



roclawska

Politechnika Wroclawska

Bazy Danych

dr inż. Roman Ptak

Katedra Informatyki Technicznej

roman.ptak@pwr.edu.pl



roclawska

Politechnika Wroclawska

Techniki organizacji danych

Wykład 2



Plan wykładu 2.

- Techniki organizacji danych
- Struktura fizyczna systemu
- Struktura logiczna systemu
- Przykład Microsoft SQL Server
- Inne systemy baz danych: MySQL, SQLite, PostgreSQL
- Hurtownie danych
- Fragmentacja danych



roclawska

Politechnika Wroclawska

Przechowywanie danych



Przechowywanie danych

- W jaki sposób w systemie bazodanowym przechowuje się duże ilości danych?
- W jaki sposób zarządzać zasobami?



Hierarchia pamięci

- Wiele różnych urządzeń do przechowywania danych
- Różne wartości parametrów:
 - szybkość
 - pojemność
 - koszt



Rodzaje pamięci

- Pamięć podręczna
- Pamięć operacyjna (RAM)
- Pamięć pomocnicza (dyski)
- Pamięć trzeciego rzędu

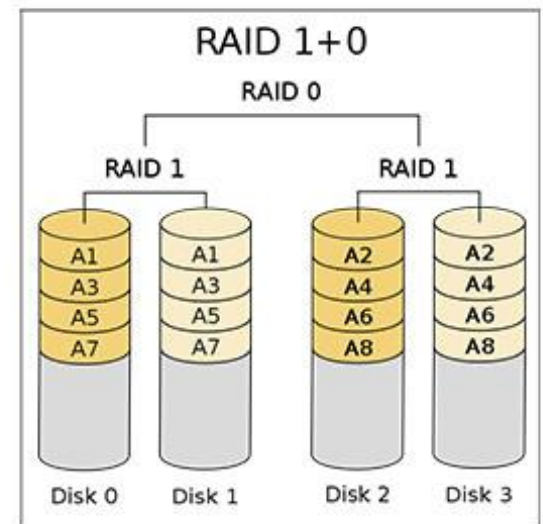
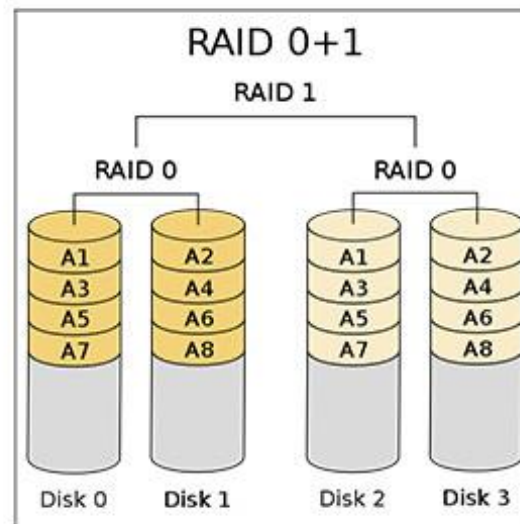
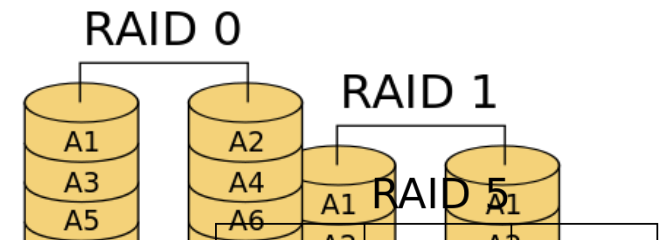


Pamięć pomocnicza

- Dyski HDD (ang. *Hard Disk Drive*)
 - do 16 TB
- Dyski SSD (ang. *Solid-State Drive*)

Macierze dyskowe (RAID)

- RAID 0 (stripping)
- RAID 1 (lustrzany)
- RAID 2
- RAID 3
- RAID 4
- RAID 5
- RAID 6
- RAID 0+1
- RAID 1+0





RAID

Poziom RAID	Minimalna liczba dysków (N)	Dostępna przestrzeń	Max liczba dysków, które mogą ulec awarii
0	2	N	0
1	2	1	N-1
2	3	$N - \log N$	1
3	3	N-1	1
4	3	N-1	1
5	3	N-1	1
6	4	N-2	2
0+1	4	zależnie od konfiguracji	zależnie od konfiguracji
1+0	4	zależnie od konfiguracji	zależnie od konfiguracji

Pamięć trzeciego rzędu

- Kasety z taśmami (ang. streamery)
- Silosy taśmowe
- „szafry grające” (ang. Jukebox) - dyski optyczne





roclawska

Politechnika Wroclawska

Microsoft SQL Server

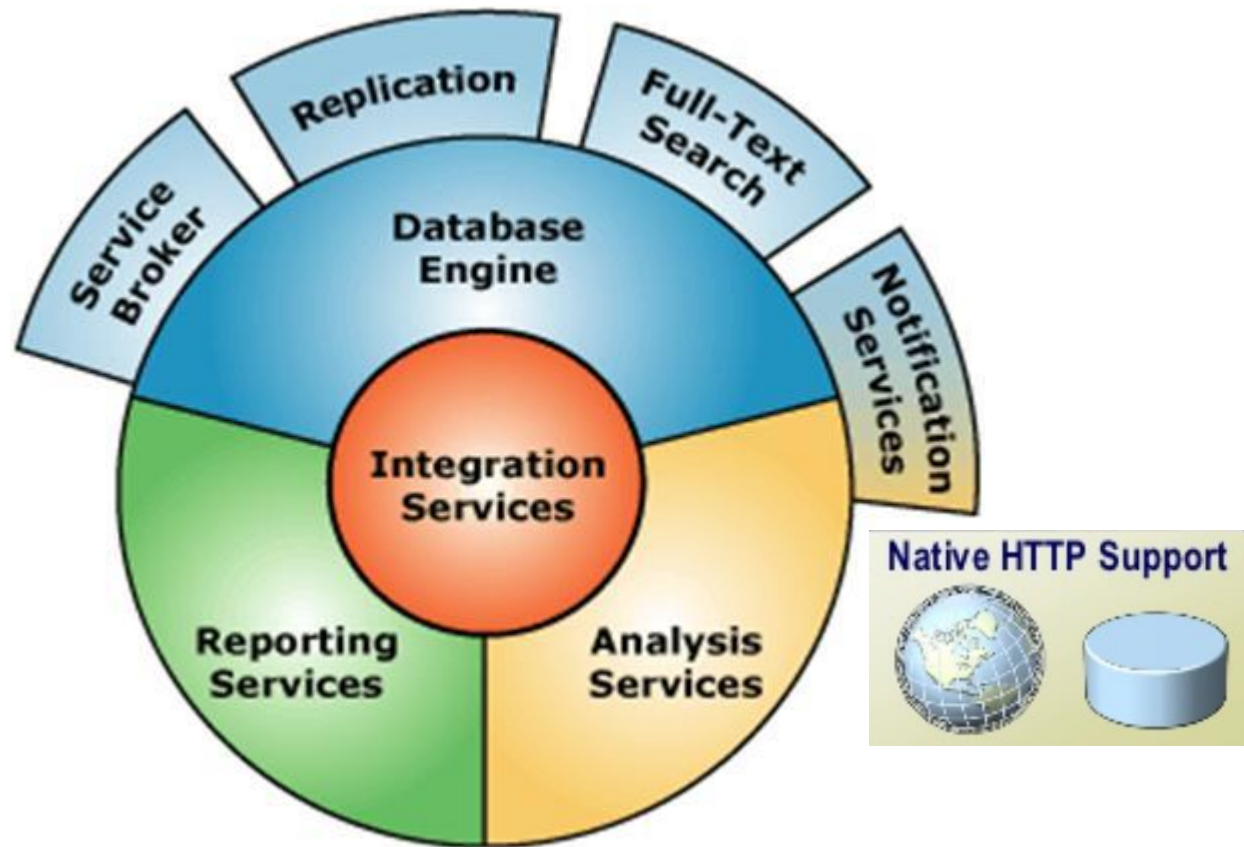


Microsoft SQL Server

- Producent: Microsoft
- Licencja: Microsoft EULA
- Najnowsza wersja stabilna: 2019
- Na serwerze MS SQL Server jedna instalacja (instancja) serwera zawiera wiele baz danych.
- Bazy danych dzielą się na systemowe (jak np. *master*) i użytkownika.



Architektura MS SQL Server





Główne składniki silnika bazy danych SQL Server

Protokoły

Część relacyjna

- Parser, Optymalizator, Menadżer SQL
- Menadżer baz danych, Moduł wykonujący kwerendy, ...

Interfejs OLE DB

Aparat składowania danych

- Menadżer transakcji, Menadżer blokad
- Menadżer indeksów, Menadżer buforowania, ...

SQLOS API

- Menadżer Wejścia/Wyjścia



MS SQL Server

- Struktura logiczna
- Struktura fizyczna



STRUKTURA LOGICZNA SYSTEMU MS SQL SERVER



Struktura logiczna systemu:

- Tabele
- Widoki
- Indeksy
- Procedury składowane
- Wyzwalacze
 - typu AFTER - po operacji DML
 - typu BEFORE - przed operacją DML
 - typu INSTEAD of - zamiast operacji DML



STRUKTURA FIZYCZNA SYSTEMU MS SQL SERVER



Struktura fizyczna - pliki:

- MDF (ang. *main data file, master data file*) - główny plik bazy danych
- NDF (ang. *next data file*) - pliki pomocnicze (grupa plików)
- LDF (ang. *log data file*) - plik dziennika powtórzeń (logi)



Pliki danych (MDF)

- Plik główny danych (ang. *primary file*) stanowi punkt startowy bazy danych i zawiera informacje o innych plikach w bazie danych.
- Każda baza danych ma dokładnie jeden plik główny danych - ma on zwykle rozszerzenie .mdf.

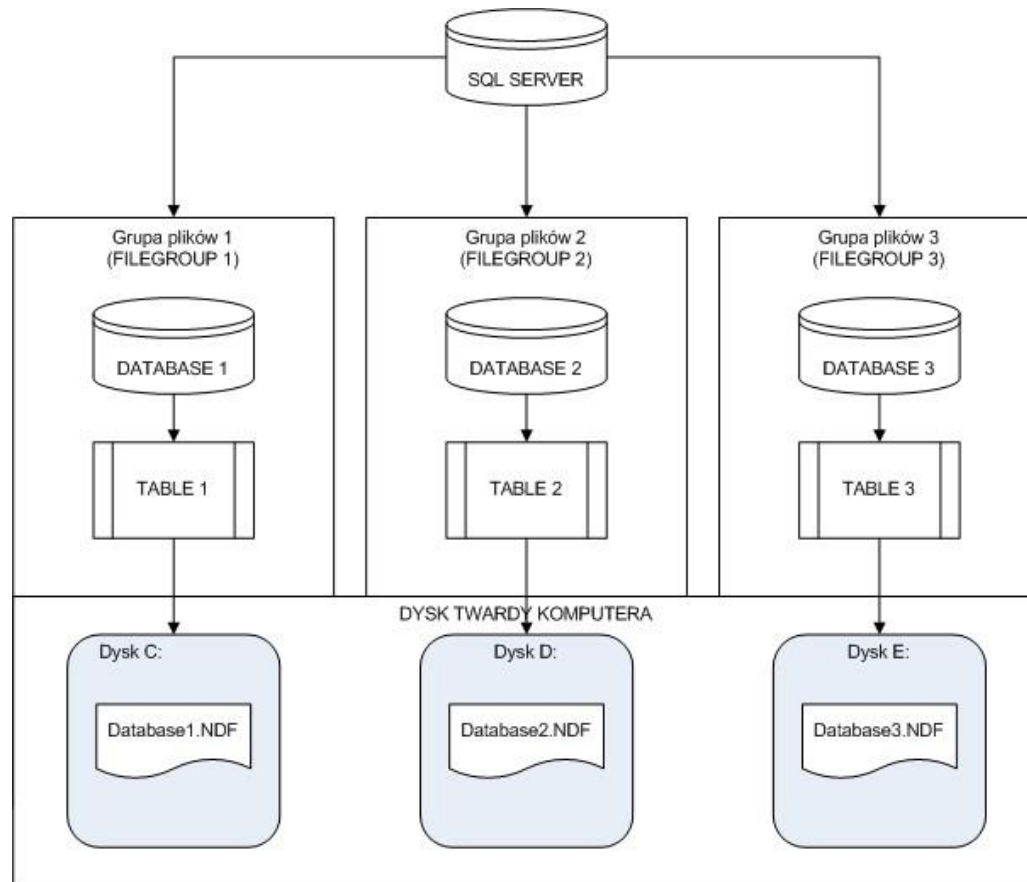


Pliki pomocnicze (NDF)

- Partycjonowanie danych
- Artykuł:
<https://msdn.microsoft.com/pl-pl/library/tworzenie-bazy-danych-w-sql-server.aspx>

The screenshot shows the Microsoft Developer Network (MSDN) website interface. At the top, there is a navigation bar with the Microsoft logo, 'Developer Network', and links for 'Zaloguj się', 'Subskrypcje MSDN', and 'Pobierz narzędzia'. Below this is a secondary navigation bar with categories like 'Technologie', 'Materiały do pobrania', 'Programy', 'Społeczność', 'Dokumentacja', and 'Przykłady'. The main content area displays a breadcrumb trail: 'Publikacje techniczne > Aplikacje bazodanowe > Bazy danych'. The article title is 'Tworzenie bazy danych w SQL Server - czyli na co zwracać uwagę podczas tworzenia i konfigurowania bazy danych'. The author is listed as 'Autor: Paweł Wilkosz' and the publication date is 'Opublikowano: 2011-06-08'. The article begins with the heading 'Wprowadzenie' and the first paragraph: 'W poprzednich publikacjach czytelnik zapoznał się ze sposobami tworzenia baz danych, tabel oraz procedur i funkcji w przestrzeni SQL Server. Kolejna część cyklu skupiać się będzie na wewnętrznych mechanizmach organizacji plików danych w zasobach dyskowych (Files and Filegroups), konfigurowania podstawowych opcji oraz zarządzania integralnością danych. Wskazania z tego zakresu jest niezwykle przydatne w następnym milowym kroku do tworzenia...'

