rocławska

Politechnika Wrocławska

Bazy Danych

dr inż. Roman Ptak
Katedra Informatyki Technicznej
roman.ptak@pwr.edu.pl

Plan wykładu 4.

- Microsoft SQL Server
  - Struktura logiczna systemu
  - Struktura fizyczna systemu
  - Język Transact-SQL
- Inne systemy baz danych: MySQL, SQLite, PostgreSQL
- Hurtownie danych
- Cele i techniki replikacji i fragmentacji danych

rocławska

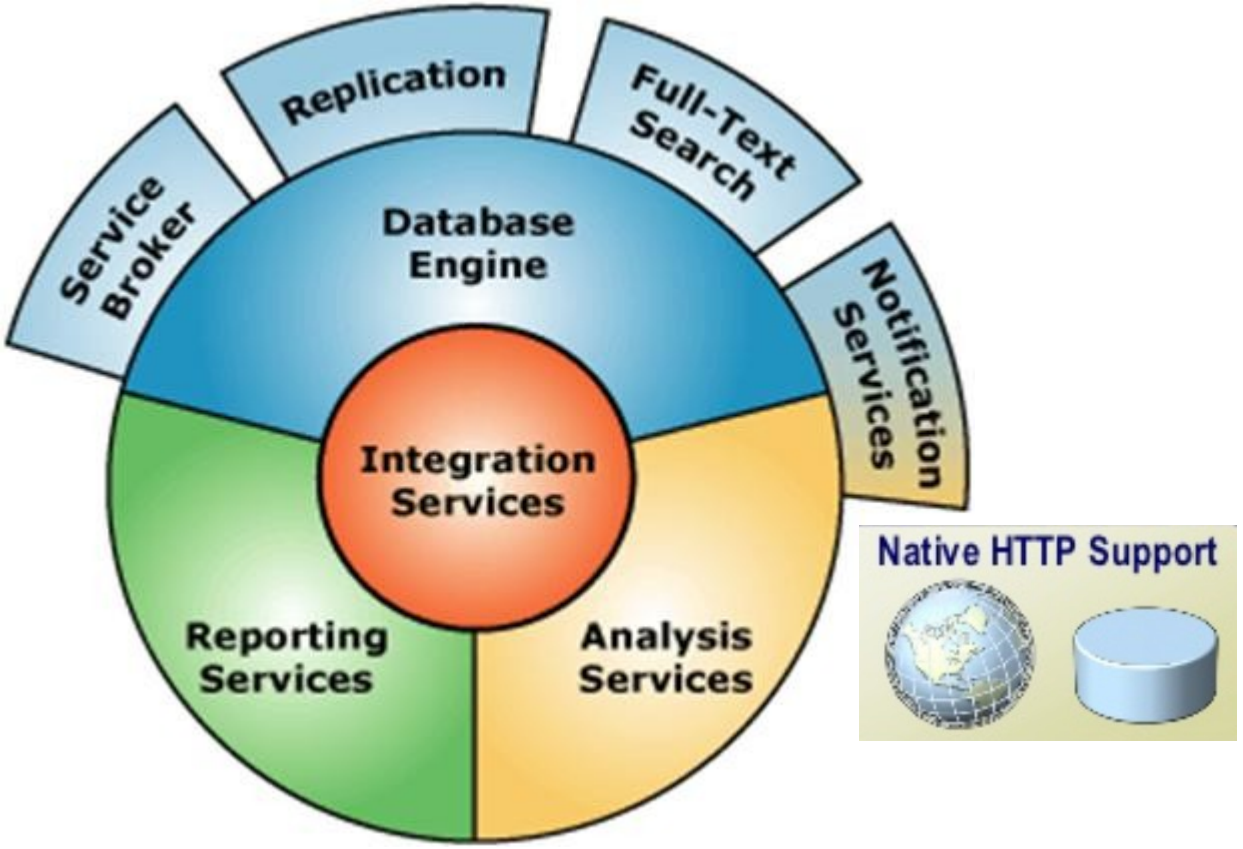
Politechnika Wrocławska

Microsoft SQL Server

Microsoft SQL Server

- Producent: Microsoft
- Licencja: Microsoft EULA
- Najnowsza wersja stabilna: 14.00 (X 2017 r.)
- 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



źródło: http://it-szkola.edu.pl/upload/kuznia/akademia_bd/main.html

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. ***m**ain **d**ata **f**ile*, *master data file*) – główny plik bazy danych
- NDF (ang. ***n**ext **d**ata **f**ile*) – pliki pomocnicze (grupa plików)
- LDF (ang. *log **d**ata **f**ile*) – plik dziennika powtórzeń (logi)

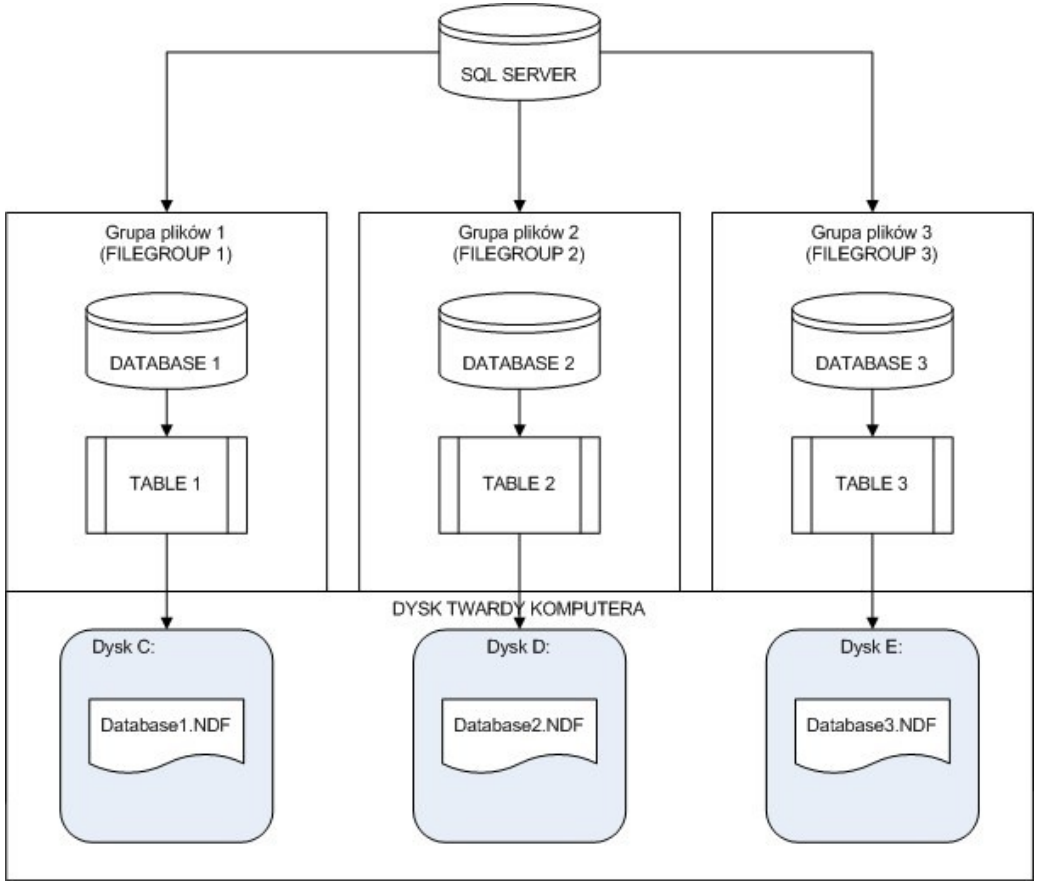
Pliki danych (MDF)

- 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

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

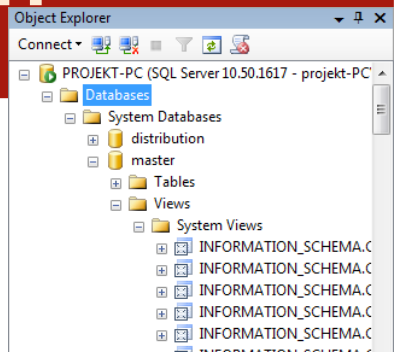
Pliki pomocnicze (NDF)

```
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
```

