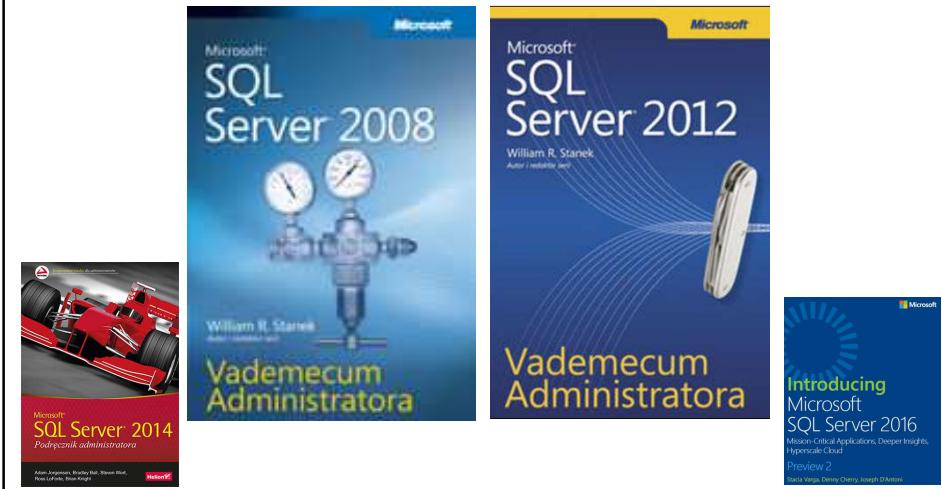


# Projektowanie i Administracja baz danych



1

## Sprawy organizacyjne

dr inż. Henryk Welfle

Instytut Mechatroniki i Systemów Informatycznych  
e-mail: [henryk.welfle@p.lodz.pl](mailto:henryk.welfle@p.lodz.pl)

Godziny przyjęć : sprawdzić na stronie <http://imsi.pl>  
pokój 8 (IMSI)

Zaliczenie części wykładowej na podstawie testu  
przeprowadzonego na ostatnim wykładzie (8 tydzień)  
- 28.11.2025 piątek godzina 14.15

Ocena końcowa =  $\frac{1}{2}$  \*ocena wykład +  $\frac{1}{2}$  ocena laboratorium  
Laboratorium: kolokwium przy komputerach (ocena)  
+ projekt bazy danych(zal.)

2

## Ranking relacyjnych systemów baz danych - RDBMS

Ranking > Relational DBMS [RSS Feed](#)

**DB-Engines Ranking of Relational DBMS**

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

This is a partial list of the [complete ranking](#) showing only relational DBMS.

Read more about the [method](#) of calculating the scores.

[Include secondary database models](#)

Rank			DBMS	Database Model	Score		
Oct 2023	Sep 2023	Oct 2022			Oct 2023	Sep 2023	Oct 2022
1.	1.	1.	Oracle	Relational, Multi-model	1261.42	+20.54	+25.05
2.	2.	2.	MySQL	Relational, Multi-model	1133.32	+21.83	-72.06
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	896.88	-5.34	-27.80
4.	4.	4.	PostgreSQL	Relational, Multi-model	638.82	+18.06	+16.10
5.	5.	5.	IBM Db2	Relational, Multi-model	134.87	-1.85	-14.79
6.	6.	7.	SQLite	Relational	125.14	-4.05	-12.66
7.	7.	6.	Microsoft Access	Relational	124.31	-4.25	-13.85
8.	8.	9.	Snowflake	Relational	123.24	+2.35	+16.51
9.	9.	8.	MariaDB	Relational, Multi-model	99.66	-0.79	-9.65
10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model	80.93	-1.80	-4.03

164 systems in ranking, October 2023

<https://db-engines.com/en/ranking>

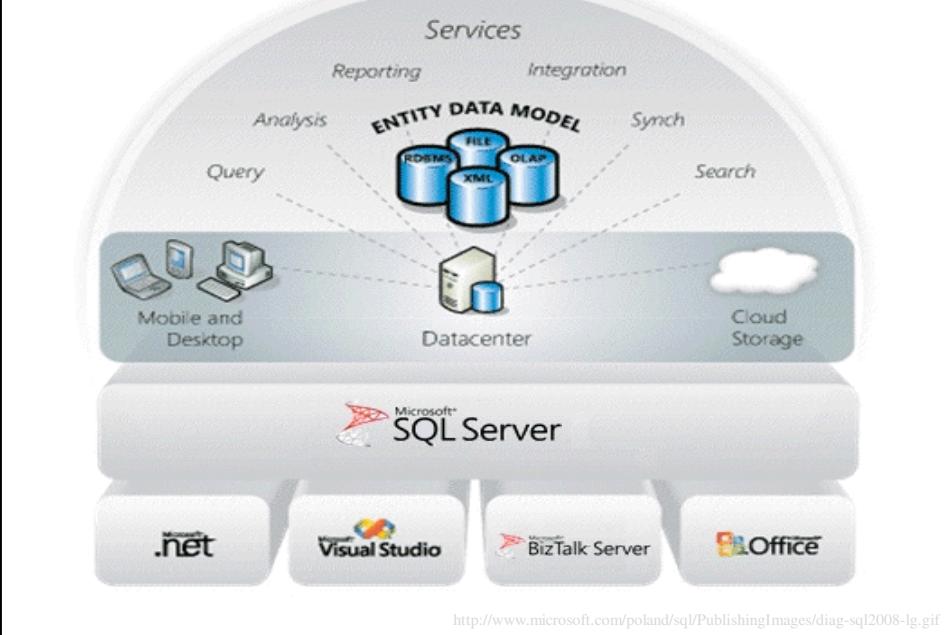
3

## Wersje SQL Server

Wersje	Rok	Nazwa produktowa	Nazwa kodowa
1.0 (OS/2)	1989	SQL Server 1.0 (16 bit)	-
1.1 (OS/2)	1991	SQL Server 1.1 (16 bit)	-
4.21 (WinNT)	1993	SQL Server 4.21	SQLNT
6.0	1995	SQL Server 6.0	SQL95
6.5	1996	SQL Server 6.5	Hydra
7.0	1998	SQL Server 7.0	Sphinx
8.0	2000	SQL Server 2000	Shiloh
8.0	2003	SQL Server 2000 64-bit	Liberty
9.0	2005	SQL Server 2005	Yukon
10.0	2008	SQL Server 2008	Katmai
10.5	2010	SQL Server 2008 R2	Kilimanjaro
11.0	2012	SQL Server 2012	Denali
12.0	2014	SQL Server 2014	Hekaton
13.0	2016	SQL Server 2016	-
14.0	2017	SQL Server 2017	-
15.0	2019	SQL Server 2019	-
16.0	2022	SQL Server 2022	-

4

## Platforma bazodanowa SQL Server



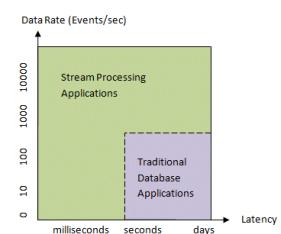
5

## Komponenty SQL Server

- Silnik baz danych (**Database Engine**)
- Usługi integracyjne (**Integration Services**)
- Usługi replikacji danych (**Replication Services**)
- Usługi powiadamiania (**Notification Services**)
- Usługi analityczne (**Analysis Service**)
- Usługi raportowania (**Reporting Services**) – Power BI
- Zapytania pełnotekstowe (**Full-Text Search**)
- Eksploracja danych (**Data Mining**)
- Usługi jakości danych (**Data Quality Services**)
- Usługi danych wzorcowych (**Master Data Services**)
- Przetwarzanie strumieniowe i zdarzeń (**StreamInsight**)
- **Machine Learning Services**
- **Narzędzia do zarządzania**
- **Narzędzia programistyczne**

### Komponenty SQL Server

Database Engine	Analysis Services
Integration Services	Reporting Services
Master Data Services	StreamInsight
Data Mining	Full-Text Search
Data Quality Services	Replication



Przykładowe obszary zastosowań dla systemu StreamInsight:  
 - analiza zagrożeń obejmują aplikacje finansowe,  
 - wykrywanie oszustw,  
 -produkcja,  
 -nadzór ruchu internetowego,  
 -analistyka biznesowej w czasie rzeczywistym.

6

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Silnik baz danych (Database Engine)

W chwili obecnej funkcjonalności dostępne w SQL Server, pozwalają wykorzystać platformę do budowy zarówno małych i średnich baz danych, w których wymagane są podstawowe elementy funkcjonalne (Express, Standard). Natomiast w przypadku dużych baz, gdzie liczy się wydajność (In-Memory OLTP, Column Store Index, Buffer Pool Extension) i bezpieczeństwo (technologia AlwaysOn, kopie bezpieczeństwa w chmurze - Azure Backup) wykorzystujemy edycję Business Intelligence lub Enterprise.

Silnik baz danych SQL Server zawiera silnik bazy danych jako podstawowa usługa do przechowywania, przetwarzania i zabezpieczania danych, replikacji, przeszukiwania pełnotekstowego, narzędzia do zarządzania danymi relacyjnymi, XML czy JSON. Wykorzystywany w integracji z usługami analizy bazy danych i integracji PolyBase w celu dostęp do usługi Hadoop i innych niejednorodnych źródeł danych oraz serwera usług Data Quality Services (DQS).

7

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi replikacji danych (Replication Services)

Obsługa replikacji danych w zastosowaniach takich jak rozproszone lub mobilne aplikacje przetwarzania danych, systemy wysokiej dostępności, skalowalne systemy przetwarzania równoległego z dodatkowymi składnicami danych dla korporacyjnych rozwiązań raportowania czy integracja z systemami heterogenicznymi, w tym z istniejącymi bazami danych Oracle.

8

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi powiadamiania (Notification Services)

Zaawansowane możliwości powiadamiania pozwalające na tworzenie i wdrażanie skalowalnych aplikacji, służących do terminowego dostarczania spersonalizowanych i aktualnych informacji na dowolne przenośne lub podłączone do sieci urządzenie.

9

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi integracyjne (Integration Services)

SSIS, który wprowadzono w SQL Server 2005, z dużym sukcesem zastąpił dostępne w SQL Server 2000 Data Transformation Services (DTS). Nowa odsłona rozwiązań, oparta na .NET Framework jest powszechnie stosowną usługą we wszystkich rozwiązańach, gdzie potrzebna jest interakcja z danymi, ich transformacja. SSIS w obecnej wersji, dostępnej w SQL Server to funkcjonalna i stabilna platforma dla wydajnych rozwiązań ETL, która ze względu na możliwość tworzenia własnych komponentów (.NET) daje bardzo duże możliwości skalowania rozwiązania.

10

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi analityczne (Analysis Service)

Począwszy od wersji 2005 SQL Server intensywny rozwój zaobserwować można także w obszarze funkcjonalności analitycznych dostarczanych wraz z produktem, jakim jest SQL Server. Możliwość wykorzystania wielowymiarowych baz danych OLAP, połączona z nowym wydajnym silnikiem dla rozwiązań działających w trybie Tabular Model (począwszy od wersji 2012, a nawet 2008 R2 (jako element funkcjonalności Power Pivot for SharePoint) sprawiają, że SQL Server stał się jedną z najczęściej wykorzystywanych platform dla budowy rozwiązań analityczno-raportowych. Dodatkowo wprowadzenie w wersji 2012 edycji SQL Server Business Intelligence, spowodowało zwiększenie dostępności tej platformy, która przy bardziej atrakcyjnej cenie (w porównaniu do Edycji Enterprise), gwarantuje olbrzymie możliwości analityczne.

11

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi raportowe (Reporting Services)

Podobnie jak Analysis Services, usługi Raportowe stanowią jeden z kluczowych elementów większości rozwiązań analityczno-raportowych. Stosunkowo proste w budowie i wdrożeniu raporty oparte o SSRS, w połączeniu z możliwością integracji usługi z platformą SharePoint stanowią interesujące rozwiązanie dla wszystkich, zainteresowanych budową efektywnych i efektownych rozwiązań raportowych. Nowsze podejście to usługa [Power BI](#). [Power BI](#) oferuje oparte na chmurze obliczeniowej usługi klasy Business Intelligence (analizy biznesowej), znane jako "Power BI Services", wraz z interfejsem opartym na aplikacji przeznaczonej na PC o nazwie "Power BI Desktop".

12

## Platforma bazodanowa SQL Server od 2008 - komponenty

### Usługi uczenia maszynowego (Machine Learning Services (In-Database))

Usługi Machine Learning Services (In-Database) obsługują rozproszone, skalowalne rozwiązania uczenia maszynowego wykorzystujące źródła danych przedsiębiorstwa. W SQL Server 2016 obsługiwany był język R. SQL Server 2017 obsługuje R i Python.

13

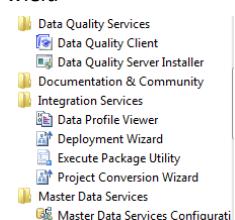
## Platforma bazodanowa SQL Server od 2008 - komponenty

Popularne elementy SQL Server rozszerzone zostały o dodatkowe komponenty (w zależności od edycji) pozwalające na:

- **Kontrolę jakości danych - Data Quality Service (DQS).** Usługa, która pozwala z wykorzystaniem prostych interfejsów użytkownika zapanować nad jakością danych wykorzystywanych w organizacji.

- **Zarządzanie i kontrolę danych (rozproszonych) – Master Data Services**  
Potrzeby aktualności i dostępności, a także wiarygodności najważniejszych danych rozproszonych wykorzystywanych w różnych procesach biznesowych i serwerach. MDS można skonfigurować do zarządzania dowolną domeną (produkty, klienci, konta) i obejmuje hierarchie, szczegółowe zabezpieczenia, transakcje, wersjonowanie danych i reguły biznesowe.

- **Integrację ze środowiskami firm trzecich** np. CDC for Oracle. W wielu przypadkach niezbędna jest grupa funkcjonalności, umożliwiających efektywną wymianę danych z systemów zewnętrznych.



14

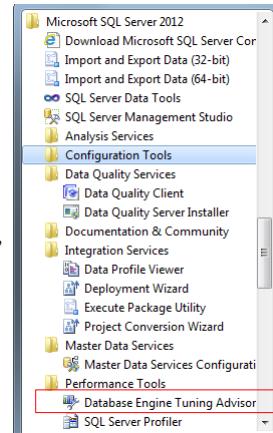
## Platforma bazodanowa SQL Server od 2008 - komponenty

Dodatkowo, niezmiennie wraz z głównymi komponentami funkcjonalnymi, dostarczane są narzędzia umożliwiające efektywne zarządzanie i wspierające optymalizację np.:

- **SQL Server Management Studio** - zintegrowane narzędzie do zarządzania, konfiguracji i tworzenia elementów kodu T-SQL,

- **SQL Server Profiler / SQL Distributed Replay** narzędzia, które w połączeniu z danymi uzyskanymi z **SQL Trace / Performance Monitor** pozwalają na obserwację i ocenę wydajności pracy środowiska,

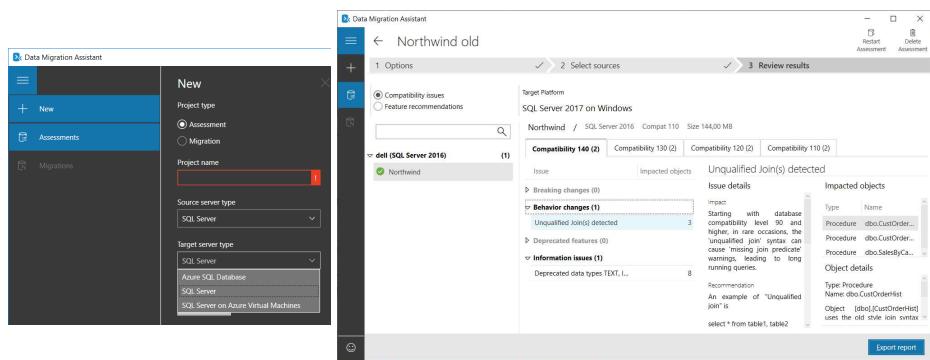
- **Database Engine Tuning Advisor**



15

## Dodatkowe komponenty – Data Migration Assistant

Asystent migracji danych (**DMA**) umożliwia uaktualnienie do nowszej platformy danych przez wykrywanie problemów ze zgodnością, które mogą wpływać na funkcjonalność bazy danych w nowej wersji SQL Server. Zaleca się ulepszenie wydajności i niezawodności dla środowiska docelowego. Umożliwia nie tylko przenoszenie schematu i danych, ale także nieokreślonych obiektów z serwera źródłowego na serwer docelowy.



16

## Cechy systemu SQL Server

### • Łatwość obsługi

- **SQL Server Management Studio (SSMS)** – graficzne narzędzie do zarządzania bazą danych, tworzenia zapytań i administracji.
- **Monitorowanie i dostosowanie wydajności** – funkcje do analizy obciążenia serwera i optymalizacji zapytań.
- **SQL Management Objects (SMO)** – biblioteka obiektów .NET do programowego zarządzania SQL Server.

### • Dostępność

- **Mirroring baz danych** – mechanizm utrzymywania kopii bazy na innym serwerze dla wysokiej dostępności.
- **Konfiguracje klastrowe odporne na awarie (fail-over)** – automatyczne przełączanie na węzeł zapasowy w przypadku awarii.
- **Migawkowa kopia bazy danych** – szybkie tworzenie kopii stanu bazy w danym momencie.
- **Szybkie odtwarzanie** – skrócenie czasu przywracania bazy po awarii.
- **Dedykowane połączenie administracyjne (SQLCMD)** – specjalny kanał do zarządzania serwerem w sytuacjach krytycznych.
- **Operacje w trybie on-line** – możliwość przebudowy indeksów i odtwarzania kopii zapasowych bez zatrzymywania pracy bazy.
- **Replikacja** – kopiowanie danych pomiędzy serwerami w celu synchronizacji lub dystrybucji.

### • Skalowalność

- **Partycjonowanie tabel i indeksów** – dzielenie dużych tabel na mniejsze segmenty dla lepszej wydajności.
- **Izolacja migawkowa** – mechanizm izolacji transakcji bez blokowania odczytów.
- **Baza danych w pamięci (In-Memory OLTP) zoptymalizowana tempdb**.
- **PolyBase i wirtualizacja danych – zapytania na różnych źródłach danych**.

### • Bezpieczeństwo

- **Transparent Data Encryption (TDE)** – szyfrowanie w spoczynku i w ruchu.
- **Dynamiczne maskowanie danych** i zabezpieczenia na poziomie wiersza.
- **SQL Ledger** – niezmienność danych dla scenariuszy audytowych.
- **Obsługa TLS 1.3** i protokołu MS-TDS 8.0

17

## Wersje SQL Server

Edition	Definition
Enterprise	<p>The premium offering, SQL Server Enterprise edition delivers comprehensive high-end datacenter capabilities with blazing-fast performance, unlimited virtualization <sup>1</sup>, and end-to-end business intelligence, enabling high service levels for mission-critical workloads and end-user access to data insights.</p> <p>Enterprise edition is available for evaluation. An evaluation deployment is available for 180 days. For more information, see <a href="#">SQL Server Licensing Resources and Documents</a> .</p>
Standard	<p>SQL Server Standard edition delivers basic data management and business intelligence database for departments and small organizations to run their applications and supports common development tools for on-premises and cloud, enabling effective database management with minimal IT resources.</p>
Web	<p>SQL Server Web edition is a low total-cost-of-ownership option for Web hosters (including choosing Web edition on IaaS on Azure) and Web VAPs to provide scalability, affordability, and manageability capabilities for small to large-scale Web properties.</p>
Developer	<p>Enterprise SQL Server Developer edition lets developers build any kind of application on top of SQL Server. It includes all the functionality of Enterprise edition, but is licensed for use as a development and test system, not as a production server. SQL Server Developer is an ideal choice for people who build and test applications.</p>
Express	<p>SQL Server Express edition is the entry-level, free database and is ideal for learning and building desktop and small server data-driven applications. It is the best choice for independent software vendors, developers, and hobbyists building client applications. If you need more advanced database features, SQL Server Express can be seamlessly upgraded to other higher end versions of SQL Server. SQL Server Express LocalDB is a lightweight version of Express edition that has all of its programmability features, runs in user mode and has a fast, zero-configuration installation and a short list of prerequisites.</p>

18

## Narzędzia do zarządzania SQL Server

Feature	Enterprise	Standard	Web	Express with Advanced Services	Express
SQL Management Objects (SMO)	Yes	Yes	Yes	Yes	Yes
SQL Assessment API	Yes	Yes	Yes	Yes	Yes
SQL Vulnerability Assessment	Yes	Yes	Yes	Yes	Yes
SQL Configuration Manager	Yes	Yes	Yes	Yes	Yes
SQLCMD (Command Prompt tool)	Yes	Yes	Yes	Yes	Yes
SQL Profiler	Yes	Yes	No <sup>1</sup>	No <sup>1</sup>	No <sup>1</sup>
SQL Server Agent	Yes	Yes	Yes	No	No
Microsoft System Center Operations Manager Management Pack	Yes	Yes	Yes	No	No
Database Tuning Advisor (DTA)	Yes	Yes <sup>2</sup>	Yes <sup>2</sup>	No	No

<https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver16>

19

## Programmability

Feature	Enterprise	Standard	Web	Express with Advanced Services	Express
Basic R integration <sup>1</sup>	Yes	Yes	Yes	Yes	No
Advanced R integration <sup>2</sup>	Yes	No	No	No	No
Basic Python integration	Yes	Yes	Yes	Yes	No
Advanced Python integration	Yes	No	No	No	No
Machine Learning Server (Standalone)	Yes	No	No	No	No
Data Virtualization with PolyBase	Yes	Yes	Yes	Yes	Yes
JSON	Yes	Yes	Yes	Yes	Yes
Query Store	Yes	Yes	Yes	Yes	Yes
Query Store on by default for new databases	Yes	Yes	Yes	Yes	Yes
Query Store hint support	Yes	Yes	Yes	Yes	Yes
Query Store read-replica support	Yes	No	No	No	No
Temporal	Yes	Yes	Yes	Yes	Yes
Common Language Runtime (CLR) integration	Yes	Yes	Yes	Yes	Yes
Java Language Runtime Integration	Yes	Yes	Yes	Yes	Yes
Native XML support	Yes	Yes	Yes	Yes	Yes
XML indexing	Yes	Yes	Yes	Yes	Yes
XML compression	Yes	Yes	Yes	Yes	Yes
MERGE and UPSERT capabilities	Yes	Yes	Yes	Yes	Yes
FILESTREAM support	Yes	Yes	Yes	Yes	Yes
FILEtable	Yes	Yes	Yes	Yes	Yes
Date and Time datatypes	Yes	Yes	Yes	Yes	Yes
Time series support	Yes	Yes	Yes	Yes	Yes
Internationalization support	Yes	Yes	Yes	Yes	Yes
Ful-text and semantic search	Yes	Yes	Yes	No	No
Specification of language in query	Yes	Yes	Yes	Yes	No
Service Broker (Messaging)	Yes	Yes	No <sup>3</sup>	No <sup>3</sup>	No <sup>3</sup>
Transact-SQL endpoints	Yes	Yes	Yes	No	No
Graph	Yes	Yes	Yes	Yes	Yes
UTF-8 support	Yes	Yes	Yes	Yes	Yes

<https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver16>

20

10

## Ograniczenia skalowalności SQL Server

Feature	Enterprise	Standard	Web	Express with Advanced Services	Express
Maximum compute capacity used by a single instance - SQL Server Database Engine <sup>1</sup>					
Maximum compute capacity used by a single instance - Analysis Services or Reporting Services					
Maximum memory for buffer pool per instance of SQL Server Database Engine					
Maximum memory-optimized data size per database in SQL Server Database Engine					
Maximum memory utilized per instance of Analysis Services					
Maximum memory utilized per instance of Reporting Services					
Maximum relational database size	524 PB	524 PB	524 PB	10 GB	10 GB

<https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver17>

21

## Ograniczenia wysokiej dostępności SQL Server

Feature	Enterprise	Standard	Web	Express with Advanced Services	Express
Server core support <sup>1</sup>	Yes	Yes	Yes	Yes	Yes
Log shipping	Yes	Yes	Yes	No	No
Database mirroring <sup>7</sup>	Yes	Yes <sup>2</sup>	Yes <sup>3</sup>	Yes <sup>3</sup>	Yes <sup>3</sup>
Backup compression	Yes	Yes	No	No	No
Database snapshot	Yes	Yes	Yes	Yes	Yes
Always On failover cluster instances <sup>4</sup>	Yes	Yes	No	No	No
Always On availability groups <sup>5</sup>	Yes	No	No	No	No
Basic availability groups <sup>6</sup>	No	Yes	No	No	No
Contained availability groups	Yes	No	No	No	No

<https://learn.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2022?view=sql-server-ver17>

22

Cecha	Mirroring	Failover Cluster Instance	Always On AG
Poziom ochrony	Pojedyncza baza	Cała instancja	Grupa baz
Wymaga wspólnego storage	Nie	Tak	Nie
Automatyczny failover	Tak (High Safety)	Tak	Tak
Skalowanie odczytu	Nie	Nie	Tak
Backup na replikach	Nie	Nie	Tak
Status	Deprecated	Aktualne	Aktualne

**Wysoka dostępność (High Availability, HA)  
Odporność na awarie (Disaster Recovery, DR)**

23

Database Mirroring (Mirroring baz danych)	
<b>Mirroring</b> polega na <b>replikacji całej bazy danych w czasie rzeczywistym</b> z jednego serwera ( <b>Principal</b> ) na drugi ( <b>Mirror</b> ). Działa na poziomie pojedynczej bazy danych, a nie całego serwera.	
◆ <b>Jak to jest realizowane</b>	Istnieją dwie (lub trzy) instancje SQL Server:
	<ul style="list-style-type: none"> <li>- <b>Principal</b> – główna baza danych, z której korzystają aplikacje.</li> <li>- <b>Mirror</b> – kopia bazy danych w trybie „standby”, aktualizowana na bieżąco.</li> <li>- <b>Witness</b> (opcjonalny) – trzeci serwer, który monitoruje stan Principal i Mirror i może automatycznie przełączyć ruch w razie awarii.</li> </ul>
◆ <b>Tryby pracy</b>	<b>High Safety (Synchronous)</b> Transakcje są zapisywane jednocześnie na Principal i Mirror. Gwarantuje zerową utratę danych. Może wspierać automatyczny failover (z serwerem Witness).
	<b>High Performance (Asynchronous)</b> Mirror jest aktualizowany z opóźnieniem. Lepsza wydajność, ale możliwa utrata danych przy awarii.
◆ <b>Zalety</b>	Szybkie przełączenie po awarii. Chroni przed utratą danych (w trybie synchronous).
	Łatwa konfiguracja.
◆ <b>Wady</b>	Działa tylko dla <b>jednej bazy na raz</b> (nie obejmuje całego serwera). Wymaga, aby <b>obie bazy miały identyczną strukturę</b> . Od SQL Server 2012 funkcja została <b>zastąpiona przez Always On</b> (choć wciąż działa w trybie kompatybilności).

24

## Failover Clustering (SQL Server Failover Cluster Instance – FCI)

**Failover** clustering chroni całą instancję SQL Server, a nie tylko pojedynczą bazę danych.

Jest to rozwiązańe sprzętowo-systemowe oparte na Windows Server Failover Cluster (WSFC).

### ◆ Jak to jest realizowane

Mamy kilka węzłów (nodes) w klastrze Windows.

Instancja SQL Server jest zainstalowana jako usługa klastrowa.

W danym momencie aktywny jest tylko jeden węzeł (active node), a pozostałe są w trybie pasywnym.

Gdy aktywny serwer ulegnie awarii, WSFC automatycznie przenosi zasoby (dyski, IP, nazwę instancji) na inny węzeł.

### ◆ Charakterystyka

Wspólny magazyn danych (shared storage) – wszystkie węzły korzystają z tych samych plików bazy.

Nie zapewnia replikacji danych – tylko redundancję usług.

Użytkownicy łączą się zawsze z tą samą nazwą instancji (adres IP wirtualny).

### ◆ Zalety

Chroni całą instancję SQL Server (wszystkie bazy, loginy, ustawienia).

Automatyczne przełączenie (failover).

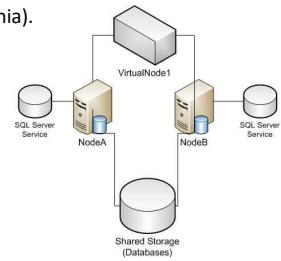
Brak utraty danych (bo dane są wspólne).

### ◆ Wady

Wymaga dostępu do współdzielonego magazynu (SAN, iSCSI).

Nie zapewnia kopii geograficznej (bo to nie replikacja).

Nie jest rozwiązaniem Disaster Recovery, tylko High Availability.



25

## Always On Availability Groups (AG)

To nowoczesny następca mirroringu, dostępny od SQL Server 2012 Enterprise Edition.

Działa na poziomie grupy baz danych, a nie pojedynczej instancji.

### ◆ Jak to jest realizowane

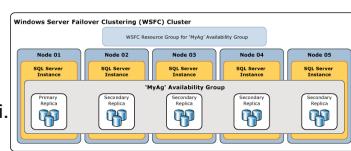
Tworzymy tzw. Availability Group, która zawiera kilka baz danych.

Grupa ta jest synchronizowana pomiędzy replikami (instancjami SQL Server):

Primary replica – główna instancja, przyjmująca zapisy.

Secondary replicas – kopie (maks. 8), które mogą:

działać synchronicznie lub asynchronicznie,  
być tylko do odczytu (np. raporty, backupy),  
automatycznie przejmować rolę Primary przy awarii.



### ◆ Tryby pracy

Synchronous commit – transakcje zapisywane na obu replikach przed potwierdzeniem (HA).

Asynchronous commit – replikacja z opóźnieniem (DR, między odległymi lokalizacjami).

### ◆ Zalety

Wysoka dostępność + Disaster Recovery w jednym rozwiązaniu.

Obsługuje automatyczny failover (dla synchronous replicas).

Możliwość odczytu z replik wtórnych (redukcja obciążenia).

Brak potrzeby wspólnego magazynu (każdy serwer ma własne dane).

Integracja z Windows Clusterem (WSFC), ale bez wspólnego storage'u.

### ◆ Wady

Tylko w SQL Server Enterprise Edition (choć w nowszych wersjach część funkcji dostępna w Standard).

Wymaga Active Directory i WSFC.

Konfiguracja bardziej złożona niż mirroring.

26

## Licencjonowanie SQL Server

Platforma SQL Server licencjonowana jest w dwóch podstawowych modelach:

- **Licencja "na serwer"**, która wymagana jest na każde urządzenie, na którym instalowane są komponenty serwerowe rozwiązania. Ten model wymaga dodatkowo (poza licencjami na warstwę serwerową), również licencji dostępowych dla użytkowników i/lub urządzeń uzyskujących dostęp do danego wystąpienia komponentu SQL Server (Standard) - **Client Access License (CAL)**
- **Licencja "na rdzeń"**, która począwszy od SQL Server 2012 zastąpiła licencjonowanie „na procesor” i wg. której, licencję należy nabyć na każdy rdzeń fizyczny (bądź wirtualny) urządzenia, na którym zainstalowano komponent SQL Server. W tym modelu nie są wymagane licencje dostępowe (CAL) dla użytkowników i/lub urządzeń uzyskujących dostęp do wystąpienia serwera SQL.

TRZY GŁÓWNE EDYCJE	MODEL LICENCJONOWANIA	
	Serwer + CAL (Client Access License)	Model na rdzeń
Enterprise		☒
Business Intelligence	☒	
Standard	☒	☒

### How to license

1. License all of the physical cores on the hardware
2. A minimum of 4 core licenses are required per physical processor



PHYSICAL CORES IN THE PROCESSOR:	1	2	4	6	8
CORE LICENSES REQUIRED	4	4	4	6	8

27

## Separacja i dostępność usług/funkcjonalności



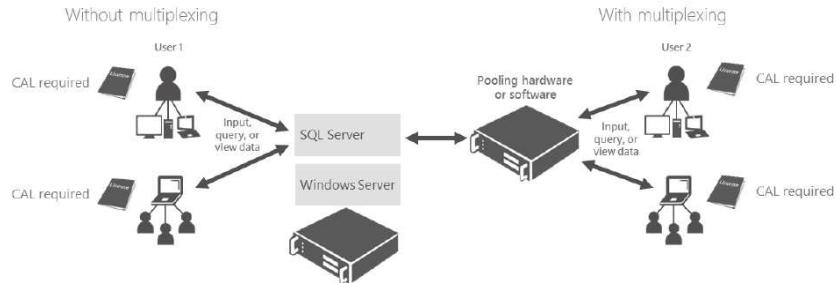
W przypadku instalacji wszystkich komponentów serwera SQL na jednym serwerze sposób licencjonowania nie zmienia się

W przypadku, gdy poszczególne komponenty serwera instalowane są na oddzielnych serwerach tzn. SQL Server DB Engine na jednym serwerze, Analysis Services na drugim, a Reporting Services nathirdem, licencje potrzebne są na każdy z ww. serwerów



28

## Wymagania w zakresie dostępu użytkowników do danych



Zastosowanie oprogramowania/urządzenia, których użycie zmniejsza ilość bezpośrednich połączeń do serwera SQL, nie zmniejsza natomiast ilości wymaganych licencji dostępowych (CAL)

29

## Wprowadzenie do SQL Server

Co to jest serwer SQL

Architektura serwera SQL

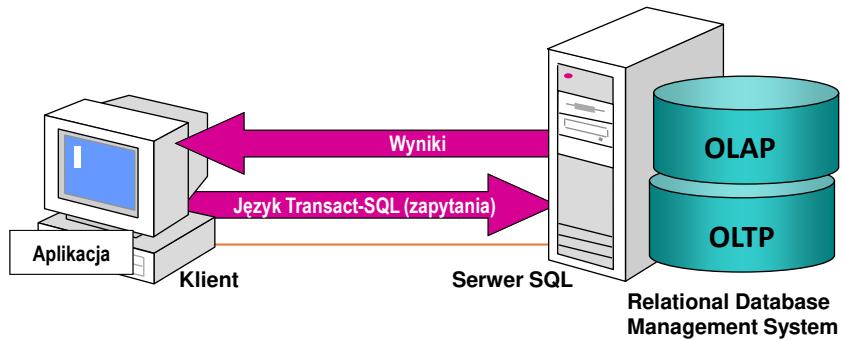
Zabezpieczenia serwera SQL

Bazy danych serwera SQL

Praca z serwerem SQL

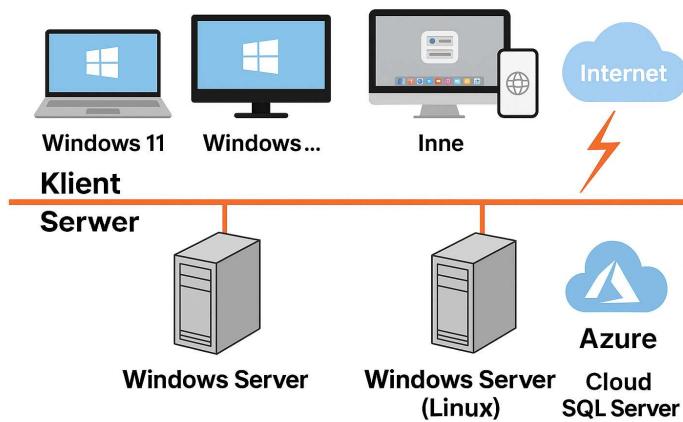
30

## Co to jest SQL Server

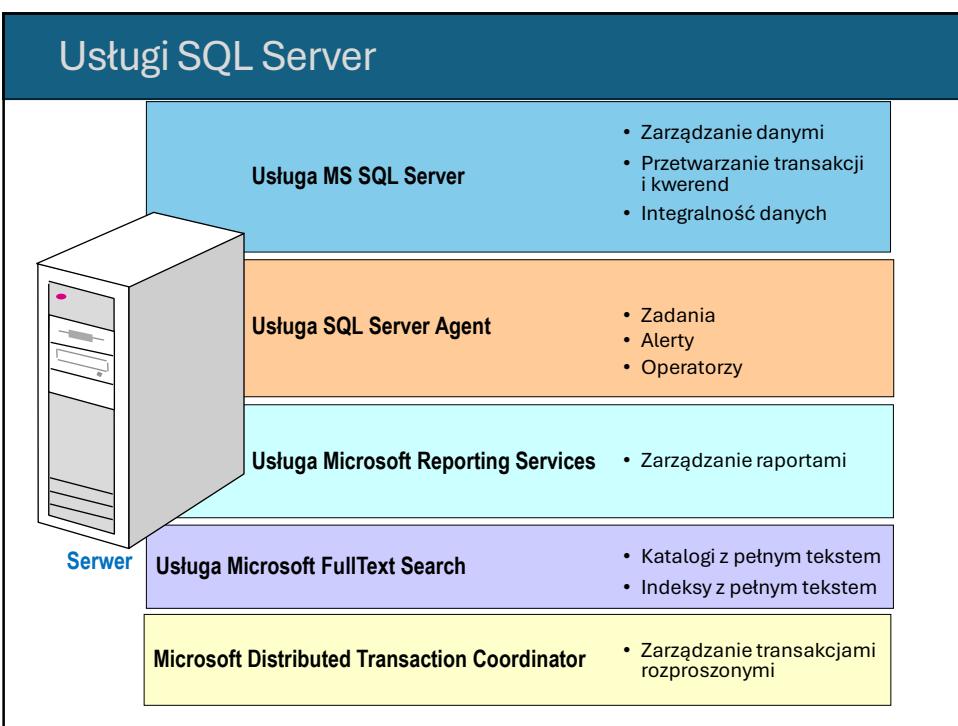


31

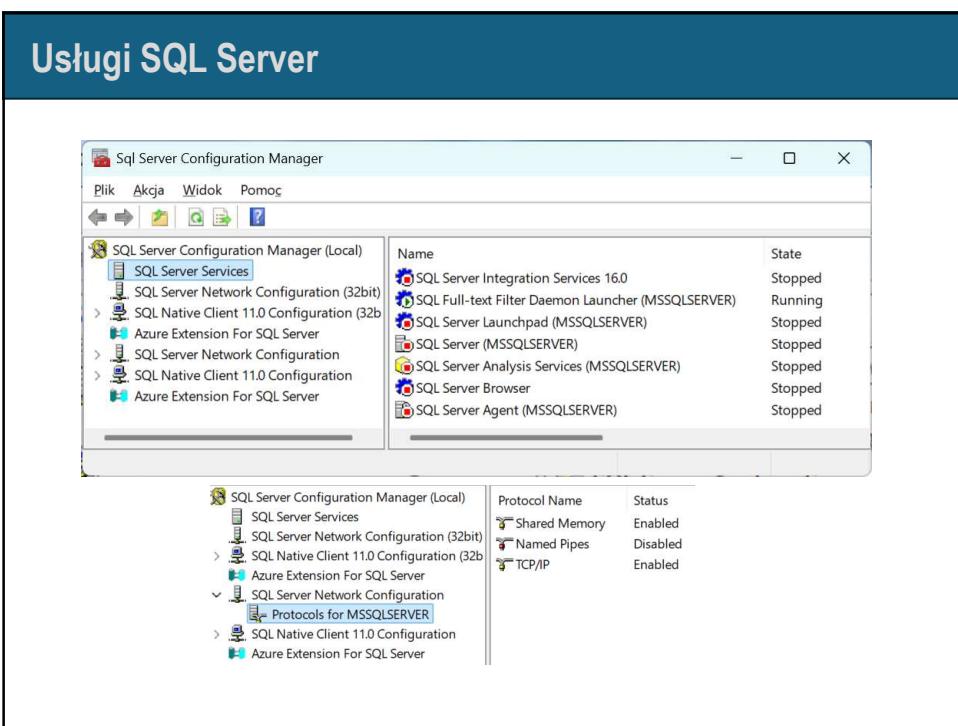
## Platformy SQL Server



32



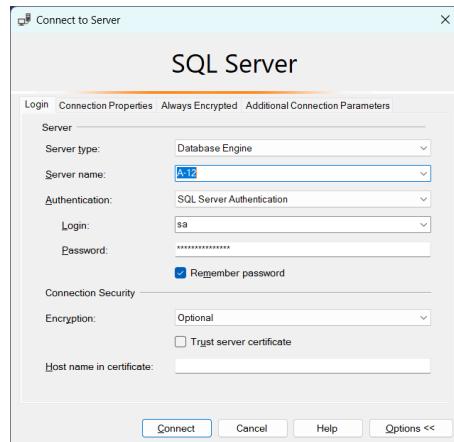
33



34

## Oprogramowanie serwera SQL

- Program SQL Server Management Studio
- Narzędzia administracyjne i kreatorzy
- Narzędzia zarządzania dostępne w wierszu polecenia
- Pomoc i podręczniki SQL Server Documentation – Microsoft Learn
- Usługi OLAP serwera Microsoft SQL Server
- ...



35

## Architektura SQL Server

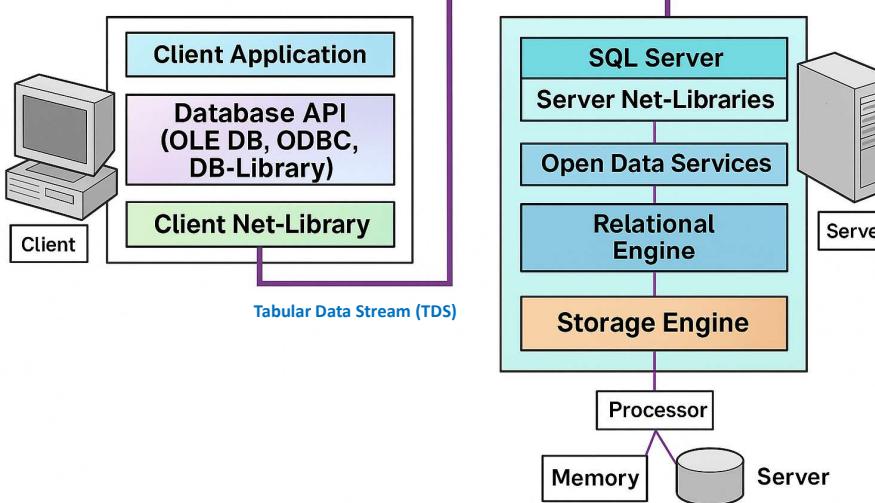
Komunikacja

Opracowanie aplikacji

Administracja

36

## Komunikacja



37

## Opracowanie aplikacji

Aktualny model komunikacji w SQL Server :

### Warstwa aplikacji:

- .NET (ADO.NET) – główny interfejs dla aplikacji .NET,
- Entity Framework Core (ORM) – popularny framework do mapowania obiektowo-relacyjnego,
- JDBC – dla aplikacji Java,
- ODBC – nadal używany w wielu środowiskach,

### Interfejsy API i protokoły:

- OLE DB – wciąż dostępny,
- ODBC – standardowy interfejs dla wielu języków,
- REST API / OData – dla usług webowych i integracji z aplikacjami w chmurze,
- TDS (Tabular Data Stream) – protokół komunikacyjny SQL Server.

### Źródła danych:

- SQL Server (on-premises),
- Azure SQL Database,
- Synapse Analytics,
- Inne źródła danych (np. pliki, API, Big Data).

### Nowe technologie:

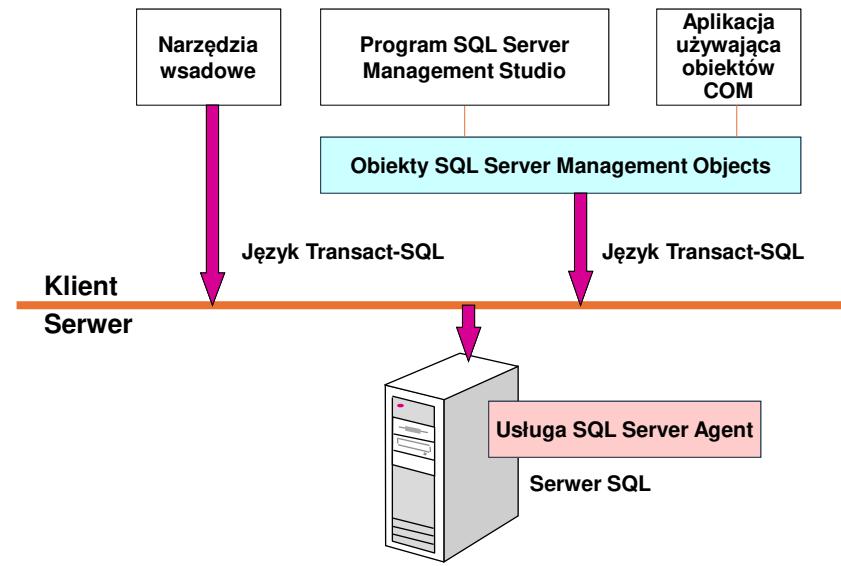
- PowerShell + SMO – do automatyzacji zarządzania,
- SQLCMD – narzędzie wiersza poleceń,
- Data APIs – np. dla Power BI, Analysis Services.

38

## Administracja

- Automatyzacja zadań
- Zarządzanie obiektami
- Monitorowanie i raportowanie

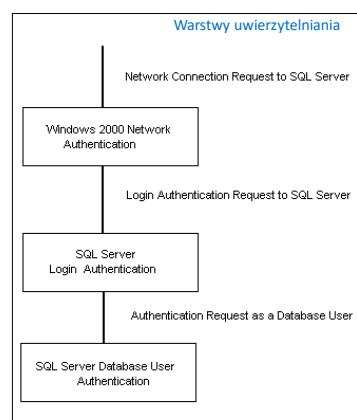
COM (Component Object Model)  
SMO (SQL Server Management Objects)



39

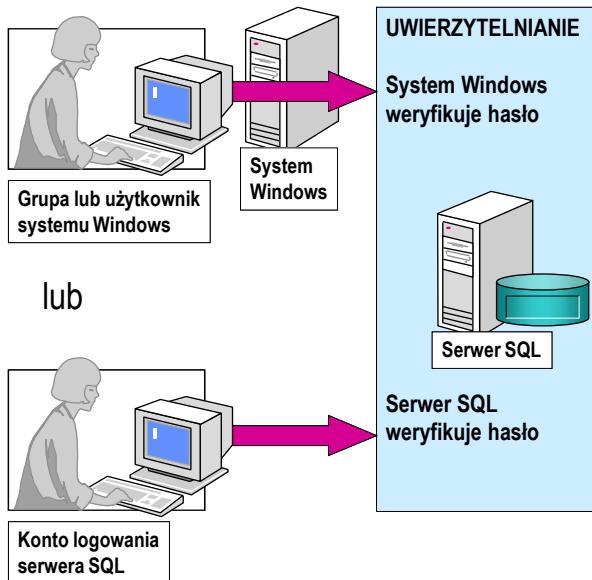
## Zabezpieczenia serwera SQL

- Uwierzytelnianie logowania ([Login Authentication](#))
- Typy ról serwera ([Types of Roles](#))
- Konta użytkowników baz danych i role ([Database User Accounts and Roles](#))
- Sprawdzanie poprawności uprawnień ([Permission Validation](#))



40

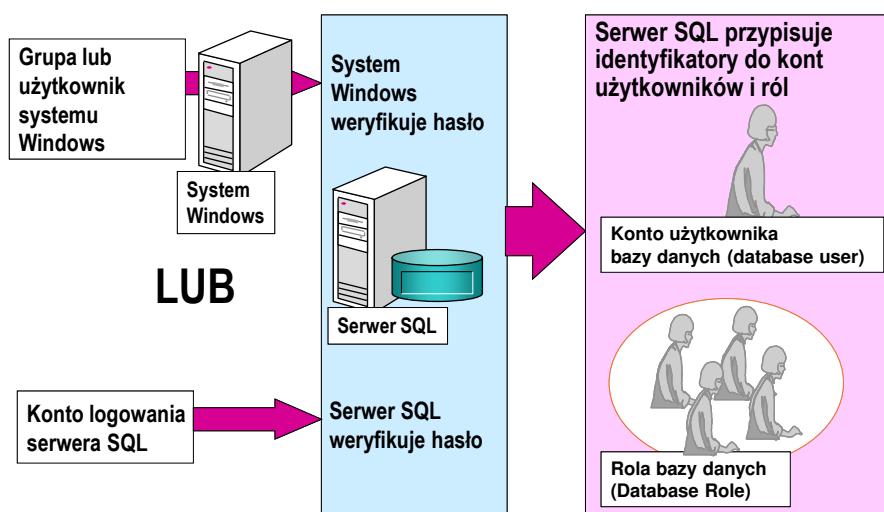
## Uwierzytelnianie logowania (Login Authentication)



41

## Konta użytkowników baz danych i role

(Database User Accounts and Roles)



42

## Typy ról (Types of Roles)

### Role serwera

- Grupa uprawnień administracyjnych na poziomie serwera

### Role serwera zdefiniowane przez użytkownika

### Role bazy danych

- Grupa uprawnień administracyjnych na poziomie bazy danych

### Role bazy danych zdefiniowane przez użytkownika

### Role aplikacyjne bazy danych

43

## Stałe role serwera

### Role [Uprawnienia]

- Database creators ([dbcreator](#)) [Tworzenia i modyfikowanie baz danych]
- Disk administrators ([diskadmin](#)) [Zarządzanie plikami na dysku]
- Process administrators ([processadmin](#)) [Zarządzanie procesami SQL Server]
- Security administrators ([securityadmin](#)) [Zarządzanie i audyt logowania serwera]
- Server administrators ([serveradmin](#)) [Konfiguracja ustawień dla całego serwera]
- Setup administrators ([setupadmin](#)) [Instalacja replikacji]
- System administrators ([sysadmin](#)) [Pełne uprawnienia]
- Bulk administrators ([bulkadmin](#)) [Wykonywanie składni BULK INSERT]
- Public ([public](#)) – rola domyślna. Członkowie widzą bazy danych w instancji

44

22

## Stałe role bazy danych

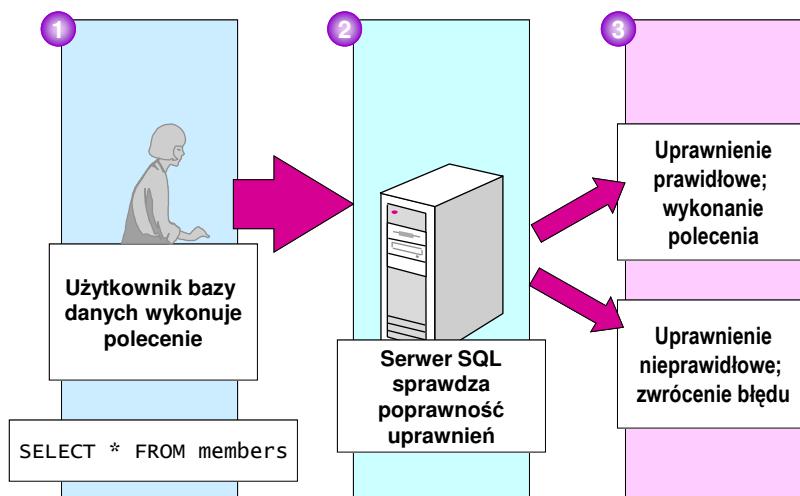
### Role [Uprawnienia]

- **public** [domyślne uprawnienia dla użytkowników bazy danych]
- **db\_owner** [właściciel bazy danych]
- **db\_accessadmin** [dodawanie lub usuwanie użytkowników b.d., grupy i roli]
- **db\_ddladmin** [dodawanie, modyfikowanie lub usuwanie obiektów b.d.]
- **db\_securityadmin** [Przypisywanie uprawnień do obiektów]
- **db\_backupoperator** [Backup bazy danych]
- **db\_datareader** [Odczyt danych z dowolnej tabeli]
- **db\_datawriter** [Dodawanie, modyfikowanie lub usuwanie danych ze wszystkich tabel]
- **db\_denydatareader** [Nie można odczytać danych z wszystkich tabel]
- **db\_denydatawriter** [Nie można zmieniać dane z wszystkich tabel]

45

## Sprawdzanie poprawności uprawnień

(Permission Validation)



46

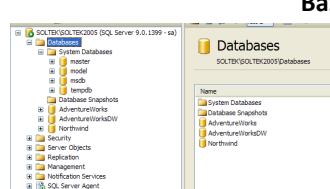
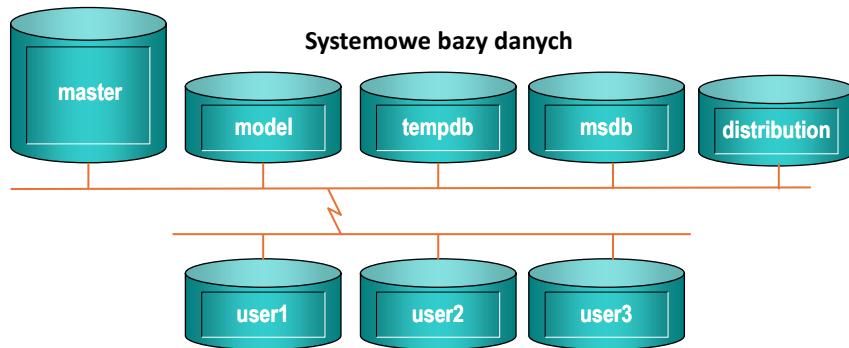
## Bazy danych SQL Server

- Typy baz danych
- Obiekty baz danych
- Tabele systemowe
- Pobieranie metadanych

47

### Typy baz danych

+ baza systemowa niewidoczna **RESOURCE**, która jest tylko do odczytu i przechowuje obiekty systemowe, które są odwzorowane na schemat **SYS** we wszystkich bazach danych.