Pliki pomocnicze (NDF)



Logiczna struktura przechowywania danych w SQL Server

źródło: <https://msdn.microsoft.com/pl-pl/library/tworzenie-bazy-danych-w-sql-server.aspx>



Pliki pomocnicze (NDF)

```
CREATE DATABASE [Test_Database] ON PRIMARY
(
NAME = N'my_database_primary',
FILENAME = N'C:\Temp\Database\my_database_primary.mdf',
SIZE = 3072KB,
FILEGROWTH = 1024KB),

FILEGROUP [Filegroup1]
(
NAME = N'my_database_secondary',
FILENAME = N'C:\Temp\Database\Secondary\my_database_secondary.ndf',
SIZE = 3027KB,
FILEGROWTH = 1024KB)

LOG ON
(
NAME = N'my_database_log',
FILENAME = N'C:\Temp\Database\Log\my_database_log.ldf',
SIZE = 1024KB,
FILEGROWTH = 10%)
GO
```




Pliki dziennika powtórzeń (LDF)

- *.ldf
- Dziennik powtórzeń - chronologiczny rejestr działań na bazie danych
- Wykorzystywany do odzyskiwania danych po awarii



Informacje o plikach fizycznych

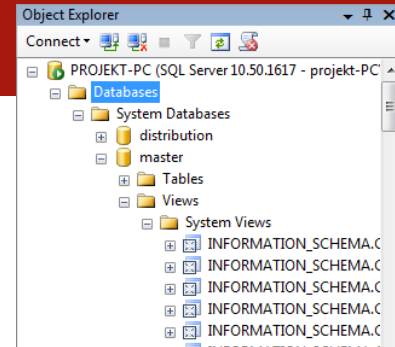
- Informacje o bazach danych
widok *sys.master_files*

(Databases → System Databases → master → Views → System Views)

```
SELECT * FROM sys.master_files
```

- Informacje o plikach fizycznych

```
sp_helpdb BazaDanych
```





Struktura widoku *sys.master_files*

- źródło: <https://msdn.microsoft.com/en-us/library/ms186782.aspx>

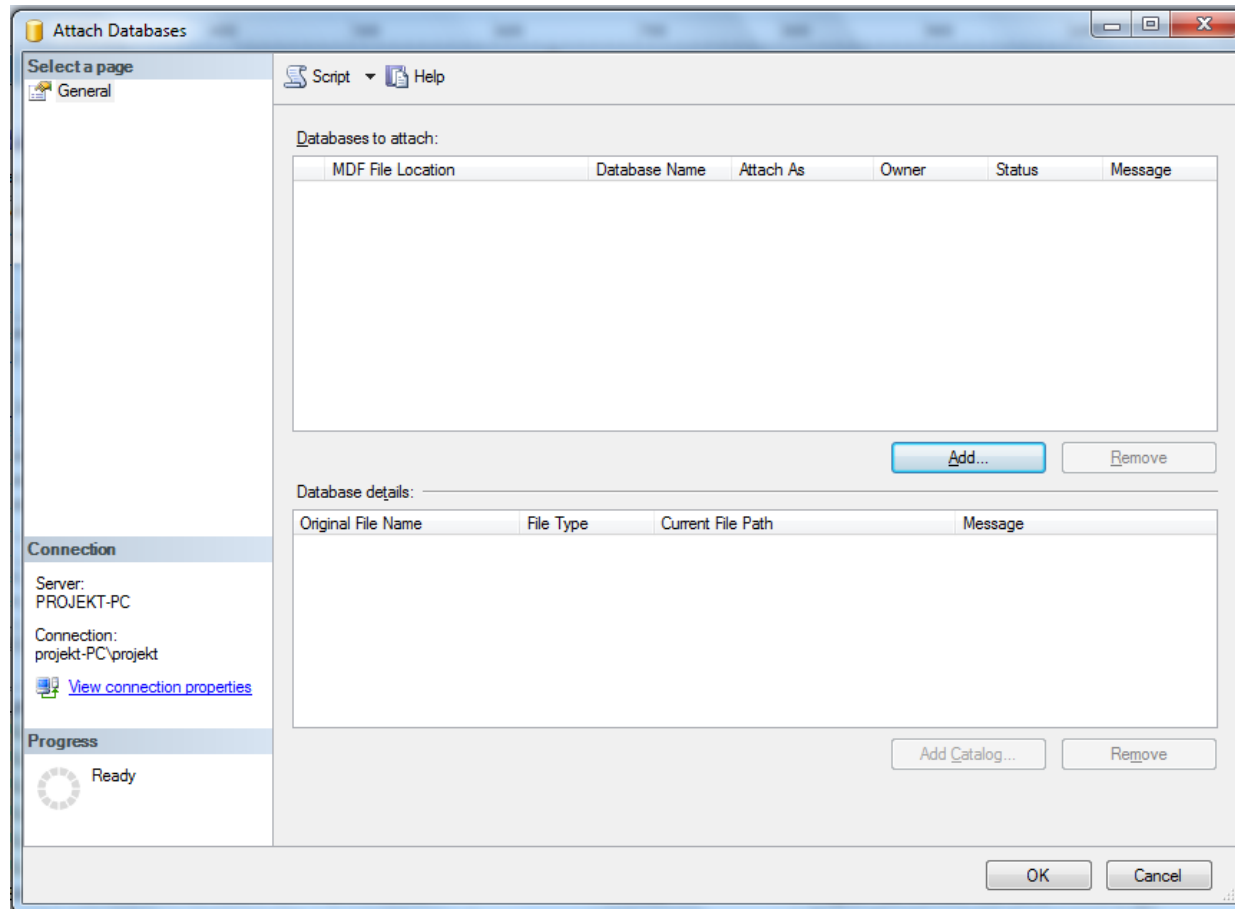
file_id, file_guid, type, type_desc, data_space_id, name, physical_name, state, state_desc, size, max_size, growth, is_media_read_only, is_read_only, is_sparse, is_percent_growth, is_name_reserved, create_lsn, drop_lsn, read_only_lsn, read_write_lsn, differential_base_lsn, differential_base_guid, differential_base_time, redo_start_lsn, redo_start_fork_guid, redo_target_lsn, redo_target_fork_guid, backup_lsn

	database_id	file_id	f
1	1	1	
2	1	2	
3	2	1	
4	2	2	
5	3	1	
6	3	2	
7	4	1	
8	4	2	
9	5	1	
10	5	2	
11	6	1	
12	6	2	

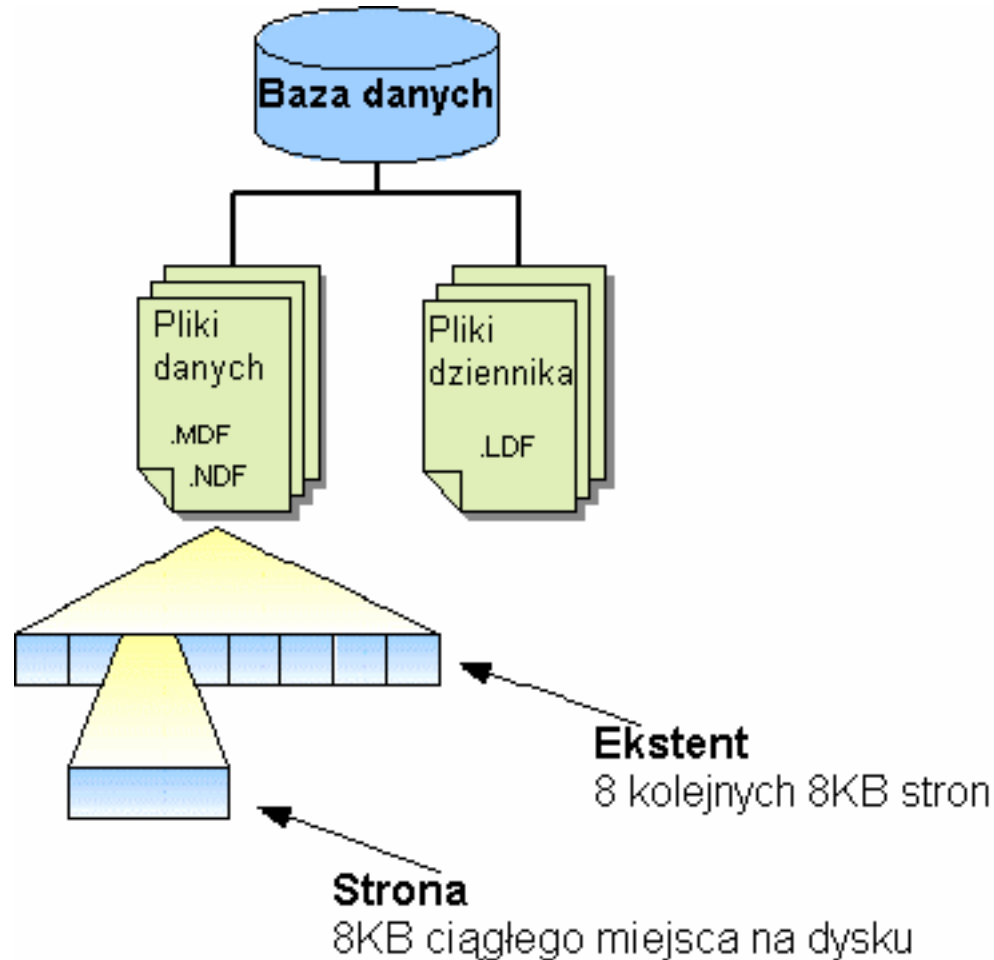


Dołączanie bazy danych

- Databases → Attach...



