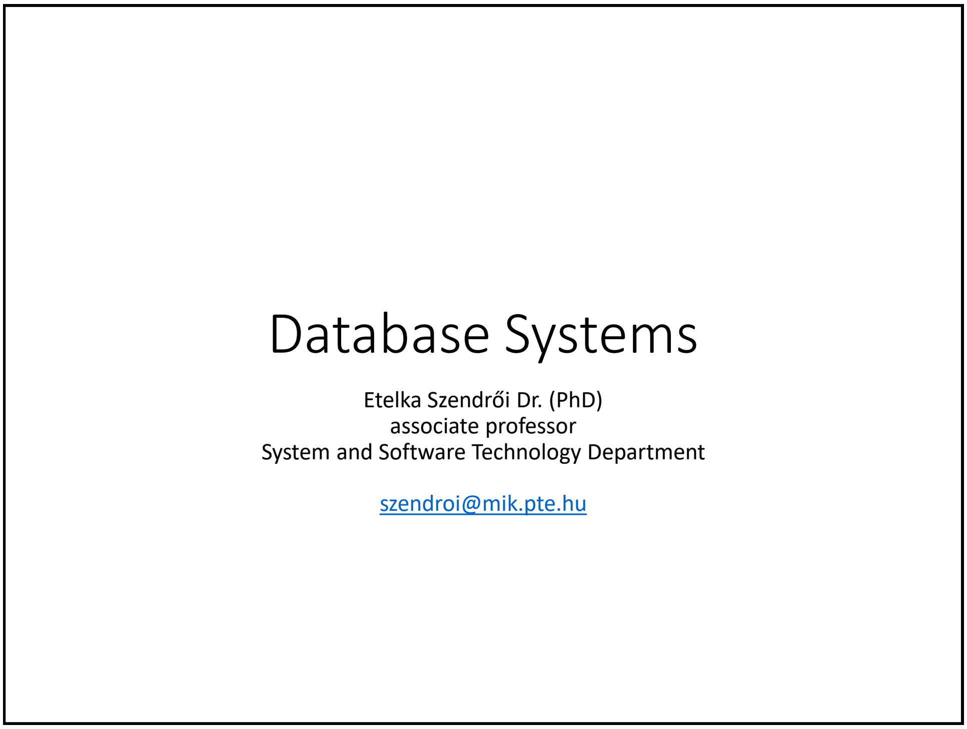
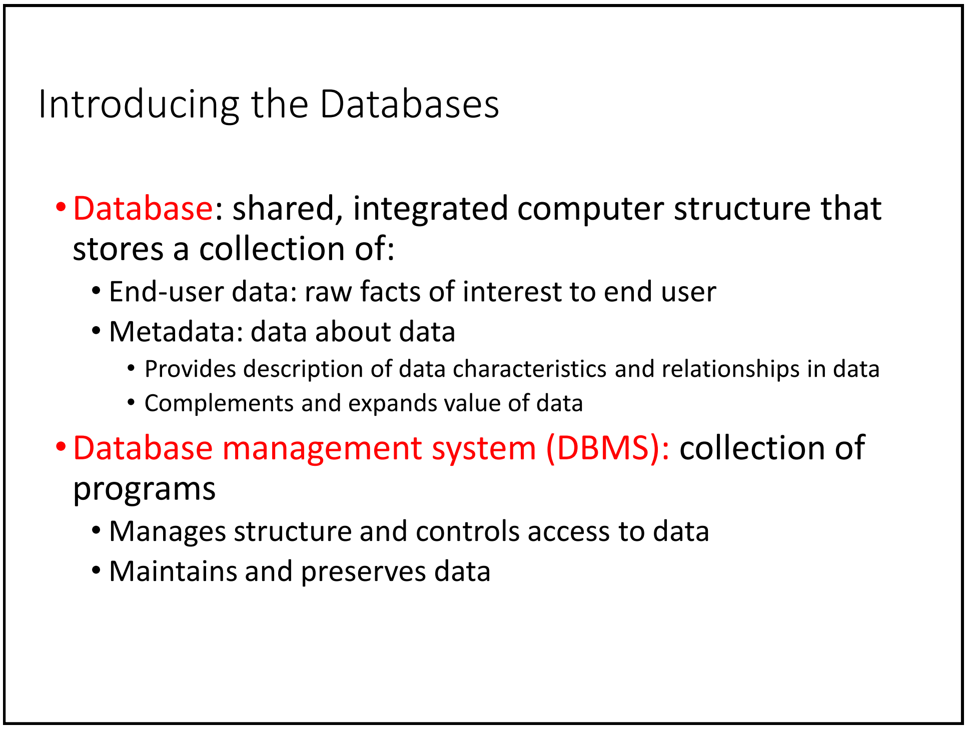


Database Systems

Etelka Szendrői Dr. (PhD)
associate professor
System and Software Technology Department

szendroi@mik.pte.hu



Introducing the Databases

- Database: shared, integrated computer structure that stores a collection of:
  - End-user data: raw facts of interest to end user
  - Metadata: data about data
    - Provides description of data characteristics and relationships in data
    - Complements and expands value of data
- Database management system (DBMS): collection of programs
  - Manages structure and controls access to data
  - Maintains and preserves data

Database Management System (DBMS)