48

## Systemowe bazy danych

- **Baza danych master**

Kluczowa baza danych dla uruchomionego serwera SQL Server. Zawiera wskaźnik do podstawowego pliku z danymi dla każdej następnej bazy danych instalowanej w systemie jak również kluczowe informacje na temat serwera. Informacja ta zawiera takie elementy jak komunikaty błędów, informacje o logowaniu, systemowe procedury składowane i serwery połączzone.

- **Baza danych model**

Baza danych model jest szablonem bazy danych. Za każdym razem, tworząc nową bazę danych, kopiwana jest baza model, a następnie są dokonywane zmiany np.: rozmiaru, wymagane dla nowej bazy danych. Dlatego, każdy obiekt istniejący w bazie danych model jest kopowany do nowopowstającej bazy danych.

49

## Systemowe bazy danych

- **Baza danych tempdb**

Baza danych tempdb to miejsce, gdzie wykonywane są operacje wymagające przestrzeni tymczasowej, jak np.: sortowania, złączenia i inne. Baza ta (podobnie jak inne domyślne bazy serwera SQL Server) może się zwiększyć, gdy okaże się, że potrzeba więcej przestrzeni. Baza tempdb jest ponownie inicjalizowana za każdym razem gdy SQL Server (usługa SQL Server) jest uruchamiany.

- **Baza danych msdb**

Baza danych msdb obsługuje usługę SQL Server Agent, włączając w to przechowywanie informacji na temat zadań, zdarzeń i replikacji. W tej bazie danych trzymana jest historia wszystkich kopii zapasowych i wykonywanych działań dotyczących odzyskiwania.

50

## Obiekty baz danych

						<b>Indeks grupowany</b>		<b>Procedura przechowywana</b>	<b>Wyzwalacz</b>
						<b>PK</b>	<b>FK</b>		
<b>EmpNum</b>	<b>Lastname</b>	<b>Firstname</b>	<b>CtryCode</b>	<b>Extension</b>	<b>LastMod</b>				
integer	longstring	varchar(20)	char(2)	char(6)	longstring	Anderson Anderson Barr ...		UpdatePhone	
10191	Labrie	Angela	FR	x19891	\HR\KarID				
10192	Labrie	Eva	FR	x19433	\HR\KarID				
10202	Martin	Jose	SP	x21467	\HR\AmyL				

**EmployeeName**

```
SELECT lastname, firstname
FROM emp
```

**Sprawdzanie**

```
x#####
```

51

## Odwoływanie się do obiektów SQL Server

- Pełne nazwy kwalifikowane
  - serwer.baza\_danych.właściciel.obiekt** (SQLServer 2000)
  - serwer.baza\_danych.schemat.obiekt** (od SQLServer 2008)
- Nazwy częściowo określone
  - Jako nazwa serwera przyjmowana jest domyślnie nazwa lokalnego serwera
  - Jako nazwa bazy danych przyjmowana jest domyślnie nazwa bieżącej bazy danych
  - Jako nazwa właściciela przyjmowana jest domyślnie nazwa użytkownika w bazie danych (lub nazwa schematu domyślnego)

SQL Server name	SQL-92 name
Database	catalog
Owner	schema
Object	object

```
CREATE TABLE northwind.dbo.order_history
```

52

## Tabele systemowe

- Przechowują informacje (metadane) dotyczące systemu i obiektów baz danych
- Katalog bazy danych przechowuje metadane dotyczące określonej bazy danych
- Katalog systemowy przechowuje metadane dotyczące całego systemu i wszystkich innych baz danych

53

## Pobieranie metadanych

- Systemowe procedury przechowywane

```
Exec sp_help employees
```

- Funkcje systemowe

```
SELECT USER_NAME(10)
```

- Widoki schematów informacji

```
SELECT * FROM INFORMATION_SCHEMA.TABLES
```

54

## Widoki schematów informacji

Information schema view	Description
<b>INFORMATION_SCHEMA.TABLES</b>	List of tables in the database
<b>INFORMATION_SCHEMA.COLUMNS</b>	Information on columns defined in the database
<b>INFORMATION_SCHEMA.TABLE_PRIVILEGES</b>	Security information for tables in the database

55

## Praca z SQL Server

Administrowanie bazą danych SQL Server

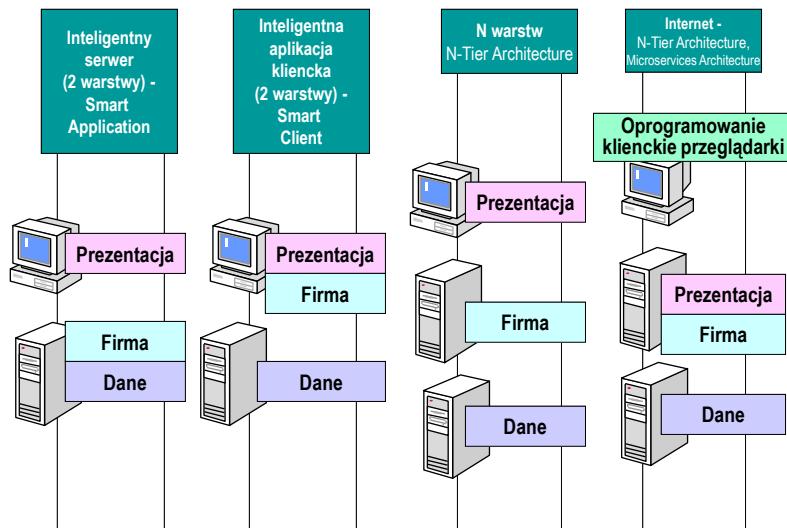
Wdrożenie bazy danych SQL Server  
(implementowanie)

Wybór architektury aplikacji dla SQL Server

Projektowanie aplikacji z wykorzystaniem  
API dla bazy danych

56

## Projektowanie aplikacji dla SQL Server



57

## Implementowanie bazy danych SQL Server

- Projektowanie bazy danych
- Tworzenie bazy danych i obiektów bazy danych
- Testowanie i dopracowywanie aplikacji i bazy danych
- Planowanie zastosowania
- Administrowanie aplikacją po zastosowaniu

58

## Administrowanie bazą danych SQL Server

- Instalowanie serwera SQL
  - Tworzenie zabezpieczeń sieciowych
  - Konfigurowanie serwera SQL
- Tworzenie baz danych
- Zarządzanie bieżącymi działaniami
  - Importowanie i eksportowanie danych
  - Wykonywanie kopii zapasowej i przywracanie bazy danych oraz dziennika
  - Monitorowanie i dopracowywanie bazy danych



# Instalacja i konfiguracja SQL Server

1

## Planowanie systemu

Oto wybrane, niezbędne zadania i aspekty, które trzeba uwzględnić w trakcie planowania systemu bazodanowego.

- Ustalenie obecnego obciążenia systemu,
- Oszacowanie wzrostu obciążenia,
- Ustalenie minimalnych wymagań sprzętowych i programowych,
- Właściwe ustawienie pamięci dla systemu i ustalenie wymagań dla operacji wejścia-wyjścia,
- Ustalenie odpowiedniej edycji systemu SQL Server,
- Określenie kolacji (ang. collation), lokalizacji plików i wielkości bazy tempdb dla systemu SQL Server,
- Wybranie kont powiązanych z usługami,
- Opracowanie planu konserwacji i tworzenia kopii zapasowych bazy danych,
- Ustalenie minimalnych poziomów czasu nieprzerwanej pracy i czasu reakcji,
- Przygotowanie strategii przywracania stanu po awarii.

Microsoft SQL Server 2014 Podręcznik administratora  
A.Jorgensen, B.Ball, S.Wort, R.LoForte, B.Knight. Helion 2015

2

## Połączenie SQL Server

- Używanie narzędzia graficznego
  - SQL Server Enterprise Manager (do wersji 2000)
  - Microsoft SQL Server Management Studio (od wersji 2005)
- Użycie narzędzia poprzez program osql(2000,2005,2008)  
lub sqlcmd(od 2005) (opcja -A połączenie w trybie Admin)
  - Komunikowanie się przy użyciu ODBC

Np. C:\sqlcmd  
-S ACER1\MSSQLSERVER  
-U sa -P hw  
-Q  
"select productname from products"  
-d northwind

```
C:\>xsql  
Microsoft (R) SQL Server Command Line Tool  
Version 12.0 NT  
Copyright (c) 2014 Microsoft. All rights reserved.  
  
Note: osql does not support all features of SQL Server 2014.  
Use sqlcmd instead. See SQL Server Books Online for details.  
  
usage: osql [ -U login_id ] [ -P password ]  
[ -S server ] [ -H hostname ] [ -E trusted connection ]  
[ -d use database name ] [ -l login timeout ] [ -t query timeout ]  
[ -h headers ] [ -s colseparator ] [ -w columnwidth ]  
[ -q "cmdline query" ] [ -e echo input ] [ -I Enable Quoted Ident ]  
[ -L list servers ] [ -c cmdexec ] [ -D default db name ]  
[ -q "cmdline query" ] [ -Q "cmdline query" and exit ]  
[ -m errorlevel ] [ -V severitylevel ] [ -W remove trailing ]  
[ -u unicode output ] [ -r[0|1] msgs to stderr ] .
```

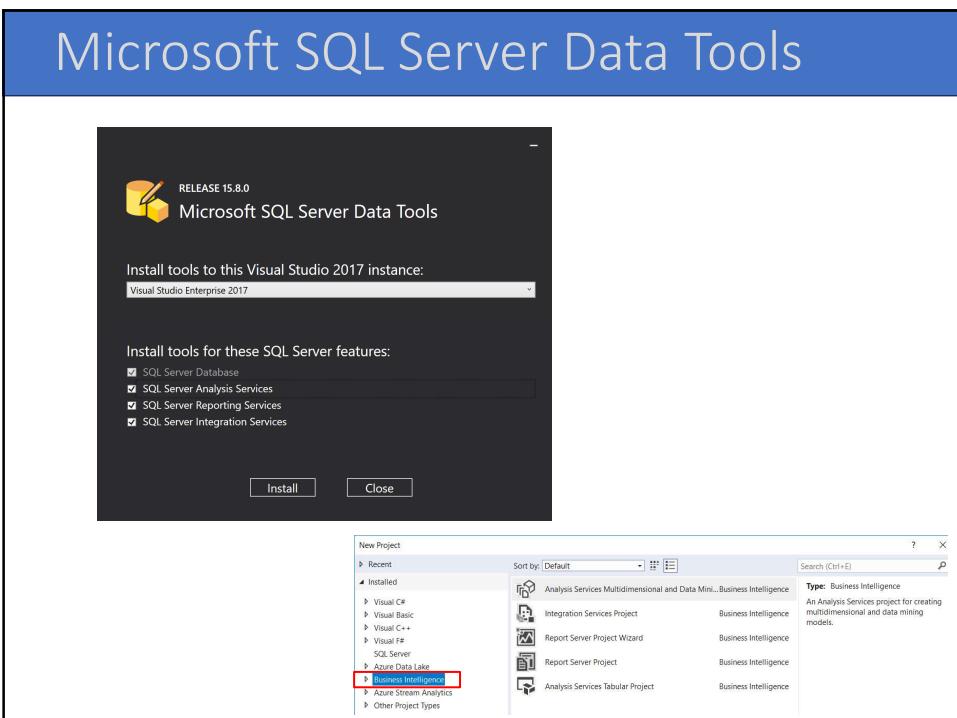
```
C:\>sqlcmd -?  
Microsoft (R) SQL Server Command Line Tool  
Version 12.0.5000.0 NT  
Copyright (c) 2014 Microsoft. All rights reserved.  
  
usage: Sqlcmd [ -U login_id ] [ -P password ]  
[ -S server ] [ -H hostname ] [ -E trusted connection ]  
[ -N Encrypt Connection ][ -C Trust Server Certificate ]  
[ -d use database name ] [ -l login timeout ] [ -t query timeout ]  
[ -h headers ] [ -s colseparator ] [ -w screen width ]  
[ -a packetsize ] [ -e echo input ] [ -I Enable Quoted Ident ]  
[ -c cmdexec ] [ -L[c] list servers[clear output] ]  
[ -q "cmdline query" ] [ -Q "cmdline query" and exit ]  
[ -m errorlevel ] [ -V severitylevel ] [ -W remove trailing ]  
[ -u unicode output ] [ -r[0|1] msgs to stderr ]
```

3

## SQL Server Management Studio

- Zintegrowane środowisko do pracy z SQL Server
  - Wykonywanie zapytań
  - Konfiguracja serwera
  - Zarządzanie serwerem
  - Administracja bazą danych
- Graficzne narzędzie do pracy z wieloma komponentami
  - Połączenie z Database Engine, Analysis Services, Reporting Services, Integration Services, Azure Blob Storage
- Zarządzanie projektami i skryptami (wsparcie Intelisense)

4



5

## Narzędzia

### Narzędzie PowerShell (`Invoke-Sqlcmd`) oraz wiersz poleceń cmd (`Sqlcmd`)

PowerShell i wiersz poleceń (cmd) to oba interfejsy wiersza poleceń w systemie Windows, ale różnią się one językami skryptowymi i możliwościami. Oto podstawowe sposoby interakcji zarówno z PowerShell, jak i Wierszem poleceń (cmd) przy ustawieniu `sp_configure` z SQL Server:

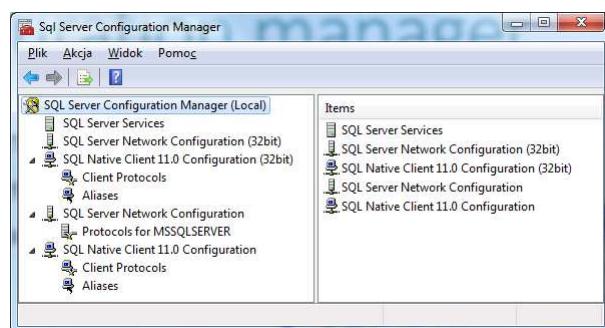
```
PS C:\> Invoke-Sqlcmd -Query "exec sp_configure 'show advanced options', 1; RECONFIGURE" -ServerInstance ...
```

```
C:\>Sqlcmd -q "exec sp_configure 'show advanced options', 1; RECONFIGURE;" Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.
```

6

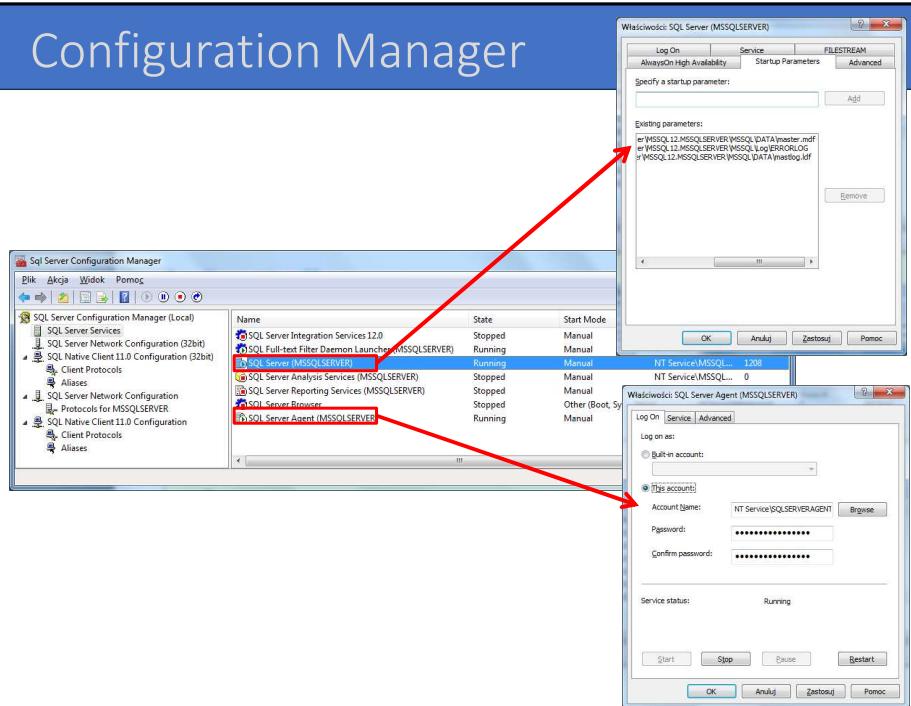
# Configuration Manager

- Protokoły komunikacyjne
  - Shared Memory (na lokalnym komputerze)
  - Named pipes (potoki)
  - TCP/IP
- Porty
- Aliasy

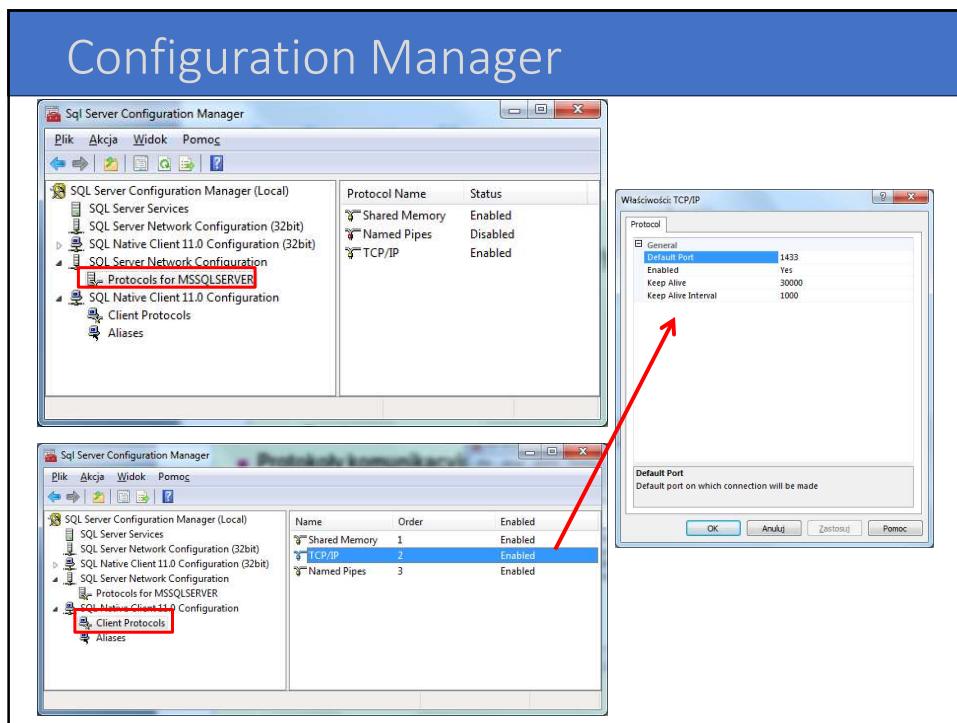


7

# Configuration Manager



8



9

## Ustawianie opcji konfiguracyjnych

Opcje konfiguracyjne mogą być ustawiane dla:

- **instancji serwera** (polecenie `sp_configure`),
- **bazy danych** - poziom bazy danych konfigurowalny poprzez instrukcję `ALTER DATABASE` (poziom zgodności bazy danych `sp_dbcmptlevel`),
- **połączenia** - opcje połączenia są ustawiane podczas zestawiania połączenia za pomocą właściwości dostawcy Microsoft OLE DB dla serwera SQL Server lub sterownika SQL Server ODBC, albo za pomocą opcji `ANSI SET`,
- **polecenia lub partii poleceń** - opcje działające na poziomie partii polecen określane za pomocą instrukcji `SET`.

Każdy z obszarów konfiguracyjnych możemy traktować jako osobny poziom w hierarchii konfiguracji serwera. Jeśli dana opcja jest obsługiwana na więcej niż jednym poziomie, to zastosowana wartość ustawienia zostanie określona przy użyciu następującej kolejności:

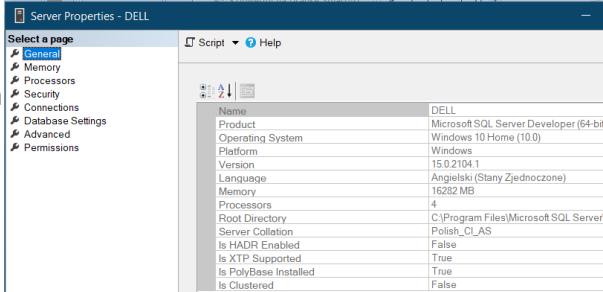
1. *Opcja serwera*
2. *Opcja bazy danych*
3. *Opcja połączenia (ANSI SET) lub opcja partii poleceń (SET)*
4. *Opcja konkretnego polecenia (HINT)*

10

## Konfiguracja instancji SQL Server

**Przykładowe właściwości instancji:**

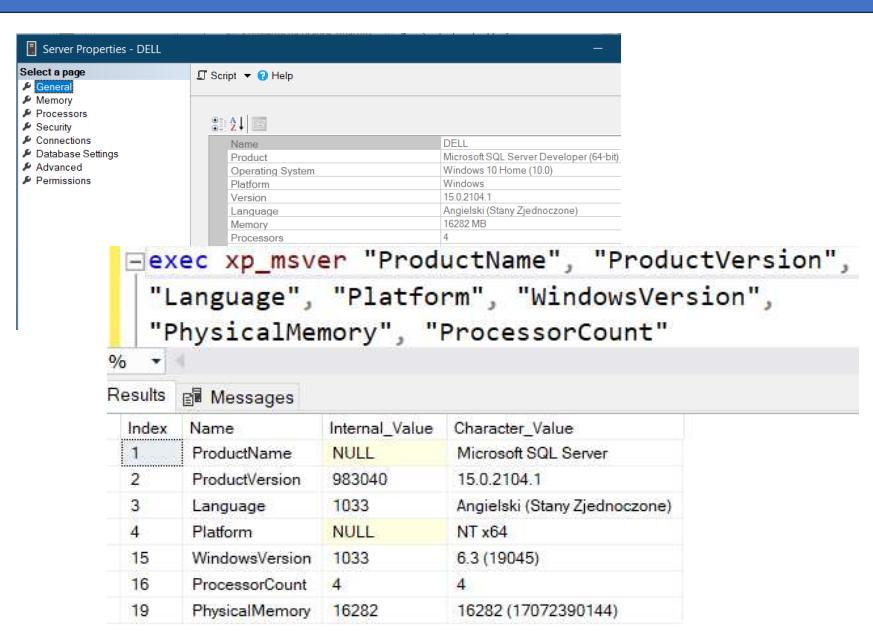
- Pamięć (Min/Max Memory),
- Planowanie wykonania (Affinity Mask, Max Worker),
- Operacje I/O (Affinity I/O Mask),
- Przetwarzanie zapytań (Query Wait, Min Memory Per Query, Blocked Process Treshold, Max Degree Of Parallelism, Index Create Memory),
- Domyślne wartości bazy danych
- Ustawienie FileStream



Name	Value
Name	DELL
Product	Microsoft SQL Server Developer (64-bit)
Operating System	Windows 10 Home (10.0)
Platform	Windows
Version	15.0.2104.1
Language	angielski (Stany Zjednoczone)
Memory	16282 MB
Processors	4
Root Directory	C:\Program Files\Microsoft SQL Server
Server Collation	Polish_CI_AS
Is HADR Enabled	False
Is XTP Supported	True
Is PolyBase Installed	True
Is Clustered	False

11

## Konfiguracja instancji SQL Server



```
exec xp_msver "ProductName", "ProductVersion",
"Language", "Platform", "WindowsVersion",
"PhysicalMemory", "ProcessorCount"
```

**Results**

Index	Name	Internal_Value	Character_Value
1	ProductName	NULL	Microsoft SQL Server
2	ProductVersion	983040	15.0.2104.1
3	Language	1033	angielski (Stany Zjednoczone)
4	Platform	NULL	NT x64
15	WindowsVersion	1033	6.3 (19045)
16	ProcessorCount	4	4
19	PhysicalMemory	16282	16282 (17072390144)

12

## Konfiguracja instancji SQL Server

select \* from sys.configurations

configuration_id	name	value	minimum	maximum	value_in_use	description	is_dynamic	is_advanced
1543	min server memory (MB)	0	0	2147483647	16	Minimum size of server memory (MB)	1	1
1544	max server memory (MB)	2147483647	128	2147483647	2147483647	Maximum size of server memory (MB)	1	1
1545	query governor cost limit	0	0	2147483647	0	Maximum estimated cost allowed by ...	1	1
1546	lightweight pooling	0	0	1	0	User mode scheduler uses lightweig...	0	1
1547	scan for startup procs	0	0	1	0	scan for startup stored procedures	0	1
1549	affinity64 mask	0	-2147...	2147483647	0	affinity64 mask	1	1
1550	affinity I/O mask	0	-2147...	2147483647	0	affinity I/O mask	0	1
1551	affinity64 I/O mask	0	-2147...	2147483647	0	affinity64 I/O mask	0	1

**Memory**

Minimum server memory (in MB): 0  
Maximum server memory (in MB): 2147483647

Other memory options

Index creation memory (in KB, 0 = dynamic memory): 0  
Minimum memory per query (in KB): 1024

```
EXEC sys.sp_configure N'show advanced options', N'1' RECONFIGURE WITH OVERRIDE
GO
EXEC sys.sp_configure N'min server memory (MB)', N'1000'
GO
EXEC sys.sp_configure N'max server memory (MB)', N'10000'
GO
RECONFIGURE WITH OVERRIDE
```

Configured values      Running values

13

## Konfiguracja instancji SQL Server

### Memory

Server Properties - ACER1

Select a page: General, Memory, Processors, Security, Connections, Database Settings, Advanced, Permissions

Connection: Server: ACER1, Connection: sa, Progress: Ready

Script, Help

Server memory options

Minimum server memory (in MB): 0  
Maximum server memory (in MB): 2147483647

Other memory options

Index creation memory (in KB, 0 = dynamic memory): 0  
Minimum memory per query (in KB): 1024

Configured values      Running values

**Index creation memory**  
- domyślnie oprogramowanie SQL Server ustala dynamicznie ilość pamięci alokowanej na potrzeby operacji tworzenia indeksów.

**Minimum memory per query**  
- domyślnie serwer SQL Server alokuje na potrzeby wykonywania zapytań minimum 1024 KB pamięci. Alokowanie takiej ilości pamięci jest gwarantowane dla każdego użytkownika i wielkość tę można zmieniać w przedziale od 512 KB do 2 GB.

14

## Konfiguracja instancji SQL Server

```
ALTER SERVER CONFIGURATION SET PROCESS AFFINITY CPU = 0 TO 1
GO
EXEC sys.sp_configure N'affinity I/O mask', N'12'
GO
RECONFIGURE
GO
```

**Processors**

Server Properties - DELL

Select a page: General, Memory, Processors, Security, Connections, Database Settings, Advanced, Permissions.

Processor Affinity table:

Processor	Processor Affinity	I/O Affinity
ALL	<input type="checkbox"/>	<input type="checkbox"/>
NumaNode0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU0	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU1	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CPU2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CPU3	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Configuration table:

configuration_id	name	value	minimum	maximum	value_in_use	description	is_dynamic	is_advanced
503	max worker threads	0	128	65535	0	Maximum worker threads	1	1

Threads section:

Maximum worker threads:

Domyślne wartości opcji Maximum Worker Threads (Maksymalna liczba wątków roboczych) używane przez serwer SQL Server:

Liczba procesorów	32-bitowe systemy operacyjne	64-bitowe systemy operacyjne
1-4	256	512
5-8	288	576
9-16	352	704
17-32	480	960

Buttons: OK, Cancel.

<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/configure-the-priority-boost-server-configuration-option?view=sql-server-ver15>

15

## Konfiguracja instancji SQL Server

**Security**

- Management
  - Policy Management
  - Data Collection
  - Resource Governor
  - Extended Events
  - Maintenance Plans
  - SQL Server Logs
- Current - 2023-11

Departament Obrony Stanów Zjednoczonych stworzył standard Trusted Computer System Evaluation Criteria (Kryteria oceny zaufanych systemów komputerowych), mający służyć do oceny poziomu bezpieczeństwa systemów komputerowych. Kryteria dla klas od C2 do A1 wymagają, by wykonywane przez użytkownika działania mogły być przedmiotem inspekcji oraz by w przedsiębiorstwie zatrudniony był personel odpowiedzialny za procedury związane z inspekcją. Wymagania inspekcji na poziomie klasy C2 definiują określone zdarzenia i informacje, które będą podlegać inspekcji.

Standard CC (Common Criteria) to międzynarodowy standard w zakresie bezpieczeństwa, który powinien służyć jako podstawa przy ocenie właściwości zabezpieczeń aplikacji i serwerów.

Server Properties - ACER1

Select a page: General, Memory, Processors, Security, Connections, Database Settings, Advanced, Permissions.

Server authentication:

- Windows Authentication mode (radio button)
- SQL Server and Windows Authentication mode

Login auditing:

- None (radio button)
- Failed logins only
- Successful logins only
- Both failed and successful logins

Server proxy account:

Proxy account:

Options:

- Enable server proxy account (checkbox)
- Enable Common Criteria compliance (checkbox)
- Enable C2 audit tracing (checkbox)
- Cross database ownership chaining (checkbox)

Buttons: View connection properties, OK, Cancel.

16

Konfiguracja instancji SQL Server				
Opcja	Opis	Zalety	Ryzyka / Wady	Kiedy używać
<b>Enable Common Criteria compliance</b>	Włącza tryb zgodności z normą <b>Common Criteria (ISO/IEC 15408)</b> .	- Spełnia wymagania certyfikacji bezpieczeństwa. - Zwiększa kontrolę nad uprawnieniami.	- Może ograniczyć funkcjonalność. - Spadek wydajności. - Wymaga testów aplikacji.	Gdy system musi być zgodny z Common Criteria (np. w sektorze publicznym).
<b>Enable C2 audit tracing</b>	Włącza audit zgodny z klasą <b>C2</b> (TCSEC). Rejestruje wszystkie zdarzenia bezpieczeństwa.	- Pełna inspekcja działań użytkowników. - Spełnia wymagania audytowe.	- Duże obciążenie serwera. - Szybkie zapełnianie dysku. - Brak miejsca = zatrzymanie SQL Server.	Tylko gdy wymagają tego przepisy lub audit bezpieczeństwa.
<b>Cross database ownership chaining</b>	Pozwala na łańcuchowe przekazywanie uprawnień między bazami o tym samym właściwcu.	- Ułatwia działanie aplikacji korzystających z wielu baz.	- Ryzyko eskalacji uprawnień. - Potencjalne naruszenia bezpieczeństwa.	Tylko w kontrolowanych środowiskach, gdy jest to absolutnie konieczne.

**Jak to wpływa na bazę danych w praktyce?**  
**Audit i logowanie:** Włączenie audytu (np. logowanie SELECT, INSERT, UPDATE, DELETE) → większe zużycie dysku i CPU.  
**Polityki haseł i autoryzacji:** Wymuszenie silnych haseł, rotacji, blokady kont.  
**Szyfrowanie:** Transparent Data Encryption (TDE), szyfrowanie połączeń (SSL/TLS).  
**Zabezpieczanie serwera:** Wyłączenie xp\_cmdshell, ograniczenie ról sysadmin, konfiguracja firewalli.

17

### Konfiguracja instancji SQL Server

**Connections**

Server Properties - ACER1

Select a page: General, Memory, Processors, Security, Connections, Database Settings, Advanced, Permissions.

**Connections**

Maximum number of concurrent connections (0 = unlimited):

Use query governor to prevent long-running queries  
 *Użyj zarządcy zapytań, aby zapobiec długotrwałym zapytaniom*

**Default connection options:**

- implicit transactions
- cursor close on commit
- ansi warnings
- ansi padding
- ANSI NULLS
- arithmetic abort
- arithmetic ignore
- quoted identifier
- no count
- ANSI NULL Default On
- ANSI NULL Default Off
- concat null yields null
- numeric round about
- xact abort

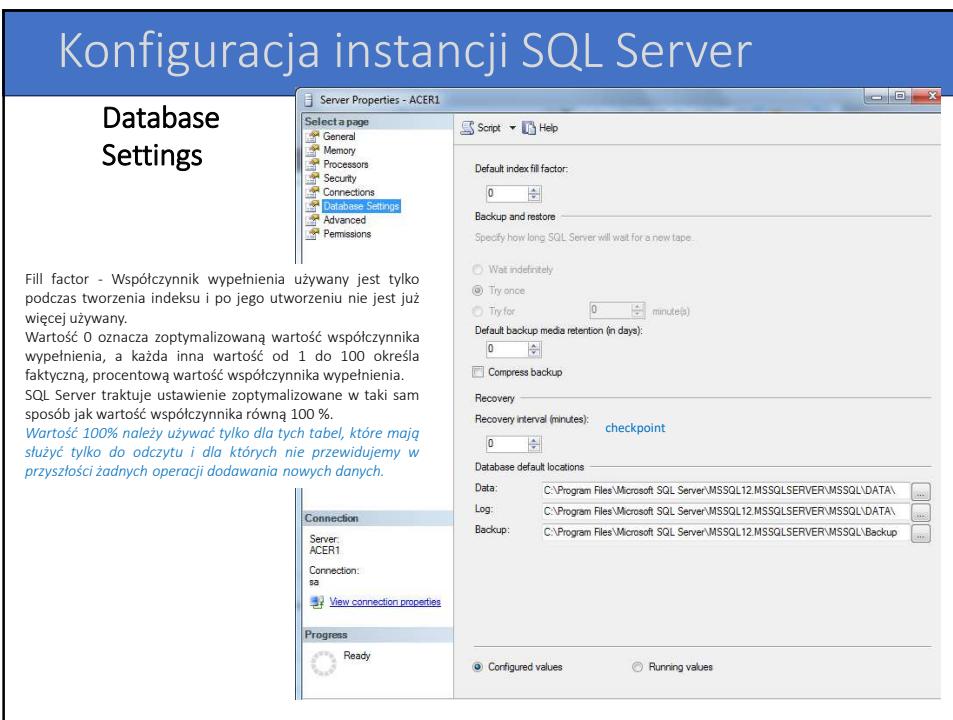
**Connection**

Server: ACER1  
Connection: sa  
[View connection properties](#)

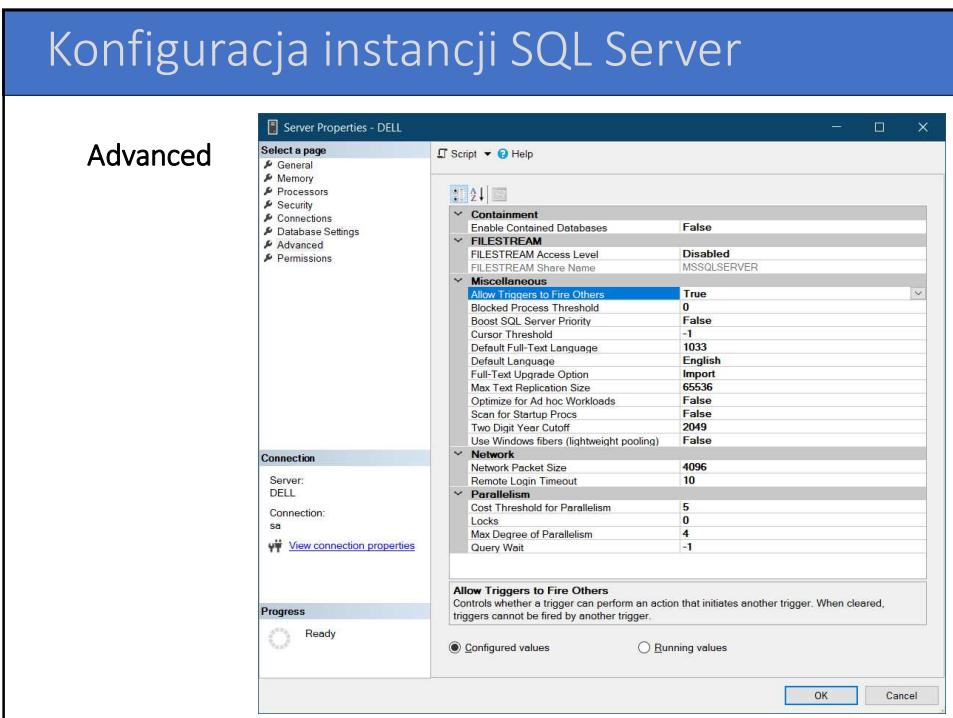
Remote server connections

Allow remote connections to this server  
Remote query timeout (in seconds, 0 = no timeout):   
 Require distributed transactions for server-to-server communication

18



19



20

## Konfiguracja instancji SQL Server

```
use [master]
GO -- przyznanie uprawnień
GRANT CREATE ANY DATABASE TO [user]

GO -- przyznanie uprawnień
GRANT CONNECT SQL TO [user]

GO -- przyznanie uprawnień z opcją Grant Option
GRANT CONNECT SQL TO [user] WITH GRANT OPTION

GO -- zabronienie uprawnień
DENY CONNECT SQL TO [user]

GO -- zabranie uprawnień przyznanych lub zabronionych
REVOKE CONNECT SQL TO [user] AS [sa]
```

21

## Collation

**Collations** - systemu SQL Server wpływają na stronę kodową używaną do zapisywania danych w kolumnach typów char, varchar.

Powoduje to przypisanie do danych pewnych cech.

- interpretacja znaków (strona kodowa),
- sortowanie znaków,
- rozpoznawanie dużych i małych liter,
- akcenty,
- ... .

### Poziomy sortowania

- Sortowanie na poziomie serwera
- Sortowanie na poziomie bazy danych
- Sortowanie na poziomie kolumny
- Sortowanie na poziomie wyrażenia

22

## Zestaw znakowy – strona kodowa

- Domyślnym zestawem znakowym jest strona kodowa 1252 (ISO Character Set) dla LATIN1\_GENERAL\_CI\_AS
- Kiedy dokonasz wyboru zestawu znakowego:
  - Wybierz stronę kodową, która zawiera znaki dla języka używanego przez twoją bazę danych,
  - Wybierz standardowy zestaw znakowy i używaj typów danych Unicode jeżeli masz zamiar zapisywać dane w wielu językach,
  - Używaj tego samego zestawu znakowego dla wszystkich klientów i serwerów, chyba że masz zamiar używać tylko pierwszych 128 znaków.
- Zmiana porządku sortowania po instalacji wymaga przebudowania wszystkich baz danych  
(Zmiana strony kodowej to operacja zaawansowana i ma wpływ na działanie bazy danych i aplikacji. Starannie należy przemyśleć tę zmianę i przetestować ją na środowisku testowym, zanim zastosujemy ją na produkcji)

```
SELECT SERVERPROPERTY('Collation') AS 'Collation of the Instance'  
-- Polish_CI_AS  
SELECT DATABASEPROPERTYEX('AdventureWorks2019', 'Collation') AS 'Collation of the Database'  
-- SQL_Latin1_General_CI_AS
```

23

## Porządek sortowania

- Domyślnym porządkiem sortowania jest sortowanie wg słownikowego porządku sortowania (nie binarnego) np. Polish\_CI\_AS.
- Wybieranie sortowania może mieć wpływa na:
  - Końcowe ustawienia,
  - Sposób działania,
  - Klientów.
- Zmiana porządku sortowania po instalacji wymaga przebudowania wszystkich baz danych

```
select COLLATIONPROPERTY('Latin1_General_CI_AS', 'Codepage')  
--CodePage - 1252  
select COLLATIONPROPERTY('Polish_CI_AS', 'Codepage')  
--CodePage - 1250
```

	binary
1	itd
2	0
3	5
4	baran
5	owca
6	źródło
7	iącki
8	ęcki

	binary2
1	itd
2	0
3	5
4	baran
5	owca
6	iącki
7	ęcki
8	źródło

	dictionary
1	itd
2	0
3	5
4	iącki
5	baran
6	ęcki
7	owca
8	źródło

24

## Zestawienie Unicode

- Domyślne zestawienie Unicode często jest ogólną wersją Unicode z odpowiednią opcją porównywania i sortowania danych.
- Kiedy wybieramy zestawienie Unicode:
  - Wybór tego samego porządku sortowania w przypadku danych Unicode i nie-Unicode jest szczególnie zalecane,
  - Zmiana w standardowym zestawieniu Unicode powoduje, że migracja danych pomiędzy danymi Unicode a nie-Unicode staje się trudniejsza i powoduje różnice w sortowaniu,
  - Wybranie tego samego zestawienia Unicode dla wszystkich serwerów jest wymagane do backupu i replikacji bazy danych pomiędzy serwerami.

25

Zestawienie Unicode

```
select * from Categories
order by CategoryName
collate Polish_BIN
```

select \* from fn\_helpcollations()

name	description
2697 Polish_BIN	Polish, binary sort
2698 Polish_BIN2	Polish, binary code point comparison sort
2699 Polish_CI_AI	Polish, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
2700 Polish_CI_AI_WS	Polish, case-insensitive, accent-insensitive, kanatype-insensitive, width-sensitive
2701 Polish_CI_AI_KS	Polish, case-insensitive, accent-insensitive, kanatype-sensitive, width-insensitive
2702 Polish_CI_AI_KS_WS	Polish, case-insensitive, accent-insensitive, kanatype-sensitive, width-sensitive
2703 Polish_CI_AS	Polish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
2704 Polish_CI_AS_WS	Polish, case-insensitive, accent-sensitive, kanatype-insensitive, width-sensitive
2705 Polish_CI_AS_KS	Polish, case-insensitive, accent-sensitive, kanatype-sensitive, width-insensitive
2706 Polish_CI_AS_KS_WS	Polish, case-insensitive, accent-sensitive, kanatype-sensitive, width-sensitive
2707 Polish_CS_AI	Polish, case-sensitive, accent-insensitive, kanatype-insensitive, width-insensitive
2708 Polish_CS_AI_WS	Polish, case-sensitive, accent-insensitive, kanatype-insensitive, width-sensitive
2709 Polish_CS_AI_KS	Polish, case-sensitive, accent-insensitive, kanatype-sensitive, width-insensitive
2710 Polish_CS_AI_KS_WS	Polish, case-sensitive, accent-insensitive, kanatype-sensitive, width-sensitive
2711 Polish_CS_AS	Polish, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive
2712 Polish_CS_AS_WS	Polish, case-sensitive, accent-sensitive, kanatype-insensitive, width-sensitive
2713 Polish_CS_AS_KS	Polish, case-sensitive, accent-sensitive, kanatype-sensitive, width-insensitive
2714 Polish_CS_AS_KS_WS	Polish, case-sensitive, accent-sensitive, kanatype-sensitive, width-sensitive

SQL\_Polish\_CI\_AS Polish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 88 on Code Page 1250 for non-Unicode Data  
SQL\_Polish\_CI\_AS Polish, case-sensitive, accent-sensitive, kanatype-insensitive, width-insensitive for Unicode Data, SQL Server Sort Order 87 on Code Page 1250 for non-Unicode Data

SQL Server Collation

Windows Collation

26

## Porządek sortowania znaków

Windows collation suffix	Sort order description
_BIN	Binary sort.
_BIN2	BIN2 binary sort order, since SQL Server 2005.
_CI_AI	Case-insensitive, accent-insensitive, kana-insensitive, width-insensitive.
_CI_AI_KS	Case-insensitive, accent-insensitive, kana-sensitive, width-insensitive
_CI_AI_KS_WS	Case-insensitive, accent-insensitive, kana-sensitive, width-sensitive
_CI_AI_WS	Case-insensitive, accent-insensitive, kana-insensitive, width-sensitive
_CI_AS	Case-insensitive, accent-sensitive, kana-insensitive, width-insensitive
_CI_AS_KS	Case-insensitive, accent-sensitive, kana-sensitive, width-insensitive
_CI_AS_KS_WS	Case-insensitive, accent-sensitive, kana-sensitive, width-sensitive
_CI_AS_WS	Case-insensitive, accent-sensitive, kana-insensitive, width-sensitive
_CS_AI	Case-sensitive, accent-insensitive, kana-insensitive, width-insensitive
_CS_AI_KS	Case-sensitive, accent-insensitive, kana-sensitive, width-insensitive
_CS_AI_KS_WS	Case-sensitive, accent-insensitive, kana-sensitive, width-sensitive
_CS_AI_WS	Case-sensitive, accent-insensitive, kana-insensitive, width-sensitive
_CS_AS	Case-sensitive, accent-sensitive, kana-insensitive, width-insensitive
_CS_AS_KS	Case-sensitive, accent-sensitive, kana-sensitive, width-insensitive
_CS_AS_KS_WS	Case-sensitive, accent-sensitive, kana-sensitive, width-sensitive
_CS_AS_WS	Case-sensitive, accent-sensitive, kana-insensitive, width-sensitive

Sortowanie systemu Windows jest łączone w postaci serii przyrostków w celu zdefiniowania wielkości liter, akcentu, szerokości lub kana.

27

## Porządek sortowanie znaków

przedrostek	Sposób przetwarzania
Binary (_BIN)	starsze sortowania BIN, wykonują niekompletne porównanie punktów kodowych z punktami kodowymi dla danych Unicode.
BIN2 (_BIN2)	jest to sortowanie danych dla znaków Unicode
Case-sensitive (_CS)	z rozróżnianiem wielkości liter
Accent-sensitive (_AS)	z rozróżnieniem znaków akcentowanych np. 'a' oraz 'â'
Kana-sensitive (_KS)	Dotyczy japońskich znaków
Width-sensitive (_WS)	Rozróżnianie znaków jedno i dwubajtowych

binary
1 .itd
2 0
3 5
4 baran
5 owca
6 źródło
7 ącki
8 ęcki

binary2
1 .itd
2 0
3 5
4 baran
5 owca
6 ącki
7 ęcki
8 źródło

dictionary
1 .itd
2 0
3 5
4 ącki
5 baran
6 ęcki
7 owca
8 źródło

28

# Zarządzanie plikami bazy danych (partycjonowanie)

1

## Partycjonowanie tabel - Wprowadzenie

Partycjonowanie tabel w SQL Server to technika organizacji danych w tabeli poprzez ich podział na logiczne jednostki zwane partycjami. Każda partycja przechowuje określony zakres danych, co może przynieść korzyści związane z zarządzaniem danymi oraz optymalizacją wydajności zapytań. Warto jednak dokładnie ocenić potrzeby aplikacji i rodzaj danych, aby zdecydować, czy partycjonowanie będzie korzystne w danym przypadku.

### Zalety:

- **Poprawa wydajności zapytań:** Dzięki partycjonowaniu można ograniczyć zakres przeszukiwania danych, co skraca czas odpowiedzi zapytań, szczególnie w przypadku dużych tabel.
- **Łatwiejsze zarządzanie danymi:** Partycjonowanie ułatwia zarządzanie dużymi zbiorami danych, umożliwiając efektywne przeglądanie, utrzymywanie i optymalizowanie poszczególnych fragmentów danych.
- **Poprawa wydajności operacji ETL:** Procesy ETL (Extract, Transform, Load) mogą być bardziej wydajne, ponieważ można manipulować tylko jednym fragmentem danych zamiast całej tabeli.
- **Ułatwione operacje na partycjach:** Partycje mogą być łatwiej zarządzane i utrzymywane, co obejmuje zarówno ich dodawanie, usuwanie, jak i przenoszenie między różnymi tabelami.
- **Lepsza obsługa przeglądania danych historycznych:** Partycjonowanie ułatwia zarządzanie danymi historycznymi, umożliwiając na przykład przechowywanie starszych danych na oddzielnych partycjach.