Podział bazy danych na pliki, ekstenty i strony



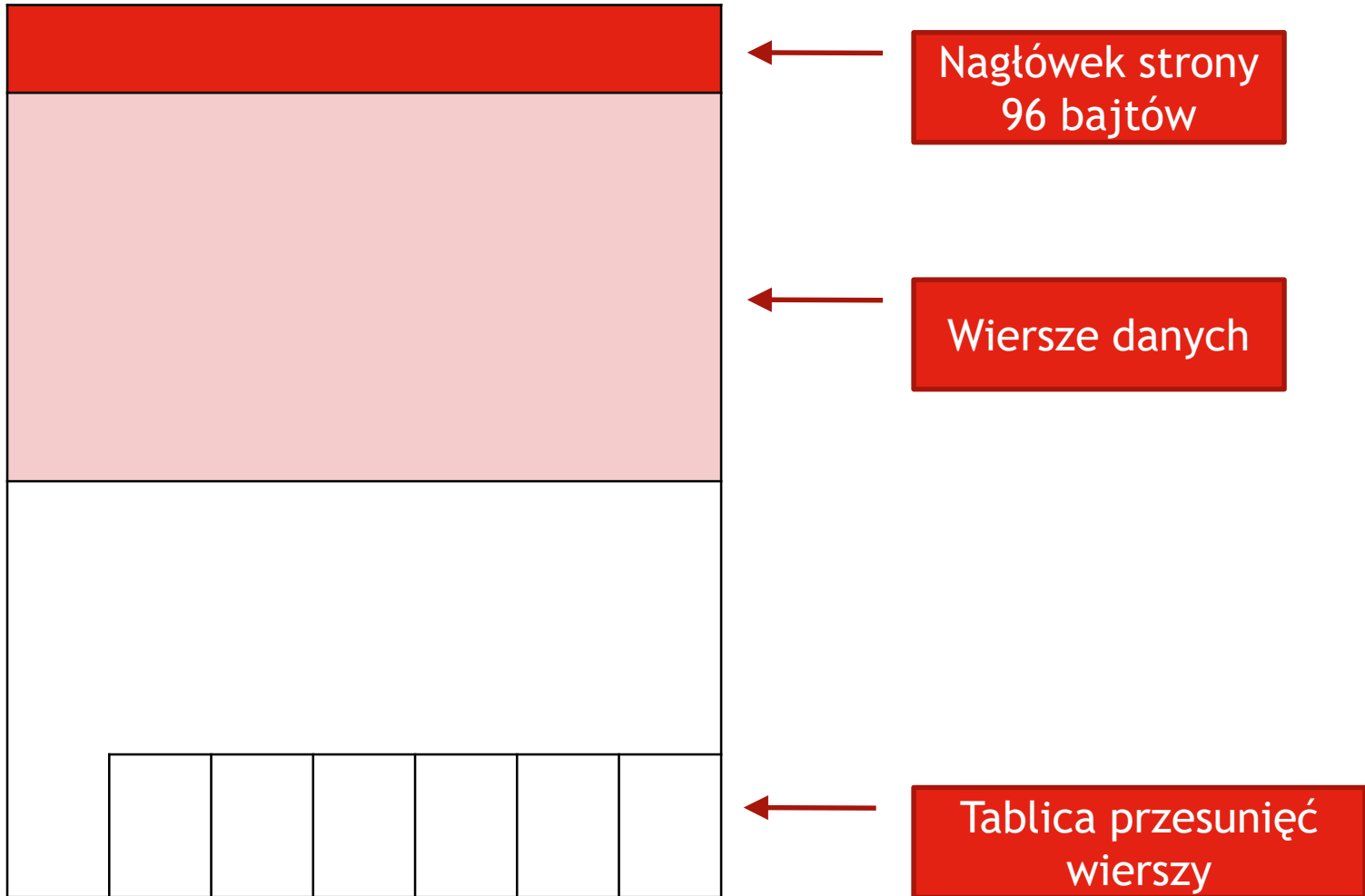


Rodzaje stron

- **data** - wszystkie dane z wyjątkiem atrybutów typów LOB (ang. *Large Object*)
- **index** - wpisy indeksów
- **text/image** - typy LOB: text, ntext, image, varchar(max), nvarchar(max), varbinary(max), xml
- **GAM** (*Global Allocation Map*), **SGAM** (*Shared GAM*), **IAM** (*Index Allocation Map*) - bitowe mapy alokacji

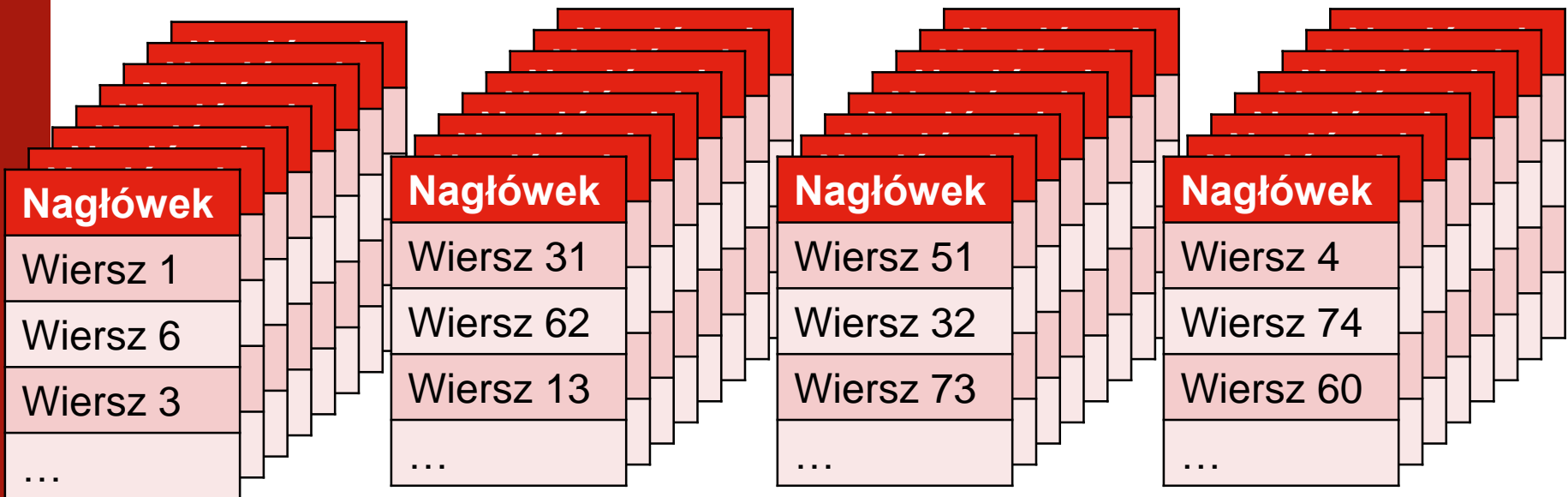


Struktura strony danych



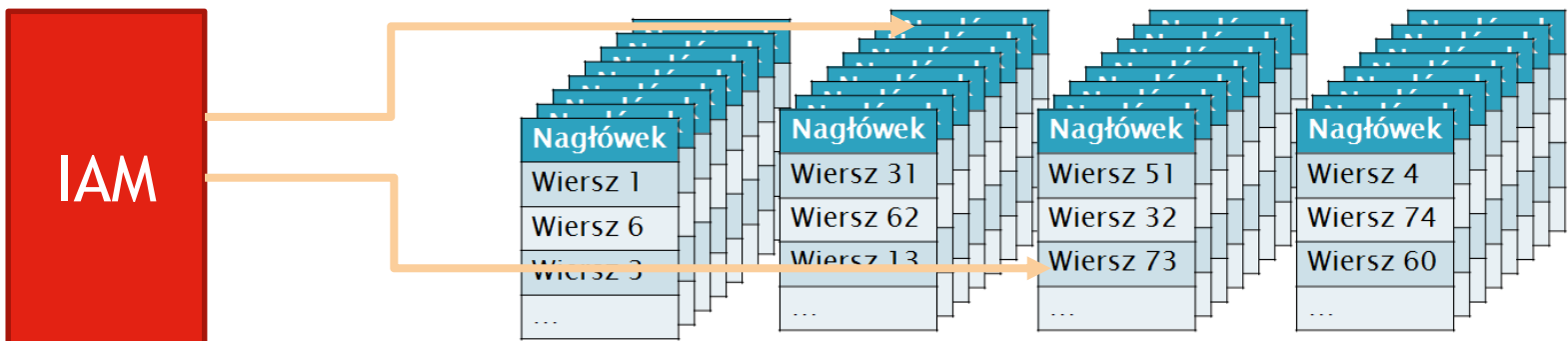
Szeregi (ang. *heap*)

- Zbiór obszarów zawierających dane z jednej tabeli (lub partycji)
- Dane nie są ze sobą powiązane
- Wyszukiwanie wymaga przejrzania wszystkich stron



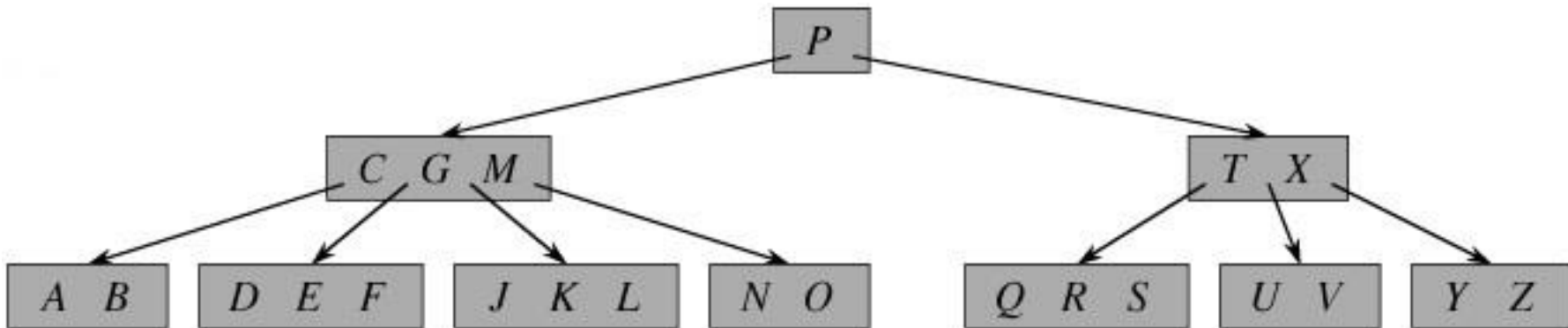
Fizyczna organizacja danych w SQL Server

- Strony GAM, SGAM informują, które obszary są wolne, zajęte
 - **GAM** - informacje o zajętych obszarach jednolitych (*uniform*)
 - **SGAM** - informacje o zajętych obszarach mieszanych (*mixed*)
- Strony **IAM** - informacje o przynależności obszarów do obiektów



B-drzewo

- Drzewo zbalansowane
- Węzły mogą mieć strukturę listy dwukierunkowej
- W praktyce do 5 poziomów



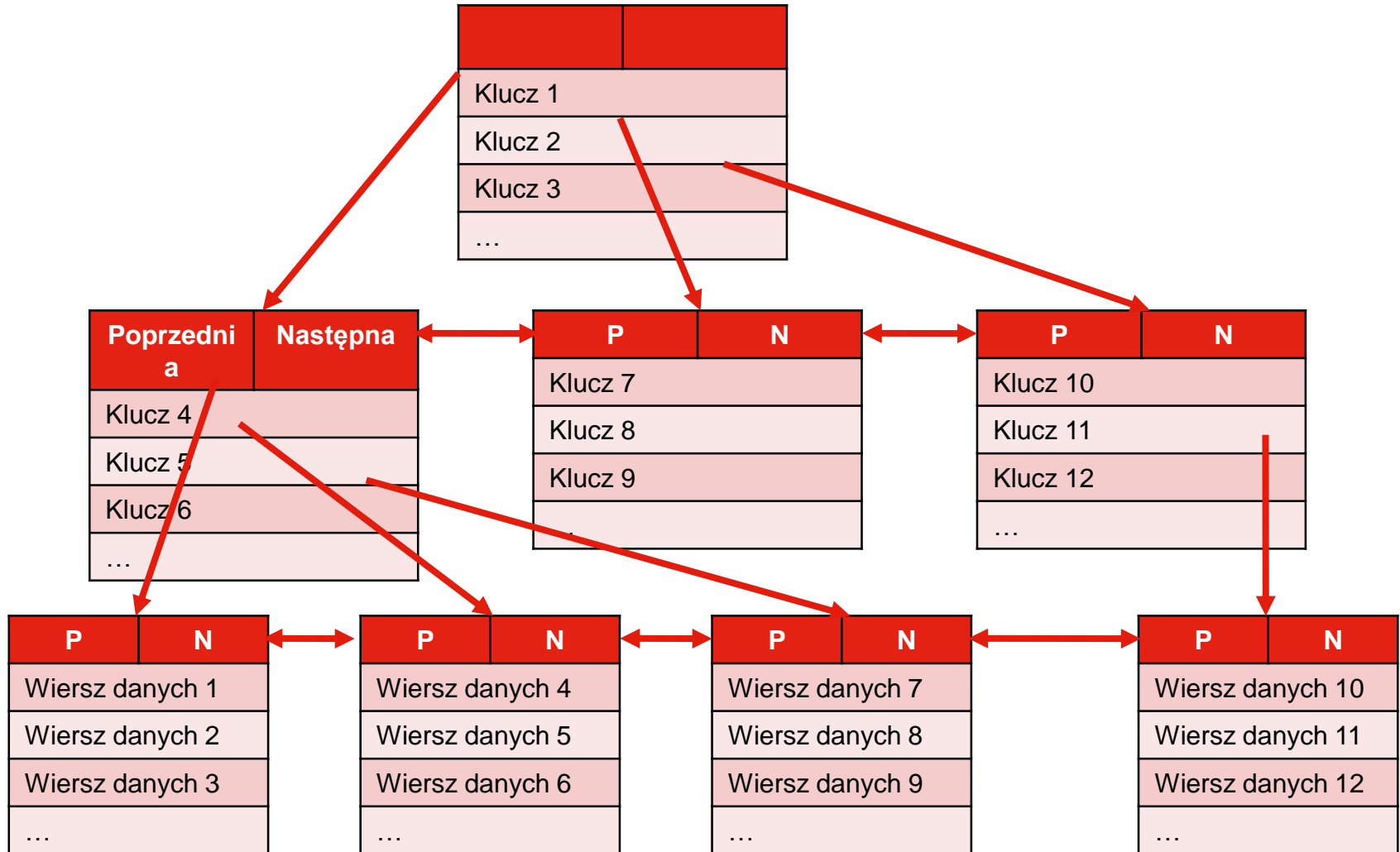


Indeks zgrupowany (z danymi)

- ang. *clustered index*
- Struktura drzewa (B-tree)
- Na poziomie korzeni i gałęzi - **strony indeksu**
- Na poziomie liści - **strony z danymi z tabel**
- Fizycznie porządkuje dane
- Dane fizyczne uporządkowane rosnące wg klucza indeksu
- Może istnieć tylko 1 indeks zgrupowany
- Zakładany najczęściej na sztucznym kluczy podstawowym tzw. id



Indeks zgrupowany





Na jakich kolumnach tworzyć indeks zgrupowany?

