

## Wstęp do SQL

SQL (Structured Query Language) – strukturalny język zapytań używany do tworzenia, modyfikowania baz danych oraz do umieszczania i pobierania danych z baz danych. Język SQL jest językiem deklaratywnym. Decyzję o sposobie przechowywania i pobrania danych pozostawia się systemowi zarządzania bazą danych (DBMS). W bazach danych Microsoft Access, Libre Base SQL pozwala na wykonanie zapytań szybciej i bardziej efektywnie.

### 1. Podstawowe komendy, funkcje i klauzule:

#### a. Wybieranie danych: SELECT

```
SELECT [DISTINCT]
wyrażenie [AS nazwa_kolumny] [...],
[FROM źródło_elementów]
[WHERE warunek]
[GROUP BY wyrażenie [...]]
[HAVING warunek]
[{UNION | INTERSECT | EXCEPT} instrukcja_SELECT]
[ORDER BY wyrażenie [ASC | DESC]]
```

Np.

- SELECT \* FROM studenci
- SELECT \* FROM oceny  
WHERE pid = '0001'
- SELECT \* FROM studenci NATURAL JOIN oceny
- SELECT \* FROM studenci NATURAL JOIN oceny  
WHERE pid='0017'
- SELECT DISTINCT studenci.nazwisko, studenci.imie, oceny.przedmiot,  
oceny.ocena, oceny.data  
FROM studenci, oceny  
WHERE oceny.przedmiot = 'OISA'  
AND studenci.pid = oceny.pid  
ORDER BY studenci.nazwisko

Klauzula SELECT określa wynikowe kolumny i może zawierać:

- Nazwy kolumn wynikowych
- Wyrażenia zawierające funkcje, nazwy kolumn i stałe
- Stosując AS można nadać wynikowym kolumnom nazwy inne od domyślnych
- Symbol \* oznacza wszystkie kolumny
- Dyrektywę DISTINCT likwidującą powtórzenia wierszy

Aby odwołać się do nazwy kolumny możemy użyć:

`nazwa_tabeli.nazwa_kolumny`

lub

`nazwa_kolumny`

jeśli nazwa kolumny jednoznacznie określa tabelę, z której pochodzi.

## 2. Wyrażenia arytmetyczne:

- Operatory: +, -, \*, /
  - Funkcje matematyczne: ABS(x), POWER(x,y), SQRT(x), RAND() – generuje liczbę losową z zakresu od 0 do 1
  - Funkcje zaokrąglania liczb:
    - a) CEILING(x) – funkcja zaokrągla swój argument w górę do najbliższej liczby całkowitej (Oracle – CEIL(x))
    - b) FLOOR(x) – funkcja zaokrągla swój argument w dół do najbliższej liczby całkowitej
    - c) ROUND(x,n) – zaokrągla liczbę x do n miejsc po przecinku
- Jeśli w wyrażeniu matematycznym choć jeden z argumentów jest NULL, to wynik tego wyrażenia zawsze jest NULL

## 3. Wyrażenia tekstowe:

- Operatory: +, &  
Np. 'tekst' + NULL = NULL      lub      'tekst' & NULL = 'tekst'
- LOWER(x), UPPER(x)
- CHARACTER\_LENGTH(x), LENGTH(x)
- SUBSTRING(x FROM n FOR m)
- POSITION(x1 IN x)
- TRIM(ltb wzorzec FROM napis)  
ltb – LEADING, TRAILING, BOTH

## 4. Konwersja typów

- CAST(wyrażenie AS typ\_danych)

Np. CAST(ocena AS NUMERIC(2, 1))

```
SELECT ocena,
CASE
    WHEN ocena > 4 THEN 'gratuluję'
    WHEN ocena > 3 THEN 'dobrze'
    WHEN ocena > 2 THEN 'nienajlepiej'
    WHEN ocena = 2 THEN 'może następnym razem'
    END AS "komentarz 1",
CASE ocena
    WHEN 5 THEN 'gratuluję'
    WHEN 4.5 THEN 'też gratuluję'
    WHEN 4 THEN 'dobrze'
    WHEN 3.5 THEN 'no średnio'
    WHEN 3 THEN 'nienajlepiej'
    ELSE 'może następnym razem'
    END AS "komentarz 2"
FROM oceny
```