2

## Partycjonowanie tabel

Możemy używać wszystkich standardowych typów skalarnych:

- liczbowych (bigint, int, smallint, tinyint, decimal/numeric, float, real, money, smallmoney),
- znakowych i binarnych o stałej lub zmiennej długości, ale nie MAX (char, varchar, nchar, nvarchar, binary, varbinary),
- daty/czasu (date, time, datetime, smalldatetime, datetime2, datetimeoffset).

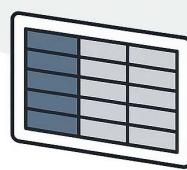
Uwagi:

- Definicja funkcji partycjonującej wspiera tylko jedną kolumnę (jeden parametr). Liczba partycji to liczba wartości granicznych +1.
- Jeżeli podamy literały dat/czasu jako granice, to używamy formatów jednoznacznych (np. yyyyymmdd) albo jawniej konwersji, aby uniknąć zależności od ustawienia języka sesji.

3

## Partycjonowanie tabel

Dane są partycjonowane poziomo według zakresu



Sales.Orders

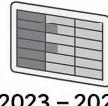
4

## Partycjonowanie tabel

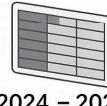
- Funkcje partycjonujące definiują granice partycjonowania
- Wartości graniczne mogą być przypisane do LEWEJ lub PRAWEJ



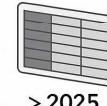
< 2023



2023 – 2024



2024 – 2025



> 2025

```
CREATE PARTITION FUNCTION pf_OrderDate (datetime)
AS RANGE RIGHT
FOR VALUES ('01/01/2023', '01/01/2024', '01/01/2025')
```

```
CREATE PARTITION FUNCTION partition_function_name ( input_parameter_type )
AS RANGE [ LEFT | RIGHT ]
FOR VALUES ( [ boundary_value [ ,...n ] ] )
[ ; ]
```

5

## Partycjonowanie tabel

- Schemat partycjonowania przypisuje partie do grup plików
- Można również zdefiniować grupę plików „next”



< 2023



2023 – 2024



2024 – 2025



> 2025

```
CREATE PARTITION SCHEME ps_OrderDate
AS PARTITION pf_OrderDate
TO (fg1, fg2, fg3, fg4, fg5)
```

```
CREATE PARTITION SCHEME partition_scheme_name
AS PARTITION partition_function_name
[ ALL ] TO ( { file_group_name | [ PRIMARY ] } [ , ...n ] )
[ ; ]
```

6

## CREATE PARTITION FUNCTION

```
CREATE PARTITION FUNCTION myRangePF1 (int)
AS RANGE LEFT FOR VALUES (1, 100, 1000);
```

Copy

The following table shows how a table that uses this partition function on partitioning column **col1** would be partitioned.

Partition	1	2	3	4
Values	<b>col1</b> <= 1	<b>col1</b> > 1 AND <b>col1</b> <= 100	<b>col1</b> > 100 AND <b>col1</b> <= 1000	<b>col1</b> > 1000

```
CREATE PARTITION FUNCTION myRangePF2 (int)
AS RANGE RIGHT FOR VALUES (1, 100, 1000);
```

Copy

The following table shows how a table that uses this partition function on partitioning column **col1** would be partitioned.

Partition	1	2	3	4
Values	<b>col1</b> < 1	<b>col1</b> >= 1 AND <b>col1</b> < 100	<b>col1</b> >= 100 AND <b>col1</b> < 1000	<b>col1</b> >= 1000

7

## CREATE PARTITION FUNCTION

```
CREATE PARTITION FUNCTION [myDateRangePF1] (datetime)
AS RANGE RIGHT FOR VALUES ('20030201', '20030301', '20030401',
                           '20030501', '20030601', '20030701', '20030801',
                           '20030901', '20031001', '20031101', '20031201');
```

Partition	1	2	...	11	12
Values	<b>datecol</b> < February 1, 2003	<b>datecol</b> >= February 1, 2003 AND datecol < March 1, 2003		<b>datecol</b> >= November 1, 2003 AND <b>datecol</b> < December 1, 2003	

```
CREATE PARTITION FUNCTION myRangePF3 (char(20))
AS RANGE RIGHT FOR VALUES ('EX', 'RXE', 'XR');
```

Partition	1	2	3	4
Values	<b>col1</b> < EX...	<b>col1</b> >= EX AND <b>col1</b> < RXE...	<b>col1</b> >= RXE AND <b>col1</b> < XR...	<b>col1</b> >= XR

8

## CREATE PARTITION SCHEME

```
CREATE PARTITION FUNCTION myRangePF1(INT)
AS RANGE LEFT FOR VALUES (1, 100, 1000);
GO

CREATE PARTITION SCHEME myRangePS1
AS PARTITION myRangePF1
TO (test1fg, test2fg, test3fg, test4fg);
```

Filegroup	Partition	Values
test1fg	1	col1 <= 1
test2fg	2	col1 > 1 AND col1 <= 100
test3fg	3	col1 > 100 AND col1 <= 1000
test4fg	4	col1 > 1000

9

## CREATE PARTITION SCHEME

```
CREATE PARTITION FUNCTION myRangePF3(INT)
AS RANGE LEFT FOR VALUES (1, 100, 1000);
GO

CREATE PARTITION SCHEME myRangePS3
AS PARTITION myRangePF3
ALL TO (test1fg);
```

```
CREATE PARTITION FUNCTION myRangePF2(INT)
AS RANGE LEFT FOR VALUES (1, 100, 1000);
GO

CREATE PARTITION SCHEME myRangePS2
AS PARTITION myRangePF2
TO (test1fg, test1fg, test1fg, test2fg);
```

Partycje w tym samym filegroup – po co? Fizycznie dane razem, ale daje korzyści:

- Przygotowanie na skalowanie
  - Łatwe przenoszenie partycji na inne filegroup'y bez przebudowy tabeli.
- Operacje administracyjne
  - SWITCH PARTITION – szybkie przenoszenie danych (np. archiwizacja).
  - MERGE / SPLIT – dodawanie/usuwanie granic bez przebudowy tabeli.
- Zarządzanie indeksami
  - Przebudowa indeksów per partycja (mniej blokad, krótszy czas).

Choć fizycznie nie daje to korzyści w I/O, logiczna struktura partycji daje elastyczność w zarządzaniu danymi i indeksami oraz przygotowuje grunt pod przyszłe zmiany.

10

## Partycjonowanie

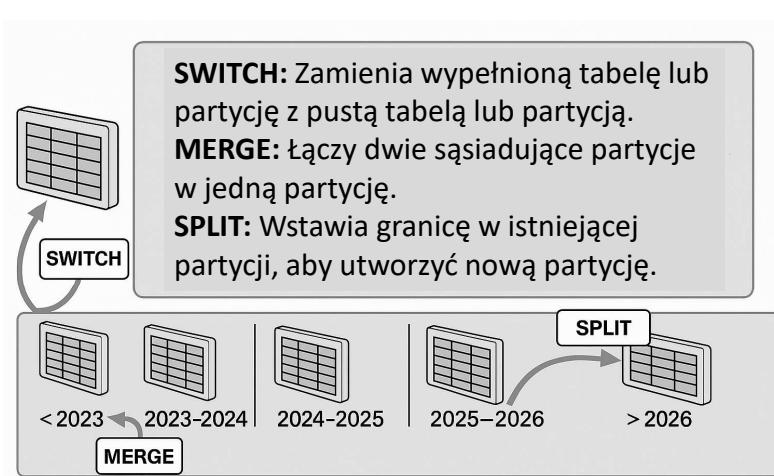
```
-- Krok 1. Utwórz funkcję partycjonowania
CREATE PARTITION FUNCTION myRangePF1(int)
AS
    RANGE LEFT FOR VALUES (1, 100)
GO

-- Krok 2. Utwórz schemat partycjonowania powiązany z funkcją
CREATE PARTITION SCHEME myRangePS1
AS
    PARTITION myRangePF1
        TO (fg1, fg2, fg3)
GO

-- Utwórz tabelę na schemacie partycjonowania
CREATE TABLE dbo.PartitionTable
(
    col1 int,
    col2 char(10)
)
ON myRangePS1 (col1)
GO
```

11

## Partycjonowanie tabel



12

## Partycjonowanie tabel - SWITCH (ALTER TABLE)

```
-- Wypełniamy przykładowymi danymi
INSERT INTO dbo.PartitionTable values (0, 'dane0'),(50, 'dane1'),(110, 'dane2')
GO
-- Tworzymy pustą tabelę identyczną z tabelą partycjonowaną
CREATE TABLE dbo.NonPartitionTable
(
    col1 INT,
    col2 CHAR (10)
) ON fg2;
-- warto dodać ograniczenie CHECK, aby dane spełniały kryterium partycji
ALTER TABLE dbo.NonPartitionTable WITH CHECK ADD CONSTRAINT [chk_partition_2]
    CHECK ([col1] > 1 AND [col1] <= 100 AND [col1] IS NOT NULL)
-- Dane z tej tabeli mają trafić do tej partycji, gdzie są dane
-- Nie mogą być z innej partycji
GO
-- Sprawdzenie numerów partycji, gdzie są dane z danej partycji o wartości np. 10
SELECT $PARTITION.myRangePF1(10) AS PartitionNumber;
GO
-- Przenosimy dane w ramach tej samej grupy plików
ALTER TABLE PartitionTable SWITCH PARTITION 2 TO NonPartitionTable;
GO
-- Wyświetl dane z tabeli PartitionTable
SELECT * FROM PartitionTable
-- Wyświetl dane z tabeli NonPartitionTable
SELECT * FROM NonPartitionTable
```

	col1	col2
1	0	dane0
2	110	dane2
	col1	col2
1	50	dane1

13

## Partycjonowanie tabel - (SPLIT, MERGE)

```
ALTER PARTITION FUNCTION partition_function_name()
{
    SPLIT RANGE ( boundary_value )
    | MERGE RANGE ( boundary_value )
} [ ; ]
```

```
ALTER PARTITION SCHEME partition_scheme_name
NEXT USED [ filegroup_name ] [ ; ]
```

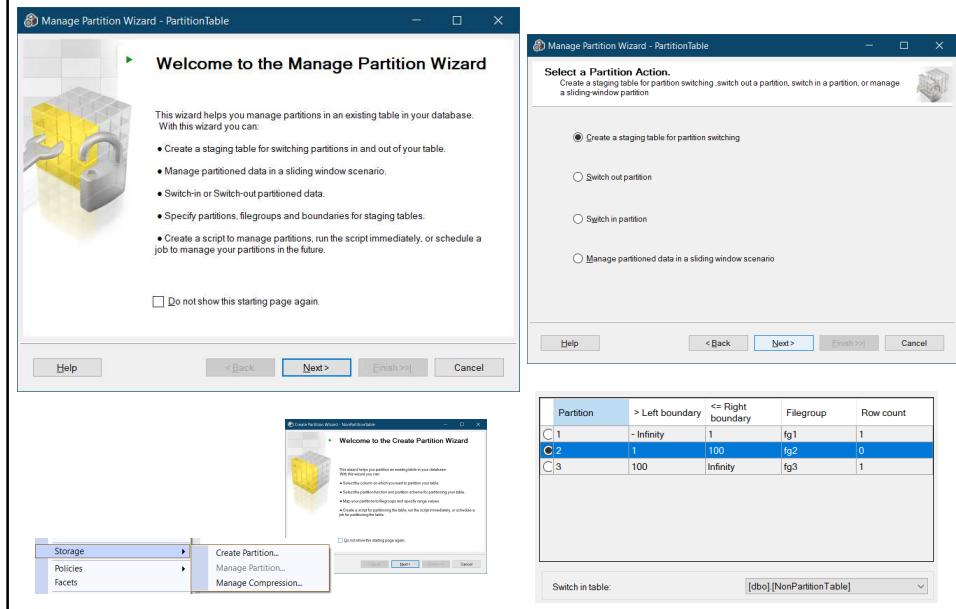
```
-- Ustaw następny docelowy filegroup dla nowo tworzonej partycji
ALTER PARTITION SCHEME myRangePS1
NEXT USED fg4
GO

-- Scal (usuń) granicę 1 w funkcji partycjonowania
ALTER PARTITION FUNCTION myRangePF1() MERGE RANGE (1);
GO

-- Podziel zakres, dodając nową granicę 50 (RANGE LEFT: 50 należy do lewej partycji)
ALTER PARTITION FUNCTION myRangePF1() SPLIT RANGE (50);
GO
```

14

## Partycjonowanie tabel - *switch*



15

# Stretch Database (przechowywanie części danych w Chmurze)

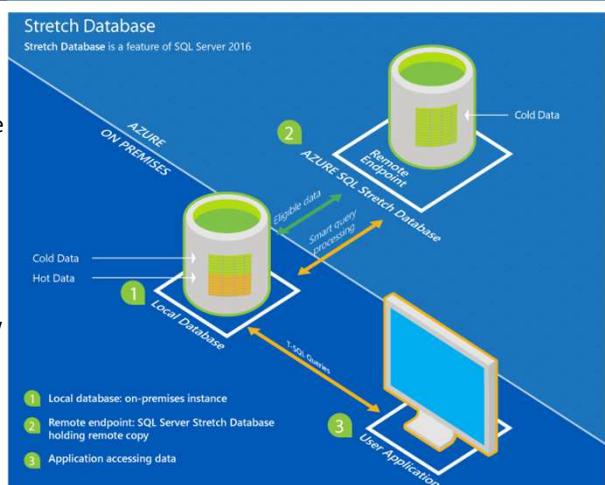
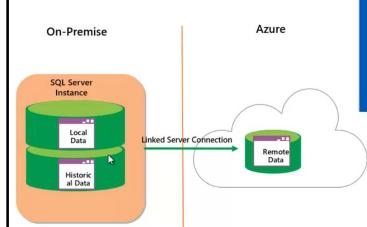
(deprecated)

### Czym jest Stretch Database?

- Dane historyczne – przechowywane w chmurze
  - Dane bieżące – przechowywane lokalnie (on-premise)

#### **Korzyści:**

- Szybsze kopie zapasowe
  - Zmniejszone wymagania dotyczące lokalnej przestrzeni dyskowej
  - Przecroczyste dla użytkowników



16

# Stretch Database

(przechowywanie części danych w Chmurze)

## Jakie są zalety Stretch Database?

- Zapewnia ekonomiczną dostępność do danych rzadko używanych (danych historycznych)
- Nie wymaga zmian zapytań ani aplikacji
- Usprawnia lokalną konserwację danych (szybszy backup lokalnej bazy)
- Zapewnia bezpieczeństwo danych nawet podczas migracji
- Redukuje zasoby dyskowe na serwerze lokalnym

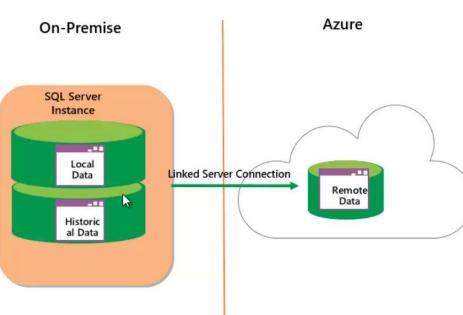
## Do czego służy Stretch Database?

Po włączeniu Stretch Database dla instancji SQL Server i bazy danych oraz wybraniu co najmniej jednej tabeli, Stretch Database cicho rozpoczyna migrację danych na platformę Azure.

- Jeśli przechowujesz dane w osobnej tabeli, możesz migrować całą tabelę.
- Jeśli tabela zawiera zarówno normalne i historyczne dane, możesz określić funkcję filtrowania, aby wybrać wiersze do migracji.



Identyfikowanie bazy danych i tabel dla Stretch Database można realizować za pomocą Asystenta migracji danych





## Zarządzanie bazą danych i plikami bazy danych

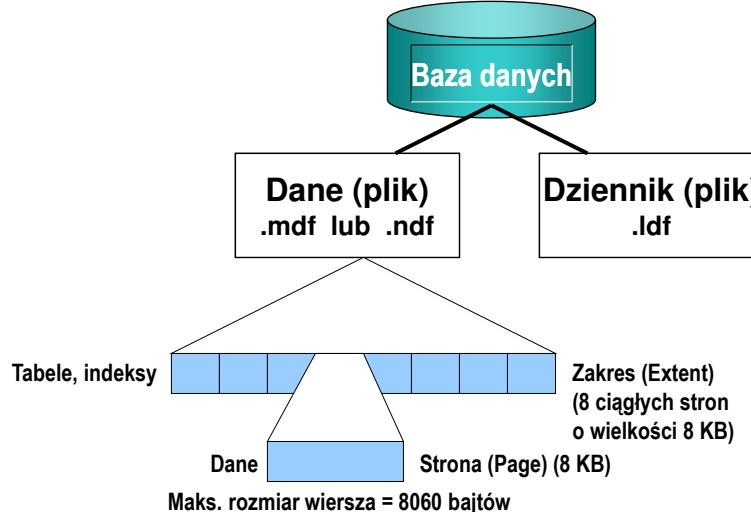
1

### Omówienie

- Jak są przechowywane dane
- Tworzenie baz danych
- Opcje bazy danych
- Modyfikowanie baz danych
- Tworzenie grup plików (zarządzanie na poziomie wielu dysków)
- Planowanie pojemności

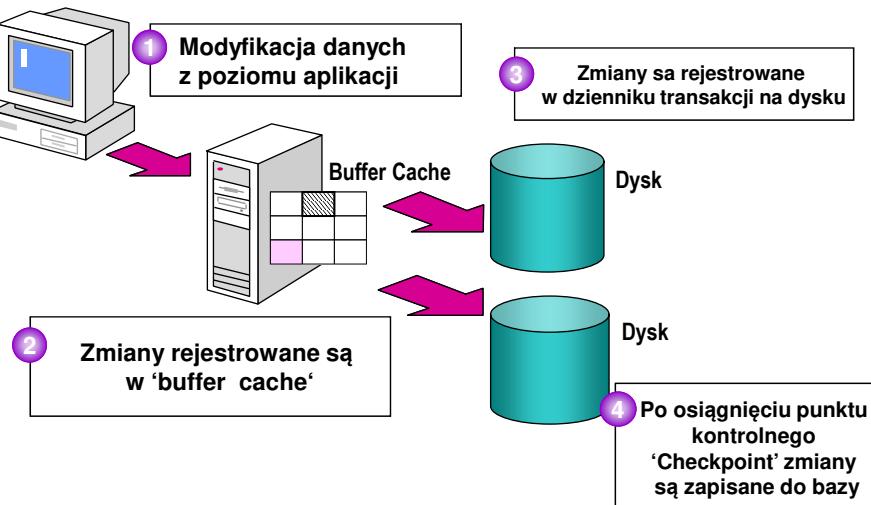
2

## Jak są przechowywane dane



3

## Jak pracuje dziennik transakcji



4

## Tworzenie baz danych

- Tworzenie bazy danych obejmuje definiowanie:
  - Nazwy bazy danych
  - Rozmiaru bazy danych
  - Plików, w których będzie rezydować baza danych

```
CREATE DATABASE sample
ON
    PRIMARY ( NAME=sample_data,
    FILENAME='c:\mssql7\data\sample.mdf',
    SIZE=10MB,
    MAXSIZE=15MB,
    FILEGROWTH=20%)
LOG ON
    ( NAME=sample_log,
    FILENAME='c:\mssql7\data\sample.ldf',
    SIZE=3MB,
    MAXSIZE=5MB,
    FILEGROWTH=1MB)
```

5

## Ustawianie opcji baz danych

- Ustawianie opcji baz danych, polecenie **ALTER DATABASE**

- Często wykorzystywane opcje

- dbo use only

```
USE [master]
GO
ALTER DATABASE [Northwind] SET RESTRICTED_USER WITH NO_WAIT
GO
```

- read only

```
USE [master]
GO
ALTER DATABASE [Northwind] SET READ_ONLY WITH NO_WAIT
GO
```

- trunc. log on chkpt.  
(związane z Recovery model)
  - autoshrink






```
USE [master]
GO
ALTER DATABASE [Northwind] SET AUTO_SHRINK ON WITH NO_WAIT
GO
```

**sp\_configure**

6

## Przeglądanie opcji baz danych

- Przeglądanie opcji baz danych

```
DBCC USEROPTIONS;
```

textsize	2147483647
language	us_english
dateformat	rndy
datefirst	7
lock_timeout	-1
quoted_identifier	SET
arithabort	SET
ansi_null_dflt_on	SET
ansi_warnings	SET
ansi_padding	SET
ansi_nuts	SET
concat_null_yields_null	SET
isolation level	read committed

Database Properties - Northwind

Select a page: General, Files, Filegroups, Options, Transaction Tracking, Permissions, Extended Properties, Mirroring, Transaction Log Shipping.

Collation: Polish\_CI\_AS  
Recovery model: Full  
Compatibility level: SQL Server 2014 (120)  
Containment type: None  
Other options:

**Automatic**

Auto Close	False
Auto Create Incremental Statistics	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False

**Containment**

Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049

**Cursor**

Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL

**FILESTREAM**

FILESTREAM Directory Name	
FILESTREAM Non-Transacted Access	Off

**Management**

Allow Replication	False
ANSI NULL Default	False
ANSI NULLS Enabled	False
ANSI Padding Enabled	False
ANSI Warnings Enabled	False
Arithmetic Abort Enabled	False
Concatenate Null Yields Null	False
Cross-database Ownership Chaining Enabled	False
Date Correlation Optimization Enabled	False
Delayed Durability	Disabled
Is Read Committed Snapshot On	False
Numeric Round-Abort	False
Parameterization	Simple

```
DECLARE @ANSI_NULL VARCHAR(3) = 'OFF';
IF ( 32 & @@OPTIONS ) = 32 SET @ANSI_NULL = 'ON';
SELECT @ANSI_NULL AS ANSI_NULL;
```

7

## Przeglądanie opcji baz danych

```
use Northwind
EXEC sp_spaceused N'dbo.Categories';
```

name	rows	reserved	data	index_size	unused
Categories	9	168 KB	112 KB	24 KB	32 KB

**select \* from sys.databases**

- lista wszystkich dostępnych opcji,

**sp\_helpdb**

- raport o wszystkich bazach na serwerze,

**sp\_helpdb database\_name**

- raport o konkretnej bazie na serwerze,

**sp\_spaceused [object\_name]**

- miejsce na obiekty bazy danych.

```
EXEC sp_helpdb Northwind|
```

name	db_size	owner	dbid	created	status	compatibility_level
Northwind	6.19 MB	sa	14	Oct 15 2017	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	120

name	fileid	filename	filegroup	size	maxsize	growth	usage
Northwind	1	C:\Program Files\Microsoft SQL Server\MSSQL12.MSS...	PRIMARY	5312 KB	Unlimited	1024 KB	data only
Northwind_log	2	C:\Program Files\Microsoft SQL Server\MSSQL12.MSS...	NULL	1024 KB	2147483648 KB	10%	log only

8

## Ustawianie opcji baz danych

**auto create statistics** - brakujące statystyki będą automatycznie tworzone przez proces optymalizatora wykonania instrukcji Transact-SQL,

**auto update statistics** - nieaktualne statystyki będą automatycznie aktualizowane,

**autoclose** - po rozłączeniu się ostatniego użytkownika baza danych zostanie automatycznie zamknięta,

**autoshrink** - pliki bazy danych zawierające ponad 25% ciągłej, pustej przestrzeni będą automatycznie zmniejszone,

**ANSI null default** - nowo utworzone tabele będą zawierały kolumny dopuszczające wartość nieokreśloną w myśl standardu SQL-92,

**ANSI nulls** - jakiekolwiek porównania z wartością NULL zwracają NULL. Po wyłączeniu porównanie dwóch wartości NULL zwraca wartość TRUE,

**ANSI warnings** - wykonanie poprawnych, ale niezgodnych ze standardem SQL-2 instrukcji języka Transact-SQL, spowoduje wyświetlenie ostrzeżenia,

**concat null yields null** - połączenie (konkatenacja) dowolnego ciągu znaków z wartością NULL zwraca wartość NULL,

**cursor close on commit** - otwarte kursory zostaną automatycznie zamknięte podczas zatwierdzania lub wycofywania transakcji,

**dbo use only** - tylko członkowie roli dbo mają dostęp do bazy,

```
ALTER DATABASE [Northwind] SET AUTO_CLOSE ON WITH NO_WAIT
```

9

## Ustawianie opcji baz danych

**default to local cursor** - domyślną lokalizacją tworzonych kurSORów jest LOCAL,

**merge publish** - baza danych może brać udział w replikacji scalanej,

**offline** - baza danych została wyłączona,

**published** - baza danych może zostać opublikowana,

**quoted identifier** - do identyfikowania obiektów można używać cudzysłowu,

**read only** - baza danych umożliwia jedynie odczyt danych,

**recursive triggers** - wyzwalacz może wywoływać inny wyzwalacz,

**select into/bulkcopy** - dopuszczalne jest wykonywanie operacji masowego ładowania danych,

**single user** - baza danych pozwala na pojedyncze połączenie klienckie. Uwaga — w przypadku bazy msdb pierwsze połączenie zostanie nawiązane przez rejestrujący się w systemie proces SQL Server Agent,

**subscribed** - baza danych może pobierać dane w procesie replikacji,

**torn page detection** - niekompletne (np. uszkodzone podczas zapisywania) strony zostaną oznaczone jako błędne,

**trunc. log on chkpt** - podczas punktu kontrolnego nieaktywna część dziennika transakcyjnego jest nieodwracalnie usuwana.

```
ALTER DATABASE [Northwind] SET READ_ONLY WITH NO_WAIT
```

10

## Ustawianie opcji połączenia z bazą danych - SET

**Date and time statements:** SET DATEFIRST, SET DATEFORMAT

**Locking statements :** SET DEADLOCK\_PRIORITY, SET LOCK\_TIMEOUT

**Miscellaneous statements :** SET CONCAT\_NULL\_YIELDS\_NULL, SET CURSOR\_CLOSE\_ON\_COMMIT, SET FIPS\_FLAGGER, SET IDENTITY\_INSERT, SET LANGUAGE, SET OFFSETS, SET QUOTED\_IDENTIFIER

**Query Execution Statements :** SET ARITHABORT, SET ARITHIGNORE, SET NOCOUNT, SET NOEXEC, SET NUMERIC\_ROUNDABORT, SET PARSEONLY, SET QUERY\_GOVERNOR\_COST\_LIMIT, SET ROWCOUNT, SET TEXTSIZE

**ISO Settings statements :** SET ANSI\_DEFAULTS, SET ANSI\_NULL\_DFLT\_OFF, SET ANSI\_NULL\_DFLT\_ON, SET ANSI\_NULLS, SET ANSI\_PADDING, SET ANSI\_WARNINGS

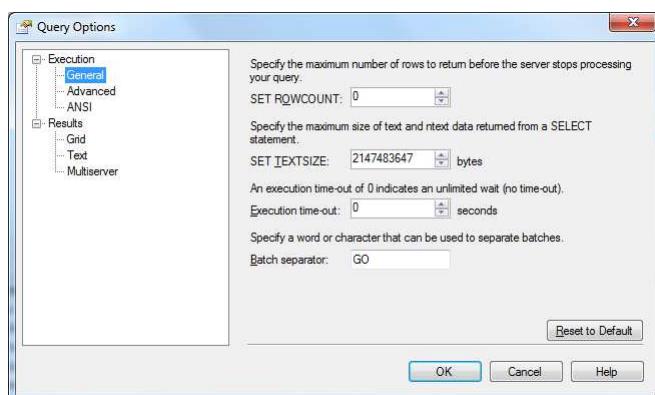
**Statistics statements :** SET FORCEPLAN, SET SHOWPLAN\_ALL, SET SHOWPLAN\_TEXT, SET SHOWPLAN\_XML, SET STATISTICS IO, SET STATISTICS XML, SET STATISTICS PROFILE, SET STATISTICS TIME

**Transactions statements :** SET IMPLICIT\_TRANSACTIONS, SET REMOTE\_PROC\_TRANSACTIONS, SET TRANSACTION ISOLATION LEVEL, SET XACT\_ABORT

SET  
ANSI\_DEFAULTS  
ANSI\_NULL\_DFLT\_OFF  
ANSI\_NULL\_DFLT\_ON  
ANSI\_NULLS  
ANSI\_PADDING  
ANSI\_WARNINGS  
ARITHABORT  
ARITHIGNORE  
CONCAT\_NULL\_YIELDS\_NULL  
CONTEXT\_INFO  
CURSOR\_CLOSE\_ON\_COMMIT  
DATEFIRST  
DATEFORMAT  
DEADLOCK\_PRIORITY  
FIPS\_FLAGGER  
FMTONLY  
FORCEPLAN  
IDENTITY\_INSERT  
IMPLICIT\_TRANSACTIONS  
LANGUAGE  
LOCK\_TIMEOUT  
NOCOUNT  
NOEXEC

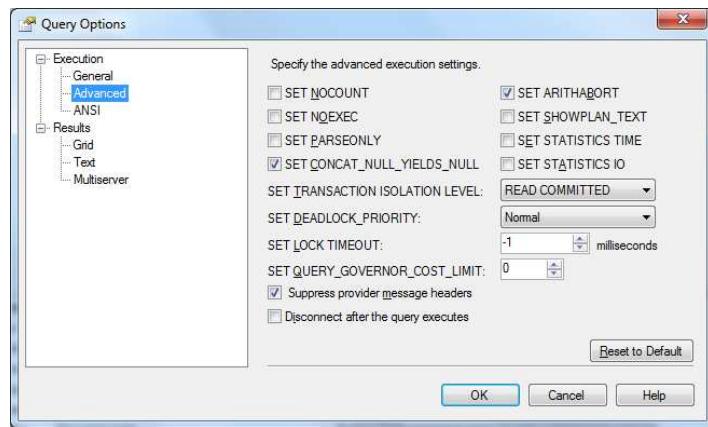
11

## Ustawianie opcji danego połączenia (SSMS)



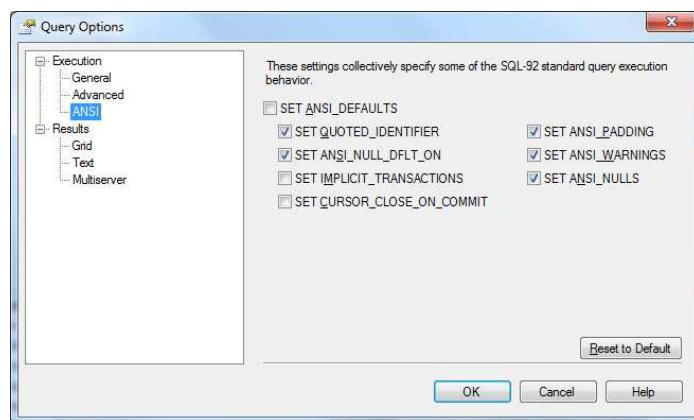
12

## Ustawianie opcji połączenia (SSMS)



13

## Ustawianie opcji połączenia (SSMS)



14

## Ustawianie opcji dla danego połączenia (query)

**SET NOCOUNT** - wyłączenie wyświetlania informacji o liczbie zwracanych wierszy.

**SET NOEXEC** - instrukcja zostanie skompilowana, ale nie wykonana. W rezultacie zostanie sprawdzona poprawność syntaktyczna i semantyczna.

**SET PARSEONLY** - sprawdzona zostaje poprawność syntaktyczna instrukcji.

**SET CONCAT\_NULL\_YIELDS\_NULL** - opcja domyślnie włączona. Połączenie (konkatenacja) dowolnego ciągu z wartością NULL da w wyniku wartość NULL, a nie ciąg źródłowy.

**SET ROWCOUNT** - zwraca jedynie określona liczbę wierszy. Domyślana wartość 0 oznacza zwrócenie wszystkich wierszy wyniku.

**SET ARITHABORT** - opcja domyślnie włączona. Jeżeli podczas wykonywania instrukcji wystąpi błąd arytmetyczny (np. dzielenie przez 0), jej wykonanie zostanie przerwane.

**SET SHOWPLAN\_TEXT** - wyświetlenie jedynie planu wykonania instrukcji bez jego wykonania przez SQL Server.

**SET STATISTICS TIME** - instrukcja zostanie wykonana oraz zostaną zwrócone informacje o liczbie ms potrzebnych na analizę, sprawdzenie poprawności i wykonanie instrukcji.

**SET STATISTICS IO** - dodatkowo zostaną wyświetlane informacje o liczbie i czasie operacji IO przeprowadzonych przez SQL Server w celu wykonania instrukcji.

15

## Ustawianie opcji dla danego połączenia (query)

Grupa opcji **SET ANSI\_DEFAULTS** wymusza zgodność ze standardem SQL-92:

**SET ANSI\_NULLS** — domyślnie ON. Wynikiem jakichkolwiek porównań z wartością NULL będzie wartość nieokreślona. Ponieważ wartość NULL reprezentuje wartość nieokreśloną lub nieznaną, standard SQL-92 uniemożliwia sprawdzenie, czy jakakolwiek wartość jest równa wartości NULL (np. @adres = NULL), pozwala jedynie stwierdzić, czy wartość nie jest wartością nieokreśloną (@adres is NULL). Po wyłączeniu opcji Set ansi\_nulls wynikiem porównania kilku wartości nieokreślonych będzie TRUE (prawda).

**SET ANSI\_NULL\_DFLT\_ON** — domyślnie ON. Nadpisuje domyślne ustawienia dla nowych kolumn, umożliwiając wstawianie w nich wartości nieokreślonej.

**SET ANSI\_PADDING** — domyślnie ON. Wyłącza automatyczne usuwanie wiodących zer z liczb i spacji z tekstów.

**SET ANSI\_WARNINGS** — domyślnie ON. Wykonanie instrukcji, które naruszają zasady standardu SQL-92, ale są zgodne z językiem Transact-SQL, spowoduje wyświetlenie komunikatu ostrzeżenia.

**SET CURSOR\_CLOSE\_ON\_COMMIT** — automatyczne zamykanie kursora w momencie zatwierdzania transakcji.

**SET IMPLICIT\_TRANSACTIONS** — rozpoczęcie nowej transakcji następuje automatycznie, bez konieczności wpisywania instrukcji *BEGIN TRANSACTION*.

**SET QUOTED\_IDENTIFIER** — domyślnie True. Identyfikatory obiektów mogą być umieszczane w cudzysłówach, a literałы muszą być umieszczane w apostrofach.

16

## Ustawianie opcji sesji - SET

```
SET QUOTED_IDENTIFIER ON  
SET ARITHABORT ON  
SET NUMERIC_ROUNDABORT OFF  
SET ANSI_WARNINGS ON  
SET ANSI_PADDING ON  
SET ANSI_NULLS ON  
SET CONCAT_NULL_YIELDS_NULL ON  
SET CURSOR_CLOSE_ON_COMMIT OFF  
SET IMPLICIT_TRANSACTIONS OFF  
SET LANGUAGE US_ENGLISH  
SET DATEFORMAT MDY  
SET DATEFIRST 7  
SET TRANSACTION ISOLATION LEVEL READ COMMITTED
```

```
DBCC USEROPTIONS;
```

textsize	2147483647
language	us_english
dateformat	mdy
datefirst	7
lock_timeout	-1
quoted_identifier	SET
arithabort	SET
ansi_null_dflt_on	SET
ansi_warnings	SET
ansi_padding	SET
ansi_nulls	SET
concat_null_yields_null	SET
isolation level	read committed

```
DECLARE @ANSI_NULL VARCHAR(3) = 'OFF';  
IF ( (32 & @@OPTIONS) = 32 ) SET @ANSI_NULL = 'ON';  
SELECT @ANSI_NULL AS ANSI_NULL;
```

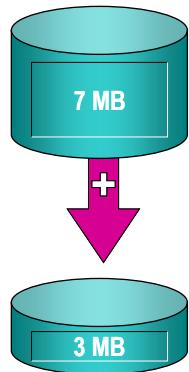
17

## Modyfikowanie baz danych

- Zarządzanie danymi i wzrostem rozmiaru pliku dziennika
- Rozszerzanie dziennika transakcji
- Zmniejszanie bazy danych lub pliku
- Usuwanie bazy danych

18

## Zarządzanie danymi i wzrostem rozmiaru pliku dziennika



```
ALTER DATABASE sample
    MODIFY FILE ( NAME = 'sample_log',
    SIZE = 10MB)
GO

ALTER DATABASE sample
ADD FILE
(NAME = sample_data2 ,
FILENAME='c:\mssql17\data\sample2.ndf',
SIZE=10MB ,
MAXSIZE=20MB)
GO
```

19

## Rozszerzanie dziennika transakcji

- Monitorowanie dziennika
- Rozszerzanie dziennika w razie potrzeby
- Monitorowanie sytuacji powodujących wzmożoną aktywność dziennika
  - Masowe ładowanie danych do tabeli indeksowanej
  - Duże transakcje
  - Wykonywanie operacji związanych z tekstem i obrazami zarejestrowanymi w dzienniku

20

## Zmniejszanie bazy danych lub pliku

- Zmniejszanie całej bazy danych

```
DBCC SHRINKDATABASE (sample, 25)
```

- Zmniejszanie pliku danych w bazie danych

```
DBCC SHRINKFILE (sample_data, 10)
```

- Automatyczne zmniejszanie bazy danych

21

## Usuwanie bazy danych

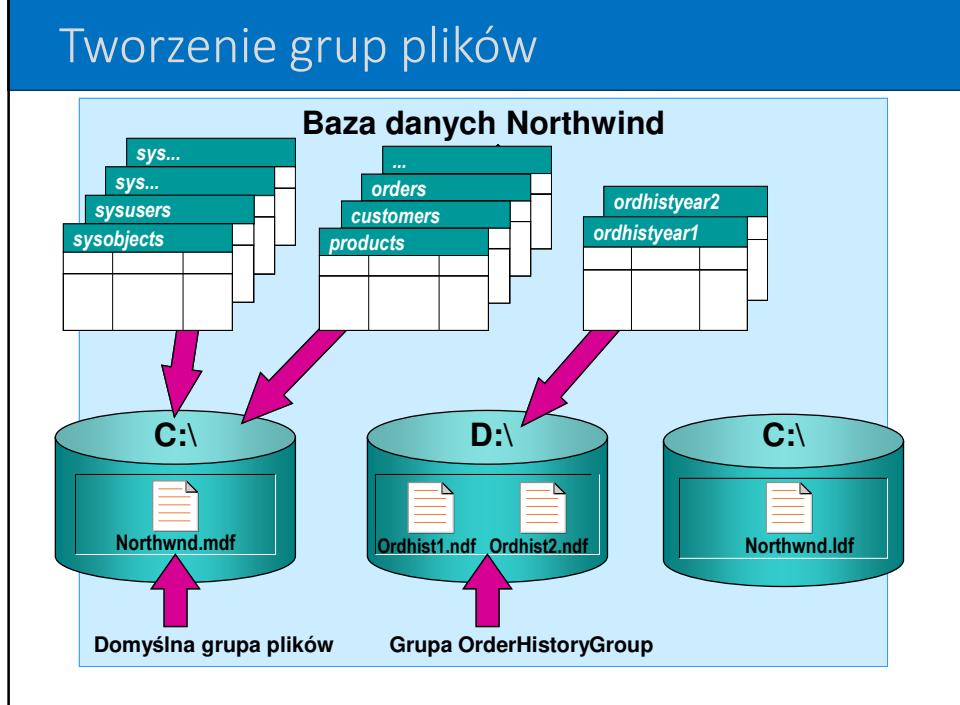
- Metody usuwania bazy danych
  - Program Microsoft SQL Server Management Studio
  - Instrukcja DROP DATABASE

```
DROP DATABASE Northwind, Advantureworks
```

- Ograniczenia dotyczące usuwania bazy danych
  - Podczas przywracania
  - Kiedy użytkownik nawiąże połączenie z bazą danych
  - Podczas publikowania stanowiącego część replikacji

22

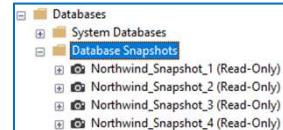
## Tworzenie grup plików



23

## Tworzenie migawki bazy danych

```
CREATE DATABASE database_snapshot_name
ON
(
    NAME = logical_file_name,
    FILENAME = 'os_file_name'
) [ ,...n ]
AS SNAPSHOT OF
[]
```



```
-- Create the database snapshot
CREATE DATABASE Northwind_Snapshot_1 ON
( NAME = Northwind, FILENAME =
'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Northwind_Snapshot_1.ss' )
AS SNAPSHOT OF Northwind;
```

24

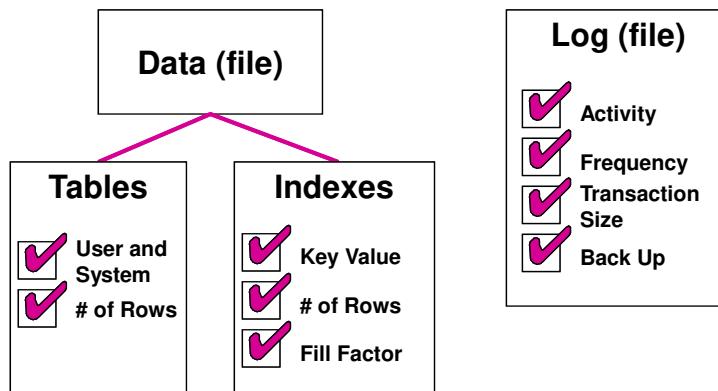
## Planowanie pojemności

- Ustalanie rozmiaru modelu bazy danych i tabel systemowych
- Oszacowywanie ilości danych w tabelach

25

### Ustalanie rozmiaru modelu bazy danych i tabel systemowych

Czynniki, które należy wziąć pod uwagę podczas oszacowywanie rozmiaru bazy danych



26

## Szacowanie ilości danych w tabelach

- Kalkulowanie ilości bajtów w rekordzie
  - sumujemy liczbę bajtów w rekordzie
  - uśredniamy kolumny zależne od wielkości
- Ustalanie ilość rekordów w stronie danych
  - dzielimy 8060 przez zsumowaną ilość bajtów w rekordzie
  - zaokrąglamy do najbliższej liczby całkowitej
- Dzielimy ilość rekordów w tabeli przez ilość rekordów na stronie danych

27

## Szacowany rozmiar wielkości pliku danych - przykład

### *Rozmiar danych*

Znając strukturę tabel oraz typy danych kolumn w bazie danych, możemy oszacować rozmiar pliku danych.

Procedura szacowania rozmiaru pliku danych przebiega według następującego schematu:

1. Na podstawie tabeli określamy rozmiar pojedynczego wiersza. Dla typów danych o stałym rozmiarze dodajemy liczbę bajtów poszczególnych kolumn, zaś dla typów o zmiennym rozmiarze przyjmujemy wartość średnią.
2. Następnie maksymalną liczbę bajtów na pojedynczej stronie danych (8060) dzielimy przez szacunkową wielkość wiersza. W wyniku otrzymujemy liczbę wierszy mieszczących się na stronie.
3. Następnie szacunkową liczbę wierszy dzielimy przez otrzymaną liczbę wierszy na stronie. W ten sposób otrzymujemy liczbę stron niezbędnych do zapisania danych znajdujących się w tabeli.
4. W kolejnym kroku otrzymaną liczbę stron mnożymy przez 8196 B (rozmiar pojedynczej strony), co daje nam w wyniku wielkość pliku danych.

28

## Szacowany rozmiar wielkości pliku danych - przykład

O określamy rozmiar pojedynczego wiersza w tabeli **Ksiazki**, **Czytelnicy** oraz **Ksiazki\_Czytelnicy**.

Tabela **Ksiazki**

Kolumna	Typ	Rozmiar
ID_Ksiazka	int	4 bajty
Tytul	varchar(200)	200 bajtów
Autor	varchar(50)	50 bajtów
Rok	char(4)	4 bajty
ID_Czytelnik	int	4 bajty
Suma		262 bajtów

Tabela **Czytelnicy**

Kolumna	Typ	Rozmiar
ID_Czytelnik	int	4 bajty
Nazwisko	varchar(50)	50 bajtów
Imie	varchar(30)	30 bajtów
Data_urodzenia	smalldatetime	4 bajty
ID_Ksiazka	int	4 bajty
Suma		92 bajtów

Tabela **Ksiazki\_Czytelnicy**

Kolumna	Typ	Rozmiar
ID_Ksiazka	int	4 bajty
ID_Czytelnik	int	4 bajty
Suma		8 bajtów

29

## Szacowany rozmiar wielkości pliku danych - przykład

2. Wyznaczamy liczbę wierszy mieszczących się na stronie. W naszym przypadku mamy:

**Ksiazki = 8060 / 262 B = 31 wierszy na stronie**

**Czytelnicy = 8060 / 92 B = 88 wierszy na stronie**

**Ksiazki\_Czytelnicy = 8060 / 8 B = 1008 wierszy na stronie**

3. Przy założeniu, że w tabeli **Ksiazki** będzie znajdowało się 5000 rekordów, w tabeli **Czytelnicy** 2000 rekordów a w tabeli **Ksiazki\_Czytelnicy** 22000 rekordów, wyznaczamy liczbę stron niezbędnych do zapisania danych znajdujących się w tabeli.

**Ksiazka = 5000 rekordów / 31 wiersze = 162 strony**

**Czytelnicy = 2000 rekordów / 88 wierszy = 23 strony**

**Ksiazki\_Czytelnicy = 22000 rekordów / 1008 wierszy = 22 strony**

4. Sumujemy liczbę wszystkich stron

**162 strony + 23 strony + 22 strony = 207 stron**

5. Obliczamy wielkość pliku danych.

**8196 B \* 207 stron = 1696572 B = 1,7 MB**

30

# Typy danych

1

## Typy danych udostępniane przez system

- **Exact numerics**  
– typy liczbowe o dokładnej wartości (np. int, decimal, numeric).
- **Approximate numerics**  
– typy liczbowe przybliżone (np. float, real).
- **Date and time**  
– typy przechowujące daty i czas (np. date, datetime, time).
- **Character strings**  
– typy znakowe (np. char, varchar).
- **Unicode character strings**  
– typy znakowe obsługujące Unicode (np. nchar, nvarchar).
- **Binary strings**  
– typy przechowujące dane binarne (np. binary, varbinary).
- **Other data types**  
– pozostałe typy, np. xml, cursor, sql\_variant.