- Mała długość atrybutu (klucza)
- Wysoka selektywność (mało powtarzających się wartości klucza indeksu)
- Rzadko bądź wcale nie zmieniane wartości
- Wartości klucza dla kolejno dodawanych wierszy są rosnące

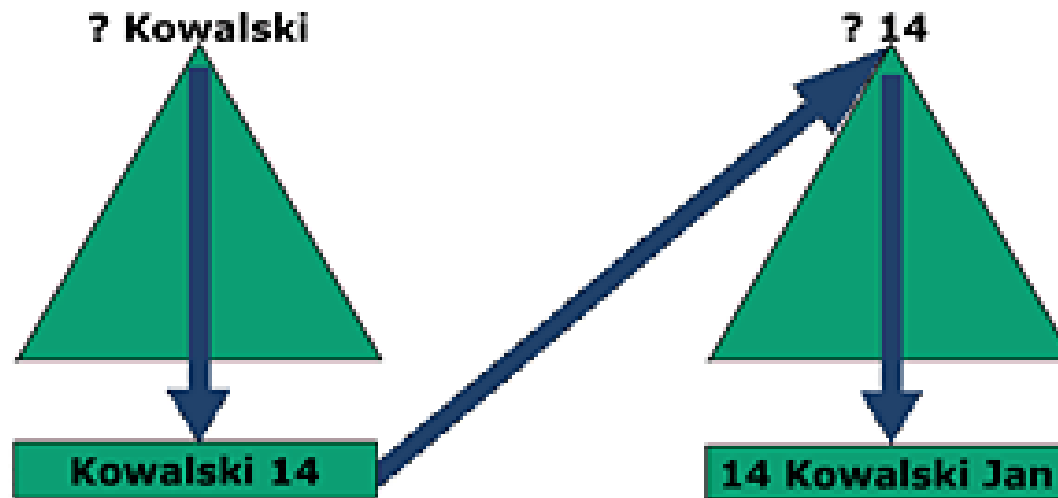
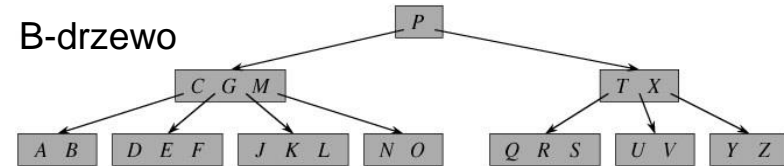


Indeks niezgrupowany

- Struktura drzewa (B-tree)
- Na **wszystkich** poziomach drzewa mamy **strony indeksu**
- Może być budowany na stercie lub na indeksie zgrupowanym
- Można stworzyć do 249 indeksów niezgrupowanych na tabeli (od SQL Server 2008 do 999)
- Stosowane są, gdy dane wyszukiwane są według wielu kryteriów
- Maksymalnie 16 kolumn w kluczu

Indeksy niezgrupowany budowany na indeksie zgrupowanym

- Indeks niezgrupowany
- Indeks zgrupowany



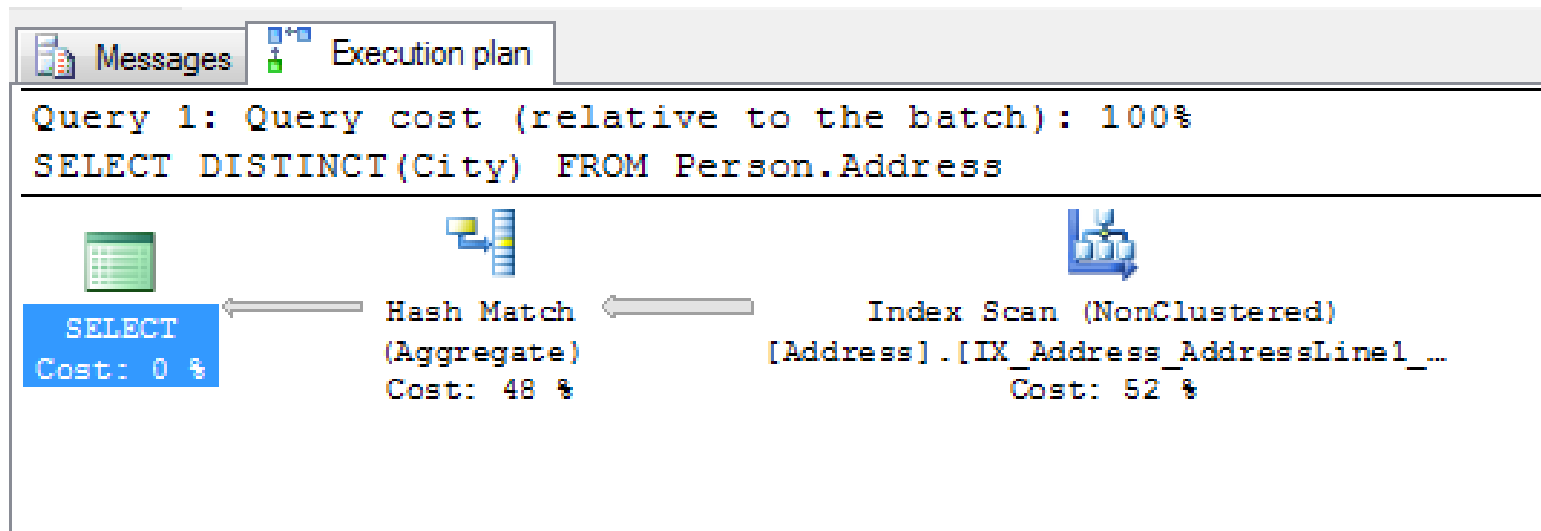
Liść indeksu niegrupowanego

Rekord z danymi



Kwestie wydajnościowe

- Fizyczna organizacja danych wpływa na wydajność pracy BD
- Plan wykonania





Na przykładzie MS SQL Server

OPTYMALIZACJA BAZ DANYCH

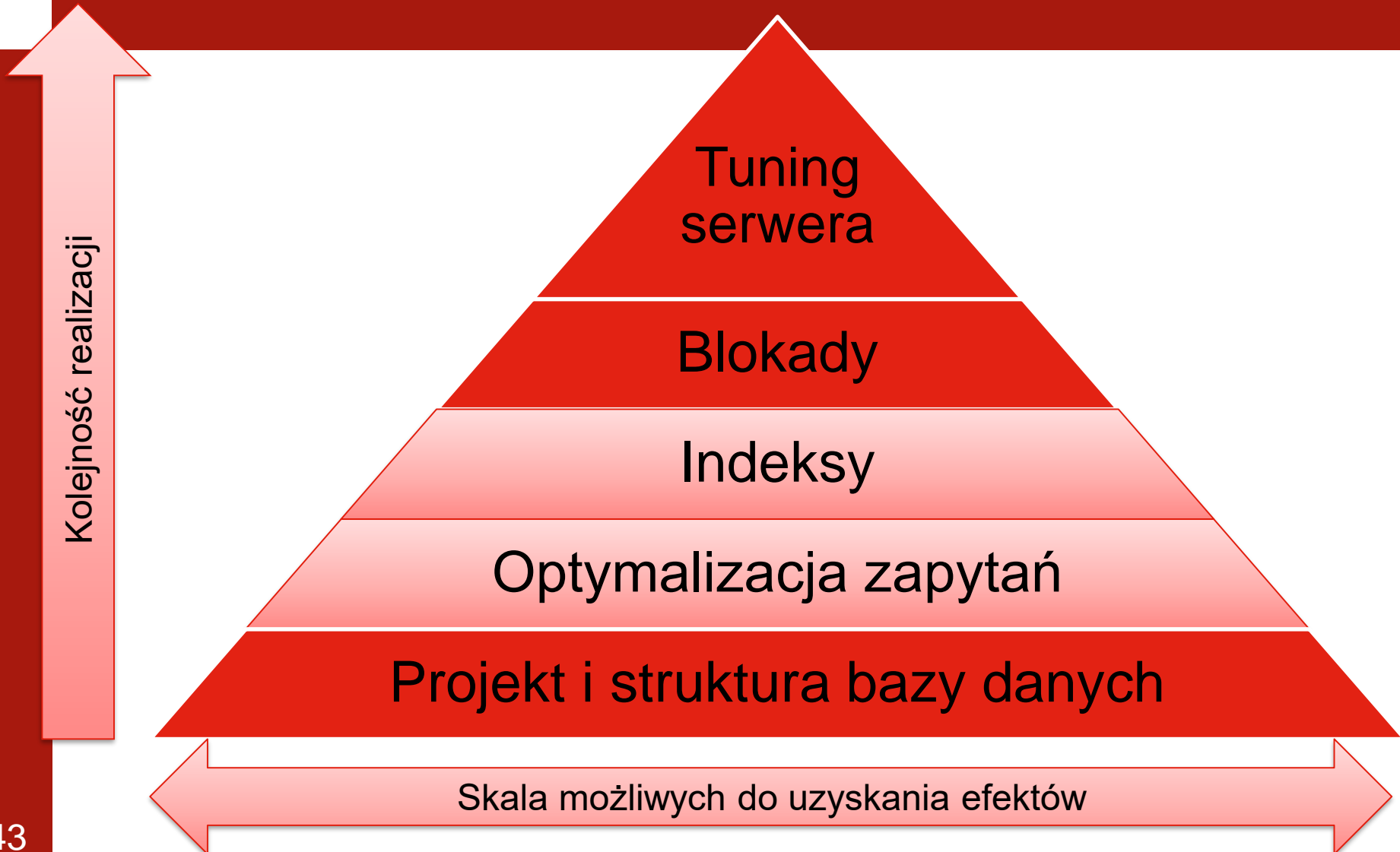


Optymalizacja BD - wprowadzenie

- Problem C10k
- Optymalizacja serwerów BD
- Zwiększenie przepustowości łącza sieciowego
- Optymalizacji BD: normalizacja, denormalizacja, zastosowanie indeksów, procedury składowane i inne



Piramida optymalizacyjna





Model optymalizacji wydajności

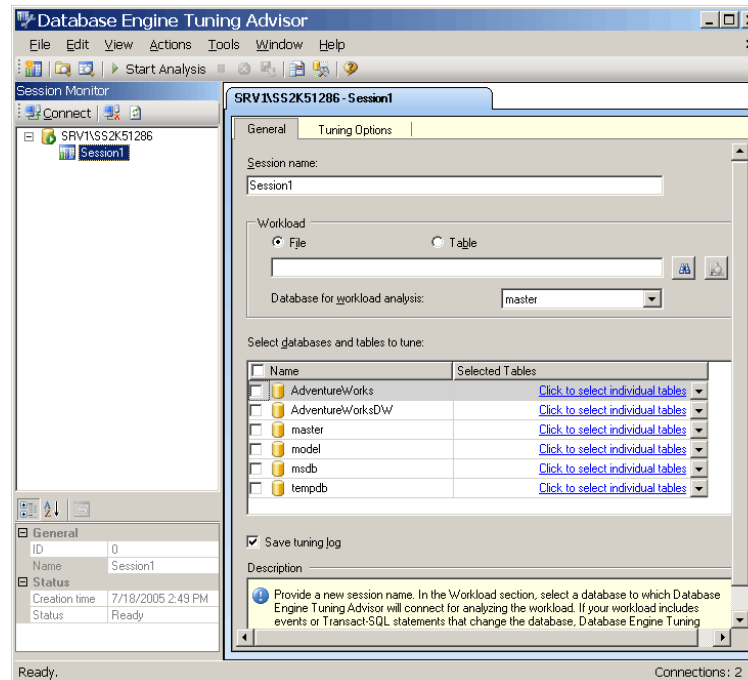
- Projekt struktury bazy danych
 - Najważniejsze zadanie w procesie optymalizacji
 - Skutek złego projektu trudno poprawić
 - Warstwa abstrakcji danych
- **Optymalizacja zapytań**
- **Indeksy**
- **Blokady**
 - Baza danych działa prawidłowo z jednym użytkownikiem a przy kilku zaczyna „spowalniać”
- **Tuning serwera**
 - Dodawanie procesorów, zwiększanie pamięci RAM, zmiana dysków twardych itd.