## 5. Funkcje agregujące:

Wyznaczają jedną wartość z grupy wartości w kolumnie – grupy wierszy określa klauzula GROUP BY

- a) COUNT – zlicz ilość wszystkich wartości (w kolumnie) różnych od NULL
  - i. COUNT(\*) – zlicza wszystkie wiersze
- b) AVG – oblicza średnią ze wszystkich wartości
- c) SUM – sumuje wartości
- d) MAX – zwraca wartość największą
- e) MIN – zwraca wartość najmniejszą

## 6. Klauzula FROM

- Klauzula FROM zawiera „wyrażenie tabelowe” określające źródło danych dla zapytania
- Wyrażenie tabelowe klauzuli FROM może zawierać listę tabel i perspektyw źródłowych albo np. A, B – oznacza iloczyn kartezjański (złączenie krzyżowe) tabel A i B  
A, B = A CROSS JOIN B
- Złączenia tabel:
  - A NATURAL [typ] JOIN B – złączenie naturalne (po kolumnach o tych samych nazwach w obu tabelach)
  - A [typ] JOIN B ON warunek
  - A [typ] JOIN B USING (kolumna [,...])
  - Typy złączeń zewnętrznych:  
{LEFT | RIGHT | FULL} [OUTER]
  - Podzapytania – umieszczone w nawiasach i z nadanymi aliasami
- Tabelom można nadać zastępcze nazwy nazywane: aliasami, nazwami skorelowanymi i zmiennymi zakresu
  - format: nazwa\_tabeli\_źródłowej [AS] alias
  - nadawanie aliasów tabelom źródłowym stosuje się przy „samozłączeniach” tabel
  - alias trzeba nadać wynikowi podzapytania, które może być użyte w zastępstwie tabeli
- np.  
select  
"PRACOWNICY"."FIRST\_NAME" as "Imię",  
"PRACOWNICY"."LAST\_NAME" as "Nazwisko",  
"SZELOWIE"."FIRST\_NAME" as "Imię(szeł)",  
"SZELOWIE"."LAST\_NAME" as "Nazwisko(szeł)"  
from  
"EMPLOYEES" "PRACOWNICY" LEFT JOIN  
"EMPLOYEES" "SZELOWIE" ON  
"PRACOWNICY"."MANAGER\_ID"="SZELOWIE"."EMPLOYEE\_ID"

## 7. *Klauzula WHERE:*

- Zawiera warunek selekcji wierszy tabeli wynikowej nazywany często wyrażeniem ograniczającym
- Warunek (predykat) jest wyrażeniem logicznym (funkcją zdaniową) – w tabeli wynikowej znajdują się tylko wiersze spełniające warunek
- Predykat może zawierać: nazwy kolumn, wyrażenia logiczne, operatory porównań, funkcje i zagnieżdżone instrukcje SELECT (podzapytania)
- Predykaty mogą mieć wartość logiczną TRUE, FALSE lub NULL
- Na logicznym rachunku predykatów opiera się rachunek relacyjny – w przypadku języka SQL jest to rachunek relacyjny na krotkach

Predykaty:

Porównania: =, <, >, <=, >=, <>

a >= b; a < b; itp.

BETWEEN: x BETWEEN 23 AND 143    x >= 23 AND x <= 143

IN: x IN (1, 2, 3); x IN (SELECT y FROM tabela\_A)

LIKE: nazwisko LIKE 'Kowal%'

'\_' – zastępuje dowolny pojedynczy znak

'%' – zastępuje ciąg pusty lub składający się z dowolnej ilości znaków

W programie MS Access znaki '\_', '%' są zastąpione przez '?', '\*\*'

NULL: x IS NULL; y IS NOT NULL

EXISTS:

EXISTS (SELECT \* FROM zakupy WHERE id\_klienta='0123')

