

Laboratorium 5. Modyfikacja bazy danych

XX = numer indeksu

1. Napisz kursor, który wyświetli imię i nazwisko wszystkich pracowników.
2. Napisz kursor, który wyświetli wszystkie nazwy baz danych na serwerze.
3. Stwórz tabelę INF_XX_KOSZYK:

	Column Name	Data Type	Allow Nulls
	k_id_koszyka	int	<input type="checkbox"/>
	k_data_zakupu	date	<input type="checkbox"/>
	k_data_dostarczenia	date	<input checked="" type="checkbox"/>
▶	k_czy_faktura	varchar(1)	<input type="checkbox"/>
			<input type="checkbox"/>

← IDENTITY

4. Stwórz tabelę INF_XX_PRODUKT:

	Column Name	Data Type	Allow Nulls
▶	p_id_produktu	int	<input type="checkbox"/>
	p_k_id_koszyka	int	<input type="checkbox"/>
	p_lp	int	<input type="checkbox"/>
	p_nazwa	varchar(200)	<input type="checkbox"/>
	p_ilosc	int	<input type="checkbox"/>
	p_cena_jednostkowa	numeric(6, 2)	<input type="checkbox"/>
	p_rabat	numeric(6, 2)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

← IDENTITY

5. Wprowadź następujące dane do tabeli INF_XX_KOSZYK:

	k_id_koszyka	k_data_zakupu	k_data_dostarczenia	k_czy_faktura
1	1	2023-04-15	2023-04-20	T
2	2	2023-08-10	2023-08-20	N
3	3	2024-04-10	2024-04-15	T
4	4	2024-04-15	NULL	N
5	5	2024-05-04	2024-05-10	T

6. Wprowadź następujące dane do tabeli INF_XX_PRODUKT:
PAMIĘTAJ O ZACHOWANIU ID KOSZYKA Z POPRZEDNIEGO ZADANIA

	p_id_produktu	p_k_id_koszyka	p_lp	p_nazwa	p_ilosc	p_cena_jednostkowa	p_rabat
1	1	1	1	Mleko UHT	2	2.50	0.00
2	2	1	2	Kecup Heinz	1	4.99	0.10
3	3	1	3	płatki Nestle	1	7.99	0.20
4	4	1	4	masło domowe	1	5.99	0.15
5	5	2	1	ser żółty	0	24.49	0.00
6	6	2	2	Mleko UHT	10	4.99	0.10
7	7	2	3	kielbasa śląska	0	17.99	0.30
8	8	2	4	cukier	1	3.99	0.00
9	9	3	1	mąka pszenna	1	3.99	0.10
10	10	3	2	woda mineralna	6	1.99	0.00
11	11	3	3	czekolada Goplana	2	5.45	0.20
12	12	3	4	sok jabłkowy	1	4.99	0.00
13	13	4	1	musztarda francuska	2	3.99	0.00
14	14	4	2	piwo Żywiec	6	2.99	0.15
15	15	4	3	chleb razowy	2	2.49	0.10
16	16	4	4	jabłka Jonagold	3	4.45	0.00
17	17	5	1	banany	1	2.49	0.00
18	18	5	2	sok gruszkowy	2	5.99	0.10
19	19	5	3	musztarda francuska	1	3.99	0.00
20	20	5	4	płatki Nestle	2	7.99	0.00

7. Wstęp do zadania:

Firma X sprzedaje produkty w sklepie wysyłkowym. Każdy koszyk posiada unikalne pole (`k_id_koszyka`), posiada datę zakupu (`k_data_zakupu`) i może mieć datę dostarczenia, która może być pusta (`k_data_dostarczenia`). Mamy również informację czy klient życzy sobie wystawienia faktury (`k_czy_faktura`, pole przyjmuje wartości 'T' lub 'N'). Koszyk składa się z produktów, które posiadają unikalne pole (`p_id_produkту`) oraz klucz obcy wskazujący z jakiego koszyka to produkt (`p_k_id_koszyka`).

Produkty w koszyku ma pole (`p_lp`), produkty w jednym koszyku są numerowane od 1. Łączna cena produktu to iloczyn liczby sztuk i ceny jednostkowej, pomniejszony o rabat (`p_rabat`). Każdy produkt w koszyku może mieć indywidualny rabat. Atrybut `p_rabat` zawiera rabat wyrażony procentowo (np. 0.2, 0.3, 0.4 oznacza rabat o wysokości 20%, 30% i 40%, odpowiednio).