PLAN WYKONANIA W MS SQL SERVER

Narzędzia wspierające optymalizację

- MS SQL Server Management Studio
 - Plan wykonania
- Database Engine Tuning Advisor





Rola optymalizatora zapytań

- T-SQL jest językiem deklaratywnym.
- Używając go, stwierdzamy **co** chcemy zrobić, a nie **jak** to zrobić.
- System zarządzania bazą danych posługując się **optymalizatorem zapytań** decyduje o sposobie wykonania zapytania.



Fazy przetwarzania zapytania



Plan wykonania MS SQL Server

- Estymowany plan wykonania (ang. *estimated execution plan*)
- Rzeczywisty plan wykonania (ang. *actual execution plan*)



Graficzny plan wykonania w SQL Server Management Studio

- Dostarcza możliwość reprezentacji graficznej planów zapytań w postaci drzewa operatorów
- Podpowiada kroki warte podjęcia celem optymalizacji zapytań (np. wykrywa brakujące indeksy)

Query 1: Query cost (relative to the batch): 100%

```
SELECT [member].[member_no], [member].[lastname], [member].[fi...
```










The graphical plan shows the following operators and their costs:

- Hash Match (Inner Join) Cost: 43 %
- Clustered Index Scan (Clustered) [corporation].[corporation_ident] Cost: 3 %
- Clustered Index Scan (Clustered) [member].[member_ident] Cost: 53 %






The right-hand pane displays the following properties:

Property	Value
Misc	
Cached plan size	40 KB
CompileCPU	4
CompileMemory	296
CompileTime	4
Degree of Parallelism	1
Estimated Number of Rows	1295.63
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	0.223126
Logical Operation	
Memory Grant	1352
MemoryGrantInfo	
Optimization Level	FULL
OptimizerHardwareDependentProperties	
Physical Operation	
QueryHash	0xAB3CE18EFBC5E187
QueryPlanHash	0x64A3D75BFCBA3094
Reason For Early Termination Of Statement	Good Enough Plan Found
RetrievedFromCache	true
Set Options	
Statement	SELECT [member].[member_no], [member].[

Ikony planu wykonania (wybór)

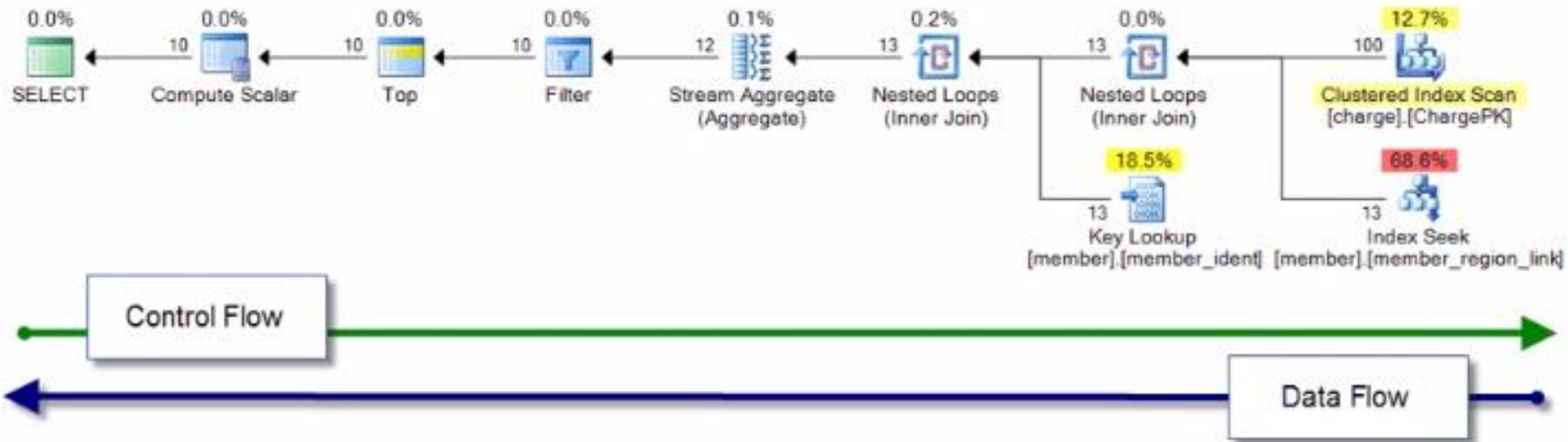
Ikona	Element języka/Operator	(ang.)
	Wynik	Result
	Skanowanie indeksu zgrupowanego	Clustered Index Scan
	Szukanie indeksu zgrupowanego	Clustered Index Seek
	Skanowanie indeksu zwykłego	Nonclustered Index Scan
	Szukanie indeksu zwykłego	Nonclustered Index Seek
	Pętla zagnieżdżona	Nested Loops
	Wyszukiwanie zakładki	Bookmark Lookup
	Sortowanie	Sort
	Dopasowanie wartości mieszającej	Hash Match

Ikony planu wykonania (cd.)

Ikona	Element języka/Operator	(ang.)
	Obliczanie wartość skalarna	Compute Scalar
	Top	Top
	Filtruj	Filtr
	Obliczanie wartość skalarnej	Compute Scalar
	Agregat strumienia	Stream Aggregate
	(...)	



Interpretacja planu wykonania





Okno podpowiedzi z informacjami o operatorze

```

SELECT Nazwisko
      , Imie
FROM BazaRelacyjna
JOIN [BazaRelacyjna].[dbo].[Klienci] k
ON k.IdMiasta=m.IdMiasta
AND m.IdWojewodztwa=k.IdWojewodztwa
GO
        
```

Hash Match

Use each row from the top input to build a hash table, and each row from the bottom input to probe into the hash table, outputting all matching rows.

Physical Operation	Hash Match
Logical Operation	Inner Join
Actual Number of Rows	26
Estimated I/O Cost	0
Estimated CPU Cost	0,0196235
Estimated Number of Executions	1
Number of Executions	1
Estimated Operator Cost	0,0197561 (72%)
Estimated Subtree Cost	0,0275778
Estimated Number of Rows	148,412
Estimated Row Size	27 B
Actual Rebinds	0
Actual Rewinds	0
Node ID	0

Output List
[BazaRelacyjna].[dbo].[Klienci].Nazwisko;
[BazaRelacyjna].[dbo].[Klienci].Imie

Probe Residual
[BazaRelacyjna].[dbo].[Klienci].[IdMiasta] as [k].
[IdMiasta]=[BazaRelacyjna].[dbo].[Miasta].[IdMiasta] as [m].[IdMiasta]

Hash Keys Probe
[BazaRelacyjna].[dbo].[Klienci].IdMiasta

Query 2: Query cost 0,0275778

```

SELECT Nazwisko , Imie
FROM [BazaRelacyjna].[dbo].[Klienci] k
JOIN [BazaRelacyjna].[dbo].[Miasta] m ON k.IdMiasta=m.IdMiasta
        
```

Disconnected.




















Przykład z BD AdventureWorks2008

The screenshot displays the schema for the `Person.Person` table in the AdventureWorks2008 database. The schema is organized into several categories:

- Columns:**
 - `BusinessEntityID` (PK, FK, int, not null)
 - `PersonType` (nchar(2), not null)
 - `NameStyle` (NameStyle(bit), not null)
 - `Title` (nvarchar(8), null)
 - `FirstName` (Name(nvarchar(50)), not null)
 - `MiddleName` (Name(nvarchar(50)), null)
 - `LastName` (Name(nvarchar(50)), not null)
 - `Suffix` (nvarchar(10), null)
 - `EmailPromotion` (int, not null)
 - `AdditionalContactInfo` (XML(Person.AdditionalContactInfoSchemaCollection), null)
 - `Demographics` (XML(Person.IndividualSurveySchemaCollection), null)
 - `rowguid` (uniqueidentifier, not null)
 - `ModifiedDate` (datetime, not null)
- Keys:**
 - `PK_Person_BusinessEntityID`
 - `FK_Person_BusinessEntity_BusinessEntityID`
- Constraints:** (Folder icon with a plus sign)
- Triggers:** (Folder icon with a plus sign)
- Indexes:** (Folder icon with a plus sign)
- Statistics:** (Folder icon with a plus sign)



Indeksy w tabeli Person.Person

- [-]  Person.Person
 - [+]  Columns
 - [-]  Keys
 -  PK_Person_BusinessEntityID
 -  FK_Person_BusinessEntity_BusinessEntityID
 - [+]  Constraints
 - [+]  Triggers
 - [-]  Indexes
 -  AK_Person_rowguid (Unique, Non-Clustered)
 -  IX_Person_LastName_FirstName_MiddleName (Non-Unique, Non-Clustered)
 -  PK_Person_BusinessEntityID (Clustered)
 -  PXML_Person_AddContact (Primary XML)
 -  PXML_Person_Demographics (Primary XML)
 -  XMLPATH_Person_Demographics (Secondary XML, Path)
 -  XMLPROPERTY_Person_Demographics (Secondary XML, Property)
 -  XMLVALUE_Person_Demographics (Secondary XML, Value)
 - [+]  Statistics



Przykład 1: Proste plany wykonania

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT * FROM [AdventureWorks2008].[Person].[Person]
```

Clustered Index Scan (Clustered)
[Person].[PK_Person_BusinessEntityI...]
Cost: 100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT * FROM [AdventureWorks2008].[Person].[Person] WHERE [BusinessEntityID]=@1
```

Clustered Index Seek (Clustered)
[Person].[PK_Person_BusinessEntityI...]
Cost: 100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person]
```

Index Scan (NonClustered)
[Person].[IX_Person_LastName_FirstN...]
Cost: 100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person] WHERE [BusinessEntityID]=@1
```

Clustered Index Seek (Clustered)
[Person].[PK_Person_BusinessEntityI...]
Cost: 100 %



Przykład 1 cd.: Proste plany wykonania

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person] WHERE [FirstName]=@1
```

Missing Index (Impact 96.4982): CREATE NONCLUSTERED INDEX [<Name of Missing Index, sysname,>] ON [Person].[Person] ([FirstName])

```
SELECT  
Cost: 0 %
```

Index Scan (NonClustered)
[Person].[IX_Person_LastName_FirstN...]
Cost: 100 %

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

```
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person] WHERE [rowguid]=@1
```

```
SELECT  
Cost: 0 %
```

Nested Loops (Inner Join)
Cost: 0 %

Index Seek (NonClustered)
[Person].[AK_Person_rowguid]
Cost: 50 %

Key Lookup (Clustered)
[Person].[PK_Person_BusinessEntityL...]
Cost: 50 %



Dwie wersje zapytania SQL

