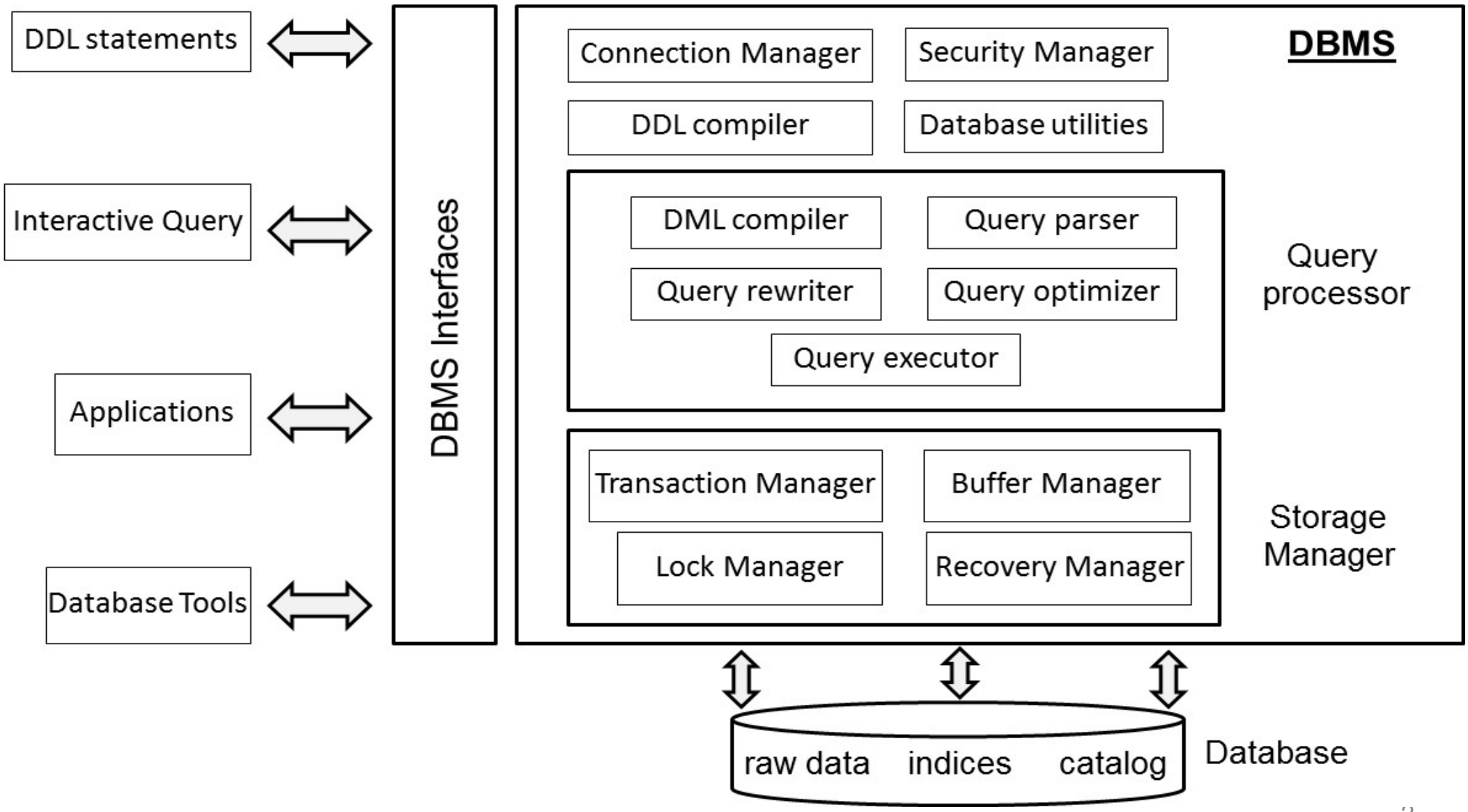
Wstęp

- Architektura DBMS
- Kategoryzacja DBMS

Architektura DBMS



Architektura DBMS

- Menadżer połączeń i bezpieczeństwa (Connection and Security Manager)
- Kompilator DDL (DDL Compiler)
- Procesor zapytań (Query Procesor)
- Menadżer przechowywania (Storage Manager)
- Narzędzia DBMS (DBMS Utilities)
- Interfejsy DBMS (DBMS Interfaces)