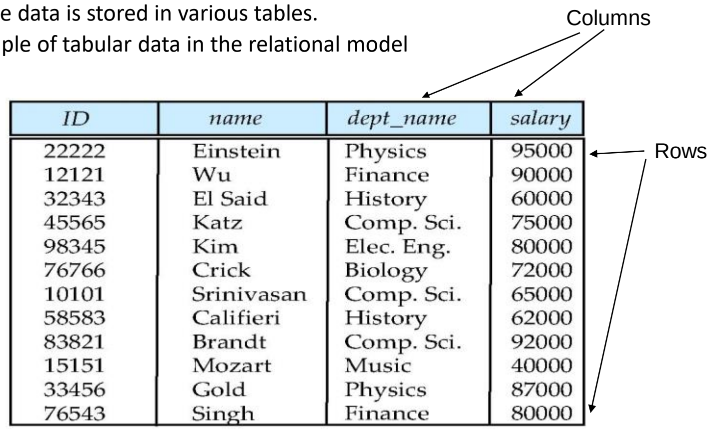
- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use
- Database Applications:
  - Banking: transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Online retailers: order tracking, customized recommendations
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
- Databases can be very large.
- Databases touch all aspects of our lives

Data Models

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semistructured data model (XML)
- Other older models:
  - Network model
  - Hierarchical model

Relational Model

- All the data is stored in various tables.
- Example of tabular data in the relational model



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Columns
Rows

(a) The *instructor* table

Data Definition Language (DDL)

- Specification notation for defining the database schema

  Example:
```
create table instructor (
                 ID          char(5),
                 name        varchar(20),
                 dept_name   varchar(20),
                 salary      numeric(8,2))
```

- DDL compiler generates a set of table templates stored in a ***data dictionary***
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - **Pure** – used for proving properties about computational power and for optimization
    - Relational Algebra
    - Tuple relational calculus
    - Domain relational calculus
  - **Commercial** – used in commercial systems
    - SQL is the most widely used commercial language

SQL

- The most widely used commercial language
- To be able to compute complex functions SQL is usually embedded in some higher-level language
- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Database Design

The process of designing the general structure of the database:

- Logical Design – Deciding on the database schema. Database design requires that we find a "good" collection of relation schemas.
  - Business decision – What attributes should we record in the database?
  - Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?
- Physical Design – Deciding on the physical layout of the database

Design Approaches

- Need to come up with a methodology to ensure that each of the relations in the database is "good"
- Two ways of doing so:
  - Entity Relationship Model
    - Models an enterprise as a collection of *entities* and *relationships*
    - Represented diagrammatically by an *entity-relationship diagram:*
  - Normalization Theory
    - Formalize what designs are bad, and test for them

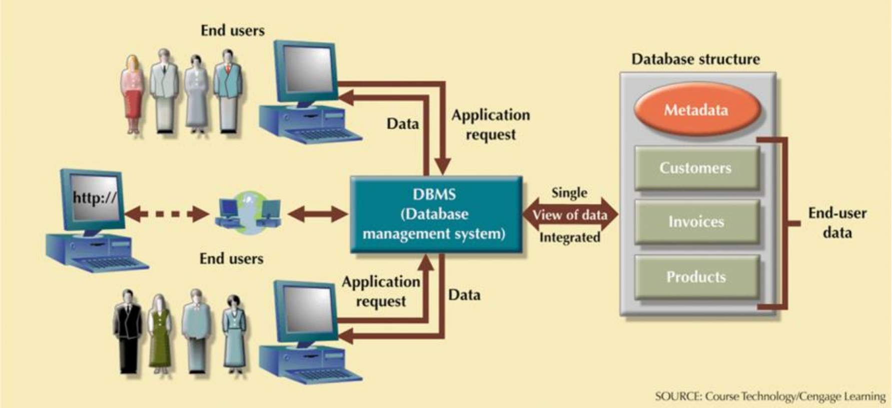
Object-Relational Data Models

- Relational model: flat, “atomic” values
- Object Relational Data Models
  - Extend the relational data model by including object orientation and constructs to deal with added data types.
  - Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
  - Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
  - Provide upward compatibility with existing relational languages.

XML: Extensible Markup Language

- Defined by the WWW Consortium (W3C)
- Originally intended as a document markup language not a database language
- The ability to specify new tags, and to create nested tag structures made XML a great way to exchange **data**, not just documents
- XML has become the basis for all new generation data interchange formats.
- A wide variety of tools is available for parsing, browsing and querying XML documents/data

The DBMS manages the interaction between the end user and the database

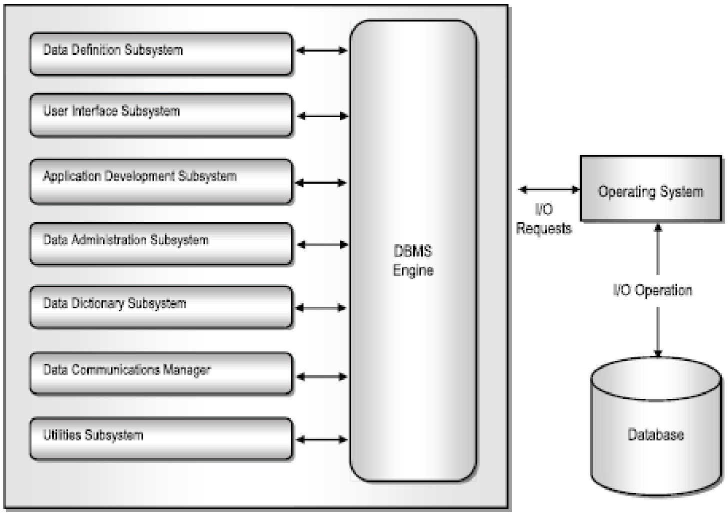


SOURCE: Course Technology/Cengage Learning

Role and Advantages of the DBMS (cont'd.)