```
SELECT *
FROM BazaRelacyjna.dbo.Klienci k
WHERE k.IdMiasta IN
    (SELECT [IdMiasta]
     FROM [BazaRelacyjna].[dbo].[Miasta] m
     WHERE IdWojewodztwa=12)
GO
```

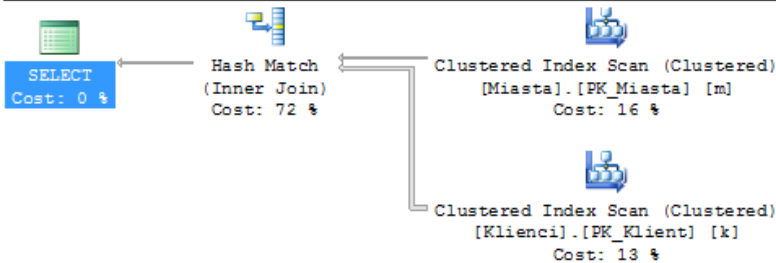
```
SELECT *
FROM BazaRelacyjna.dbo.Klienci k
JOIN [BazaRelacyjna].[dbo].[Miasta] m
ON k.IdMiasta=m.IdMiasta
AND m.IdWojewodztwa=12
GO
```



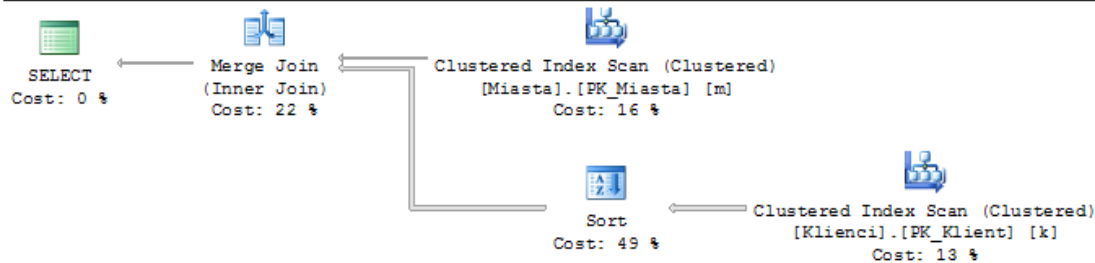
Uzyskane plany wykonania

Results Messages Execution plan

Query 1: Query cost (relative to the batch): 50%
SELECT * FROM BazaRelacyjna.dbo.Klienci k WHERE k.IdMiasta IN (SELECT [IdMiasta] FROM [BazaRelacyjna].[dbo].[Miasta] m WHERE IdWojewodztwa=12)



Query 2: Query cost (relative to the batch): 50%
SELECT * FROM BazaRelacyjna.dbo.Klienci k JOIN [BazaRelacyjna].[dbo].[Miasta] m ON k.IdMiasta=m.IdMiasta AND m.IdWojewodztwa=12





roclawska

Politechnika Wroclawska

Przegląd innych systemów baz danych

MySQL, SQLite, PostgreSQL



DB-Engines Ranking (X'2020)

Oct 2020	Oct 2019	Change	Programming Language	Ratings	Change
1	1		Oracle	1368.77	+12.89
2	2		MySQL	1256.38	-26.69
3	3		Microsoft SQL Server	1043.12	-51.60
4	5		PostgreSQL	542.40	+58.49
5	4		MongoDB	448.02	+35.93
6	6		IBM DB2	161.90	-8.87
7	8		Elasticsearch	153.84	+3.67
8	7		Redis	153.28	+10.37
9	11	↑	SQLite	125.43	+2.80
10	10		Cassandra	119.10	-4.12



Wersje BD (X'2020)

Produkt	Producent	Najnowsza wersja	Licencja
MySQL	Oracle	8.0.21	GPL lub komercyjna
MariaDB	Monty Program Ab	10.5.6	GPL v. 2
Microsoft SQL Server	Microsoft	2019 (15.0.2000.5)	Microsoft EULA
Oracle Database	Oracle	19c (12.2.0.3)	komercyjna
PostgreSQL	PostgreSQL Global Development Group	13.0	PostgreSQL
SQLite	D. Richard Hipp	3.33.0	Public domain
Informix	IBM	14.10.FC3	EULA
DB2	IBM	11.5	EULA



Duże bazy danych

- Microsoft SQL Server 2019, Azure SQL Database
- Oracle 11g, 12c, 18c, 19c
- IBM Informix Extended Parallel Server (XPS)
- SAP Sybase Adaptive Server Enterprise 16.0 (ASE)
- Interbase XE3



Średnie bazy danych

- MySQL
- MariaDB
- PostgreSQL
- Firebird 3.0.2
- Microsoft Visual FoxPro 9.0
- MS Access 2016



Małe bazy danych

- mSQL 4.0
- SQLite

- Nie nadają się do tworzenia wielodostępowych aplikacji internetowych.



MYSQL/MARIADB



MySQL

- SZRBD dla Linux/Unix, Windows, ...
- Pierwsze wydanie: 1995 r.
- Licencja: GPL lub komercyjna (Oracle)
- ⇒ MariaDB - GPL wersja 2
- Aktualna wersja stabilna: 8.0.21
- Posiada wiele silników bazy danych (motor, ang. *engine, back end*)
- Dostęp z poziomu wielu języków programowania: m.in. C/C++, Delphi, PHP.



Silniki baz danych MySQL

- InnoDB - domyślny od wersji 5.5, obsługuje transakcje
- MyISAM - starszy typ, nie obsługuje transakcji
- MEMORY (HEAP) - najszybszy typ silnika, gdyż wszystko jest przechowywane wyłącznie w pamięci RAM. Posiada jednak kilka ograniczeń, między innymi nie przechowuje danych po wyłączeniu serwera MySQL.
- Inne: XtraDB, ISAM, BerkeleyDB (BDB), MERGE, ...