Menadżer połączeń i bezpieczeństwa

- Menadżer połączeń zapewnia możliwość konfiguracji połączenia z bazą danych (lokalnie lub przez sieć)
  - weryfikuje dane logowania i zwraca handler połączenia (connection handle)
  - połączenie z bazą danych może działać jako pojedynczy proces lub jako wątek w ramach procesu
- Menadżer bezpieczeństwa weryfikuje, czy użytkownik ma odpowiednie uprawnienia
  - dostęp do odczytu vs dostęp do zapisu

Kompilator DDL

- Kompiluje definicje danych określone w DDL
- 3 DDL'e (wewnętrzny/logiczny/zewnętrzny model)
- Kompilator DDL najpierw analizuje definicje DDL i sprawdza ich poprawność składniową
- Kompilator DDL następnie tłumaczy definicje danych na format wewnętrzny i w razie potrzeby generuje błędy
- Po udanej kompilacji kompilator DDL rejestruje definicje danych w katalogu

Procesor zapytań

- Procesor zapytań asystuje w wykonywaniu zapytań do bazy danych, takich jak pobieranie, wstawianie, aktualizacja lub usuwanie danych
- Kluczowe komponenty:
  - Kompilator DML (DML Compiler)
  - Parser zapytań (Query Parser)
  - Przepisywacz zapytań (Query Rewriter)
  - Optymalizator zapytań (Query Optimizer)
  - Wykonawca zapytań (Query Executor)

Kompilator DML

- Kompilator DML kompiluje instrukcje DML
- Proceduralny DML
  - DML w sposób jawny określa jak poruszać się po bazie danych
  - record-at-a-time DML (zorientowany na rekord)
  - brak procesora zapytań
- Deklaratywny DML
  - DML określa jakie dane należy pobrać lub jakie zmiany należy wprowadzić
  - set-at-a-time DML (zorientowany na zbiór)
  - procesor zapytań

Kompilator DML

```
import java.sql.*;
public class JDBCExample1 {
public static void main(String[] args) {
try {
System.out.println("Loading JDBC driver...");
Class.forName("com.mysql.jdbc.Driver");
System.out.println("JDBC driver loaded!");
} catch (ClassNotFoundException e) {
throw new RuntimeException(e);
}
String url =
"jdbc:mysql://localhost:3306/employeeschema";
String username = "root";
String password = "mypassword123";
String query = "select E.Name, D.DName" +
"from employee E, department D" +
"where E.DNR=D.DNR;";
Connection connection = null;
Statement stmt=null;
```

```
try {
System.out.println("Connecting to database");
connection = DriverManager.getConnection(url,
username, password);
System.out.println("MySQL Database connected!");
stmt = connection.createStatement();
ResultSet rs = stmt.executeQuery(query);
while (rs.next()) {
System.out.print(rs.getString(1));
System.out.print(" ");
System.out.println(rs.getString(2));
}
stmt.close();
} catch (SQLException e) {
System.out.println(e.toString());
} finally {
System.out.println("Closing the connection.");
if (connection != null) {
try {
connection.close();
} catch (SQLException ignore) {}}}}
```

Kompilator DML

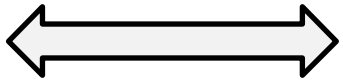
- Problem niedopasowania impedancji
  - mapowanie między pojęciami OO (np. Java) a relacyjnymi (np. SQL)
- Rozwiązania niedopasowania impedancji
  - język hosta i DBMS o porównywalnych strukturach danych (np. Java i OODBMS)
  - oprogramowanie pośredniczące do mapowania struktur danych z DBMS na język hosta i odwrotnie

Kompilator DML

Java

```
public class Employee {
private int EmployeeID;
private String Name;
private String Gender;
private int DNR;

public int getEmployeeID() {
return EmployeeID;
}
public void setEmployeeID( int id ) {
this.EmployeeID = id;
}
public String getName() {
return Name;
}
public void setName( String name ) {
this.Name = name;
}
…}
```



