

Temat: Obliczenia w programie MS Access.

1. Operatory działań matematycznych w programie Access:

Operator...	i jego działanie:
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
^	potęgowanie
\	dzielenie całkowite
Mod	reszta z dzielenia całkowitego

Ćwiczenie

Przepisz do zeszytu temat i przerysuj tabelkę.

Ćwiczenie 1

1. Utwórz nową bazę danych – **Obliczenia.accdb**:

- utwórz tabelę z dwoma polami o nazwach **a** i **b**,
- **Typ danych: Liczba**,
- we **Właściwościach pola** ustal **Rozmiar pola: Podwójna precyzja**; to zapewni swobodne używanie liczb rzeczywistych,
- zapisz tabelę pod nazwą **Liczby**,
- w **Widoku arkusza danych** wprowadź liczby:

a:	1	3	1,5	4,25	1
b:	0	-2	2,75	-1,125	2
- utwórz kwerendę – przenieś do niej oba pola **a** i **b**, zapisz ją pod nazwą **K_Działania**,
- w siatce projektowania wpisz cztery podstawowe działania: **a+b**, **a-b**, **a*b**, **a/b** i uruchom kwerendę

a	b	a+b	a-b	a*b	a/b
1	0	1	1	0	#Błąd
3	-2	1	5	-6	-1,5
1,5	2,75	4,25	-1,25	4,125	0,545454545454545
4,25	-1,125	3,125	5,375	-4,78125	-3,77777777777778
1	2	3	-1	2	0,5

suma różnica iloczyn iloraz

- w nagłówkach kolumn wyświetlanego arkusza pojawiły się nazwy **Wyr1:**, **Wyr2:** itd.
 - jeżeli w wierszu Pole siatki projektowania kwerendy umieszczamy coś, co nie jest nazwą pola tabeli źródłowej, wówczas powinniśmy poprzedzić to nazwą zakończoną dwukropkiem; taka nazwa jest wówczas wyświetlana w wynikowym arkuszu, jako nagłówek kolumny; jeśli tego nie zrobimy – komputer samodzielnie dopisze nazwy: **Wyr1:**, **Wyr2:** itd.
 - wróć do fazy projektu i skoryguj odpowiednio wpisy: **Wyr1:** suma
Wyr2: różnica
Wyr3: iloczyn
Wyr4: iloraz.
2. Utwórz kwerendę obliczającą kolejne potęgi liczby 2. Zaprojektuj najpierw odpowiednią tabelę (dwa pola **a** i **b** typu liczba; dla pola **a** ustaw wartość domyślną **2**). Zapisz kwerendę pod nazwą **K_Potęgi**.
3. Utwórz kwerendę obliczającą pole i objętość sześcianu o danym boku **a**. Zaprojektuj najpierw odpowiednią tabelę. Zapisz kwerendę pod nazwą **K_Sześcian**.
4. Utwórz kwerendę obliczającą pole i objętość prostopadłościanu o danych bokach **a**, **b** i **c**. Zaprojektuj najpierw odpowiednią tabelę. Zapisz kwerendę pod nazwą **K_Prostopadłościan**.
5. Utwórz kwerendę wykorzystującą dzielenie całkowite do sprawdzenia czy dana liczba dzieli się np. przez 3. Zaprojektuj najpierw odpowiednią tabelę. Zapisz kwerendę pod nazwą **K_Dzielenie**.

2. Funkcje matematyczne MS Access (przepisz funkcje do zeszytu lub wydrukuj je i wklej).

ABS(liczba) – funkcja zwraca wartość bezwzględną argumentu liczba.

ATN(liczba) – funkcja zwraca wartość równą argusowi tangensowi argumentu liczba.

COS(liczba) – funkcja zwraca wartość równą cosinusowi argumentu liczba (argument liczba wyraża miarę kąta w radianach).

EXP(liczba) – funkcja zwraca wartość równą stałej e (podstawa logarytmu naturalnego) podniesionej do potęgi argumentu liczba.