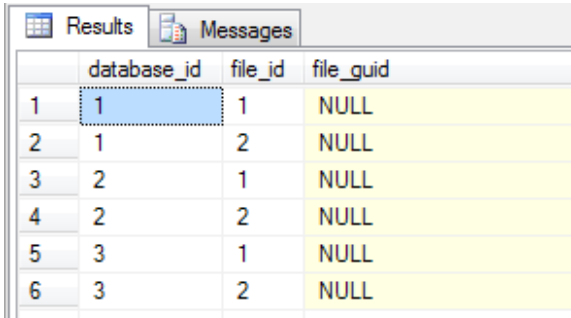
- Informacje o plikach fizycznych

```
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

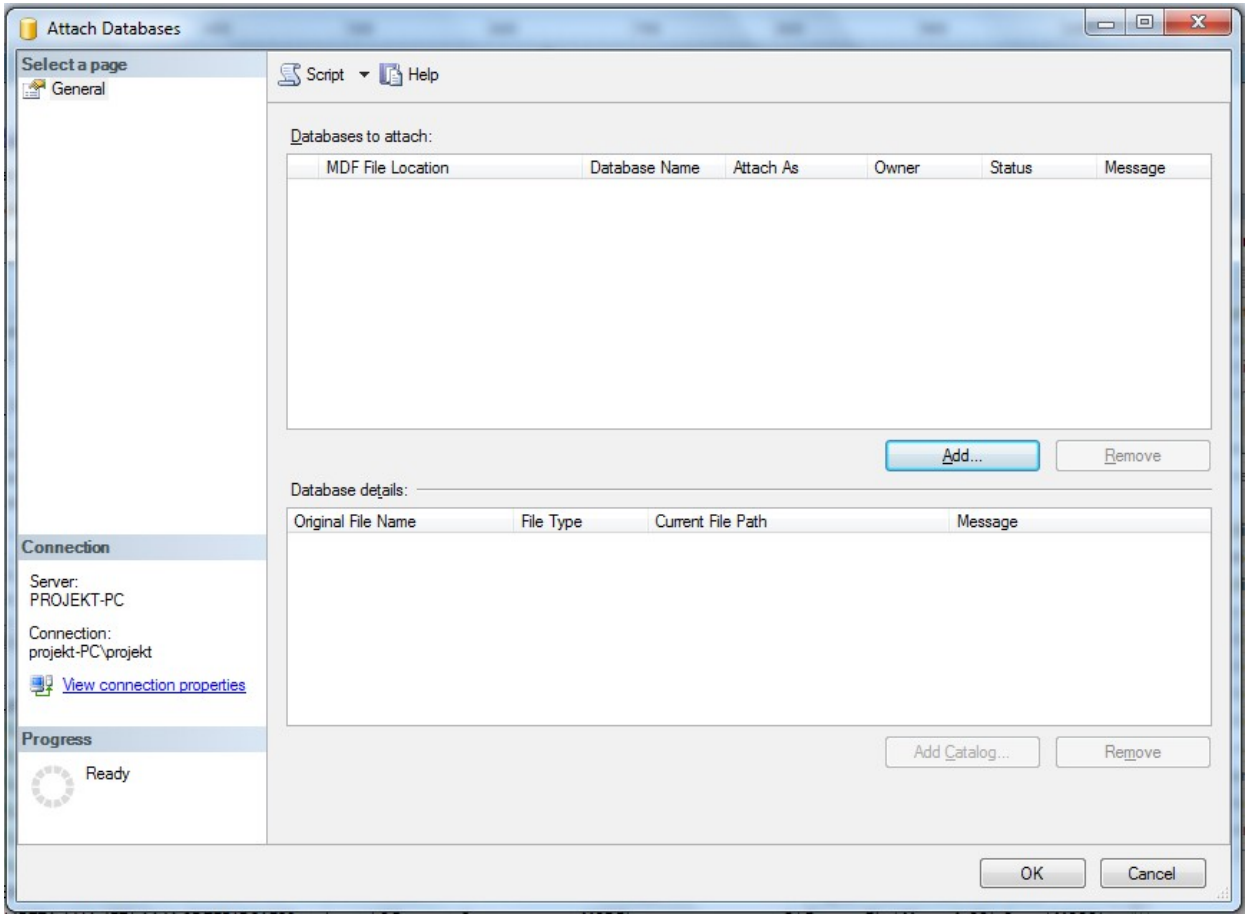


Results | Messages

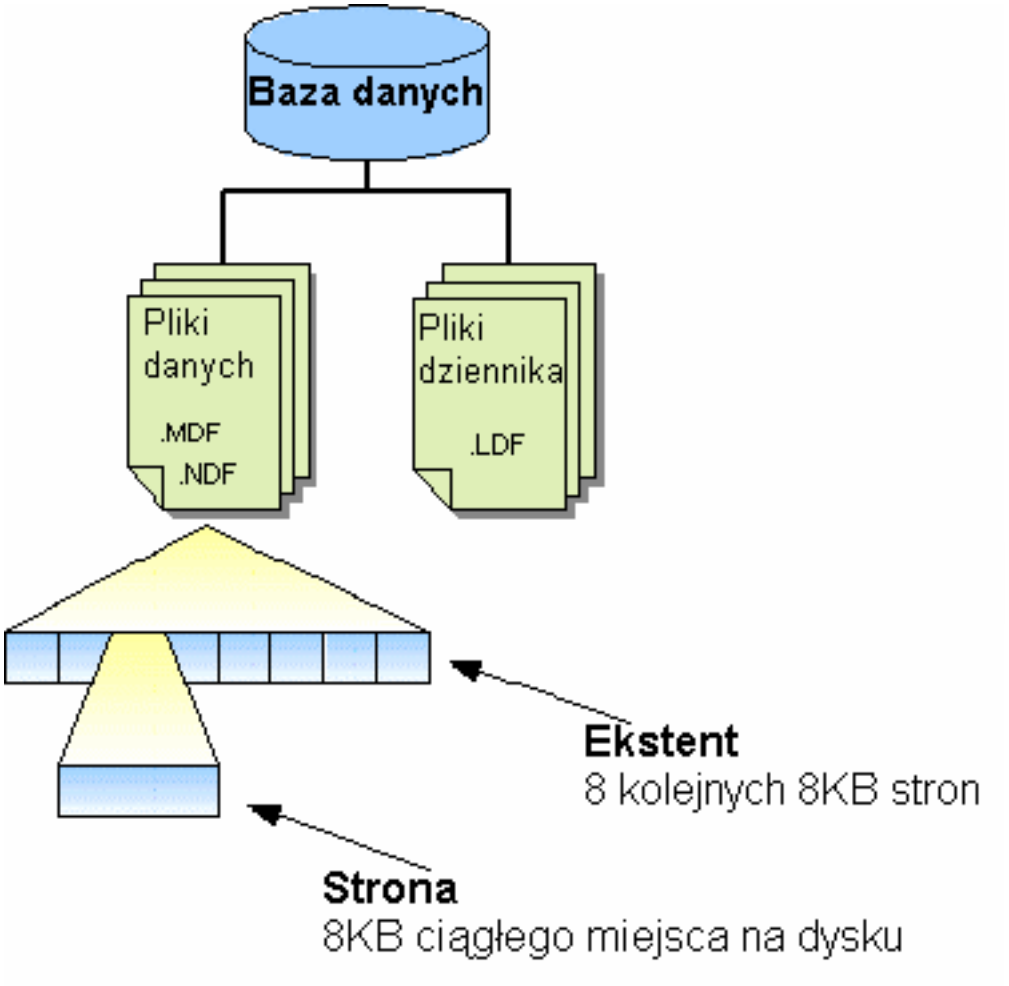
	database_id	file_id	file_guid
1	1	1	NULL
2	1	2	NULL
3	2	1	NULL
4	2	2	NULL
5	3	1	NULL
6	3	2	NULL

Dołączanie bazy danych

- Databases → Attach…



Podział bazy danych na pliki, ekstenty i strony

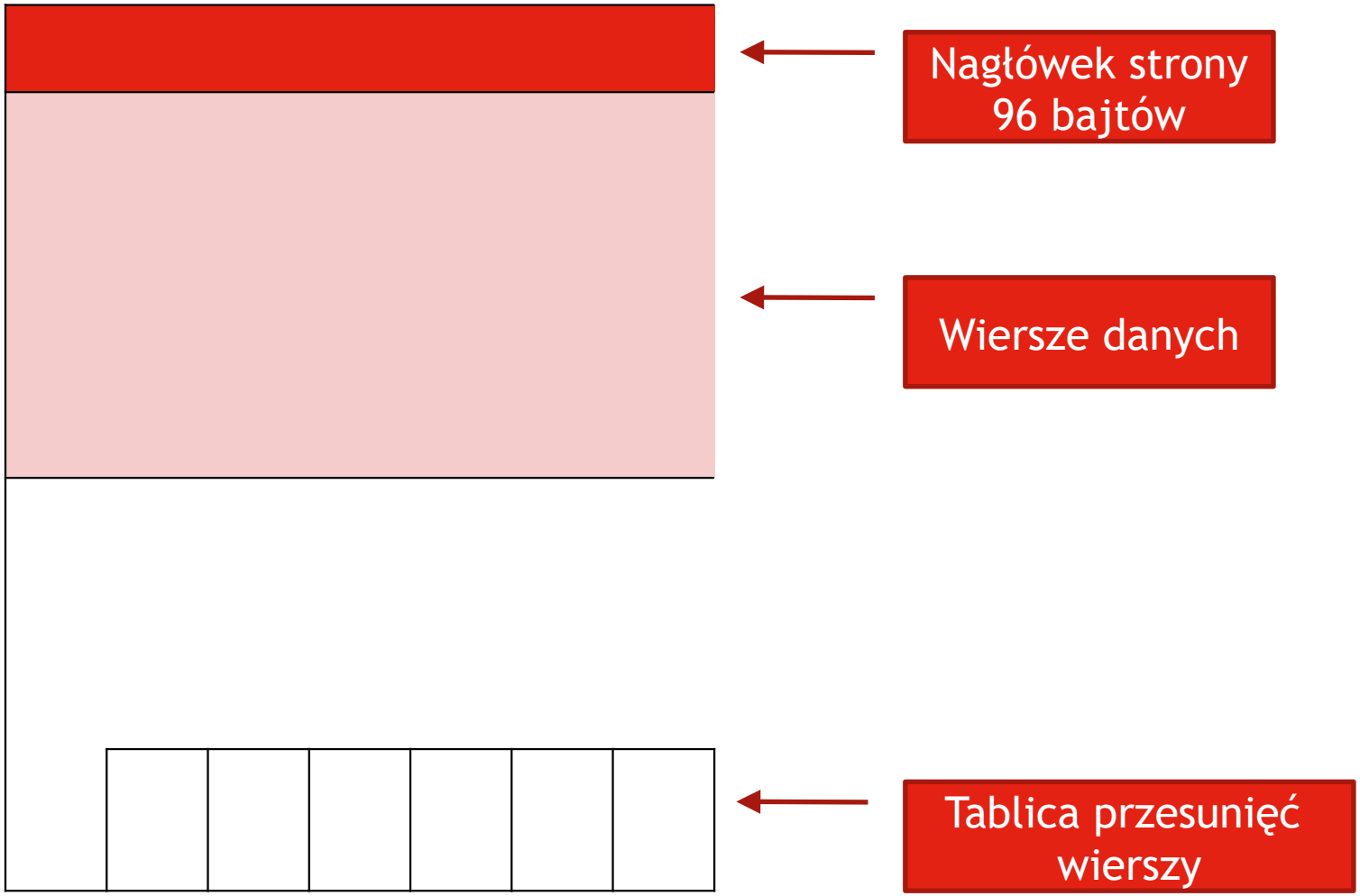


źródło: http://edu.pjwstk.edu.pl/wyklady/szb/scb/wyklad15/w15.htm

Rodzaje stron

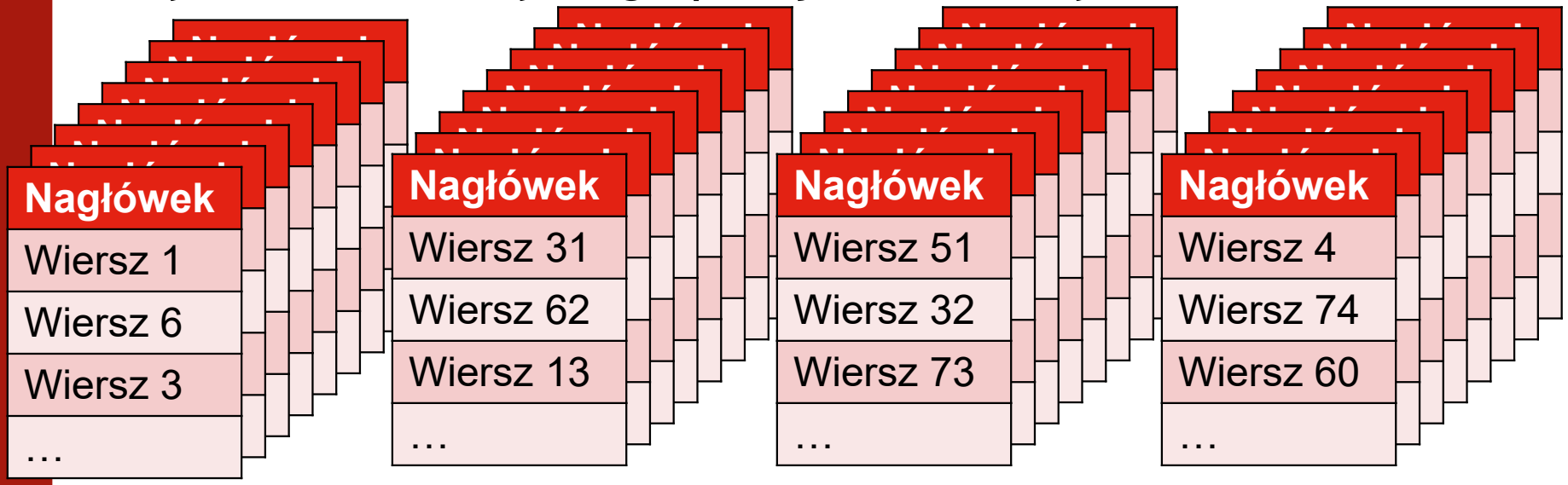
- **data** – wszystkie dane z wyjątkiem atrybutów typów LOB (ang. *Large OBject*)
- **index** – wpisy indeksów
- **text/image** – typy LOB: text, ntext, image, varchar(max), nvarchar(max), varbinary(max), xml
- **GAM** (*Global Allocation Map*), **SGAM** (*Shered* GAM), **IAM** (*Index Allocation Map*)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

Struktura strony danych



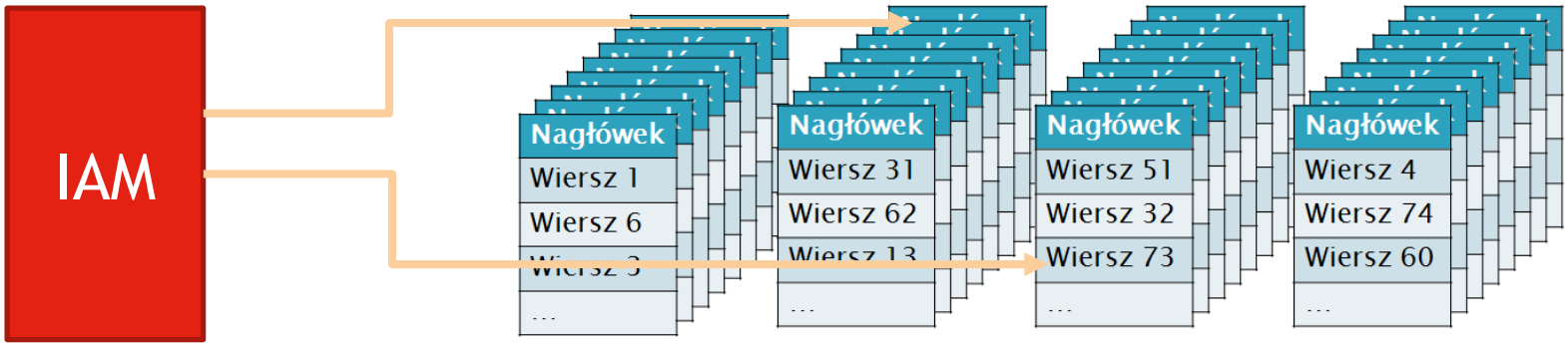
Sterta (ang. *heap*)

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



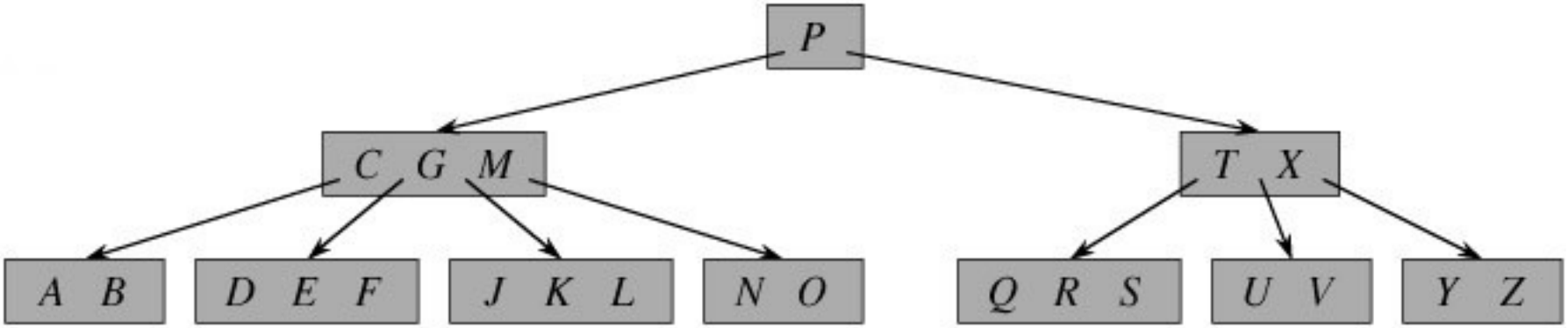
