie przechowywała wartość wszystkich sprzedanych produktów i będzie zawsze zawierała tylko jeden wiersz. Utwórz jeden wyzwalacz, który będzie pilnował, aby wartość w tabeli *T_SprzedaneProdukty* była zawsze aktualna. Przy wszystkich operacjach aktualizujących tabelę *T_ListaProduktow* (*INSERT*, *UPDATE*, *DELETE*), wyzwalacz powinien aktualizować wartość w tabeli *T_SprzedaneProdukty*.

Tworzenie tabeli:

```
CREATE TABLE T_SprzedaneProdukty (wartosc INT NOT NULL)
```

Obliczamy wartość początkową

```
DECLARE @Wartosc money;  
SELECT @Wartosc = SUM(lp.ilosc * p.cena)  
FROM T_ListaProduktow lp JOIN T_Produkt p ON lp.produkt = p.id;  
INSERT INTO T_SprzedaneProdukty (wartosc) VALUES (@Wartosc);
```