FIX(liczba) – Funkcja zwraca część całkowitą argumentu liczba. Dla wartości ujemnych funkcja Fix zwraca najbliższą liczbę całkowitą ujemną. Np. Fix(-3,4)= -3

INT(liczba) – funkcja zwraca część całkowitą argumentu liczba. Dla wartości ujemnych funkcja Int zwraca najbliższą liczbę całkowitą ujemną nie większą od podanej liczby. Np. (Int(-3,4)) = -4.

LOG(liczba) – funkcja zwraca wartość równą logarytmowi naturalnemu argumentu liczba. Podstawą logarytmu naturalnego jest liczba $e \approx 2,718282$. Aby otrzymać logarytm liczby o dowolnej podstawie a, należy podzielić wartości logarytmu naturalnego tej liczby przez logarytm naturalny z a, czyli: $\text{Log}_a(x) = \text{Log}(x) / \text{Log}(a)$

RND[(liczba)] – funkcja zwraca liczbę losową z przedziału [0; 1]

SGN(liczba) – funkcja zwraca wartość typu Integer reprezentującą znak liczby.

Jeśli:	Funkcja Sgn zwraca
liczba > 0	1
liczba = 0	0
liczba < 0	-1

SIN(liczba) – funkcja zwraca wartość odpowiadającą sinusowi argumentu liczba (argument liczba wyraża miarę kąta w radianach).

SQR(liczba) – funkcja zwraca wartość równą pierwiastkowi kwadratowemu argumentu liczba (argument musi być liczbą nieujemną).

TAN(liczba) – funkcja zwraca wartość równą tangensowi argumentu liczba (argument liczba wyraża miarę kąta w radianach).

Ćwiczenie 2

Utwórz kwerendę na bazie tabeli **Liczby** wykorzystującą funkcje matematyczne:

abs(b): Abs([b])

sgn(b): Sgn([b])

int(b): Int([b])

rnd(b): Rnd([b])

sqr(b): Sqr([b])

Zapisz ją pod nazwą **Funkcje matematyczne**.

Ćwiczenie 3

Utwórz kwerendę na bazie tabeli **Liczby** obliczającą funkcje trygonometryczne dla pola a. Zapisz ją pod nazwą **Funkcje trygonometryczne**.

Ćwiczenie 4

Utwórz kwerendę obliczającą deltę i pierwiastki równań kwadratowych. Zaprojektuj najpierw tabelę z polami a, b, c (Tabela4). Zapisz kwerendę pod nazwą **Pierwiastki**.

a	b	c	delta	x1	x2
-1	0	4	16	2	-2
1	2	-3	16	-3	1
1	5	-6	49	-6	1
1	4	4	0	-2	-2

3. Operatory Logiczne: (przerysuj do zeszytu tabelkę oraz przepisz uwagę.)

Operator...	i jego działanie	
And	koniunkcja	$a \text{ i } b$
Imp	implikacja	$z \text{ a wynika } b$
Or	alternatywa	$a \text{ lub } b$
Eqv	równoważność	$a \text{ równoważne } b$
Not	negacja	$\text{nie } a$
Xor	alternatywa wykluczająca	$a \text{ albo } b$

Ćwiczenie 5

W Widoku Projektu utwórz tabelę **PQ** zawierającą dwa pola: **p** i **q**:

p, q – wybierz typ danych: **Tak/Nie**, we **Właściwościach pola** nadaj im format: **Prawda/Falsz**.

Wypełnij tabelę danymi jak w polach p i q poniżej:

p	q
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

☒ - odpowiada wartości logicznej Prawda

☐ - odpowiada wartości logicznej Falsz

W widoku projektu utwórz nową kwerendę, jako źródło danych do kwerendy wykorzystaj tabelę **PQ**. Przenieś z tabeli do siatki projektowania dwa pola: p i q, następnie w kolejnych wolnych pozycjach wiersza **Pole** siatki projektowania wpisz:

Pole:	p	q	nie p: Not[p]	nie q: Not[q]	p i q: [p] And [q]	p równoważne q: [p] Eqv [q]	z p wynika q: [p] Imp [q]	p lub q: [p] Or [q]	p albo q: [p] Xor [q]
Sortuj:									
Pokaż:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kryteria:									
lub:									

Zapisz kwerendę pod nazwą **Operatory logiczne** i uruchom ją.