### Data Type Categories

Data types in SQL Server are organized into the following categories:

Exact numerics	Unicode character strings
Approximate numerics	Binary strings
Date and time	Other data types
Character strings	

2

## Typy danych dokładne numeryczne

Typy_dokladne_numeryczne				
Column Name	Data Type	Length	Precision	Scale
Typ_bigint	bigint	8	19	0
Typ_int	int	4	10	0
Typ_smallint	smallint	2	5	0
Typ_tinyint	tinyint	1	3	0
Typ_money	money	8	19	4
Typ_smallmoney	smallm...	4	10	4
Typ_bit	bit	1	0	0

Exact Numerics	
bigint	numeric
bit	smallint
decimal	smallmoney
int	tinyint
money	

Typ	Zakres	Przechowywanie
bigint	$-2^{63}$ ( $-9\,223\,372\,036\,854\,775\,808$ ) do $2^{63}-1$ ( $9\,223\,372\,036\,854\,775\,807$ )	8 bajtów
int	$-2^{31}$ ( $-2147\,483\,648$ ) do $2^{31}-1$ ( $2147\,483\,647$ )	4 bajty
smallint	$-2^{15}$ ( $-32\,768$ ) do $2^{15}-1$ ( $32\,767$ )	2 bajty
tinyint	0 do 255	1 bajt
money	$-922\,337\,203\,685\,477,5808$ do $922\,337\,203\,685\,477,5807$	8 bajtów
smallmoney	$-214\,748,3648$ do $214\,748,3647$	4 bajty

3

## Typy danych dokładne numeryczne

Exact Numerics	
bigint	numeric
bit	smallint
decimal	smallmoney
int	tinyint
money	

Typ	Opis	Przechowywanie
decimal(p,s)	Liczby o stałej precyzyji i skali. Zakres od $-10^{38}$ do $10^{38}$ . Parametr p określa całkowitą liczbę cyfr (1–38, domyślnie 18). Parametr s określa liczbę cyfr po przecinku (0–p, domyślnie 0).	5–17 bajtów

Typy_dokladne_numeryczne1				
Column Name	Data Type	Length	Precision	Scale
[Typ_decimal_1-9]	decimal(9, 0)	5	Typy_dokladne_numeryczne1	
[Typ_decimal_10-19]	decimal(19, 0)	9	19	0
[Typ_decimal_20-28]	decimal(28, 0)	13	28	0
[Typ_decimal_29-38]	decimal(38, 0)	17	38	0
[Typ_numeric_1-9]	numeric(9, 2)	5	9	2
[Typ_numeric_10-19]	numeric(19, ...)	9	19	12
[Typ_numeric_20-28]	numeric(28, 5)	13	28	5
[Typ_numeric_29-38]	numeric(38, ...)	17	38	30

4

## Typy danych przybliżone numeryczne

### Approximate Numerics

float      real

**Typy\_pryblizone\_numeryczne**

Column Name	Data Type	Length	Precision	Scale
float	float	8	53	0
real	real	4	24	0

Typ	Zakres	Przechowywanie
float	od -1,79E+308 do -2,23E-308, 0 oraz od 2,23E-308 do 1,79E+308	Zależne od wartości n (precyzji)
real	od -3,40E+38 do -1,18E-38, 0 oraz od 1,18E-38 do 3,40E+38	4 bajty

Wartość n	Precyzja	Rozmiar przechowywania
1–24	7 cyfr	4 bajty
25–53	15 cyfr	8 bajtów

5

## Typy danych związanych z datą i godziną

Typ	Opis	Przechowywanie
datetime	Od 1 stycznia 1753 do 31 grudnia 9999 z dokładnością 3,33 ms.	8 bajtów
datetime2	Od 1 stycznia 0001 do 31 grudnia 9999 z dokładnością 100 ns.	6–8 bajtów
smalldatetime	Od 1 stycznia 1900 do 6 czerwca 2079 z dokładnością 1 minutę.	4 bajty
date	Przechowuje tylko datę (od 1 stycznia 0001 do 31 grudnia 9999).	3 bajty
time	Przechowuje tylko czas z dokładnością 100 ns.	3–5 bajtów
datetimeoffset	Jak datetime2, ale z informacją o strefie czasowej.	8–10 bajtów

Date and Time

date	datetimeoffset
datetime2	smalldatetime
datetime	time

**Typy\_data\_godzina**

Column Name	Data Type	Length	Precision	Scale
Typ_datetime	datetime	8	23	3
Typ_datetime2	datetime2(7)	8	27	7
Typ_smalldatetime	smalldatetime	4	16	0
Typ_date	date	3	10	0
Typ_time	time(7)	5	16	7
Typ_datetimeoffset	datetimeoffset(7)	10	34	7
Typ_timestamp	timestamp	8	0	0

6

Typy danych znakowe								
Character Strings				Unicode Character Strings				Typy_znakowe
char	varchar	nchar	nvarchar	Column Name	Data Type	Length	Precision	Scale
text		ntext		Typ_char	char(5)	5	0	0
				Typ_nchar	nchar(5)	5	0	0
				Typ_varchar	varchar(10)	10	0	0
				Typ_nvarchar	nvarchar(10)	10	0	0
				Typ_varcharmax	varchar(MAX)	-1	0	0
				Typ_nvarchar...	nvarchar(MAX)	-1	0	0
				Typ_text	text	16	0	0
				Typ_ntext	ntext	16	0	0

Typ	Opis	Przechowywanie
char(n)	Stała długość ciągu znaków. Maksymalnie 8 000 znaków.	Stała szerokość (n bajtów)
varchar(n)	Zmienna długość ciągu znaków. Maksymalnie 8 000 znaków.	2 bajty + liczba znaków
varchar(max)	Zmienna długość ciągu znaków. Maksymalnie 1 073 741 824 znaków.	2 bajty + liczba znaków
text	Zmienna długość ciągu znaków. Maksymalnie 2 GB danych tekstowych.	4 bajty + liczba znaków
nchar(n)	Stała długość ciągu Unicode. Maksymalnie 4 000 znaków.	Stała szerokość × 2 (każdy znak = 2 bajty)
nvarchar(n)	Zmienna długość ciągu Unicode. Maksymalnie 4 000 znaków.	2 bajty + liczba znaków × 2
nvarchar(max)	Zmienna długość ciągu Unicode. Maksymalnie 536 870 912 znaków.	2 bajty + liczba znaków × 2
ntext	Zmienna długość ciągu Unicode. Maksymalnie 2 GB danych tekstowych.	4 bajty + liczba znaków

7

Typy danych binarne													
Binary Strings													
Typ danych	Opis												
binary(n)	Stała długość ciągu binarnego. Maksymalnie 8 000 bajtów.												
varbinary(n)	Zmienna długość ciągu binarnego. Maksymalnie 8 000 bajtów.												
varbinary(max)	Zmienna długość ciągu binarnego. Maksymalnie 2 GB.												
image	Zmienna długość ciągu binarnego. Maksymalnie 2 GB. (Typ przestarzały – używaj varbinary(max))												

8

Typy danych - pozostałe		Other Data Types
<b>timestamp</b> Przechowuje unikalny numer, który jest aktualizowany za każdym razem, gdy wiersz zostaje utworzony lub zmodyfikowany. Wartość timestamp opiera się na wewnętrznym zegarze i nie odpowiada rzeczywistemu czasowi. Każda tabela może mieć tylko jedną kolumnę typu timestamp.		
<b>Typ danych</b>	<b>Opis</b>	
<b>sql_variant</b>	Przechowuje do 8 000 bajtów danych różnych typów, z wyjątkiem text, ntext i timestamp	
<b>uniqueidentifier</b>	Przechowuje globalnie unikalny identyfikator (GUID) 6F9619FF-8B86-D011-B42D-00C04FC964FF	
<b>xml</b>	Przechowuje dane w formacie XML. Maksymalnie 2 GB	
<b>cursor</b>	Przechowuje referencję do kursora używanego w operacjach bazodanowych	
<b>table</b>	Przechowuje zestaw wyników do późniejszego przetwarzania	
<b>geometry</b>	Przechowuje dane przestrzenne w układzie płaskim (2D)	
<b>geography</b>	Przechowuje dane przestrzenne w układzie elipsoidalnym (np. współrzędne geograficzne)	
<b>timestamp</b>	Przechowuje znacznik czasu używany do wersjonowania wierszy w tabeli	

9

Typy danych - Hierarchical Data (SQL Server)	
• <b>Typ danych hierarchyid</b>	- Wbudowany typ w SQL Server do przechowywania danych hierarchicznych. - Zoptymalizowany do reprezentowania struktur drzewiastych. - Ułatwia wykonywanie zapytań na danych hierarchicznych.
• <b>Definicja danych hierarchicznych</b>	- Zbiór elementów powiązanych relacjami nadzędny–podzędny. - Każdy element może mieć element nadzędny i podzędne.
• <b>Przykłady zastosowań</b>	- Struktura organizacyjna (np. pracownicy → działy). - System plików (foldery → pliki). - Zestaw zadań w projekcie (zadania → podzadania). - Taksonomia terminów językowych. - Wykres powiązań stron internetowych.
<pre>CREATE TABLE Tree ( ID int identity(1,1) Primary Key, Node hierarchyid, Title nvarchar(50) )  select * from Tree insert into Tree (Node,Title) Values(hierarchyid::GetRoot(),'Project Manager'); GO DECLARE @Manager hierarchyid set @Manager=hierarchyid::GetRoot(); insert into Tree (Node,Title) Values(@Manager.GetDescendant(null,null),'Senior Software Developer 1'); select * from Tree</pre>	

10



# Zarządzanie plikami bazy danych (FILESTREAM)

1

## FILESTREAM - Wprowadzenie

Jest to mechanizm dostępny w SQL Server, który pozwala na przechowywanie i zarządzanie dużymi plikami binarnymi (np. zdjęcia, dokumenty, pliki wideo) w bazie danych. Jest to alternatywa dla tradycyjnego przechowywania plików na dysku lub w bazie danych jako BLOB (Binary Large Object).

### Zalety ogólne:

- **Integralność danych:** Pliki są zsynchronizowane z bazą danych, co zapewnia integralność danych.
- **Wysoka wydajność:** Bezpośredni dostęp do plików na dysku zapewnia wysoką wydajność przy pracy z dużymi plikami.
- **Skalowalność:** Łatwe zarządzanie dużą ilością dużych plików.
- **Backup i przywracanie danych:** Pliki FILESTREAM są uwzględniane w procedurach kopii zapasowych i przywracania.
- **Zachowanie spójności i transakcje:** Zintegrowane z mechanizmami transakcji SQL Server.
- **Wsparcie dla wyszukiwania pełnotekstowego:** Umożliwia zaawansowane wyszukiwanie tekstu w przechowywanych plikach.
- **Bezpieczeństwo dostępu:** Konfiguracja uprawnień dostępu do plików.
- **Kompatybilność:** Dostępny w różnych wersjach SQL Server i kompatybilny z narzędziami.

2

## Co może FILESTREAM?

### ZALETY

- Spójność transakcyjna,
- Wszystkie dane w jednym miejscu,
- Jedno środowisko przechowywania i odpytywania danych,
- Oszczędność miejsca w bazie,
- Rozmiar pliku ograniczony tylko przez system operacyjny i miejsce na dysku,
- Wydajne pobieranie danych,
- Zmniejsza rozmiar logu transakcyjnego.
- Integracja z backupem i HA/DR
- Obsługa FILETABLE
- Lepsza skalowalność dla dużych BLOB – nie obciąża bufora stron i mechanizmu I/O SQL Server tak jak VARBINARY(MAX).

3

## Jak przechowywać pliki: w bazie czy w systemie?

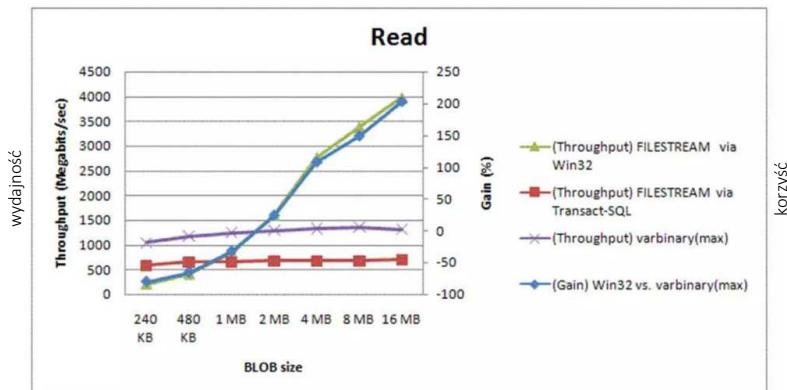
	Dane w bazie	Dane w systemie plików
Zalety:	<ul style="list-style-type: none"><li>• Spójność transakcyjna</li><li>• Wszystkie dane w jednym miejscu</li><li>• Jedno środowisko przechowywania i odpytywania danych</li></ul>	<ul style="list-style-type: none"><li>• Oszczędność miejsca w bazie</li><li>• Rozmiar pliku ograniczony tylko przez system operacyjny i miejsce na dysku</li><li>• Wydajne pobieranie danych</li></ul>
Wady:	<ul style="list-style-type: none"><li>• Duże obiekty potrzebują dużo pamięci<ul style="list-style-type: none"><li>• Varbinary(max) przechowa najwyżej 2GB danych</li><li>• Aktualizacja może spowodować dużą fragmentację bazy danych</li><li>• Duże pliki bazy danych</li></ul></li></ul>	<ul style="list-style-type: none"><li>• Separacja logiczna i fizyczna danych</li><li>• Baza danych nie wie o zmianach w systemie plików</li><li>• Operacje ręczne na poziomie systemu plików wpływają na spójność danych</li><li>• Wymagana dedykowana obsługa plików</li></ul>

4

2

## Czy T-SQL jest wydajny?

- Po przekroczeniu ~1 MB odczyt **FILESTREAM via Win32** (strumieniowanie po API systemu plików) coraz wyraźniej wygrywa z **VARBINARY(MAX)** i **FILESTREAM via T-SQL**, bo korzysta z natywnych mechanizmów I/O i systemowej pamięci podręcznej (NT cache), a nie z bufora stron SQL.
- **FILESTREAM via T-SQL** pozostaje najwolniejszy w miarę wzrostu rozmiaru — wygodny, ale nie wykorzystuje przewagi strumieniowania Win32.



Read performance of various BLOB sizes, źródło: FILESTREAM Storage in SQL Server 2008,  
<https://msdn.microsoft.com/en-us/library/hh461480.aspx>

5

## Kilka zaleceń

- Stosuj **FILESTREAM** dla plików > 1 MB,
- Nie nadużywaj T-SQL, raczej korzystaj z API,
- Użyj kolumny wyliczanej do przechowywania rozmiaru danych,
- Przechowuj pliki **FILESTREAM** na oddzielnych dyskach.

6

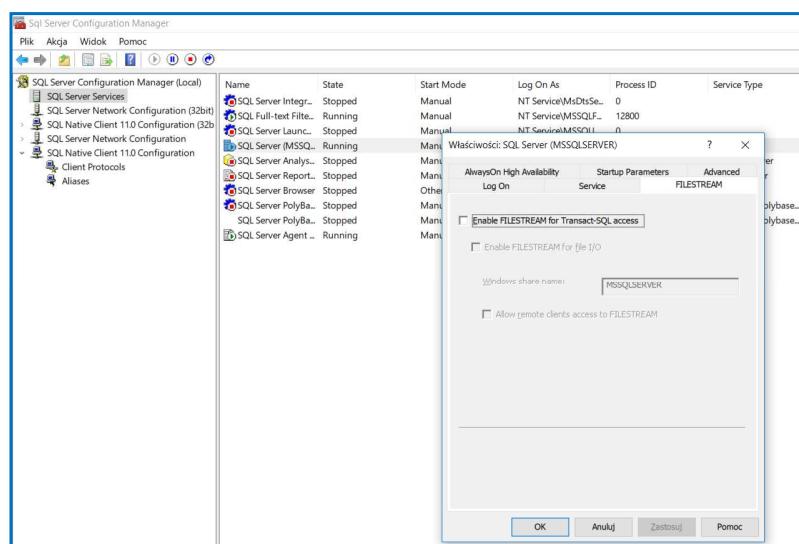
3

## Wady FILESTREAM?

- Database SNAPSHOT nie obsługuje grup FILESTREAM
- Database Mirroring nie obsługuje FILESTREAM
- TDE nie obsługuje szyfrowania plików w grupach FILESTREAM
- Kompresja SQL Server nie dotyczy plików w grupach FILESTREAM

7

## Ustawienie FILESTREAM



8

## Ustawienie FILESTREAM

**Funkcje interfejsu API**  
(ang. Application Programming Interface)  
Wymaga Microsoft.Data.SqlClient i uprawnień do tabeli/kolumny FILESTREAM.  
Kluczowe klasy/funkcje:  
- SqlFileStream,  
- PathName(),  
- GET\_FILESTREAM\_TRANSACTION\_CONTEXT().

9

## Ustawienie FILESTREAM

*EXEC sp\_configure filestream\_access\_level, 2  
RECONFIGURE*

FILESTREAM Access Level  
Sets the FILESTREAM access level.

FILESTREAM Access Level	Full access enabled
FILESTREAM Share Name	MSSQLSERVER
Default Full-Text Language	English
Full-Text Upgrade Option	Import
Max Text Replication Size	65536

<https://learn.microsoft.com/en-us/sql/relational-databases/blob/enable-and-configure-filestream?view=sql-server-ver17>

10

## Dostęp do danych Filestream za pomocą T-SQL

Za pomocą języka Transact-SQL dane FILESTREAM można wstawać, aktualizować i usuwać w następujący sposób:

- **Wstawianie pól FILESTREAM:**

Pola FILESTREAM można wypełnić operacją INSERT (pustą lub małą, niepustą wartością), lecz efektywniejsze jest korzystanie z interfejsów Win32 do przesyłania dużych ilości danych.

- **Aktualizacja danych FILESTREAM:**

Aktualizacja danych FILESTREAM modyfikuje BLOB w systemie plików. Ustawienie pola FILESTREAM na NULL skutkuje usunięciem związków z nim danych BLOB. Operacje aktualizacji Transact-SQL nie są użyteczne do częściowej aktualizacji danych FILESTREAM.

- **Usuwanie danych FILESTREAM:**

Usunięcie wiersza lub tabeli z danymi FILESTREAM powoduje usunięcie danych BLOB w systemie plików. Faktyczne usuwanie plików FILESTREAM to proces asynchroniczny

11

## Strumieniowy dostęp do danych FILESTREAM

Dostęp do danych FILESTREAM za pomocą interfejsów strumieniowych wymaga poniższych kroków, niezależnie od używanej biblioteki klienta czy języka programowania:

**1. Rozpocznij transakcję:**

Aby uzyskać dostęp do danych FILESTREAM poprzez strumieniowe API, konieczne jest rozpoczęcie transakcji.

**2. Pobierz nazwę ścieżki i kontekst transakcji:**

Należy uzyskać logiczną nazwę ścieżki do pliku danych FILESTREAM oraz wskaźnik do bieżącego kontekstu transakcji, aby uzyskać dostęp do danych FILESTREAM.

**3. Otwórz plik danych FILESTREAM:**

Uzyskaj dostęp do pliku danych FILESTREAM za pomocą klasy opakowującej .NET lub interfejsu API FILESTREAM w Win32.

**4. Odczytaj/Zapisz dane FILESTREAM:**

Odczytuj lub zapisuj dane FILESTREAM przy użyciu funkcji wejścia/wyjścia pliku.

**5. Zamknij plik danych FILESTREAM:**

Zamknij plik danych FILESTREAM w sposób kontrolowany.

**6. Zatwierdź/Anuluj transakcję:**

Zatwierdź lub anuluj transakcję po zakończeniu operacji.

12

## Grupa plików - FILESTREAM

**Database Properties - A01**

**Select a page:**

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Mirroring
- Transaction Log Shipping
- Query Store

**Rows**

Name	Files	Read-Only	Default	Autogrow All Files
PRIMARY	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>

**FILESTREAM**

Name	FILESTREAM Files	Read-Only	Default
file01	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Connection**

Server: DELL  
Connection: sa  
[View connection properties](#)

**MEMORY OPTIMIZED DATA**

Name	FILESTREAM Files

```

ALTER DATABASE A01
ADD FILEGROUP file01
CONTAINS FILESTREAM;
GO

ALTER DATABASE A01
ADD FILE (
    NAME = file02,
    FILENAME = 'C:\FILESTREAM\Data')
TO FILEGROUP file01;
GO

```

13

## Katalog na pliki - FILESTREAM

**Database Properties - A01**

**Select a page:**

- General
- Files
- Filegroups
- Options
- Change Tracking
- Permissions
- Extended Properties
- Mirroring
- Transaction Log Shipping
- Query Store

**Database name:** A01  
**Owner:** sa  
 Use full-text indexing

**Database files:**

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize	Path
A01	ROWS Data	PRIMARY	8	By 64 MB, Unlimited	<input type="button"/> C:\Program Files\Micro
file02	FILESTREAM Data	file01	1	Unlimited	<input type="button"/> C:\Program Files\Micro
A01_Log	LOG	Not Applicable	8	By 64 MB, Limited to 2097152 MB	<input type="button"/> C:\Program Files\Micro

```

ALTER DATABASE A01
ADD FILEGROUP file01
CONTAINS FILESTREAM;
GO

ALTER DATABASE A01
ADD FILE (
    NAME = file02,
    FILENAME = 'C:\FILESTREAM\Data')
TO FILEGROUP file01;
GO

```

14

## VARBINARY(MAX) FILESTREAM

```
CREATE TABLE dbo.Records01
(
[Id] [uniqueidentifier] ROWGUIDCOL NOT NULL UNIQUE,
[SerialNumber] INTEGER UNIQUE,
[Chart] VARBINARY(MAX) FILESTREAM NULL
) ON [PRIMARY] FILESTREAM_ON [file01]
GO
```

```
CREATE TABLE [dbo].[Authors]
(
[AuthorID] UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL
UNIQUE ,
[AuthorName] VARCHAR(50) ,
[AuthorImage] VARBINARY(MAX) FILESTREAM NULL
)
GO
```

15

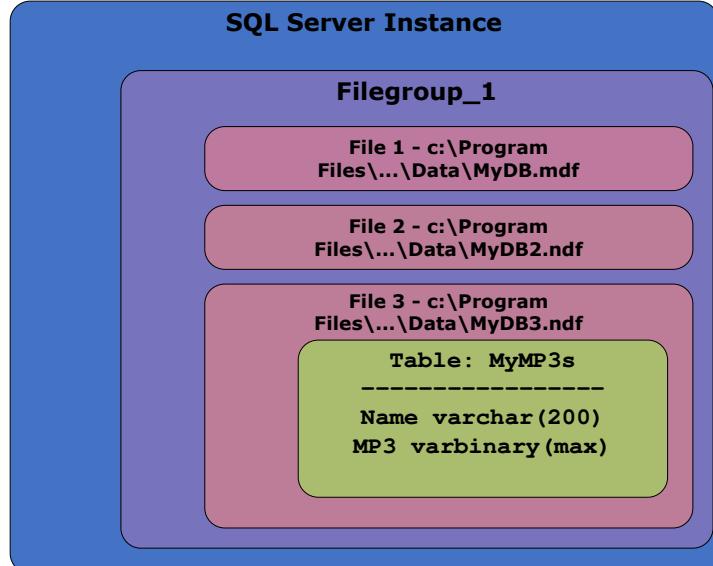
## VARBINARY(MAX) FILESTREAM

```
CREATE TABLE [dbo].[Authors]
(
[AuthorID] UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL UNIQUE,
[AuthorName] VARCHAR(50) ,
[AuthorImage] VARBINARY(MAX) FILESTREAM NULL
)
GO
-- Insert the data to the table
INSERT INTO [dbo].[Authors]
( AuthorID , AuthorName , AuthorImage)
SELECT NEWID() , 'PhilFactor' ,
bulkcolumn FROM OPENROWSET(BULK 'C:\temp\dane.jpg',
SINGLE_BLOB) AS x
GO
SELECT * FROM dbo.authors
```

AuthorID	AuthorName	AuthorImage
3EC18F5B-9C27-467E-B33A-46121BBC56D	PhilFactor	0xFFD8FFE113104578696600049492A0008000000A000F...

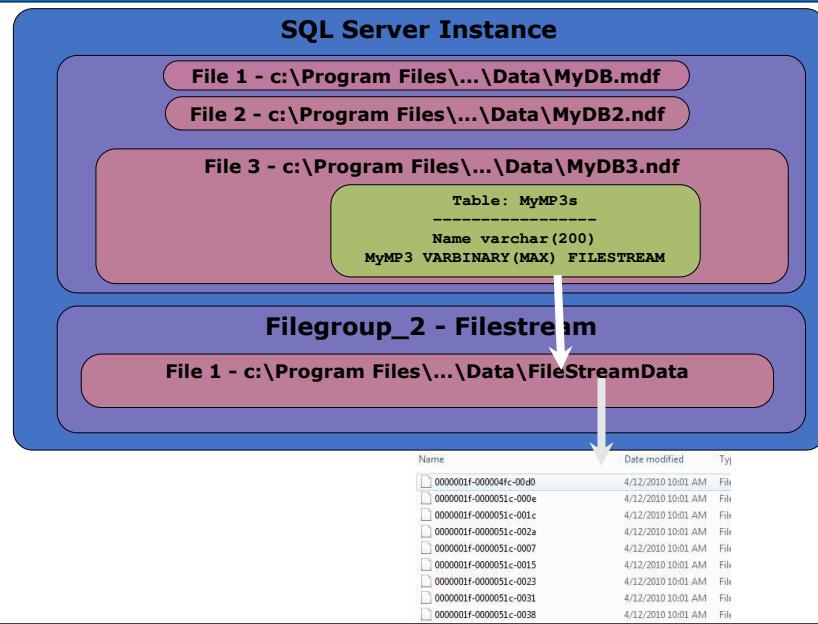
16

## FILESTREAM



17

## FILESTREAM



18

## Porównanie technologii pamięci masowej BLOB

Punkt porównawczy	Rozwiązanie do przechowywania danych		
	Serwer plików / system plików	SQL Server (przy użyciu funkcji varbinary(max))	FileStream
Maksymalny rozmiar obiektu BLOB	Rozmiar woluminu NTFS	2 GB	Rozmiar woluminu NTFS
Wydajność przesyłania strumieniowego dużych obiektów BLOB	Doskonałe	Słabe	Doskonałe
Bezpieczeństwo	Ręczne listy ACL	Zintegrowany	Zintegrowane + automatyczne listy ACL
Koszt za GB	Niski	Wysoki	Niski
Zarządzania	Trudne	Zintegrowane	Zintegrowane
Integracja z danymi strukturalnymi	Trudny	Spójność na poziomie danych	Spójność na poziomie danych
Tworzenie i wdrażanie aplikacji	Bardziej złożone	Prostsze	Prostsze
Odzyskiwanie danych po fragmentacji danych	Doskonałe	Słabe	Doskonałe
Wydajność częstych małych aktualizacji	Doskonałe	Umiarkowany	Słabe

19

# Zarządzanie bezpieczeństwem

[Secure your SQL Server - SQL Server | Microsoft Learn](#)

[SQL Server security best practices - SQL Server | Microsoft Learn](#)

[Determine effective Database Engine permissions - SQL Server | Microsoft Learn](#)

1

## Informacje podstawowe

### **Przy zarządzaniu zabezpieczeniami trzeba zapewnić:**

- Rozwiązania zapewniające potrzebę użytkowników, a także ochronę danych przed osobami nieupoważnionymi
- Ograniczenie uprawnień w taki sposób, aby zmniejszyć prawdopodobieństwo złośliwego lub przypadkowego wykonania szkodliwych poleceń przez użytkowników
- Usunięcie ewentualnych luk w systemie zabezpieczeń np. wynikających z przynależności użytkowników do grupy Administratorzy.

2

1

## Informacje podstawowe

Zarządzanie bezpieczeństwem w SQL Server wymaga zapewnienia kilku kluczowych obszarów, aby chronić dane i kontrolować dostęp.

### 1. Uwierzytelnianie (Authentication)

- **Tryby uwierzytelniania:**
  - **Windows Authentication** – zalecane, korzysta z mechanizmów domeny i Kerberos.
  - **SQL Server Authentication** – wymaga silnych haseł i polityki ich wygasania.
- **Zasady haseł:**
  - Włącz politykę haseł systemu Windows (złożoność, długość, historia).
  - Wymuszaj szyfrowanie połączeń (SSL/TLS).

3

## Informacje podstawowe

### 2. Autoryzacja (Authorization)

- **Role serwera** (np. sysadmin, securityadmin) – nadawaj tylko niezbędne.
- **Role bazy danych** (np. db\_owner, db\_datareader) – stosuj zasadę minimalnych uprawnień.
- Twórz **role użytkownika** zamiast nadawać uprawnienia indywidualnie.
- Używaj **GRANT, DENY, REVOKE** do zarządzania uprawnieniami.

4

## Informacje podstawowe

### **3. Zarządzanie loginami i użytkownikami**

- Oddziel loginy serwera od użytkowników bazy danych.
- Monitoruj i usuwaj nieużywane konta.
- Stosuj kontrolę dostępu opartą na rolach (RBAC Role-Based Access Control).

### **4. Szyfrowanie i ochrona danych**

- Transparent Data Encryption (TDE) – szyfrowanie plików bazy.
- Always Encrypted – szyfrowanie kolumn wrażliwych.
- Szyfrowanie połączeń – wymuszaj protokół TLS.

5

## Informacje podstawowe

### **5. Audyt i monitorowanie**

- Włącz SQL Server Audit – śledzenie logowań, zmian uprawnień, operacji DML.
- Monitoruj failed logins i nietypowe aktywności.
- Używaj Extended Events lub Profiler do analizy.

### **6. Aktualizacje i konfiguracja**

- Regularnie instaluj aktualizacje bezpieczeństwa.
- Wyłącz niepotrzebne funkcje (np. xp\_cmdshell).
- Włącz firewall i ogranicz dostęp do portu 1433.

### **7. Kopie zapasowe i odzyskiwanie**

- Szyfruj kopie zapasowe.
- Przechowuj je w bezpiecznej lokalizacji.
- Testuj procedury odzyskiwania.

6

## Zabezpieczenia w SQL Server – Kluczowe elementy

### 1. Tryby uwierzytelniania (Authentication)

- **Windows Authentication** – korzysta z mechanizmów domeny i Kerberos, zalecany.
- **Mixed Mode (Windows + SQL Authentication)** – umożliwia logowanie zarówno przez konta Windows, jak i konta SQL Server.

### 2. Konta serwerowe (Loginy)

- Loginy serwera umożliwiają dostęp do instancji SQL Server.
- Mogą być:
  - **Windows logins** (użytkownicy/domeny)
  - **SQL logins** (zdefiniowane w SQL Server)

### 3. Uprawnienia (Authorization)

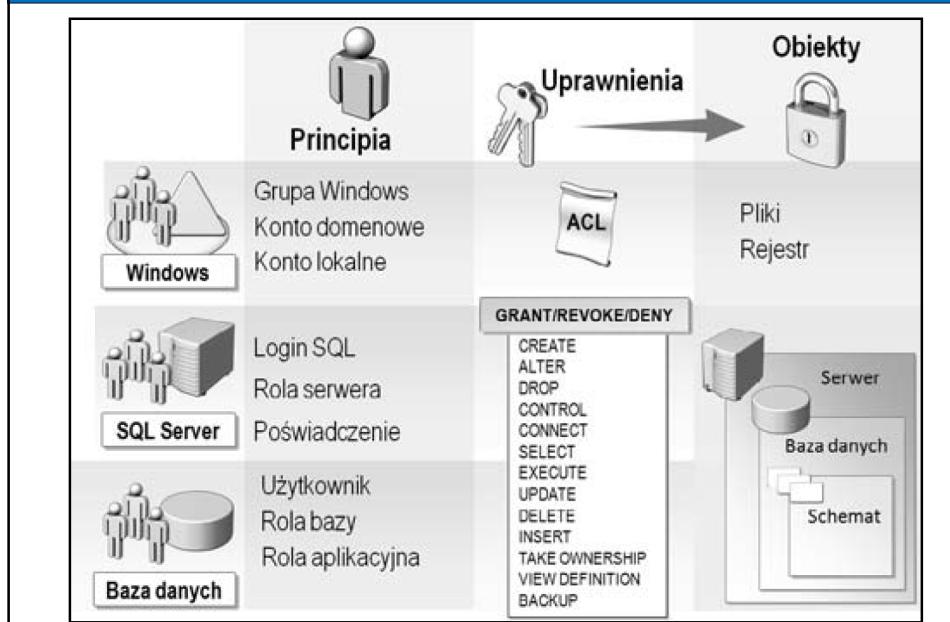
- **Na obiektach** – np. SELECT, INSERT, UPDATE na tabelach.
- **Wykonywanie poleceń** – np. EXECUTE na procedurach.
- **Domniemane** – wynikające z członkostwa w rolach lub schematach.

### 4. Role

- **Role serwera** – np. sysadmin, securityadmin, dbcreator.
- **Role bazy danych** – np. db\_owner, db\_datareader, db\_datawriter.
- Możliwość tworzenia **ról użytkownika** dla grupowania uprawnień.

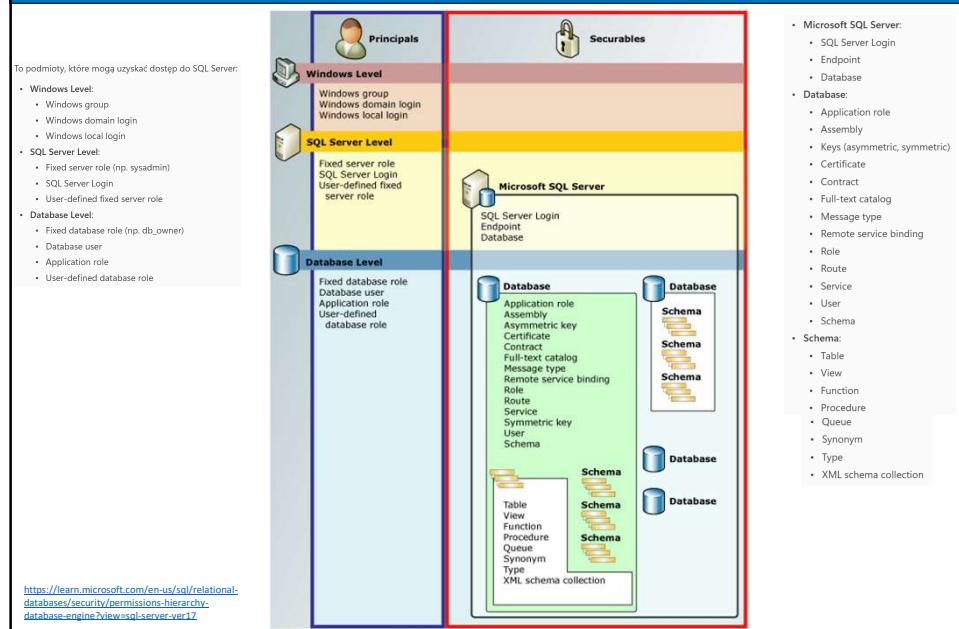
7

## Model bezpieczeństwa



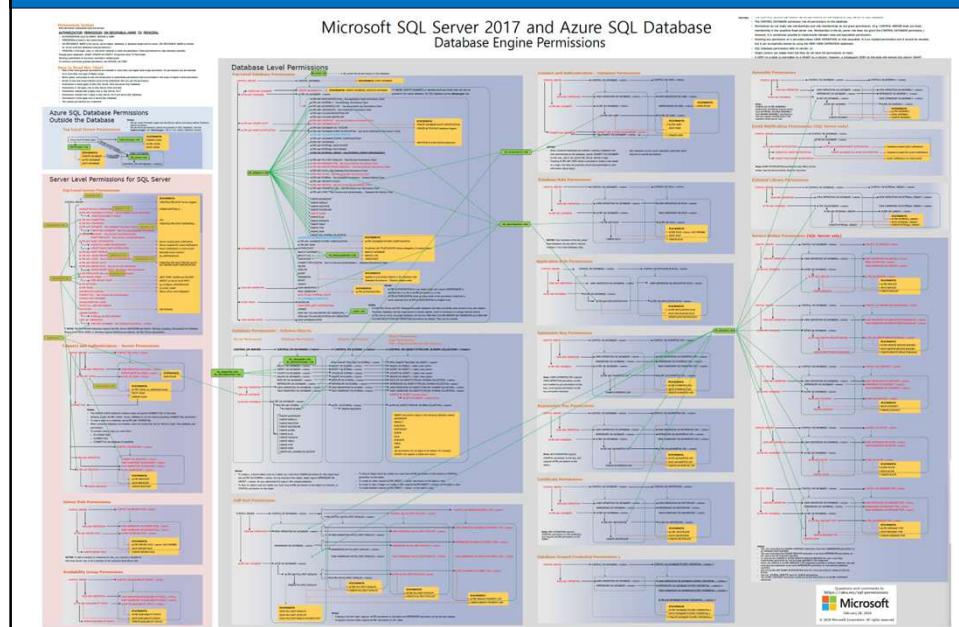
8

## Model bezpieczeństwa



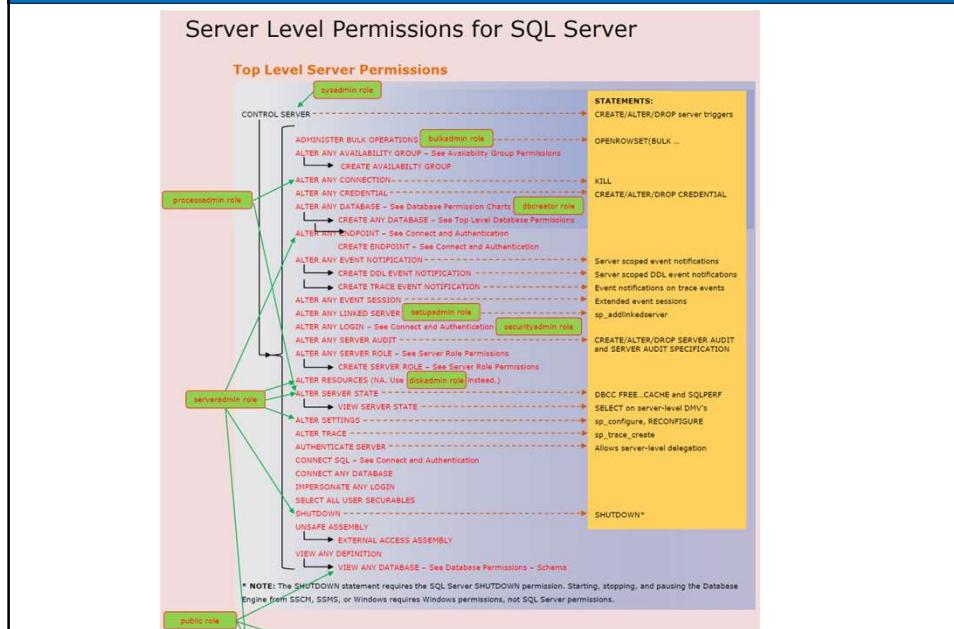
9

# Model bezpieczeństwa



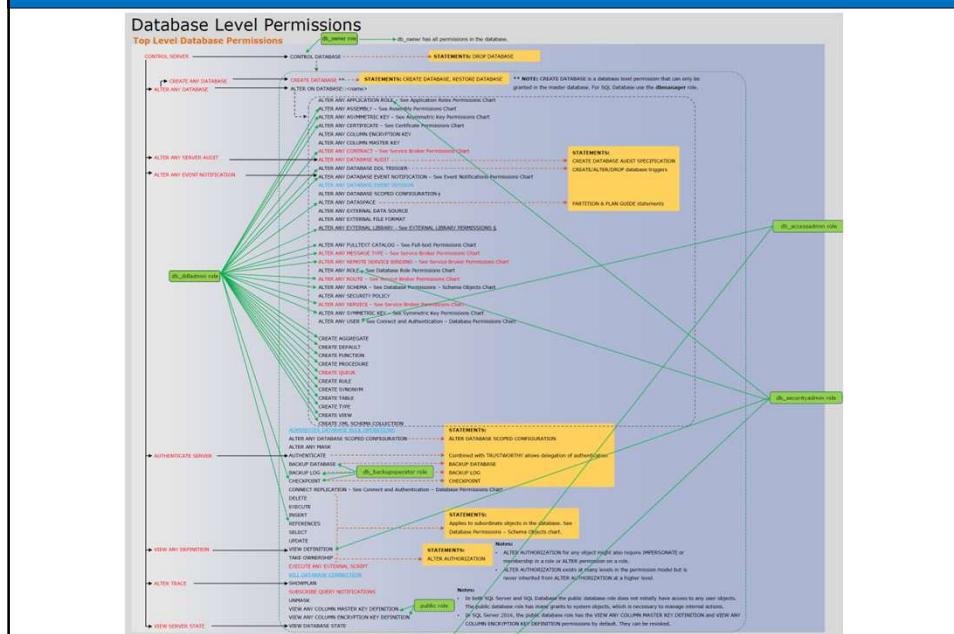
10

# Model bezpieczeństwa



11

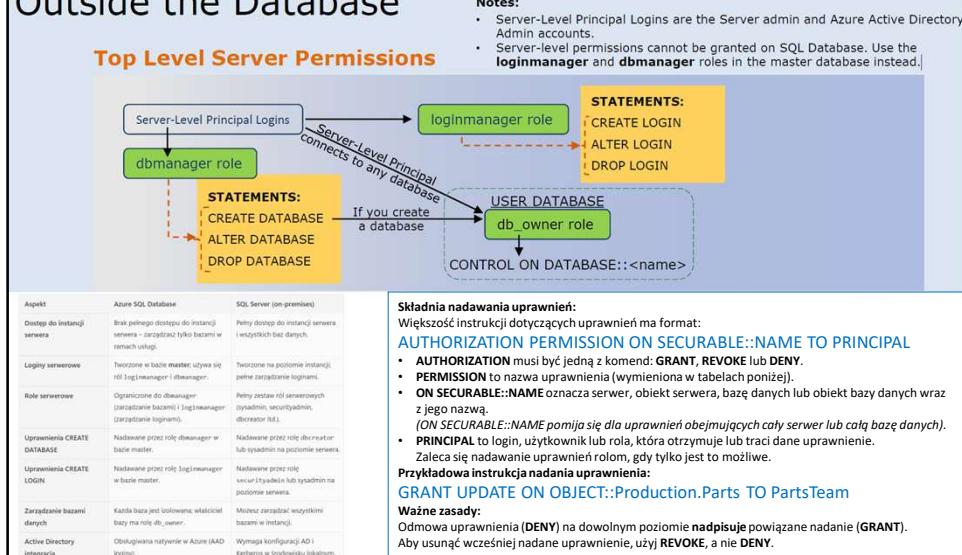
## Model bezpieczeństwa



12

## Model bezpieczeństwa

### Azure SQL Database Permissions Outside the Database



13

- Etapy zarządzania bezpieczeństwem w SQL Server

Wybór trybu uwierzytelniania

Przypisanie kont logowania do użytkowników i ról

Przypisanie uprawnień użytkownikom i rolom

Planowanie bezpieczeństwa

Role aplikacyjne

14

- Implementacja trybu uwierzytelniania

Proces uwierzytelniania

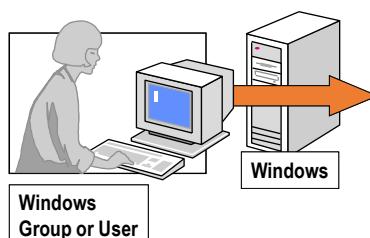
Wybranie trybu uwierzytelniania

Kolejne kroki w implementowaniu trybu uwierzytelniania

Tworzenie kont logowania

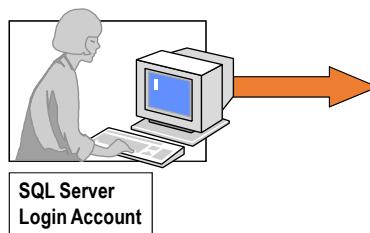
15

## Proces uwierzytelniania



Weryfikuje wpisy w tablicy syslogins ; ufa że Windows zweryfikowało hasło

syslogins			



Weryfikuje wpisy w tablicy syslogins i weryfikuje hasło

syslogins			

16

## Wybór trybu identyfikacji

### Zalety trybu identyfikacji Windows (Windows Authentication)

- zaawansowane cechy bezpieczeństwa
- dodawanie grupy jako jednego konta
- szybki dostęp

### Zalety trybu mieszanego (SQL Server Authentication)

- dostęp dla użytkowników innych niż Windows i dla użytkowników Internetu
- dodatkowa warstwa bezpieczeństwa ponad Windows

17

## Kroki w implementacji trybu identyfikacji

1 Zweryfikuj protokół połączenia (autentykacja Windows)

2 Ustaw tryb uwierzytelniania

3 Zaloguj się do serwera

4 Stwórz grupy i użytkowników Windows

5 Upoważnij grupy i użytkowników Windows do logowania w SQL Server

6 Stwórz konta SQL Server dla użytkowników, którzy łączą się przy użyciu trybu autentykacji SQL Server

18

## Tworzenie kont logowania

1  
2     `select sid, name from master..syslogins order by name desc`

120 % No issues found

Results Messages

sid	name
1 0x010100000000000514000000	ZARZĄDZANIE NT\USŁUGA SIECIOWA
2 0x010100000000000512000000	ZARZĄDZANIE NT\SYSTEM
3 0x01	sa
4 0x010600000000000500000005A048DDFF9C7430AB450D4E...	NT SERVICE\Winmgmt
5 0x01060000000000050000000732B9753646EF90356745CB6...	NT SERVICE\SQLWriter
6 0x010600000000000500000000DC488F14B79FD47A992A3D8...	NT SERVICE\SQLSERVERAGENT
7 0x010600000000000500000000E20F4FE7B15874E48E190264...	NT Service\MSSQLSERVER
8 0x01050000000000051500000A018D5D7841D4667D80E110...	IMSI\welfie

`select * from sys.syslogins`  
- loginy SQL (polityka, wygaśnięcie, domyślne DB/ język)  
`select * from sys.server_principals`  
- wszystkie loginy (SQL, Windows, cert, klucz asym.)

Security  
Logins  
##MS\_PolicyEventProcessingLogin##  
##MS\_PolicyTsqlExecutionLogin##  
IMSI\adminjs  
IMSI\welfie  
NT Service\MSSQLSERVER  
NT SERVICE\SQLSERVERAGENT  
NT SERVICE\SQLWriter  
NT SERVICE\Winmgmt  
sa  
ZARZĄDZANIE NT\SYSTEM  
ZARZĄDZANIE NT\USŁUGA SIECIOWA

19

## Tworzenie kont logowania

MS SQL Server 2000 - deprecated

//SQL Server Authentication

```
sp_addlogin
[ @loginame = ] N'loginame'
[ , [ @passwd = ] N'passwd' ]
[ , [ @defdb = ] N'defdb' ]
[ , [ @deflanguage = ] N'deflanguage' ]
[ , [ @sid = ] sid ]
[ , [ @encryptopt = ] 'encryptopt' ]
[ ; ]
```

//Windows Authentication

```
sp_grantlogin [ @loginame = ] N'loginame'
```

20

## Tworzenie kont logowania

### MS SQL Server od wersji 2005

```
-- Syntax for SQL Server
CREATE LOGIN login_name { WITH <option_list1> | FROM <sources> }

<option_list1> ::=
    PASSWORD = { 'password' | hashed_password HASHED } [ MUST_CHANGE ]
    [ ,<option_list2> [ ,... ] ]

<option_list2> ::=
    SID = sid
    | DEFAULT_DATABASE = database
    | DEFAULT_LANGUAGE = language
    | CHECK_EXPIRATION = { ON | OFF }
    | CHECK_POLICY = { ON | OFF }
    | CREDENTIAL = credential_name

<sources> ::=
    WINDOWS [ WITH <windows_options>[ ,... ] ]
    | EXTERNAL PROVIDER [WITH OBJECT_ID = 'objectid']
    | CERTIFICATE certname
    | ASYMMETRIC KEY asym_key_name

<windows_options> ::=
    DEFAULT_DATABASE = database
    | DEFAULT_LANGUAGE = language
```

//SQL Server Authentication

//Windows Authentication

//EXTERNAL - uwierzytelnianie przez zewnętrzny provider,

//CERTIFICATE - obiekt kryptograficzny do szyfrowania/podpisywania,

//ASYMMETRIC KEY - para kluczy (RSA) do szyfrowania i podpisywania.

21

## Przypisanie kont logowania do użytkowników i ról bazy

Prawa do bazy są przechowywane w

use Northwind  
select \* from sysprotects  
select \* from sysusers

*northwind..sysprotects*

sid	uid	action	protecttype
117575457	0	193	205
117575457	0	195	205
117575457	0	196	205
117575457	7	193	205

Użytkownicy bazy są przechowywani w

*northwind..sysusers*

uid	name
0	public
1	dbo
3	INFORMATION_SCHEMA
4	sys
7	payroll

id	sid	action	protecttype	columns	grantor
8	1410104064	2	205	NULL	1
9	1429104121	0	224	205	NULL
10	1429104121	2	224	206	NULL
11	1442104178	0	224	205	NULL
12	1442104178	2	224	206	NULL
13	1459116452	0	224	205	NULL
14	1459116452	2	205	NULL	1
15	667	0	193	205	0x01

*sys.syslogins*

22

11

## Przypisanie kont logowania do kont użytkowników bazy

- Dodanie kont użytkowników bazy danych  
(SQL Server 2000)

- *sp\_grantdbaccess*

- Dodanie kont użytkowników bazy danych  
(od SQL Server 2005)

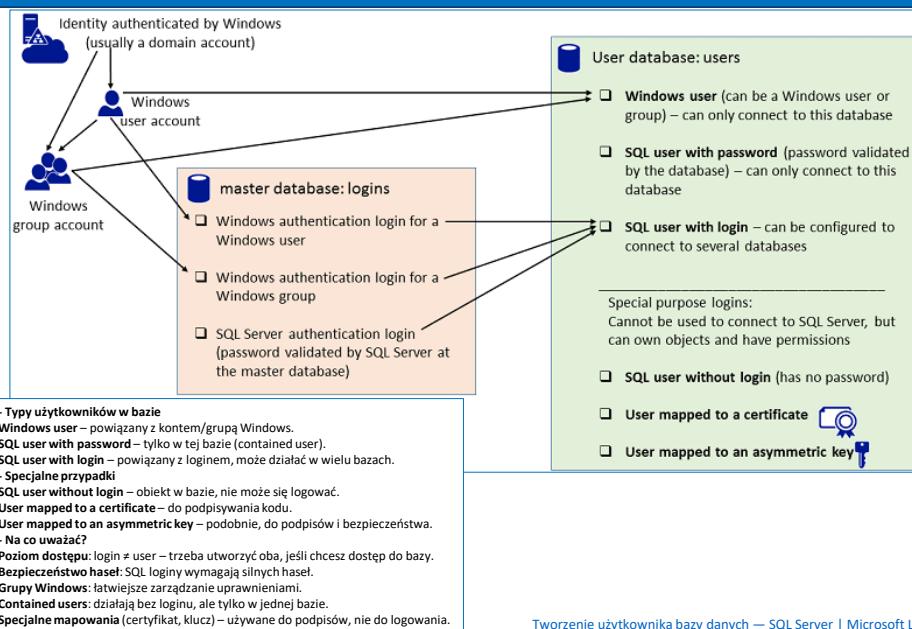
- ***CREATE USER***

- Konto użytkownika dbo

- Konto użytkownika guest

23

## Konta użytkownika bazy danych



24

## Konta użytkownika bazy danych – Contained user

Na serwerze (instancji) włącz uwierzytelnianie contained:

```
-- W SSMS na instancji:  
EXEC sp_configure 'show advanced options', 1;  
RECONFIGURE;  
EXEC sp_configure 'contained database authentication', 1;  
RECONFIGURE;
```

### Logowanie przez SSMS

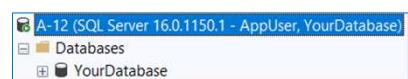
- **Server name:** Serwer\Instancja albo tcp:nazwa\_serwera, port
- **Authentication:** SQL Server Authentication
  - **Login:** AppUser (to jest contained user w bazie)
  - **Password:** SilneHaslo#2025
- Kliknij Options » Connection Properties i ustaw: **Connect to database:** YourDatabase - to ważne, bo użytkownik istnieje tylko w tej bazie

W docelowej bazie ustaw containment (PARTIAL):

```
ALTER DATABASE YourDatabase SET CONTAINMENT = PARTIAL;
```

Utwórz użytkownika w bazie:

```
USE YourDatabase;  
CREATE USER AppUser WITH PASSWORD = 'SilneHaslo#2025';  
-- przyznaj minimalne uprawnienia  
ALTER ROLE db_datareader ADD MEMBER AppUser;
```



25

## Przypisanie kont do ról bazy danych

Role serwera (wbudowane)

Role serwera zdefiniowane przez użytkownika

Role bazy danych (wbudowane)

Role bazy danych zdefiniowane przez użytkownika

26

## Predefiniowane role serwera

Rola	Uprawnienia
bulkadmin	członkowie roli mogą przeprowadzać masowe ładowanie danych (mogą wykonać instrukcję BULK INSERT).
dbcreator	członkowie roli mogą tworzyć i modyfikować bazy danych
diskadmin	członkowie roli mogą zarządzać plikami i urządzeniami pamięci masowej.
processadmin	członkowie roli mogą zarządzać procesami SQL Servera (mogą wykonać instrukcję KILL).
securityadmin	członkowie roli mogą zarządzać dostępem użytkowników do SQL Servera.
serveradmin	członkowie roli mogą konfigurować parametry pracy serwera bazodanowego, w tym zatrzymywać serwer.
setupadmin	członkowie roli mogą zarządzać replikacją danych pomiędzy serwerami.
sysadmin	członkowie roli są administratorami serwera bazodanowego i mogą wykonywać każdą operację.