OVERLAPS

## 8. *Klauzule GROUP BY i HAVING:*

- Klauzula GROUP BY jest używana do definiowania grup wyjściowych wierszy
- Dla grup wierszy można używać funkcji agregujących (COUNT, MIN, MAX, SUM, AVG)
- Klauzula HAVING (podobnie jak WHERE) odrzuca grupy wierszy nie spełniające warunku (predykatu)

Np.

```
SELECT studenci.nazwisko || studenci.imie AS student, srednie."średnia"
```

```
FROM
```

```
(SELECT oceny.id_studenta, AVG(oceny.ocena) AS "średnia"
```

```
FROM oceny
```

```
GROUP BY oceny.id_studenta) srednie
```

```
NATURAL JOIN studenci
```

## 9. *UNION, INTERSECT, EXCEPT, ORDER BY*

- UNION – operator sumowania wyników dwóch instrukcji SELECT
- INTERSECT – operator przecięcia wyników dwóch instrukcji SELECT
- EXCEPT – operator różnicy wyników dwóch instrukcji SELECT
- ORDER BY – klauzula określająca kryterium sortowania
  - ASC – oznacza kolejność sortowania rosnącą (domyślna)
  - DESC – oznacza kolejność sortowania malejącą

Np.  
SELECT 'student' AS funkcja, nazwisko AS "Nazwisko", imie AS "Imię"  
FROM studenci  
UNION  
SELECT "Stopien\_naukowy", "Nazwisko", "Imie"  
FROM wykadowcy  
ORDER BY 2

#### 10. Aktualizacja danych

- UPDATE nazwa\_tabeli SET nazwa\_kolumny = wyrażenie [...]  
WHERE wyrażenie\_ograniczające

Np.  
• UPDATE płace SET płaca = płaca \* 2 WHERE nazwisko = 'Kowalski'  
• UPDATE płace SET płaca = płaca + 10 WHERE nazwisko LIKE 'Kow%'

Rodzaj zgodności

Wzorzec

- Zgodność (zwraca True)
- Brak zgodności (zwraca False)
- Wiele znaków  
a\*a aa, aBa, aBBB aBC  
\*ab\* abc, AABb, Xa
- Znak specjalny  
a[\*]a a\*a
- Wiele znaków  
ab\* abaaa aba ab
- Jeden znak  
a?a aaa, a3a, aBa
- Jedna cyfra  
a#a a0a, a1a, a2a
- Zakres znaków  
[a & z] f, p, j 2, &
- Poza zakresem  
[!a & z] 9, &, % b, a
- Nie cyfra  
[!0 & 9] A, a, &, ~ 0, 1, 9
- Połączenie  
a[!b & m]# An9, az0, a99 abc, aj0

## ZADANIA

### Otwórz bazę danych Cukiernia

Wyświetl zawartość tabeli:

- a) Wyświetl wszystkie rekordy z tabeli Czekoladki (wszystkie atrybuty).
- b) Wyświetl dane (wszystkie atrybuty) wszystkich klientów.
- c) Wyświetl wszystkie rekordy z tabeli Klienci (wszystkie atrybuty) posortowane alfabetycznie według miasta, a następnie według nazwiska.
- d) Wyświetl wszystkie rekordy z tabeli Czekoladki (wszystkie atrybuty) posortowane malejąco według ceny jednostkowej (koszt).
- e) Wyświetl nazwę, opis i cenę poszczególnych opakowań czekoladek posortowane alfabetycznie według nazwy i ceny.
- f) Wypisz pierwsze 4 pudełka czekoladek i ich opis.
- g) Wyświetl wszystkie informacje na temat 5 najdroższych opakowań Czekoladek.
- h) Wypisz pierwsze 25% wszystkich zamówień.
- i) Wyświetl wszystkie Pudełka czekoladek, których stan magazynowy jest mniejszy niż 400.
- j) Wyświetl nazwy, adresy Klientów z Katowic posortowane według Nazwy Klientów.
- k) Wyświetl wszystkie czekoladki których koszt jest między 30 a 40 groszy.
- l) Wyświetl informacje na temat wszystkich klientów , których nazwisko ma przedostatnią literę ó.
- m) Wypisz wszystkie miasta, z których pochodzą klienci (tak aby nazwa miasta się nie powtarzała).