W obliczanych polach pojawiły się liczby **0** i **-1** zamiast napisów **Prawda** czy **Falsz**. Otóż wartości logicznej **Falsz** odpowiada liczba **0**, a wartości logicznej **Prawda** – odpowiada liczba **-1**. Jeżeli chcemy mieć konkretne napisy zamiast liczb 0 i -1, to musimy zmienić format wyświetlania pól:

- wrót do trybu projektowania kwerendy,
- ustaw się w pierwszym z wyliczanych pól i z menu podręcznego wybierz **Właściwości**,
- ponieważ lista rozwijana Format nie proponuje nam niczego sensownego, sam(a) wpisz format w postaci: **;"Prawda";"Falsz"**
- kliknij teraz kolejne pola w siatce projektowania i wpisz identyczny format... (Aby się zanadto nie przemieszczać, wystarczy raz wpisany format zaznaczyć i skopiować do Schowka. Następnie kliknąć kolejne pole w polu projektowania i w okienku **Właściwości** w wierszu **Format** wkleić zawartość Schowka.)
- zapisz zmodyfikowaną kwerendę i uruchom ją.

Uwaga!

Dla danych typu **Tak/Nie** mamy dostępne tylko 3 gotowe formaty: **Tak/Nie**, **Prawda/Falsz**, **Wł/Wył**. Użytkownik może wprowadzić własny format w postaci:

;"tekst dla wartości logicznej Prawda";" tekst dla wartości logicznej Falsz"

Tworząc własne, niestandardowe formaty dla danych typu **Tak/Nie** należy pamiętać o tym, że składają się one z trzech sekcji:

Sekcja	Dotyczy
Pierwsza	Nie ma wpływu na format danych typu Tak/Nie. Jednak średnik (;) jest tu obowiązkowy.
Druga	Tekst wyświetlany zamiast standardowej wartości „Tak”, „Prawda” czy „Wł”.
Trzecia	Tekst wyświetlany zamiast standardowej wartości „Nie”, „Falsz” czy „Wył”.

Temat: Funkcje tekstowe w kwerendach.

UCase(ciąg_znaków) – funkcja zamienia podany ciąg znaków na duże litery.

LCase(ciąg_znaków) – funkcja zamienia podany ciąg znaków na małe litery.

Left(ciąg_znaków; długość) – funkcja zwraca podaną liczbę znaków począwszy od lewej strony ciągu.

Right(ciąg_znaków; długość) – funkcja zwraca podaną liczbę znaków począwszy od prawej strony ciągu.

Mid(ciąg_znaków; start; długość) – funkcja zwraca podaną liczbę znaków począwszy od lewej strony ciągu.

Len(ciąg_znaków) – funkcja zwraca liczbę znaków w ciągu znaków.

StrConv(ciąg_znaków;konwersja) – funkcja zwraca ciąg znaków poddany żądanej konwersji.

Argument	Opis	
ciąg_znaków	Wyrażenie znakowe podlegające konwersji.	
konwersja	Argument typu Integer – suma wartości określających typ przeprowadzanych konwersji.	
	Wartość	Opis
	1	Zamienia znaki ciągu na wielkie litery.
	2	Zamienia znaki ciągu na małe litery.
	3	Zamienia pierwszą literę każdego wyrazu w ciągu na wielką literę.

Ćwiczenie

Przepisz temat i funkcje tekstowe wraz z opisem. Możesz pogłębić swoją wiedzę na ten temat korzystając z pomocy programu Access lub Internetu.

Ćwiczenie 1

Utwórz nową bazę danych i zapisz ją w pliku **Tekst.accdb**. Następnie w *Widoku Projektu* utwórz tabelę **Dane tekstowe** zawierającą trzy pola: **tekst**, **ile**, **start**:

tekst – wybierz typ danych: **Tekst**

ile – wybierz typ danych: **Liczba**, rozmiar pola: **Liczba całkowita**

start – wybierz typ danych: **Liczba**, rozmiar pola: **Liczba całkowita**.

