

BAZY DANYCH

Wykład 7

Tematyka wykładu



2

- Baza danych MS SQL Server i optymalizacja zapytań
- Zarządzanie użytkownikami, grupami i rolami w bazie danych
- Zakładanie nowej bazy danych
- Instrukcje uzupełniające poznane funkcje
 - ▣ Formatowanie daty
 - ▣ Typy tablicowe

3 Baza danych MS SQL Server

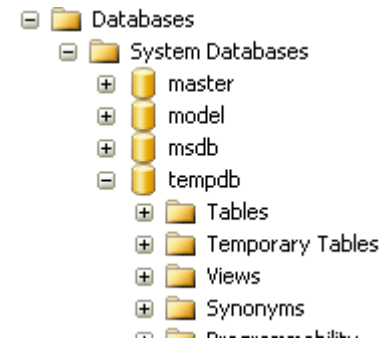
Bazy w MS SQL Server

Optymalizacja zapytań

Nazwy baz danych MS SQL Server

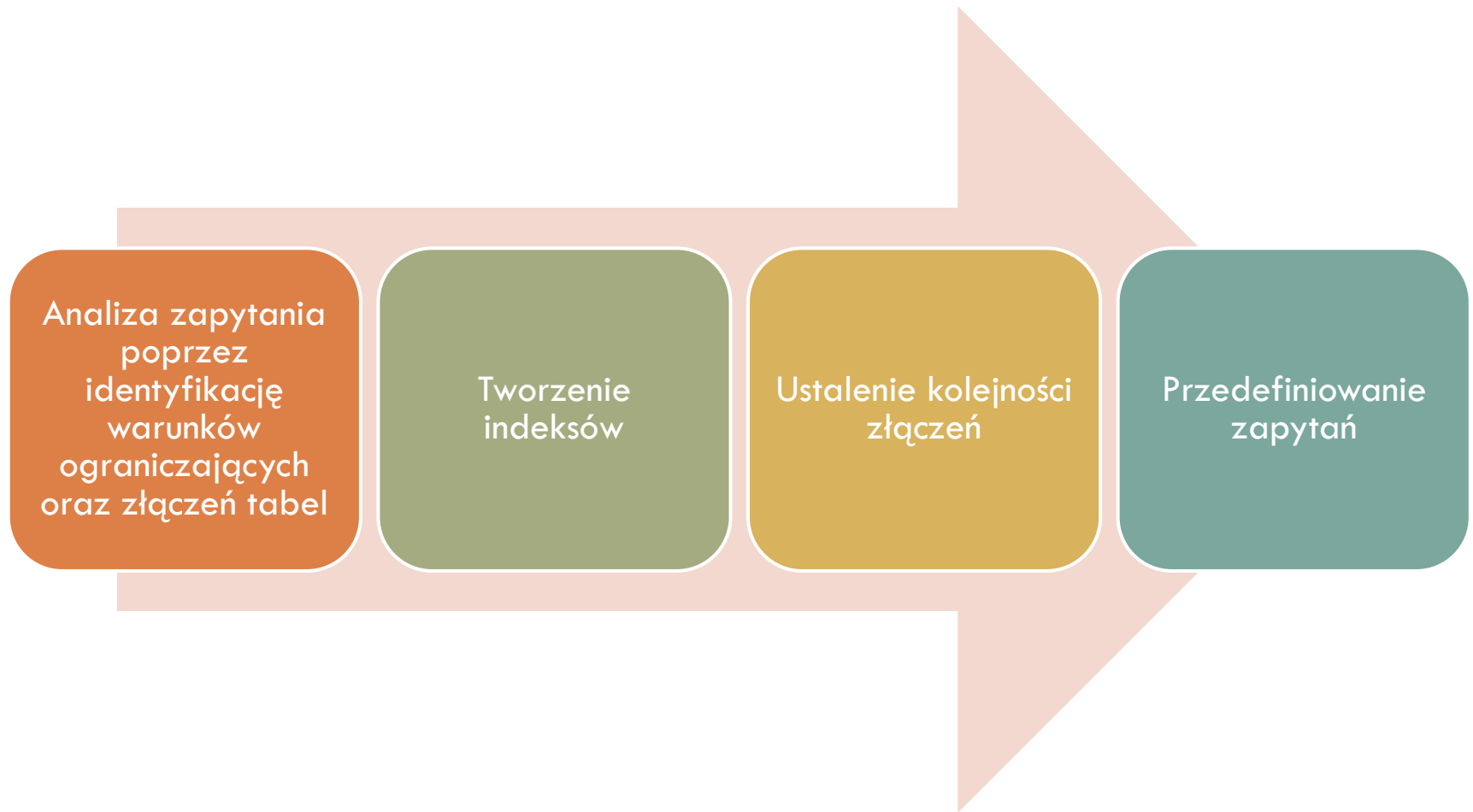
4

1. **Master** – dane konfiguracyjne, dzienniki zdarzeń
2. **Resource** – baza ukryta, posiada obiekty związane z bazą danych
3. **Model** – wzorzec dla nowych baz danych
4. **MSDB** – harmonogramowanie danych, dane historyczne, odtwarzanie bazy danych (kopie zapasowe)
5. **TempDB** – przechowuje dane tymczasowe



