



BAZY DANYCH II

Wprowadzenie do SQL

mgr inż. Patryk Kaczmarek

patryk.kaczmarek@ansleszno.pl

Plan laboratorium

- Polecenie SELECT
- Klauzula WHERE
- Operatory SQL
- Sortowanie ORDER BY
- Funkcje znakowe
- Funkcje liczbowe
- Funkcje dat
- Wyrażenie CASE
- Funkcje grupowe

Projekcja – klauzula SELECT

Wybór wartości określonych atrybutów relacji

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

```
SELECT * FROM nazwa_tabeli;
```

Wyrażenie arytmetyczne: - + / *

```
SELECT nazwisko, placapod+100 FROM  
pracownicy;
```

Aliasy nazw atrybutów relacji

Alias to alternatywna nazwa atrybutu, można z niego korzystać podczas prezentacji wyników

```
SELECT placapod*12 AS roczna FROM pracownicy;
```

Konkatenacja

```
SELECT 'Pracownik ' + [nazwisko] + ' zarabia ' +[placapod] FROM  
pracownicy;
```

```
SELECT CONCAT('Pracownik ',nazwisko,' zarabia ',placapod) FROM  
pracownicy;
```

Obsługa wartości pustych

NULL – wartość niedostępna, nieprzypisana

```
SELECT placapod*12 + placadod FROM pracownicy;
```

```
SELECT placapod*12 + ISNULL(placadod,0) FROM pracownicy;
```

Eliminacja duplikatów - DISTINCT

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

Porządkowanie wyników zapytania

ORDER BY kolejność sortowania :

- ASC (ascending- rosnąco) lub
- DESC (descending- malejąco)

ORDER BY występuje zawsze jako ostatnia klauzula zapytania

porządek sortowania:

- liczby – od mniejszych do większych
- daty – od wcześniejszych do późniejszych
- łańcuchy znaków – alfabetycznie

```
SELECT nazwisko, etat FROM pracownicy  
ORDER BY etat DESC;
```

Selekcja krotek

Klauzula WHERE

```
SELECT * FROM pracownicy WHERE atrybut OPERATOR wartość;
```

Operatory logiczne

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

```
SELECT nazwisko FROM pracownicy WHERE placapod > 1500;
```

Operatory SQL

BETWEEN ... AND...

do przedziału wartości zalicza się wartości graniczne,
granica dolna musi poprzedzać granicę górną

```
SELECT nazwisko FROM pracownicy WHERE placapod BETWEEN 500  
AND 1500;
```

Operatory SQL

IN

jeżeli w zbiorze znajdują się dane znakowe bądź daty to należy je ująć w apostrofy

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

Operatory SQL

LIKE

do tworzenia wzorca wykorzystujemy znaki specjalne %
(dowolny ciąg znaków) i _ (pojedynczy znak)

```
SELECT nazwisko FROM pracownicy WHERE nazwisko like 'K%';
```

Funkcje znakowe

- **LOWER(atr)** – zamienia na małe litery
- **UPPER(atr)** - zamienia na duże litery

```
SELECT LOWER(nazwisko) FROM pracownicy;
```

Funkcje znakowe

- **SUBSTRING(atr,start,ile)** – wycina ciąg od pozycji start, ile – tyle znaków

```
SELECT SUBSTRING(nazwisko,1,3) FROM pracownicy;
```

- **LEN(atr)** - zwraca długość ciągu

```
SELECT LEN(nazwisko) FROM pracownicy;
```

Funkcje znakowe

- **REPLACE (atr,co,na)** – zamienia w ciągu atr znak „co” na „na”

```
SELECT REPLACE(nazwisko,'a','b') FROM pracownicy;
```

Funkcje liczbowe

- **ROUND (atr,miejsc)** – zaokrągla liczby

```
SELECT ROUND(placapod,2) FROM pracownicy;
```

- **CEILING(atr)** - w górę do liczby całkowitej
- **FLOOR(atr)** – w dół do liczby całkowitej

```
SELECT CEILING(placapod) FROM pracownicy;
```

Funkcje liczbowe

- **POWER (liczba,potega)** – podnosi liczbę do potęgi
- **SQRT(liczba)** – pierwiastek kwadratowy
- **ABS(liczba)** – wartość bezwzględna
- **SIGN(liczba)** – zwraca -1 dla liczb mniejszych od zera 0 – dla zera 1 dla większych od zera

```
SELECT POWER(2,2);
```