Wypełnij tabelę danymi – uzupełnij pola **tekst**, **ile** i **start** jak w ćwiczeniu 2.

Utwórz kwerendę **Formatowanie tekstu** w oparciu o pole **tekst**, która wyświetli tekst w różny sposób: dużymi literami, małymi oraz tylko pierwszą literę dużą.

tekst	dużymi: UCase([tekst])	małymi: LCase([tekst])	pierwsza duża: StrConv([tekst];3)
Alabama			
kot i co dalej			
krokodyl			
Ala chyba ma kota			

Ćwiczenie 2

Utwórz nową kwerendę w *Widoku Projektu* wycinającą tekst i zapisz ją pod nazwą **Wycinanki**. Przenieś do niej wszystkie trzy pola z tabeli **Dane tekstowe**.

tekst	ile	start	z lewej: Left([tekst];[ile])	z prawej: Right([tekst];[ile])	ze środka: Mid([tekst];[start];[ile])
Alabama	4	2			
kot i co dalej	8	2			
krokodyl	2	2			
Ala chyba ma kota	5	5			

Ćwiczenie 3

Utwórz nową tabelę o polach imię i nazwisko. Wypełnij ją. Utwórz kwerendę, która spowoduje wyświetlanie:

- inicjałów imienia i nazwiska w jednym polu,
- pierwszej litery imienia i całego nazwiska w następnym polu.

Temat: Funkcje sterujące.

CHOOSE(*indeks*; *wyrażenie_1*; *wyrażenie_2*; ...; *wyrażenie_n*)

Funkcja zwraca wskazaną przez argument *indeks* wartość z listy argumentów *wyrażenie1*, ... *wyrażenie_n*.

Argument	Opis
<i>indeks</i>	Wyrażenie liczbowe lub pole o wartości z przedziału od 1 do liczby dostępnych argumentów do wyboru (n).
<i>wyrażenie</i>	Wyrażenie, zawierające jedną z możliwości do wyboru.

IIF(*wyrażenie*; *gdy_prawda*; *gdy_fałsz*)

Zależnie od wartości logicznej *wyrażenia* funkcja zwraca argument *czy_prawda* lub *czy_fałsz*. Jest to po prostu instrukcja warunkowa.

Argument	Opis
<i>wyrażenie</i>	Warunek logiczny.
<i>gdy_prawda</i>	Wartość lub wyrażenie zwracane przez funkcję, gdy argument <i>wyrażenie</i> ma wartość <i>Prawda</i> .
<i>gdy_fałsz</i>	Wartość lub wyrażenie zwracane przez funkcję, gdy argument <i>wyrażenie</i> ma wartość <i>Fałsz</i> .

SWITCH(*wyrażenie_1*; *wartość_1*; [...*wyrażenie_n*; *wartość_n*])

Funkcja oblicza wartości na liście *wyrażeń* i zwraca *wartość* lub *wyrażenie* odpowiadające pierwszemu na liście *wyrażeniu* o wartości *Prawda*.

Argument	Opis
<i>wyrażenie</i>	Wyrażenie, które ma być obliczone.
<i>wartość</i>	Wartość lub wyrażenie, które zostanie zwrócone, jeśli odpowiadający argument <i>wyrażenie</i> ma wartość <i>Prawda</i> .

Ćwiczenie

Wykonaj w zeszycie notatkę wraz z jednym przykładem o funkcjach sterujących na podstawie pomocy programu Access lub Internetu.