SQL

```
CREATE TABLE Employee (
'EmployeeID' INT NOT NULL,
'Name'  VARCHAR(45) NULL,
'Gender' VARCHAR(45) NULL,
'DNR' INT NULL)
```

EmployeeID	Name	Gender	DNR
100	Bart Baesens	Male	2
110	Wilfried Lemahieu	Male	4
120	Seppe vanden Broucke	Male	6
…			

Kompilator DML

- Kompilator DML rozpoczyna od wyodrębnienia instrukcji DML z języka hosta
- Kompilator DML współpracuje następnie z parserem zapytań, przepisywaczem zapytań, optymalizatorem zapytań i wykonawcą zapytań w celu wykonania instrukcji DML
- Błędy są generowane i zgłaszane w razie potrzeby

Parser zapytań i przepisywanie zapytań

- Parser zapytań analizuje zapytanie do wewnętrznego formatu reprezentacji
- Parser zapytań sprawdza zapytanie pod kątem poprawności składniowej i semantycznej
- Przepisywanie zapytań optymalizuje zapytanie niezależnie od aktualnego stanu bazy danych

Optymalizator zapytań

- Optymalizator zapytań optymalizuje zapytanie na podstawie aktualnego stanu bazy danych (na podstawie np. predefiniowanych indeksów)
- Optymalizator zapytań opracowuje różne plany wykonania zapytań i ocenia ich koszt pod kątem szacunkowych
  - liczby operacji we/wy
  - koszt przetwarzania procesora
  - czasu wykonywania
- Szacunki oparte na informacjach katalogowych w połączeniu z wnioskami statystycznymi
- Optymalizator zapytań jest kluczowym atutem DBMS

Wykonawca zapytań

- Wynikiem optymalizacji zapytania jest ostateczny plan wykonania
- Wykonawca zapytania zajmuje się faktycznym wykonaniem, wywołując menedżera pamięci w celu pobrania żądanych danych

Menadżer przechowywania

- Menadżer przechowywania zarządza fizycznym dostępem do plików i nadzoruje prawidłowe i wydajne przechowywanie danych
- Składa się z
  - menadżera transakcji
  - menadżera buforów
  - menadżera blokad
  - menadżera odzyskiwania

Menadżer transakcji

- Menadżer transakcji nadzoruje realizację transakcji bazodanowych
  - transakcja bazodanowa to sekwencja operacji odczytu/zapisu uważana za jednostkę atomową
- Menadżer transakcji tworzy harmonogram z przeplatanymi operacjami odczytu/zapisu
- Menadżer transakcji gwarantuje właściwość ACID
- COMMIT transakcji po pomyślnym wykonaniu i ROLLBACK po nieudanej realizacji

Menadżer buforów

- Menadżer buforów zarządza pamięcią buforową DBMS
  - inteligentnie buforuje dane w buforze
- Przykładowe strategie:
  - Lokalizacja danych: ostatnio pobrane dane prawdopodobnie zostaną ponownie pobrane
  - Prawo 20/80: 80% transakcji odczytuje lub zapisuje tylko 20% danych
- Menadżer buforów musi przyjąć strategię inteligentnej wymiany w przypadku zapełnienia bufora
- Menadżer bufora musi współpracować z menedżerem blokad

Menadżer blokad

- Zapewnia kontrolę współbieżności, co zapewnia integralność danych przez cały czas
- Dwa rodzaje blokad: blokady odczytu i zapisu
- Menadżer blokad odpowiedzialny za przypisywanie, zwalnianie i rejestrowanie blokad w katalogu
- Menadżer blokad korzysta z protokołu blokowania, który opisuje zasady blokowania, oraz tabeli blokad z informacjami o blokadzie

Menadżer odzyskiwania

- Nadzoruje poprawność wykonywania transakcji bazodanowych
- Śledzi wszystkie operacje bazy danych w pliku dziennika
- Zostanie wezwany do cofnięcia czynności przerwanych transakcji lub podczas odzyskiwania po awarii