Fizyczna organizacja danych w SQL Server

- 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

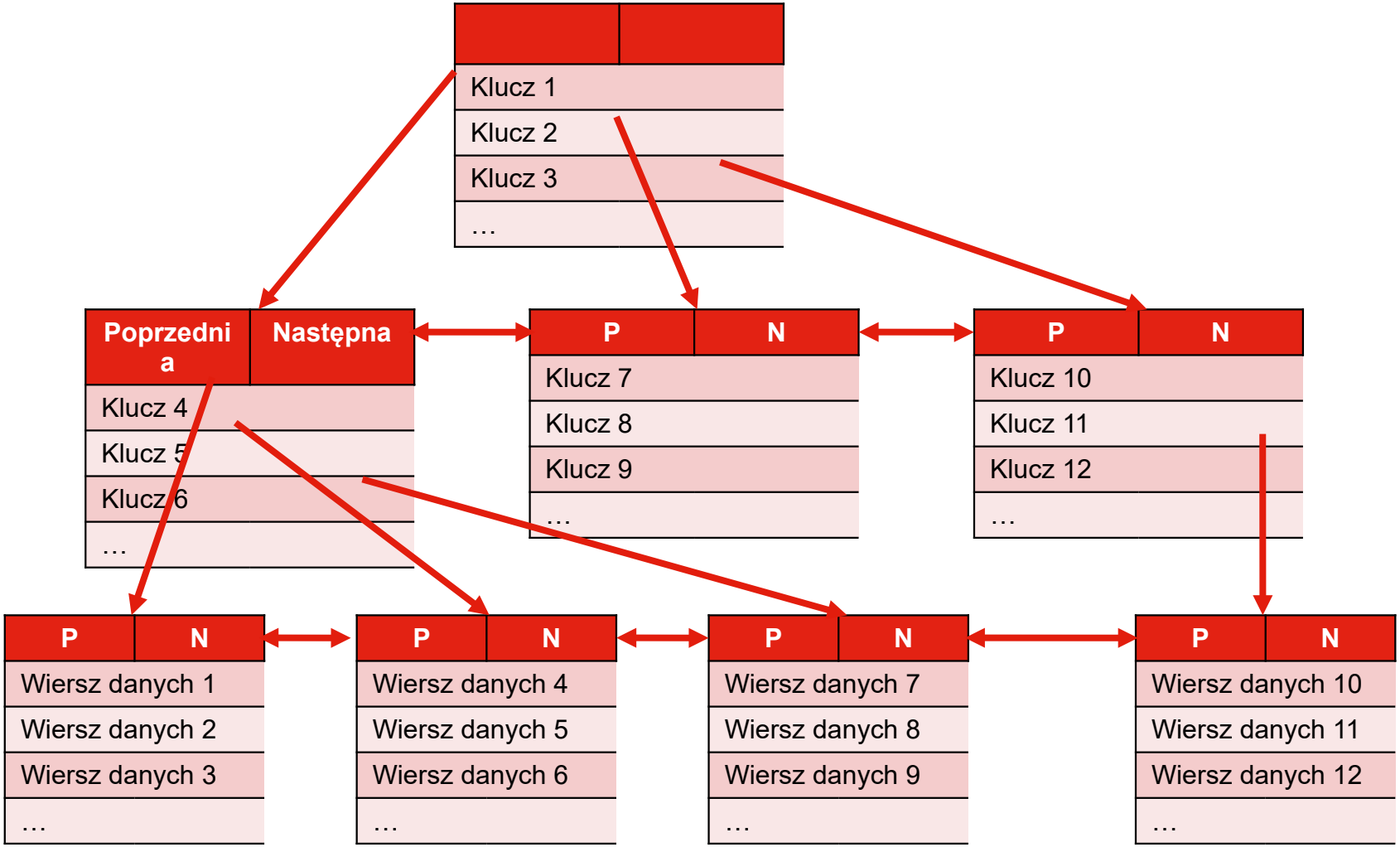
- Drzewo zbalansowane
- Węzły mogą mieć strukturę listy dwukierunkowej
- W praktyce do 5 poziomów



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?

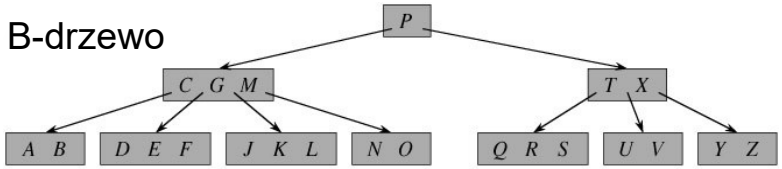
- Mała długość atrybutu (klucza)
- Wysoka selektywność (mało powtarzających się wartości klucza indeksu)
- Rzadko bądź wcale nie zmieniane wartości
- Wartości klucza dla kolejno dodawanych wierszy są rosnące

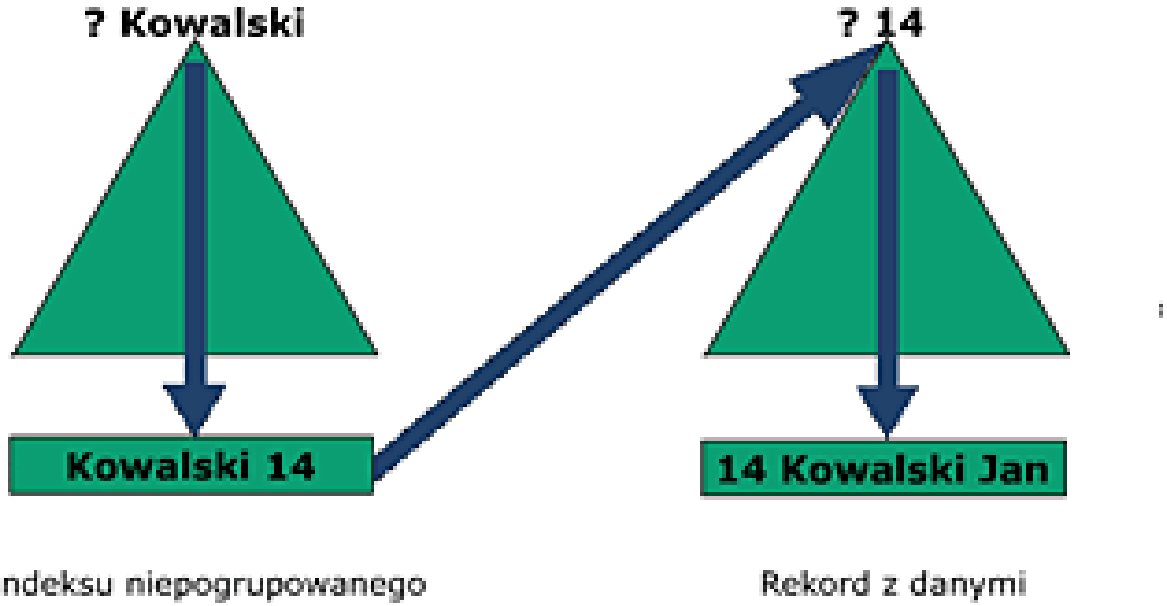
Indeks niezgrupowany

- 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

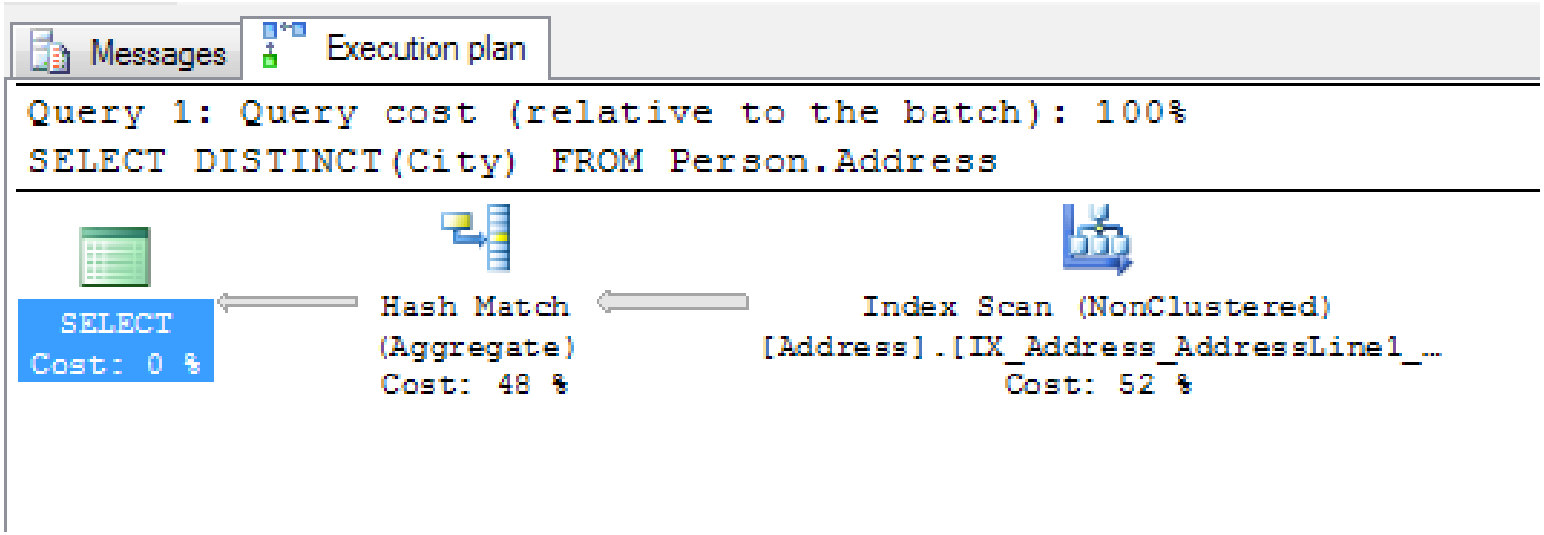




źródło: http://edu.pjwstk.edu.pl/wyklady/szb/scb/wyklad15/w15.htm

Kwestie wydajnościowe

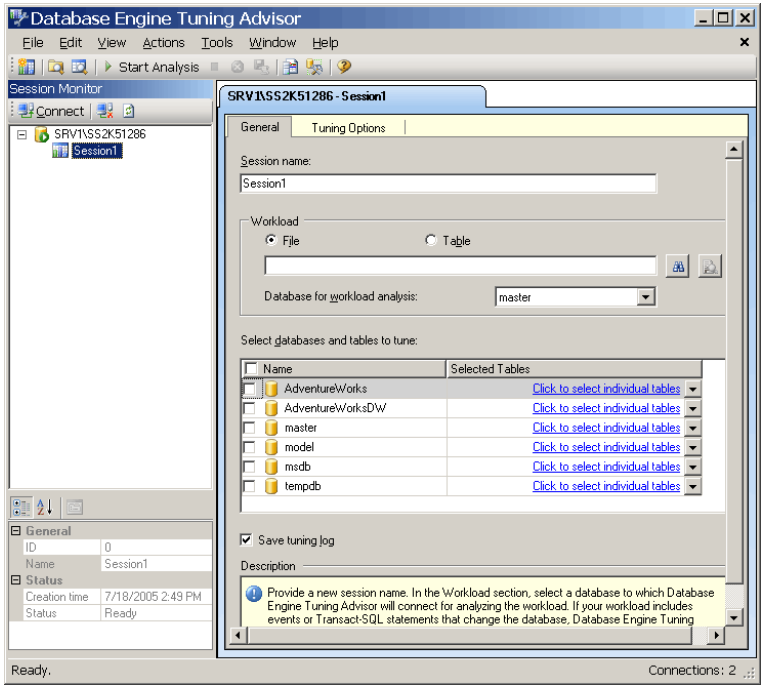
- Fizyczna organizacja danych wpływa na wydajność pracy BD
- Plan wykonania



PLAN WYKONANIA W MS SQL SERVER

Narzędzia wspierające optymalizację

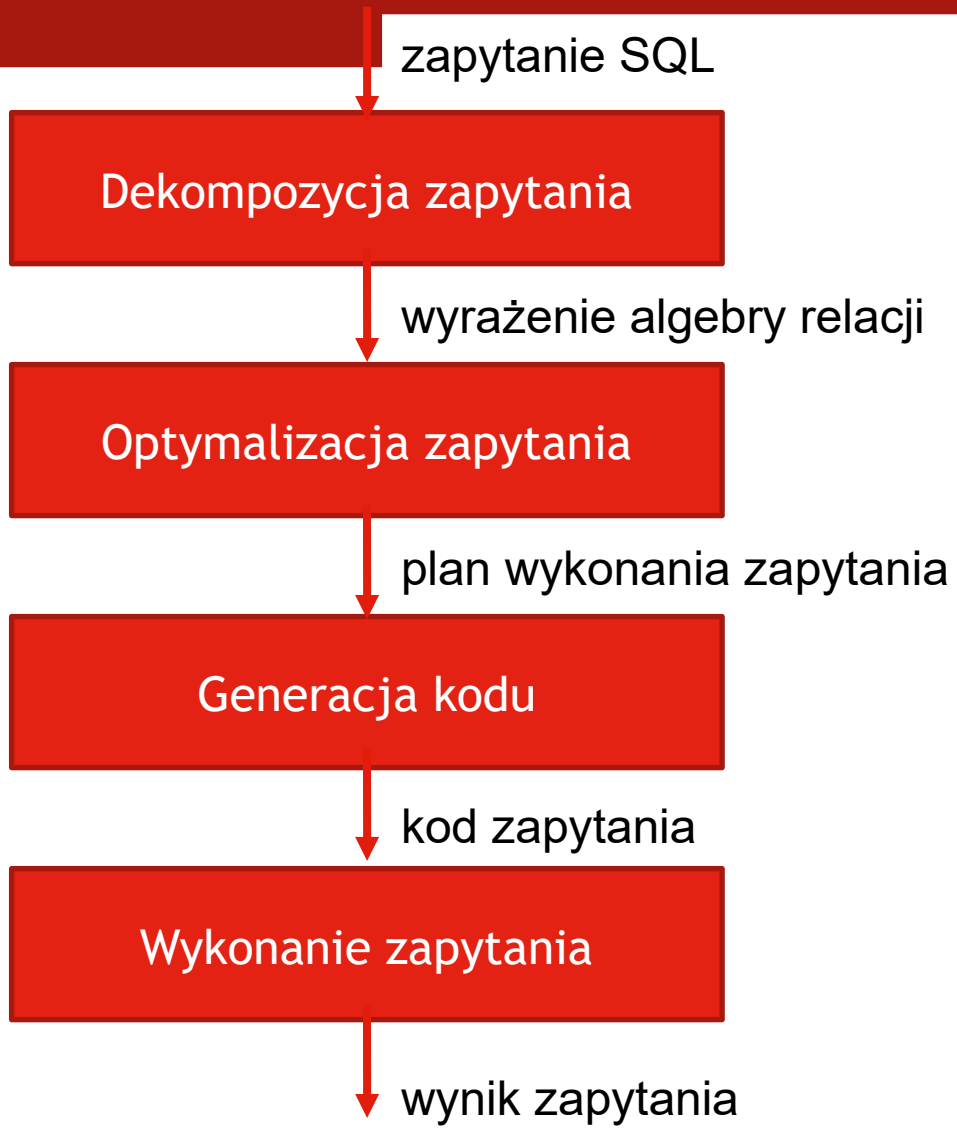
- MS SQL Server Management Studio
  - Plan wykonania
- Database Engine Tuning Advisor



Rola optymalizatora zapytań