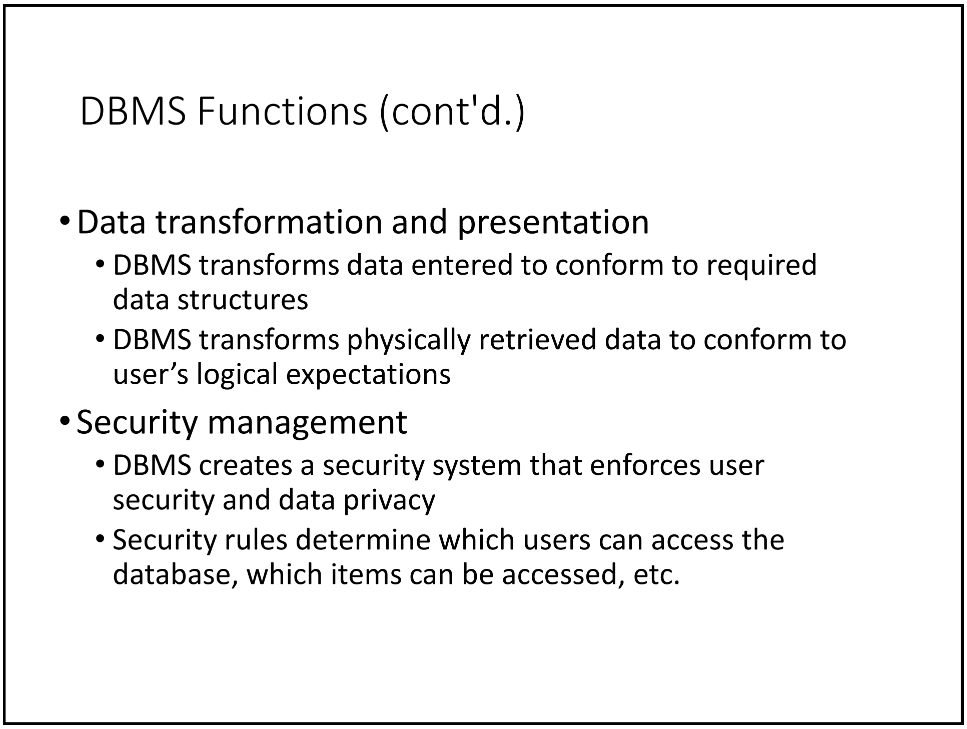
- Advantages of a DBMS:
  - Improved data sharing
  - Improved data security
  - Better data integration
  - Minimized data inconsistency
  - Improved data access
  - Improved decision making
  - Increased end-user productivity

Functional components of DBMS


Data Definition Subsystem
User Interface Subsystem
Application Development Subsystem
Data Administration Subsystem
Data Dictionary Subsystem
Data Communications Manager
Utilities Subsystem
DBMS Engine
I/O Requests
Operating System
I/O Operation
Database

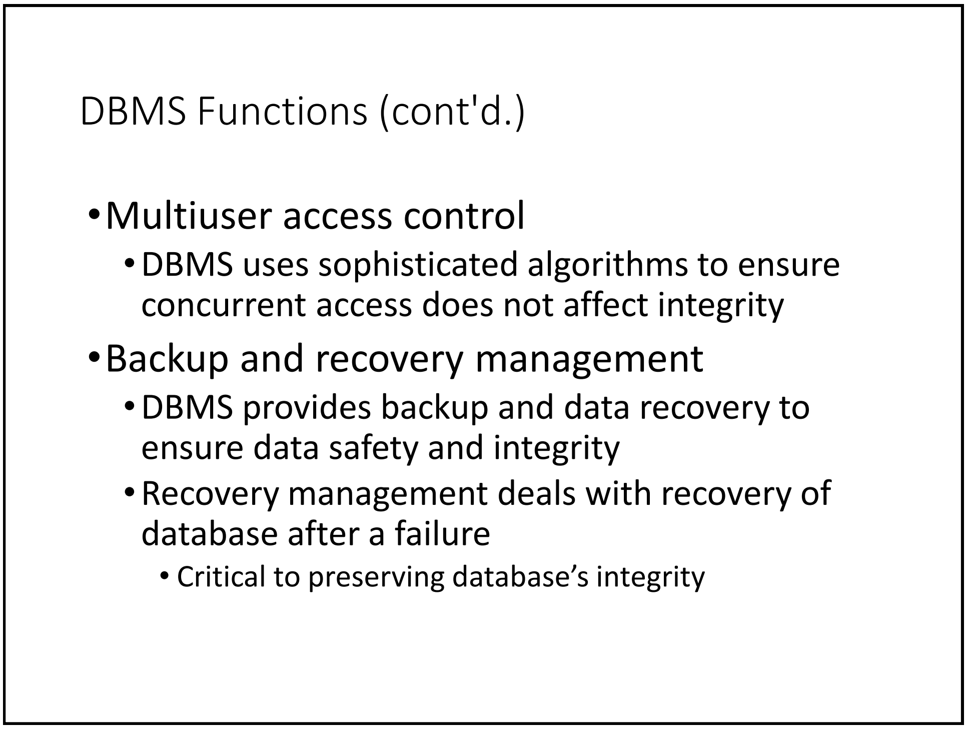
DBMS Functions (cont'd.)