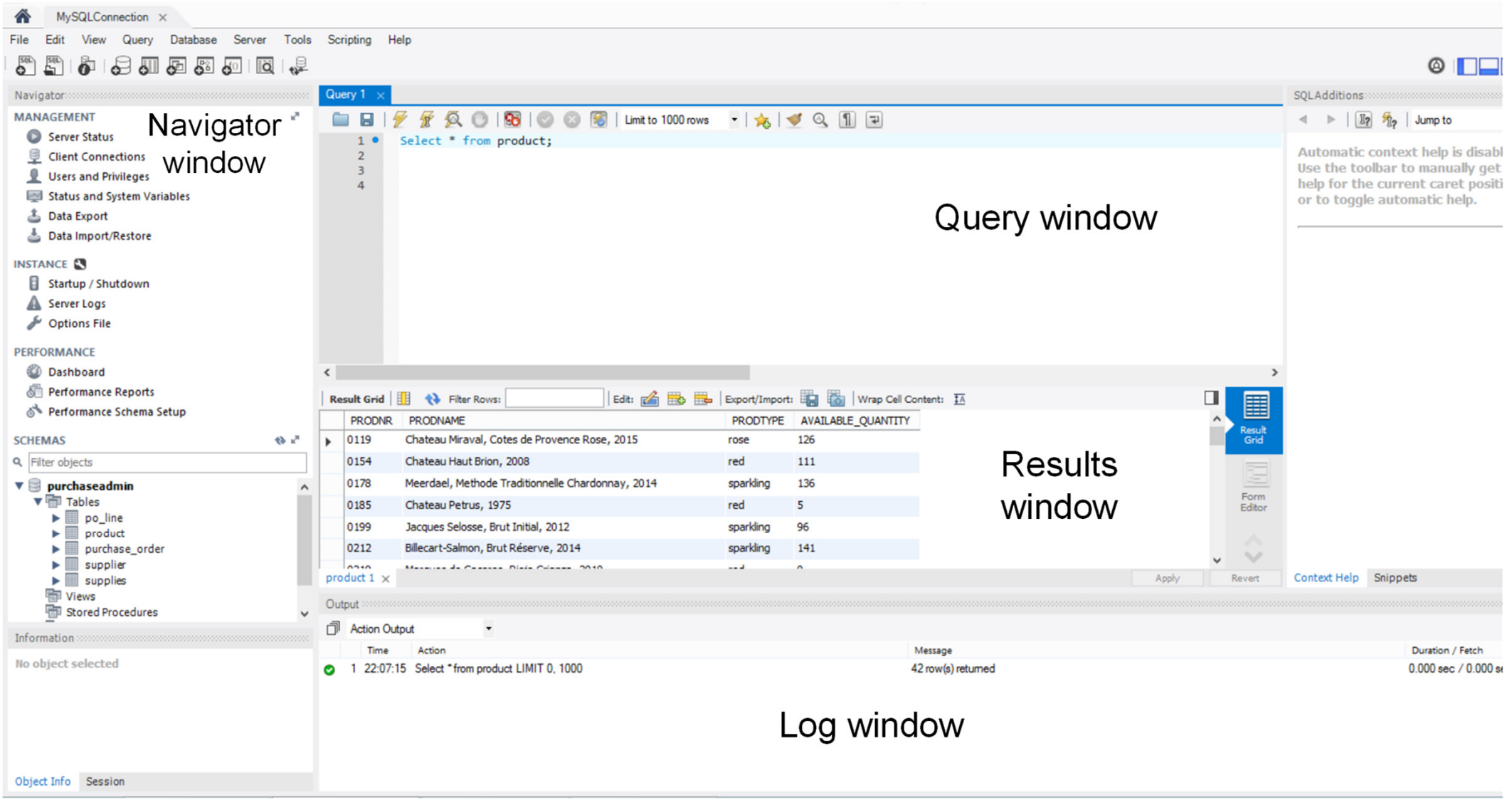
Narzędzia DBMS

- Narzędzia do ładowania
- Narzędzia do reorganizacji
- Narzędzia do monitorowania wydajności
- Narzędzia do zarządzania użytkownikami
- Narzędzia do tworzenia kopii zapasowych i odzyskiwania