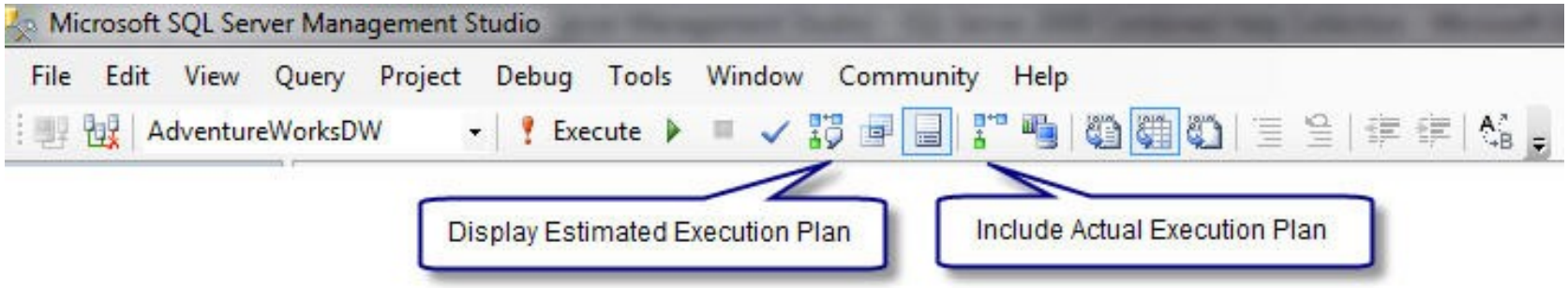
- T-SQL jest językiem deklaratywnym.
- Używając go, stwierdzamy **co** chcemy zrobić, a nie **jak** to zrobić.
- System zarządzania bazą danych posługując się **optymalizatorem zapytań** decyduje o sposobie wykonania zapytania.

Fazy przetwarzania zapytania



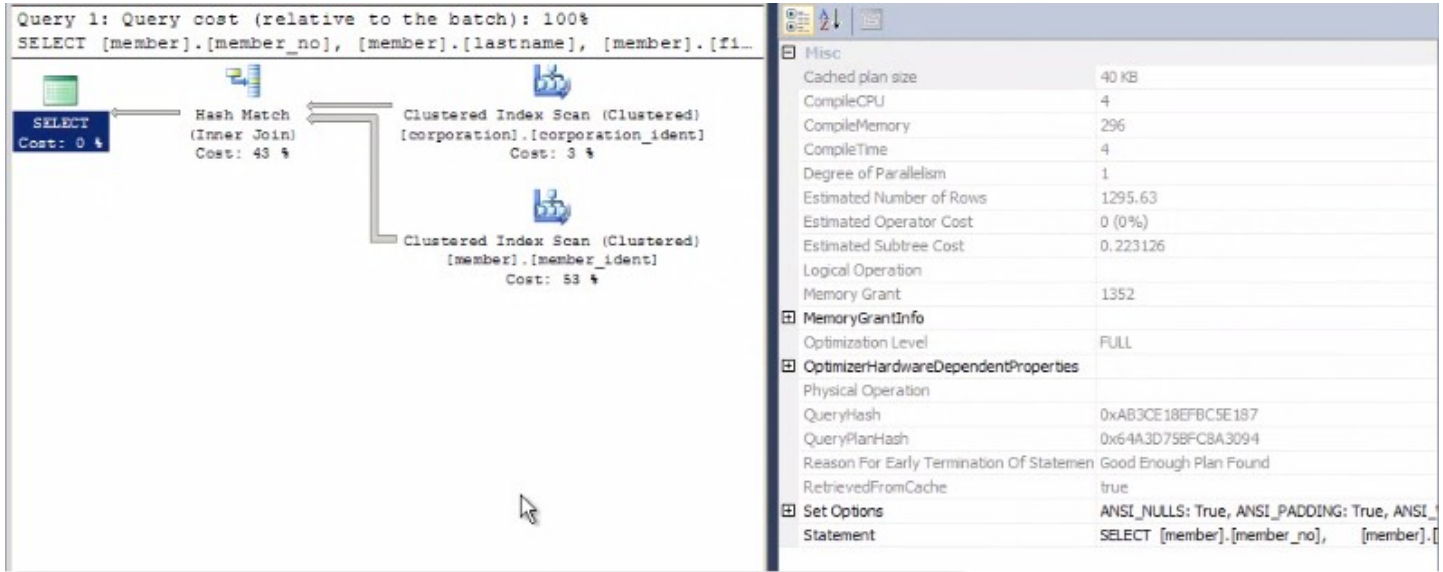
Plan wykonania MS SQL Server

- 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)



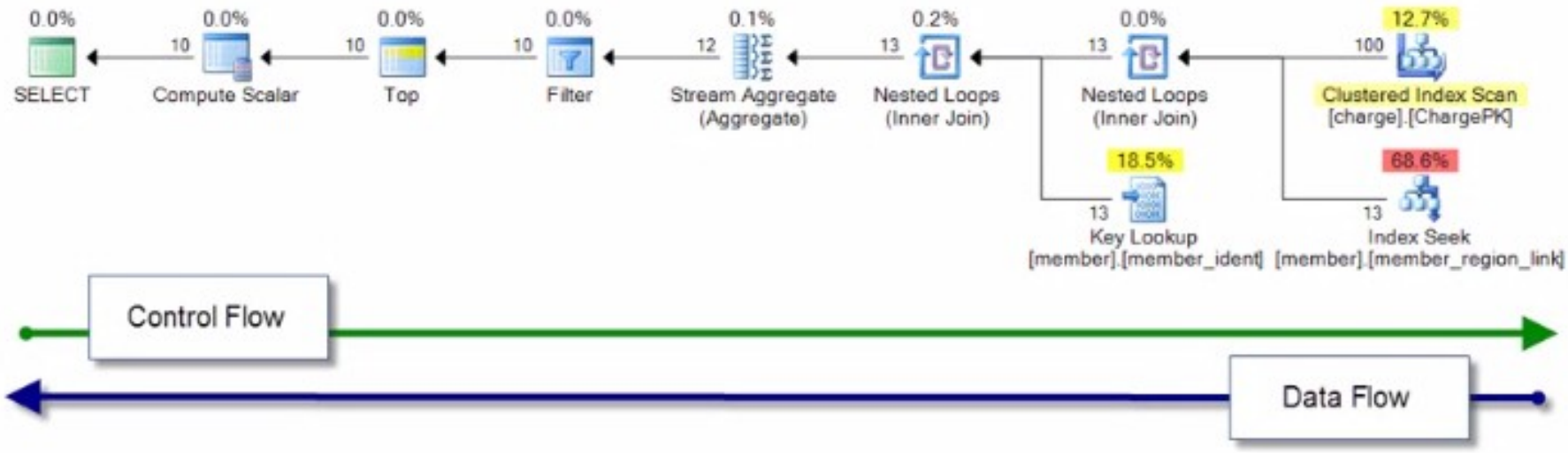
Ikony planu wykonania (wybór)

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

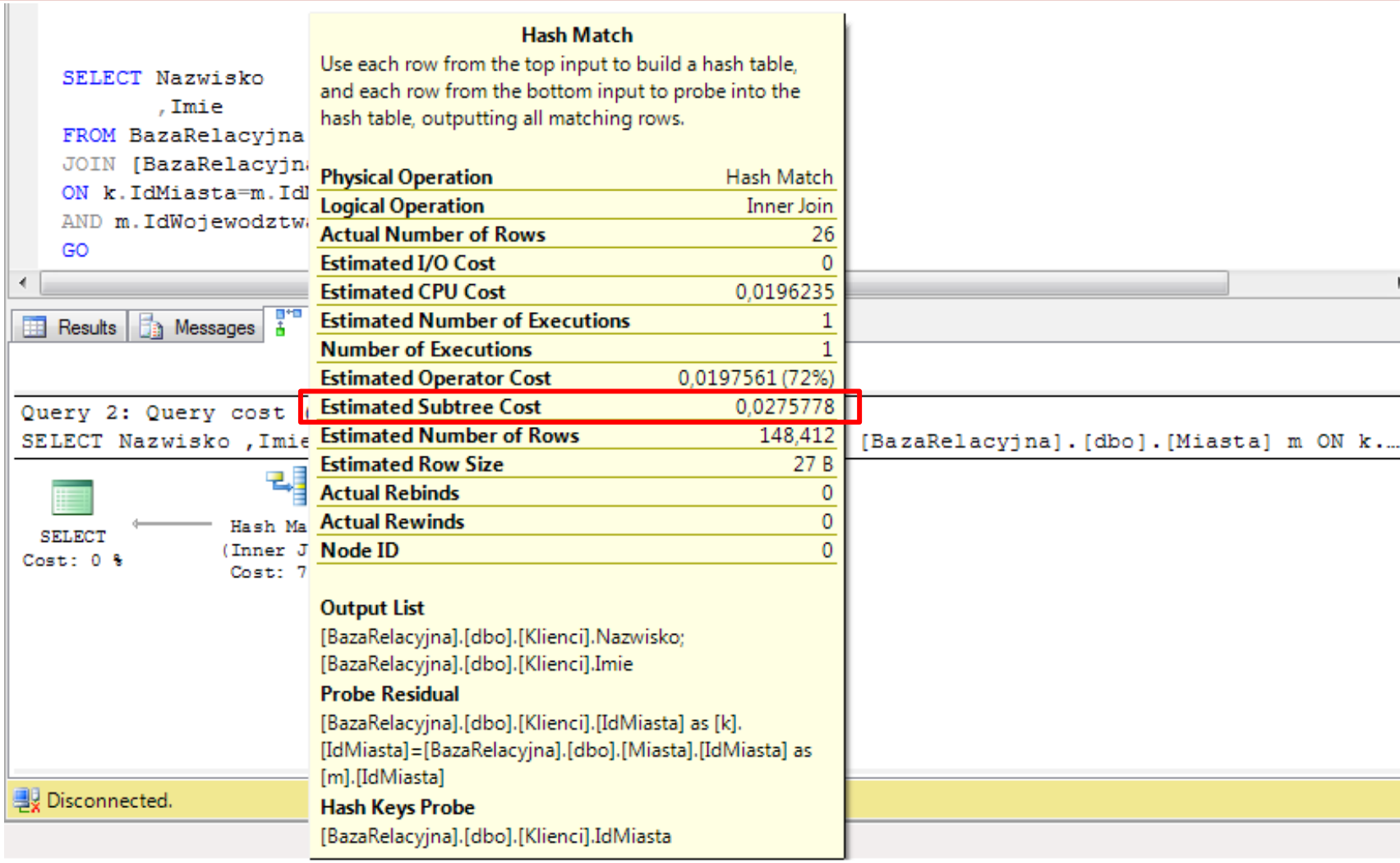
Ikony planu wykonania (cd.)

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

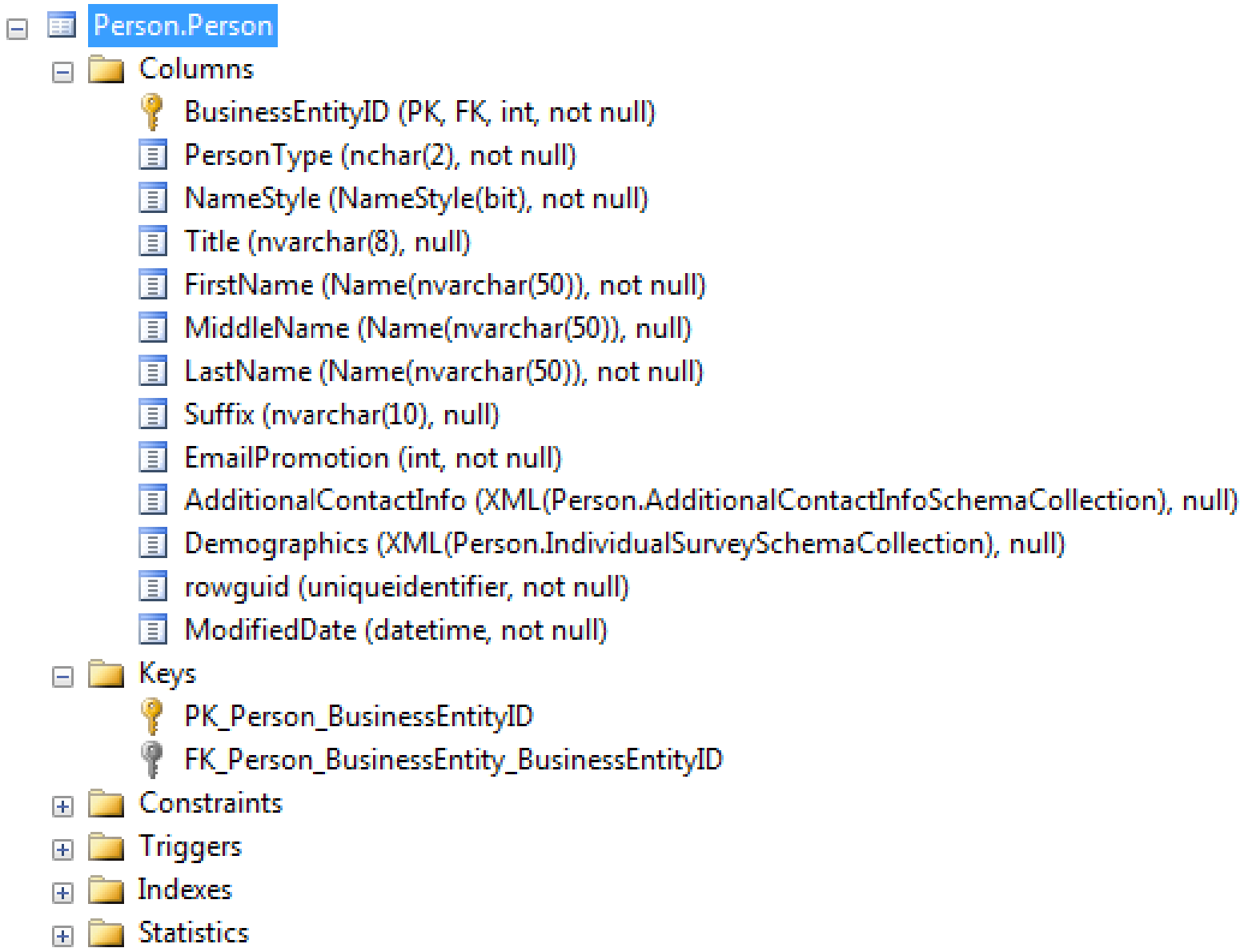
Interpretacja planu wykonania



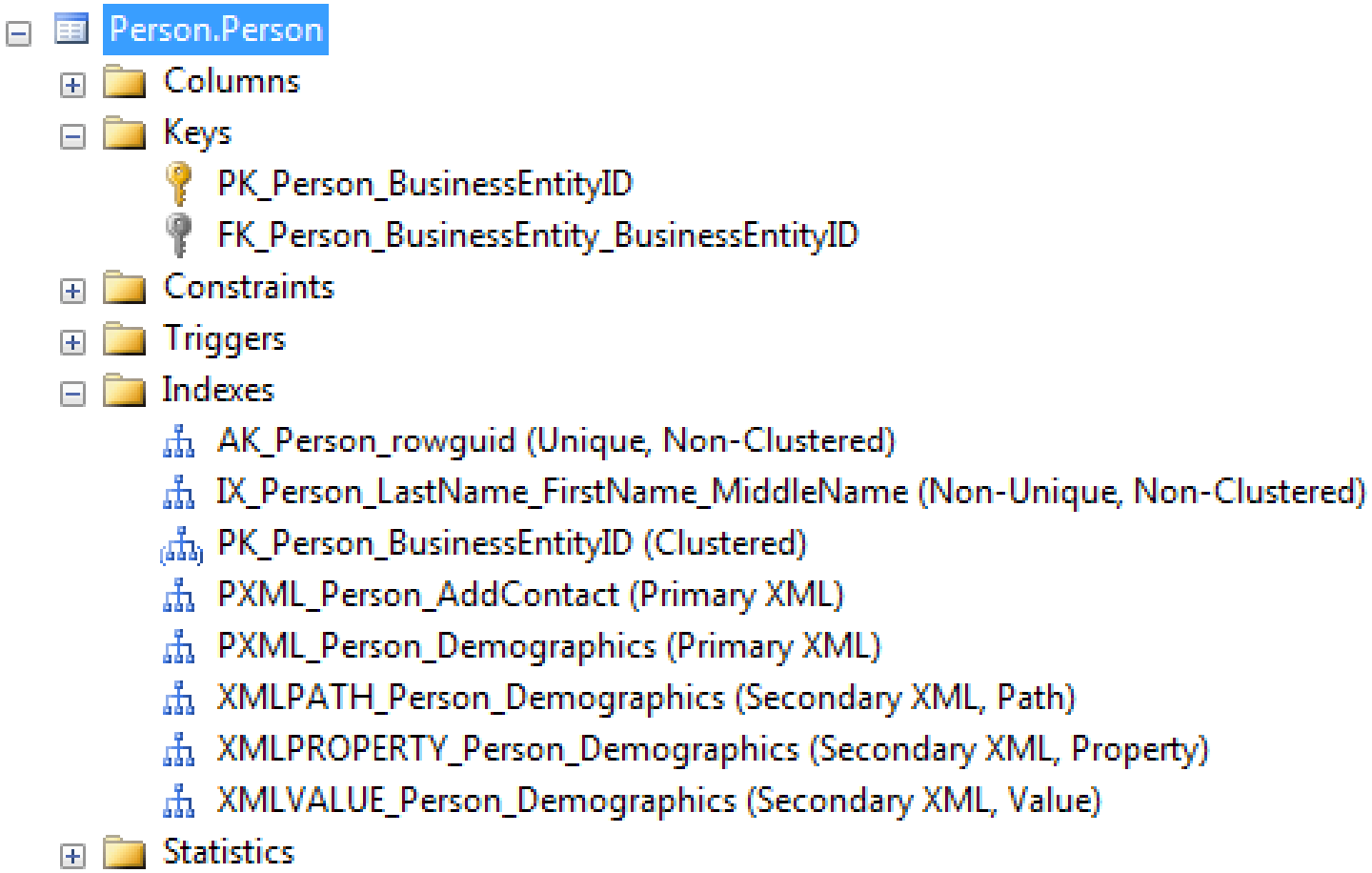
Okno podpowiedzi z informacjami o operatorze



Przykład z BD AdventureWorks2008



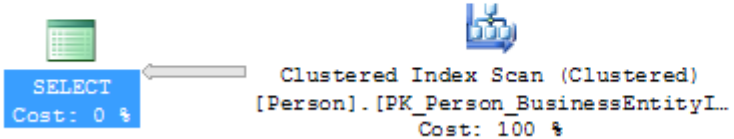
Indeksy w tabeli Person.Person



Przykład 1: Proste plany wykonania

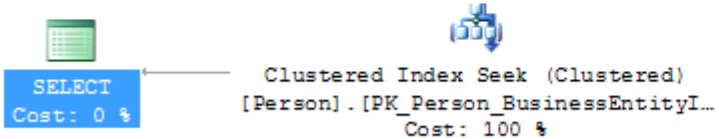
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [AdventureWorks2008].[Person].[Person]
```