27

## Predefiniowane role serwera

```
-- Dodawanie do danej roli serwera użytkowników
ALTER SERVER ROLE [sysadmin] ADD MEMBER user1;

-- Usuwanie z danej roli serwera użytkowników
ALTER SERVER ROLE [sysadmin] DROP MEMBER user1;

-- Przeglądanie szczegółowych praw dla wybranej roli wbudowanej
EXEC sp_srvrolepermission 'bulkadmin';

-----
-- Tworzenie roli serwerowej zdefiniowanej przez użytkownika
CREATE SERVER ROLE [MyServerRole01];

-- Dodawanie do danej roli serwera użytkowników
ALTER SERVER ROLE [MyServerRole01] ADD MEMBER [user1];

-- Dodawanie do danej roli serwera uprawnień szczegółowych
GRANT CONNECT SQL TO [MyServerRole01];

-- Dodawanie do danego użytkownika serwera (loginu) uprawnień szczegółowych
GRANT ALTER ANY DATABASE TO [user1];
```

```
EXEC master..sp_addsrvrolemember @loginame = 'u1', @rolename = 'sysadmin'
EXEC master..sp_dropsrvrolemember @loginame = 'u1', @rolename = 'sysadmin'
```

28

## Predefiniowane role serwera

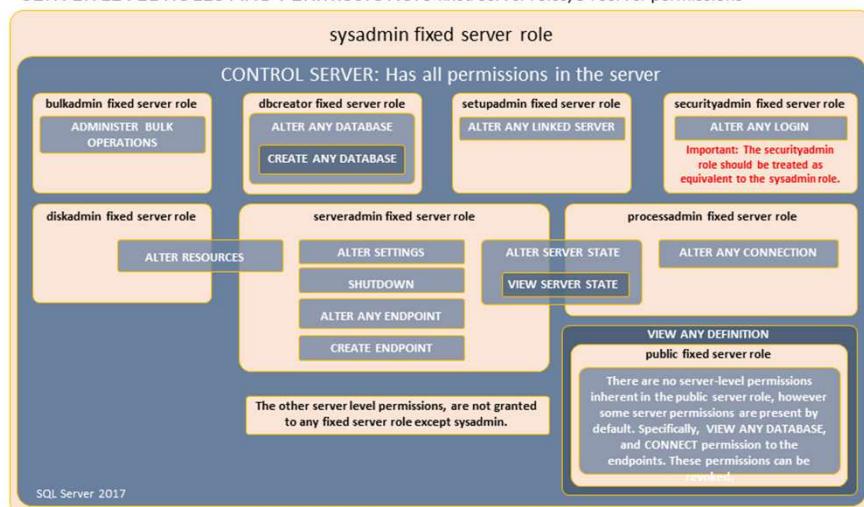
The figure consists of four separate SSMS result sets:

- bulkadmin:** Shows permission to add members to the bulkadmin role and BULK INSERT permission.
- processadmin:** Shows permission to add members to the processadmin role and KILL permission.
- diskadmin:** Shows permission to add members to the diskadmin role and DISK INIT, sp\_addumpdevice, sp\_diskdefault, and sp\_dropdevice permissions.
- dbcreator:** Shows a long list of permissions including ADD MEMBER TO securityadmin, GRANT/DENY/REVOKE CREATE DATABASE, READ THE ERROR LOG, SP\_ADDLINKEDSRVLOGIN, SP\_ADDLOGIN, SP\_DEFAULTDB, SP\_DEFAULTLANGUAGE, SP\_DENYLOGIN, SP\_DROPLINKEDSRVLOGIN, SP\_DROPLOGIN, SP\_DROPREMOTELOGIN, SP\_GRANTLOGIN, SP\_HELPLOGINS, SP\_PASSWORD, SP\_REMOTEOPTION (UPDATE), and SP\_REVOKELOGIN.

29

## Predefiniowane role serwera

SERVER LEVEL ROLES AND PERMISSIONS: 9 fixed server roles, 34 server permissions



30

## Polecenia związane z rolami serwera

Feature	Type	Description
<code>sp_helpsrvrole</code>	Metadata	Returns a list of server-level roles.
<code>sp_helpsrvrolemember</code>	Metadata	Returns information about the members of a server-level role.
<code>sp_srvrolepermission</code>	Metadata	Displays the permissions of a server-level role.
<code>IS_SRVROLEMEMBER</code>	Metadata	Indicates whether a SQL Server login is a member of the specified server-level role.
<code>sys.server_role_members</code>	Metadata	Returns one row for each member of each server-level role.
<code>CREATE SERVER ROLE</code>	Command	Creates a user-defined server role.
<code>ALTER SERVER ROLE</code>	Command	Changes the membership of a server role or changes name of a user-defined server role.
<code>DROP SERVER ROLE</code>	Command	Removes a user-defined server role.
<code>sp_addsrvrolemember</code>	Command	Adds a login as a member of a server-level role. Deprecated. Use <code>ALTER SERVER ROLE</code> instead.
<code>sp_dropsrvrolemember</code>	Command	Removes a SQL Server login or a Windows user or group from a server-level role. Deprecated. Use <code>ALTER SERVER ROLE</code> instead.

<https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/server-level-roles?view=sql-server-2017>

31

## Predefiniowane role bazy danych

Rola	Uprawnienia
<code>db_accessadmin</code>	członkowie roli mogą zarządzać dostępem użytkowników do bazy danych.
<code>db_backupoperator</code>	członkowie roli mogą tworzyć kopie zapasowe bazy danych.
<code>db_datareader</code>	członkowie roli mogą odczytywać dane zapisane w dowolnej tabeli bazy danych.
<code>db_datawriter</code>	członkowie roli mogą modyfikować dane zapisane w dowolnej tabeli bazy danych.
<code>db_ddladmin</code>	członkowie roli mogą tworzyć, modyfikować i usuwać obiekty bazodanowe.
<code>db_denydatareader</code>	członkowie roli nie mogą odczytywać danych zapisanych w jakiejkolwiek tabeli bd.
<code>db_denydatawriter</code>	członkowie roli nie mogą modyfikować danych zapisanych w jakiejkolwiek tabeli bd.
<code>db_owner</code>	członkowie roli są administratorami bd i mogą w niej wykonywać każdą operację.
<code>db_security admin</code>	członkowie roli mogą nadawać i odbierać uprawnienia użytkownikom.
<code>public</code>	specjalna rola serwera, której członkami są wszyscy użytkownicy bazy danych.

32

## Predefiniowane role bazy danych

```
-- Dodawanie do danej roli bazy danych użytkowników bd  
ALTER ROLE [db_owner] ADD MEMBER user_db_1;  
  
-- Usuwanie z danej roli bazy danych użytkowników bd  
ALTER ROLE [db_owner] DROP MEMBER user_db_1;  
  
-- Przeglądanie szczegółowych praw dla wybranej roli wbudowanej  
EXEC sp_dbfixedrolepermission 'db_datawriter';  
-----  
-----  
-- Tworzenie roli bazy danych zdefiniowanej przez użytkownika  
CREATE ROLE [MyDbRole01];  
  
-- Dodawanie do danej roli bazy danych użytkowników  
ALTER ROLE [MyDbRole01] ADD MEMBER [user_db_1];  
  
-- Dodawanie do danej roli bazy danych uprawnień szczegółowych  
GRANT UPDATE ON [Tab1] TO [MyDbRole01];  
  
-- Dodawanie do danego użytkownika bazy danych uprawnień szczegółowych  
GRANT UPDATE ON [Tab1] TO [user_db_1];
```

```
EXEC sp_addrolemember 'db_owner', 'u1'  
EXEC sp_droptablemember 'db_owner', 'u1'
```

33

## Predefiniowane role bazy danych

EXEC sp_dbfixedrolepermission 'db_backupoperator'	
DbFixedRole	Permission
db_backupoperator	BACKUP DATABASE
db_backupoperator	BACKUP LOG
db_backupoperator	CHECKPOINT

EXEC sp_dbfixedrolepermission 'db_ddadmin'	
DbFixedRole	Permission
db_ddadmin	All DDL but GRANT, REVOKE, DENY
db_ddadmin	dbcc cleantable
db_ddadmin	dbcc show_statistics
db_ddadmin	dbcc showcontig
db_ddadmin	REFERENCES permission on any table
db_ddadmin	sp_changeobjectowner
db_ddadmin	sp_fulltext_column
db_ddadmin	sp_fulltext_table
db_ddadmin	sp_recompile
db_ddadmin	sp_rename
db_ddadmin	sp_tableoption
db_ddadmin	TRUNCATE TABLE

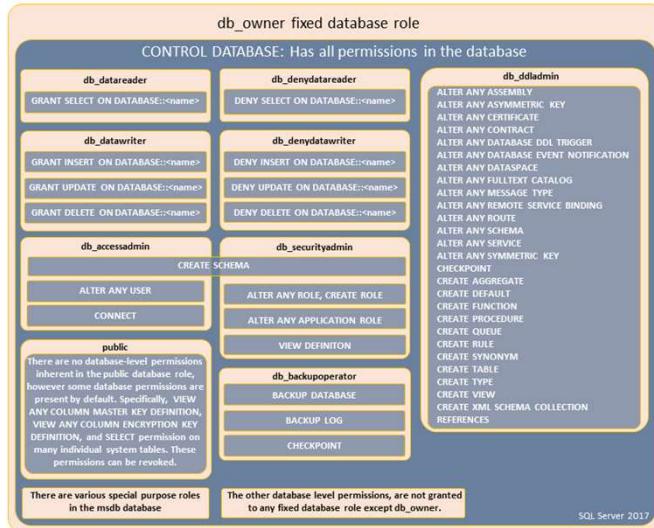
  

EXEC sp_dbfixedrolepermission 'db_datawriter'	
DbFixedRole	Permission
db_datawriter	DELETE permission on any object
db_datawriter	INSERT permission on any object
db_datawriter	UPDATE permission on any object

34

## Predefiniowane role bazy danych

DATABASE LEVEL ROLES AND PERMISSIONS: 11 fixed database roles, 77 database permissions



<https://docs.microsoft.com/en-us/sql/relational-database/security/authentication-access/database-level-roles?view=sql-server-2017>

35

## Role bazy danych definiowane przez użytkownika

Dodawanie ról:

- Kiedy grupa ludzi potrzebuje wykonać tę samą czynność w SQL Server
- Jeżeli nie masz uprawnień do zarządzania kontami Windows

```
CREATE ROLE [Managers]
```

```
ALTER ROLE [Managers] ADD MEMBER [u1]
```

```
EXEC sp_addrole Managers, dbo  
EXEC sp_addrolemember Managers, Alicia
```

36

18

## Polecenia związane z rolami baz danych

Feature	Type	Description
<a href="#">sp_helpdbfixedrole</a>	Metadata	Returns a list of the fixed database roles.
<a href="#">sp_dbfixedrolepermission</a>	Metadata	Displays the permissions of a fixed database role.
<a href="#">sp_helprole</a>	Metadata	Returns information about the roles in the current database.
<a href="#">sp_helprolemember</a>	Metadata	Returns information about the members of a role in the current database.
<a href="#">sys.database_role_members</a>	Metadata	Returns one row for each member of each database role.
<a href="#">IS_MEMBER</a>	Metadata	Indicates whether the current user is a member of the specified Microsoft Windows group, Microsoft Entra group, or Microsoft SQL Server database role.
<a href="#">CREATE ROLE</a>	Command	Creates a new database role in the current database.
<a href="#">ALTER ROLE</a>	Command	Changes the name or membership of a database role.
<a href="#">DROP ROLE</a>	Command	Removes a role from the database.
<a href="#">sp_addrole</a>	Command	Creates a new database role in the current database.
<a href="#">sp_droprole</a>	Command	Removes a database role from the current database.
<a href="#">sp_addrolemember</a>	Command	Adds a database user, database role, Windows login, or Windows group to a database role in the current database. All platforms except Analytics Platform System (PDW) and Azure Synapse should use <a href="#">ALTER ROLE</a> instead.
<a href="#">sp_droprolemember</a>	Command	Removes a security account from a SQL Server role in the current database. All platforms except Analytics Platform System (PDW) and Azure Synapse should use <a href="#">ALTER ROLE</a> instead.
<a href="#">GRANT</a>	Permissions	Adds permission to a role.
<a href="#">DENY</a>	Permissions	Denies a permission to a role.
<a href="#">REVOKE</a>	Permissions	Removes previously granted or denied permissions.

<https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/database-level-roles?view=sql-server-2017>

37

- Przypisanie uprawnień do użytkowników i ról

Rodzaje uprawnień

Przyznawanie, zaprzeczanie,  
unieważnianie uprawnień

38

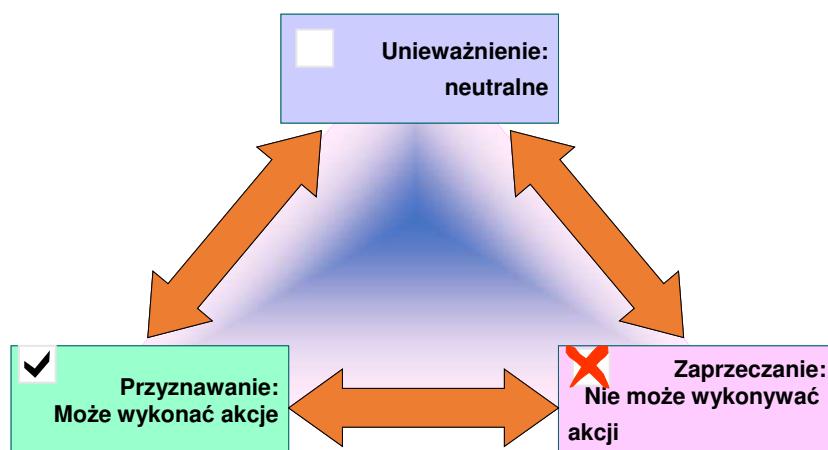
19

## Rodzaje uprawnień

Polecenie	Obiekt	Uprawnienia predefiniowane
CREATE DATABASE		
CREATE TABLE	SELECT INSERT UPDATE DELETE REFERENCES	TABLE VIEW
CREATE VIEW		
CREATE PROCEDURE		
CREATE RULE		
CREATE DEFAULT	SELECT UPDATE REFERENCES	COLUMN
BACKUP DATABASE		
BACKUP LOG	EXEC	STORED PROCEDURE

39

## Przyznawanie, zaprzeczanie, unieważnianie uprawnień



40

## Przyznawanie uprawnień umożliwiając dostęp

```
GRANT SELECT  
ON products  
TO order
```

```
GRANT INSERT, UPDATE  
ON products  
TO Eva, Ivan, David
```

User/Role	Insert	Update	Select
Eva	✓	✓	□
Ivan	✓	✓	□
David	✓	✓	□
Order	□	□	✓

41

## Zaprzeczanie uprawnień uniemożliwiając dostęp

```
DENY SELECT, INSERT, UPDATE  
ON products  
TO Eva, Ivan, David
```

User/Role	Insert	Update	Select
Eva	✗	✗	✗
Ivan	✗	✗	✗
David	✗	✗	✗
Order	□	□	✓

42

## Unieważnianie przyznanych i zaprzeczonych uprawnień

```
REVOKE SELECT, INSERT, UPDATE  
ON products  
FROM Eva, Ivan
```

User/Role	Insert	Update	Select
Eva	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ivan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
David	<b>X</b>	<b>X</b>	<b>X</b>
Order	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

43

## Planowanie bezpieczeństwa

### Ustalanie użytkowania domyślnych kont

- sysadmin [sa]
- BUILTIN\Administrators

### Ustalanie funkcji konta guest (gość)

### Ustalanie zasad uprawnień roli Public

### Zastosowanie uprawnień do ról

- stworzenie ról definiowanych przez użytkownika
- zastosowanie uprawnień do ról def. przez użytkownika
- dodanie użytkowników do zdefiniowanych ról

### Tworzenie obiektów jako właściciel - dbo

44

## Przypisanie konta logowania do kont użytkowników i ról

```
USE NORTHWIND;
GO

-- Create a Login
CREATE LOGIN [u1]
WITH PASSWORD = 'u1',
DEFAULT_DATABASE = [Northwind],
DEFAULT_LANGUAGE = [polski],
CHECK_POLICY = OFF;
GO

-- Add the user to the database using their Login name and the user name
CREATE USER u11
FOR LOGIN u1
WITH DEFAULT_SCHEMA = 'dbo';
GO

-- Add user to db_datareader role
EXEC SP_ADDROLEMEMBER 'db_datareader', 'u11';
GO

-- Add user to the database owner role
EXEC SP_ADDROLEMEMBER 'db_owner', 'u11';
GO
```

45

## Zarządzanie uprawnieniami

```
GRANT { ALL [ PRIVILEGES ] | permission [ ( column [, ...n] ) ] [, ...n] }
[ ON [ class :: ] securable ]
TO principal [, ...n]
[ WITH GRANT OPTION ]
[ AS principal ];
```

### Elementy składni poleceń GRANT, REVOKE, DENY w SQL Server

- **ALL** - Oznacza wszystkie możliwe uprawnienia dla obiektu danego typu (np. wszystkie prawa do tabeli – **bez administracyjnych**).
  - **permission** - Konkretnie uprawnienie lub lista uprawnień, które mają zostać przyznane (np. SELECT, UPDATE, EXECUTE).
  - **column** - Określa poziom kolumny – można nadać uprawnienia tylko do wybranych kolumn (dotyczy np. SELECT i UPDATE).
  - **principal** - Podmiot, któremu nadajemy uprawnienia: użytkownik bazy danych, rola bazy danych, login SQL Server lub grupa Windows.
  - **WITH GRANT OPTION** - Pozwala użytkownikowi, który otrzymał dane uprawnienie, przekazać je innym użytkownikom.
  - **AS principal** - Określa, w czym imieniu nadawane są uprawnienia (np. gdy administrator nadaje prawa jako inna rola).
- Dla instrukcji REVOKE i DENY**
- **CASCADE** - Usuwa lub blokuje również uprawnienia, które zostały nadane przez użytkownika korzystającego z opcji **WITH GRANT OPTION**.

46

## Zarządzanie uprawnieniami

```
GRANT { ALL [ PRIVILEGES ] | permission [ ( column [, ...n] ) ] [, ...n] }  
[ ON [ class :: ] securable ]  
TO principal [, ...n]  
[ WITH GRANT OPTION ]  
[ AS principal ];
```

```
GRANT ALTER, CONTROL, DELETE  
ON SCHEMA::dbo TO grupa1
```

ALL jest skrótem dla operacyjnych uprawnień typowych dla obiektów (np. SELECT, INSERT, UPDATE, DELETE dla tabeli), ale nie obejmuje uprawnień administracyjnych ani specjalnych, które mogą wpływać na bezpieczeństwo lub strukturę bazy.

**Funkcje skalarne:**

- Uprawnienia: [EXECUTE, REFERENCES](#).

**Funkcje tabelaryczne:**

- Uprawnienia: [EXECUTE, DELETE, INSERT, REFERENCES, SELECT, UPDATE](#).

**Procedury przechowywane:**

- Uprawnienia: [EXECUTE](#).

**Tabele:**

- Uprawnienia: [SELECT, INSERT, UPDATE, DELETE, REFERENCES](#).

**Widoki:**

- Uprawnienia: [SELECT, INSERT, UPDATE, DELETE, REFERENCES](#).

**Baza danych:**

- Uprawnienia: [BACKUP DATABASE, BACKUP LOG, CREATE DATABASE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE RULE, CREATE TABLE, CREATE VIEW](#).

47

## Zarządzanie uprawnieniami

```
GRANT { ALL [ PRIVILEGES ] | permission [ ( column [, ...n] ) ] [, ...n] }  
[ ON [ class :: ] securable ]  
TO principal [, ...n]  
[ WITH GRANT OPTION ]  
[ AS principal ];
```

- **ALTER** - Pozwala użytkownikowi zmieniać definicję obiektu (np. edytować kod procedury, zmieniać strukturę tabeli).
- **CONTROL** - Daje pełną kontrolę nad obiektem, w tym możliwość usuwania obiektu, zmieniając jego definicję czy nadawanie uprawnień innym użytkownikom (działa jak właściciel obiektu).
- **TAKE OWNERSHIP** - Pozwala przejąć własność obiektu – użytkownik staje się jego właścicielem.
- **VIEW DEFINITION** - Umożliwia podgląd definicji obiektu (np. kod procedury, strukturę tabeli, zapytanie widoku), bez prawa do jego modyfikacji czy wykonania.

**Procedury przechowywane, Funkcje, Tabele, Widoki:**

- Uprawnienia: [ALTER, CONTROL, TAKE OWNERSHIP, VIEW DEFINITION](#).

**Bazy danych:**

- Uprawnienia: [ALTER, CONTROL, TAKE OWNERSHIP, VIEW DEFINITION](#)

48

## Zarządzanie uprawnieniami

```
USE Northwind
```

```
GO
```

```
GRANT ALL ON [dbo].[Categories] TO [u1]
```

// bez uprawnień:  
// - ALTER,  
// - CONTROL,  
// - TAKE OWNERSHIP,  
// - VIEW DEFINITION

Permission	Grantor	Grant	With Grant	Deny
Alter	dbo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Control	dbo	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delete	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Insert	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
References	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Select	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Take ownership	dbo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Update	dbo	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View definition	dbo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

49

## Uprawnienia na poziomie serwera:

uprawnienia pozwalają na zarządzanie serwerem SQL Server, wykonywanie operacji administracyjnych oraz kontrolowanie dostępu do baz danych i innych zasobów serwera.

- CONTROL SERVER
- CREATE DATABASE
- ALTER ANY LOGIN
- ALTER ANY LINKED SERVER
- ADMINISTER DATABASE BULK OPERATIONS
- BACKUP DATABASE
- BACKUP LOG
- CREATE ENDPOINT
- CREATE ANY DATABASE
- CREATE ANY LOGIN
- CREATE ANY LINKED SERVER
- CREATE ANY SERVER ROLE
- CREATE TRACE EVENT SESSION
- SHUTDOWN
- VIEW SERVER STATE
- VIEW ANY DEFINITION
- IMPERSONATE ANY LOGIN
- CONTROL ANY LOGIN
- CREATE ASYMMETRIC KEY
- CREATE CERTIFICATE

50

## Uprawnienia

### Uprawnienia na poziomie schematu w bazie danych:

uprawnienia kontrolują dostęp do obiektów w obrębie jednej bazy danych i umożliwiają różne operacje administracyjne oraz zarządzanie danymi.

- ALTER
- CONTROL
- DELETE
- INSERT
- REFERENCES
- SELECT
- UPDATE
- VIEW DEFINITION

### Uprawnienia na poziomie obiektów w bazie danych:

uprawnienia kontrolują uprawnienia na poziomie obiektów bazy danych.

- SELECT
- INSERT
- UPDATE
- DELETE
- REFERENCES
- ALTER
- CONTROL
- EXECUTE
- TAKE OWNERSHIP
- VIEW DEFINITION

51

## Zarządzanie uprawnieniami za pomocą ról

```
-- =====
-- Create Database Role GRUPA1
-- =====
USE NORTHWIND
GO
-- Create the database role
CREATE ROLE Grupa1 AUTHORIZATION [dbo]
GO
-- Grant access rights to a specific schema in the database
GRANT ALTER, CONTROL, DELETE, EXECUTE, INSERT, REFERENCES, SELECT, TAKE OWNERSHIP, UPDATE,
VIEW DEFINITION
ON SCHEMA::dbo TO Grupa1
GO
-- Add an existing user to the role
EXEC SP_ADDROLEMEMBER 'Grupa1', 'u1'
GO
```

52

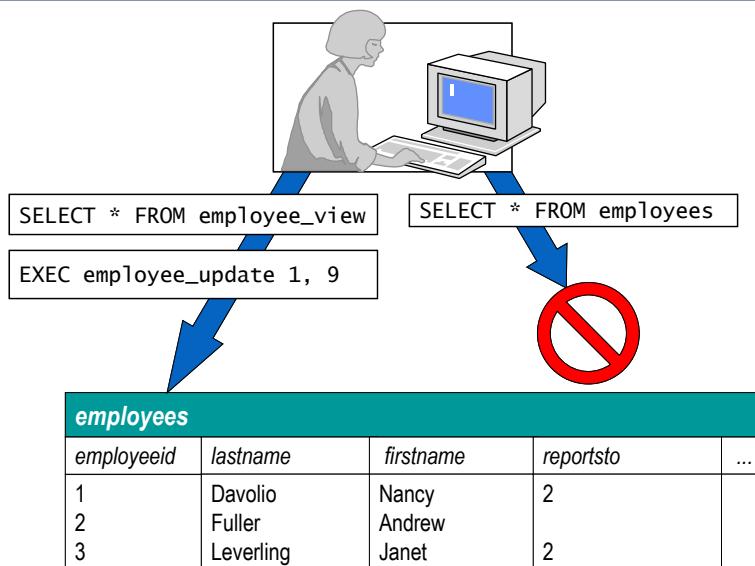
- Zarządzanie bezpieczeństwem aplikacji

Zarządzanie bezpieczeństwem przy użyciu widoków i przechowywanych procedur

Zarządzanie bezpieczeństwem przy pomocy ról aplikacyjnych

53

Zarządzanie bezpieczeństwem przy użyciu widoków i procedur przechowywanych



54

## Zarządzanie bezpieczeństwem aplikacji przy pomocy ról aplikacyjnych

Role aplikacyjne w SQL Server to specjalne role przeznaczone do organizowania dostępu do bazy danych w kontekście aplikacji.

### Główne cechy ról aplikacyjnych:

- **Dedykowane dla aplikacji** - Role aplikacyjne są tworzone z myślą o przypisywaniu uprawnień dostępu do bazy danych użytkownikom lub grupom użytkowników korzystających z aplikacji.
- **Bez przypisywania do użytkowników** - W przeciwieństwie do standardowych ról serwera lub bazy danych, role aplikacyjne nie są przypisywane bezpośrednio do użytkowników, lecz działają w kontekście sesji.
- **Ograniczanie uprawnień** - Umożliwiają bardziej precyzyjne nadawanie uprawnień, co pozwala ograniczyć dostęp do konkretnych zasobów bazodanowych na poziomie aplikacji.
- **Zmiana kontekstu** - Użytkownicy mogą przełączyć się do roli aplikacyjnej podczas sesji, co zmienia kontekst bezpieczeństwa sesji.
- **Kontrola dostępu do danych** - Role aplikacyjne mogą być używane do nadawania uprawnień tylko na poziomie bazy danych, zapewniając dodatkową warstwę kontroli dostępu do danych.

55

## Kiedy warto użyć roli aplikacyjnej?

### - Ukrycie logiki dostępu

- Użytkownicy końcowi łączą się przez wspólne konto (lub różne konta), ale aplikacja po zalogowaniu aktywuje rolę, która daje dodatkowe uprawnienia.
- Dzięki temu użytkownik nie ma bezpośredniego dostępu do wrażliwych danych w SSMS – tylko aplikacja może aktywować rolę.

### - Centralne zarządzanie uprawnieniami aplikacji

- Zamiast nadawać uprawnienia każdemu użytkownikowi, nadajesz je roli aplikacyjnej.
- Aplikacja zna hasło roli i aktywuje ją w sesji.

### - Scenariusze z wieloma użytkownikami

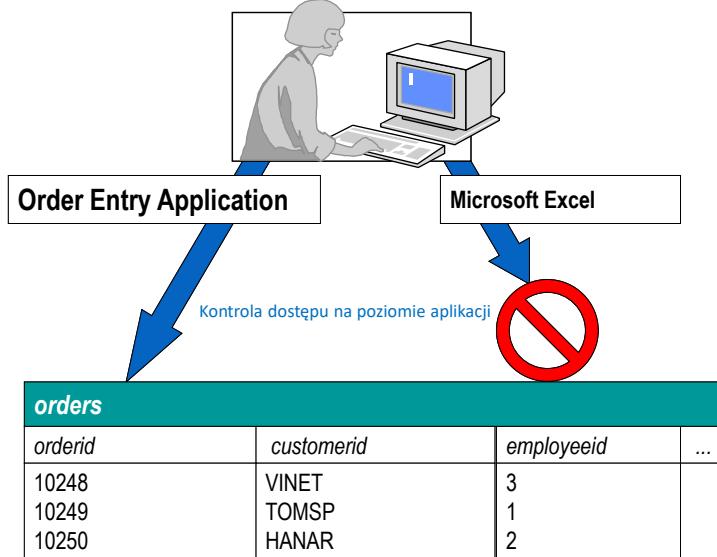
- Każdy użytkownik ma minimalne uprawnienia (np. SELECT na kilku tabelach).
- Aplikacja włącza rolę, aby uzyskać dodatkowe prawa (np. INSERT, UPDATE) tylko w kontekście działania aplikacji.

### - Bezpieczeństwo w środowisku z connection pooling

- Rola działa tylko w sesji, więc połączenie w puli nie „dziedziczy” roli, jeśli użyjesz sp\_unsetapprole.

56

## Zarządzanie bezpieczeństwem aplikacji przy pomocy ról aplikacyjnych



57

## Tworzenie roli aplikacyjnej

Wykonanie polecenia CREATE APPLICATION ROLE  
[ sp\_addapprole () – deprecated ]

Tylko członkowie db\_owner, db\_securityadmin, i sysadmin mogą ją wykonywać

```
USE YourDatabase;
GO

CREATE APPLICATION ROLE AppRoleOrders
WITH PASSWORD = 'Super_Tajne_Haslo#2025';
GO

-- Nadaj uprawnienia roli (np. SELECT na tabeli orders)
GRANT SELECT ON dbo.orders TO AppRoleOrders;
```

58

## Aktywowanie roli aplikacyjnej

Użytkownik musi wyspecyfikować hasło

Zasięgiem działania jest dana baza - jeżeli użytkownik przełączy się do innej bazy danych, użytkownik ma prawa użytkownika we wcześniejszej bazie danych

Rola nie może być dezaktywowana aż do czasu jak użytkownik się nie rozłączy

```
USE YourDatabase;
GO

DECLARE @cookie varbinary(8000); -- „żeton” do powrotu
EXEC sp_setapprole
    @rolename = N'AppRoleOrders',
    @password = N'Super_Tajne_Haslo#2025',
    @fCreateCookie = true,
    @cookie = @cookie OUTPUT;

-- Od teraz sesja działa z uprawnieniami roli aplikacyjnej
SELECT * FROM dbo.orders; -- działa, jeśli rola ma SELECT

-- Powrót do stanu sprzed roli
EXEC sp_unsetapprole @cookie = @cookie;
```

59

## Zalecane praktyki

- ✓ Używaj **Mixed Mode** tylko gdy konieczne; preferuj Windows/Entra dla reszty,
- ✓ Nie używaj konta **sa**; nadawaj rolę **sysadmin** dedykowanym loginom,
- ✓ Zapewnij, by **dbo** był właścicielem wszystkich obiektów,
- ✓ Udostępniaj dane przez **procedury składowane** i **widoki**, nie bezpośrednio tabele,
- ✓ Stosuj **zasadę najmniejszych uprawnień** i role zamiast pojedynczych GRANT'ów,
- ✓ Wyłącz/ogranicz **guest** w bazach użytkowych; nie dawaj przywilejów roli **public**,
- ✓ Unikaj **db\_owner** dla kont aplikacyjnych; twórz role „read-only” i „read-write”.

60

# Backup bazy danych

1

## Informacje podstawowe

- Zapobieganie utracie danych
- Zmiana ustawień i modele odzyskiwania bazy danych
- Kopie zapasowe w SQL Server
- Kiedy robić backup bazy danych
- Wykonywanie kopii zapasowych
- Rodzaje metod tworzenia kopii zapasowych
- Planowanie strategii kopii zapasowych

2

## Zapobieganie utracie danych

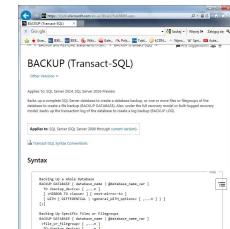
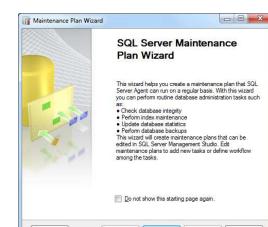
**Dane można utracić w wyniku:**

- awarii sprzętu lub oprogramowania,
  - przypadkowego lub złośliwego użycia instrukcji np. DELETE, UPDATE bez klauzuli WHERE,
  - działania niebezpiecznych wirusów lub ataków,
  - katastrofy naturalne np. pożar, powódź, trzęsienia ziemi, ataki terrorystyczne,
  - Kradzież.

3

## Zapobieganie utracie danych

- Posiadanie strategii kopii zapasowych
    - minimalizowanie utraty danych
    - odzyskiwanie utraconych danych
    - odzyskiwanie danych w minimalnym czasie
  - Regularne robienie kopii zapasowej



4

## Wstępne planowanie operacji tworzenia i przywracania kopii zapasowych

Jakiego rodzaju bazę danych rozpatrujemy?

Jak ważne są dane przechowywane w bazie danych?

Jak często aktualizowane są dane przechowywane w bazie danych?

Jak szybko wymagane jest przywrócenie danych?

Czy posiadamy odpowiedni sprzęt do wykonywania kopii zapasowych?

Czy tworzone kopie zapasowe mogą być kompresowane?

Kiedy należy wykonywać kopie zapasowe?

Czy konieczne jest przechowywanie tworzonych kopii zapasowych poza siedzibą firmy (oprogramowania)?

5

## Modele odzyskiwania bazy danych



### Prosty (Simple Recovery model)

- Przeznaczony dla baz, gdzie wystarczy odtworzenie do stanu z ostatniego pełnego lub różnicowego backupu.
- Nie obsługuje backupów dziennika transakcji – brak point-in-time recovery.
- Po każdym checkpoint system automatycznie czyści nieaktywne wpisy w dzienniku (opcja Truncate Log on Checkpoint).
- Zalecany dla baz testowych, pomocniczych lub systemowych, gdzie pełna historia transakcji nie jest wymagana.



### Pełny (Full Recovery model)

- Pełne logowanie wszystkich operacji.
- Obsługuje backup dziennika transakcji → możliwe point-in-time recovery.
- Wymaga regularnych backupów logu, inaczej dziennik rośnie.
- Zalecany dla baz krytycznych, gdzie wymagana jest pełna historia transakcji.



### Z rejestracją masową (Bulk-Logged Recovery model)

- Minimalne logowanie dla operacji masowych (np. BULK INSERT, SELECT INTO).
- Obsługuje backup dziennika, ale brak point-in-time recovery w trakcie operacji masowych.
- Backup logu po operacjach masowych jest większy.
- Dobry dla dużych operacji masowych, gdy wydajność jest priorytetem.

6

3

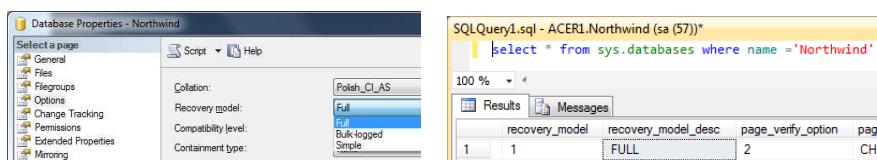
## Modele odzyskiwania bazy danych z rejestrowaniem masowym

- Bulk-Logged Recovery Model minimalizuje rejestrowanie w dzienniku transakcji dla operacji masowych (bulk operations), aby poprawić wydajność.
- Operacje takie jak:
  - SELECT INTO
  - BULK INSERT
  - Bcp (polecenie)
  - WRITETEXT (deprecated)
  - UPDATETEXT (deprecated)są rejestrowane w sposób minimalny, a nie w pełni (instrukcje nie są całkowicie pomijane w dzienniku – są rejestrowane w sposób minimalny (np. tylko alokacje stron, a nie pełne dane)).
- Po wykonaniu takich operacji nie można odtworzyć bazy do punktu w czasie w trakcie operacji masowej (point-in-time recovery), ponieważ dziennik nie zawiera pełnych danych.
- Po zakończeniu operacji masowych wykonaj pełny backup bazy danych.

```
BULK INSERT Pracownicy  
FROM 'C:\Import\dane.csv'  
WITH (  
FIELDTERMINATOR = ',',  
ROWTERMINATOR = '\n',  
FIRSTROW = 1  
)
```

7

## Zmiana modelu odzyskiwania bazy danych



### Składnia:

```
ALTER DATABASE database_name  
SET RECOVERY [FULL | SIMPLE | BULK_LOGGED]
```

### Przykład:

```
ALTER DATABASE [Northwind] SET RECOVERY FULL WITH NO_WAIT  
-- NO_WAIT | ROLLBACK AFTER 5 SECONDS | ROLLBACK IMMEDIATE
```

```
select * from sys.databases where name ='Northwind'
```

8

## Backup SQL Server

- Umożliwia robienie kopii zapasowej podczas gdy użytkownicy kontynuują pracę ze swoją bazą danych,
- Robi kopię zapasową na wskazanych nośnikach i zapisuje ich lokalizację,
- Wychwytuje działania na bazie danych, które zdarzają się podczas procesu Backupu
  - wywołuje punkt kontrolny i rejestruje LSN (log sequence number) najstarszego aktywnego rekordu dziennika trans.
  - zapisuje wszystkie strony na medium wykorzystywanym do robienia kopii zapasowej
  - zapisuje wszystkie transaction log zarejestrowane podczas procesu tworzenia kopii zapasowej

9

## Wykonywanie i składowanie kopii zapasowych



### Kto wykonyuje kopie zapasowe

- członkowie roli serwera **sysadmin**
- członkowie roli baz danych **db\_owner**,  
**db\_backupoperator**

### Gdzie przechowywać kopie zapasowe

- Dyski (systemy archiwizujące oparte na dyskach)
- Napędy taśmowe (z modułem automatycznego zmieniania kaset)

10

## Wybór urządzeń i nośników dla kopii zapasowych

Wybór właściwego rozwiązania dla potrzeb archiwizacji danych uzależniony jest od wielu czynników obejmujących między innymi:

- Pojemność
- Niezawodność
- Rozszerzalność
- Szybkość
- Koszt

Liczba potrzebnych taśm lub dysków uzależniona jest od:

- Ilości archiwizowanych danych
- Zakładanej częstotliwości wykonywania kopii zapasowych
- Wymaganego okresu przechowywania dodatkowych zestawów danych

11

## Kiedy robić kopie zapasowe baz danych

Tworzenie kopii zapasowych systemowych baz danych

Tworzenie kopii zapasowych baz danych użytkowników

Czynności, które są zabronione podczas robienia kopii zapasowej

12

## Backup systemowych baz danych

### Po zmodyfikowaniu bazy danych MASTER (SIMPLE)

- Gdy używamy polecień CREATE DATABASE, ALTER DATABASE lub DROP DATABASE, CREATE LOGIN ...
- Wykonujemy systemowe procedury przechowywane do zmiany konfiguracji serwera (warto zachować kilka kopii)
- Pewne zmiany realizowane są automatycznie (np. zmiana rozmiaru bazy danych)

### Po zmodyfikowaniu bazy danych MSDB (SIMPLE)

- Zmiana harmonogramów zadań przez SQL Server Agent
- Dodawanie zadań, alertów czy operatorów
- Przechowuje historię kopii zapasowych

### Po zmodyfikowaniu bazy danych MODEL (default FULL)

13

## Backup baz danych użytkownika



Po utworzeniu baz danych



Po utworzeniu indeksów

dziennik transakcji rejestruje tylko fakt, że indeks został utworzony, nie rzeczywiście zmiany stron danych. Czas, który jest wymagany, aby odbudować indeks może być dłuższy niż czas potrzebny do przywrócenia pełnej kopii zapasowej.



Po wyczyszczeniu dziennika transakcji

polecenia BACKUP LOG WITH TRUNCATE\_ONLY  
lub BACKUP LOG WITH NO LOG (deprecated)



Po wykonaniu nierejestrowanych operacji

polecenie SELECT...!INTO, bulk copy (bcp)  
polecenie WRITETEXT lub UPDATETEXT (deprecated)

14

## Czynności zabronione podczas tworzenia kopii zapasowej bazy danych

- Tworzenie i modyfikowanie bazy danych
- Wykonywanie operacji zmiany wielkości bazy danych
- Tworzenie indeksów
- Wykonanie nierejestrowanych operacji
- Zmniejszanie rozmiaru bazy danych

15

## Wykonywanie kopii zapasowych

Tworzenie urządzenia do tworzenia kopii zapasowej

Tworzenie trwałych plików kopii zapasowych

Tworzenie tymczasowych plików kopii zapasowych

Użycie wielokrotnych plików kopii zapasowych do przechowywania backupu

Użycie polecenia BACKUP

Backup na urządzeniu taśmowym

16

## Tworzenie urządzenia kopii zapasowych



Dlaczego tworzyć urządzenia do kopii zapasowych

celem użycia w przyszłości w procesie tworzenia kopii zapasowych  
aby zautomatyzować zadanie tworzenia kopii zapasowych



Użycie systemowej przechowywanej procedury sp\_addumpdevice

określa logiczną nazwę urządzenia logiczne i fizyczne nazwy są przechowywane w systemowej tablicy sysdevices

```
USE [master]
EXEC sp_addumpdevice @devtype ='disk',
@logicalname = 'backup_file',
@physicalname = 'c:\backup\backup_file.bak';
BACKUP DATABASE [Northwind] TO [backup_file];
```

17

## Tworzenie tymczasowych plików kopii zapasowych

### Dlaczego tworzy się takie pliki

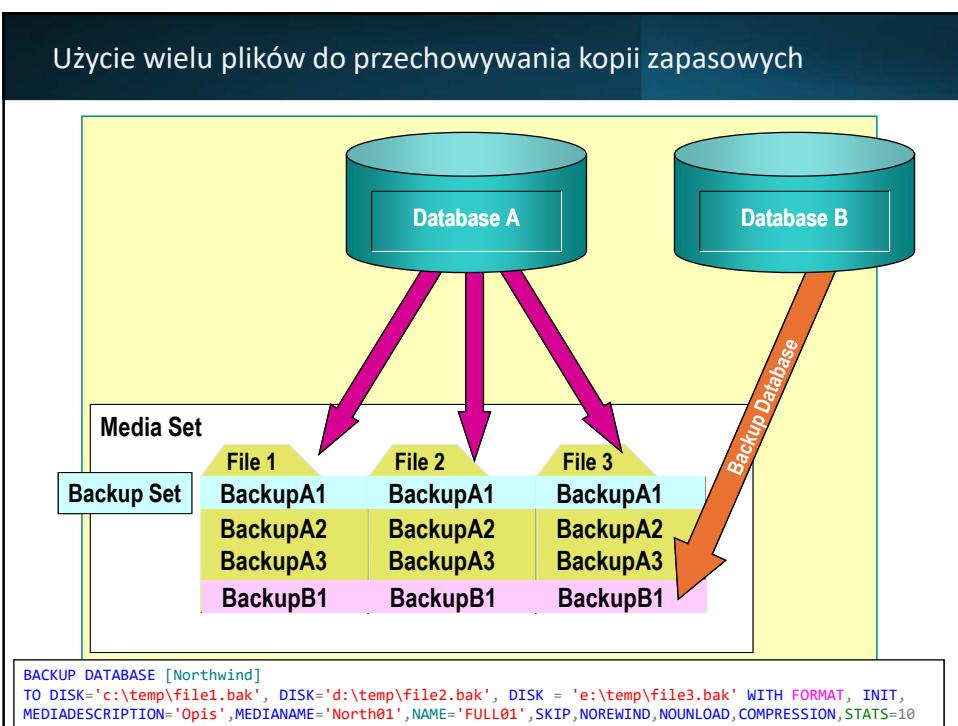
- Aby wykonać pojedynczy proces archiwizacji danych
- Aby sprawdzić operacje archiwizacji danych, którą zamierza się zautomatyzować

### Użycie wyrażenia BACKUP DATABASE

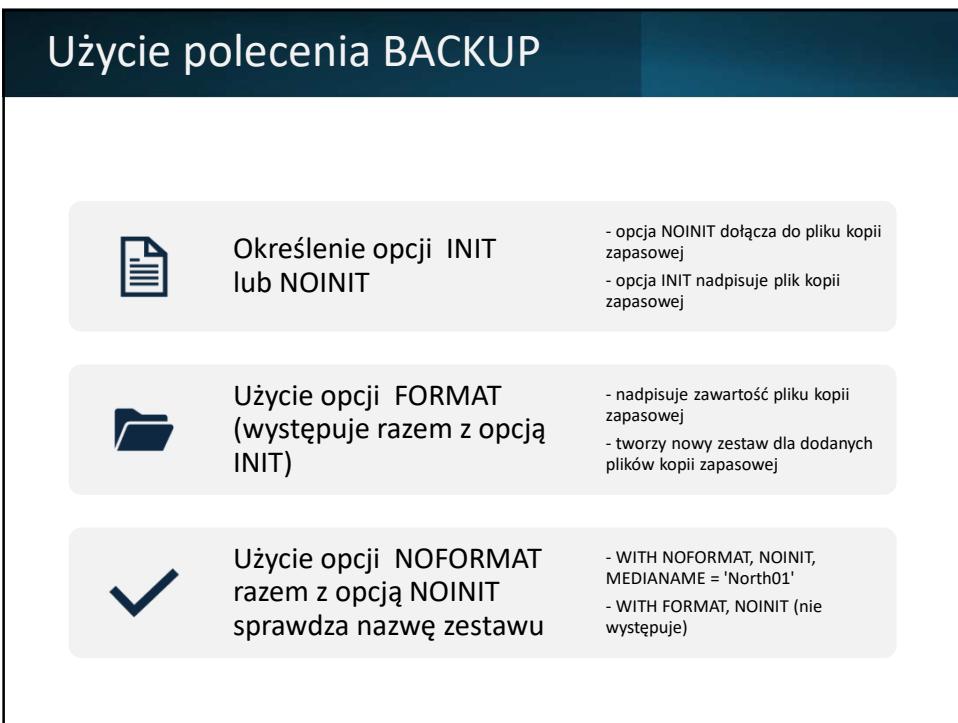
- określamy typ medium (urządzenie dyskowe czy taśmowe)
- określamy dokładną ścieżkę i nazwę pliku

```
USE [master]
BACKUP DATABASE northwind
TO DISK = 'c:\backup\temp_backup_file.bak'
```

18



19



20

## Backup na urządzeniu taśmowym

Wymaga urządzenia taśmowego lokalnie dołączonego do SQL Server

Nagrywa informacje o kopii zapasowej na etykiecie taśmy

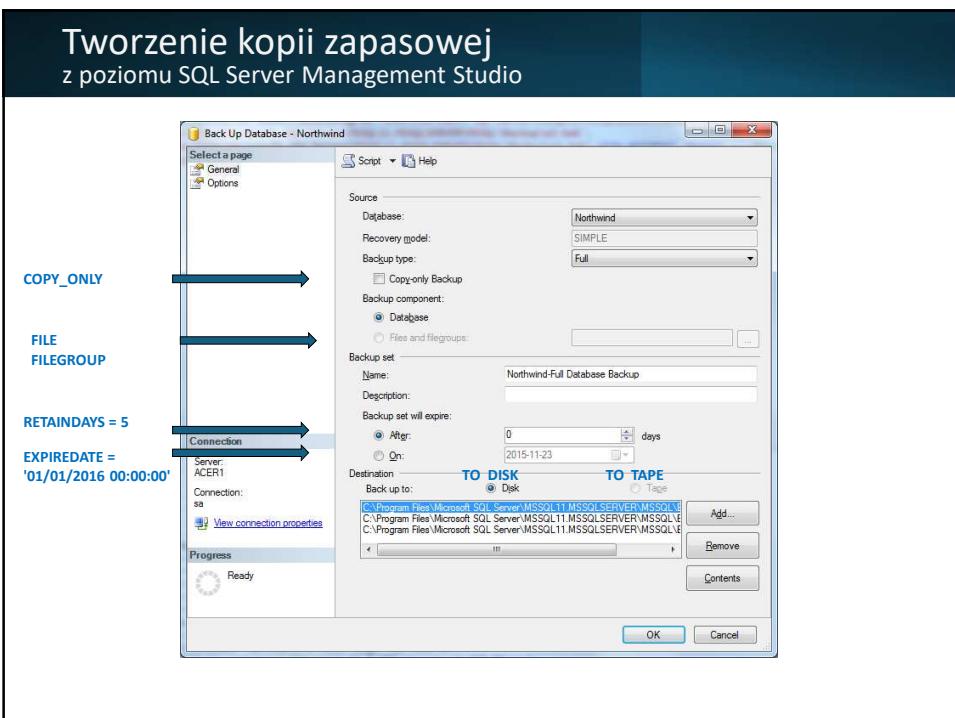
przechowuje kopie zapasowe MS SQL Server i innych danych

21

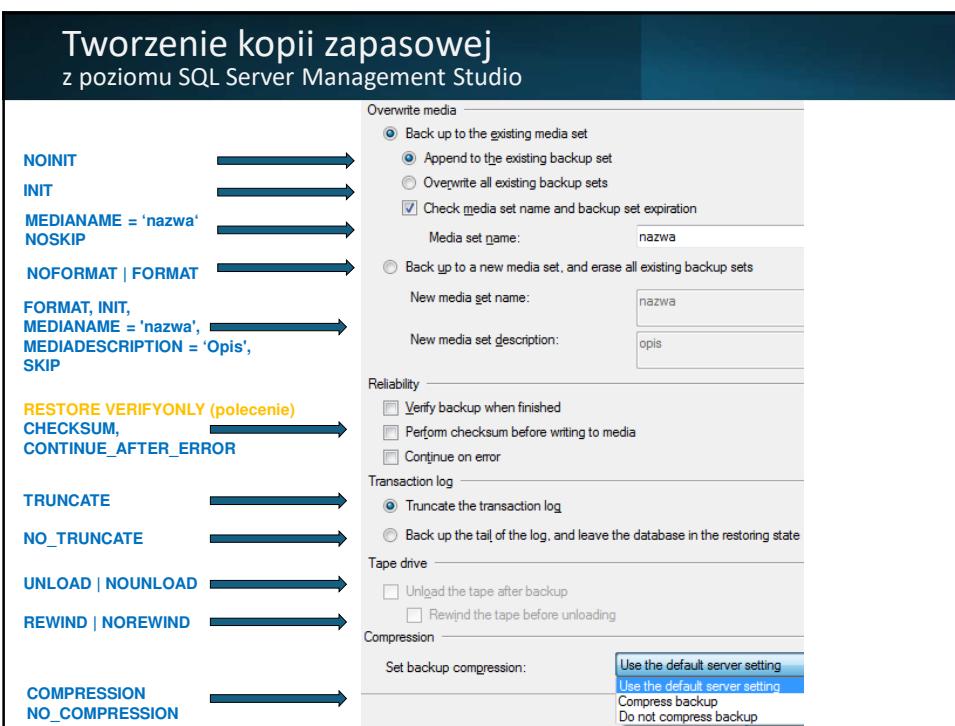
## Określanie opcji dla taśmy

Opcja taśmy	Opis
UNLOAD (default)	Przewija i rozładowywuje taśmę
NOUNLOAD	Nie przewija i rozładowywuje taśmę
BLOCKSIZE	Zmienia fizyczny rozmiar bloku w bajtach
FORMAT	Nadpisuje nagłówki w plikach, które są używane do Backup'u
SKIP	Ignoruje etykiety taśm ANSI
NOSKIP (default)	Czyta etykiety taśm ANSI
RESTART	Restartuje proces kopii zapasowej od momentu jej przerwania (realizowany dla wielu taśm)

22



23



24

## Typy metod tworzenia kopii zapasowych

- **Full Backup** – kopia całej bazy.
- **Differential Backup** – zmiany od ostatniego pełnego backupu.
- **Transaction Log Backup** – zapis transakcji od ostatniego backupu logu.
- **File/Filegroup Backup** – dla dużych baz.
- **Copy-Only Backup** – specjalny backup bez zmiany standardowego łańcucha backupów (backup chain).
- **Tail-Log Backup** – przed restore w przypadku awarii.