Procedura optymalizacji zapytań

5



Analiza zapytania



6

- Przekształcenie zapytania, aby posiadał jedynie następujące operatory porównania:
 - ▣ =, >, <, >=, <=, BETWEEN, LIKE – tylko w sytuacji, gdy porównywany jest wzorzec zawierający pierwszy znak inny niż %

- W miarę możliwości należy rezygnować z porównań typu `lower(imie)='jan'`

Przykłady przekształceń (1 / 3)



7

□ Zamiast:

□ `select * from tabela where cena <> 0`

□ należy używać:

□ `select * from tabela where cena > 0`

Przykłady przekształceń (2/3)



8

□ Zamiast:

▣ `select * from tabela where substring(nazwisko,1,1) = 'K'`

□ należy używać:

▣ `select * from tabela where nazwisko LIKE 'K%'`

Przykłady przekształceń (3/3)



9

□ Zamiast:

▣ `select * from tabela where wartosc * 1000 = 5000`

□ należy używać:

▣ `select * from tabela where wartosc = 5`

Optymalizacja – inne przykłady

10

- Dla tablic wykorzystujących JOIN powinno się stosować zmaterializowane widoki
- Indeksy powinny być dobrane pod względem częstości korzystania z danych



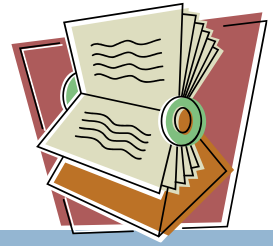
Tworzenie indeksów



11

- Indeksy pozwalają na szybsze działanie bazy danych. Zazwyczaj są tworzone podczas tworzenia tabeli, jednak istnieje możliwość tworzenia indeksów po jej utworzeniu.
- Do tworzenia indeksów używana jest instrukcja CREATE INDEX.
 - ▣ CREATE [UNIQUE] INDEX nazwaIndeksu ON nazwaTabeli(nazwaIndeksowanejKolumny, ...)
- gdzie:
 - ▣ nazwaKolumnyIndeksu: nazwaKolumny [ASC | DESC]
 - ▣ UNIQUE – tworzy indeksy unikalne
 - ▣ ASC oraz DESC – tworzenie indeksów z rosnącym lub malejącym porządkiem sortowania

Przykłady poleceń tworzących indeks



12

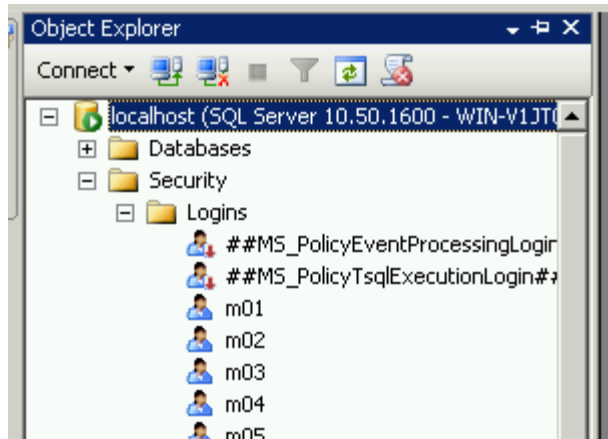
- ❑ `create unique index fragmentNazwiskaInd
on tablica(nazwa_towaru desc,id);`
- ❑ `drop index fragmentNazwiskaInd on tablica;`