Funkcje dat

- **GETDATE ()** – zwraca aktualną datę i godzinę

```
SELECT GETDATE();
```

- **DATEDIFF(co, start,koniec)** – zwraca różnice w określonym formacie

co – mm : miesiąc, yy : rok, dd : dzień

```
SELECT DATEDIFF(mm,'2016-01-01','2016-12-01');
```

Funkcje dat

- GETDATE () – modyfikacje
- **DAY(GETDATE())** – zwraca dzień
- **MONTH(GETDATE())** – zwraca miesiąc
- **YEAR(GETDATE())** – zwraca rok

```
SELECT DAY(GETDATE());
```

- **DATEANAME(month,GETDATE())** – zwraca nazwę miesiąca, lub dnia tygodnia itp

Wyrażenie CASE

CASE WHEN warunek **THEN** wynik **ELSE** innyWynik
END

```
SELECT JakisWynik = (CASE WHEN symbol > 0 THEN 0 ELSE 1 END);
```

FUNKCJE GRUPOWE

Bazują na podzbiorach nazywanych grupami, operują na zbiorach wielu krotek

- **AVG(atr)** – średnia
- **COUNT(atr)** – licznik
- **SUM(atr)** – suma
- **MAX(atr)** – maksymalna wartość
- **MIN(atr)** – minimalna wartość

FUNKCJE GRUPOWE

GROUP BY atrybut

	NAZWISKO	ID_ZESP
1	Dolny	NULL
2	Marecki	10
3	Makowski	10
4	Kotarski	20
5	Nowak	20
6	Kowalski	20
7	Grzybowska	20
8	Krakowska	20
9	Opolski	20
10	Siekierski	20
11	Nowicki	30
12	Przywarek	30
13	Kotlarczyk	30

```
SELECT ID_ZESP FROM PRACOWNICY GROUP BY ID_ZESP
```

FUNKCJE GRUPOWE

UWAGA

Na liście atrybutów mogą być tylko funkcje grupowe (agregujące) i atrybuty grupujące.

```
SELECT NAZWISKO, ID_ZESP FROM PRACOWNICY GROUP BY ID_ZESP
```

Spowoduje błąd

```
SELECT ID_ZESP FROM PRACOWNICY GROUP BY ID_ZESP;
```

OK

FUNKCJE GRUPOWE

HAVING – pozwala na wybór grup spełniających określone warunki

```
SELECT ID_ZESP FROM PRACOWNICY GROUP BY ID_ZESP HAVING ID_ZESP > 30;
```


Microsoft SQL Server Management



SQL Server Management Studio

Aplikacja komputerowa

Logowanie do BD laboratorium



Microsoft SQL Server 2012

Server type: Database Engine

Server name: db.labs.local

Authentication: SQL Server Authentication

Login: student

Password: *****

Remember password

Connect Cancel Help Options >>

db.labs.local

student

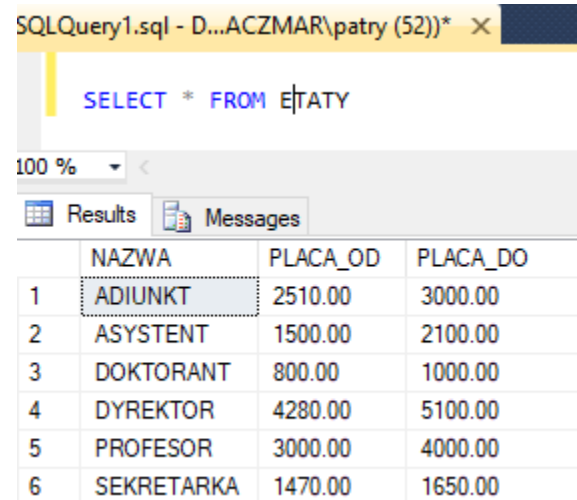
student

Przykład sprawozdania

- 1. Wyświetl wszystkie dane z tabeli:

Polecenie: **SELECT * FROM ETATY;**

Zrzut ekranu:



SQLQuery1.sql - D...ACZMAR\patry (52))* ×

```
SELECT * FROM ETATY
```

100 % <

Results Messages

	NAZWA	PLACA_OD	PLACA_DO
1	ADIUNKT	2510.00	3000.00
2	ASYSTENT	1500.00	2100.00
3	DOKTORANT	800.00	1000.00
4	DYREKTOR	4280.00	5100.00
5	PROFESOR	3000.00	4000.00
6	SEKRETARKA	1470.00	1650.00

Do każdego zadania w sprawozdaniu dołączamy 2 elementy: użyte polecenia (tekst) oraz zrzut ekranu dokumentujący wykonanie polecenia