



UNIWERSYTET IM. ADAMA MICKIEWICZA W POZNANIU

---

# ***Wprowadzenie do projektowania i wykorzystania baz danych***

*Katarzyna Klessa*



## Informacje organizacyjne

### 1. Dyżury **II semestr**:

wtorek 11:50-12:40, piątek 13:20-14:00

pokój / room 312aB Coll. Novum

*- proszę o indywidualne umawianie się na dyżury*

### 2. Warunki zaliczenia przedmiotu:

**semestr I** - test zaliczeniowy zaliczony do końca sesji zimowej (w przypadku r. 2014/15 - luty 2015)

**semestr II** - wykonanie ćwiczeń zaliczeniowych i egzamin końcowy (egz. praktyczny, z wykorzystaniem komputera)



## Zapytanie SELECT: wyświetlanie wartości bez użycia tabel

Polecenia SELECT można użyć, aby wyświetlić wybraną wartość lub wyrażenie, wynik funkcji, które nie są pobierane z tabel w bazie danych, np. funkcje matematyczne.

```
SELECT 2+2
```

```
SELECT 2-2
```

```
SELECT 2*2
```

```
SELECT 2/2
```

```
SELECT 2*2*2, POWER(2,3)
```



## Jeszcze o Select: wybieranie danych innych niż te w tabelach

Polecenia SELECT można użyć, aby wyświetlić wybraną wartość lub wyrażenie, wynik funkcji, które nie są pobierane z tabel w bazie danych, np. funkcje matematyczne.

```
SELECT 2+2
```

```
SELECT 2-2
```

```
SELECT 2*2
```

```
SELECT 2/2
```

```
SELECT 2*2*2, POWER(2,3)
```



## Jeszcze o Select: wybieranie danych innych niż te w tabelach. Funkcje GETDATE, DATEADD

### GETDATE

Zwraca aktualną datę serwera.

```
select GETDATE()
```



## Jeszcze o Select: wybieranie danych innych niż te w tabelach. Funkcje GETDATE, DATEADD

Funkcja dodaje lub odejmuje określone części daty (zob. też: [http://www.w3schools.com/sql/func\\_dateadd.asp](http://www.w3schools.com/sql/func_dateadd.asp)). Pierwszy parametr określa, którą część daty chcemy zmienić (dodać od niego lub odjąć). Drugi parametr informuje, o ile chcemy zmienić. Trzeci parametr to data wejściowa, którą będziemy modyfikować.

```
select DATEADD(YEAR, -21, '2015-02-22')
```

```
select DATEADD(YEAR, -21, '2015-02-22 12:37')
```

```
select DATEADD(YEAR, -21, GETDATE())
```



## Jeszcze o Select: wybieranie danych innych niż te w tabelach. Funkcje GETDATE, DATEADD

100 % <

Results Messages

	(No column name)
1	1994-02-22 00:00:00.000
1	1994-02-22 12:37:00.000
1	1994-02-25 10:42:31.630

✓ Query executed successfully.