```
CREATE TABLE test ENGINE=MEMORY
```



Silniki baz danych MariaDB

- MyISAM
- XtraDB - kompatybilny z InnoDB
- Aria - nowy silnik bazy danych
- Inne: FederatedX, OQGRAPH, SphinxSE, ...



MyISAM

- Domyślny silnik dla wcześniejszych wersji MySQL
- Rozszerzenie silnika ISAM
- Zaprojektowane z myślą o podstawowych aplikacjach internetowych i prostocie działania
- Stosowany ciągle w hurtowniach danych ze względu na możliwość zapytań do bardzo dużych tabel



MyISAM - architektura fizyczna

- Każda tabela zapisywana w trzech plikach:
 - *nazwa_tabeli_.FRM* - definicja tabeli
 - *nazwa_tabeli_.MYD* - dane
 - *nazwa_tabeli_.MYI* - indeksy



MyISAM

- Zalety
 - + szybki odczyt z tabel
 - + prostsze wykonywanie kopii zapasowych
- Wady
 - brak obsługi transakcji
 - brak mechanizmów odpowiedzialnych za integralność danych
 - przy dużych tabelach, długie czasy wykonywania REPAIR TABLE po awarii serwera



InnoDB

- Zapewnia integralności danych oraz spójności i bezpieczeństwa **transakcji**.
- Standard **ACID**
- ACID jest skrótem od angielskich słów: *atomicity* - atomowość, *consistency* - spójność, *isolation* - izolacja, *durability* - trwałość.
- Mechanizm kluczy obcych



ACID

- **Atomowość** transakcji oznacza, że albo wykonujemy ją w całości albo wcale. Nie może dojść do sytuacji, w której wykona się część zapytań.
- **Spójność** oznacza, że po wykonaniu transakcji system będzie spójny, czyli nie zostaną naruszone żadne **zasady integralności**.
- **Izolacja** transakcji oznacza, iż jeżeli dwie transakcje wykonują się współbieżnie, to zazwyczaj (zależnie od poziomu izolacji) nie widzą zmian przez siebie wprowadzanych.
- **Trwałość** danych oznacza, że system potrafi uruchomić się i udostępnić spójne, nienaruszone i aktualne dane zapisane w ramach zatwierdzonych transakcji, na przykład po nagłej awarii zasilania.



InnoDB

- Zalety
 - + obsługa transakcji
 - + gwarantuje integralność danych
 - + lepiej sprawuje się podczas replikacji typu master - slave
- Wady
 - wolniejszy odczyt danych
 - trudniejsze wykonywanie backupów

Zestaw oprogramowania

- LAMP - Linux + MySQL/MariaDB + PHP/Perl/Python
- WAMP - Windows + Apache + MySQL + PHP
- MAMP - Mac OS X + Apache + MySQL + PHP
- XAMPP - X (Cross-platform) Apache, MySQL, PHP, Perl

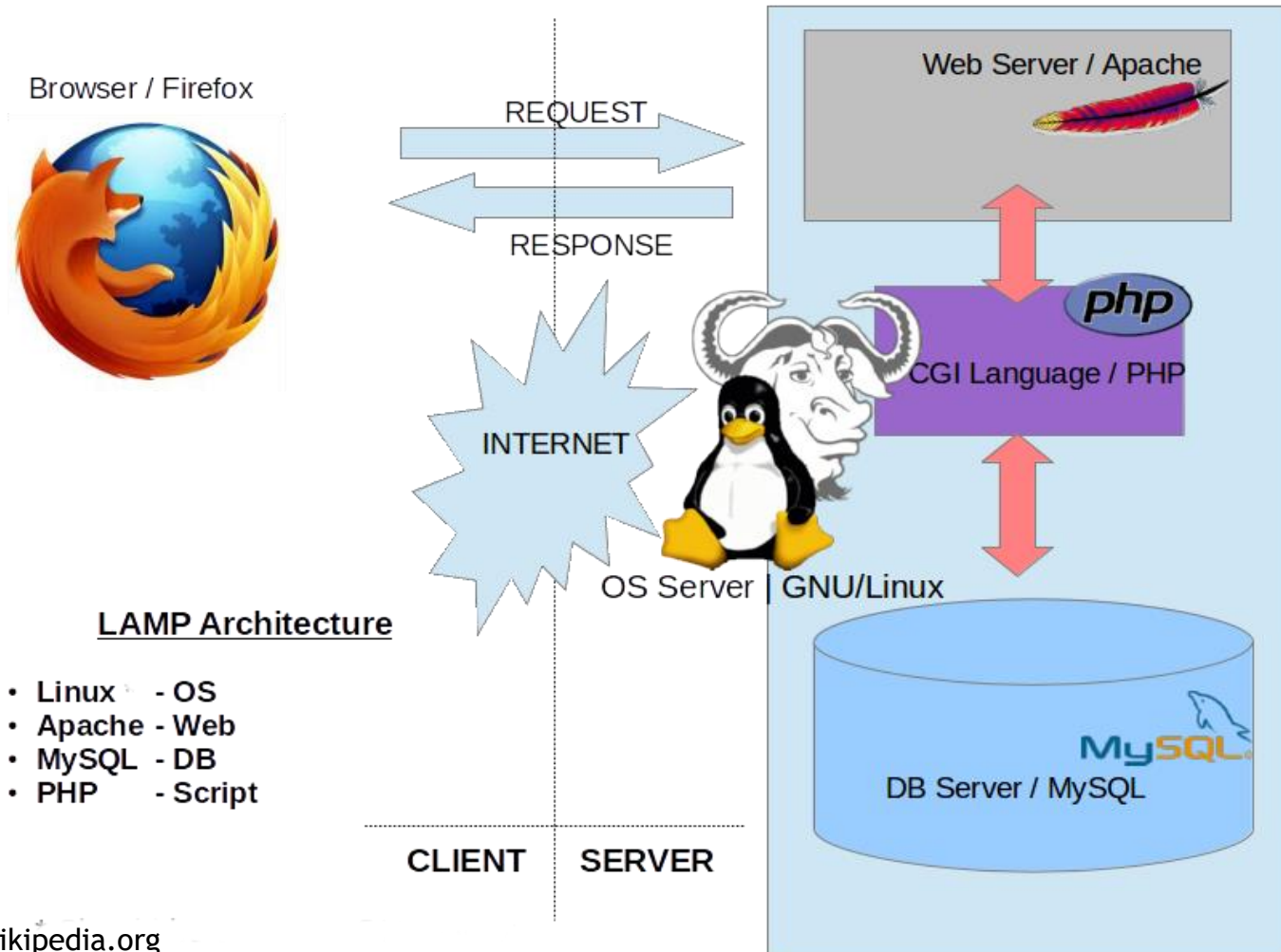


LAMP:





Uproszczony schemat architektury LAMP



LAMP Architecture

- Linux - OS
- Apache - Web
- MySQL - DB
- PHP - Script

Narzędzia administracyjne

- phpMyAdmin - za pomocą przeglądarki internetowej
- MySQL Workbench





SQLITE





SQLite

- Bezserwerowa relacyjna baza danych
- Pierwsze wydanie: 2000 r.
- Licencja: Public domain
- Aktualna wersja: 3.33.0



SQLite - cechy

- Cała baza znajduje się w jednym pliku (do 140 TB):
 - tabele, indeksy, widoki, wyzwalacze itd.
- Baza jest utrzymywana na dysku przy użyciu B-drzew.
 - osobne drzewa są używane dla każdej z tabel i każdego z indeksów.
- Posiada wsparcie dla prawie całego standardu SQL-92.
- Wydajność SQLite jest porównywalna do popularnych SZBD typu klient-serwer.
- Silnik jest napisany w języku C i korzysta tylko z podstawowych funkcji bibliotecznych (sqlite3.c z plikiem nagłówkowym sqlite3.h).
- Nie zawiera systemu uwierzytelniania.



Zastosowania SQLite

- Mozilla Firefox - przeglądarka internetowa
- PHP, Python - jedna z dostępnych baz danych
- Ruby on Rails - domyślna baza danych w Ruby on Rails 2.0.
- Android - środowisko programistyczne dla urządzeń przenośnych
- Oprogramowanie antywirusowe: Avira, Avast
- Zotero - oprogramowanie do zarządzania źródłami informacji.
- Inne

Narzędzia administracyjne

- SQLite Manager - dodatek przeglądarki Mozilla Firefox do zarządzania bazami danych SQLite.
- DaDaBIK Database Interface Kreator (Open Source)
- SQLite Database Browser - graficzne narzędzie do zarządzania bazami danych SQLite.
- SQLiteSpy (Freeware).
- Lita - aplikacja Adobe AIR do zarządzania bazami danych SQLite.
- SQLiteStudio - darmowa aplikacja do zarządzania bazami danych SQLite w wersjach 2 i 3.





POSTGRESQL





PostgreSQL

- Wielodostępny, serwerowy SZBD
- Pierwsze wydanie: 1995 r.
- Platformy: Linux/Unix, Windows, ...
- Licencja: PostgreSQL
- Aktualna wersja: 13.0

PostgreSQL - cechy

- Indeksy: B-drzewo, Hash, R-drzewo, GIST, GIN, SP-GIST
- Funkcje składowane np. w PL/pgSQL
- Wyzwalacze
- Rozszerzona definicja typów danych - m.in. typy danych geograficznych na potrzeby systemów informacji geograficznej (GIS) w module PostGIS.



Narzędzia administracyjne

- pgAdmin
- phpPgAdmin - administracja za pomocą przeglądarki internetowej



phpPgAdmin



roclawska

Politechnika Wroclawska

Hurtownie danych.

**Cele i techniki replikacji
oraz fragmentacji danych**

Serwery lustrzane, obiekty partycjonowane



HURTOWNIE DANYCH

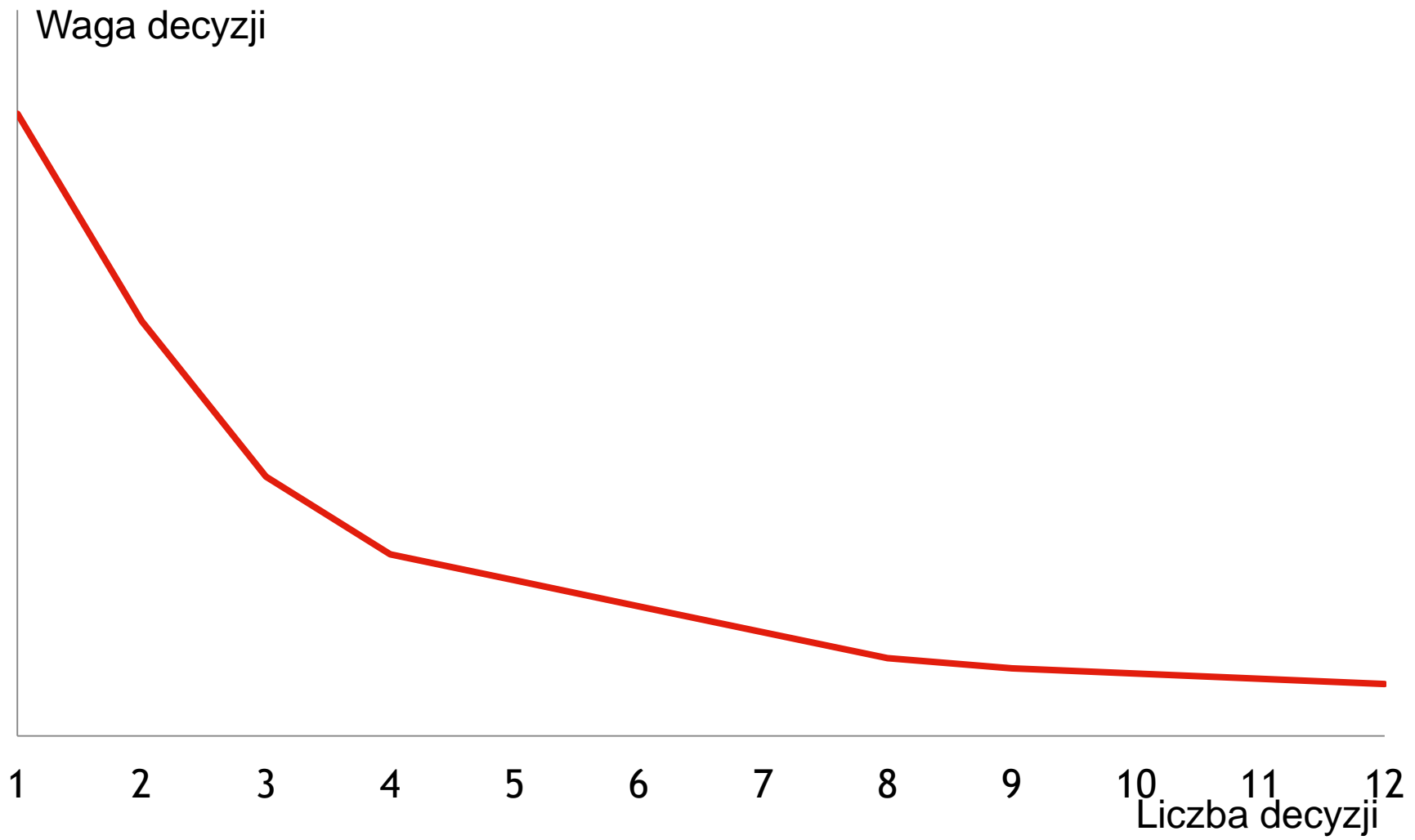


Podstawowe pojęcia

- Przetwarzanie operacyjne