Ćwiczenie 1

Utwórz tabelę o polach a, b, c przechowujących liczby rzeczywiste. Utwórz kwerendę **Trójmian** obliczając pierwiastki równania kwadratowego $ax^2 + bx + c = 0$.

a	b	c	delta:	wniosek1:	wniosek2:	wniosek3:	x1:	x2:
1	0	4						
3	-2	1						
1,5	2,75	-7						
4,25	-1,13	0						
1	2	-3						
1	5	-6						
1	1	1						
1	4	4						

- Przenieś do kwerendy pola a, b i c. Utwórz nowe pole o nazwie (delta), gdzie obliczysz wartość delty.
- W nowym polu (wniosek1) wyświetl: „funkcja ma pierwiastki” lub „funkcja nie ma pierwiastków” na podstawie wartości delty. Przepisz do zeszytu funkcję **IIF**, która rozwiązała to polecenie.
- W nowym polu (wniosek2) wyświetl: „większa” „zero” „mniejsza” na podstawie wartości delty. Przepisz do zeszytu funkcję **IIF**, która rozwiązała to polecenie.
- W nowym polu (wniosek3) wyświetl: „większa” „zero” „mniejsza” na podstawie wartości delty z użyciem funkcji **SWITCH**. Przepisz do zeszytu funkcję **SWITCH**, która rozwiązała to polecenie.
- Oblicz pierwiastki równania kwadratowego x1 i x2. Błędy wystąpiły w tych rekordach, gdzie delta była ujemna. Popraw konstrukcję tego wyrażenia wykorzystując funkcję **IIF** i przepisz ją zeszytu.

Temat: Funkcje daty i czasu.

DATE() – funkcja zwraca bieżącą datę wskazywaną przez zegar systemowy.

DATEADD(przedział; liczba; data) - funkcja zwraca datę, do której dodano określony przedział czasu.

Argument	Opis			
<i>przedział</i>	Wyrażenie znakowe będące przedziałem czasu, który ma zostać dodany.			
	Ustawienie	Opis	Ustawienie	Opis
	yyyy	Rok	ww	Tydzień
	q	Kwartał	g	Godzina
	m	Miesiąc	n	Minuta
	y	Dzień roku	s	Sekunda
<i>liczba</i>	Wyrażenie numeryczne będące liczbą przedziałów, które mają zostać dodane.			
<i>data</i>	Data, do której dodawany jest przedział czasu.			

DATEDIFF(przedział;data1;data2;[pierwszy_dzień_tygodnia;pierwszy_tydzień_roku])

Funkcja zwraca wartość określającą liczbę przedziałów czasowych pomiędzy dwoma datami.

Argument	Opis		
<i>przedział</i>	Wyrażenie znakowe, będące przedziałem czasu używanym do wyliczenia różnicy pomiędzy datami <i>data1</i> i <i>data2</i>		
<i>data1, data2</i>	Skrajne daty, które mają być użyte do obliczeń.		
<i>pierwszy_dzień_tygodnia</i>	Stała określająca pierwszy dzień tygodnia (domyślnie – niedziela).		
	Stała	Wartość	Opis
	vbSunday	1	Niedziela
	vbMonday	2	Poniedziałek
	vbTuesday	3	Wtorek
	vbWednesday	4	Środa
	vbThursday	5	Czwartek
	vbFriday	6	Piątek
	vbSaturday	7	Sobota
<i>pierwszy_tydzień_roku</i>	Stała określająca pierwszy tydzień roku.		
	Stała	Wartość	Opis
	vbFirstJan1	1	Rozpoczyna liczenie od tygodnia zawierającego dzień 1 stycznia.
	vbFirstFourDays	2	Rozpoczyna liczenie od tygodnia zawierającego, co najmniej cztery dni nowego roku.
	vbFirstFullWeek	3	Rozpoczyna liczenie od pierwszego pełnego tygodnia nowego roku.

WEEKDAY(data;[pierwszy_dzień_tygodnia]) - funkcja zwraca liczbę całkowitą określającą dzień tygodnia.

Argument	Opis
<i>data</i>	Wyrażenie numeryczne, wyrażenie znakowe lub ich dowolna kombinacja określająca datę.
<i>pierwszy_dzień_tygodnia</i>	Stała określająca pierwszy dzień tygodnia (domyślnie – niedziela). Ustawienia tego argumentu są identyczne jak w funkcji DateDiff.

YEAR(data) - funkcja zwraca liczbę całkowitą określającą rok.