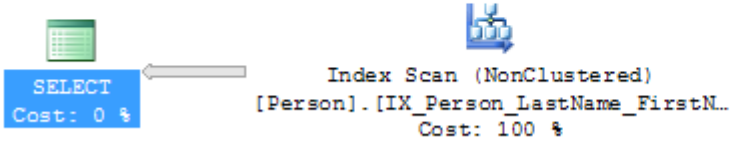
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT * FROM [AdventureWorks2008].[Person].[Person] WHERE [BusinessEntityID]=@1
```



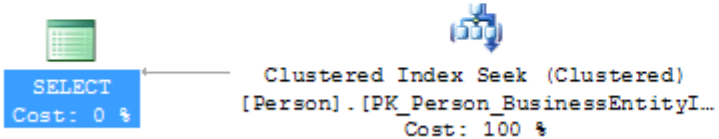
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person]
```

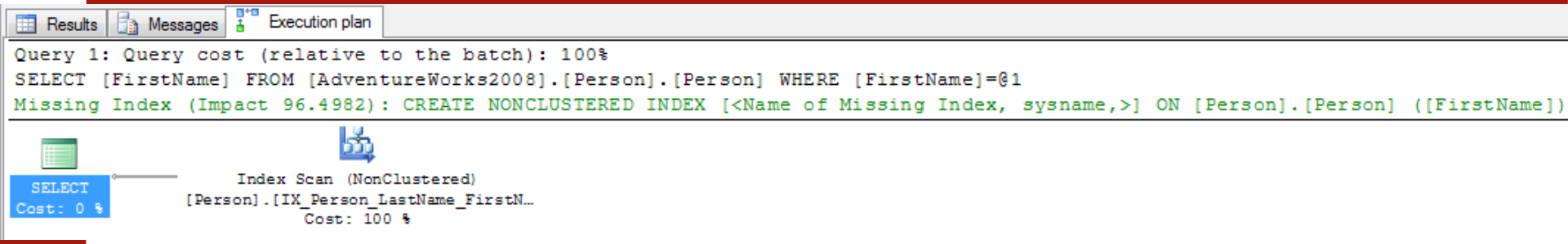


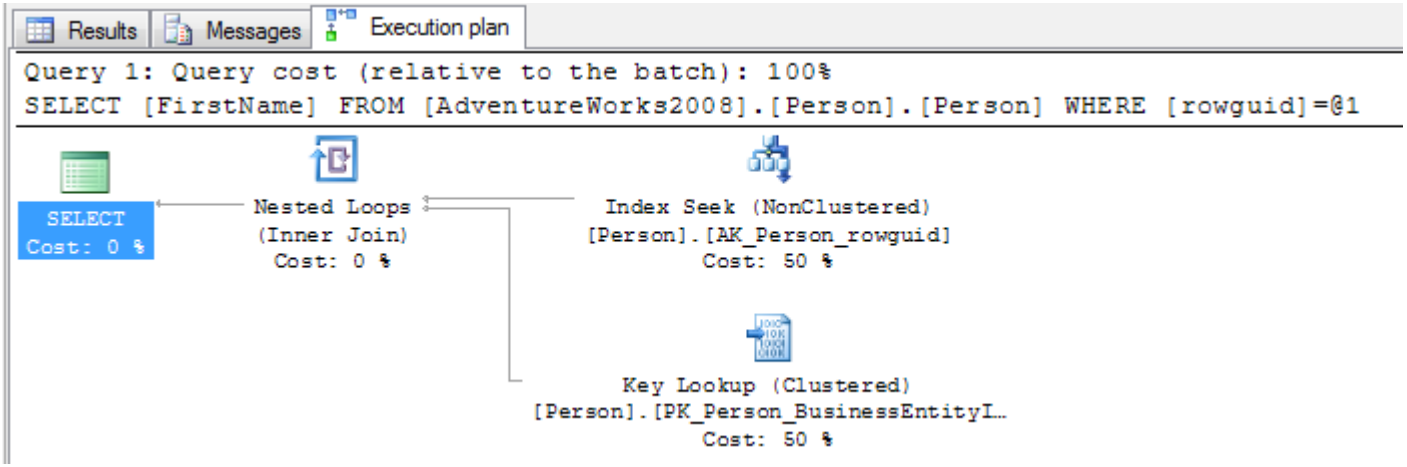
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT [FirstName] FROM [AdventureWorks2008].[Person].[Person] WHERE [BusinessEntityID]=@1
```



Przykład 1 cd.: Proste plany wykonania





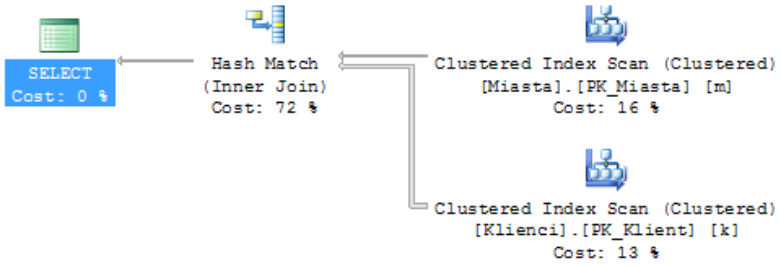
Dwie wersje zapytania SQL

```
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
```

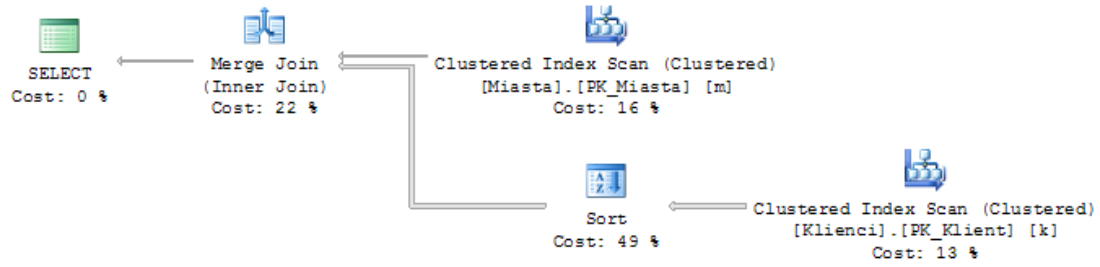
Uzyskane plany wykonania

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
```



TRANSACT-SQL

Transact-SQL (T-SQL)

- Został stworzony przez Sybase; później prawa kupiła firma Microsoft i wykorzystuje ten język w kolejnych wersjach MS SQL Server

Klasyczny język SQL a T-SQL

- SQL – język deklaratywny w SZBD
- Sekwencje poleceń
- Nie ma konstrukcji sterujących: instrukcji warunkowych, pętli, zmiennych

- T-SQL – rozszerzenie języka ale nie „pełny” język programowania
- Programowanie zadań po stronie serwera BD
- W T-SQL osadzony jest standardowy język SQL

T-SQL

- Rozszerzenie języka SQL umożliwiające tworzenie konstrukcji takich jak:
  - pętle, instrukcje warunkowe oraz zmienne
  - tabele tymczasowe.
- Zastosowanie:
  - do tworzenia wyzwalaczy,
  - procedur i funkcji składowanych w bazie.