Zarządzanie użytkownikami i grupami

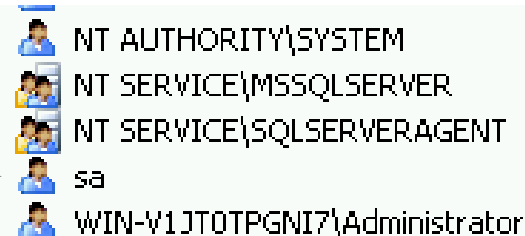


SQL Server Management Studio - Logins

14



**Administrator
serwera**

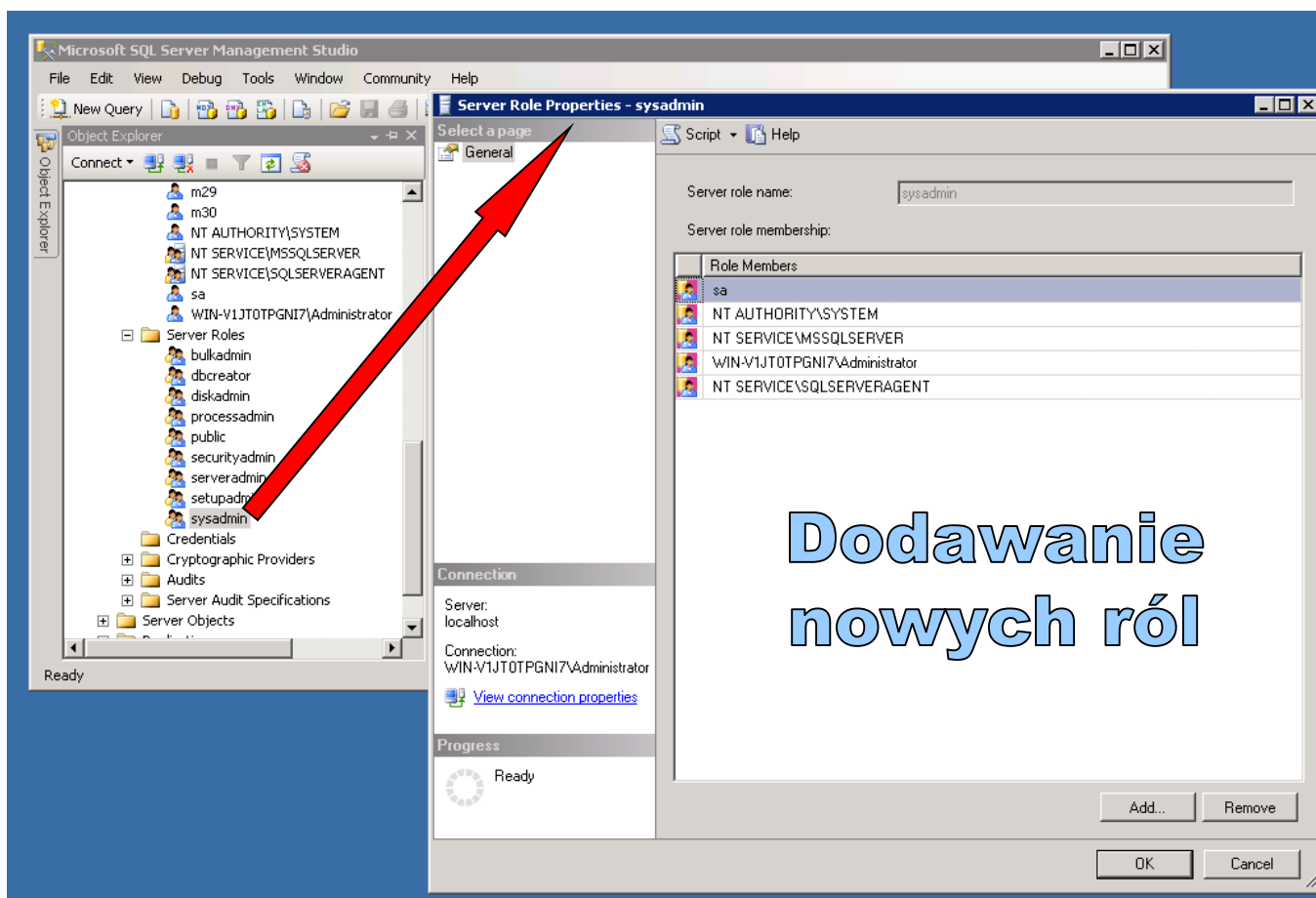


A list of SQL Server logins with their corresponding icons. The logins are: NT AUTHORITY\SYSTEM, NT SERVICE\MSSQLSERVER, NT SERVICE\SQLSERVERAGENT, sa, and WIN-V1JT0TPGNI7\Administrator.

NT AUTHORITY\SYSTEM
NT SERVICE\MSSQLSERVER
NT SERVICE\SQLSERVERAGENT
sa
WIN-V1JT0TPGNI7\Administrator

SQL Server Management Studio – Roles

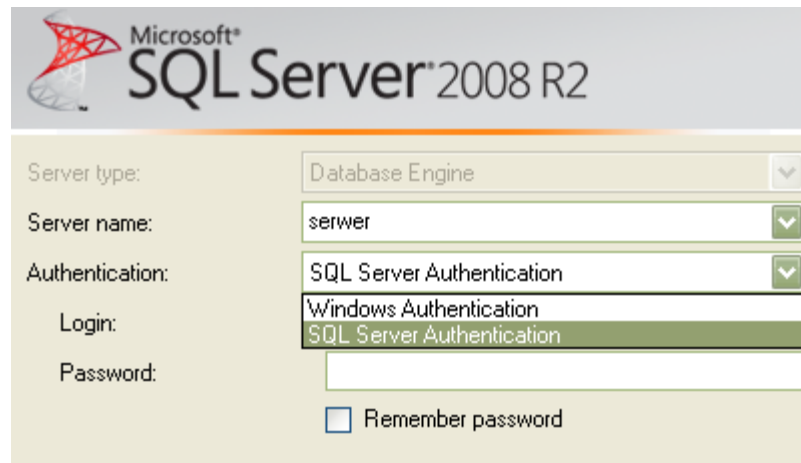
15



Typy uwierzytelniania

16

- ❑ **Windows only** – poprzez konta w systemie Windows
- ❑ **SQL Server and Windows** – poprzez konto SQL Server



The screenshot shows the 'Server type' configuration window for Microsoft SQL Server 2008 R2. The 'Server type' is set to 'Database Engine'. The 'Server name' is 'serwer'. The 'Authentication' dropdown menu is open, showing 'SQL Server Authentication' (selected), 'Windows Authentication', and 'SQL Server Authentication'. The 'Login' field is empty. The 'Password' field is empty. There is a checkbox for 'Remember password' which is unchecked.

