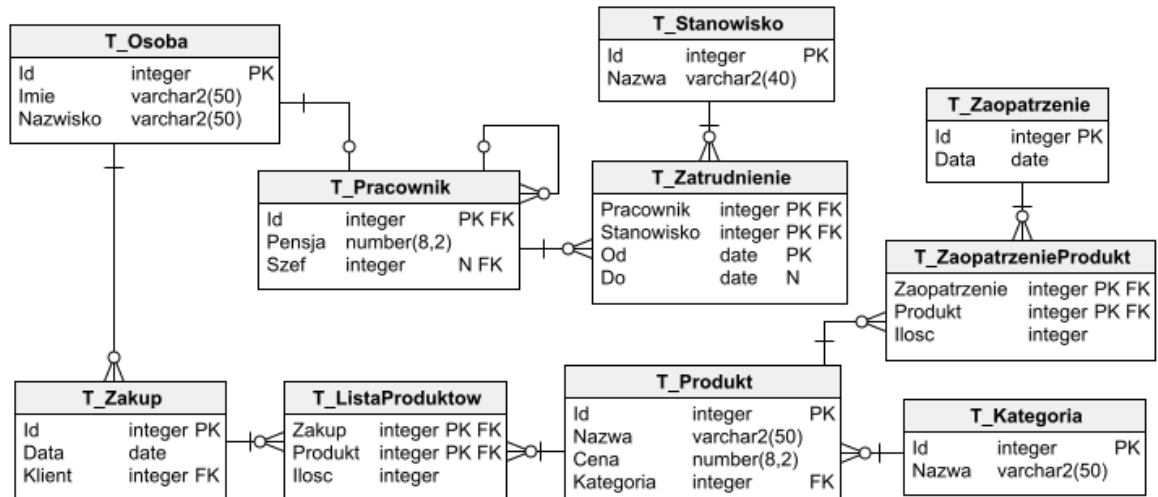


# SBD Lab11

## PL/SQL Kursory

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

Link do skryptu drop: [link](#).



### Zad 1

Przy pomocy kursora przejrzyj wszystkie produkty (tabela *T\_Produkt*) i zmodyfikuj ceny tak, aby produkty droższe niż \$2 staniały o 10%, a produkty tańsze niż \$1 podrożały o 5%. Dla każdego zmienionego rekordu wypisz informację: „Cena {nazwa\_produkту} została zmieniona na: {nowa\_cena}\$”. Wykorzystaj pętlę *LOOP*. Cenę zaokrąglaj do 2 miejsc po przecinku.

Oczekiwany wynik:

```
Cena cod została zmieniona na: 1,98$
Cena herring została zmieniona na: 2,48$
Cena garlic została zmieniona na: ,2$
Cena potato została zmieniona na: ,05$
Cena mango została zmieniona na: 1,04$
Cena grapefruit została zmieniona na: ,41$
Cena trout została zmieniona na: 1,98$
Cena haddock została zmieniona na: 3,11$
```

### Zad 2

Przerób kod z zadania 1 na procedurę wykorzystującą kursor tak, aby wartości cen produktów do obniżki i podwyżki nie były stałe, tylko były parametrami procedury. Nie korzystaj z *IF-a*, zamiast tego użyj *CASE*.

### Zad 3

Utwórz nowe zlecenie zaopatrzenia sklepu (tabela *T\_Zaopatrzenie*) z dzisiejszą datą. Następnie, korzystając z kursora, przypisz do tego zlecenia wszystkie produkty, które sprzedały się w więcej niż 10 sztukach w grudniu 2022. W *T\_ZaopatrzenieProdukt* jako ilość podaj dwukrotność sprzedanej ilości sztuk danego produktu z grudnia 2022. Po dodaniu każdego produktu wypisz informację: „Zamówiono produkt o ID= {id} w ilości= {ilość}”. Nie korzystaj z *IF-a*.

Podpowiedź:

Najpierw należy utworzyć nowy rekord w tabeli *T\_Zaopatrzenie* i przechwycić wygenerowane id (PK ma właściwość Identity). Następnie przy użyciu kursora należy wstawić do tabeli *T\_ZaopatrzenieProdukt* rekordy dla produktów, które nas interesują.

