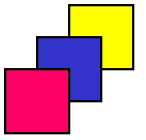
Język SQL. Rozdział 2.
Proste zapytania

**Polecenie SELECT, klauzula WHERE,
operatory SQL, klauzula ORDER BY.**



Wprowadzenie do języka SQL

- **Język dostępu do bazy danych.**
- **Język deklaratywny, zorientowany na przetwarzanie zbiorów.**
- **Grupy poleceń języka:**
  - **DML (ang. *Data Manipulation Language*),**
  - **DDL (ang. *Data Definition Language*),**
  - **DCL (ang. *Data Control Language*).**
- **Polecenie SQL może być zapisane:**
  - **w jednym bądź wielu wierszach,**
  - **dużymi lub małymi literami.**
- **Polecenie SQL zawsze kończymy średnikiem.**

```
SELECT *
FROM pracownicy;
```

```
select * from
PRACOWNICY;
```

Projekcja

- **Wybór wartości określonych atrybutów relacji.**

```
SELECT nazwisko, etat
FROM pracownicy;
```

```
SELECT id_prac, nazwisko,
placa_pod, zatrudniony, id_szefa
FROM pracownicy;
```

Wyrażenia arytmetyczne

- **Operatory arytmetyczne:**
  - **+, -, *, /**

```
SELECT nazwisko, placa_pod*12
FROM pracownicy;
```

```
SELECT etat, placa_pod/30,
placa_pod*12
FROM pracownicy;
```

Polecenie DESCRIBE

- **Wyświetla strukturę relacji, przekazanej jako parametr polecenia.**

```
SQL> describe zespoly
 Nazwa                          Wartość NULL? Typ
 ------------------------------ ------------ --------------------
 ID_ZESP                        NOT NULL     NUMBER(4)
 NAZWA                                       VARCHAR2(15)
 ADRES                                       VARCHAR2(10)
```

- **Typy atrybutów:**
  - **NUMBER(precyzja, skala) – liczba,**
  - **VARCHAR2(długość) – ciąg znaków**
  - **DATE – data.**

Uwaga!

DESCRIBE jest poleceniem narzędzia SQL*Plus, nie języka SQL!

Aliasy nazw atrybutów relacji

- **Alias to alternatywna nazwa atrybutu, z aliasów można korzystać podczas sortowania i prezentacji wyników.**

```
SELECT nazwisko,
        placa_pod AS placa_podstawowa,
        placa_pod*12 AS "roczna płaca",
        placa_pod/20 AS dniówka
FROM pracownicy;
```

Operator konkatenacji - ||

- **Umożliwia łączenie (sklejanie) wartości wyświetlanych atrybutów tekstowych i literałów.**

```
SELECT 'Pracownik ' || nazwisko || ' zarabia ' || placa_pod
AS tekst
FROM pracownicy;
```

Obsługa wartości pustych

- **Wartość pusta – wartość niedostępna, nieprzypisana, nieznana lub nieistotna; oznaczana jako NULL.**
- **Może negatywnie wpływać na wyniki operacji arytmetycznych.**

```
SELECT nazwisko, placa_pod*12 + placa_dod AS zarobki_razem
FROM pracownicy;
```

- **Zastępowanie wartości pustych:**
  - `COALESCE(wyr_1, wyr_2, ...)` **– element standardu SQL,**

```
SELECT nazwisko, placa_pod*12 +
       COALESCE(placa_dod, 0) AS zarobki_razem
FROM pracownicy;
```

  - `NVL(wyr_1, wyr_2)` **– dostępna w SZBD Oracle.**

```
SELECT nazwisko, placa_pod*12 +
       NVL(placa_dod, 0) AS zarobki_razem
FROM pracownicy;
```

Eliminowanie powtórzeń

- Realizowane przy użyciu klauzuli DISTINCT lub UNIQUE.
- Klauzule umieszczane bezpośrednio po klauzuli SELECT.

```
SELECT etat FROM pracownicy;
```

```
SELECT DISTINCT etat FROM pracownicy;
```

```
SELECT UNIQUE etat FROM pracownicy;
```

- Możliwa jest eliminacja powtórzeń z wielu wyrażeń.

```
SELECT DISTINCT etat, id_zesp FROM pracownicy;
```

Porządkowanie wyników zapytania (1)

- **Realizowane przy użyciu klauzuli ORDER BY.**
- **ORDER BY występuje najczęściej jako ostatnia klauzula zapytania.**
- **Wskazanie kolejności sortowania:**
  - **ASC – rosnąco (domyślnie), DESC – malejąco.**
- **Porządek sortowania:**
  - **liczby – od mniejszych do większych,**
  - **daty – od wcześniejszych do późniejszych,**
  - **łańcuchy znaków – alfabetycznie.**
- **Jeżeli klauzula ORDER BY nie zostanie użyta, wiersze zostaną zwrócone w całkowicie losowej kolejności.**
- **Można korzystać z aliasów i numerów kolumn (użycie numerów kolumn jest niezgodne ze standardem SQL3).**

```
SELECT nazwisko, etat,
placa_pod * 12 AS zarobki
FROM pracownicy
ORDER BY nazwisko,
                zarobki DESC, etat;
```

```
SELECT nazwisko, etat,
placa_pod * 12 AS zarobki
FROM pracownicy
ORDER BY 1, 3 DESC, 2;
```

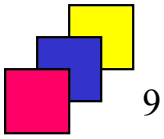
Porządkowanie wyników zapytania (2)

- **Wyrażenie porządkujące nie musi być elementem zbioru wynikowego.**

```
SELECT nazwisko FROM pracownicy
ORDER BY placa_pod DESC;
```