Microsoft®
SQL Server® 2008 R2

Server type: Database Engine

Server name: serwer

Authentication: SQL Server Authentication

Login:

Password:

☐ Remember password

Role

17



- **Login security** – łączenie się z serwerem
- **Database security** – dostęp do bazy danych
- **Database objects** – dostęp do indywidualnych obiektów bazy danych

Wbudowane konta i grupy



18

□ Dla serwera:

- **sa** – domyślne konto administratora
- **administratorzy i użytkownicy** – grupy utworzone w systemie MS Windows

□ Dla bazy danych:

- **dbo** – domyślne administratora
- **guest** – konto gościa
- **sys** – konto systemowe

Wbudowane role serwera



19

- SysAdmin – jakikolwiek członek może wykonywać jakiekolwiek działania na serwerze
- ServerAdmin – jakikolwiek członek może ustawiać opcje konfiguracyjne na serwerze
- SetupAdmin – jakikolwiek członek może zarządzać zadaniami i opcjami startowymi dla SQL Server i powiązanych
- SecurityAdmin – jakikolwiek członek może zarządzać bezpieczeństwem serwera
- ProcessAdmin – jakikolwiek członek może zabijać procesy utworzone na SQL Server
- DbCreator – jakikolwiek członek może tworzyć, modyfikować, usuwać oraz odtwarzać bazy danych
- DiskAdmin – jakikolwiek członek może zarządzać plikami na dysku dla SQL Server
- BulkAdmin – jakikolwiek członek może wykonywać instrukcje wstawiające dane

Wbudowane role bazy danych



20

- db_owner – członkowie mają pełny dostęp
- db_accessadmin – członkowie mogą zarządzać grupami MS Windows oraz loginami SQL Server
- db_datareader – członkowie mogą odczytywać wszystkie dane
- db_datawriter – członkowie mogą dodawać, usuwać oraz modyfikować dane w tabelach
- db_ddladmin – członkowie mogą uruchamiać składnie DDL
- db_securityadmin – członkowie mogą modyfikować role członkostwa oraz zarządzać uprawnieniami
- db_backupoperator – członkowie mogą tworzyć kopię zapasową bazy danych
- db_denydatareader – członkowie nie mogą przeglądać zawartości tabel w bazie danych
- db_denydatawriter – członkowie nie mogą zmieniać lub usuwać danych w tabelach lub perspektywach

Uprawnienia do tabel, perspektyw, procedur i innych obiektów

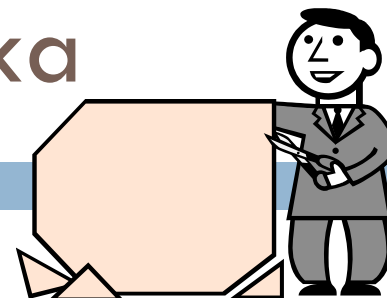
21



- **Alter** – tworzenie, modyfikacja i usuwanie tabel
- **Control** – pełne uprawnienia do zarządzania obiektem
- **Delete, insert, select, update** – tylko dla pojedynczej grupy instrukcji (nie dotyczy procedur i funkcji)
- **Take ownership** – pełne uprawnienia
- **View definition** – dostęp do metadanych obiektu
- **Execute** – wykonanie procedury lub funkcji

Dodawanie nowego użytkownika

22



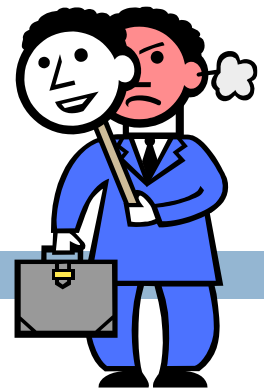
□ Ogólne (SQL Server i MS Windows)

▣ CREATE USER nazwa_użytkownika [{ { FOR | FROM } {
LOGIN nazwa_logowania | CERTIFICATE nazwa_certyfikatu
| ASYMMETRIC KEY nazwa_klucza_asym }] [WITH
DEFAULT_SCHEMA = nazwa_schematu]

□ Domenowego (MS Windows)

▣ CREATE LOGIN
[<nazwa_domeny>\<nazwa_użytkownika>] FROM
WINDOWS;

Zmiana nazwy użytkownika



23

- ALTER USER nazwa_użytkownika WITH < element_set > [,...n] < element_set > ::= NAME = nazwa_nowego_użytkownika | DEFAULT_SCHEMA = nazwa_schematu

Przykład – zakładanie nowego użytkownika

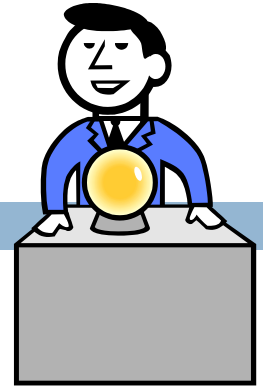
24



1. USE master;
2. GO
3. CREATE LOGIN JM WITH PASSWORD = 'haslo123';
4. GO
5. CREATE USER JM FOR LOGIN JM;
6. GO
7. GRANT ALTER To JM;
8. GO
9. GRANT CONTROL To JM;
10. GO
11. sp_addrolemember @rolename = 'db_owner', @membername = 'jm' ; go
12. sp_addrolemember @rolename = 'db_accessadmin', @membername = 'jm'; go
13. sp_addrolemember @rolename = 'db_datareader', @membername = 'jm'; go