25

## Pełna kopia zapasowa bazy danych

- Dostarcza punkt bazowy
- Tworzy wszystkie obiekty, tabele systemowe i dane
- Tworzy część Transaction Log używanego w trakcie tworzenia kopii zapasowej

```
USE master
EXEC sp_addumpdevice 'disk', 'disk_c',
'C:\backup\dysk_c.bak'
BACKUP DATABASE northwind TO disk_c
```



Pełna kopia zapasowa pozwala na kompletne odtworzenie stanu wszystkich danych znajdujących się w bazie w chwili zakończenia operacji tworzenia kopii zapasowej.

26

## Pełna kopia zapasowa bazy danych

**Przykład 1.** tworzy urządzenie kopii zapasowej o nazwie logicznej 'disk\_c' i wykonuje pełną kopię zapasową bazy danych.

```
USE master  
EXEC sp_addumpdevice 'disk', 'disk_c','c:\backup\disk_c.bak';  
BACKUP DATABASE Northwind TO disk_c;
```

**Przykład 2.** wykonuje pełną kopię zapasową bazy danych do pliku na urządzeniu disk\_c i zastępuje poprzednie kopie zapasowe.

```
BACKUP DATABASE Northwind TO disk_c WITH INIT
```

**Przykład 3.** dołącza pełną kopię zapasową bazy danych do pliku na urządzeniu 'disk\_c'. Wszystkie wcześniejsze pliki kopii zapasowych pozostają nienaruszone.

```
BACKUP DATABASE Northwind TO disk_c WITH NOINIT
```

**Przykład 4.** tworzy kopię zapasową pliku na dysku i wykonuje pełną kopię zapasową bazy danych do tego pliku.

```
BACKUP DATABASE Northwind TO DISK='c:\backup\disktemp_c.bak'
```

27

## Różnicowa kopia zapasowa bazy danych

- Używana do często modyfikowanych baz danych
- Wymaga pełnego backupu bazy danych
- Tworzy kopie zapasową zmian w bazie danych od czasu jej ostatniej pełnej kopii zapasowej
- Oszczędza czas zarówno w samym procesie backupu jak i procesie odtwarzania
- Tworzy część Transaction Log używanego w trakcie tworzenia kopii zapasowej

```
BACKUP DATABASE northwind TO  
DISK = 'c:\backup\disktemp_c.bak'  
WITH DIFFERENTIAL
```

Pełna i różnicowa kopia zapasowa pozwala na kompletne odtworzenie stanu wszystkich danych znajdujących się w bazie w chwili zakończenia operacji tworzenia różnicowej kopii zapasowej.

28

## Kopia zapasowa dziennika transakcji

- Wymaga pełnego backup-u bazy danych
- Tworzy wszystkie zmiany bazy danych od ostatniego polecenia BACKUP LOG do końca obecnego dziennika transakcji
- Przycina dziennik transakcji

```
USE master
EXEC sp_addumpdevice 'disk', 'disk_d',
'd:\backup\disk_d.bak'
BACKUP LOG northwind TO disk_d
```

Kopia zapasowa dziennika transakcji zapisuje stan istniejący w chwili operacji tworzenia (rozpoczęcia) kopii zapasowej.

29

## Użycie opcji NO\_TRUNCATE

- Tworzy kopię zapasową dziennika transakcji ([Transaction Log](#)) dla bazy danych.
- Może być wykonana nawet gdy baza jest w stanie SUSPECT lub częściowo offline, czyli w sytuacji awaryjnej, gdy normalny backup logu nie jest możliwy.

```
-- Wymaga uprawnień sysadmin
ALTER DATABASE Test SET EMERGENCY;      -- tylko do odczytu, pomija zabezpieczenia
ALTER DATABASE Test SET SINGLE_USER;     -- przygotowanie do naprawy

-- Diagnostyka/naprawa (tylko lab! REPAIR_ALLOW_DATA_LOSS może powodować utratę danych)
-- DBCC CHECKDB (Test) WITH NO_INFOMSGS;
DBCC CHECKDB (Test, REPAIR_ALLOW_DATA_LOSS);

ALTER DATABASE Test SET MULTI_USER;
-- Powrót z EMERGENCY:
ALTER DATABASE Test SET ONLINE;    -- jeżeli wcześniej OFFLINE
```

- Zachowuje wszystkie wpisy w dzienniku transakcji – nie czyści logu po wykonaniu backupu.
- Umożliwia odzyskanie danych do momentu awarii ([Point-in-Time Recovery](#)), jeśli mamy pełny backup + wszystkie logi.
- [Wymaga modelu odzyskiwania Full lub Bulk-Logged](#) – w trybie Simple backup logu nie działa.
- [Nie przerwa łańcucha logów](#) – backup jest częścią chain, więc można go użyć w sekwencji restore.
- Opcja NO\_TRUNCATE jest typowa dla sytuacji awaryjnych, np. gdy baza jest uszkodzona, ale chcemy uratować transakcje przed naprawą.

```
BACKUP LOG Northwind TO DISK = 'C:\Backup\Northwind_Log.bak' WITH NO_TRUNCATE
```

30

## Czyszczenie dziennika transakcji

- Użycie opcji TRUNCATE\_ONLY lub NO\_LOG (*deprecated*)
  - wykonywać przed wykonaniem pełnego backupu bazy danych
  - nie można odzyskać zmian
  - nie jest nagrywany

--Truncating the transaction log  
**BACKUP LOG NORTHWIND WITH NO\_LOG**  
-- **WITH TRUNCATE\_ONLY**

- Ustawienie opcji ‘trunc. log na chkpt’ (model simple)
  - zapisuje wszystkie zatwierdzone transakcje
  - wykonuje się automatycznie po ustawieniu na true

31

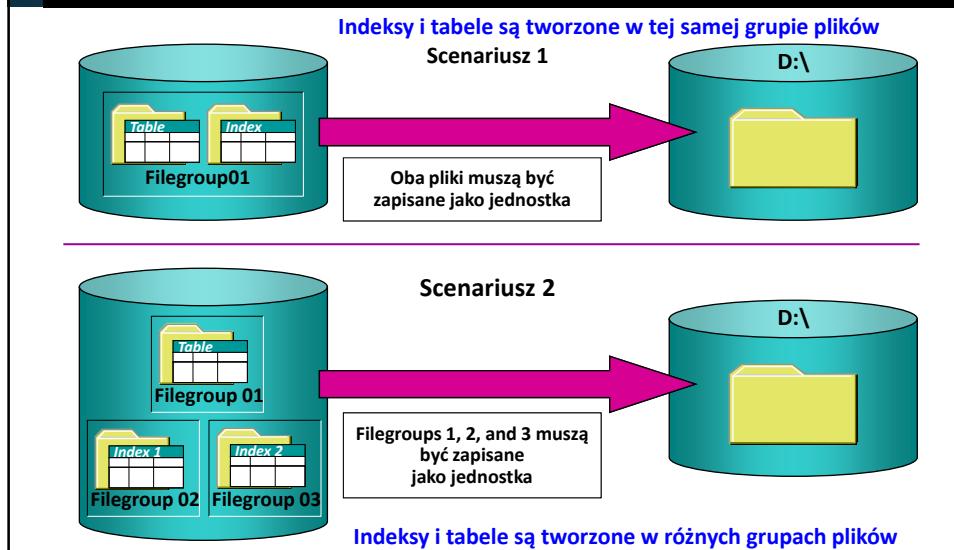
## Kopia zapasowa pliku bazy danych lub grupy plików

- Używane przy bardzo dużych bazach danych
- Tworzy kopię plików bazy danych oddzielnie
- Zapewnia, że wszystkie pliki bazy danych znajdujące się w grupie będą kopiowane
- Należy utworzyć kopię dziennika transakcji w celu zgodności danych

```
BACKUP DATABASE Northwind
FILE = 'northwind5' TO northwind5 -- FILE|FILEGROUP
BACKUP LOG Northwind to Northlog
```

32

## Ograniczenia - kopia zapasowa pliku lub grupy plików



33

## Planowanie strategii kopii zapasowej

Strategia pełnej kopii zapasowej bazy danych (Full)

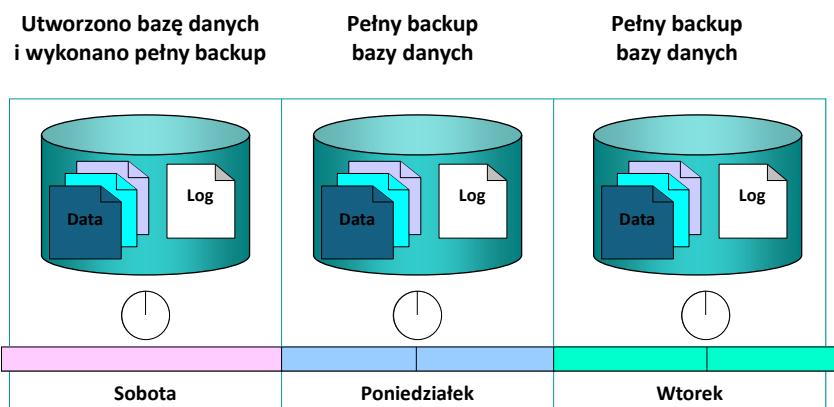
Strategia pełnej kopii zapasowej bazy danych i kopii zapasowej dziennika transakcji (Transaction Log)

Strategia różnicowej kopii zapasowej bazy danych (Differential)

Strategia kopii zapasowej plików bazy danych lub grupy plików

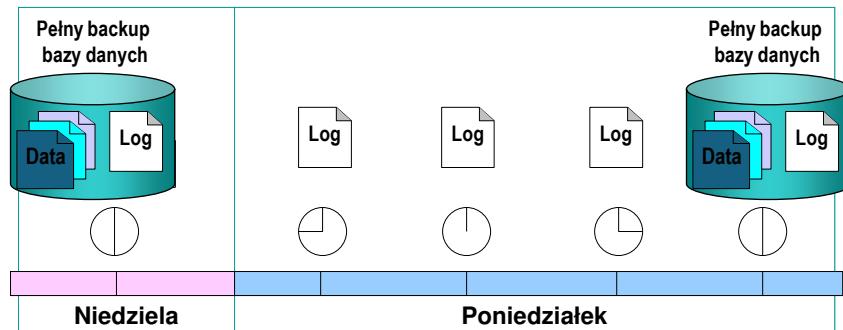
34

## Strategia pełnej kopii zapasowej bazy danych



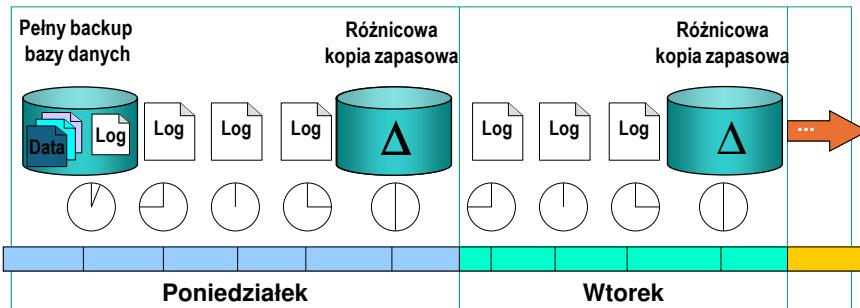
35

## Strategia pełnej kopii zapasowej bazy danych i kopii zapasowej dziennika transakcji



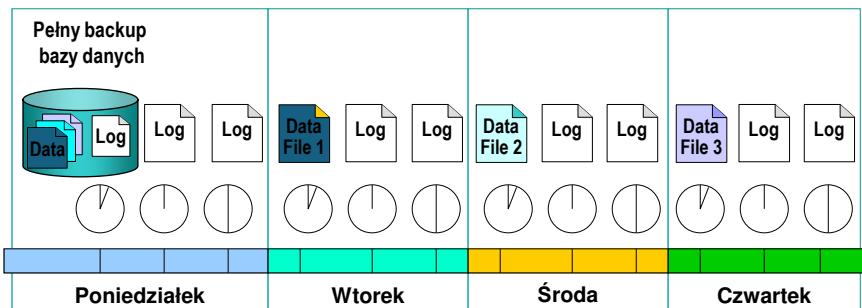
36

## Strategia różnicowej kopii zapasowej baz danych



37

## Strategia kopii zapasowej plików bazy danych lub grupy plików

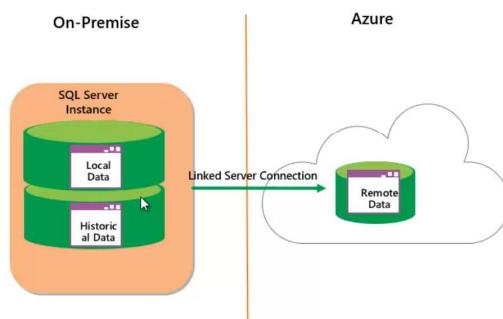


38

## Stretch Database (deprecated)

### Stretch Database

- What is Stretch Database?
  - Historical data – cloud storage
  - Current data – on-premise
- Benefits
  - Quicker backups
  - Reduced on-premise storage requirements
  - Seamless to users



39

### Rozważania

- Twórz kopię zapasową na wielu fizycznych urządzeniach
- Typ urządzenia wpływa na szybkość backupu
- Minimalizuj inne operacje SQL Server podczas backupu
- Używaj kompresji backupu
- Rozważ szyfrowanie kopii zapasowych
- Testuj odtwarzanie kopii zapasowych regularnie
- Planuj harmonogram backupów w oknach serwisowych
- Przechowuj kopie w różnych lokalizacjach (on-site i off-site)
- Monitoruj czas trwania i integralność backupów
- Automatyzuj proces backupu (SQL Agent, Maintenance Plans)

40

## Polecane praktyki



Posiadaj i przetestuj strategie tworzenia kopii zapasowej



Twórz kopię zapasową systemowej bazy danych zaraz po tym jak była modyfikowana



Rozplanuj operacje Backupu, kiedy aktywność bazy danych jest mniejsza



Twórz trwałe pliki kopii zapasowej



Wykonaj kopie zapasową plików bazy danych lub grupy plików, aby zaoszczędzić czas

41

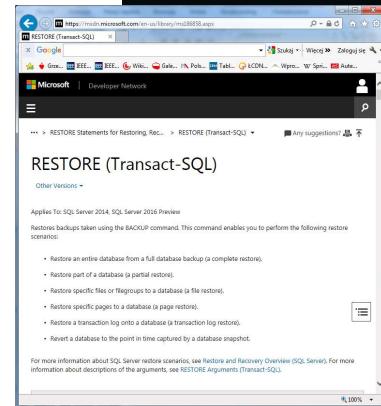
## Przywracanie bazy danych

42

21

## Informacje podstawowe

- Proces SQL Server Recovery
- Przygotowania do odtworzenia bazy danych
- Odtworzenie kopii zapasowej bazy danych
- Odtworzenie bazy z różnych typów kopii zapasowych
- Odtworzenie zepsutej systemowej bazy danych



43

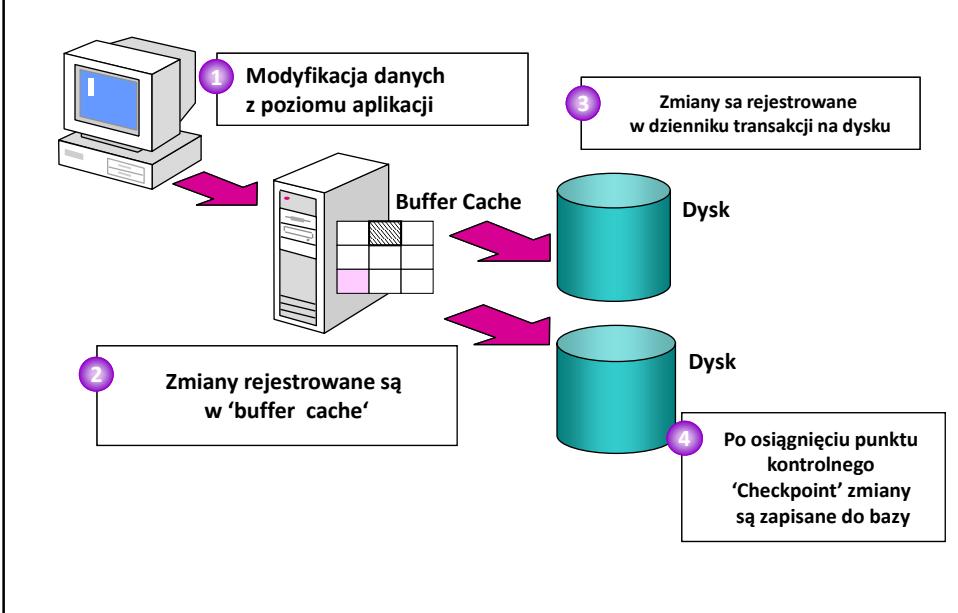
## Proces SQL Server Recovery

mechanizm przywracania spójności bazy po awarii



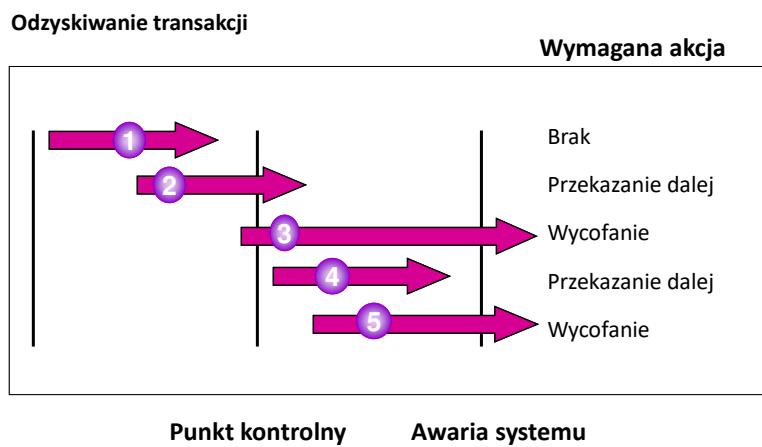
44

## Jak pracuje dziennik transakcji



45

## Odzyskiwanie transakcji i punkty kontrolne



46

## SQL Server - proces odtwarzania

### Recovery wewnętrzny - po starcie instancji

- Analysis → Redo → Undo ( kolejne etapy pracy z dziennikiem transakcyjnym po restarcie bazy).

### Przygotowania do odtworzenia bazy danych

- Weryfikacja kopii (RESTORE HEADERONLY, RESTORE FILELISTONLY)
- Ścieżki i uprawnienia do plików .bak / .trn
- Kolejność: pełna → różnicowa → logi
- Ustalenie trybu: **WITH RECOVERY** vs **WITHOUT RECOVERY**
- Opcje **MOVE/REPLACE** dla nowych lokalizacji plików
- (Opcjonalnie) DBCC CHECKDB, VERIFYONLY, checksums

### Scenariusze odtworzenia bazy danych (RESTORE)

- Pełne przywrócenie (single shot)
- Łańcuch: pełna + różnicowa + wiele logów (NORECOVERY, aż do ostatniego kroku)
- Point-in-time z WITH STOPAT
- File/Filegroup restore
- Bazy systemowe (master/msdb/model) – single-user + WITH REPLACE
- Tail-log backup (gdy baza wciąż działa), następnie końcowe RESTORE ... WITH RECOVERY

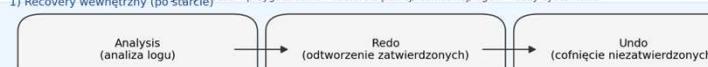
Na koniec: DBCC CHECKDB (sprawdzenie integralności bazy danych), przełączenie aplikacji do aktualnej bazy, odbudowa indeksów/statystyk

```
USE Northwind
ALTER INDEX ALL ON [Products] REBUILD;
UPDATE STATISTICS [Products];
```

47

## SQL Server - proces odtwarzania

Przegląd procesu • przygotowania • restore z pełnej/różnicowej/logów • bazy systemowe



2) Przygotowania do odtworzenia

Zweryfikuj kopie:  
RESTORE HEADERONLY  
RESTORE FILELISTONLY

Ścieżki/Uprawnienia  
Dostęp do plików .bak/.trn

Sekwencja backupów  
pełna → różnicowa → logi

Tryb docelowy:  
RECOVERY vs NORECOVERY

Opcje MOVE/REPLACE  
nowe lokalizacje plików

Opcjonalnie: CHECKDB,  
VERIFYONLY, checksums

3) Scenariusze odtworzeń (RESTORE)

A) Pełne przywrócenie (single file):  
RESTORE DATABASE ... WITH RECOVERY

B) Łańcuch: pełna + różnicowa + logi:  
FULL (NORECOVERY) → DIFF (NORECOVERY) → LOGI (RECOVERY)

C) Point-in-time (log):  
WITH STOPAT = YYYY-MM-DD hh:mm:ss

D) File/Filegroup/Piecemal restore  
(duże bazy, etapowe udostępnianie)

E) Bazy systemowe (master/msdb/model):  
tryb single-user → RESTORE ... WITH REPLACE

F) Tail-log backup (jeśli baza działa):  
BACKUP LOG ... WITH NO\_TRUNCATE / NORECOVERY

Kontrola końcowa: DBCC CHECKDB,  
przełączenie aplikacji, indeksy/STATISTICS

Legenda:  
RECOVERY - kończ odtwarzanie  
NORECOVERY - przygotuj do kolejnych RESTORE

48

## Czynności SQL Server podczas procesu odtwarzania – status bazy

- Wykonanie bezpiecznego sprawdzenia
  - Czy baza już istnieje
  - Czy pliki bazy są różne od zawartych w backupie
  - Czy pliki bazy są niekompletne
- Odtworzenie bazy i wszystkich powiązanych plików

```
Sprawdzenie trybu i stanu pracy bazy danych  
(stare podejście):  
select databaseproperty('master','IsShutdown')  
select databaseproperty('master','IsEmergencyMode')  
select databaseproperty('master','IsSuspect')  
select databaseproperty('master','IsInLoad')  
select databaseproperty('master','IsInRecovery')
```

```
SELECT  
    DB_NAME() AS current_db,  
    DATABASEPROPERTYEX('master', 'Status') AS Status,  
    -- ONLINE | RESTORING | RECOVERING | RECOVERY_PENDING | SUSPECT | EMERGENCY | OFFLINE  
    DATABASEPROPERTYEX('master', 'Updateability') AS Updateability,  
    -- READ_WRITE | READ_ONLY  
    DATABASEPROPERTYEX('master', 'IsInStandby') AS IsInStandby;  
    -- 1/0 (log shipping standby)
```

current_db	Status	Updateability	IsInStandby
master	ONLINE	READ_WRITE	0

```
SELECT  
    name,  
    state_desc,  
    -- ONLINE | RESTORING | RECOVERING | RECOVERY_PENDING | SUSPECT | EMERGENCY | OFFLINE  
    user_access_desc,  
    -- MULTI_USER | RESTRICTED_USER | SINGLE_USER  
    is_read_only,  
    is_in_standby, -- odpowiada "standby" (log shipping)  
    is_cleanly_shutdown -- odpowiednik "IsShutdown": 1 = zamknięta czysto  
FROM sys.databases  
WHERE name = 'master'; -- usuń WHERE, aby zobaczyć wszystkie bazy
```

name	state_desc	user_access_desc	is_read_only	is_in_standby	is_cleanly_shutdown
master	ONLINE	MULTI_USER	0	0	0

49

## SPRAWDZANIE INTEGRALNOŚCI BAZY DANYCH

Database Console Commands – zestaw poleceń diagnostycznych i konserwacyjnych, które działają bezpośrednio na bazach danych. Sprawdza różne aspekty integralności bazy danych.

- DBCC CHECKDB ('*database\_name*' | *database\_id* / 0, NOINDEX | {REPAIR\_ALLOW\_DATA\_LOSS | REPAIR\_FAST | REPAIR\_REBUILD} ) WITH { ALL\_ERRORMSG, NO\_INFOMSGS, TABLOCK, ESTIMATEONLY , PHYSICAL\_ONLY | DATA\_PURITY }

```
DBCC CHECKDB (Northwind, REPAIR_ALLOW_DATA_LOSS)
```

- DBCC CHECKCATALOG – sprawdza integralność referencyjną tabel systemowych
- DBCC CHECKALLOC
- DBCC CHECKTABLE

- Aby skorzystać z polecenia DBCC trzeba należeć do roli SYSADMIN lub roli DB\_OWNER

50

## DBCC - Database Console Commands

Polecenie	Opis	Zastosowanie
DBCC CHECKDB	Sprawdza integralność bazy danych (struktury, indeksy, alokacje).	Regularne sprawdzanie spójności bazy, diagnostyka po awarii lub restore.
DBCC CHECKTABLE	Sprawdza integralność pojedynczej tabeli.	Gdy podejrzewasz uszkodzenie konkretnej tabeli lub po operacjach naprawczych.
DBCC CHECKALLOC	Sprawdza alokację stron w bazie danych.	Kontrola poprawności alokacji po problemach z dyskiem lub błędach I/O.
DBCC FREEPROCCACHE	Czyści pamięć podręczną planów zapytań.	Po zmianach w strukturze bazy lub konfiguracji, aby wymusić nowe plany zapytań.
DBCC SHRINKDATABASE	Zmniejsza rozmiar bazy danych.	Po dużych operacjach usuwania danych, aby odzyskać miejsce na dysku.
DBCC TRACEON	Włącza określony znacznik śledzenia (trace flag).	Do aktywacji specjalnych opcji diagnostycznych lub optymalizacyjnych.

```
DBCC CHECKDB('Northwind') WITH NO_INFOMSGS, ALL_ERRORMSG;
DBCC CHECKTABLE('products') WITH NO_INFOMSGS;
DBCC CHECKALLOC('Northwind');
DBCC FREEPROCCACHE;
DBCC SHRINKDATABASE('Northwind', 10);
DBCC TRACEON(3604);
```

51

## Przygotowania do odtworzenia bazy

Weryfikacja kopii zapasowych

Wykonanie określonych zadań przed odtworzeniem kopii zapasowych

52

## Weryfikowanie Backupów

Wyrażenie  
RESTORE HEADERONLY

- Zwraca informację o pliku backupu

```
RESTORE HEADERONLY FROM DISK = 'C:\xxx.bak'  
RESTORE HEADERONLY FROM dysk_c
```

Wyrażenie  
RESTORE FILELISTONLY

- Zwraca informacje o bazie danych i jej plikach

```
RESTORE FILELISTONLY FROM dysk_c  
RESTORE FILELISTONLY FROM DISK = 'C:\xxx.bak'
```

Wyrażenie  
RESTORE LABELONLY

- Zwraca informacje o medium backup-u

```
RESTORE LABELONLY FROM dysk_c  
RESTORE LABELONLY FROM DISK = 'C:\xxx.bak'
```

Wyrażenie  
RESTORE VERIFYONLY

- Weryfikuje czy indywidualne pliki są kompletne i możliwe do odczytania

```
RESTORE VERIFYONLY FROM dysk_c  
RESTORE VERIFYONLY FROM DISK = 'C:\xxx.bak'
```

53

## Wykonanie określonych zadań przed odtworzeniem backupu

Ustawienie opcji **dbo use only** dla danej bazy

- Musi być ustawiona przez członka roli **sysadmin** lub **db\_owner**
- Ogranicza dostęp do bazy

Backup Transaction Log

- Zapewnia spójność bazy danych
- Przechowuje zmiany pomiędzy ostatnim backupem dziennika transakcji i kiedy baza danych działała w trybie offline

54

## Odtwarzanie kopii zapasowej

Użycie polecenia  
RESTORE

Inicjowanie  
procesu  
odtwarzania

Określenie opcji  
odtwarzania

55

## Użycie polecenia RESTORE

Użycie opcji  
odtworzenia

```
USE master  
RESTORE DATABASE northwind  
FROM dysk_c
```

Odtworzenie  
uszkodzonej bazy  
użytkownika

- SQL Server automatycznie odtwarza pliki i obiekty bazy danych
- Nie trzeba usuwać uszkodzonej bazy.

56

## RECOVERY i NORECOVERY

– różnice w odtwarzaniu bazy

### Opcja RECOVERY

- Kończy proces odtwarzania, wycofuje niezatwierdzone transakcje, przywraca spójność.
- Po tym kroku **nie można** zastosować kolejnych backupów (logów/różnicowych).
- Domyślna opcja, jeśli nie podano innej.
  - używaj przy **ostatnim backupie**,
  - umożliwia dostęp do bazy.

### Opcja NORECOVERY

- Nie wycofuje transakcji, baza **nie jest spójna** i pozostaje w trybie RESTORING.
- Używaj przy **wszystkich wcześniejszych backupach** (pełne, różnicowe, logi).
- Brak dostępu do bazy do czasu RECOVERY.

[RESTORE DATABASE nazwa\\_db WITH RECOVERY](#)

57

## Opcje odtwarzania bazy danych

### Użycie opcji FILE

- Odtwarza określony backup
- Musisz określić numer pliku (wymaga numeru pliku w zestawie)

### Użycie opcji MOVE...TO

- MOVE pozwala wskazać nową ścieżkę i nazwę pliku dla każdego logicznego pliku z backupu podczas odtwarzania
  - Określa gdzie odtworzyć pliki backupu
  - Używane do odtworzenia różnych dysków, serwerów lub standby SQL Server

### Użycie opcji REPLACE

- REPLACE oznacza, że odtwarzanie powinno zastąpić każdą bazę danych o tej samej nazwie istniejącą na danym serwerze. Zamienia istniejącą bazę danych
  - SQL Server nie wykonuje bezpiecznego sprawdzenia

58

## Odtworzenie bazy danych

z różnych typów kopii zapasowych

- Odtwarzanie z backupu Full Database
- Odtwarzanie z backupu Differential
- Odtwarzanie z backupu Transaction Log
- Odtwarzanie z backupu pliku lub grupy plików

59

## Odtwarzanie z pełnej kopii zapasowej

- Kiedy używać
  - Fizyczny dysk jest zepsuty
  - Cała baza jest zniszczona, zepsuta lub skasowana
  - Aby utrzymać identyczną kopię bazy na innym serwerze
- Określanie opcji odtworzenia
  - Rozpocząć proces odtworzenia z opcją RECOVERY
  - Odłożyć proces odtworzenia z opcją NORECOVERY

```
USE master
RESTORE DATABASE northwind
FROM dysk_c
WITH FILE = 2, RECOVERY
```

60

## Odtwarzanie z różnicowych kopii zapasowych

- Odtwarza tylko część bazy danych, które zmieniły się od ostatniego Full Database Backup
- Zwraca bazę do dokładnego stanu jaki był wykonywany Differential Backup
- Zajmuje mniej czasu niż seria Transaction Logs

Składnia jest ta sama kiedy odtwarzasz pełną bazę

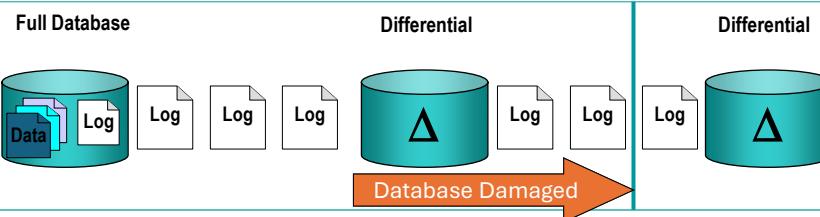
Określ plik backup, który zawiera differential backup

```
USE master  
RESTORE DATABASE northwind  
FROM dysk_c_diff  
WITH NORECOVERY
```

61

## Odtwarzanie - z kopii zapasowych dzienników transakcji

Northwind Database Backups

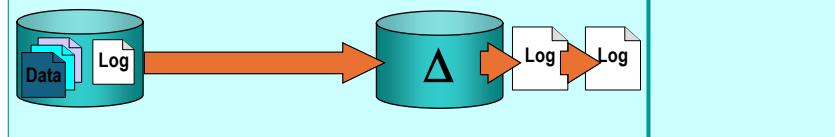


Restore Northwind Database

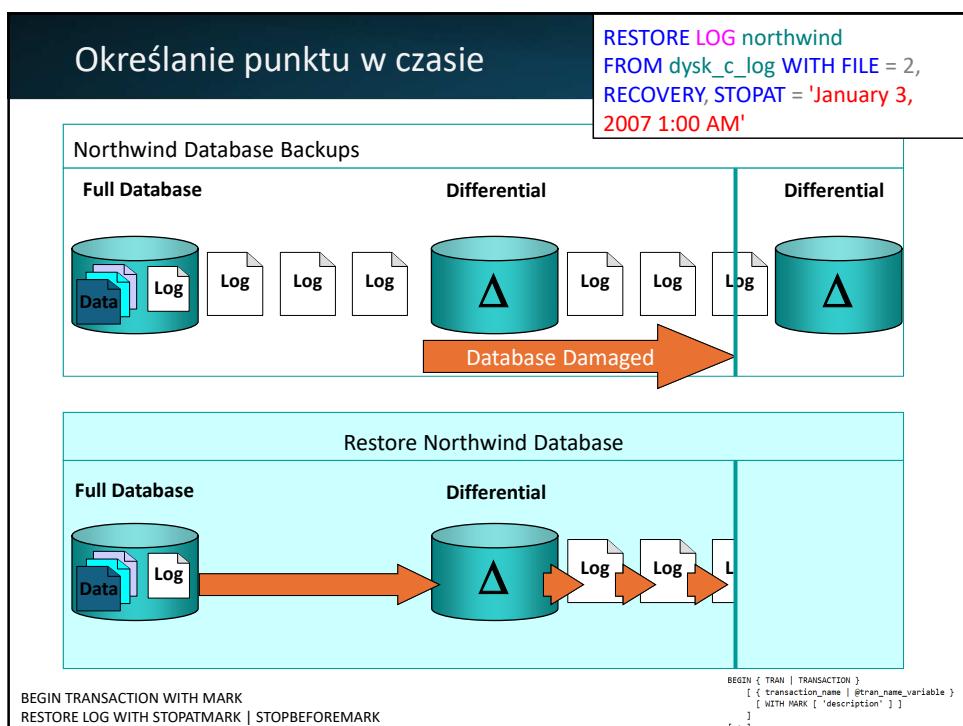
```
USE master  
RESTORE LOG northwind  
FROM nwindbaclog  
WITH FILE = 2, RECOVERY
```

Full Database

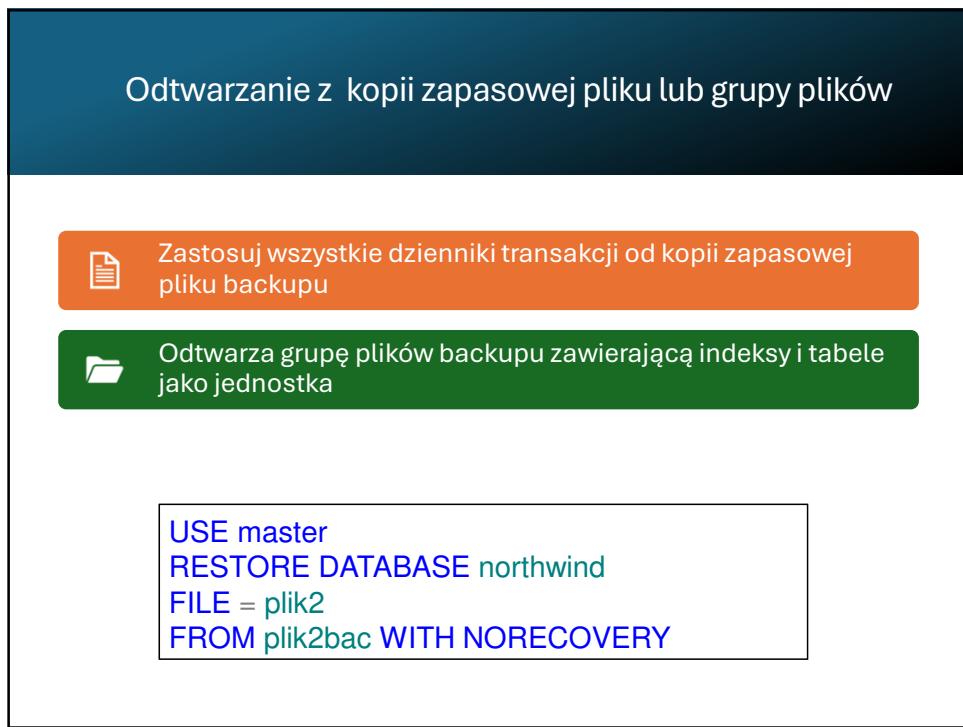
Differential



62



63



64

## Przywracanie uszkodzonych systemowych baz danych

Przywrócenie systemowej bazy z kopii zapasowej

Przebudowanie systemowych baz danych

Dołączenie lub przywrócenie baz danych użytkowników

- Odtworzenie z kopii zapasowej
- Dołączenie - CREATE DATABASE ... FOR ATTACH;
- Dołączenie z wykorzystaniem `sp_attach_db` lub `sp_attach_single_file_db` (deprecated)

65

## Polecane praktyki

### 1. Uzyskaj informacje o kopiah zapasowych przed odtworzeniem

- Sprawdź typ backupu (pełny, różnicowy, log), datę, LSN, lokalizację.

**Polecenie:** RESTORE HEADERONLY, RESTORE FILELISTONLY.

### 2. Określ opcję NORECOVERY na wszystkich backupach z wyjątkiem ostatniego

- Pozwala zastosować kolejne backupy (różnicowe, logi) bez zamknięcia bazy.

### 3. Używaj opcji RECOVERY na ostatnim backupie

- Przywraca bazę do spójnego stanu i udostępnia ją do pracy.

### 4. Dodawaj markery przed wykonaniem operacji wysokiego ryzyka

- np. BEGIN TRANSACTION lub logowanie w tabeli audytowej.

### 5. Sprawdzaj pliki kopii zapasowej określając poleceniem RESTORE VERIFYONLY

- Weryfikuje integralność backupu bez odtwarzania.

### 6. Testuj odtwarzanie na środowisku testowym

- Regularne DR testy minimalizują ryzyko w produkcji.

### 7. Monitoruj przestrzeń dyskową przed restore

- Upewnij się, że jest miejsce na pliki danych i logów.

### 8. Używaj opcji MOVE przy zmianie lokalizacji plików

- Szczególnie przy odtwarzaniu na innym serwerze lub dysku.

### 9. Po restore wykonaj DBCC CHECKDB

- Sprawdzenie integralności bazy po odtworzeniu.

### 10. Odbuduj indeksy i zaktualizuj statystyki

- Poprawa wydajności po dużych operacjach.

66

# Automatyzacja zadań administratora

1

## Informacje podstawowe

- Powody automatyzacji
- Wstęp do automatyzacji SQL Server
- Automatyzowanie rutynowych zadań utrzymywania
- Tworzenie alarmów
- Diagnozowanie serwera

2

1

## Powody automatyzacji

- **Wykonywanie regularnie planowanych zadań**
  - Backup bazy danych
  - Importowanie i eksportowanie danych
- **Rozpoznawanie i odpowiadanie na potencjalne problemy**
  - Odpowiedź na zdarzenia SQL Server ( błędy)
  - Definiowanie warunków wykonania

3

## Wstęp do automatyzacji SQL Server

- Automatyzowanie administracji SQL Server
- Zapisywanie zdarzeń do Application Log
- Przygotowania do automatyzacji
- Konfigurowanie poczty

4

## Zapisywanie zdarzeń do Application Log

- Poziomy ważności błędów SQL Server od 19 do 25
- Systemowe przechowywane procedury `sp_addmessage` lub `sp_altermessage`
- Wyrażenie `RAISERROR WITH LOG`
- Rozszerzona przechowywana procedura `xp_logevent`

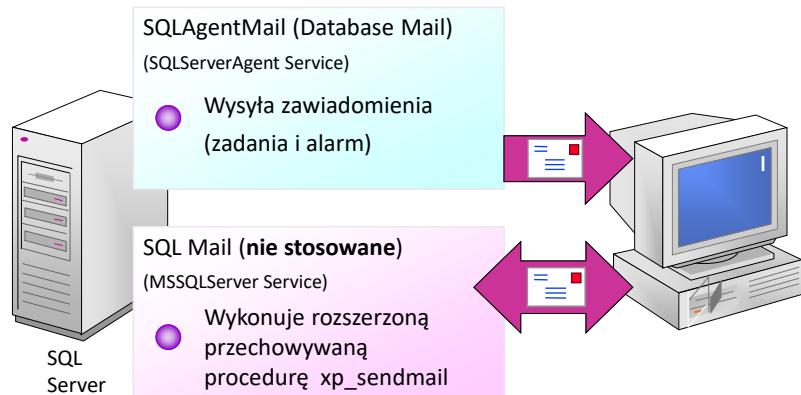
5

## Przygotowania do automatyzacji

- Upewnij się że SQL Server Agent jest uruchomiony
- Zweryfikuj czy SQL Server Agent Logon Account jest mapowany do roli sysadmin
  - Lokalne konto systemowe
  - Lokalne konto użytkownika
  - Domenowe konto użytkownika
- Zweryfikuj czy SQL Server Agent może się połączyć do lokalnego SQL Server

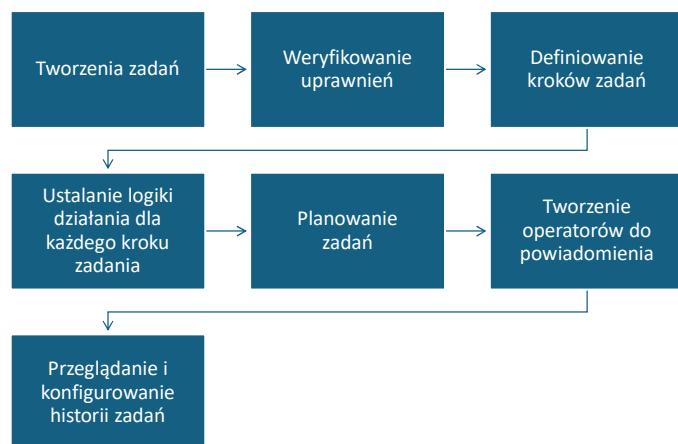
6

## Konfigurowanie poczty



7

## Automatyzowanie rutynowych zadań utrzymywania



8

## Tworzenie zadań

Upewnij się, że zadanie jest możliwe

Określ właściciela zadania

Ustal gdzie zadanie będzie wykonywane

Stwórz kategorię zadania

9

## Weryfikowanie uprawnień

Wykonywanie Transact-SQL

- SQL Server Agent przywołuje wyrażenie EXECUTE AS (dawniej SETUSER), kiedy właściciel zadania nie jest członkiem roli **sysadmin**

Wykonywanie komend systemu operacyjnego lub aktywnego języka skryptowego

- Członkowie roli **sysadmin** używają konta logowania SQL Server Agent
- Właściciele zadania, którzy nie są członkami roli **sysadmin** używają SQLAgentCmdExec dla kont Windows

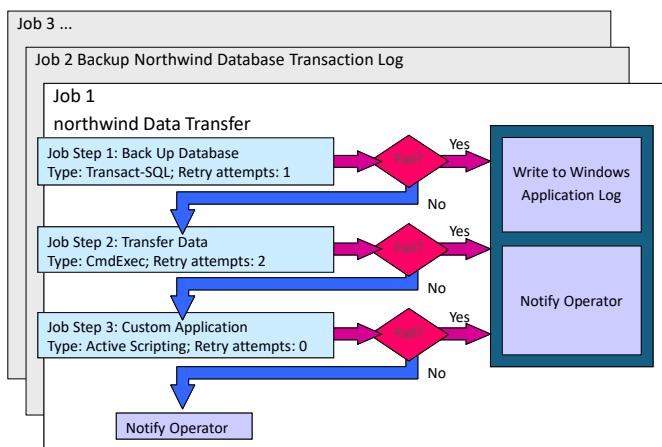
10

## Definiowanie kroków zadania

- Używanie poleceń języka Transact-SQL
- Używanie komend system operacyjnego
- Używanie aktywnych języków skryptowych Microsoft Visual Basic, Scripting Edition (VBScript) lub JavaScript
- Używanie procesów Replikacji

11

## Ustalanie logicznego przepływu danych dla każdego kroku zadania



12

## Planowanie zdań

### Job 2: Backup Northwind Database Transaction Log

#### Schedule: M-F Shift 1

Sun	Mon	Tues	Wed	Thur	Fri	Sat
	✓	✓	✓	✓	✓	

Every 2 Hours  
From: 8:00 A.M.  
To: 5:00 P.M.

#### Schedule: Weekend

Sun	Mon	Tues	Wed	Thur	Fri	Sat
✓						✓

Every 8 Hours  
From: 12:00 A.M.  
To: 11:59 P.M.

#### Schedule: M-F Shift 2

Sun	Mon	Tues	Wed	Thur	Fri	Sat
	✓	✓	✓	✓	✓	

Every 4 Hours  
From: 5:01 P.M.  
To: 7:59 A.M.