- **Pozycja wartości pustych w zbiorze wyników:**
  - **różnie w różnych RDBMS,**
  - **w SZDB Oracle: na końcu dla porządku rosnącego i na początku dla porządku malejącego,**
  - **jawne wskazanie pozycji – dodatkowe klauzule:**
    - **NULLS FIRST – na początku zbioru rekordów (domyślne dla porządku malejącego),**
    - **NULLS LAST – na końcu zbioru rekordów (domyślne dla porządku rosnącego).**

```
SELECT nazwisko, placa_dod
FROM pracownicy
ORDER BY placa_dod DESC NULLS LAST;
```

Selekcja krotek relacji

- Klauzula WHERE

```
SELECT atrybut1, atrybut2, ... FROM relacja
WHERE [atrybut | wyrażenie] operator [atrybut | wyrażenie | wartość];
```

Operatory

- operatory logiczne:

  =, !=, <>, >, >=, <, <=

```
SELECT nazwisko, placa_pod, etat
FROM pracownicy
WHERE placa_pod > 400;
```

```
SELECT nazwisko, id_zesp
FROM pracownicy
WHERE placa_dod > (placa_pod/10);
```

```
SELECT id_prac, nazwisko, etat
FROM pracownicy
WHERE etat != 'ASYSTENT';
```

Operatory (1)

- operatory SQL:
  - BETWEEN ... AND ... – zawieranie wartości w przedziale,
    - do przedziału wartości zalicza się wartości graniczne, granica dolna musi poprzedzać granicę górną.

```
SELECT nazwisko, placa_pod, etat
FROM pracownicy
WHERE placa_pod BETWEEN 208 AND 1070;
```

  - IN – zawieranie wartości w zbiorze,
    - elementy zbioru muszą być tego samego typu.

```
SELECT nazwisko, placa_pod, id_zesp
FROM pracownicy
WHERE etat IN ('PROFESOR', 'DYREKTOR');
```

Operatory (2)

- **LIKE – zgodność wartości wyrażenia ze wzorcem,**
  - **do tworzenia wzorca wykorzystujemy znaki specjalne % (dowolny ciąg znaków, również pusty) i _ (pojedynczy znak).**

```
SELECT nazwisko, placa_pod, id_zesp
FROM pracownicy
WHERE nazwisko LIKE 'M%';
```

```
SELECT nazwisko, zatrudniony
FROM pracownicy
WHERE nazwisko LIKE 'KOWALSK_';
```

- **IS NULL – czy wartość wyrażenia jest pusta.**

```
SELECT nazwisko, placa_pod
FROM pracownicy
WHERE placa_dod = NULL;
```

```
SELECT nazwisko, placa_pod
FROM pracownicy
WHERE placa_dod IS NULL;
```

Operatory (3)

- **negacje operatorów SQL:**
  - **NOT BETWEEN ... AND ...**
  - **NOT IN**
  - **NOT LIKE**
  - **IS NOT NULL**

```
SELECT nazwisko, placa_pod, id_zesp
FROM pracownicy
WHERE etat NOT IN ('PROFESOR', 'DYREKTOR');
```

```
SELECT nazwisko, etat, placa_pod + NVL(placa_dod,0)
FROM pracownicy
WHERE nazwisko NOT LIKE '%SKI';
```

Warunki złożone klauzuli WHERE (1)

- Operatory logiczne w klauzuli WHERE
  - AND
  - OR
- Tabele wartości logicznych

	PRAWDA	FAŁSZ	NIEOKREŚLONE
NOT	FAŁSZ	PRAWDA	NIEOKREŚLONE

AND	PRAWDA	FAŁSZ	NIEOKREŚLONE
PRAWDA	PRAWDA	FAŁSZ	NIEOKREŚLONE
FAŁSZ	FAŁSZ	FAŁSZ	FAŁSZ
NIEOKREŚLONE	NIEOKREŚLONE	FAŁSZ	NIEOKREŚLONE

OR	PRAWDA	FAŁSZ	NIEOKREŚLONE
PRAWDA	PRAWDA	PRAWDA	PRAWDA
FAŁSZ	PRAWDA	FAŁSZ	NIEOKREŚLONE
NIEOKREŚLONE	PRAWDA	NIEOKREŚLONE	NIEOKREŚLONE

Warunki złożone klauzuli WHERE (2)

- **Operatory logiczne mogą być stosowane jednocześnie w tej samej klauzuli WHERE, przy czym AND posiada wyższy priorytet niż OR, zmiana priorytetu jest możliwa za pomocą nawiasów.**

```
SELECT nazwisko, etat
FROM pracownicy
WHERE placa_pod > 500 AND etat = 'ADIUNKT'
          OR etat = 'ASYSTENT';
```

```
SELECT nazwisko, etat
FROM pracownicy
WHERE placa_pod > 500 AND
          (etat = 'ADIUNKT' OR etat = 'ASYSTENT');
```

Podsumowanie polecenia SELECT

```
SELECT [DISTINCT] { * , kolumna [AS alias], ... }
FROM relacja
WHERE warunek [ AND | OR warunek ... ]
ORDER BY { kolumna, wyrażenie } [ASC | DESC];
```

SELECT	Wybiera listę kolumn
alias	Można stosować tylko do kolumn i wyrażeń (nie do *)
*	Oznacza wszystkie kolumny
DISTINCT, UNIQUE	Eliminuje duplikaty ze zbioru wynikowego
FROM t	Określa relację, z której odczytujemy dane
WHERE	Określa warunki wyboru wierszy, zawiera wartości kolumn, wyrażenia i literały
AND/OR	Łączy warunki w klauzuli WHERE
()	Pozwala na zmianę priorytetu operatorów
ORDER BY	Służy do określenia kryterium sortowania
ASC	Rosnący porządek sortowania
DESC	Malejący porządek sortowania