Usuwanie użytkownika

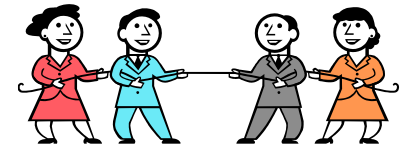
25



- DROP USER nazwa_użytkownika

- Przykład:
 - ▣ DROP USER JM;

Lista członków roli serwera



26

- `sp_helpsrvrolemember [[@srvrolename =] 'rola']`
- Przykłady:
 - `sp_helpsrvrolemember`
 - `sp_helpsrvrolemember @srvrolename=sysadmin`

	ServerRole
1	sysadmin
2	sysadmin
3	sysadmin
4	sysadmin
5	sysadmin
6	securityadmin
7	securityadmin
8	serveradmin
9	serveradmin

	ServerRole	MemberName	MemberSID
1	sysadmin	sa	0x01
2	sysadmin	ZARZĄDZANIE NT\SYSTEM	0x010100000000000512000000

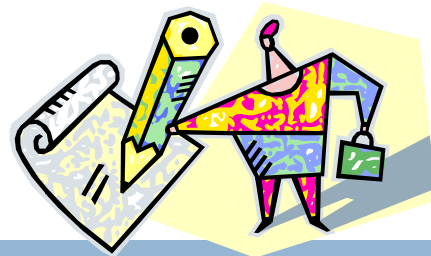
Zarządzanie standardowymi rolami



27

- ❑ `CREATE ROLE nazwa_rol [AUTHORIZATION nazwa_właściciela]`
- ❑ `ALTER ROLE nazwa_rol WITH NAME = nowa_nazwa`
- ❑ `DROP ROLE nazwa_rol sp_helprole`
`[[@srvrolename =] 'rola']`

Przykład zarządzania rolami



28

- ❑ `CREATE ROLE nowa_rola AUTHORIZATION jacek;`
- ❑ `GO`

- ❑ `ALTER ROLE nowa_rola WITH NAME =
nowa_nazwa_3;`
- ❑ `GO`

- ❑ `DROP ROLE nowa_nazwa_2;`
- ❑ `GO`

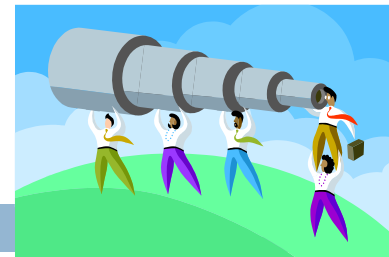
Zarządzanie członkami roli bazy danych

29



- `sp_addrolemember @rolename = 'rola',
@membername = 'użytkownik'`
- `sp_droprolemember @rolename = 'rola',
@membername = 'użytkownik'`
- `sp_helprolemember [[@rolename =] 'rola']`

Przykład dodawania ról



30

- `sp_addrolemember @rolename = 'nowa_rola',
@membername = 'jacek'`
- `sp_helprolemember @rolename='nowa_rola'`

	DbRole	MemberName	MemberSID
1	nowa_rola	Jacek	0xDA66215D578BC347A1798F3B79986250

Zarządzanie rolami aplikacji

31



1. `CREATE APPLICATION ROLE nazwa_rol_i_aplikacji
WITH PASSWORD = 'hasło'`
2. `[, DEFAULT_SCHEMA = nazwa_schematu]`
3. `ALTER APPLICATION ROLE nazwa_rol_i_aplikacji
WITH < zestaw_elementów > [,...n] < element >
::= NAME = nowa_nazwa_rol_i_aplikacji`
4. `| PASSWORD = 'hasło'`
5. `| DEFAULT_SCHEMA = nazwa_schematu DROP
APPLICATION ROLE nazwa_rol_i`

Przykład nadawania uprawnień

32



- GRANT SELECT ON tablica TO student (zezwolenie na dostęp do obiektu)
- DENY SELECT ON tablica TO student (zablokowanie dostępu do obiektu)
- REVOKE SELECT ON tablica TO student (cofnięcie nadanego wcześniej uprawnienia)

Przykład nadania uprawnień do roli



33

1. `CREATE PROCEDURE procedura`
2. `@imie varchar(20) , @identyfikator INT AS`
3. `UPDATE pracownik SET imie= @imie WHERE`
`pracownik_id = @identyfikator`
4. `GO`
5. `GRANT EXECUTE ON procedura`
6. `TO nowa_rola`
7. `GO`

Polecenia weryfikujące

34



□ Lista ról

▣ sp_helprole

Results		Messages	
	RoleName	RoleId	IsAppRole
1	public	0	0
2	nowa_rola2	14	0
3	nowa_nazwa_3	15	0
4	nowa_rola	16	0