## Zapisywanie danych do pliku

```
/*WHERE*/
```

```
SELECT * FROM student WHERE BirthDate < '1994-01-01'
```

100 %

Results Messages

	StudentID	BookNumber	FirstName	LastName	BirthDate	Sex	Faculty	Specializati...	AverageNote
1	1	fa622ca7-bc0b-4ca7-9d78-fd8572bb197c	Andrzej	Kowalski	1985-05-21	male	English	NULL	4,127242
2	2	8d37620c-3fbb-402d-b98d-db75b71b49de	Justyna	Stefańska	1993-06-24	female	English	NULL	2,52235
3	3	b112b500-72c8-4029-a98a-65f9bab64dd9	Marek	Andrzejewski	1992-06-19	male	Modern Languages and Literatures	NULL	3,505093
4	4	807f0c01-f33d-4979-9834-69df9e92090c	Jan					NULL	4,015246
5	5	a830cd03-25e0-4e3b-a988-44c607fcd6a8	Mag				Languages and Literatures	NULL	3,164437
6	6	d0b5f140-8f6d-49e8-b77d-960806cef30e	Jac				Languages and Literatures	NULL	3,547136
7	7	e5fb6031-0bfd-4743-a5df-497119bc6b1c	Mor					NULL	3,806427
8	8	bfd3f33-4c97-4781-a867-20e0a96e7cff	Woj				Languages and Literatures	NULL	3,155375
9	9	595264c5-34d6-443f-ae77-93ec777481ad	Mar					NULL	3,845906
10	10	e5bd1c09-5338-4964-863e-b9130c869b70	Mor				Languages and Literatures	NULL	3,491307
11	13	f2b6eddf-da34-40d5-b5e7-1110d07285ba	And				Languages and Literatures	NULL	2,069869
12	14	f3cd9ae1-eeb1-4891-9151-abd4607d3b31	Kat				Languages and Literatures	NULL	3,624685
13	15	84b6b4e1-cc5b-424e-821f-b6cf64e93c05	Agnieszka	Steranska	1986-03-24	female	Modern Languages and Literatures	NULL	2,356331
14	16	116929d0-c7b6-438d-a7ae-dcfbcb30d335	Patrycja	Kowalewska	1987-09-09	female	Modern Languages and Literatures	NULL	3,297905
15	18	0f2c2eae-af26-4d84-86c4-7ba80547936c	Andrzej	Kowalski	1987-11-24	male	Modern Languages and Literatures	NULL	3,408858
16	20	9372d363-760d-48d6-b212-2b504453e5df	Anna	Węglarczyk	1986-11-03	female	English	NULL	3,939689
17	22	a5bcf7a7-f60d-41f4-aa5b-638200d28eb9	Magdalena	Węglarczyk	1993-02-05	female	Modern Languages and Literatures	NULL	3,111405
18	23	bf5fa6bd-e9ab-4b63-bc09-ac2accd59978	Patrycja	Piasecka	1985-04-15	female	English	NULL	3,411335
19	24	6b00354f-d31d-45d1-8cda-6e945630e29c	Patrycja	Węglarczyk	1987-10-17	female	Modern Languages and Literatures	NULL	4,711864

Query executed successfully.

Lenovo-Kasia\SQLXPRESS (12...





## Klauzula WHERE: wstęp

Klauzula WHERE pozwala wybrać na podstawie określonych warunków, które wiersze powinny być wyświetlone za pomocą polecenia SELECT.

Za wyrażeniem WHERE podajemy warunki logiczne, które muszą być spełnione, aby wiersz pojawił się jako wynik zapytania.

Warto zapamiętać, że WHERE jest używane również w innych zapytaniach, nie tylko w SELECT, np. w INSERT, UPDATE i DELETE, o których powiemy później.



## Klauzula WHERE: składnia

```
SELECT [TOP LICZBA] KolumnyDoPokazania [AS  
NowaNazwaKolumny]  
FROM Tabela [AS NowaNazwaTabeli]  
[WHERE WarunkiFiltrowania]  
[GROUP BY KolumnyPoKtórychGrupować]  
[ORDER BY KolumnyPoKtórychSortować]  
[HAVING WarunkiFiltrowania]
```

[] - opcjonalne, czyli można to pominąć



## Klauzula WHERE

Chcemy zobaczyć studentów, którzy mają 21 lat lub więcej (i np. mogą kupować alkohol i papierosy w USA). Mamy rok 2015, więc zakładamy, że data graniczna to rok 1994, a więc więcej niż 21 lat będą miały osoby urodzone przed rokiem 1994. Piszemy zapytanie SELECT, a we frazie WHERE określamy, że ma być wybrany każdy kto urodził się wcześniej niż 1 stycznia 1994.

```
select * from student where BirthDate < '1994-01-01'
```



## Klauzula WHERE

Chcemy zobaczyć studentów, którzy mają 21 lat lub więcej (i np. mogą kupować alkohol i papierosy w USA). Mamy rok 2015, więc zakładamy, że data graniczna to rok 1994, a więc więcej niż 21 lat będą miały osoby urodzone przed rokiem 1994. Piszemy zapytanie SELECT, a we frazie WHERE określamy, że ma być wybrany każdy kto urodził się wcześniej niż 1 stycznia 1994.

```
select * from student where BirthDate < '1994-01-01'
```

	Student...	BookNumber	FirstName	LastName	BirthDate	Sex	Faculty	Specializati...	AverageN...
1	1	fa622ca7-bc0b-...	Andrzej	Kowalski	1985-05-21	male	English	NULL	4,127242
2	2	8d37620c-3fbb-...	Justyna	Stefańska	1993-06-24	female	English	NULL	2,52235
3	3	b112b500-72c8...	Marek	Andrzeje...	1992-06-19	male	Modern L...	NULL	3,505093