 - **OLTP** (*On-line Transaction Processing*) - duża ilość prostych transakcji zapisu i odczytu. Główny nacisk kładziony jest na zachowanie integralności danych w środowisku wielodostepowym oraz na efektywność mierzona liczbą transakcji w danej jednostce czasu.
- Przetwarzanie analityczne
 - **OLAP** (*On-line Analytical Processing*) - stosunkowo nieliczne, ale za to złożone transakcje odczytu. Miarą efektywności jest czas i jakość odpowiedzi. Powszechnie wykorzystuje się go w technikach związanych z **Data Mining**'iem.



Zależność wagi i liczby decyzji



Hurtownia danych

- ang. *Data Warehouse* - DW
- Analityczna baza danych wykorzystywana jako podstawa systemu wspomagającego podejmowanie decyzji.
- ang. *Data mining* - eksploracja danych, drążenie danych



Cele stosowania hurtowni danych

- Przetwarzanie analityczne (OLAP)
- Wspomaganie decyzji (DSS)
- Archiwizacja danych
- Analiza efektywności
- Wsparcie dla systemów CRM (np. poprzez precyzyjne dobieranie strategii marketingowych na podstawie danych o klientach i sprzedaży)



Architektura hurtowni danych



Raporty systemowe

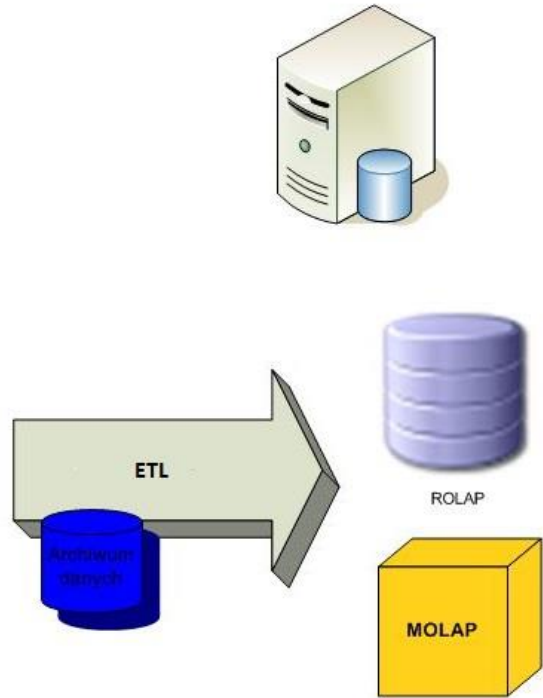


Transakcyjne bazy danych



Słowniki użytkowników

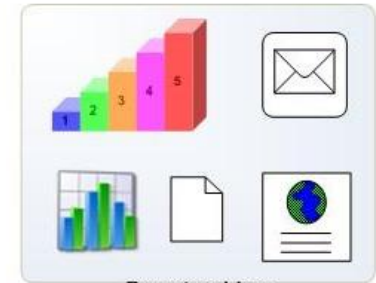
Dane Źródłowe



Hurtownia danych
SQL Server
Oracle
Access



ODBC, OLEDB,
OLAP, ADO,
ADOMD,
WebServices



Raporty ad-hoc



Publikacja



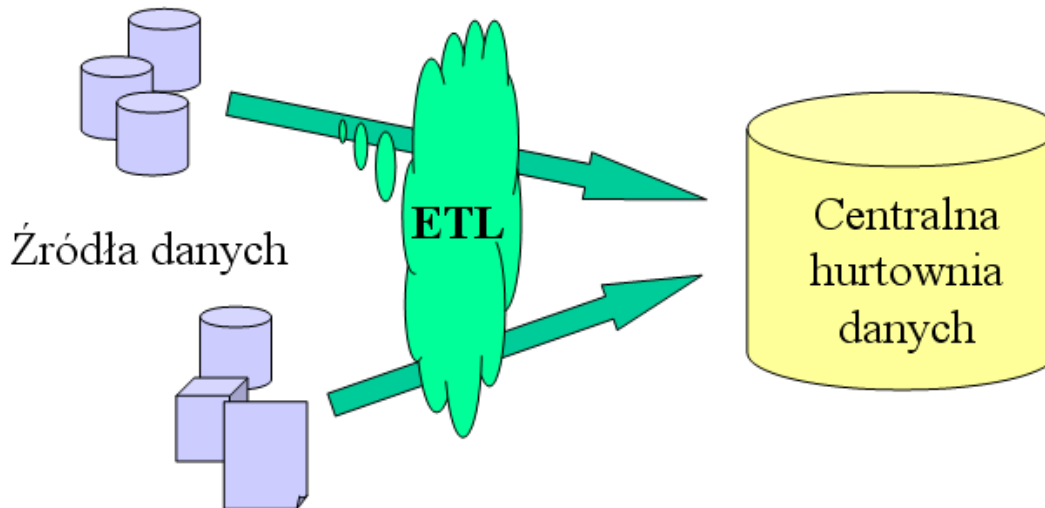
Raporty standardowe



Analiza
Excel
Excel+

Proces integracji danych - ETL

- *Extract, Transform, Load*



- *SQL Server Integration Services (SSIS) - graficzne narzędzie ETL firmy Microsoft włączone do Microsoft SQL*

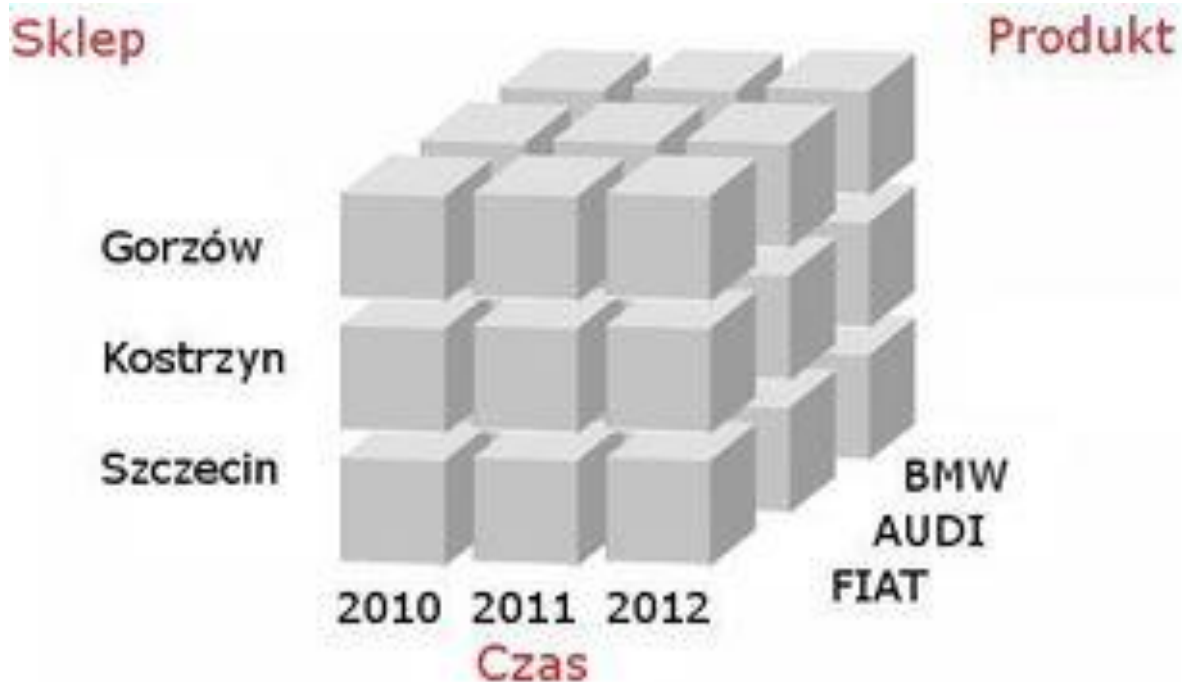


Model danych MOLAP

- *Multidimensional* OLAP
 - implementacja w serwerach wielowymiarowych
 - dane przechowywane w wielowymiarowych tabelach (ang. *data cubes*), zwanych potocznie **kostkami**
- + **najlepsza efektywność**
- **wymaga dużej dodatkowej pamięci**

Kostka MOLAP - przykład

- Trójwymiarowa kostka z wymiarami: Sklep, Czas, Produkt i zagregowanymi danymi o sprzedaży.





Model danych ROLAP

- *Relational* OLAP
 - fakty przechowywane w tabelach faktów
 - wymiary przechowywane w tabelach wymiarów
- + nie potrzeba dodatkowej pamięci
- niska efektywność

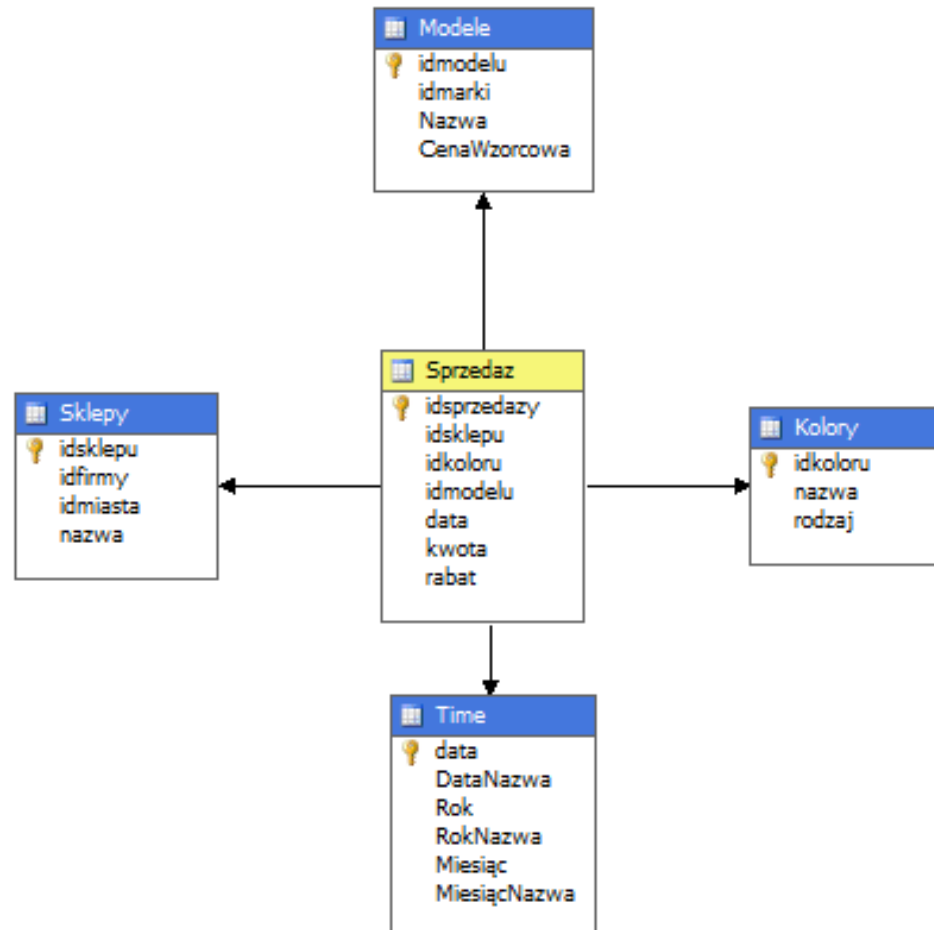


Stosowane schematy w ROLAP

- Schematy podstawowe
 - gwiazda
 - płatek śniegu
- Schematy pochodne
 - Konstelacja faktów
 - Gwiazda - płatek śniegu

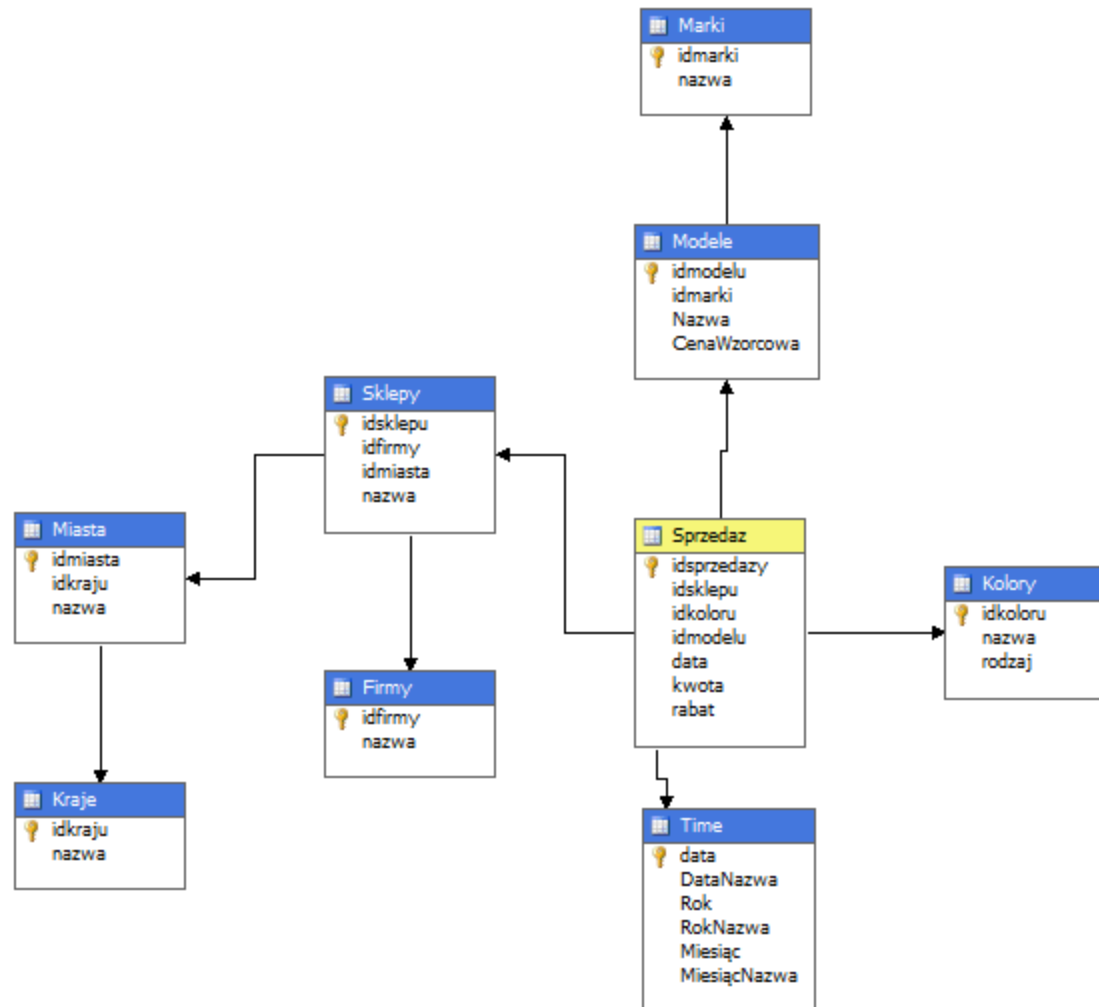


Schemat gwiazdy





Schemat płatka śniegu





Model danych HOLAP

- *Hybrid* OLAP - implementacja hybrydowa (relacyjno-wielowymiarowa)
- dane elementarne (źródłowe) przechowywane w tabelach
- dane zintegrowane przechowywane w kostkach
- pośrednia struktura pomiędzy ROLAP a MOLAP („przyśpieszczacz ROLAP”)



Systemy hurtowni danych

- Oracle Data Warehousing
- Microsoft SQL Server Business Intelligence
- IBM InfoSphere Warehouse
- Teradata Enterprise Data Warehouse
- IBM Netezza Data Warehouse
- Sybase IQ
- Infobright
- SAP NetWeaver Business Intelligence



Skala wdrożenia

- Rozmiar danych $> 1\text{TB}$
- Liczba użytkowników - rzędu 100 (analityków)
- Typowy czas wdrożenia - od 6 miesięcy do 3 lat



Inne przyszłościowe modele

- Data Warehouse Appliances
- „Big data”; paradygmat firmy Google
MapReduce (np. Apache Hadoop)
- Kolumnowe bazy danych
- Bazy danych klasy NoSQL
- Rozwiązania in-memory



CELE I TECHNIKI FRAGMENTACJI DANYCH



Fragmentacja (partycjonowanie) danych

- Obiekty partycjonowane
- Podział tabeli na części, zwane fragmentami lub partycjami
- Techniki fragmentacji
 - pozioma
 - pionowa
 - mieszana

Cel fragmentacji

- Zwiększenie efektywności dostępu do danych
 - zmniejszenie rozmiaru danych, które należy przeszukać
 - zrównoleglenie operacji dostępu do dysków, na których umieszczono fragmenty
 - alokowanie fragmentów „blisko” miejsca ich wykorzystania - redukcja kosztów transmisji sieciowej



Fragmentacja pozioma

- Podział zbioru **rekordów** (krotek) na podzbiory
- Każdy podzbiór opisany jest identyczną liczbą atrybutów
- Wybór fragmentu (partycji) do której trafia rekord realizowany na podstawie wartości jednego lub kilku wybranych atrybutów tabeli - tzw. **atrybutów fragmentujących** (partycjonujących)



Fragmentacja pionowa

- Rozbicie tabeli na fragmenty złożone z podzbioru **atrybutów**
- Każdy fragment zawiera identyczną liczbę rekordów
- Atrybut niekluczowy A_n może się znaleźć tylko w jednym fragmencie
- Atrybut kluczowy znajduje się w każdym fragmencie - służy do łączenia fragmentów

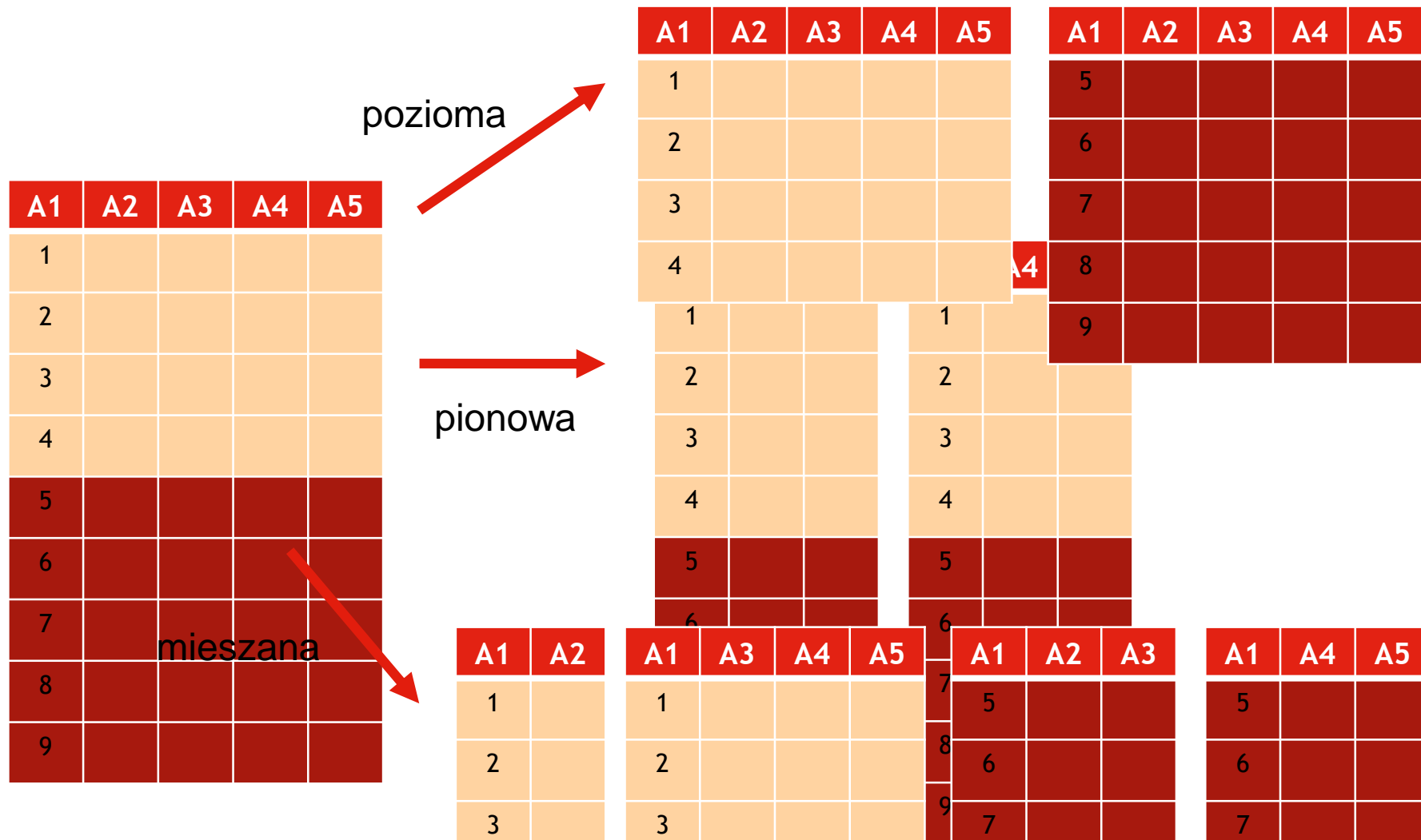


Fragmentacja mieszana

- Połączenie fragmentacji poziomej i pionowej
- Warianty
 - Fragmentacja pozioma + pionowa
 - Fragmentacja pionowa + pozioma



Rodzaje fragmentacji - przykład





Podsumowanie

- Przegląd systemów zarządzania relacyjnymi bazami danych
- Hurtownie danych - specyficzna architektura i zastosowania baz danych
- Cele i techniki fragmentacji danych



Pytania?

DZIĘKUJĘ ZA UWAGĘ