Oczekiwany wynik:

```
Zamówiono produkt o ID= 7 w ilości= 44
Zamówiono produkt o ID= 9 w ilości= 48
Zamówiono produkt o ID= 8 w ilości= 24
Zamówiono produkt o ID= 2 w ilości= 30
Zamówiono produkt o ID= 6 w ilości= 28
```

### Zad 4

Dodaj kolumnę „Bonus” typu *money* z opcją *NULL* do tabeli *T\_Pracownik*. Następnie, korzystając z kursora, przypisz wartość bonusu każdemu aktualnie zatrudnionemu pracownikowi. Bonus jest wyliczany na podstawie stażu w miesiącach i wynosi:  $\text{pensja} * \text{ilość\_miesięcy} / 100$ . Jest on przyznawany tylko osobom, które pracowały co najmniej 6 miesięcy i nie może wynieść więcej niż 30% pensji. Stwórz perspektywę przechowującą id\_pracownika i jego staż w miesiącach, która będzie wykorzystywana przez kursor. Po dodaniu bonusu wypisz informację: „Pracownik od id= {id} ma przypisany bonus w wysokości= {bonus} % pensji”.

Na przykład:

- bonus dla pracownika, który przepracował 35 miesięcy wyniesie 30% pensji
- bonus dla pracownika, który przepracował 27 miesięcy wyniesie 27% pensji
- pracownik, który przepracował tylko 3 miesiące nie dostanie bonusu

Podpowiedź:

Do obliczenia stażu w miesiącach użyj funkcji *months\_between()*. Aktualnie zatrudnieni pracownicy mają przynajmniej jednego *NULLa* w kolumnie „Do” w tabeli *T\_Zatrudnienie*. Jako datę rozpoczęcia pracy należy uznać najwcześniejszą datę „Od” z tabeli *T\_Zatrudnienie* dla danego pracownika, biorąc pod uwagę wszystkie stanowiska na których pracował. Jako datę końcową należy uznać dzisiejszą datę (*current\_date*).

Oczekiwany wynik (może się różnić w zależności od dzisiejszej daty):

```
Pracownik od id= 2 ma przypisany bonus w wysokości= 30 % pensji
Pracownik od id= 3 ma przypisany bonus w wysokości= 30 % pensji
Pracownik od id= 5 ma przypisany bonus w wysokości= 27 % pensji
Pracownik od id= 6 ma przypisany bonus w wysokości= 15 % pensji
```

Tworzenie kolumny Bonus:

```
ALTER TABLE T_Pracownik
ADD Bonus Number(8,2) NULL;
```

## Zad 5

Dodaj kolumnę „Ulubiony\_produkty” typu *integer* z opcją *NULL* do tabeli *T\_Osoba*. Następnie, korzystając z kursora, jako wartość ulubionego produktu przypisz produkt, który dana osoba kupiła w największej ilości we wszystkich zakupach. Po dodaniu produktu wypisz informację: „Dodano ulubiony produkt o id= {id} dla osoby o id= {id}”.

Oczekiwany wynik:

```
Dodano ulubiony produkt o id= 7 dla osoby o id= 7
Dodano ulubiony produkt o id= 6 dla osoby o id= 8
Dodano ulubiony produkt o id= 5 dla osoby o id= 9
Dodano ulubiony produkt o id= 10 dla osoby o id= 3
Dodano ulubiony produkt o id= 2 dla osoby o id= 10
Dodano ulubiony produkt o id= 1 dla osoby o id= 2
Dodano ulubiony produkt o id= 7 dla osoby o id= 6
Dodano ulubiony produkt o id= 6 dla osoby o id= 4
Dodano ulubiony produkt o id= 5 dla osoby o id= 5
```

Tworzenie kolumny Ulubiony\_produkty:

```
ALTER TABLE T_Osoba
ADD Ulubiony_produkty integer null;

ALTER TABLE T_Osoba
ADD CONSTRAINT FK_Osoba_Produkt FOREIGN KEY (Ulubiony_produkty)
REFERENCES T_Produkt (Id);
```

Po dodaniu danych poniższe zapytanie powinno zwracać wynik widoczny na obrazku:

```
SELECT o.nazwisko, p.nazwa AS "ulubiony produkt"
FROM T_Osoba o LEFT JOIN T_Produkt p ON o.ulubiony_produkty = p.id;
```

	NAZWISKO	ulubiony produkt
1	Clearchus	cod
2	Prepelaus	herring
3	Phrynichus	kiwi
4	Charitimides	kiwi
5	Ephialtes	mango
6	Prothytes	mango
7	Letodorus	grapefruit
8	Sitalces	grapefruit
9	Cheirisophus	pumpkin
10	Theramenes	<null>