- Data storage management
  - DBMS creates and manages complex structures required for data storage
  - Also stores related data entry forms, screen definitions, report definitions, etc.
  - Performance tuning: activities that make the database perform more efficiently
  - DBMS stores the database in multiple physical data files



DBMS Functions (cont'd.)

- Data transformation and presentation
  - DBMS transforms data entered to conform to required data structures
  - DBMS transforms physically retrieved data to conform to user’s logical expectations
- Security management
  - DBMS creates a security system that enforces user security and data privacy
  - Security rules determine which users can access the database, which items can be accessed, etc.



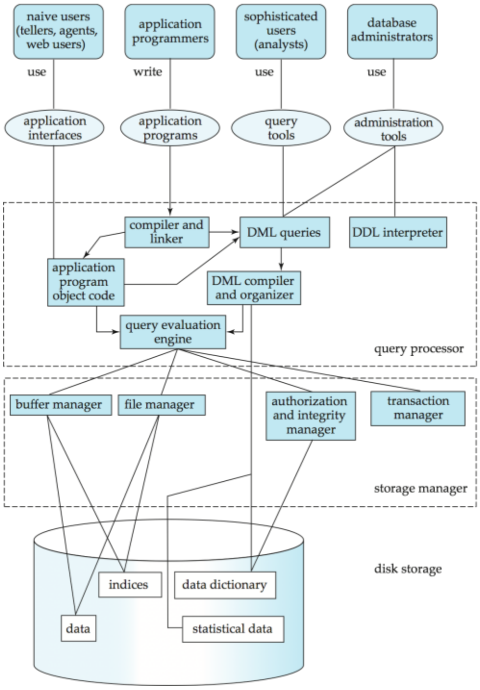
DBMS Functions (cont'd.)

- Multiuser access control
  - DBMS uses sophisticated algorithms to ensure concurrent access does not affect integrity
- Backup and recovery management
  - DBMS provides backup and data recovery to ensure data safety and integrity
  - Recovery management deals with recovery of database after a failure
    - Critical to preserving database’s integrity

DBMS Functions (cont'd.)

- Data integrity management
  - DBMS promotes and enforces integrity rules
    - Minimizes redundancy
    - Maximizes consistency
  - Data relationships stored in data dictionary used to enforce data integrity
  - Integrity is especially important in transaction-oriented database systems

Database System Internals



Database Engine

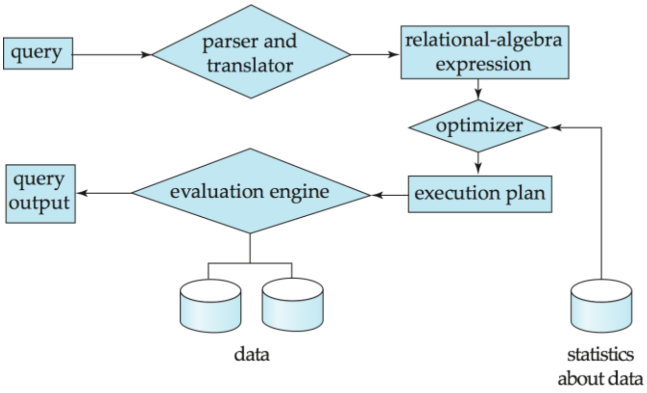
- Storage manager
- Query processing
- Transaction manager

Storage Management

- **Storage manager** is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible to the following tasks:
  - Interaction with the OS file manager
  - Efficient storing, retrieving and updating of data
- Issues:
  - Storage access
  - File organization
  - Indexing and hashing

Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



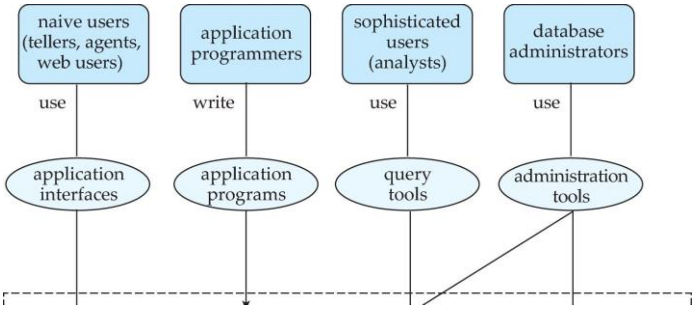
Query Processing (Cont.)