Zmienne w T-SQL

- Wymóg jawnej deklaracji
- Nazwa zmiennej lokalnej musi rozpoczyna się od znaku @
  - zmienna globalna od @@
- Typy zmiennych mogą być takie, jak wbudowane MS SQL Servera:
  - CHAR, VCHAR, NCHAR, NVCHAR, INT, DATETIME, …
- Przykład:

```
DECLARE @licznik int = 1
```

Konwersja typów

- Polecenie **CAST**

```
CAST (wyrażenie AS typ_danych)
```

- Polecenie **CONVER**

```
CONVEERT (typ_danych[(rozmiar)], wyrażenie [,styl])
```

- Przykłady:

```
PRINT CAST(12.34 AS int)
PRINT CONVERT(int 12.34)
PRINT CONVERT(char(24), GETDATE(), 9)
```

Instrukcje sterujące

- Instrukcja warunkowa:

```
IF wyrażenie_logiczne {polecenie_SQL|blok_poleceń}
ELSE {polecenie_SQL|blok_poleceń}
```

- Pętle

```
WHILE wyrażenie_logiczne
{polecenie_SQL|blok_poleceń}
```

Przykład użycia instrukcji warunkowej

```
DECLARE @i int
SET @i = 1
IF @i = 1
BEGIN
    SELECT 'tak'
END
ELSE
BEGIN
     SELECT 'nie'
END
```

Przykład użycia pętli

```
SET @licznik = 1
WHILE @licznik < 100
BEGIN
   INSERT INTO #tmp(liczba) VALUES (@licznik)
   SET @licznik=@licznik+1
END
```

Przykład tworzenia widoku

```
CREATE VIEW Klienci
AS
SELECT imie, nazwisko
FROM BazaFirmy.dbo.Klienci_firmy
WHERE rodzaj = 'staly'
```

Przykład tworzenia wyzwalacza

```
CREATE TRIGGER t_zamowienie
ON dbo.Orders
FOR INSERT, UPDATE, DELETE
AS EXEC master..xp_sendmail
  'kontroler1', 'Zmieniono zawartość
  tabeli Orders - proszę uaktualnić
  zestawienia'
GO
```

Obsługa błędów

```
BEGIN TRY
  Instrukcje
END TRY
BEGIN CATCH
  Instrukcje_obsługi_błędów
ENT CATCH
```

Funkcje

- Konfiguracyjne
- Kursora
- Daty i czasu
- Matematyczne
- Metadanych
- Bezpieczeństwa
- Operujące na łańcuchach znaków
- Systemowe
- Statystyki systemowej
- Tekstu i obrazu

rocławska

Politechnika Wrocławska

Przegląd innych systemów baz danych

MySQL, SQLite, PostgreSQL

Wersje BD

Produkt	Producent	Najnowsza wersja	Licencja
Oracle Database	Oracle	18c	komercyjna
Microsoft SQL Server	Microsoft	14.00	Microsoft EULA
MySQL	Oracle	8.0.15	GPL lub komercyjna
MariaDB	Monty Program Ab	10.3.14	GPL v. 2
PostgreSQL	PostgreSQL Global Development Group	11.2	PostgreSQL
SQLite	D. Richard Hipp	3.27.2	Public domain
Informix	IBM	12.10.xC12	EULA
DB2	IBM	12.1	EULA

Duże bazy danych

- Microsoft SQL Server 2017, Azure SQL Database
- Oracle 11g, 12c, 18c
- 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 wielodstępowych 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: 8.0.15
  - 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

źródło: http://osworld.pl/czym-sie-rozni-innodb-od-myisam/

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

źródło: http://osworld.pl/czym-sie-rozni-innodb-od-myisam/

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

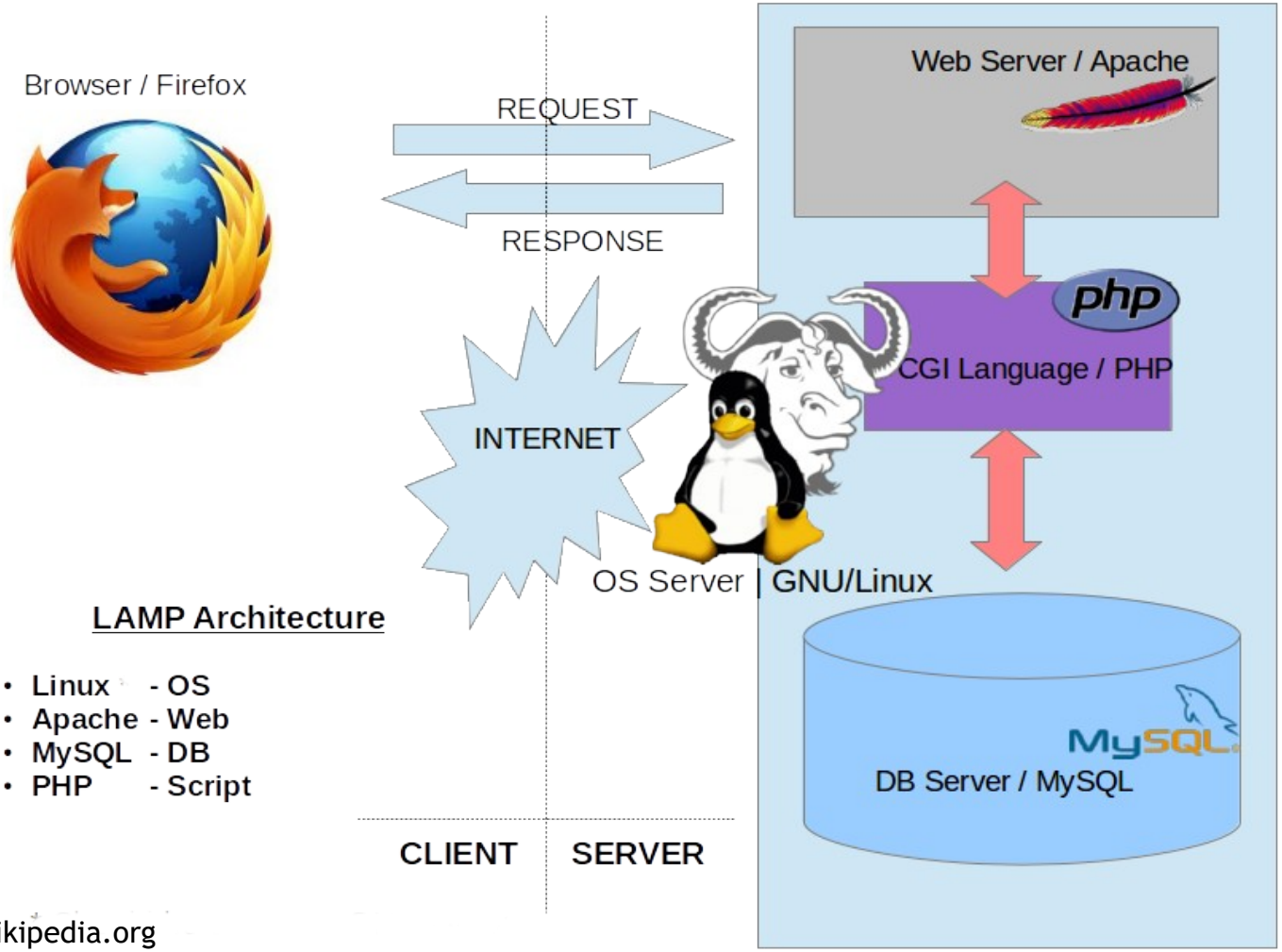
- Zalety
  - + obsługa transakcji
  - + gwarantuje integralność danych
  - + lepiej sprawuje się podczas replikacji typu master - slave
- Wady
  - – wolniejszy odczyt danych
  - – trudniejsze wykonywanie backupów

źródło: http://osworld.pl/czym-sie-rozni-innodb-od-myisam/

Zestaw oprogramowania

- LAMP – **L**inux + **M**ySQL/MariaDB + **P**HP/Perl/Python
- WAMP – **W**indows + **A**pache + **M**ySQL + **P**HP
- MAMP – **M**ac OS X + **A**pache + **M**ySQL + **P**HP
- XAMPP – X (Cross-platform) **A**pache, **M**ySQL, **P**HP, **P**erl

Uproszczony schemat architektury LAMP



źródło: wikipedia.org

Funkcje MySQL w PHP

- http://php.net/manual/en/ref.mysql.php

- **mysql_close** – Close MySQL connection
- **mysql_connect** – Open a connection to a MySQL Server
- **mysql_create_db** – Create a MySQL database
- **mysql_db_query** – Selects a database and executes a query on it
- **mysql_error** – Returns the text of the error message from previous MySQL operation
- **mysql_fetch_array** – Fetch a result row as an associative array, a numeric array, or both
- **mysql_info** – Get information about the most recent query
- **mysql_num_rows** – Get number of rows in result
- **mysql_pconnect** – Open a persistent connection to a MySQL server
- **mysql_query** – Send a MySQL query

Łączenie z serwerem i bazą danych w PHP

```php
<?php
// nawiazujemy polaczenie
$connection = @mysql_connect('localhost', 'uzytkownik', 'haslo')
// w przypadku niepowodznie wyświetlamy komunikat
or die('Brak połączenia z serwerem MySQL.<br />Błąd: '.mysql_error());
// połączenie nawiązane ;-)
echo "Udało się połączyć z serwerem!<br />";
// nawiązujemy połączenie z bazą danych
$db = @mysql_select_db('nazwa_bazy', $connection)
// w przypadku niepowodzenia wyświetlamy komunikat
or die('Nie mogę połączyć się z bazą danych<br />Błąd:
   '.mysql_error());
// połączenie nawiązane ;-)
echo "Udało się połączyć z bazą danych!";
// zamykamy połączenie
mysql_close($connection);
?>
```

Tworzenie bazy danych

```
<?php
// otwiera połączenie
$polaczenie = mysql_connect("localhost", "juzer", "samopas");
// wybiera bazę
mysql_select_db("test",$polaczenie);
// tworzy instrukcję SQL
$sql = "CREATE TABLE tabelaProbna (id int not null primary key
   auto_increment,
   poleProbne varchar (75))";
// wykonuje instrukcję SQL
$wynik = mysql_query($sql, $polaczenie) or die(mysql_error());
// wyświetla identyfikator wyniku
echo $wynik;
?>
```

Wstawianie danych do bazy

```
<?php
// otwiera połączenie
$polaczenie = mysql_connect("localhost", "juzer",
   "samopas");
// wybiera bazę
mysql_select_db("test",$polaczenie);
// tworzy instrukcję SQL
$sql = "INSERT INTO tabelaProbna values ('', 'jakaś
   wartość')";
// wykonuje instrukcję SQL
$wynik = mysql_query($sql, $polaczenie) or
   die(mysql_error());
// wyświetla identyfikator wyniku
echo $wynik;
?>
```

Pobieranie danych z bazy

```php
<?php
// otwiera połączenie
$polaczenie = mysql_connect("localhost", "juzer", "samopas");
// wybiera bazę
mysql_select_db("test",$polaczenie);
// tworzy instrukcję SQL
$sql = "SELECT * FROM tabelaProbna";
// wykonuje instrukcję SQL
$wynik = mysql_query($sql, $polaczenie) or die(mysql_error());
//przechodzi przez każdy wiersz zbioru wyników i wyświetla dane
while ($nowaTablica = mysql_fetch_array($wynik)) {
   // nadaje polom nazwy
   $id  = $nowaTablica['id'];
   $poleProbne = $nowaTablica['poleProbne'];
   //wyświetla rezultaty na ekranie
   echo "Identyfikator to $id a tekst w polu to $poleProbne <br>";
}
?>
```

Narzędzia administracyjne

- phpMyAdmin – za pomocą przeglądarki internetowej
- MySQL Workbench

SQLITE

SQLite

- Bezserwerowa relacyjna baza danych
- Pierwsze wydanie: 2000 r.
- Licencja: Public domain
- Aktualna wersja: 3.27.2

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.

Przykład użycia biblioteki w C

```
#include <stdio.h>
#include <sqlite3.h>
static int callback(void *nic, int argc, char **argv, char **kol) {
int i;
for(i=0; i<argc; i++)
  printf("%s = %s\n", kol[i], argv[i] ? argv[i] : "NULL");
printf("\n");
return 0;
}

int main(int argc, char **argv) {
sqlite3 *db;
int rc;
rc = sqlite3_open("../demo.db", &db);
rc = sqlite3_exec(db, "SELECT * FROM pracownicy", callback, 0, NULL);
sqlite3_close(db);
return 0;
}
```

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: 11.2

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

rocławska

Politechnika Wrocławska

Hurtownie danych.

