



# BAZY DANYCH

Modyfikacja bazy danych Perspektywy – Widoki, Procedury

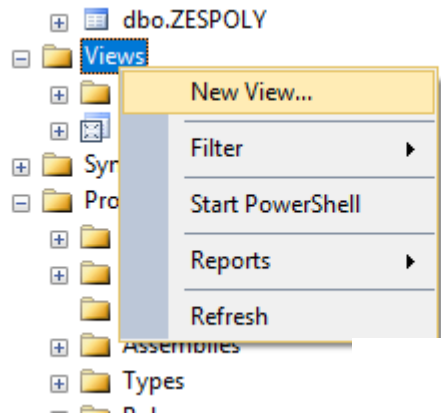
mgr inż. Patryk Kaczmarek

[patryk.kaczmarek@ansleszno.pl](mailto:patryk.kaczmarek@ansleszno.pl)

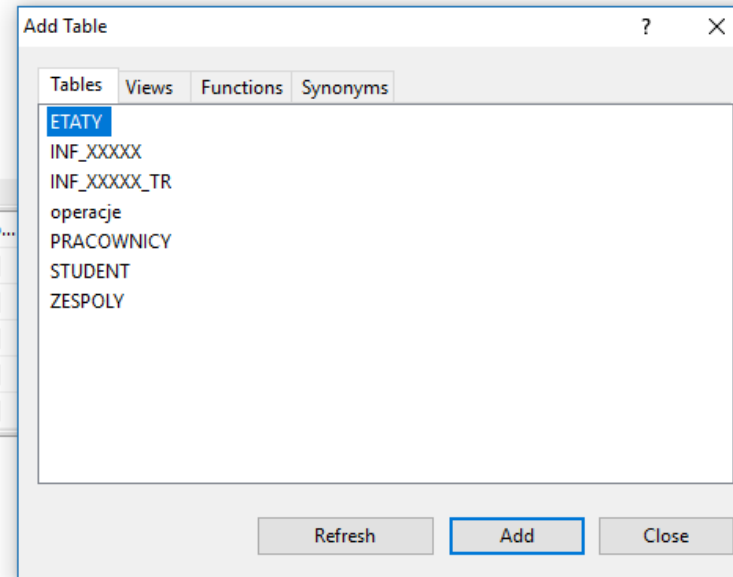
# Plan laboratorium

- Tworzenie Widoków
- Tworzenie procedur

# Tworzenie widoku

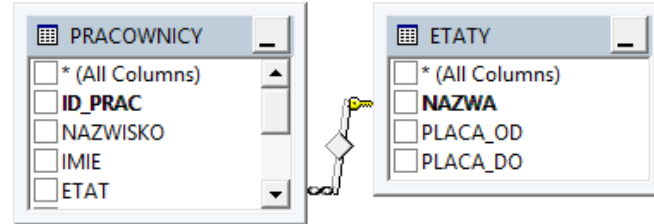


Alias	Table	Outp...
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>



Or...

# Tworzenie widoku



Column	Alias	Table	Outp...	Sort Type	Sort Order
			<input type="checkbox"/>		
dbo.PRACOWNICY.*			<input type="checkbox"/>		
dbo.PRACOWNICY.ID_PRAC			<input type="checkbox"/>		
dbo.PRACOWNICY.NAZWISKO			<input type="checkbox"/>		
dbo.PRACOWNICY.IMIE			<input type="checkbox"/>		
dbo.PRACOWNICY.ETAT			<input type="checkbox"/>		
dbo.PRACOWNICY.ID_SZEFA					
dbo.PRACOWNICY.ZATRUDNIONY					
dbo.PRACOWNICY.PLACA_POD	TAT = dbo.ETATY.NAZWA				