#### Schedule: CPU Idle

Sun	Mon	Tues	Wed	Thur	Fri	Sat

CPU Idle

13

## Tworzenie operatorów do powiadomienia

### Job: Northwind Data Transfer

#### Job Step 1: Back Up Transaction Log

#### Job Step 2: Transfer Data

#### Job Step 3: Back Up Database

Job failed

#### Notify Operator

Operator Name	E-mail	Pager	Net send
Meng Phua	mail	✓	
Nwind Admins	✓	✓	✓
Jose Lugo		✓	

#### Pager Schedule

12:01 - 8:00 A.M. Meng Phua  
8:01 - 6:00 P.M. Nwind Admins  
6:01 - 12:00 P.M. Jose Lugo

14

## Przeglądanie i konfigurowanie historii zadań

- Przeglądanie indywidualne historii zadań
  - Rezultat kroku zadania - sukces lub porażka
  - Czas wykonania
  - Błędy i uwagi
- Konfigurowanie wielkości historii zadań
  - Zachowywanie informacji o każdym zadaniu
  - Nadpisanie historii, kiedy został osiągnięty jego maksymalny rozmiar

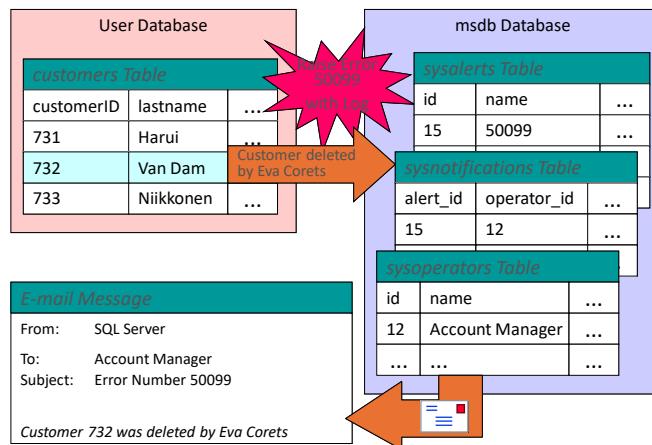
15

## Tworzenie alarmów

- Używanie alarmów do odpowiadania na ewentualne problemy
- Tworzenie alarmów do odpowiadania na błędy SQL Server
- Tworzenie alarmów na błędy zdefiniowane przez użytkownika
- Odpowiadanie na wyniki działań warunkowych alarmów
- Przypisanie operatora Fail-Safe

16

## Używanie alarmów do odpowiadania na potencjalne problemy



17

## Tworzenie alarmów reagujących na błędy SQL Server

### Definiowanie alarmów na błędy SQL Server

- dostarczane systemowo lub definiowane przez użytkownika
- Muszą być wpisane do dziennika Windows

### Definiowanie alarmów na poziomy ważności błędów

- Poziomy ważności od 19 do 25 są automatycznie zapisywane w dzienniku Windows
- Konfigurowanie wysyłania alarmów

18

## Tworzenie alarmów na błędy zdefiniowane przez użytkownika

### 1 Tworzenie komunikatu błędu

- Numer błędu musi być większy niż 50000
- Można używać parametry

```
EXEC sp_addmessage 50099, 16, 'Customer %d  
was deleted by %s', 'us_english', 'true'
```

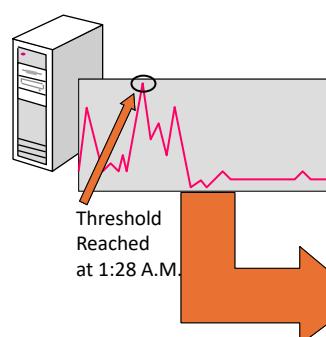
### 2 Reaguje na błędy z aplikacji bazodanowej

- Używaj operatora RAISERROR
- Deklaruj zmienne dla zwracanych parametrów

### 3 Definiuj alarmy na wiadomości o błędzie

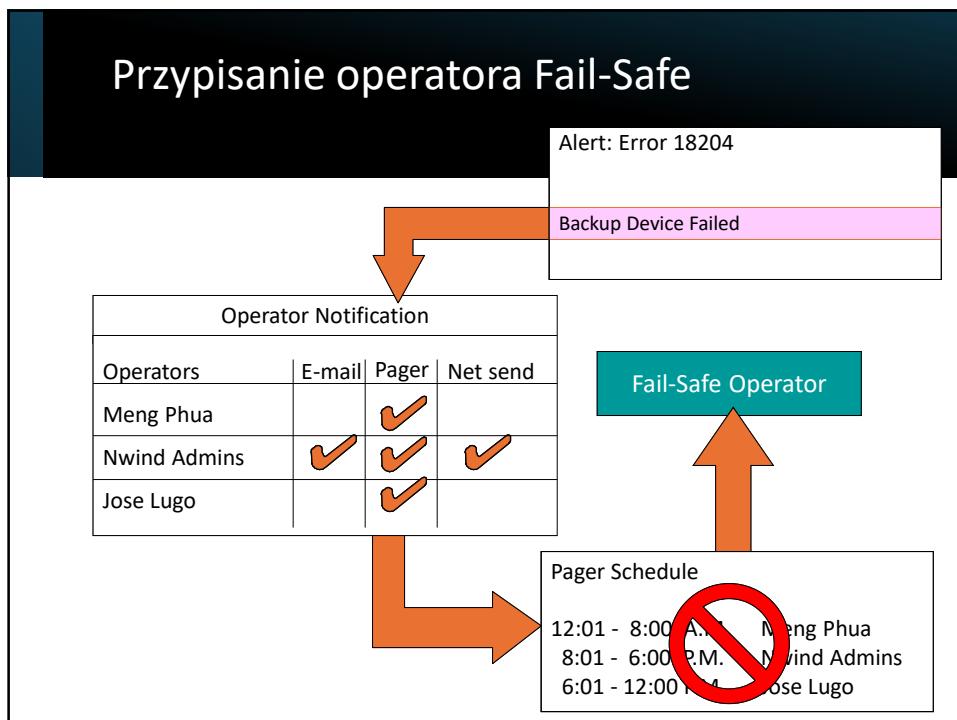
19

## Alarmy reagujące na spełnienie określonych warunków

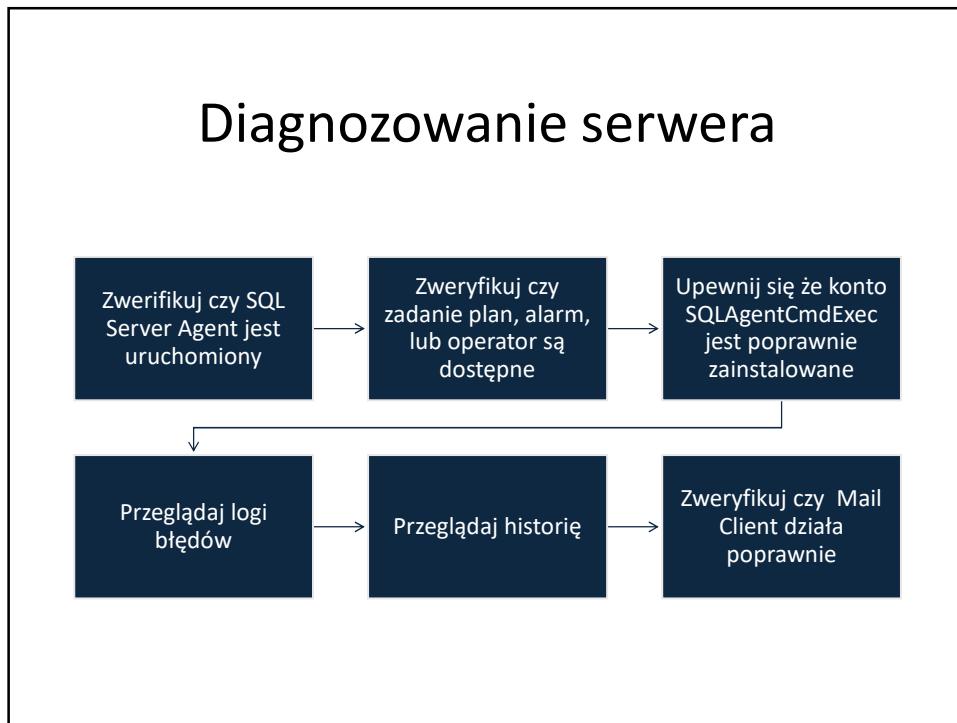


Alert 3 All Databases: Severity Level 18
Alert 2 northwind Database: Transfer Data Error
Alert 1: northwind Database: Log 75% Full
Execute Job: Job 2: Back Up Northwind Transaction Log
Operators to Notify:
Operator Name      E-mail      Pager      Net send
Meng Phua <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Nwind Admins <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Jose Lugo <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Pager Schedule
12:01 - 8:00 A.M.   Meng Phua
8:01 - 6:00 P.M.     Nwind Admins
6:01 - 12:00 P.M.   Jose Lugo

20



21



22

## Alarmy diagnostyczne

Czynniki, które powodują alarmy diagnostyczne

- Windows Application Log jest pełny
- Użycie CPU jest zbyt wysokie
- Ilość odpowiedzi alarmowych jest duża

Rozwiązywanie alarmu diagnostycznego

- Uniemożliwianie tymczasowo alarmowania
- Zwiększanie opóźnienia pomiędzy odpowiedziami
- Poprawianie globalnego problemu zasobów
- Czyszczenie Windows Application Log

23

## Monitorowanie i utrzymywanie SQL Server

24

## Informacje podstawowe

- Po co monitorować SQL Server
- Narzędzia do monitorowania SQL Server
  - Microsoft Event Viewer
  - SQL Server Performance Monitor
  - Activity Monitor in SSMS
  - Transact-SQL
  - SQL Server Profiler
  - SQL Server Query Analyzer
- Tworzenie planu utrzymywania SQL Server

25

## Po co monitorować SQL Server

- Powody monitorowania SQL Server,
- Czynniki wpływające na wydajność,
- Identyfikacja spadków wydajności.

26

## Powody monitorowania SQL Server

- Ustalenie powodu małej wydajności
  - Nadmierna ilość I/O
  - Długość czasu CPU
  - Duży ruch w sieci
- Sprawdzenie przepustowości wszystkich procesów wszystkich użytkowników
- Sprawdzenie spójności (konsystencji) danych

27

## Czynniki wpływające na wydajność

- Server Hardware
- Operating System
- Network
- SQL Server
- Database Application
- Client Application

28

## Identyfikacja spadków wydajności

- Decyzja co badać:
  - System
  - Client
  - Zapytanie
- Znajomość akceptowalnego zakresu
- Ustalenie granicy wydajności

29

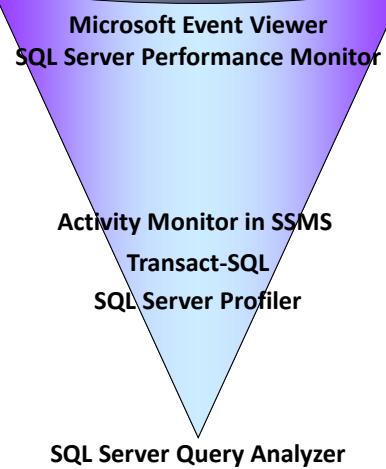
## Narzędzia do monitorowania SQL Server

- Wspólne zadania monitorowania
- Użycie Microsoft Event Viewer
- Użycie SQL Server Performance Monitor
- Użycie Activity Monitor w SSMS
- Użycie Transact-SQL do monitorowania SQL Server
- Użycie SQL Server Profiler
- Generowanie historii zapytań
- Użycie SQL Server Query Analyzer

30

## Wspólne zadania monitorowania

- Poziom monitorowania systemu:
  - Sprzęt
  - System operacyjny
  - Aplikacja
- SQL Server-Specific Monitoring
  - czynności SQL Server
  - Konsystencja danych (spójność)
- Określona wydajność zapytań



31

## Użycie Microsoft Event Viewer

- Microsoft Event Viewer wyświetla błędy, uwagi, i wiadomości informacyjne
- Microsoft Event Viewer umożliwia przeglądanie Windows NT Event Logs
  - Windows NT application log
  - Windows NT system log
  - Windows NT security log

32

## Użycie SQL Server Performance Monitor

- SQL Server - Specific Counters w Performance Monitor
- Stosowna ochrona
- Liczniki (wskaźniki) monitorowania
  - predefiniowane liczniki
  - liczniki SQL Server
  - liczniki Windows NT
  - liczniki zdefiniowane przez użytkownika

33

## Użycie Activity Monitor w SQL Server Management Studio

- SQL Server Process Information
- klucze, blokowanie i zakleszczenia
  - przegląd Process ID
  - przegląd obiektów

34

## Użycie Transact-SQL do monitorowania SQL Server

- Systemowe przechowywane procedury
- Zmienne globalne
- Polecenie Transact-SQL
- Polecenia DBCC
- Flagi śledzenia

35

## Użycie SQL Server Profiler

- Monitorowanie bieżącej aktywności serwera
  - Wybór zdarzeń do monitorowania
  - Wybrać kryteria śledzenia
  - Wybór danych do wychwycenia (np. czas rozpoczęcia, identyfikator sesji)
- Wychwytywanie danych w czasie rzeczywistym
- Wychwytywanie danych do pliku (\*.trc)

36

## Tworzenie planu utrzymania dla SQL Server

- Tworzenie planu utrzymania bazy danych
  - Aktualizowanie danych o optymalizacji danych
  - Weryfikowanie integralności danych periodycznie
  - Wykonywanie kopii bezpieczeństwa - backup-ów
  - Utrzymywanie historii
- Automatyzowanie zadań planu utrzymania bazy danych

37

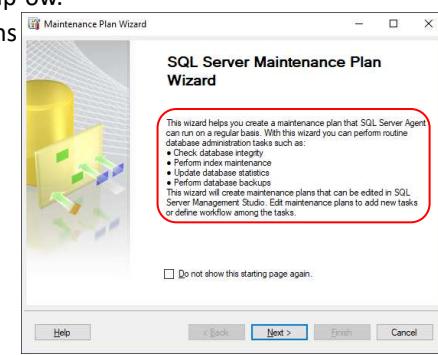
## Tworzenie planu utrzymania bazy danych

- Aktualizowanie informacji o optymalizacji danych
  - utrzymanie indeksowania poprzez użycie opcji **fillfactor**
  - UPDATE STATISTICS
  - DBCC SHRINKDATABASE
- Weryfikowanie integralności danych periodycznie
  - DBCC CHECKALLOC
  - DBCC CHECKDB
- Wykonywanie backup-ów
- Utrzymanie historii

38

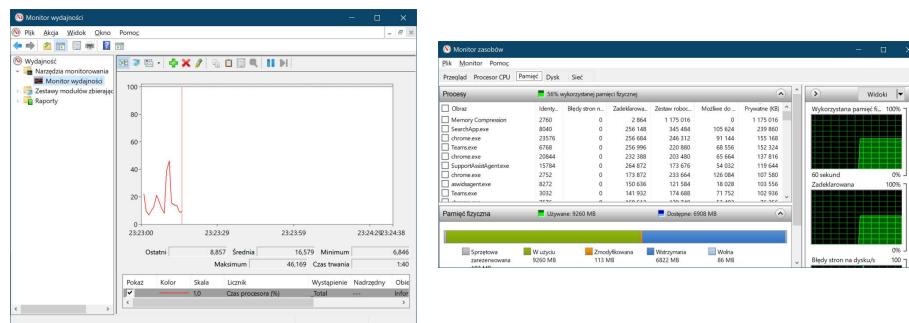
## Automatyzowanie zadań utrzymania bazy danych

- Kreator planu utrzymania bazy danych określa:
  - bazy danych objętych planem utrzymania,
  - testy weryfikacji danych (Check Database Integrity),
  - optymalizację danych (Reorganize Index i Rebuild Index),
  - częstotliwość i miejsce docelowe backup-ów.
- Korzystamy z SQL Server Maintenance Plans  
(plany konserwacji)  
w SQL Server Management Studio (SSMS)



39

## Co warto monitorować



Wskaźniki monitorowania można podzielić na dwie kategorie:

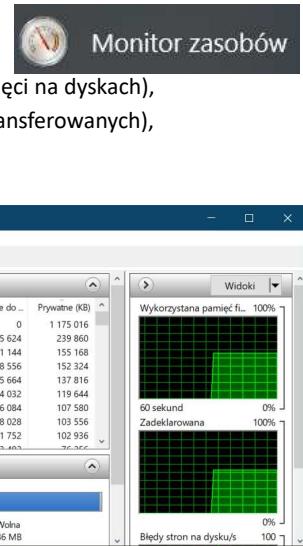
- Wskaźniki (liczniki) sprzętowe,
- Wskaźniki (liczniki) monitorujące SZBD.

40

## Wskaźniki sprzętowe (Resource Monitor)

Wskaźniki sprzętowe dotyczą monitorowania:

- procesorów serwera (procentowe użycie procesora),
- pamięci RAM (ilość wolnej pamięci, poziom stronicowania pamięci na dyskach),
- interfejsów sieciowych (zużycie przepustowości, ilość danych transferowanych),
- zapisu / odczytu z dysków twardych,
- ilości wolnego miejsca na dyskach twardych.



41

## Wskaźniki monitorujące SZBD (Performance Monitor)

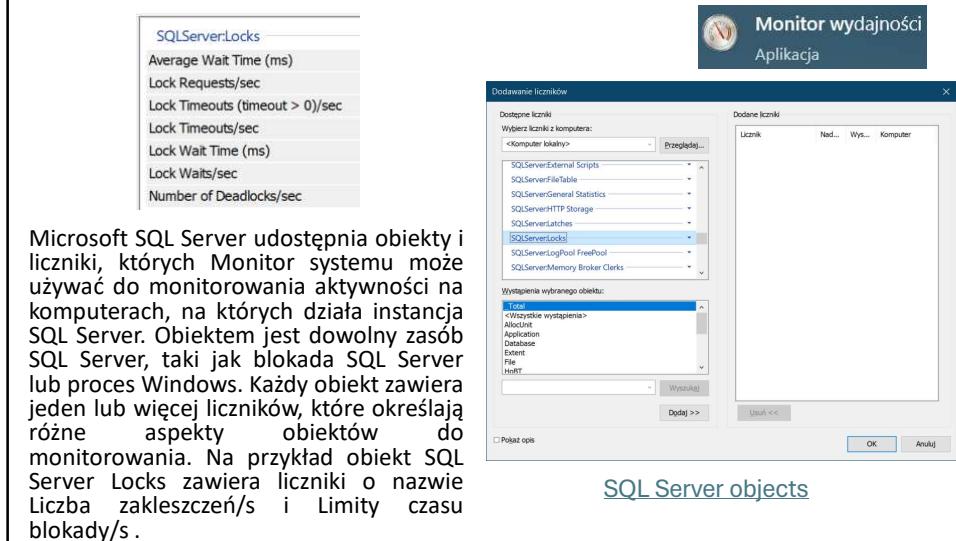
Wskaźniki monitorujące SZBD dotyczą monitorowania:

- rozmiaru plików baz danych,
- eskalacji blokad i zakleszczeń,
- ilości aktywnych transakcji,
- ilości aktywnych połączeń z bazami danych,
- czasu wykonania zapytań do baz danych,
- udanych / nieudanych prób logowania,
- i wiele innych



42

## Wskaźniki monitorujące SZBD (Performance Monitor)



Microsoft SQL Server udostępnia obiekty i liczniki, których Monitor systemu może używać do monitorowania aktywności na komputerach, na których działa instancja SQL Server. Obiektem jest dowolny zasób SQL Server, taki jak blokada SQL Server lub proces Windows. Każdy obiekt zawiera jeden lub więcej liczników, które określają różne aspekty obiektów do monitorowania. Na przykład obiekt SQL Server Locks zawiera liczniki o nazwie Liczba zakleszczeń/s i Limity czasu blokady/s.

43

## Wskaźniki monitorujące SZBD: DMV T-SQL

### SQL Server Operating System Related Dynamic Management Views (Transact-SQL)

Dynamiczne widoki zarządzania (DMV) powiązane z systemem operacyjnym SQL Server (SQLOS).

SQLOS jest odpowiedzialny za zarządzanie zasobami systemu operacyjnego specyficznymi dla SQL Server

sys.dm_os_buffer_descriptors	sys.dm_os_nodes
sys.dm_os_buffer_pool_extension_configuration	sys.dm_os_Performance_counters
sys.dm_os_child_instances	sys.dm_os_process_memory
sys.dm_os_cluster_nodes	sys.dm_osSchedulers
sys.dm_os_cluster_properties	sys.dm_os_server_diagnostics_log_configurations
sys.dm_os_dispatcher_pools	sys.dm_os_spinlock_stats
sys.dm_os_enumerate_fixed_drives	sys.dm_os_stacks
sys.dm_os_host_info	sys.dm_os_sys_info
sys.dm_os_hosts	sys.dm_os_sys_memory
sys.dm_os_latch_stats	sy
sys.dm_os_loaded_modules	s.dm_os_tasks sys.dm_os_threads
sys.dm_os_memory_brokers	sys.dm_os_virtual_address_dump
sys.dm_os_memory_cache_clock_hands	sys.dm_os_volume_stats
sys.dm_os_memory_cache_counters	sys.dm_os_waiting_tasks
sys.dm_os_memory_cache_entries	sys.dm_os_wait_stats
sys.dm_os_memory_cache_hash_tables	sys.dm_os_windows_info
sys.dm_os_memory_clerks	sys. dm_os_workers
sys.dm_os_memory_nodes	

44

## SQL Server - Transactions object

SQLQuery1.sql - (local).master (sa (75))

```
SELECT * FROM sys.dm_os_performance_counters
WHERE object_name LIKE '%Transactions%' and counter_name like 'Free space %';
```

Results

object_name	counter_name	instance_name	cntr_value	cntr_type
SQLServer:Transactions	Free Space in tempdb (KB)		29376	65792

Monitor wydajności

Liczniki transakcji

Opis
Wolne miejsce w tempdb (KB) Ilość miejsca (w kilobajtach) dostępnego w formacie tempdb. Musi być wystarczająca ilość wolnego miejsca, aby pomieścić zarówno magazyn wersji poziomu izolacji migawki, jak i wszystkie nowe obiekty tymczasowe utworzone w tej instancji aparatu bazy danych.

45

## SQL Server - Locks object

SQLQuery1.sql - (local).master (sa (75))

```
SELECT * FROM sys.dm_os_performance_counters
WHERE object_name LIKE '%Locks%' and counter_name LIKE '%Deadlocks%'
and instance_name LIKE '_Total%';
```

Results

object_name	counter_name	instance_name	cntr_value	cntr_type
SQLServer:Locks	Number of Deadlocks/sec	_Total	0	272696576

SQLServer:Locks

- Average Wait Time (ms)
- Lock Requests/sec
- Lock Timeouts (timeout > 0)/sec
- Lock Timeouts/sec
- Lock Wait Time (ms)
- Lock Waits/sec
- Number of Deadlocks/sec

Item	Description
_Total	Information for all locks.
AllocUnit	A lock on an allocation unit.
Application	A lock on an application-specified resource.
Database	A lock on a database, including all objects in the database.
Extent	A lock on a contiguous group of 8 pages.
File	A lock on a database file.
Heap/B-tree	Heap or B-tree. A lock on a heap of data pages, or on the B-tree structure of an index.
Key	A lock on a row in an index.
Metadata	A lock on a piece of catalog information, also called metadata.
Object	A lock on table, stored procedure, view, etc, including all data and indexes. The object can be anything that has an entry in sys.all_objects.
OIB	Lock resource for online index build locks, specifically for a online index build LOB tracking table.
Page	A lock on an 8-kilobyte (KB) page in a database.
RID	Row ID. A lock on a single row in a heap.
RowGroup	Lock resource for a columnstore index rowgroup.
Xact	Lock resource for transactions.

46

## Polecane praktyki



**Ustal granicę wydajności**



**Monitoruj aktualną wydajność**



**Identyfikuj zmniejszenie wydajności**

47

48

## Mechanizmy szyfrowania danych - Transparent Data Encryption (TDE)

### Transparent Data Encryption (TDE)

- **Opis:** Szyfruje dane na poziomie plików baz danych (pliki MDF, LDF i BAK) w sposób transparentny dla aplikacji.
- **Mechanizm:**
  - Klucz główny (Database Master Key) w bazie danych.
  - Certyfikat lub klucz asymetryczny w SQL Server.
  - Klucz szyfrowania bazy danych (Database Encryption Key, DEK).
- **Zastosowanie:**
  - Ochrona przed nieautoryzowanym dostępem do plików baz danych (np. w przypadku kradzieży kopii zapasowych).
- **Zalety:**
  - Transparentność – aplikacje nie wymagają zmian.
  - Szyfruje dane "w spoczynku".
- **Wady:**
  - Nie chroni danych "w ruchu" ani w pamięci RAM.
  - Możliwy wpływ na wydajność.

[Tutorial: Getting started with Always Encrypted - SQL Server | Microsoft Learn](#)

49

## Mechanizmy szyfrowania danych - Always Encrypted

### Always Encrypted

- **Opis:** Szyfrowanie danych na poziomie kolumn bazy danych z możliwością kontroli dostępu na poziomie klienta.
- **Mechanizm:**
  - Klucze szyfrowania są przechowywane poza SQL Server (np. w Azure Key Vault lub na kliencie).
  - Dwa tryby szyfrowania:
    - **Deterministyczne** – ten sam tekst szyfrowany daje taki sam wynik.
    - **Losowe** – generuje różne wartości dla tego samego tekstu.
- **Zastosowanie:**
  - Ochrona poufnych danych (np. numerów kart kredytowych, PESEL) przed administratorami bazy danych i innymi nieautoryzowanymi użytkownikami.
- **Zalety:**
  - Chroni dane "w spoczynku" i "w ruchu".
  - Wysoki poziom bezpieczeństwa dzięki oddzieleniu zarządzania kluczami od bazy danych.
- **Wady:**
  - Wymaga wsparcia po stronie aplikacji.
  - Ograniczenia w operacjach na danych (np. brak możliwości użycia LIKE).

50

25

## Mechanizmy szyfrowania danych - Backup Encryption

### Szyfrowanie kopii zapasowych (Backup Encryption)

- **Opis:** Szyfrowanie kopii zapasowych bazy danych w SQL Server.
- **Mechanizm:**
  - Wybór algorytmu szyfrowania (np. AES 256).
  - Wykorzystanie certyfikatu lub klucza asymetrycznego w SQL Server.
- **Zastosowanie:**
  - Ochrona kopii zapasowych przed nieautoryzowanym dostępem.
- **Zalety:**
  - Łatwa implementacja.
  - Brak wpływu na wydajność bazy danych.
- **Wady:**
  - Ochrona dotyczy tylko plików kopii zapasowej.

51

## Mechanizmy szyfrowania danych - Cell-Level Encryption (CLE)

### Cell-Level Encryption (CLE)

- **Opis:** Szyfrowanie danych na poziomie poszczególnych komórek (kolumn).
- **Mechanizm:**
  - Funkcje SQL Server takie jak EncryptByKey, DecryptByKey, itp.
  - Klucz symetryczny lub asymetryczny.
- **Zastosowanie:**
  - Ochrona wybranych danych poufnych.
- **Zalety:**
  - Elastyczność – możliwość szyfrowania wybranych danych.
  - Szyfrowanie i deszyfrowanie możliwe w samej bazie danych.
- **Wady:**
  - Większy narzut na aplikację (konieczność ręcznego szyfrowania/deszyfrowania).
  - Potencjalny wpływ na wydajność.

52

## Mechanizmy szyfrowania danych - Transport Layer Security (TLS)

### Transport Layer Security (TLS)

- **Opis:** Zabezpieczenie danych "w ruchu" między SQL Server a klientem.
- **Mechanizm:**
  - Szyfrowanie połączenia za pomocą certyfikatów SSL/TLS.
- **Zastosowanie:**
  - Zapobieganie podsłuchiwaniu transmisji danych.
- **Zalety:**
  - Ochrona danych przesyłanych między klientem a serwerem.
  - Transparentne dla aplikacji.
- **Wady:**
  - Nie chroni danych "w spoczynku".

53

## Mechanizmy szyfrowania danych - porównanie

Mechanizm	Poziom ochrony	Ochrona danych "w ruchu"	Ochrona danych "w spoczynku"	Łatwość wdrożenia	Wpływ na aplikacje
Transparent Data Encryption (TDE)	Cała baza danych	Nie	Tak	Łatwe	Żaden
Always Encrypted	Wybrane kolumny	Tak	Tak	Trudniejsze	Wysoki
Backup Encryption	Kopie zapasowe	Nie	Tak	Łatwe	Żaden
Cell-Level Encryption	Wybrane kolumny/ komórki	Nie	Tak	Trudne	Wysoki
Transport Layer Security (TLS)	Połączenia sieciowe	Tak	Nie	Łatwe	Żaden

54

## Mechanizmy szyfrowania danych - Podsumowanie

- **TDE** jest dobrym wyborem do ochrony całych baz danych przy minimalnym wysiłku.
- **Always Encrypted** nadaje się do ochrony najbardziej wrażliwych danych i zapewnia wysoki poziom bezpieczeństwa.
- **Backup Encryption** to podstawa w przypadku tworzenia kopii zapasowych.
- **Cell-Level Encryption** pozwala na precyzyjne szyfrowanie, ale jest trudniejsze do zarządzania.
- **TLS** to standardowa praktyka dla bezpiecznych połączeń sieciowych.

55

## Mechanizmy generowanie skrótów - HashBytes

W SQL Server funkcja HASHBYTES służy do generowania skrótów (hashy) danych za pomocą algorytmów kryptograficznych. Funkcja HASHBYTES jest używana do tworzenia jednokierunkowego skrótu, który nie może być odszyfrowany.

```
HASHBYTES ( '<algorithm>', { @input | 'input' } )
```

```
<algorithm> ::= MD2 | MD4 | MD5 | SHA | SHA1 | SHA2_256 | SHA2_512
```

Zastosowanie:

- Przechowywanie haseł:
  - Zamiast przechowywać hasła w postaci tekstu, zapisujemy je jako hasze, aby zwiększyć bezpieczeństwo.
  - Warto używać bezpiecznych algorytmów, takich jak SHA2\_256 lub SHA2\_512.

56



## Transparent Data Encryption (TDE) w SQL Server

57

### Wprowadzenie do TDE

- **Czym jest Transparent Data Encryption?**
  - Funkcja SQL Server służąca do szyfrowania danych w spoczynku.
- **Cel TDE:**
  - Ochrona danych przed nieautoryzowanym dostępem w przypadku utraty lub kradzieży plików bazy danych.
- **Wspierane wersje SQL Server:**
  - Dostępne w edycjach Enterprise, Developer, oraz od SQL Server 2019 w Standard Edition.

58

## Zasady Działania TDE

- **Szyfrowanie danych w spoczynku (data at rest):**
  - Szyfrowane są pliki bazy danych (MDF, NDF) oraz pliki dziennika transakcji (LDF).
- **Klucz szyfrowania danych (DEK - Database Encryption Key):**
  - Klucz szyfrowania przechowywany w samej bazie, chroniony przez certyfikat lub klucz asymetryczny w bazie systemowej master.
- **Przepływ procesu szyfrowania:**
  1. Tworzenie certyfikatu w bazie **master**.
  2. Tworzenie klucza szyfrowania bazy (DEK).
  3. Włączenie TDE dla bazy danych.

59

## Korzyści z użycia TDE

- **Ochrona danych:**
  - Uniemożliwia dostęp do danych w przypadku kradzieży plików bazy danych.
- **Przejrzystość dla aplikacji:**
  - Proces szyfrowania i deszyfrowania jest transparentny dla aplikacji klienckich.
- **Zgodność z regulacjami:**
  - Pomaga spełnić wymogi bezpieczeństwa danych, np. RODO.

60

## Proces wdrażania TDE

- Tworzenie certyfikatu

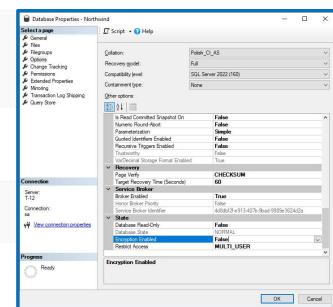
```
USE master;
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'ComplexPassword!';
CREATE CERTIFICATE TDE_Cert WITH SUBJECT = 'TDE Certificate';
```

- Tworzenie klucza szyfrowania bazy danych (DEK)

```
USE YourDatabase;
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TDE_Cert;
```

- Włączenie TDE

```
ALTER DATABASE YourDatabase
SET ENCRYPTION ON;
```



61

## Encrypt Column (Always Encrypted) w SQL Server

62

31

## Wprowadzenie do Always Encrypted

- **Czym jest Always Encrypted?**
  - Technologia SQL Server służąca do szyfrowania wybranych kolumn w tabelach bazy danych.
- **Cel Always Encrypted:**
  - Ochrona wrażliwych danych w bazie, które nigdy nie są dostępne w formie jawnej na serwerze SQL.
- **Jak działa?**
  - Szyfrowanie i deszyfrowanie odbywa się po stronie klienta (np. aplikacji). Serwer SQL przechowuje tylko dane zaszyfrowane.

63

## Zasady działania Encrypt Column

- **Proces szyfrowania:**
  - Klient szyfruje dane przed ich wysłaniem do serwera.
- **Przechowywanie kluczy:**
  - Klucz główny (Column Master Key - CMK) jest przechowywany w bezpiecznym miejscu, np. w magazynie certyfikatów lub Azure Key Vault.
  - Klucz szyfrowania kolumny (Column Encryption Key - CEK) szyfruje dane i jest przechowywany w bazie w postaci zaszyfrowanej.
- **Przykład użycia:**
  - Szyfrowanie numerów kart kredytowych w bazie danych czy haseł.

64

## Korzyści z użycia Encrypt Column

- **Ochrona prywatności danych:**
  - Wrażliwe dane są zabezpieczone na serwerze SQL.
- **Zgodność z regulacjami:**
  - Spełnia wymogi np. RODO.
- **Granularność szyfrowania:**
  - Możliwość szyfrowania wybranych kolumn w tabeli.
- **Ograniczenie dostępu:**
  - Administratorzy serwera SQL nie mają dostępu do danych w formie jawniej.

65

## Tryby szyfrowania kolumn

- **Deterministyczny (Deterministic Encryption):**
  - Taki sam plaintext zawsze generuje ten sam ciphertext.
  - Możliwość użycia w filtrach WHERE, GROUP BY, itp.
  - Przykład: Numery PESEL.
- **Losowy (Randomized Encryption):**
  - Każde zaszyfrowanie tej samej wartości generuje inny ciphertext.
  - Wyższy poziom bezpieczeństwa, ale brak możliwości wyszukiwania w danych.
  - Przykład: Numery kart kredytowych

66

## Proces wdrażania Encrypt Column

- Tworzenie Klucza Głównego (CMK)

```
CREATE COLUMN MASTER KEY CMK_Local  
WITH  
(  
    KEY_STORE_PROVIDER_NAME = 'MSSQL_CERTIFICATE_STORE',  
    KEY_PATH = 'CurrentUser/My/CertName'  
)
```

- Tworzenie Klucza Szyfrowania Kolumny (CEK)

```
CREATE COLUMN ENCRYPTION KEY CEK_Local  
WITH VALUES  
(  
    COLUMN_MASTER_KEY = CMK_Local,  
    ALGORITHM = 'RSA_OAEP',  
    ENCRYPTED_VALUE = <encrypted_value>  
)
```

- Szyfrowanie Kolumny

```
CREATE TABLE SensitiveData (  
    Id INT PRIMARY KEY,  
    EncryptedColumn NVARCHAR(100) COLLATE Latin1_General_BIN2  
    ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = CEK_Local, ENCRYPTION_TYPE = DETERMINISTIC)  
)
```

67

## Najlepsze praktyki i ograniczenia

- **Najlepsze praktyki:**

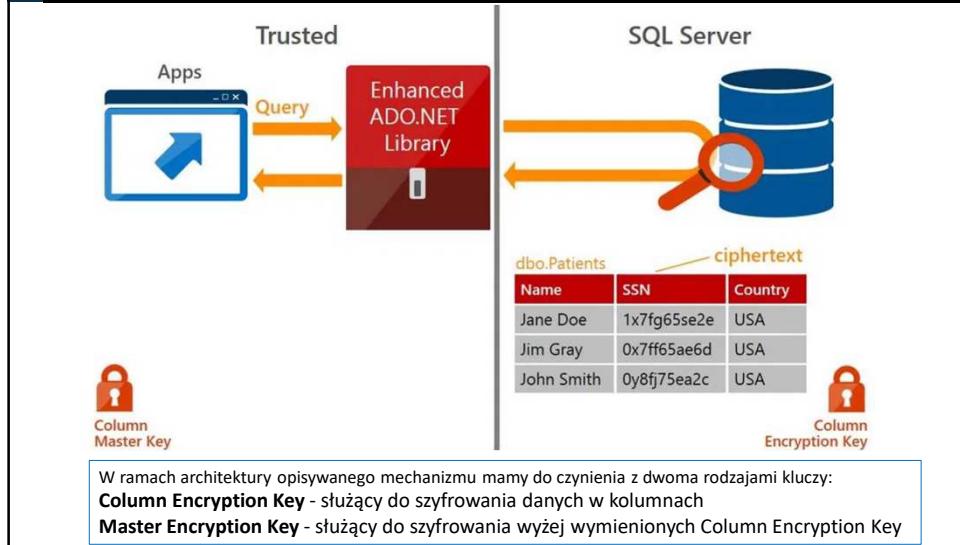
- Wybieraj odpowiedni tryb szyfrowania (deterministyczny /losowy) w zależności od wymagań aplikacji.
- Bezpiecznie przechowuj klucze szyfrowania (CMK).
- Regularnie monitoruj dostęp do danych zaszyfrowanych.

- **Ograniczenia:**

- Brak wsparcia dla funkcji takich jak LIKE, PATINDEX, SUBSTRING.
- Wymaga sterowników klienckich kompatybilnych z Always Encrypted.
- Może wpływać na wydajność operacji w aplikacjach

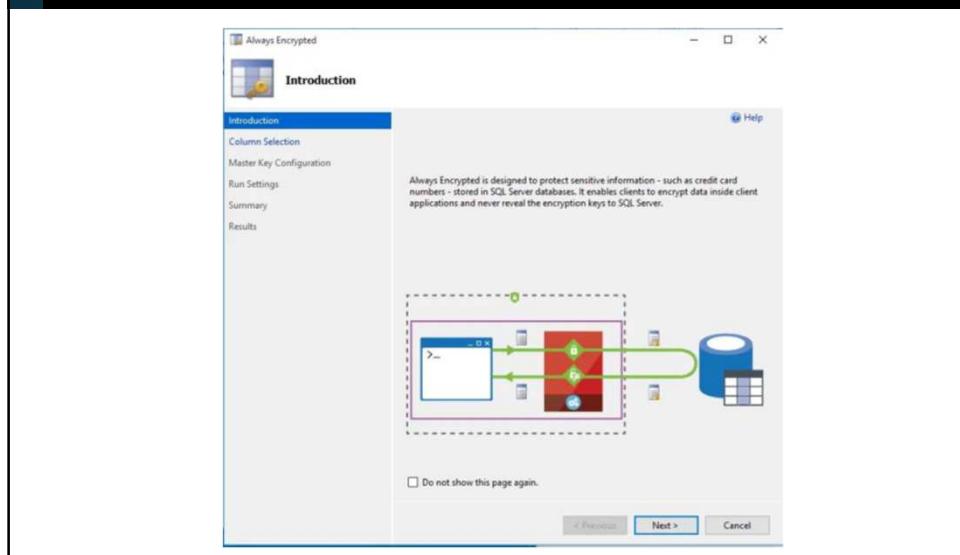
68

## Zasady działania Always Encrypted



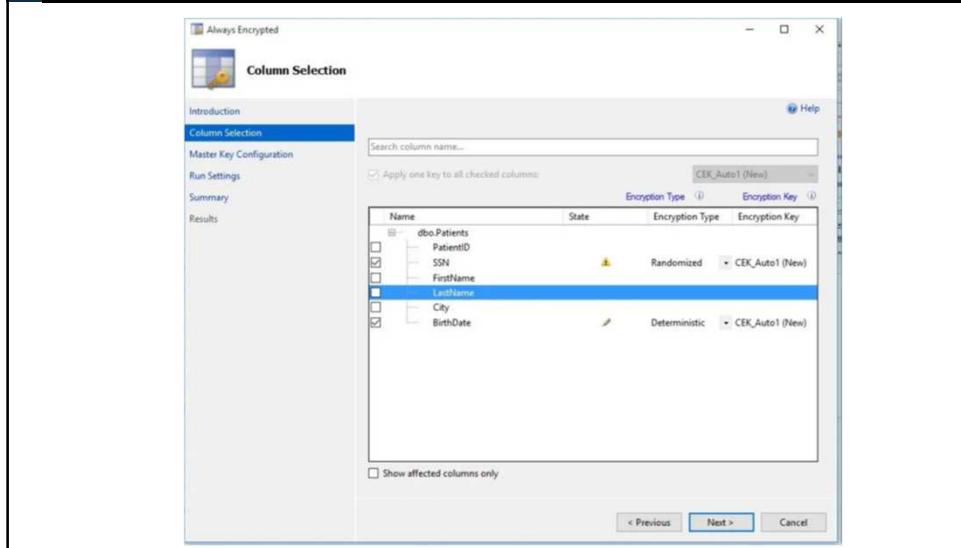
69

## Always Encrypted



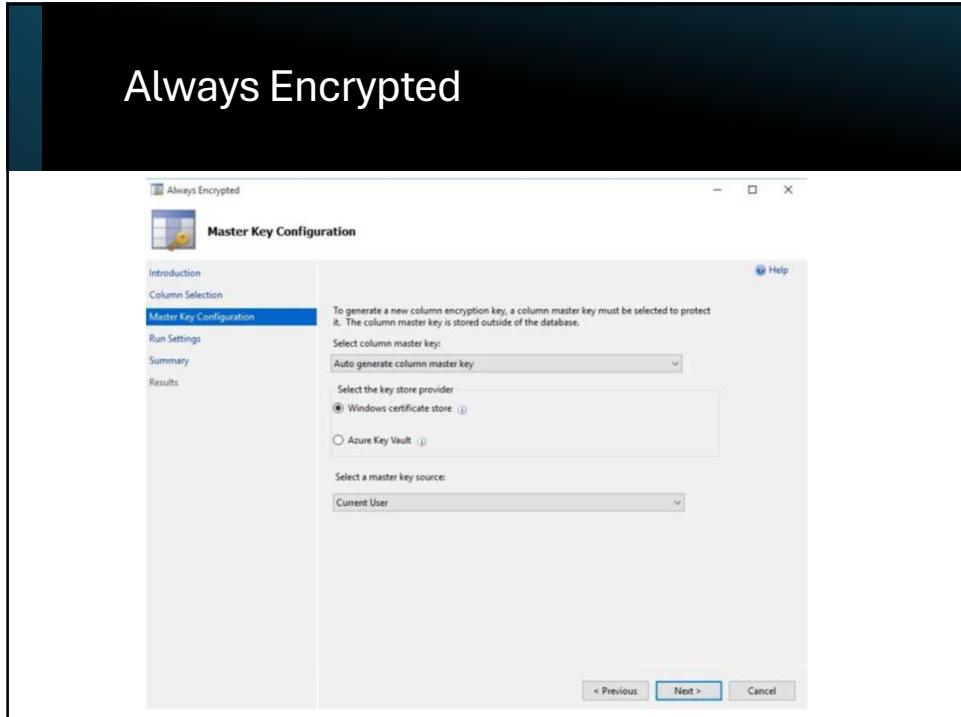
70

## Always Encrypted

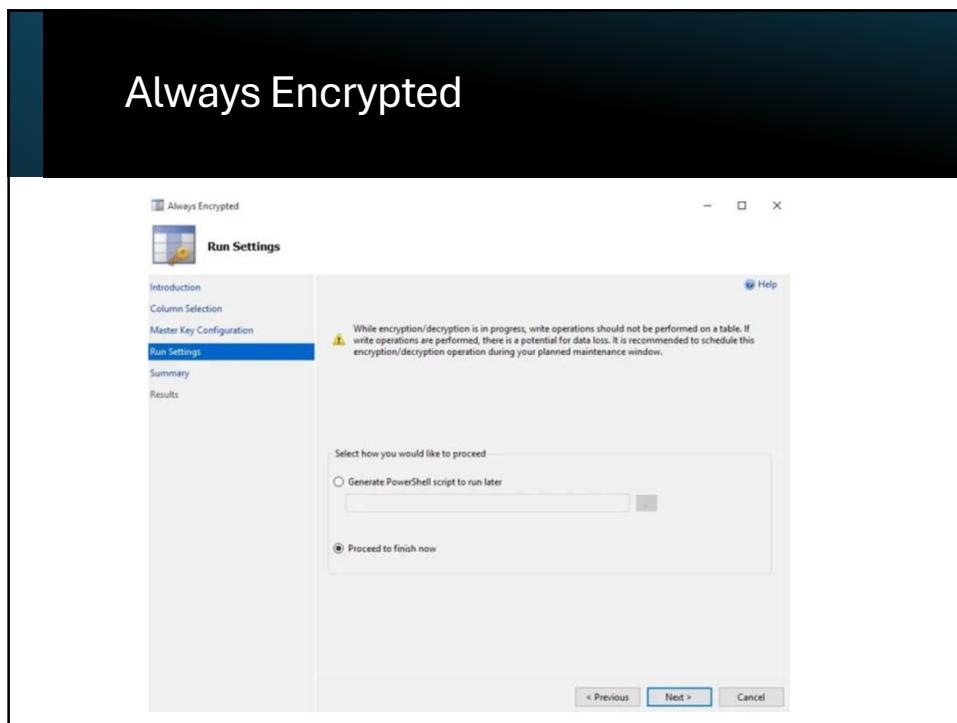


71

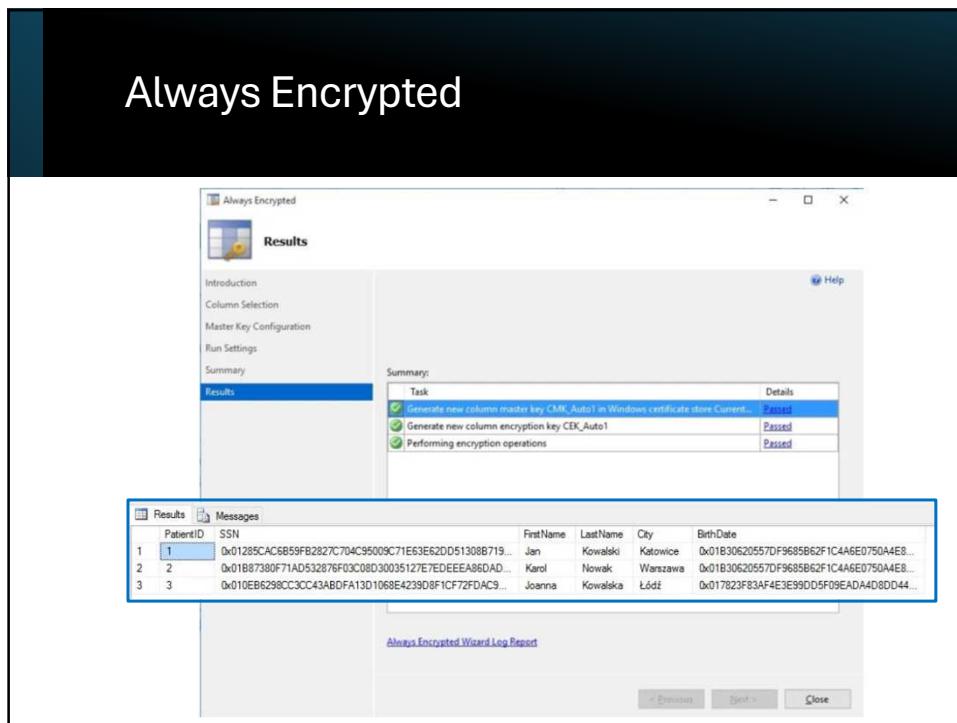
## Always Encrypted



72



73



74

## Always Encrypted

name	encryption_type	encryption_type_desc	encryption_algorithm_name	column_encryption_key_id
PatientID	NULL	NULL	NULL	NULL
SSN	2	RANDOMIZED	AEAD_AES_256_CBC_HMAC_SHA_256	1
FirstName	NULL	NULL	NULL	NULL
LastName	NULL	NULL	NULL	NULL
City	NULL	NULL	NULL	NULL
BirthDate	1	DETERMINISTIC	AEAD_AES_256_CBC_HMAC_SHA_256	1



75

## Podsumowanie

- Always Encrypted chroni wrażliwe dane nawet przed administratorami serwera.
- Szyfrowanie i deszyfrowanie odbywa się po stronie klienta.
- Planuj strategię szyfrowania zgodnie z wymaganiami biznesowymi i bezpieczeństwa.

76

## Inne podejście do szyfrowania danych

Cecha	HASHBYTES	EncryptByKey
Rodzaj operacji	Generowanie hashu	Szyfrowanie danych
Jedno-/dwukierunkowość	Jednokierunkowe (nie można odwrócić)	Dwukierunkowe (można odszyfrować dane)
Wymaga klucza	Nie	Tak (klucz symetryczny)
Zastosowanie	Weryfikacja integralności, hasła	Ochrona danych wrażliwych
Zależność od algorytmu	Obsługuje różne algorytmy hashujące	Używa algorytmów szyfrowania symetrycznego

77

Policy Management (Zarządzanie Politykami)  
Bazy Danych w SQL Server

78

39

## Wprowadzenie do Policy Management

- **Czym jest Policy Management?**

- Narzędzie w SQL Server umożliwiające tworzenie i zarządzanie politykami konfiguracji i bezpieczeństwa bazy danych.

- **Dlaczego zarządzanie politykami jest ważne?**

- Zapewnia zgodność z najlepszymi praktykami i politykami bezpieczeństwa.

- **Przykład zastosowania:**

- Automatyczne sprawdzanie, czy wszystkie indeksy są uporządkowane zgodnie z określoną regułą.

79

## Korzyści z zarządzania politykami

- **Standaryzacja konfiguracji:**

- Umożliwia automatyczne weryfikowanie, że wszystkie instancje SQL Server spełniają wymagania konfiguracji.

- **Poprawa bezpieczeństwa:**

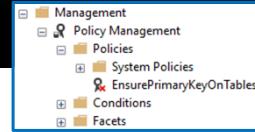
- Możliwość monitorowania zgodności z politykami bezpieczeństwa, np. silne hasła.

- **Redukcja błędów:**

- Automatyczne wykrywanie odchyleń od polityk minimalizuje ryzyko błędów konfiguracyjnych.

80

## Kluczowe pojęcia



- **Polityka (Policy):**
  - Definicja wymagań konfiguracji lub standardów, które mają być spełnione przez obiekty w bazie danych.
- **Kondycja (Condition):**
  - Zbiór warunków, które muszą być spełnione, aby polityka była uznana za zgodną.
- **Czynniki zewnętrzne (Facet):**
  - Grupowanie właściwości SQL Server, które można skonfigurować w ramach polityki.
- **Tryby oceny (Evaluation Modes):**
  - Tryby sprawdzania zgodności, np. na żądanie, podczas zmian, okresowo.

81

## Tworzenie i Konfiguracja Polityk

- **Tworzenie nowej polityki:**
  - Proces tworzenia polityki w SQL Server Management Studio (SSMS).
- **Przykład:**
  - Tworzenie polityki sprawdzającej, czy indeksy są niepofragmentowane.
- **Definiowanie kondycji:**
  - Warunki określające, jakie cechy musi posiadać obiekt.
  - Przykład: „Rozmiar danych nie większy niż 1 GB”
- **Skrypt SQL do tworzenia polityki:**

```
CREATE POLICY CheckIndexFragmentation
ON Facet = 'Index'
FOR Condition = 'NonFragmentedIndex';
```

82

## Monitorowanie zgodności z politykami

- **Tryby oceny polityk:**
  - **On Demand** (Na żądanie) – sprawdzenie zgodności tylko wtedy, gdy jest to wymagane.
  - **On Change: Prevent** (Na zmianę: zapobieganie) – blokuje zmiany naruszające politykę.
  - **On Schedule** (Na harmonogram) – okresowe sprawdzanie zgodności.
- **Przykład monitorowania zgodności z politykami:**
  - Skonfigurowanie harmonogramu do sprawdzania zgodności z politykami bezpieczeństwa bazy danych.

83

## Przykłady Polityk

- **Polityka dla bezpieczeństwa haseł:**
  - Polityka wymuszająca stosowanie złożonych haseł dla kont SQL Server.
- **Polityka dla struktur indeksów:**
  - Sprawdzanie fragmentacji indeksów i zgodności ze standardami struktury.
- **Przykład kodu do oceny polityki:**

```
EXEC sp_syspolicy_check_policy N'PolicyName';
```

84

## Podsumowanie

### Najlepsze praktyki:

- **Regularna weryfikacja polityk:**
  - Okresowe sprawdzanie polityk w celu identyfikacji i rozwiązania niezgodności.
- **Automatyzacja oceny polityk:**
  - Wykorzystanie harmonogramów SQL Server Agent do automatycznego sprawdzania zgodności.
- **Edukacja zespołu:**
  - Szkolenie zespołu dotyczące zgodności i zasad używania polityk.

### Podsumowanie:

- Zarządzanie politykami to skuteczne narzędzie zapewniające zgodność z wymogami organizacyjnymi.
- Regularnie aktualizuj polityki, aby dostosować się do zmieniających się wymagań.