Cele i techniki replikacji oraz fragmentacji danych

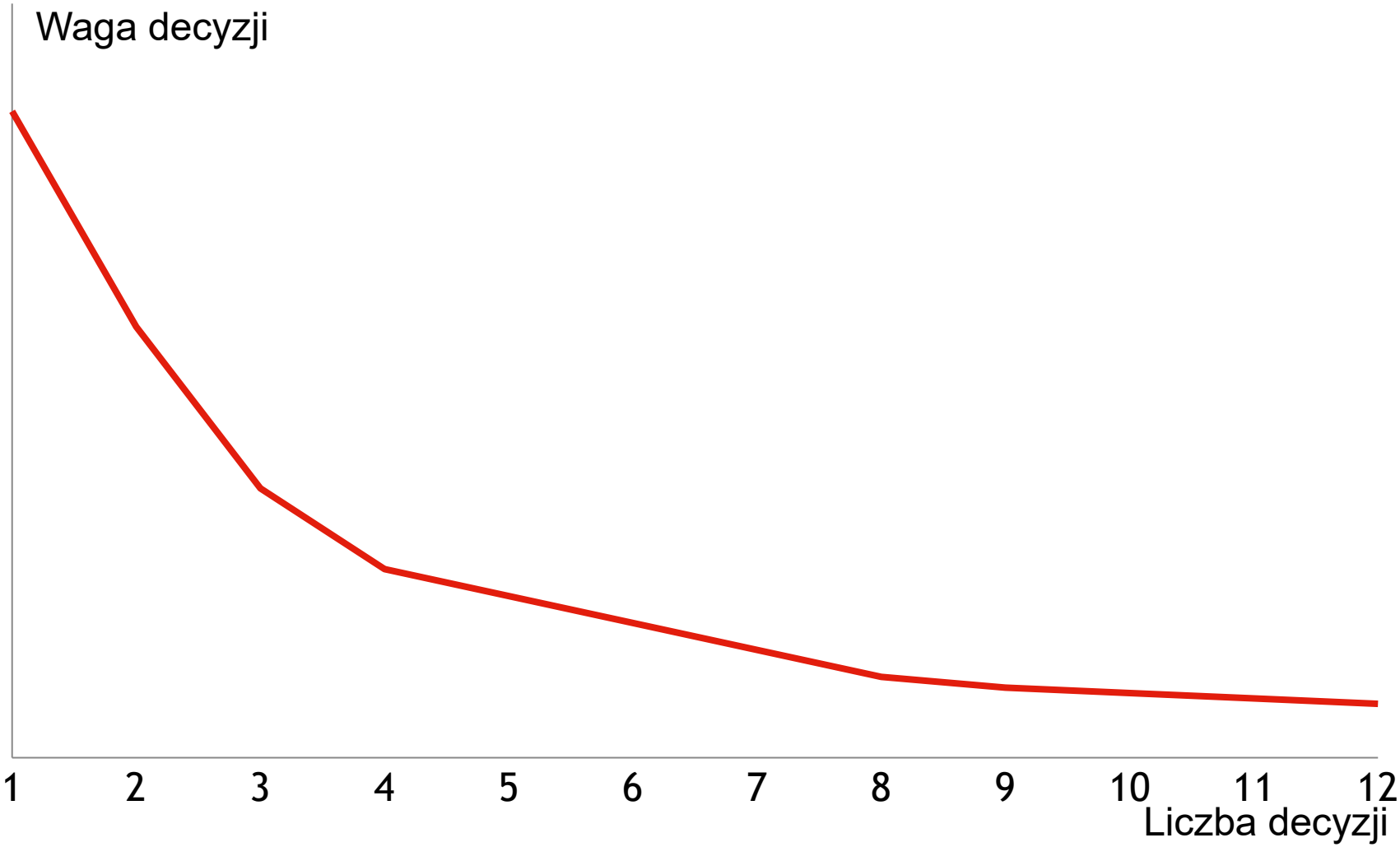
Serwery lustrzane, obiekty partycjonowane

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 liczba 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 sie go w technikach związanych z **Data Mining'iem**.

Zależność wagi i liczby decyzji



Hurtownia danych

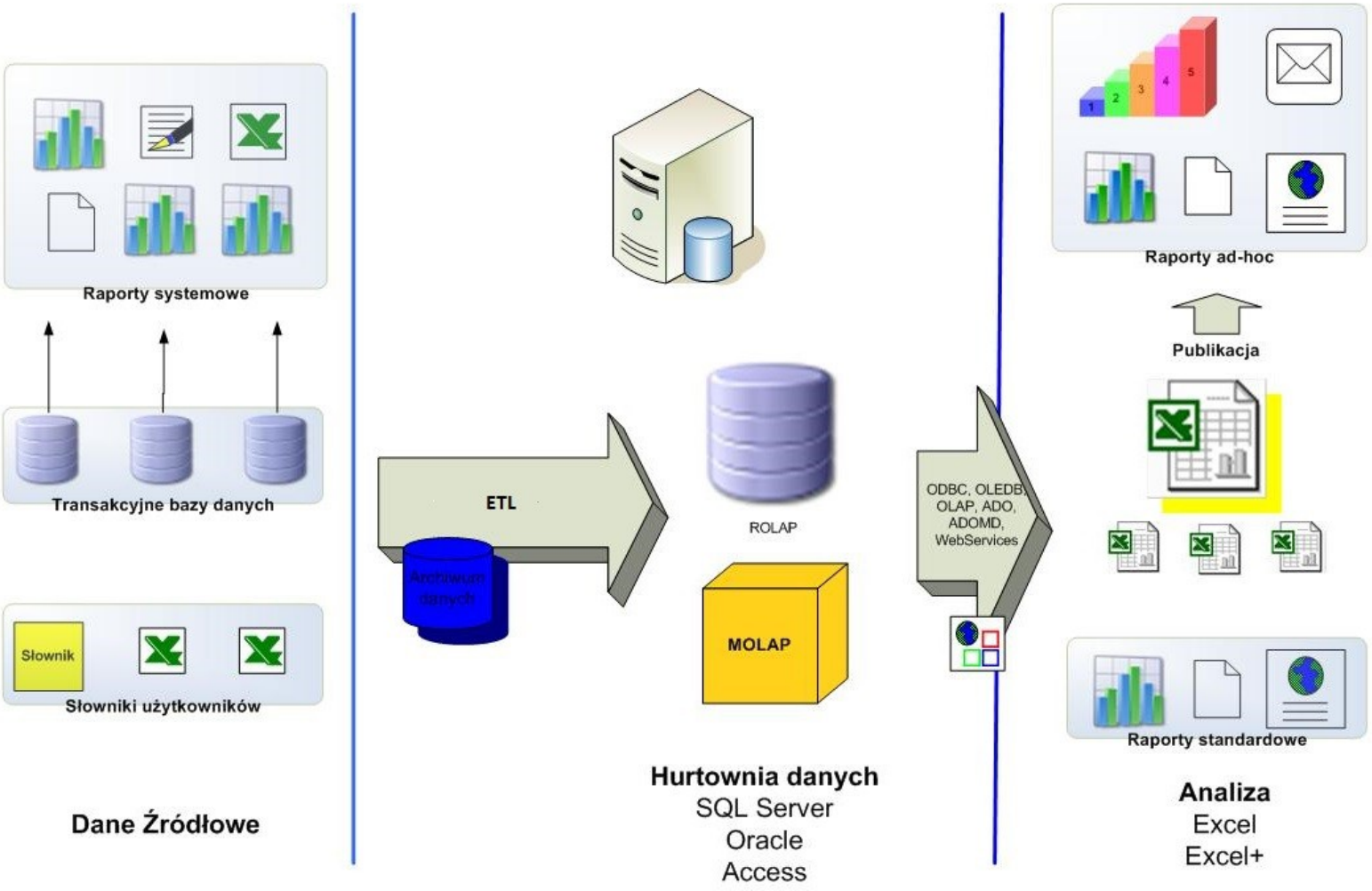
- ang. *Data Warehouse* - DW
- Analityczna baza danych wykorzystywana jako podstawa systemu wspomagającego podejmowanie decyzji.

- ang. *Data mining* - eksploracja danych, drążenie danych

Cele stosowania hurtowni danych

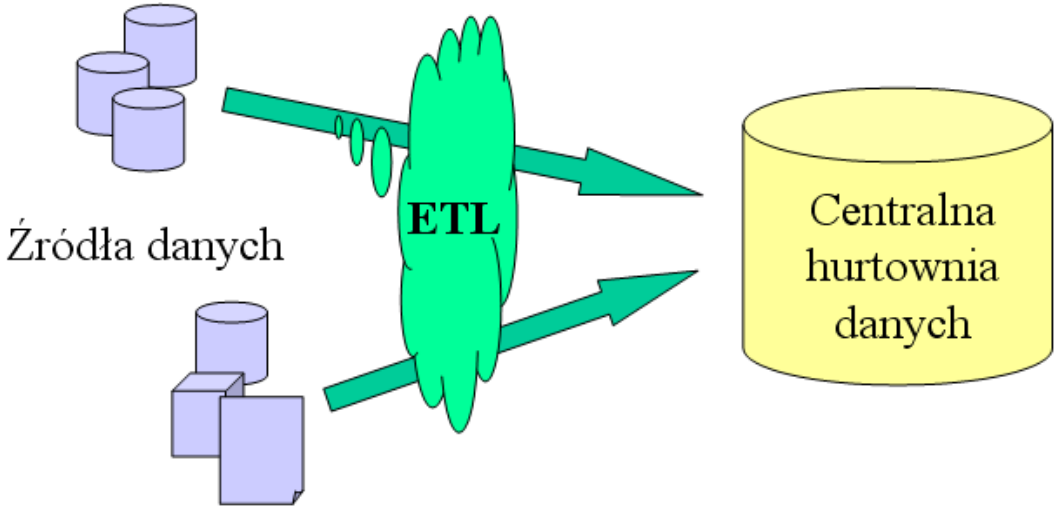
- Przetwarzanie analityczne (OLAP)
- Wspomaganie decyzji (DSS)
- Archiwizacja danych
- Analiza efektywności
- Wsparcie dla systemów CRM (np. poprzez precyzyjne dobieranie strategii marketingowych na podstawie danych o klientach i sprzedaży)

Architektura hurtowni danych



Proces integracji danych - ETL

- *Extract, Transform, Load*



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

Model danych MOLAP

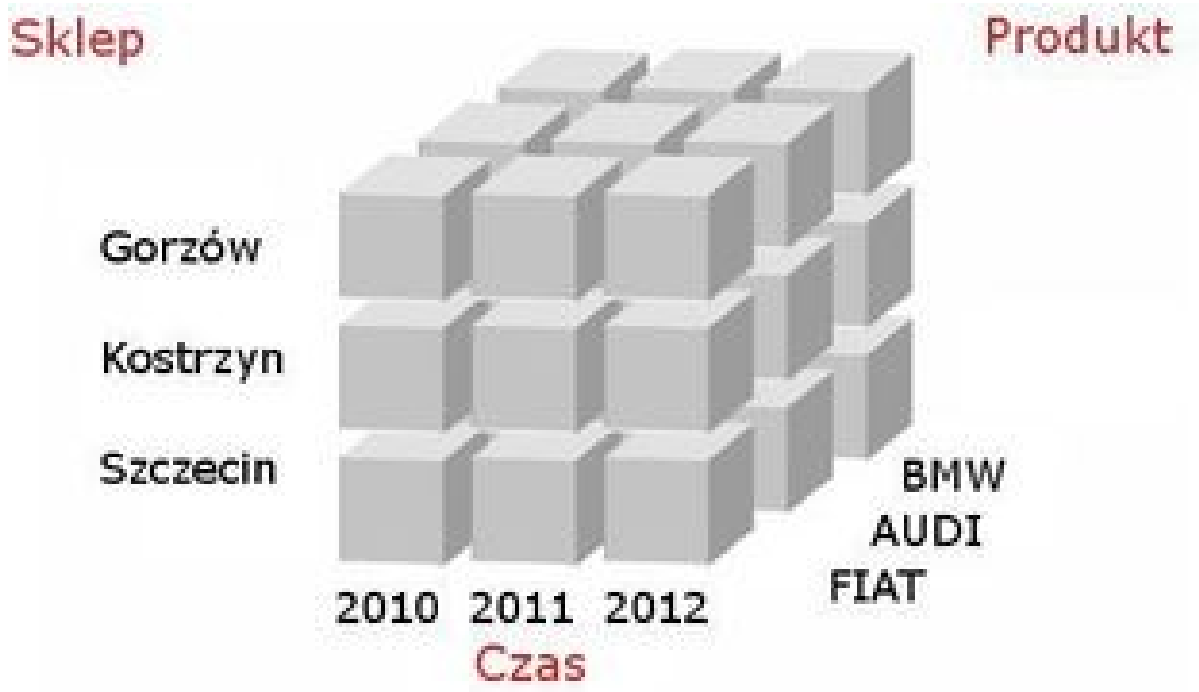
- *Multidimensional* OLAP
- implementacja w serwerach wielowymiarowych
- dane przechowywane w wielowymiarowych tabelach (ang. *data cubes*), zwanych potocznie **kostkami**