Interfejsy DBMS

- Interfejs webowy
- Interfejs linii komend
- Interfejs oparty na formularzach
- Graficzny interfejs użytkownika
- Interfejs w języku naturalnym
- Interfejs administratora
- Interfejs sieciowy
- …

Interfejsy DBMS



Kategoryzacja DBMS

- Kategoryzacja na podstawie modelu danych
- Kategoryzacja na podstawie stopnia równoczesnego dostępu
- Kategoryzacja na podstawie architektury
- Kategoryzacja na podstawie użycia

Kategoryzacja na podstawie modelu danych

- Hierarchiczne DBMS
  - model danych podobny do drzewa
  - DML jest proceduralny i zorientowany na rekordy
  - brak procesora zapytań (model logiczny i wewnętrzny przeplatają się)
  - np. IMS (IBM)
- Sieciowe DBMS
  - wykorzystują sieciowy model danych
  - CODASYL DBMS
  - DML jest proceduralny i zorientowany na rekordy
  - brak procesora zapytań (model logiczny i wewnętrzny przeplatają się)
  - CA-IDMS (Computer Associates)

Kategoryzacja na podstawie modelu danych

- Relacyjne DBMS
  - wykorzystuje relacyjny model danych
  - obecnie najpopularniejszy
  - SQL (deklaratywny i zorientowany na zbiór)
  - procesor zapytań
  - ścisły rozdział między logicznym i wewnętrznym modelem danych
  - np. MySQL (open source, Oracle), Oracle DBMS (Oracle), DB2 (IBM), Microsoft SQL (Microsoft)

Kategoryzacja na podstawie modelu danych

- Zorientowany obiektowo DBMSs (OODBMS)
  - oparty na obiektowym modelu danych
  - brak niedopasowania impedancji w połączeniu z obiektowym językiem hosta
  - np. db4o (open source, Versant), Caché (Intersystems) GemStone/S (GemTalk Systems)
  - sukcesy tylko na rynkach niszowych, ze względu na ich złożoność

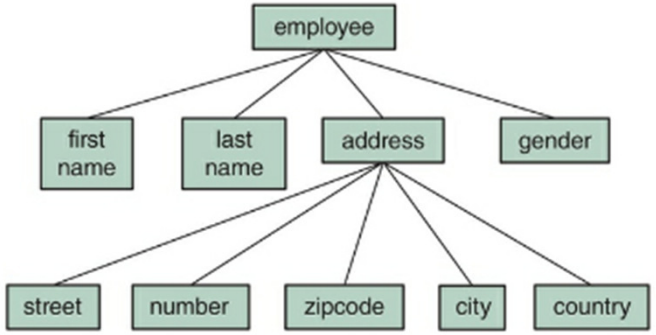
Kategoryzacja na podstawie modelu danych

- Obiektowo-relacyjny DBMSs (ORDBMSs)
  - określane również jako rozszerzone relacyjne DBMS (ERDBMSs)
  - używa modelu relacyjnego rozszerzonego o koncepcje obiektowe
  - DML jest SQL (deklaratywny i zorientowany na zbiór)
  - np. Oracle DBMS (Oracle), DB2 (IBM), Microsoft SQL (Microsoft)

Kategoryzacja na podstawie modelu danych

- XML DBMS
  - używa modelu XML do przechowywania danych
  - Natywne XML DBMS (np. BaseX, eXist) mapują strukturę drzewa dokumentu XML na fizyczną strukturę pamięci
  - DBMS z obsługą XML (np. Oracle, IBM DB2) to istniejące DBMS, które zostały rozszerzone o funkcje przechowywania danych XML

```
<employee>
    <firstname>Bart</firstname>
    <lastname>Baesens</lastname>
    <address>
      <street>Naamsestraat</street>
      <number>69</number>
      <zipcode>3000</zipcode>
      <city>Leuven</city>
      <country>Belgium</country>
    </address>
    <gender>Male</gender>
</employee>
```