8. Stwórz perspektywę (VIEW) INF_XX_vKOSZYK:

Wyświetlającą id koszyka, liczbę produktów, cenę przed rabatem i po rabacie (**CENY ZAOKRĄGLONE DO 2 MIEJSC**)

	<code>k_id_koszyka</code>	LICZBA PRODUKTÓW	CENA PRZED RABATEM	CENA PO RABACIE
1	1	4	23.97	20.97
2	2	4	53.89	48.90
3	3	4	31.82	29.24
4	4	4	44.25	41.06
5	5	4	34.44	33.24

9. Napisz procedurę, która przyjmuje parametry ID koszyka i pole czy rabat do obliczenia wartości koszyka z opcją rabat albo bez rabatu (ZAOKRĄGLENIE):

```
EXEC INF_XX_WARTOSC_KOSZYK 1, 'N';
```

WARTOŚĆ KOSZYKA
23.97

```
EXEC INF_XX_WARTOSC_KOSZYK 1, 'T';
```

WARTOŚĆ KOSZYKA
20.97

10. Zmodyfikuj powyższą procedurę – przekształcając ją na funkcję:

Różnica między funkcją a procedurą:

Funkcja musi zwracać wartość

```
CREATE PROCEDURE [INF_XX_WARTOSC_KOSZYK] @ID_KOSZYK int, @CZYRABAT varchar(1)
AS
```

```
CREATE FUNCTION [INF_XX_WARTOSC_KOSZYK_F] (@ID_KOSZYK int, @CZYRABAT varchar(1))
RETURNS decimal(6,2)
AS
BEGIN
RETURN jakas wyliczana zmienna;
END;
```

Wywołanie:

```
SELECT k_id_koszyka, dbo.INF_XX_WARTOSC_KOSZYK_F(k_id_koszyka, 'T') as 'WARTOSC
KOSZYKA' FROM INF_XX_KOSZYK;
```

	k_id_koszyka	WARTOSC KOSZYKA
1	1	20.97
2	2	48.90
3	3	29.24
4	4	41.06
5	5	33.24

```
SELECT k_id_koszyka, dbo.INF_XX_WARTOSC_KOSZYK_F(k_id_koszyka, 'n') as 'WARTOSC
KOSZYKA' FROM INF_XX_KOSZYK;
```

	k_id_koszyka	WARTOSC KOSZYKA
1	1	23.97
2	2	53.89
3	3	31.82
4	4	44.25
5	5	34.44

11. Do tabeli INF_XX_KOSZYK dodaj kolumnę k_punkty_payback:

```
ALTER TABLE INF_XX_KOSZYK  
ADD k_punkty_payback [int]
```

12. Stwórz procedurę INF_XX_NALICZ_PAYBACK, która przypisze punkty do koszyków według reguły:

Zasady przyznawania punktów lojalnościowych PayBack są następujące:

- za każde 10 zł wydane na produkty nieobjęte rabatem (p_rabat=0) przyznawany jest 1 punkt,
- za każde 15 zł wydane na produkty objęte jakimkolwiek rabatem przyznawany jest 1 punkt.

Naliczone punkty PayBack są zawsze liczbą całkowitą przyciętą w dół

```
EXEC INF_XX_NALICZ_PAYBACK;  
SELECT * FROM INF_XX_KOSZYK;
```

13. Napisz procedurę, która wyliczy wartość koszyków w roku i miesiącu podanym jako parametry (z uwzględnieniem rabatów) INF_XX_WARTOSC_PARAM

W zdaniach można wykorzystać kursor

Podpowiedź na potrzeby sumowania rekordów z wyników użyj kursora:

```
DECLARE nazwaKursora CURSOR
FOR
  'POLECENIE_SQL_KTÓRE_MOŻE_ZWRÓCIĆ_KILKA_REKORDÓW'

OPEN nazwaKursora
FETCH NEXT FROM nazwaKursora INTO 'POSZCZEGÓLNE_POŁA_DO_JAKICH_ZMIENNYCH_PRZEPISAĆ';

WHILE @@FETCH_STATUS = 0
BEGIN
  'OPERACJE_NA_KURSORE'
  FETCH NEXT FROM nazwaKursora INTO 'POSZCZEGÓLNE_POŁA_DO_JAKICH_ZMIENNYCH_PRZEPISAĆ';
END

CLOSE nazwaKursora
DEALLOCATE nazwaKursora
```