# Normalizacja baz danych

1

## Przykład:

Dostawca

Nazwa Firmy	Adres	Nazwa Produktu	Cena
Exotic Liquids	49 Gilbert St.	Aniseed Syrup	10,00
Exotic Liquids	49 Gilbert St.	Chai	18,00
Exotic Liquids	49 Gilbert St.	Chang	19,00
Forêts d'éables	148 rue Chasseur	Sirop d'éable	28,50
Forêts d'éables	148 rue Chasseur	Tarte au sucre	49,30
Fomaggi Fortini s.r.l.	Viale Dante, 75	Gorgonzola Telino	12,50
Fomaggi Fortini s.r.l.	Viale Dante, 75	Mascarpone Fabioli	32,00
Fomaggi Fortini s.r.l.	Viale Dante, 75	Mozzarella di Giovanni	34,80

- Założymy, że atrybut Nazwa Firmy jest unikalna, tj. nie ma dwóch dostawców o tej samej nazwie.
- Cechy relacji Dostawca:
  - redundancja danych - problem spójności danych
  - anomalia wprowadzania danych
  - anomalia usuwania danych
  - anomalia uaktualniania danych
- Rozwiązaniem: dekompozycja relacji Dostawca na dwie relacje: Dostawca i Dostawy

2

## Normalizacja baz danych

W 1970 E. F. Codd sformułował pewne reguły dotyczące projektowania baz danych. Reguły te zostały pierwotnie wyrażone jako trzy postacie normalne:

- **pierwsza postać normalna,**
- **druga postać normalna,**
- **trzecia postać normalna.**

W połowie lat siedemdziesiątych spostrzeżono pewne braki w trzeciej postaci normalnej i zdefiniowano mocniejszą postać normalną, znaną jako postać **normalna Boyce'a-Codda**. W późniejszym czasie opracowano **czwartą postać normalną**, a nawet **piątą**.

Proces kolejnego przekształcania projektu baz danych przez 1NF, 2NF i 3NF znany jest jako **normalizacja**, które to postacie normalne (1,2,3) są często wystarczające w praktycznych projektach baz danych.

3

## Normalizacja baz danych

### Pierwsza postać normalna 1NF

Relacja R jest w pierwszej postaci normalnej (1NF) wtedy i tylko wtedy, gdy wszystkie użyte dziedziny zawierają tylko atomowe wartości.

### Druga postać normalna 2NF

Relacja R jest w drugiej postaci normalnej (2NF) wtedy i tylko wtedy, gdy jest w postaci 1NF oraz każdy niekluczowy atrybut jest w pełni funkcjonalnie zależny od klucza głównego.

### Trzecia postać normalna 3NF

Relacja R jest w drugiej postaci normalnej (3NF) wtedy i tylko wtedy, gdy jest w postaci 2NF oraz każdy niekluczowy atrybut jest w nietracyjtynie zależny od klucza głównego.

4

## Normalizacja baz danych

### Postać normalna Boyce-Codd (BCNF)

Relacja jest w postaci normalnej Boyce'a-Codda (BCNF) wtedy i tylko wtedy, gdy każdy determinant (zbiór atrybutów po lewej stronie zależności funkcyjnej) jest kluczem kandydującym.

### Czwarta postać normalna 4NF

Relacja R jest w czwartej postaci normalnej (4NF) wtedy i tylko wtedy, gdy w każdym przypadku zależności wielowartościowej w R, powiedzmy  $A \rightarrow\!\!\! \rightarrow B$ , wszystkie atrybuty R są także funkcyjnie zależne od A.

### Piąta postać normalna 5NF

Relacja R jest w piątej postaci normalnej (5NF) wtedy i tylko wtedy, gdy każda zależność złączeniowa w R jest konsekwencją kluczy kandydujących R.

5

## Normalizacja baz danych

- Proces normalizacji relacji można traktować jako proces, podczas którego schematy relacji posiadające pewne niepożądane cechy są dekomponowane na mniejsze schematy relacji o pożądanych własnościach
- Proces normalizacji musi posiadać trzy dodatkowe własności:
  - Własność zachowania atrybutów: żaden atrybut nie zostanie zagubiony w trakcie procesu normalizacji
  - Własność zachowania informacji: dekompozycja relacji nie prowadzi do utraty informacji
  - Własność zachowania zależności: wszystkie zależności funkcyjne są reprezentowane w pojedynczych schematach relacji

6

## Pierwsza postać normalna 1NF

- Definicja:

Schemat relacji R znajduje się w pierwszej postaci normalnej(1NF), jeżeli wartości atrybutów są atomowe (niepodzielne)

Nazwa Firmy	(No column name)
Exotic Liquids	Chang, Chai, Aniseed Syrup

Relacja w 1NF

Nazwa Firmy	productname
Exotic Liquids	Aniseed Syrup
Exotic Liquids	Chai
Exotic Liquids	Chang

Nazwa Firmy	Adres
Exotic Liquids	str.49 Gilbert St London
Forêts d'éables	str.148 rue Chasseur Ste-Hyacinthe
Formaggi Fortini s.r.l.	str.Viale Dante, 75Ravenna

Relacja w 1NF

Nazwa Firmy	ulica	Adres	Miasto
Exotic Liquids	str.	49 Gilbert St.	London
Forêts d'éables	str.	148 rue Chasseur	Ste-Hyacinthe
Formaggi Fortini s.r.l.	str.	Viale Dante, 75	Ravenna

7

## Pierwsza postać normalna 1NF

Wartości atrybutów są elementarne (niepodzielne):

- brak macierzy

- brak list

- brak innych złożonych struktur danych

Tabela: Szkoła	
Klasa	Uczeń
1A	Jan, Tomasz
1B	Marek, Ewa
1C	Anna, Izabela



Tabela: Szkoła	
Klasa	Uczeń
1A	Jan
1A	Tomasz
1B	Marek
1B	Ewa
1C	Anna
1C	Izabela

8

4

## Pierwsza postać normalna 1NF

Tabela: Geografia	
Województwo	Miasto
Mazowieckie	Warszawa, Płock, Radom
Pomorskie	Gdańsk, Gdynia, Sopot

Tabela: Geografia	
Województwo	Miasto
Mazowieckie	Warszawa
Mazowieckie	Płock
Mazowieckie	Radom
Pomorskie	Gdańsk
Pomorskie	Gdynia
Pomorskie	Sopot

9

## Pierwsza postać normalna 1NF

Nazwa Firmy	Adres
Exotic Liquids	str.49 Gilbert St.London
Forêts d'éables	str.148 rue ChasseurSte-Hyacinthe
Fomaggi Fortini s.r.l.	str.Viale Dante, 75Ravenna

Nazwa Firmy	ulica	Adres	Miasto
Exotic Liquids	str.	49 Gilbert St.	London
Forêts d'éables	str.	148 rue Chasseur	Ste-Hyacinthe
Fomaggi Fortini s.r.l.	str.	Viale Dante, 75	Ravenna

10

## Druga postać normalna 2NF

- atrybuty w tabeli powinny być zależnie funkcyjnie od klucza głównego
- tabele powinny opisywać konkretny obiekt

Tabela: Zamówienie									
Nr_Fakt	Data_Fakt	Nr_Klienta	Klient	Woj_Klienta	Miasto_Klienta	Brutto	Vat	Netto	
1234	2015-09-11	1	Nowak	Mazowieckie	Warszawa	1500	23%	1155	
5678	2015-09-24	2	Kowal	Łódzkie	Łódź	4800	8%	4416	
9012	2015-10-05	3	Bąk	Wielkopolskie	Poznań	2300	23%	1771	
3456	2015-10-15	1	Nowak	Mazowieckie	Warszawa	3700	8%	3404	

11

## Druga postać normalna 2NF

Tabela: Zamówienie					Tabela: Klient			
Nr_Fakt	Data_Fakt	Nr_Klienta	Brutto	Vat	Nr_Klienta	Klient	Woj_Klienta	Miasto_Klienta
1234	2015-09-11	1	1500	23%	1155	Nowak	Mazowieckie	Warszawa
5678	2015-09-24	2	4800	8%	4416	Kowal	Łódzkie	Łódź
9012	2015-10-05	3	2300	23%	1771	Bąk	Wielkopolskie	Poznań
3456	2015-10-15	1	3700	8%	3404			



Druga postać normalna (2PN)				
Tabela: Zamówienie				
Nr_Fakt	Data_Fakt	Nr_Klienta	Brutto	Vat
1234	2015-09-11	1	1500	23%
5678	2015-09-24	2	4800	8%
9012	2015-10-05	3	2300	23%
3456	2015-10-15	1	3700	8%

Tabela: Klient		
Nr_Klienta	Klient	Miasto_Klienta
1	Nowak	Warszawa
2	Kowal	Łódź
3	Bąk	Poznań

Tabela: Miasto_Województwo	
Miasto	Województwo
Warszawa	Mazowieckie
Łódź	Łódzkie
Poznań	Wielkopolskie

12

6

## Trzecia postać normalna 3NF

- każdy atrybut jest funkcjonalnie zależny jedynie od klucza głównego, czyli nie mogą istnieć jakiekolwiek zależności przekształcające

### Trzecia postać normalna (3PN)

Tabela: Zamówienie

Nr_Fakt	Data_Fakt	Nr_Klienta	Brutto	Vat	Netto
1234	2015-09-11	1	1500	23%	1155
5678	2015-09-24	2	4800	8%	4416
9012	2015-10-05	3	2300	23%	1771
3456	2015-10-15	1	3700	8%	3404

Tabela: Klient

Nr_Klienta	Klient	Miasto_Klienta
1	Nowak	Warszawa
2	Kowal	Łódź
3	Bąk	Poznań

Tabela: Miasto\_Województwo

Miasto	Województwo
Warszawa	Mazowieckie
Łódź	Łódzkie
Poznań	Wielkopolskie

13

## Normalizacja

Tabela: Zamówienia

Nr_Fakt	Data_Fakt	Nr_Klienta	Netto	VAT
1234	2015-09-11	1	1155	23%
5678	2015-09-24	2	4416	8%
9012	2015-10-05	3	1771	23%
3456	2015-10-15	1	3404	8%

Tabela: Klient

Nr_Klienta	Klient	Miasto_ID
1	Nowak	1
2	Kowal	2
3	Bąk	3

Tabela: Miasto

Miasto_ID	Miasto	Województwo_ID
1	Warszawa	1
2	Łódź	2
3	Poznań	3

Tabela: Województwo

Województwo_ID	Województwo
1	Mazowieckie
2	Łódzkie
3	Wielkopolskie

14

## **Relacyjny model danych**

- Relacyjny model danych
- Struktury danych
- Operacje
- Ograniczenia integralnościowe

## **Model danych**

- **Model danych** definiuje
  - struktury danych
  - operacje
  - ograniczenia integralnościowe
- **Relacyjny model danych** definiuje
  - relacje
  - selekcja, projekcja, połączenie, operacje na zbiorach
    - klucz podstawowy, klucz obcy, zawężenie dziedziny, unikalność, wartość pusta/niepusta

## **Modelowanie danych**

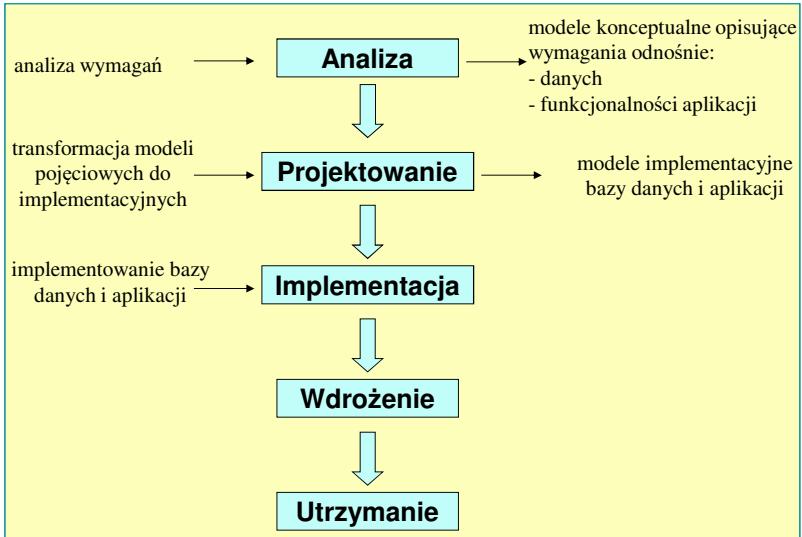
### **Model związków-encji**

- Wprowadzenie do modelowania i projektowania systemów informatycznych
- Model związków-encji
  - encje
  - atrybuty encji
  - związki pomiędzy encjami
  - hierarchia generalizacji

## **Modelowanie - modele**

- Modelowanie - odwzorowanie rzeczywistych obiektów świata rzeczywistego w systemie informatycznym (bazie danych)
- Modele
  - konceptualne (reprezentacja obiektów w uniwersalnym modelu niezależnym od modelu implementacyjnego)
    - model związków-encji
    - model UML
  - implementacyjne
    - modele wykorzystywane do implementacji modeli konceptualnych
    - modele danych (relacyjne, obiektowe, itp.)

## Cykl projektowy systemu informatycznego



## Model związków-encji

• **Model związków-encji**  
**(entity-relationship model - ER)**

- obiekty świata rzeczywistego reprezentowane za pomocą encji (entities)
- powiązania między obiektami świata rzeczywistego reprezentowane za pomocą związków (relationships) pomiędzy encjami

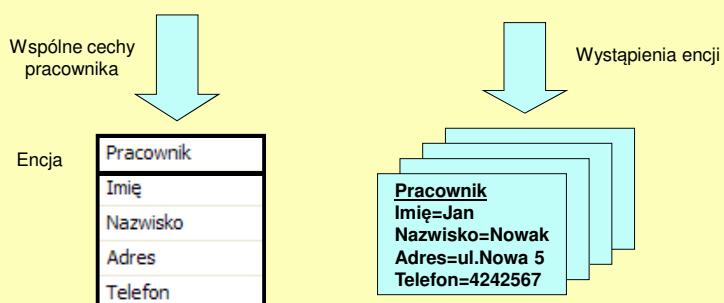
## Encja

- Reprezentuje zbiór obiektów opisany tymi samymi cechami (atrybutami, własnościami)
- Informacje o tych obiektach będą przechowywane w bazie danych
- Konkretny obiekt świata rzeczywistego jest reprezentowany jako wystąpienie encji (instancję encji)

## Modelowanie encji

### Obiekty świata rzeczywistego

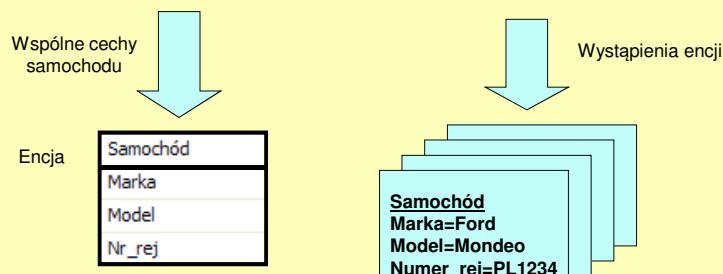
Firma zatrudnia pracowników. Chcemy przechowywać informacje nt. danych personalnych pracowników (imię, nazwisko, adres i numer telefonu).



## Modelowanie encji

### Obiekty świata rzeczywistego

Parking firmy jest przeznaczony do parkowania wielu różnych samochodów. Chcemy przechowywać informacje o samochodach (marka, model, numer rejestracyjny), które mogą parkować na parkingu firmy.



## Modelowanie encji

- Każda encja posiada
  - unikalną nazwę
  - zbiór cech (atrybutów)
- Encje wchodzą w związki z innymi encjami
  - wyjątkiem są encje reprezentujące dane słownikowe i konfiguracyjne
- Dowolna rzecz lub obiekt może być reprezentowana tylko przez jedną encję
- Nazwa encji powinna być rzeczownikiem w liczbie pojedynczej

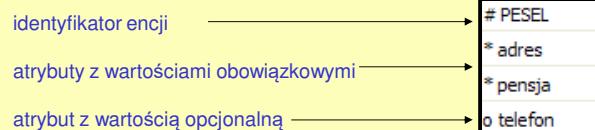
## Atrybuty encji

- Identyfikator
  - atrybut lub zbiór atrybutów jednoznacznie identyfikujący wystąpienie encji
  - zbiór atrybutów + związki
  - związki
- Identyfikatory naturalne
  - PESEL, NIP, nr dowodu, nr paszportu, nr rejestracyjny, ISBN
- Identyfikatory sztuczne
  - numer pozycji katalogowej, identyfikator pracownika
- Deskryptory (atrybuty deskrypcyjne)
  - wszystkie inne atrybuty poza identyfikatorami
  - reprezentują podstawowe cechy/właściwości encji
  - cechy te będą przechowywane w bazie danych
  - atrybuty z wartościami opcjonalnymi
  - atrybuty z wartościami obowiązkowymi

## Definicja atrybutu encji

- Nazwa
  - Dziedzina
    - typ danych i maksymalny rozmiar
    - zbiór dozwolonych wartości
    - zakres dozwolonych wartości
  - Dozwolone / niedozwolone wartości puste
  - Opcjonalnie unikalność wartości
- } ograniczenia integralnościowe

### Przykład



## Związek

- Związek (asocjacja) reprezentuje powiązania pomiędzy obiektami świata rzeczywistego
  - klienci posiadają rachunki bankowe
  - studenci otrzymują oceny z egzaminów
- W modelu ER związek łączy encje
- Związek z każdego końca posiada krótki opis ułatwiający interpretację związku

## Modelowanie związków



- Istnieje związek pomiędzy pracownikami a samochodami
- Chcemy wiedzieć:
  - Ile samochodów może posiadać pracownik?
  - Ile pracowników może posiadać ten sam samochód?
  - Czy każdy samochód musi do kogoś należeć?
  - Czy każdy pracownik musi posiadać samochód?

## Cechy związku

- **Stopień związku**

- unarny (binarny rekursywny)
- binarny
- ternarny
- n-arny

- **Typ asocjacji (kardynalność)**

- jeden-do-jeden (1:1)
- jeden-do-wiele (1:M)
- wiele-do-wiele (M:N)

- **Istnienie (klasa przynależności)**

- opcjonalny
- obowiązkowy

## Cechy związku - przykład

- Pracownicy firmy posiadają samochody

związek Pracownik-Samochód  
stopień związku: binarny

- W celu udostępnienia miejsca parkingowego należy zarejestrować pracownika i jego samochód

typ asocjacji  
Pracownik (1) : Samochód (1)

- Każdy pracownik ma prawo parkować tylko jeden konkretny samochód

istnienie  
Pracownik może posiadać

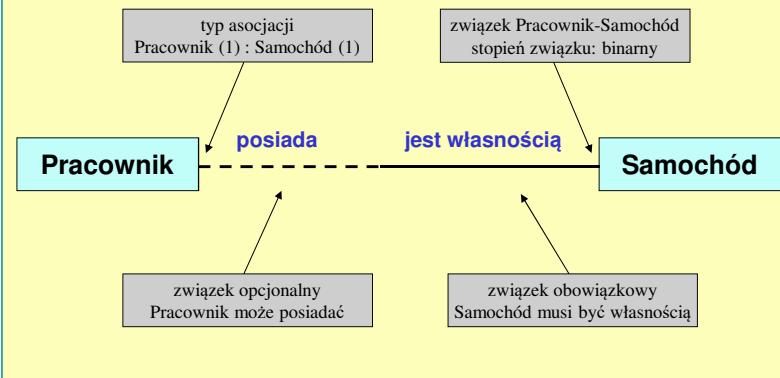
- Nie każdy pracownik ma samochód
- Zarejestrowany w rejestrze parkingowym samochód na pewno jest własnością jednego pracownika

typ asocjacji  
Pracownik (1) : Samochód (1)

istnienie  
Samochód musi być własnością

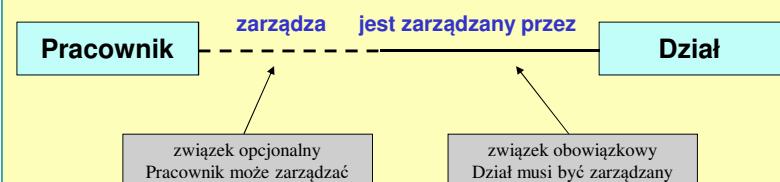
## Cechy związku - przykład

- Związek binarny (łączy dwie encje)
- Związek opcjonalny od strony pracownika (linia przerywana)
- Związek obowiązkowy od strony samochodu (linia ciągła)
- Związek 1:1 (1 pracownik posiada 1 samochód)



## Typ asocjacji 1:1

Związek binarny jeden-do-jeden (1:1)

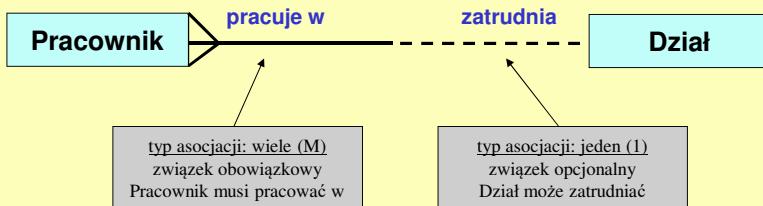


### Interpretacja

- pracownik może być kierownikiem tylko jednego działu
  - \* istnieją pracownicy, którzy nie kierują żadnym działem
- każdy dział musi być kierowany przez dokładnie jednego pracownika

## Typ asocjacji 1:M

Związek binarny typu jeden-do-wiele (1:M)

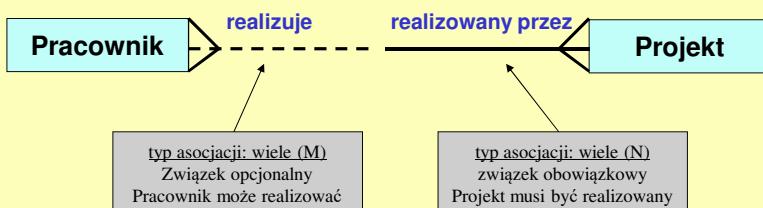


### Interpretacja

- każdy pracownik musi pracować w jakimś dziale
  - w jednym dziale pracuje jeden lub wielu pracowników
  - dział może zatrudniać pracowników
- \* istnieją działy, które nie zatrudniają pracowników

## Typ asocjacji M:N

Związek binarny typu wiele-do-wiele (M:N)



### Interpretacja

- pracownik może brać udział w projekcie
- \* istnieją pracownicy nie biorący udziału w żadnym projekcie
- projekt musi być realizowany przez przynajmniej jednego pracownika
- w tym samym projekcie może brać udział wielu pracowników

## Atrybuty związku

Związek binarny typu wiele-do-wiele (M:N)



- Jeśli związek posiada dodatkowe cechy → należy wprowadzić dodatkową encję (Realizacja)
- Do encji tej dochodzą obowiązkowe związki typu wiele – interpretacja obowiązkowości związków
  - \* jeśli istnieje wystąpienie encji Realizacja, to musi ono dotyczyć jakiegoś projektu i pracownika
  - \* nie może istnieć realizacja bez pracownika i projektu

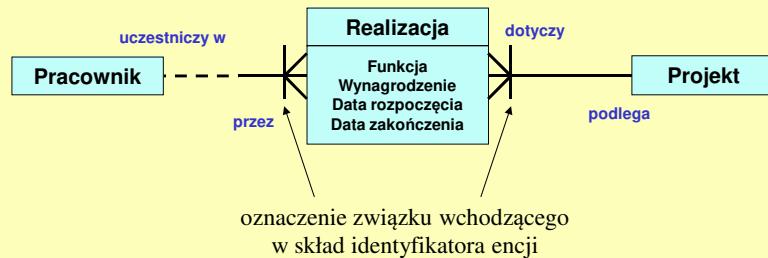
## Encja słaba

- Encja słaba (weak entity)
  - nie posiada swojego identyfikatora
  - wystąpienia encji mogą istnieć tylko w kontekście wystąpień encji powiązanych z encją słabą
  - konkretne wystąpienie encji Realizacja może wystąpić wyłącznie w kontekście konkretnego pracownika i konkretnego projektu



## Identyfikator encji słabej

- Identyfikatorem encji słabej są wszystkie związki, w które wchodzi ta encja



## Związek binarny rekursywny

- Określa powiązanie pomiędzy wystąpieniem encji a innym wystąpieniem tej samej encji
- Modelowanie zależności służbowych
- Modelowanie elementów złożonych

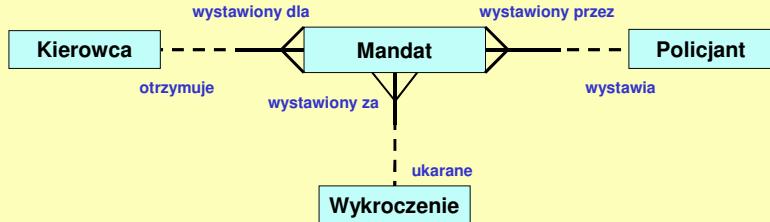
### Związek 1:M opcjonalny z obu stron.



Tego typu związek modelujący zależności hierarchiczne musi być opcjonalnym z obu stron. W przeciwnym przypadku, powstałyby hierarchia nieskończona.

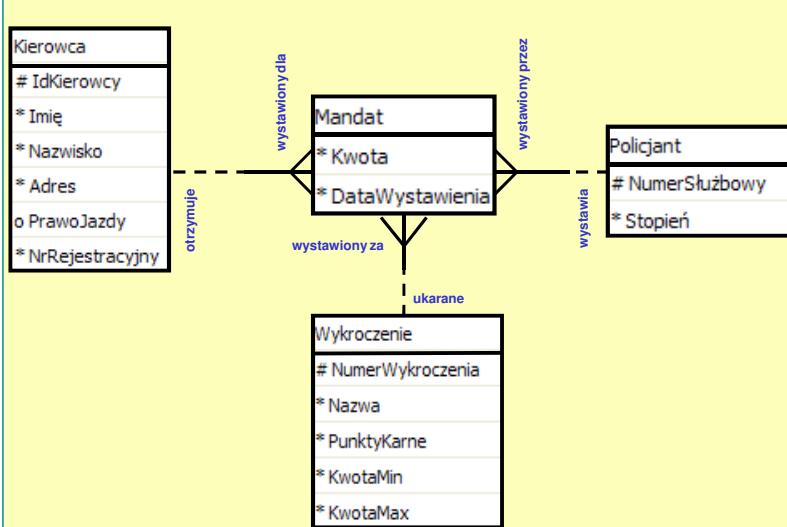
## Związki ternarne

Związek ternarny łączy 3 encje.  
"Kierowca możetrzymać mandat za popełnione wykroczenie.  
Mandat jest wystawiany przez konkretnego policjanta."



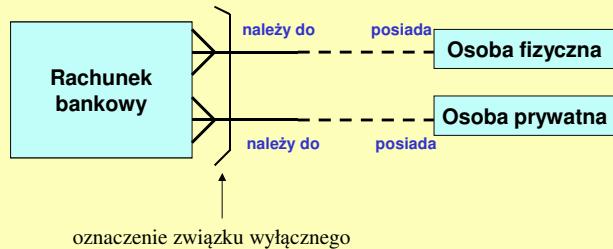
Związek ternarny jest reprezentowany jako encja (Mandat)  
– do encji Mandat dochodzą związki obowiązkowe  
\* jeśli wystawiono mandat to jest on dla konkretnej osoby,  
został wystawiony przez konkretnego policjanta i dotyczy  
konkretnego wykroczenia.

## Związki ternarne



## Związki wyłączne

- Związki wyłączne (**exclusive relationships**)
  - konkretne wystąpienie encji może w danym momencie wchodzić tylko w jeden z ze związków
  - Np. [Rachunek bankowy] wchodzi w związek z encją [Osoba fizyczna] lub [Osoba prawna].



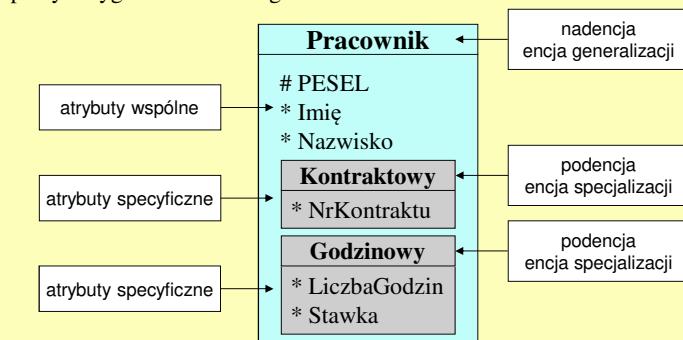
## Hierarchia encji / generalizacja

- Związek generalizacji
  - określa, że pewne encje o wspólnym zbiorze atrybutów można uogólnić i stworzyć encję wyższego poziomu → encję generalizacji
- Encje niższego poziomu w hierarchii generalizacji → encje specjalizacji
- Relacja opisująca związki typu generalizacja/specjalizacja pomiędzy encjami → hierarchia generalizacji/specjalizacji lub hierarchia encji

## Hierarchia encji

### Dziedziczenie atrybutów

Firma zatrudnia pracowników kontraktowych i godzinowych. Wszyscy pracownicy posiadają pewien zbiór wspólnych atrybutów (PESEL, imię, nazwisko, adres). Pracownicy kontraktowi i godzinowi posiadają specyficzne dla siebie atrybuty. Dla pracowników kontraktowych jest to numer kontraktu, a dla pracowników godzinowych są to: liczba godzin pracy w tygodniu i stawka godzinowa.



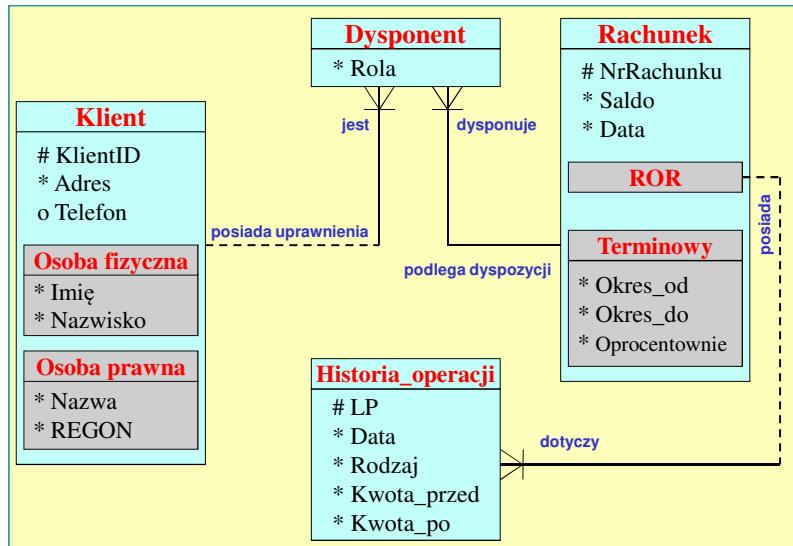
## Hierarchia encji

### Interpretacja

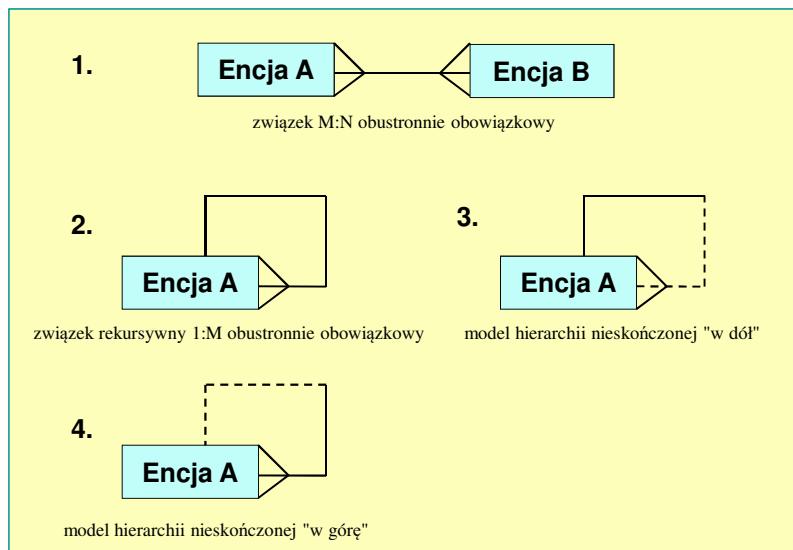
- podencje dziedziczą wszystkie atrybuty swojej nadencji
- każde wystąpienie nadencji jest zawsze wystąpieniem jednej podencji
- semantyka związku generalizacji oznacza, że każde wystąpienie podencji JEST wystąpieniem nadencji
  - \* pracownik kontraktowy JEST pracownikiem
  - \* pracownik godzinowy JEST pracownikiem
- identyfikator nadencji jest wspólny dla wszystkich jej podencji
- podencje nie posiadają swoich identyfikatorów



## Hierarchia encji



## Związki niedozwolone



## Transformacja modelu ER do modelu relacyjnego

- Transformacja encji
- Transformacja związków
- Transformacja hierarchii encji

## Pojęcia podstawowe

- **Schemat bazy danych**
  - zbiór schematów relacji
- **Relacja (tabela)**
  - dwu-wymiarowa tablica
  - kolumny → atrybuty
  - wiersze → krotki, rekordy
    - \* każda krotka reprezentuje wystąpienie encji
- **Klucz podstawowy**
  - atrybut lub zbiór atrybutów - wybrany spośród kluczy potencjalnych
- **Klucz obcy**
  - atrybut lub zbiór atrybutów wskazujący na klucz podstawowy innej relacji
    - \* atrybut lub zbiór atrybutów w relacji B, będący jednocześnie kluczem podstawowym w relacji A
    - należy zaznaczyć, że klucz obcy może odnosić się do klucza podstawowego samej relacji, w której został on zdefiniowany

## Transformacja

- Model ER → schemat relacyjny
- Transformacja
  - encji z atrybutami
  - związków
  - hierarchii encji

## Reguły transformacji encji

- Encja → relacja
- Atrybut encji → atrybut relacji
- Typ danych atrybutu encji → typ danych atrybutu relacji
- Identyfikator encji → klucz podstawowy relacji
- Obowiązkowość atrybutów encji → ograniczenie NOT NULL
- Opcjonalność atrybutów encji → ograniczenie NULL
- Pozostałe ograniczenia integralnościowe atrybutów encji → ograniczenia integralnościowe atrybutów relacji

Pracownik
# PESEL
* adres
* pensja
o telefon

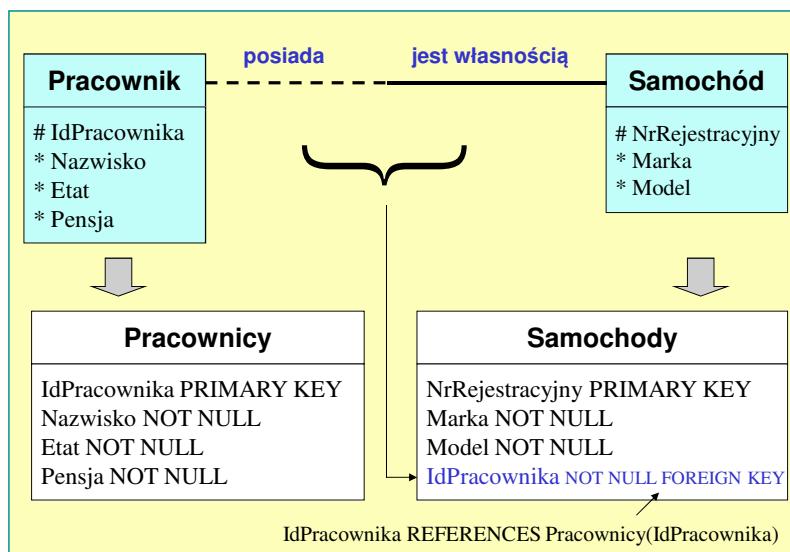


Pracownicy (  
PESEL PRIMARY KEY,  
adres NOT NULL,  
pensja NOT NULL,  
telefon NULL )

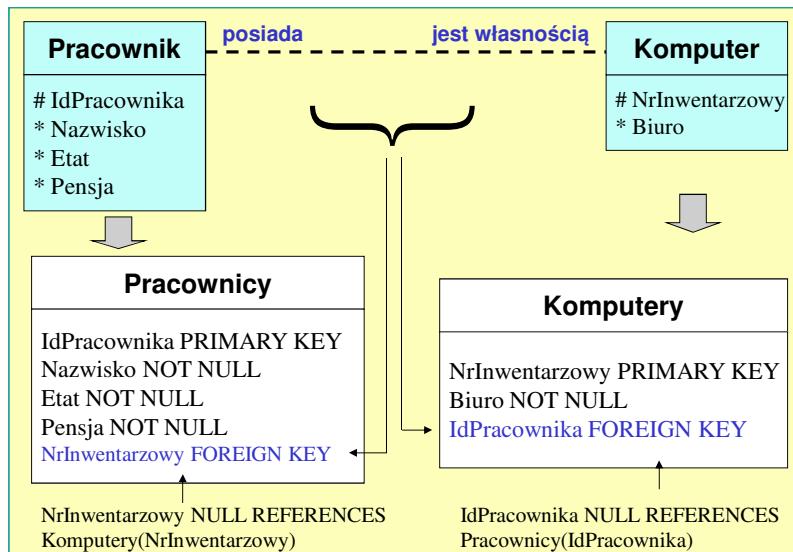
## Reguły transformacji związków

- Związek binarny 1:1 → klucz obcy we wskazanej tabeli
- Związek unarny 1:1 → klucz obcy w tej samej tabeli
- Związek binarny 1:M → klucz obcy w tabeli po stronie "wiele"
- Związek binarny M:N → tabela
- Związek unarny M:N → tabela

## Związek binarny 1:1 jednostronnie obowiązkowy



## Związek binarny 1:1 obustronne opcjonalny



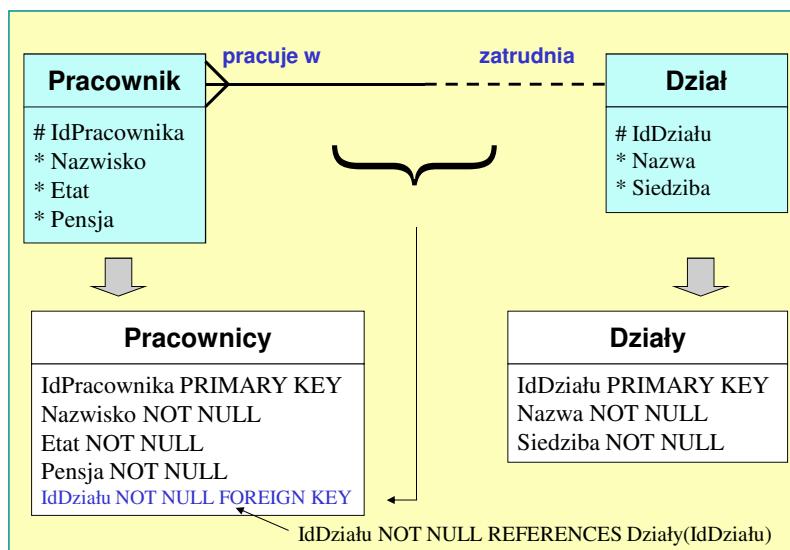
## Związek binarny 1:1

rzadko stosowany	<b>Pracownicy</b>	<b>Komputery</b>
1.	<b>Pracownicy</b> IdPracownika PRIMARY KEY ... NrInwentarzowy FOREIGN KEY	<b>Komputery</b> NrInwentarzowy PRIMARY KEY ... IdPracownika FOREIGN KEY
2.	<b>Pracownicy</b> IdPracownika PRIMARY KEY ...	<b>Komputery</b> NrInwentarzowy PRIMARY KEY ... IdPracownika FOREIGN KEY
3.	<b>Pracownicy</b> IdPracownika PRIMARY KEY ... NrInwentarzowy FOREIGN KEY	<b>Komputery</b> NrInwentarzowy PRIMARY KEY ...

## Związek 1:M

- Klucz obcy dodawany do relacji po stronie "wiele"
- Ograniczenia referencyjne definiowane dla klucza obcego
- Obowiązkowość związku po stronie "wiele" → ograniczenie NOT NULL definiowane na kluczu obcym
- Opcjonalność związku po stronie "wiele" → ograniczenie NULL definiowaną na kluczu obcym relacji
- Opcjonalność lub obowiązkowość związku po stronie "jeden" nie jest odwzorowywana w modelu relacyjnym

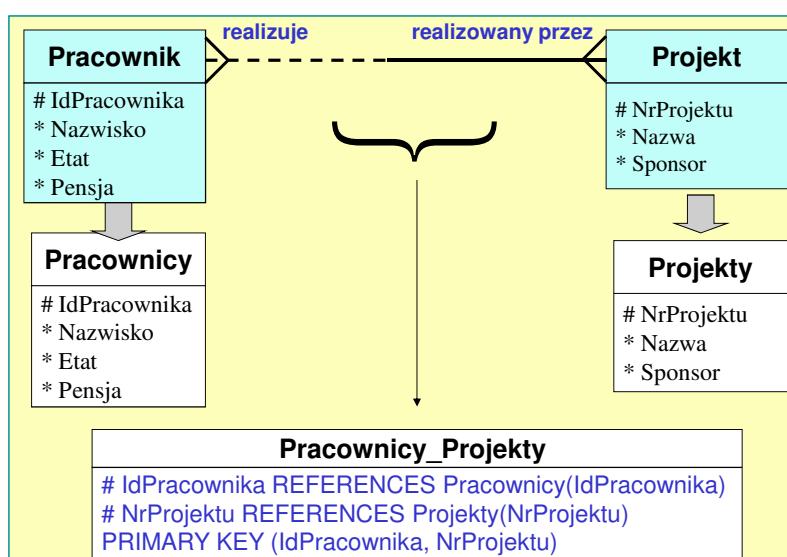
## Związek binarny 1:M jednostronnie obowiązkowy



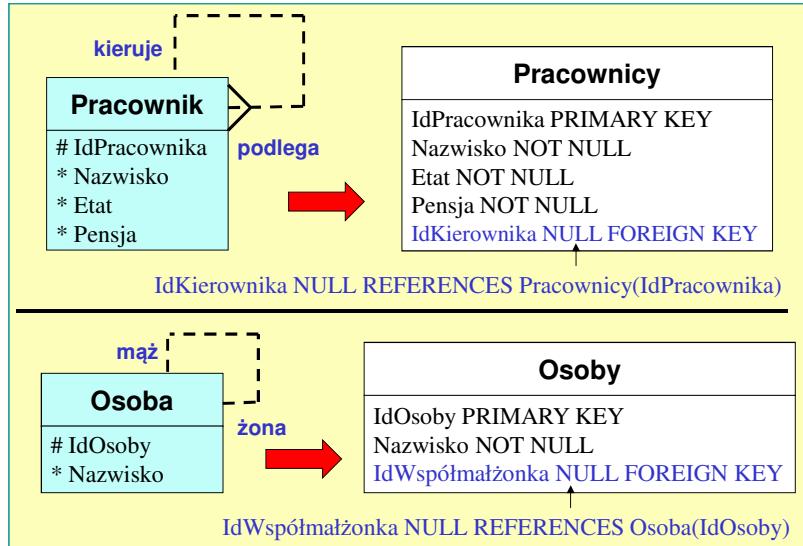
## Związek M:N

- Reprezentowany poprzez dodatkową relację
- Nazwa relacji jest złączeniem nazw relacji z powstałych encji
- Relacja zawiera klucze obce wskazujące na klucze podstawowe relacji powstały z powiązanych encji
- Ograniczenia referencyjne definiowane dla kluczy obcych
- Klucze obce tworzą klucz podstawowy relacji

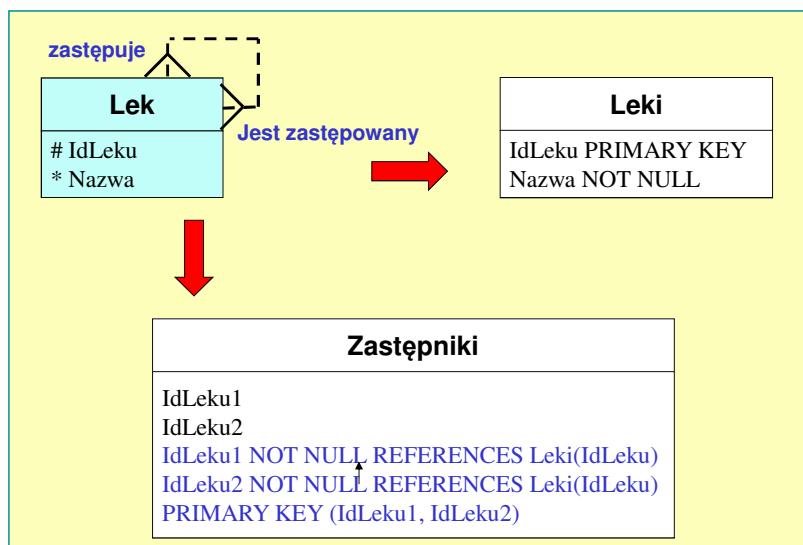
## Związek binarny M:N jednostronne obowiązkowy



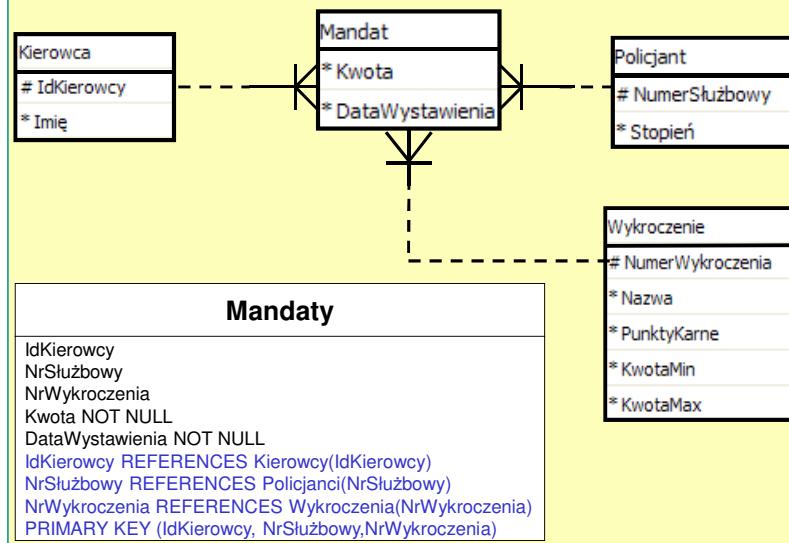
### Związek unary obustronne opcjonalny



### Związek unary obustronne opcjonalny



## Związki ternarne



## Normalizacja baz danych

## Przykład

Dostawca		Nazwa Produktu	Cena
Nazwa Firmy	Adres		
Exotic Liquids	49 Gilbert St.	Aniseed Syrup	10,00
Exotic Liquids	49 Gilbert St.	Chai	18,00
Exotic Liquids	49 Gilbert St.	Chang	19,00
Forêts d'éables	148 rue Chasseur	Sirop d'éable	28,50
Forêts d'éables	148 rue Chasseur	Tarte au sucre	49,30
Formaggi Fortini s.r.l.	Viale Dante, 75	Gorgonzola Telino	12,50
Formaggi Fortini s.r.l.	Viale Dante, 75	Mascarpone Fabioli	32,00
Formaggi Fortini s.r.l.	Viale Dante, 75	Mozzarella di Giovanni	34,80

- Założymy, że atrybut **Nazwa Firmy** jest unikalna, tj. nie ma dwóch dostawców o tym samym nazwisku.
- Cechy relacji **Dostawca**:
  - redundancja danych - problem spójności danych
  - anomalia wprowadzania danych
  - anomalia usuwania danych
  - anomalia uaktualniania danych
- Rozwiązaniem: dekompozycja relacji Dostawca na dwie relacje: Dostawca i Dostawy

## Przykład

Dostawca		Dostawy	
Nazwa Firmy	Adres	Nazwa Produktu	Cena
Exotic Liquids	49 Gilbert St.	Aniseed Syrup	10,00
Exotic Liquids	49 Gilbert St.	Chai	18,00
Exotic Liquids	49 Gilbert St.	Chang	19,00
Forêts d'éables	148 rue Chasseur	Sirop d'éable	28,50
Forêts d'éables	148 rue Chasseur	Tarte au sucre	49,30
Formaggi Fortini s.r.l.	Viale Dante, 75	Gorgonzola Telino	12,50
Formaggi Fortini s.r.l.	Viale Dante, 75	Mascarpone Fabioli	32,00
Formaggi Fortini s.r.l.	Viale Dante, 75	Mozzarella di Giovanni	34,80

### • Zależność funkcyjna (FD)

Dana jest relacja  $r$  o schemacie  $R$ .  $X, Y$  są podzbiorami atrybutów  $R$ . W schemacie relacji  $R$ ,  $X$  wyznacza funkcyjnie  $Y$ , lub  $Y$  jest funkcyjnie zależny od  $X$ , co zapisujemy  $X \rightarrow Y$ , wtedy i tylko wtedy, jeżeli dla dwóch dowolnych krotek  $t1, t2$  takich, że  $t1[X] = t2[X]$  zachodzi zawsze  $t1[Y] = t2[Y]$ , gdzie  $t_i[A]$  oznacza wartość atrybutu

A krotki  $t_i$

### • Przykłady:

- 1. Nazwa Firmy  $\rightarrow$  Adres
- 2. {Nazwa Firmy, Nazwa Produktu}  $\rightarrow$  Cena

## Normalizacja baz danych

- Proces normalizacji relacji można traktować jako proces, podczas którego schematy relacji posiadające pewne niepożądane cechy są dekomponowane na mniejsze schematy relacji o pożądanych własnościach
- Proces normalizacji musi posiadać trzy dodatkowe własności:
  - **Własność zachowania atrybutów** - żaden atrybut nie zostanie zagubiony w trakcie procesu normalizacji
  - **Własność zachowania informacji** - dekompozycja relacji nie prowadzi do utraty informacji
  - **Własność zachowania zależności** - wszystkie zależności funkcyjne są reprezentowane w pojedynczych schematach relacji

## Normalizacja baz danych

### Pierwsza postać normalna 1NF

„Relacja R jest w pierwszej postaci normalnej (1NF) wtedy i tylko wtedy, gdy wszystkie użyte dziedziny zawierają tylko atomowe wartości.”

### Druga postać normalna 2NF

„Relacja R jest w drugiej postaci normalnej (2NF) wtedy i tylko wtedy, gdy jest w postaci 1NF oraz każdy niekluczowy atrybut jest w pełni funkcyjnie zależny od klucza głównego.”

### Trzecia postać normalna 3NF

„Relacja R jest w drugiej postaci normalnej (3NF) wtedy i tylko wtedy, gdy jest w postaci 2NF oraz każdy niekluczowy atrybut jest w nietracyjtynie zależny od klucza głównego.”