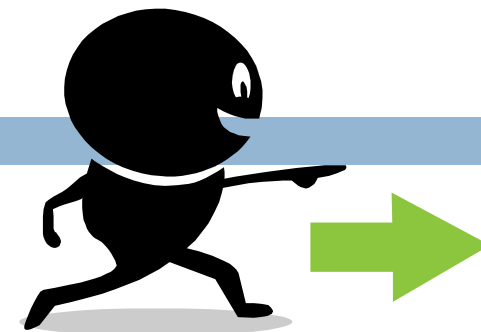
□ Uprawnienia użytkowników

▣ sp_helplogins

	LoginName	DBName	UserName	UserOrAlias
7	BUILTIN\Użytkownicy	DBDS...	db_datareader	MemberOf
8	BUILTIN\Użytkownicy	DBDS...	db_datawriter	MemberOf
9	BUILTIN\Użytkownicy	DBDS...	dbda	User

Sprawdzenie uprawnień dla użytkownika i roli

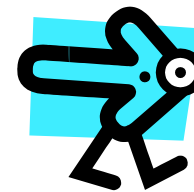
35



```
1. WITH perms_cte as
2. (
3.     select USER_NAME(p.grantee_principal_id) AS principal_name,
4.           dp.principal_id,
5.           dp.type_desc AS principal_type_desc,
6.           p.class_desc,
7.           OBJECT_NAME(p.major_id) AS object_name,
8.           p.permission_name,
9.           p.state_desc AS permission_state_desc
10.    from   sys.database_permissions p
11.   inner JOIN sys.database_principals dp
12.   on      p.grantee_principal_id = dp.principal_id
13. )
14. --users
15. SELECT p.principal_name, p.principal_type_desc, p.class_desc, p.[object_name], p.permission_name, p.permission_state_desc, cast(NULL as
sysname) as role_name
16. FROM   perms_cte p
17. WHERE  principal_type_desc <> 'DATABASE_ROLE'
18. UNION
19. --role members
20. SELECT rm.member_principal_name, rm.principal_type_desc, p.class_desc, p.object_name, p.permission_name,
p.permission_state_desc,rm.role_name
21. FROM   perms_cte p
22. right outer JOIN (
23.     select role_principal_id, dp.type_desc as principal_type_desc, member_principal_id,user_name(member_principal_id) as
member_principal_name,user_name(role_principal_id) as role_name--,*
24.     from   sys.database_role_members rm
25.     INNER JOIN sys.database_principals dp
26.     ON      rm.member_principal_id = dp.principal_id
27. ) rm
28. ON      rm.role_principal_id = p.principal_id
29. order by 1
```

181	Jacek	SQL_USER	OBJECT_OR_COLUMN	procedura	EXECUTE	GRANT	nowa_rola
-----	-------	----------	------------------	-----------	---------	-------	-----------

Sprawdzenie uprawnień użytkownika



36

□ Uprawnienia bieżącego użytkownika na bazie danych

▣ `SELECT * FROM sys.fn_my_permissions(default, 'DATABASE');`

	entity_name	subentity_name	permission_name
1	database		CREATE TABLE
2	database		CREATE VIEW
3	database		CREATE PROCEDURE
4	database		CREATE FUNCTION
5	database		CREATE RULE
6	database		CREATE DEFAULT

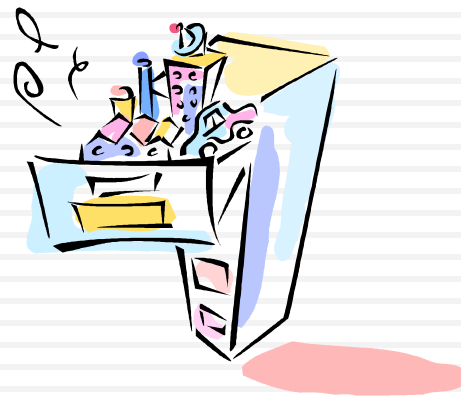
□ Uprawnienia innego użytkownika w zakresie serwera

▣ `EXECUTE AS LOGIN = 'sa';`

▣ `SELECT * FROM sys.fn_my_permissions(default, 'SERVER');`

	entity_name	subentity_name	permission_name
1	server		CONNECT SQL
2	server		SHUTDOWN
3	server		CREATE ENDPOINT
4	server		CREATE ANY DATABASE

