

Bazy danych

Ps 5 – Predefiniowane funkcje języka SQL

1. Wypisać imię, nazwisko i stanowisko pracownika według następującego wzorca: Pracownik imię nazwisko jest zatrudniony na stanowisku stanowisko.
2. Przekształcić zapytanie z zadania 1 tak, aby imię i nazwisko były wypisywane wielkimi literami, a stanowisko zaczynało się wielką literą.
3. Podać imię i nazwisko pracownika oraz cztery pierwsze litery nazwy jego działu. Skróconą nazwę działu wypisać wielkimi literami. Skrót działu księgowości zastąpić skrótem KSIG.
4. Podać listę imion i nazwisk pracowników w porządku od najkrótszego do najdłuższego nazwiska.
5. Podać pensję, datę zatrudnienia i nazwę działu pracownika, którego nazwisko jest parametrem zapytania. Przekształcić zapytanie tak, aby wielkość liter wpisywanego nazwiska mogła być dowolna.
6. Podać imię i nazwisko pracownika oraz:
 1. dzień tygodnia, w którym został zatrudniony;
 2. datę i godzinę jego zatrudnienia w formacie:
numer_dnia nazwa_miesiąca numer_roku godzina:minuta:sekunda.
7. Podać imiona, nazwiska i nipy pracowników, których nipy zaczynają się od cyfr 15.
8. Podać identyfikatory przełożonego i jego podwładnego oraz różnicę pomiędzy datami ich zatrudnienia wyrażoną w:
 1. dniach;
 2. miesiącach;
 3. latach.
9. W zapytaniu z zadania 8 zastąpić identyfikatory pracowników ich nazwiskami.
10. Dokonać symulacji zwiększenia pensji pracowników w następujący sposób: 0 20%, gdy pensja jest nie większa niż 2500, 0 10%, gdy pensja jest nie większa niż 3500, 0 5% w pozostałym przypadku.

Pomoc

1. Funkcje numeryczne
 1. ABS(n) – zwraca wartość bezwzględną liczby n.
 2. CEIL(n) – zaokrągla liczbę n do najmniejszej liczby całkowitej większej lub równej n.
 3. FLOOR(n) – zaokrągla liczbę n do największej liczby całkowitej mniejszej lub równej n.
 4. ROUND(n [, m]) – zaokrągla liczbę n do m miejsc po przecinku. Jeśli m jest pominięte, to przyjmuje się, że m=0. Gdy m jest ujemne, to liczba n jest zaokrąglana do liczby m miejsc przed przecinkiem.
2. Funkcje znakowe
 1. CONCAT(ciag1, ciag2) lub ciag1||ciag2 - łączy ze sobą ciągi znaków ciag1 i ciag2.
 2. INITCAP(ciag) – zamienia pierwszą literę ciągu znaków ciag na wielką, a pozostałe na małe.
 3. LOWER(ciag) – zamienia wszystkie litery ciągu znaków ciag na małe.
 4. UPPER(ciag) – zamienia wszystkie litery ciągu znaków ciag na wielkie.
 5. LENGTH(ciag) – zwraca długość ciągu znaków ciag.
 6. REPLACE(ciag, ciag1 [, ciag2]) – zwraca ciąg znaków ciag, w którym każde wystąpienie ciągu znaków ciag1 jest zastąpione przez ciąg znaków ciag2. Jeżeli ciag2 jest pominięte, to przyjmuje się, że ciag2="".
 7. SUBSTR(ciag, n [, m]) – zwraca podciąg ciągu znaków ciag o długości m, rozpoczynający się od znaku o numerze n. Jeżeli m jest pominięte, to zwracane są wszystkie pozostałe znaki występujące w ciągu znaków ciag, począwszy od n. Jeżeli n jest ujemna, to pozycja liczona jest od końca ciągu znaków.

3. Funkcje datowe
 1. `data1-data2` – zwraca liczbę dni pomiędzy datami `data1` i `data2`.
 2. `MONTHS_BETWEEN(data1, data2)` – zwraca liczbę miesięcy pomiędzy datami `data1` i `data2`.
 3. `TO_CHAR(data [, format])` – zwraca datę w postaci tekstu, zapisaną według podanego formatu `format`.
 4. Formaty (podawane w apostrofach)
 1. `DD` – numer dnia miesiąca;
 2. `DAY` – nazwa dnia tygodnia;
 3. `MM` – numer miesiąca;
 4. `MONTH` nazwa miesiąca;
 5. `MON` – trzyliterowy skrót nazwy miesiąca;
 6. `YYYY` – numer roku;
 7. `YY` – skrócony numer roku;
 8. `YEAR` – rok słownie;
 9. `HH` lub `HH12` – godzina (od 1 do 12);
 10. `HH24` – godzina (od 1 do 24);
 11. `MI` – minuta;
 12. `SS` – sekunda.
4. Funkcje konwertujące
 1. `TO_CHAR(n [, format])` – zamienia liczbę `n` na ciąg znaków zgodnie z formatem `format`.
 2. `TO_DATE(ciąg [, format])` – zamienia ciąg znaków `ciąg` na datę zgodnie z formatem `format`.
 3. `TO_NUMBER(ciąg [, format])` – zamienia ciąg znaków `ciąg` na liczbę zgodnie z formatem `format`.
5. Inne
 1. `NVL(wyrażenie1, wyrażenie2)` – zwraca `wyrażenie1`, jeżeli jest ono różne od `NULL`; w przeciwnym przypadku zwraca `wyrażenie2`.
 2. `¶metr` – umożliwia wprowadzenie z klawiatury wartości, która zostanie podstawiona do zapytania w miejscu występowania wyrażenia `¶metr`.
 3. `CASE`
`WHEN warunek1 THEN wartosc1`
`WHEN warunek2 THEN wartosc2`
...
`WHEN warunekn THEN wartoscn`
`[ELSE wartosc]`
`END [alias]`
Wyświetla wartość w zależności od tego, jaki warunek jest spełniony.
 4. `CASE wyrażenie`
`WHEN wartosc1 THEN wartosc1'`
`WHEN wartosc2 THEN wartosc2'`
...
`WHEN wartoscn THEN wartoscn'`
`[ELSE wartosc']`
`END [alias]`
Wyświetla wartość w zależności od tego, jaką wartość przyjmuje podane wyrażenie.