MONTH(data) - funkcja zwraca liczbę całkowitą z przedziału od 1 do 12, określającą miesiąc roku.

DAY(data) - funkcja zwraca liczbę całkowitą z przedziału od 1 do 31, określającą dzień tygodnia.

HOURL(godzina) - funkcja zwraca liczbę całkowitą z przedziału od 0 do 23, określającą godzinę.

MINUTE(godzina) - funkcja zwraca liczbę całkowitą z przedziału od 0 do 59, określającą minutę godziny.

SECOND(godzina) - funkcja zwraca liczbę całkowitą z przedziału od 0 do 59, określającą sekundę w minucie.

NOW() - funkcja zwraca bieżącą datę i godzinę pobraną z zegara systemowego.

Ćwiczenie

Przepisz do zeszytu temat i opisy funkcji daty i czasu. Możesz wkleić do zeszytu wydrukowaną kartkę. Możesz poszerzyć swoją wiedzę korzystając z pomocy programu MS Access lub Internetu.

Ćwiczenie 1

Utwórz nową bazę danych i zapisz ją w pliku **Czas.accdb**. Następnie w *Widoku Projektu* utwórz tabelę zawierającą trzy pola: **chwila1**, **przedział**, **liczba**:

chwila1 – wybierz typ: **Data/Godzina**, a we właściwościach pola nadaj mu format: **Data ogólna**

przedział – typ danych: **Tekst** (rozmiar pola: **5** znaków)

liczba - typ danych: **Liczba** (rozmiar pola: **Liczba całkowita**)

Zapisz tabelę pod nazwą **Czas**. Wprowadź do niej następujące dane:

chwila1	przedział	liczba
2009-11-12 10:15:20	ww	9
2000-06-09 04:21:19	m	5
1961-08-20 13:51:05	g	36
1988-06-01 18:11:22	s	1800
1989-08-03 14:15:10	yyyy	18

Utwórz w *Widoku Projektu* nową kwerendę (**Elementy daty**), która pokaże Ci w praktyce, jak Access wykonuje operacje na datach. Z tabeli **Czas** przenieś pole **chwila1**:

chwila1	rok: Year([chwila1])	miesiąc: Month([chwila1])	dzień: Day([chwila1])	dzień tygodnia: Weekday([chwila1])

Chcielibyśmy, aby dzień tygodnia wyświetlał się słownie. Wróć do trybu projektowania kwerendy, włącz **Powiększenie** (Shift+F2) i skorzystaj z funkcji **Choose**:

dzień tygodnia: Choose(Weekday([chwila1]);"Niedziela";"Poniedziałek";"Wtorek";"Środa";"Czwartek";"Piątek";"Sobota")

Korzystając z funkcji **Choose** spraw, aby miesiąc wyświetlał się słownie.

Ćwiczenie 2

Utwórz w *Widoku Projektu* nową kwerendę (**Elementy czasu**) opartą na tabeli **Czas**. Z tabeli **Czas** przenieś pole **chwila1**:

chwila1	godzina: Hour([chwila1])	minuta: Minute([chwila1])	sekunda: Second([chwila1])

Ćwiczenie 3

Utwórz w *Widoku Projektu* nową kwerendę (**Za chwilę**) opartą na tabeli **Czas**. Z tabeli **Czas** przenieś pole **chwila1**, **przedział**, **liczba**. Wykorzystaj funkcję **DateAdd**.

chwila1	przedział	liczba	za trochę: DateAdd([przedział];[liczba];[chwila1])

Pole **za trochę** pokazuje, jaka chwila będzie za podaną **liczbę** jednostek wyspecyfikowanych w polu **przedział**.

Ćwiczenie 4

W oparciu o tabelę **Czas** utwórz jeszcze jedną kwerendę (**różnica**) zawierającą pola: **przedział** i **chwila1**. Wykorzystaj funkcję **Now** i **DateDiff** do określenia różnicy przedziałów.

przedział	chwila1	dzisiaj: Now()	różnica: DateDiff ([przedział];[chwila1];[dzisiaj])