- Alternative ways of evaluating a given query
  - Equivalent expressions
  - Different algorithms for each operation
- Cost difference between a good and a bad way of evaluating a query can be enormous
- Need to estimate the cost of operations
  - Depends critically on statistical information about relations which the database must maintain
  - Need to estimate statistics for intermediate results to compute cost of complex expressions

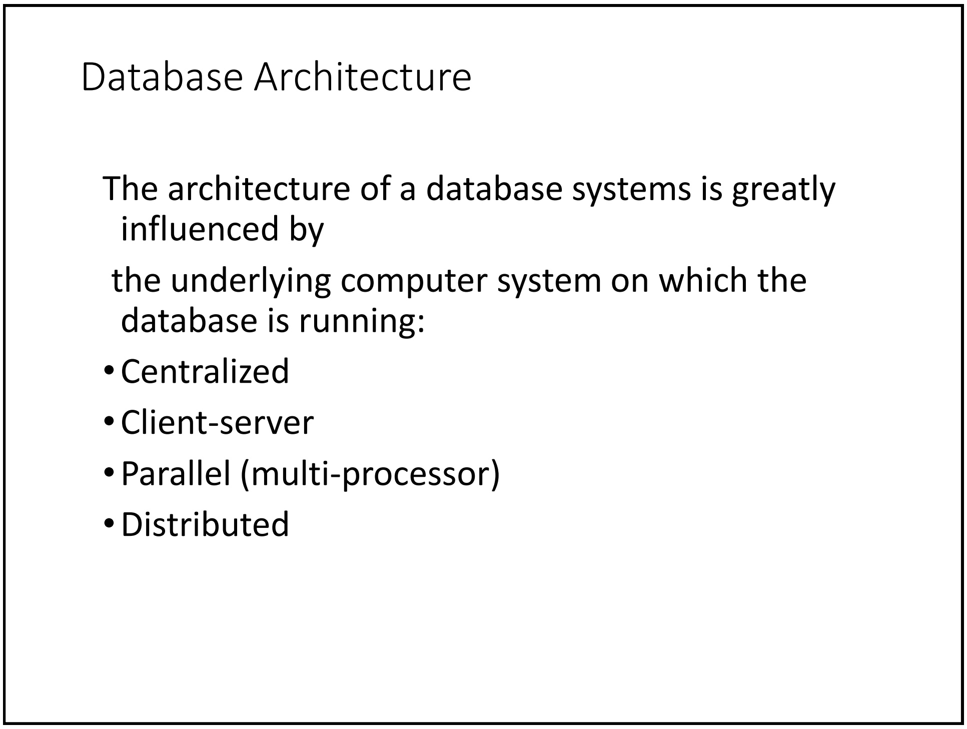
Transaction Management

- What if the system fails?
- What if more than one user is concurrently updating the same data?
- A **transaction** is a collection of operations that performs a single logical function in a database application
- **Transaction-management component** ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.
- **Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Database Users and Administrators