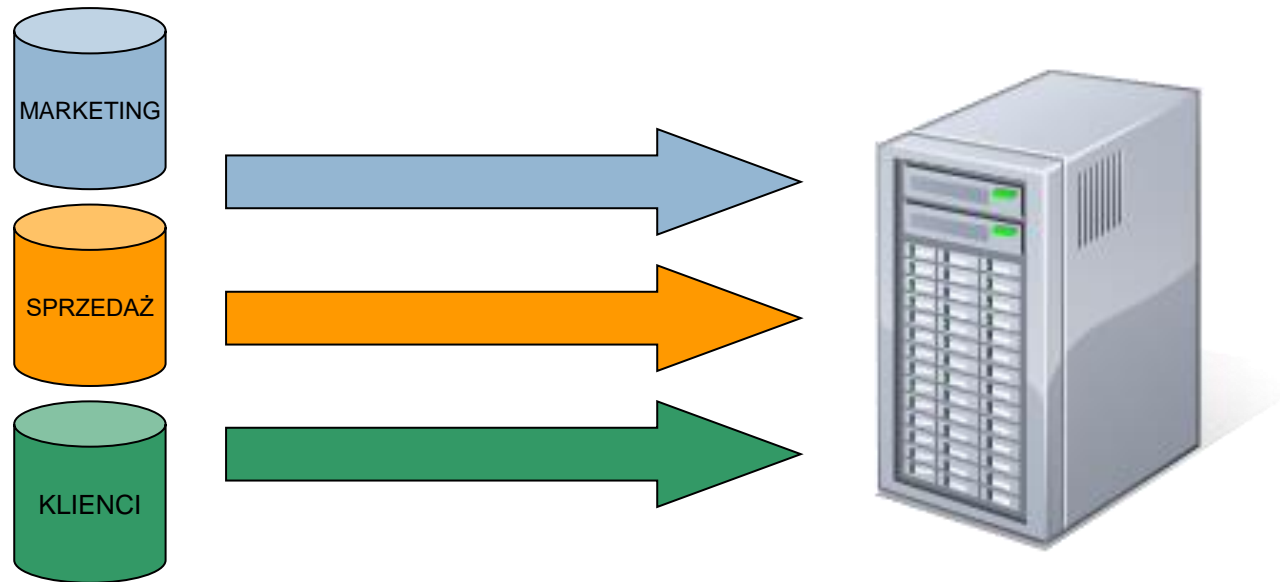
Zakładanie nowej bazy danych



Baza danych

38

- Na jednym serwerze może istnieć wiele baz danych



Tworzenie nowej bazy danych

39

1.

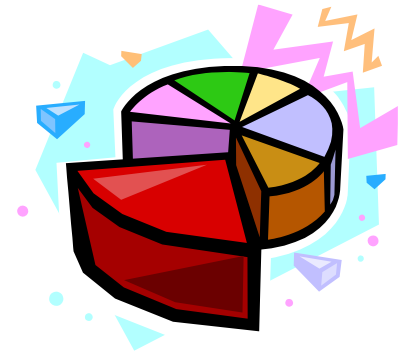
```
CREATE DATABASE nazwa_bazy_danych
[ ON
    [ < plik > [ ,...n ] ]
    [ , < grupa_plików > [ ,...n ] ]
]
[ LOG ON { < plik > [ ,...n ] } ]
[ COLLATE nazwa ]
[ FOR LOAD | FOR ATTACH ]
```
2.

< plik > ::=
3.

```
[ PRIMARY ]
( [ NAME = logiczna_nazwa_pliku , ]
  FILENAME = 'nazwa_pliku_w_systemie_operacyjnym'
  [ , SIZE = rozmiar ]
  [ , MAXSIZE = { maksymalny_rozmiar | UNLIMITED } ]
  [ , FILEGROWTH = wielkośc_skoku_rozmiaru ] ) [ ,...n ]
```
4.

< grupa_plików > ::=
5.

```
FILEGROUP nazwa_grupy_plików < plik > [ ,...n ]
```



Tworzenie prostej bazy danych

40



1. USE master
2. GO
3. **CREATE DATABASE Studenci ON (NAME = studenci_dat,
FILENAME = 'd:\bazy\studenci.mdf',**
4. **SIZE = 4,**
5. **MAXSIZE = 10,**
6. **FILEGROWTH = 1)**
7. GO

Nazwa ▲	Rozmiar	Typ	Data modyfikacji
 studenci.mdf	4 096 KB	SQL Server Databa...	2011-04-04 15:35
 studenci_log.LDF	1 024 KB	SQL Server Databa...	2011-04-04 15:35

Przykład nowej bazy danych



41

1. USE master
2. GO
3. CREATE DATABASE Absolvenci
ON
4. (NAME = Absolvenci_dat,
5. FILENAME =
'd:\bazy\absolvenci.mdf',
6. SIZE = 10,
7. MAXSIZE = 50,
8. FILEGROWTH = 5)
8. LOG ON (NAME =
'Absolvenci_log',
9. FILENAME =
'd:\bazy\absolvenci.ldf',
10. SIZE = 5MB,
11. MAXSIZE = 25MB,
12. FILEGROWTH = 5MB)
13. GO

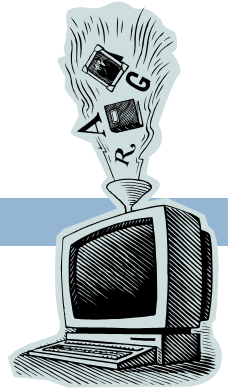
CIĄG DALSZY SKRYPTU →

Nazwa	Rozmiar	Typ	Data modyfikacji
absolvenci.ldf	5 120 KB	SQL Server Databa...	2011-04-04 15:37
absolvenci.mdf	10 240 KB	SQL Server Databa...	2011-04-04 15:37
studenci.mdf	4 096 KB	SQL Server Databa...	2011-04-04 15:36
studenci_log.LDF	1 024 KB	SQL Server Databa...	2011-04-04 15:36

Usunięcie bazy danych

42

- ❑ `DROP DATABASE nazwa_bazy_danych;`
- ❑ Usunięcie bazy danych skutkuje usunięciem plików z danymi.