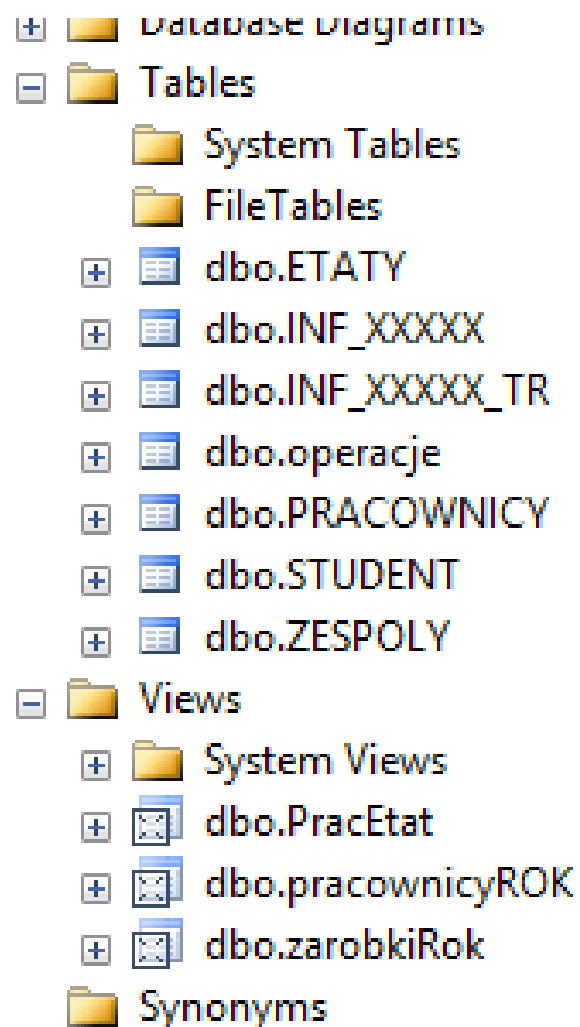
# Tworzenie widoku polecenie

```
CREATE VIEW nazwaWidoku as  
polecenie SQL;
```

# Tworzenie widoku wywołanie

```
SELECT * FROM nazwaWidoku
```

# Tworzenie widoku Podgląd



# Procedury

Model polecenia

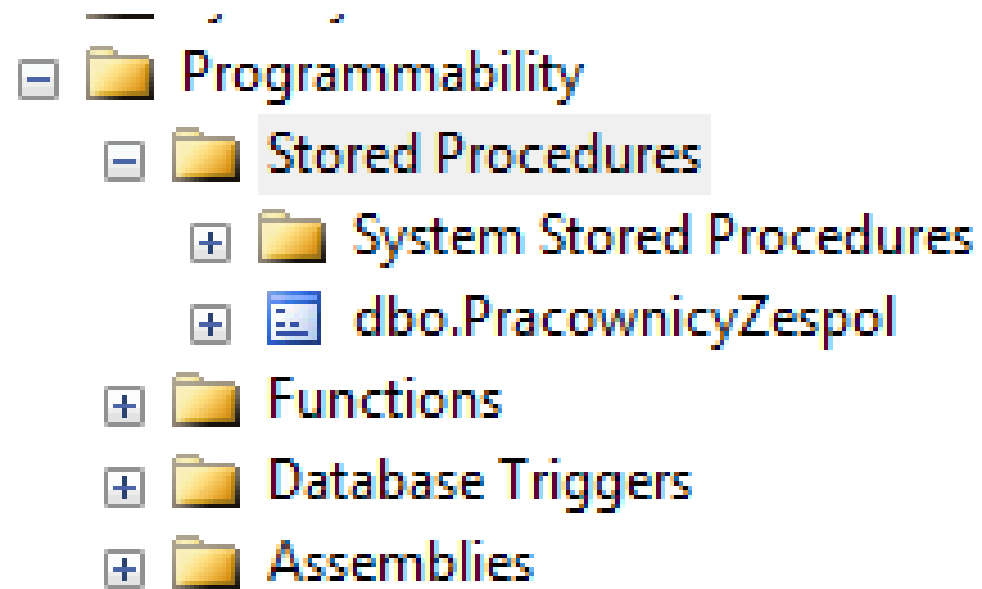
```
CREATE PROCEDURE nazwa_procedury  
    {@parametr typ}  
    AS
```

Zapytanie z wykorzystaniem parametru

# Procedure - Przykład

```
CREATE PROCEDURE [PracownicyZespol] @idZesp int
AS
SELECT *
FROM [PWSZ_test].[dbo].[PRACOWNICY]
WHERE
ID_ZESP = @idZesp;
```