## Klauzula WHERE

Chcemy zobaczyć studentów, którzy mają 21 lat lub więcej (i np. mogą kupować alkohol i papierosy w USA). Mamy rok 2015, więc zakładamy, że data graniczna to rok 1994, a więc więcej niż 21 lat będą miały osoby urodzone przed rokiem 1994. Piszemy zapytanie SELECT, a we frazie WHERE określamy, że ma być wybrany każdy kto urodził się wcześniej niż 1 stycznia 1994.

```
select * from student where BirthDate < '1994-01-01'
```



## Klauzula WHERE

W efekcie otrzymaliśmy dane ludzi urodzonych wcześniej niż 1 stycznia 1994. Niestety nie jest to lista ludzi, którzy na dzień dzisiejszy mają skończone 21 lat, bo będzie to zależne od dzisiejszej daty. Czyli prawidłowe wyniki to tylko dane tych osób, które urodziły się do dnia dzisiejszego włącznie w roku 1994, jednak osoby urodzone później nie będą miały jeszcze ukończonych 21 lat. Czyli osoba, która urodziła się 1 lutego 1994 na dzisiaj (22 luty 2015) ma dzisiaj skończone 21 lat, a osoba która urodziła się 1 października – jeszcze nie ma 21 lat.

**Jak poprawić zapytanie?**



## Klauzula WHERE

Aby poprawić nasze zapytanie spróbujmy użyć funkcji GETDATE i DATEADD.

Wybieramy studentów, którzy urodzili się wcześniej niż dzisiejsza data minus 21 lat.

```
select * from Student where BirthDate <  
DATEADD(YEAR, -21, GETDATE())
```



## Klauzula WHERE: porównanie (=)

```
SELECT FirstName, LastName FROM Student WHERE  
Faculty = 'English'
```

Zapytanie zwróci listę wszystkich studentów Wydziału Anglistyki.





## Klauzula WHERE: różnica (<>)

```
SELECT FirstName, LastName, Faculty FROM Student  
WHERE Faculty <> 'English'
```

Zapytanie zwróci listę wszystkich studentów wydziałów innych niż Wydział Anglistyki.



## Klauzula WHERE: iloczyn (cz. wspólna zbiorów)

```
SELECT FirstName, LastName, Sex, Faculty FROM  
Student WHERE Sex = 'female' AND Faculty =  
'English'
```

Zapytanie zwróci listę studentek Wydziału Anglistyki, oba warunki muszą być spełnione.



## Klauzula WHERE: suma zbiorów

```
SELECT FirstName, LastName, Faculty FROM Student  
WHERE Faculty = 'English' OR Faculty = 'Modern  
Languages and Literatures'
```

Zapytanie zwróci listę wszystkich studentów W. Anglistyki i W. Neofilologii.



## Klauzula WHERE: operator IN

```
SELECT FirstName, LastName, Sex, Faculty FROM  
Student  
WHERE FirstName IN ( 'Monika', 'Anna', 'Jacek',  
'Patrycja' )
```

Zapytanie zwróci listę studentów o imionach określonych w nawiasie, po operatorze IN.



## Klauzula WHERE: większy - mniejszy

```
SELECT FirstName, LastName, Faculty, AverageNote  
FROM Student
```

```
WHERE AverageNote > 4.3 AND AverageNote < 4.6
```

Zapytanie zwróci listę studentów o średniej wyższej niż 4,3 i niższej niż 4,6.

**\* Ten sam efekt można uzyskać, używając operatora BETWEEN:**

```
SELECT FirstName, LastName, Faculty, AverageNote  
FROM Student
```

```
WHERE AverageNote BETWEEN 4.3 AND 4.6
```



## Klauzula WHERE: warunki złożone

```
SELECT FirstName, LastName, Sex, Faculty FROM  
Student  
WHERE (Sex = 'female' AND Faculty = 'English') OR  
(Sex = 'male' AND Faculty = 'Modern Languages and  
Literatures')
```

Zapytanie zwróci listę pań z Anglistyki i panów z Neofilologii.



**Dziękuję za uwagę!**