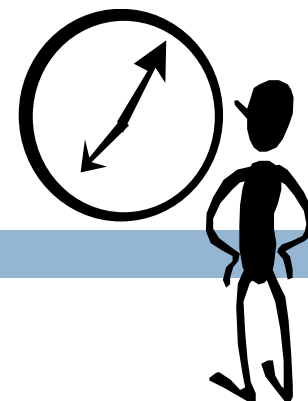


Formatowanie daty



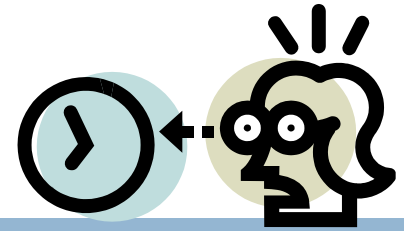
Funkcje daty (1 / 3)

44



- GETDATE() – zwraca bieżącą datę
- DATEADD(interwał, liczba, data) – zwiększenie daty o odpowiednio: dd – dni, mm – miesiące, yy – lata

Funkcje daty (2/3)



45

- DATENAME(jednostka czasu, data),
gdzie jednostka to dd, dw, mm, yy

- DAY(data)
- MONTH(data)
- YEAR(data)

Funkcje daty (3/3)



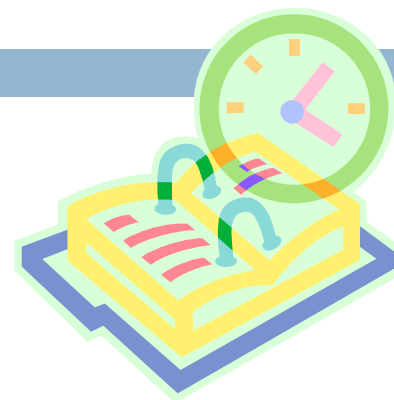
46

- DATEDIFF(jednostka czasu, data1, data2) – zwraca różnicę pomiędzy dwoma datami
- ISNULL(kolumna, 'wyświetlany tekst') – wypisuje wartość, jeżeli na pozycji kolumna jest ona pusta

Przykład – jaki będzie efekt działania?

47

1. DECLARE
2. @data Date
3. BEGIN
4. SET @data='2011-04-05';
5. SELECT GETDATE();
6. SELECT DAY(@data) AS Dzień, MONTH(@data) AS Miesiąc, YEAR(@data) AS Rok;
7. SELECT DATEADD(mm, 14, '2011-04-05') AS Za_14_miesięcy;
8. SELECT DATEDIFF(dd, @data, '2012-06-10') AS Interwał;
9. SELECT DATENAME(MONTH, @data) AS Miesiąc;
10. SELECT ISNULL(NULL, @Data) AS Z_NULLem, ISNULL('2011-04-03',@Data) AS Bez_NULLa;
11. END



Rozwiązanie

48



5		(No column name)		
	1	2011-04-03 16:16:18.360		
6		Dzień	Miesiąc	Rok
	1	5	4	2011
7		Za_14_miesięcy		
	1	2012-06-05 00:00:00.000		
8		Interwał		
	1	432		
9		Miesiąc		
	1	April		
10		Z_NULLem	Bez_NULLa	
	1	2011-04-05	2011-04-03	

Typy tablicowe



Typy tablicowe



50

- Typy tablicowe pozwalają na symulowanie tablic znanych z języków programowania w kodzie T-SQL.
- Dostęp do tablicy zdefiniowanej jako typ tablicowy jest identyczny jako do typowej tabeli SQL.

Deklaracja i usuwanie typów tablicowych



51

- Deklaracja
 - ▣ `CREATE TYPE nazwa_typu_tablicowego AS TABLE (lista_atrybutów)`
- Usuwanie
 - ▣ `DROP TYPE nazwa_typu_tablicowego`
- Przykłady:
 - ▣ `CREATE TYPE typ_tablicowy AS TABLE (id int, wartosc varchar(2) NOT NULL PRIMARY KEY)`
 - ▣ `CREATE TYPE typ_tablicowy6 AS TABLE (id int, wartosc varchar(6) NOT NULL PRIMARY KEY)`

Przykład używania typów tablicowych

- co zostanie wyświetlone?

52



```
1. DECLARE
2.   @województwo typ_tablicowy,
3.   @licznik INT,
4.   @wo VARCHAR(2)
5. BEGIN
6.     INSERT @województwo(id, wartosc) VALUES
7.       (1,'Polska'),(2,'pomorskie'),(3,'zachodniopomorskie'),(4,'kujawsko-pomorskie');
8.     SET @licznik=1;
9.     WHILE @licznik<=4
10.      BEGIN
11.        SELECT @wo=wartosc from @województwo where id=@licznik;
12.        SET @licznik+=1;
13.      END;
14.    PRINT @wo;
15.  END
```

Pytanie?

53

- Rola serwera, pozwalająca członkom na wykonywanie instrukcji wstawiających dane to:
 - ▣ ProcessAdmin
 - ▣ DbCreator
 - ▣ DiskAdmin
 - ▣ BulkAdmin