# Procedury Podgląd



# Procedury wywołanie

EXEC nazwaProcedury parametr

# Procedury wywołanie

```
EXEC PracownicyZespol 20
```

00 %

Results Messages

	ID_PRAC	NAZWISKO	IMIE	ETAT	ID_SZEFA	ZATRUDNIONY	PLACA_POD	PLACA_DOD	ID_ZESP
1	130	Nowak	Piotr	PROFESOR	100	1968-01-07 00:00:00.000	3960.00	NULL	20
2	140	Kowalski	Krzysztof	PROFESOR	130	1975-01-09 00:00:00.000	3230.00	805.00	20
3	150	Grzybowska	Maria	ADIUNKT	130	1977-01-09 00:00:00.000	2845.50	NULL	20
4	160	Krakowska	Joanna	SEKRETARKA	130	1985-01-03 00:00:00.000	1590.00	NULL	20
5	170	Opolski	Roman	ASYSTENT	130	1992-01-10 00:00:00.000	1839.70	480.50	20
6	190	Kotarski	Konrad	ASYSTENT	140	1993-01-09 00:00:00.000	1971.00	NULL	20
7	220	Siekierski	Mateusz	ASYSTENT	110	1993-01-10 00:00:00.000	1889.00	NULL	20

# Funkcje skalarne

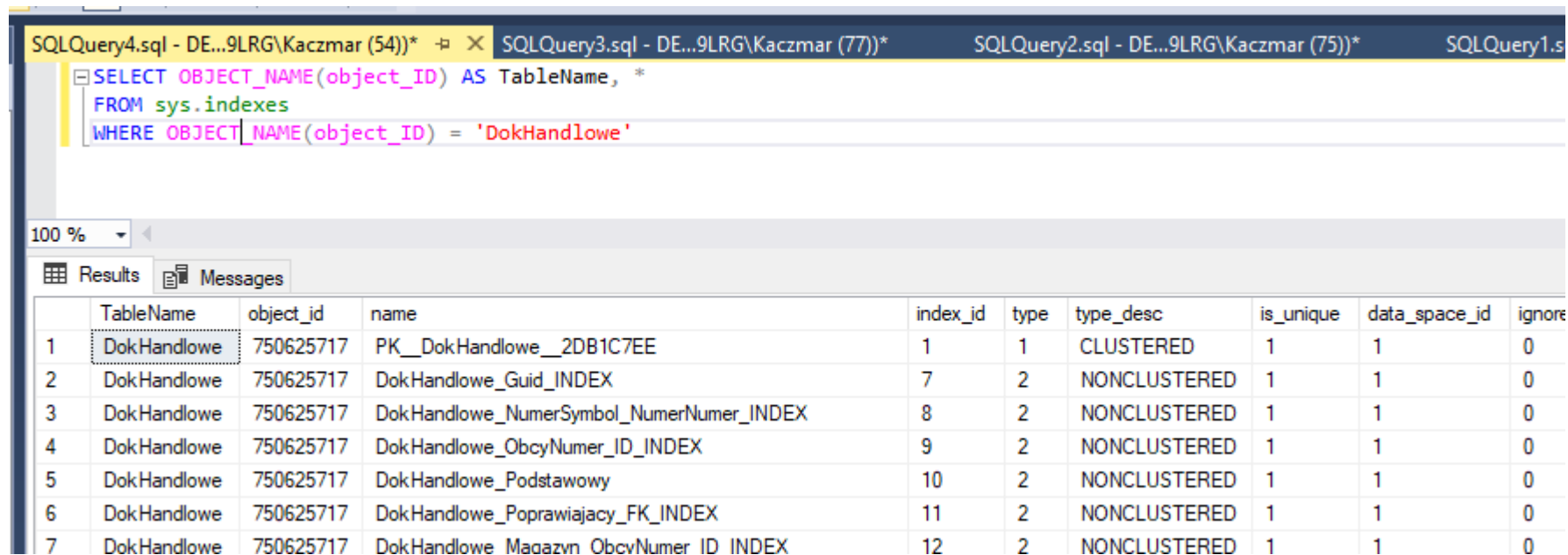
Systemowe funkcje wbudowane zwracające metadane, stosujemy zazwyczaj w procedurach utrzymaniowych, np. do identyfikacji obiektów, ich parametrów czy wykonywania na nich działań administracyjnych.

Funkcje `OBJECT_ID()` oraz `OBJECT_NAME()` pozwalają pójść na skróty w odpytywaniu widoków systemowych.