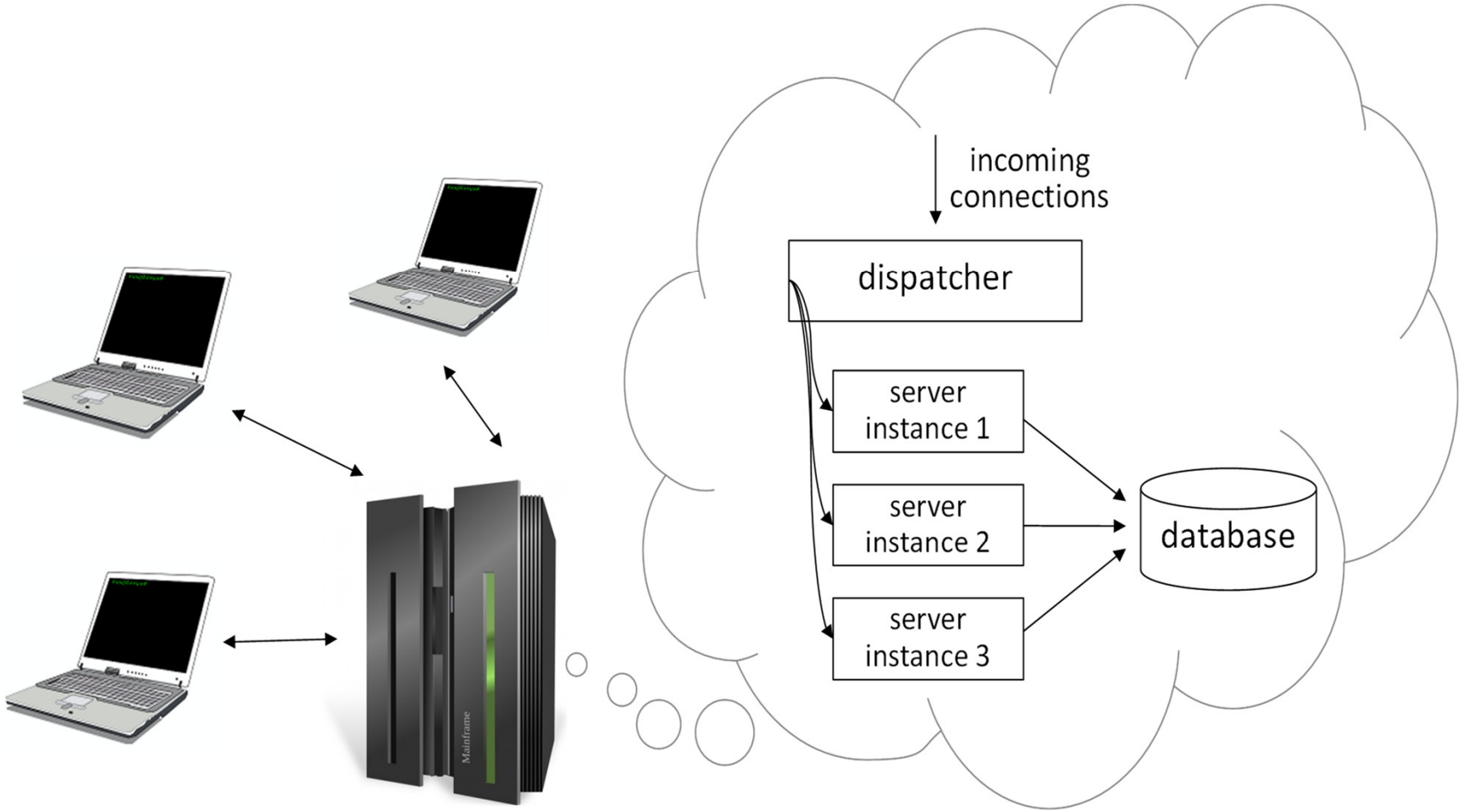


Kategoryzacja na podstawie modelu danych

- NoSQL DBMS
  - ukierunkowane na przechowywanie dużych i nieustrukturyzowanych danych
  - można podzielić na magazyny klucz-wartość, bazy danych zorientowane kolumnowo i grafowe bazy danych
  - skupiają się na skalowalności i zdolności radzenia sobie z nieregularnymi lub wysoce niestabilnymi strukturami danych
  - np. Apache Hadoop, MongoDB, Neo4j