Database

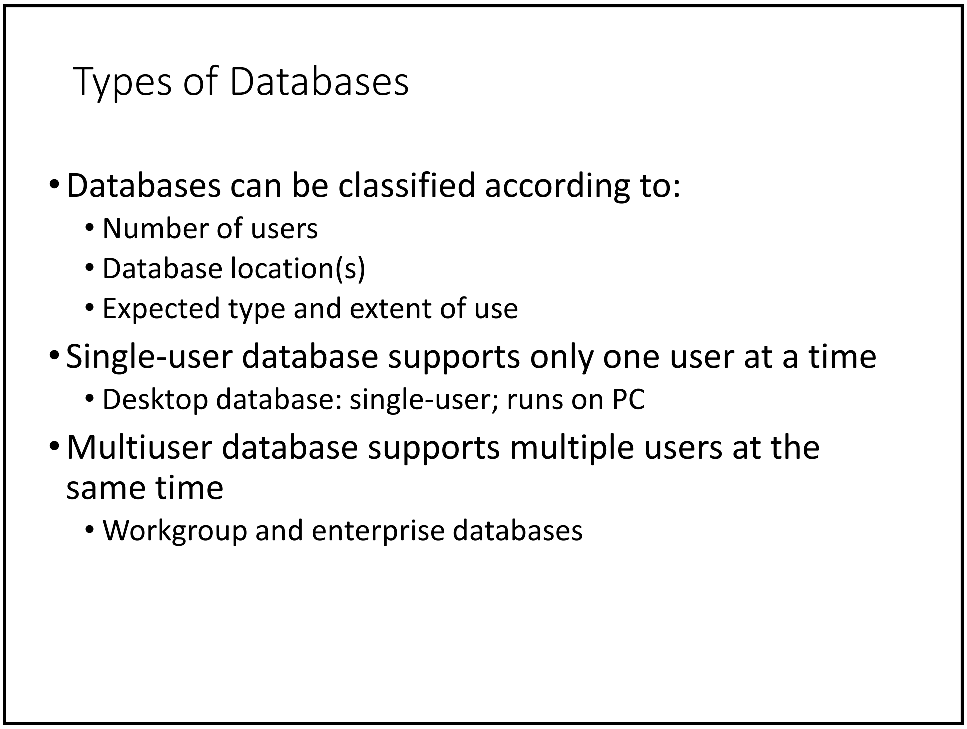


Database Architecture

The architecture of a database systems is greatly influenced by

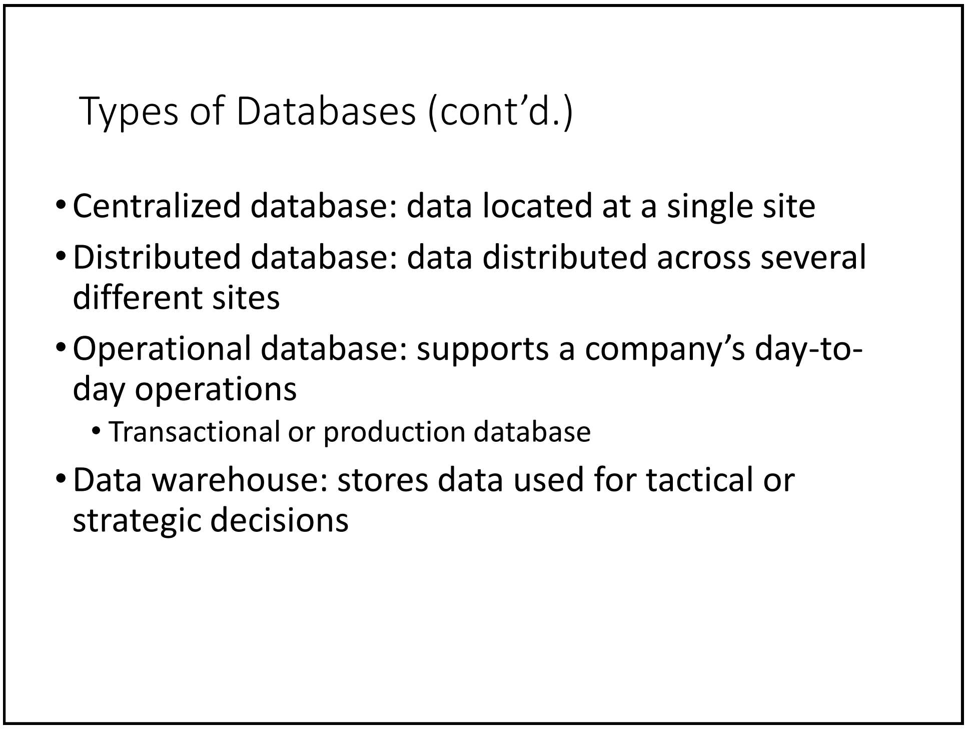
the underlying computer system on which the database is running:

- Centralized
- Client-server
- Parallel (multi-processor)
- Distributed



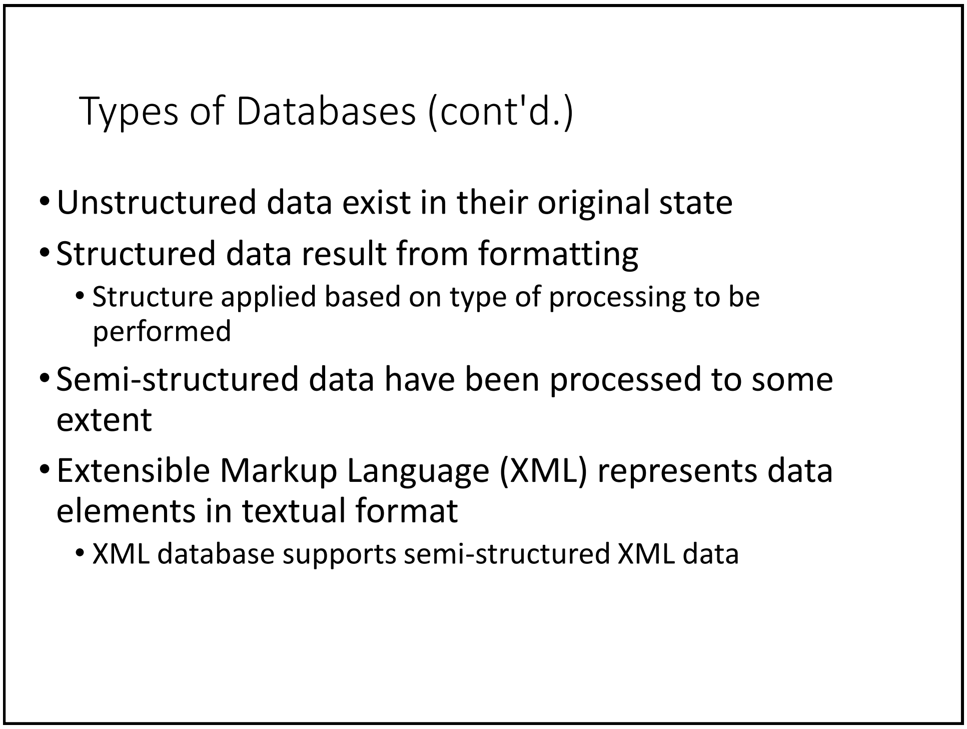
Types of Databases

- Databases can be classified according to:
  - Number of users
  - Database location(s)
  - Expected type and extent of use
- Single-user database supports only one user at a time
  - Desktop database: single-user; runs on PC
- Multiuser database supports multiple users at the same time
  - Workgroup and enterprise databases



Types of Databases (cont’d.)

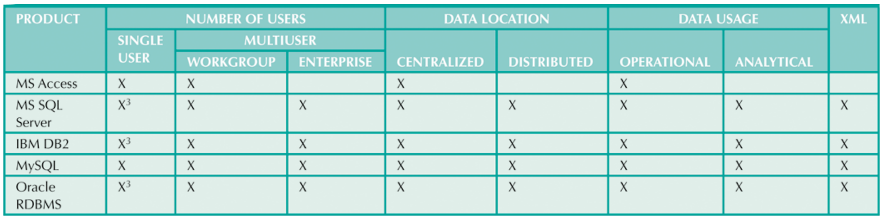
- Centralized database: data located at a single site
- Distributed database: data distributed across several different sites
- Operational database: supports a company’s day-to-day operations
  - Transactional or production database
- Data warehouse: stores data used for tactical or strategic decisions



Types of Databases (cont'd.)