Zwracają identyfikator obiektu lub jego nazwę. Bywają przydatne, szczególnie jeśli na szybko, chcemy wyciągnąć jakieś dane, a nie znamy identyfikatora danego obiektu i nie chce nam się łączyć dodatkowych widoków.

# Funkcje skalarne

Dla przykładu – chcemy na szybko wyświetlić informacje o wszystkich indeksach w tabeli. W skrótovej wersji, moglibyśmy zapisać zapytanie w taki sposób :



The screenshot shows a SQL Server query window with the following SQL code:

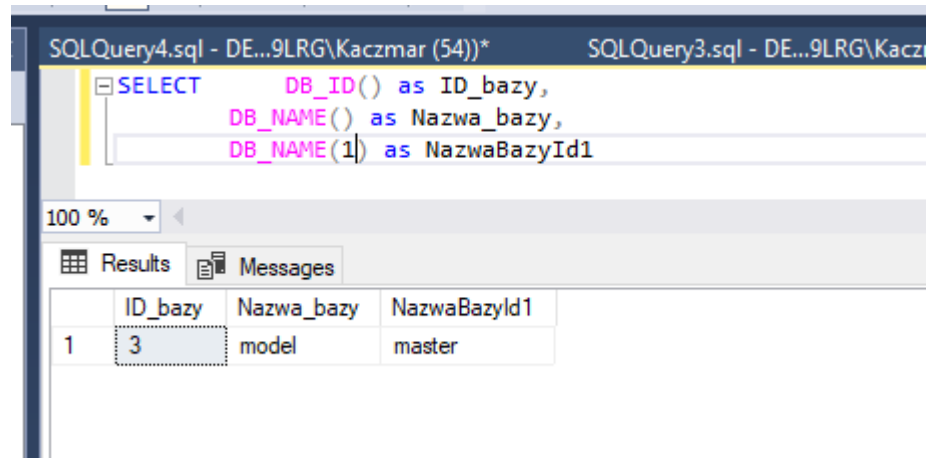
```
SELECT OBJECT_NAME(object_ID) AS TableName, *  
FROM sys.indexes  
WHERE OBJECT_NAME(object_ID) = 'DokHandlowe'
```

The results pane displays the following table:

	TableName	object_id	name	index_id	type	type_desc	is_unique	data_space_id	ignore
1	DokHandlowe	750625717	PK_DokHandlowe__2DB1C7EE	1	1	CLUSTERED	1	1	0
2	DokHandlowe	750625717	DokHandlowe_Guid_INDEX	7	2	NONCLUSTERED	1	1	0
3	DokHandlowe	750625717	DokHandlowe_NumerSymbol_NumerNumer_INDEX	8	2	NONCLUSTERED	1	1	0
4	DokHandlowe	750625717	DokHandlowe_ObcyNumer_ID_INDEX	9	2	NONCLUSTERED	1	1	0
5	DokHandlowe	750625717	DokHandlowe_Podstawowy	10	2	NONCLUSTERED	1	1	0
6	DokHandlowe	750625717	DokHandlowe_Poprawiajacy_FK_INDEX	11	2	NONCLUSTERED	1	1	0
7	DokHandlowe	750625717	DokHandlowe Magazyn ObcyNumer ID INDEX	12	2	NONCLUSTERED	1	1	0

# Funkcje skalarne

Funkcje DB\_ID() oraz DB\_NAME(), analogicznie jak w przypadku poprzednio prezentowanych funkcji, zwracają translacje nazwy na identyfikator lub odwrotnie. Samo wywołanie, bez podania parametru, zwróci aktualną nazwę / identyfikator bazy danych do której jesteś podłączony.



```
SQLQuery4.sql - DE...9LRG\Kaczmar (54))*    SQLQuery3.sql - DE...9LRG\Kaczmar (54)*
SELECT    DB_ID() as ID_bazy,
          DB_NAME() as Nazwa_bazy,
          DB_NAME(1) as NazwaBazyId1
```

100 %

Results Messages

	ID_bazy	Nazwa_bazy	NazwaBazyId1
1	3	model	master

# Zmienne globalne

## Zmienne @@GLOBALNE

Zmienne globalne to bardzo podobne do funkcji skalarnych obiekty. Ich nazwa zaczyna się od @@ i jest ich raptem kilkadziesiąt.