Kategoryzacja na podstawie stopnia równoczesnego dostępu

- Systemy jedno- i wieloużytkownikowe



Kategoryzacja na podstawie architektury

- Scentralizowana architektura DBMS
  - dane są utrzymywane na centralnym serwerze
- Architektura klient-serwer
  - aktywni klienci żądają usług od serwerów pasywnych
  - gruby serwer vs gruby klient
- n-warstwowa architektura DBMS
  - klient z funkcjonalnością GUI, serwer aplikacji z aplikacjami, serwer bazy danych z DBMS i bazą danych oraz serwer WWW do dostępu przez WWW

Kategoryzacja na podstawie architektury

- Chmurowa architektura DBMS
  - DBMS i baza danych są hostowane przez zewnętrznego dostawcę chmury
  - np. projekt Apache Cassandra i Google's BigTable
- Sfederowany DBMS
  - zapewnia jednolity interfejs do wielu źródeł danych
  - ukrywa szczegóły przechowywania danych aby ułatwić dostęp do danych

Kategoryzacja na podstawie architektury

- in-memory DBMS (w pamięci)
  - przechowuje wszystkie dane w pamięci wewnętrznej zamiast wolniejszej pamięci zewnętrznej (np. dysk)
  - często wykorzystywane do zastosowań czasu rzeczywistego
  - np. HANA (SAP), SQLite (częściowo)

Kategoryzacja na podstawie użycia

- Przetwarzanie transakcyjne w trybie on-line (On-line transaction processing - OLTP)
  - skupia się na zarządzaniu danymi operacyjnymi lub transakcyjnymi
  - serwer bazy danych musi być w stanie przetwarzać wiele prostych transakcji w jednostce czasu
  - DBMS musi mieć dobre wsparcie dla przetwarzania dużej liczby krótkich, prostych zapytań
- Przetwarzanie analityczne w trybie on-line (On-line analytical processing - OLAP)
  - skupia się na wykorzystaniu danych operacyjnych do podejmowania decyzji taktycznych lub strategicznych
  - ograniczona liczba użytkowników formułuje złożone zapytania
  - DBMS powinien wspierać wydajne przetwarzanie złożonych zapytań, które często przychodzą w mniejszych ilościach

Kategoryzacja na podstawie użycia

- Big Data & Analytics
  - bazy danych NoSQL
  - skupia się na bardziej elastycznych, a nawet pozbawionych schematów strukturach baz danych
  - przechowuje nieustrukturyzowane informacje, takie jak e-maile, dokumenty tekstowe, tweety na Twitterze, posty na Facebooku itp.
- Multimedia
  - multimedialne DBMS zapewniają przechowywanie danych multimedialnych, takich jak tekst, obrazy, audio, wideo, gry 3D itp.
  - powinny również zapewniać narzędzia do zapytań opartych na treści

Kategoryzacja na podstawie użycia

- Aplikacje przestrzenne
  - przestrzenne DBMS obsługują przechowywanie i odpytywanie danych przestrzennych (zarówno 2D, jak i 3D)
  - Geographical Information Systems (GIS)
- Czujniki
  - dane z czujników, np. do śledzenia aktywności, dane telematyczne

Kategoryzacja na podstawie użycia

- Mobilne
  - mobilne DBMS działają na smartfonach, tabletach lub innych urządzeniach mobilnych
  - powinny zawsze być online, mieć niewielką powierzchnię i być w stanie poradzić sobie z ograniczoną mocą przetwarzania, pamięcią masową i żywotnością baterii
- Open source
  - kod DBMS typu open source jest publicznie dostępny i może być rozszerzany przez każdego
  - www.sourceforge.net
  - np. MySQL (Oracle)