\+ 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**

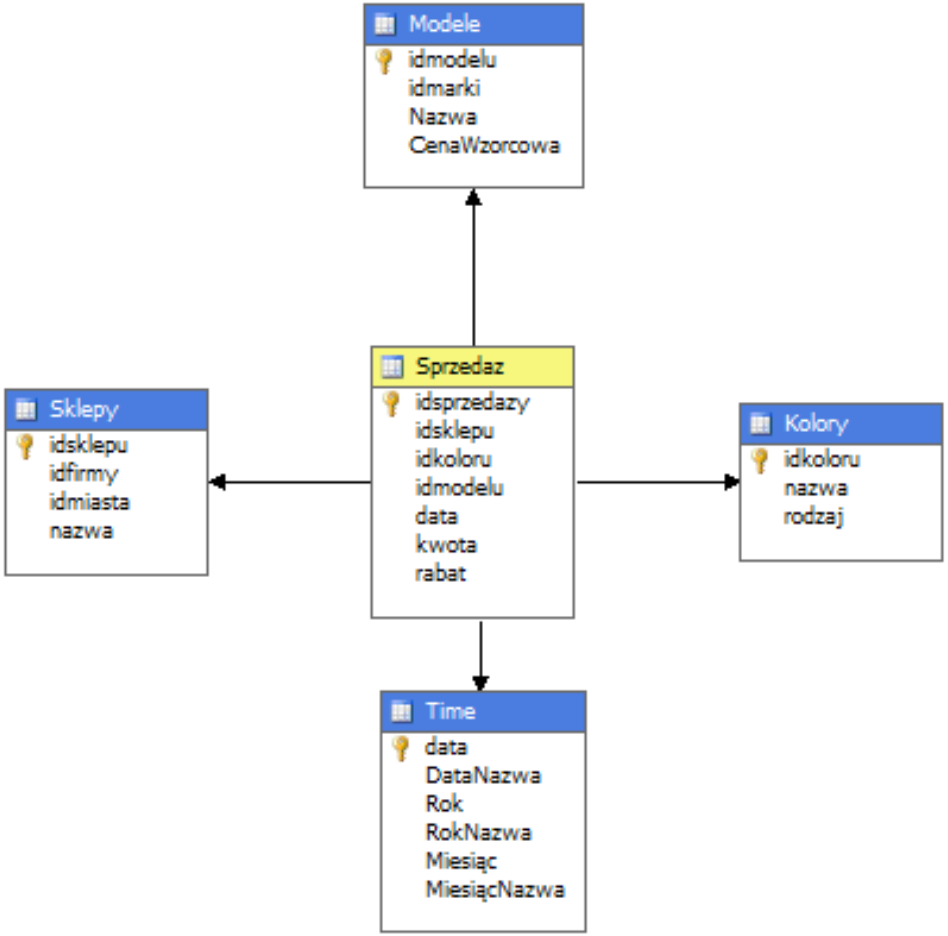
\+ nie potrzeba dodatkowej pamięci

– niska efektywność

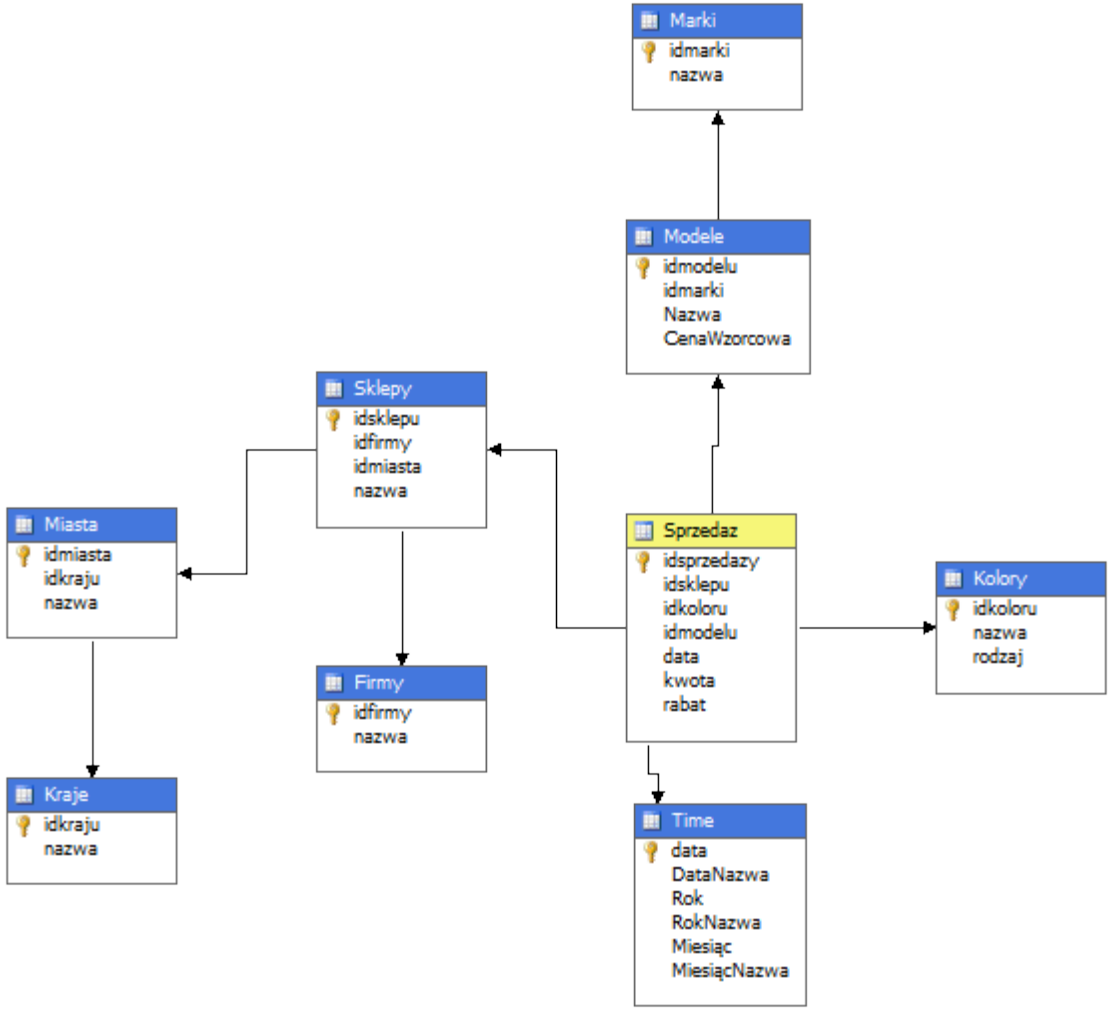
Stosowane schematy w ROLAP

- 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 zagregowane przechowywane w kostkach
- pośrednia struktura pomiędzy ROLAP a MOLAP („przyśpieszacz 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 > 1TB
- 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 REPLIKACJI/FRAGMENTACJI DANYCH

Replikacja danych

- Replikacja danych – proces powielania danych pomiędzy różnymi serwerami baz danych.
- Serwery lustrzane.

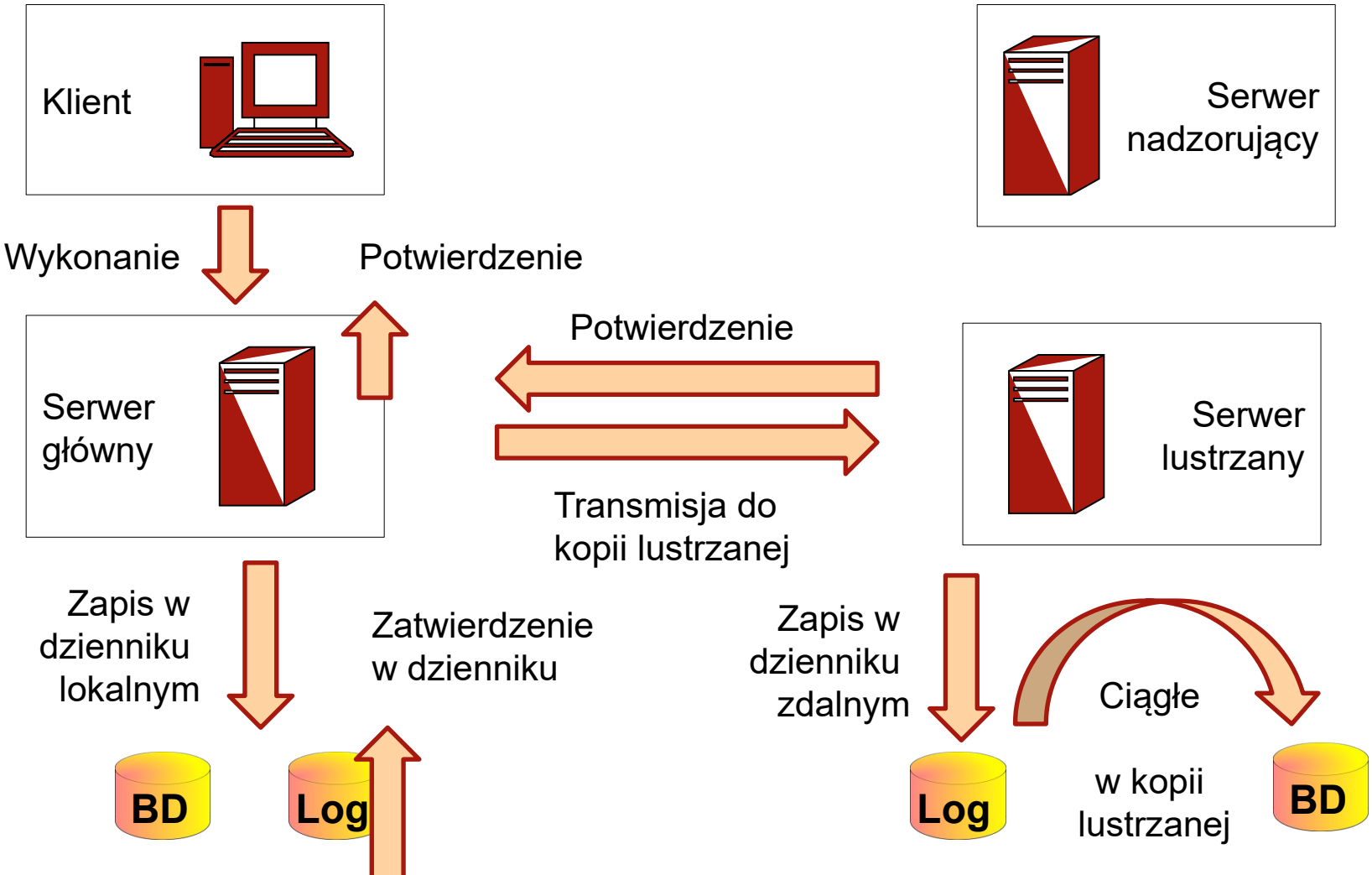
Replikacja danych - cel

- przyspieszenie dostępu do danych wykorzystywanych często
  - słownik miast, województw, kodów adresowych
  - katalog produktów
- równoważenie obciążenia węzłów
- równoległe wykonywanie zapytań
- zwiększenie efektywności dostępu do danych przez wybór repliki do której dostęp jest najszybszy
- bezpieczeństwo danych i całego systemu

Rodzaje replikacja danych

- Migawkowa – rozprowadzane dane mają stan z pewnego określonego momentu w czasie.
- Transakcyjna – dane rozprowadzane są na podstawie logów transakcji.
- Dwukierunkowa (łącząca) – serwer realizuje transakcje zarówno od innego serwera (lustrzanego), jak i od klientów.

Ogólna architektura technologii tworzenia lustrzanej kopii bazy 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)

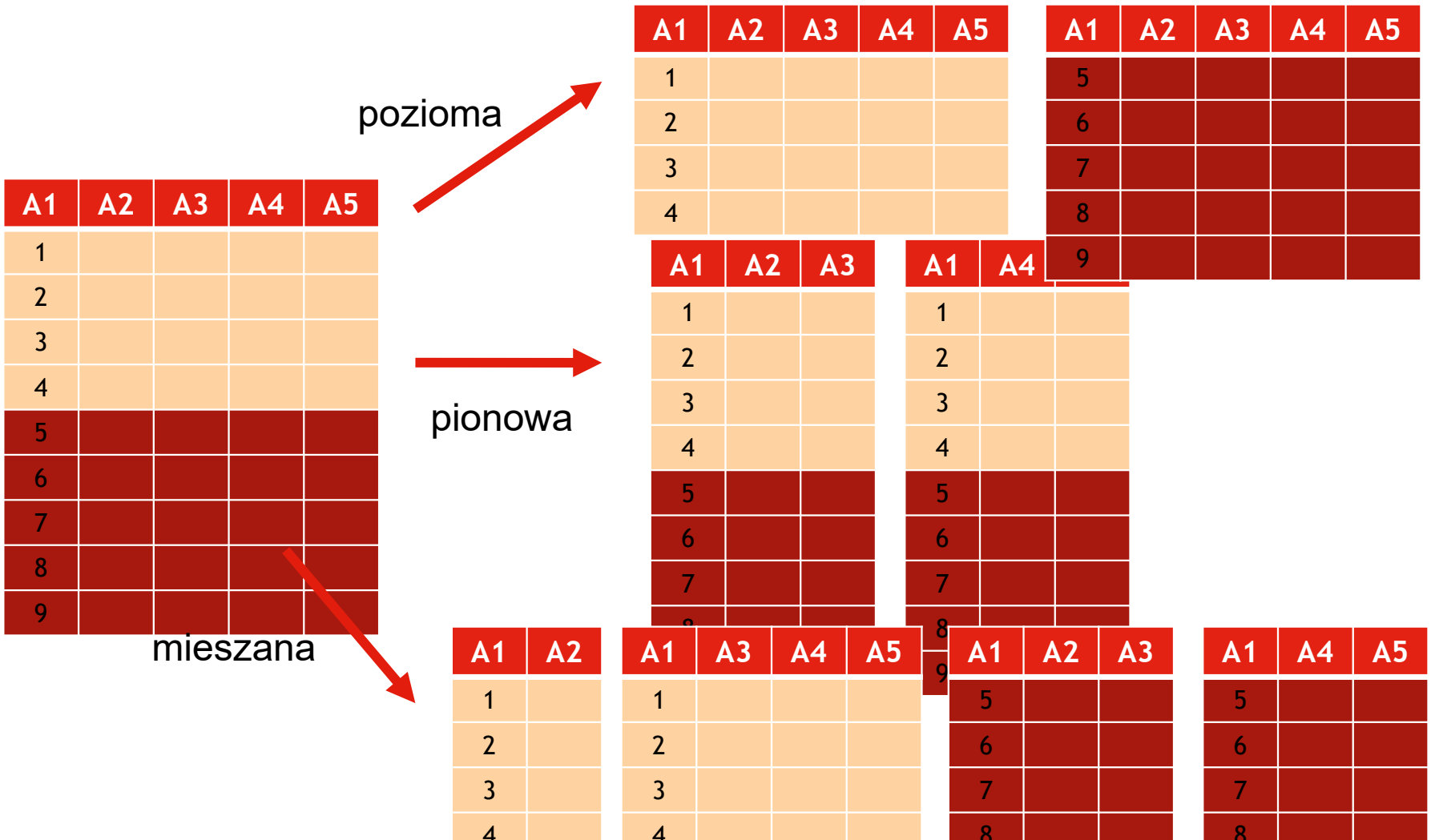
Fragmentacja pionowa

- 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 replikacji danych
- Cele i techniki fragmentacji danych

Pytania?

DZIĘKUJĘ ZA UWAGĘ