- Unstructured data exist in their original state
- Structured data result from formatting
  - Structure applied based on type of processing to be performed
- Semi-structured data have been processed to some extent
- Extensible Markup Language (XML) represents data elements in textual format
  - XML database supports semi-structured XML data

Types of Databases



PRODUCT	NUMBER OF USERS			DATA LOCATION		DATA USAGE		XML
	SINGLE USER	MULTIUSER						
		WORKGROUP	ENTERPRISE	CENTRALIZED	DISTRIBUTED	OPERATIONAL	ANALYTICAL	
MS Access	X	X		X		X		
MS SQL Server	X[3]	X	X	X	X	X	X	X
IBM DB2	X[3]	X	X	X	X	X	X	X
MySQL	X	X	X	X	X	X	X	X
Oracle RDBMS	X[3]	X	X	X	X	X	X	X

[3] Vendor offers single-user/personal DBMS version

The Database System

- Database system: the collection of Database Managements System, databases and user applications which defines and regulates the collection, storage, management, use of data
- Five major parts of a database system:
  - Hardware
  - Software
  - People
  - Procedures
  - Data

Database System (DBS)

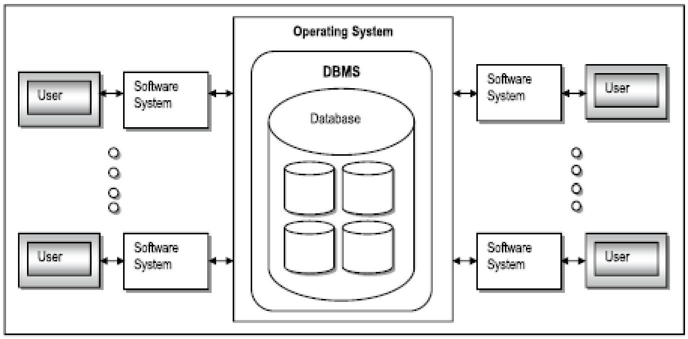
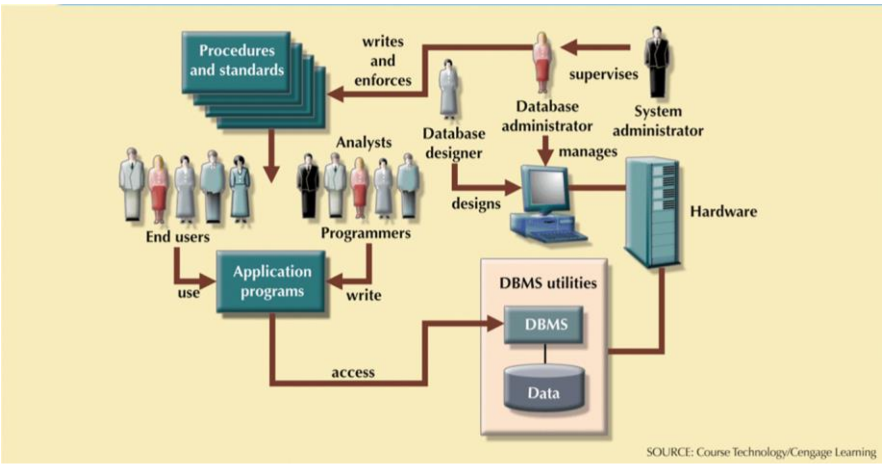


Figure 1-1. Simplified Representation of a DBS

The database system environment



SOURCE: Course Technology/Cengage Learning

The Database System Environment (cont'd.)

- Hardware: all the system’s physical devices
- Software: three types of software required
  - Operating system software
  - DBMS software
  - Application programs and utility software

The Database System Environment (cont'd.)

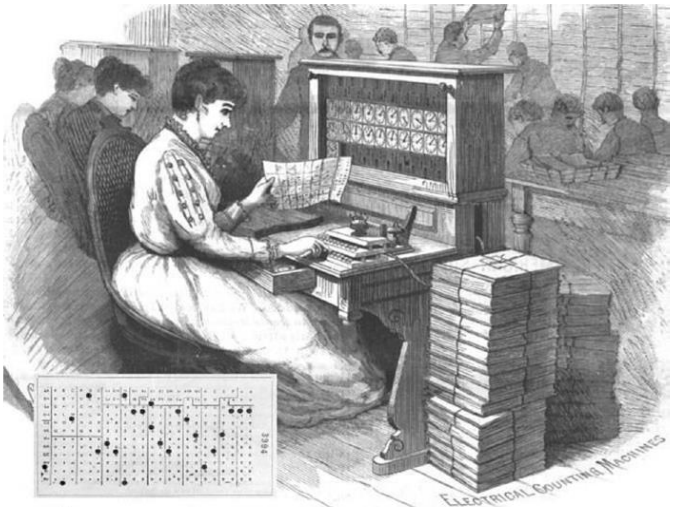
- People: all users of the database system
  - System and database administrators
  - Database designers
  - Systems analysts and programmers
  - End users
- Procedures: instructions and rules that govern the design and use of the database system
- Data: the collection of facts stored in the database

The Database System Environment (cont'd.)

- Database systems are created and managed at different levels of complexity
- Database solutions must be cost-effective as well as tactically and strategically effective
- Database technology already in use affects selection of a database system

History of DBMS

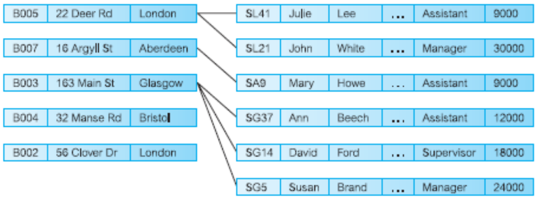
Processing data in 19th Century



a Hollerith tabulating machine being used to process the U.S. census in 1890.

The first database revolution

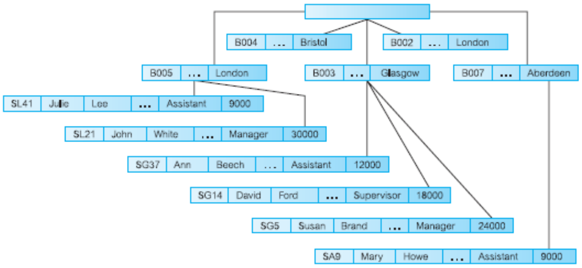
Network data model
records appearing as nodes (also called segments) and sets as edges in the graph



B005 | 22 Deer Rd | London
B007 | 16 Argyll St | Aberdeen
B003 | 163 Main St | Glasgow
B004 | 32 Manse Rd | Bristol
B002 | 56 Clover Dr | London

SL41 | Julie | Lee | ... | Assistant | 9000
SL21 | John | White | ... | Manager | 30000
SA9 | Mary | Howe | ... | Assistant | 9000
SG37 | Ann | Beech | ... | Assistant | 12000
SG14 | David | Ford | ... | Supervisor | 18000
SG5 | Susan | Brand | ... | Manager | 24000

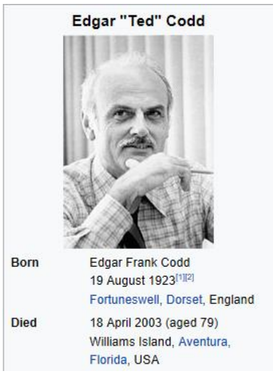
A hierarchical model can be represented as a tree graph, with records appearing as nodes (also called segments) and sets as edges. The hierarchical model allows a node to have only one parent



B004 | ... | Bristol
B002 | ... | London
B005 | ... | London
B003 | ... | Glasgow
B007 | ... | Aberdeen
SL41 | Julie | Lee | ... | Assistant | 9000
SL21 | John | White | ... | Manager | 30000
SG37 | Ann | Beech | ... | Assistant | 12000
SG14 | David | Ford | ... | Supervisor | 18000
SG5 | Susan | Brand | ... | Manager | 24000
SA9 | Mary | Howe | ... | Assistant | 9000

Second database revolution

- Edgar Codd (IBM), 1970 : *A Relational Model of Data for Large Shared Data Banks*.
- This classic paper contained the core ideas that defined the *relational database model* that became the most significant—almost universal—model for database systems for a generation.

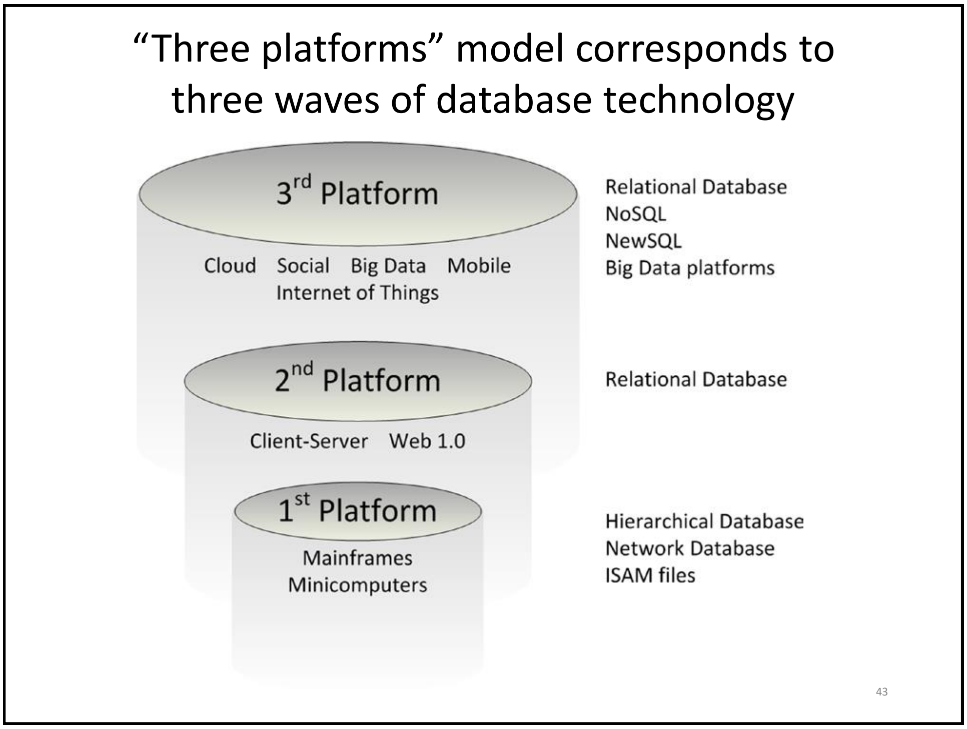


Edgar "Ted" Codd

Born	Edgar Frank Codd 19 August 1923[1][2] Fortuneswell, Dorset, England
Died	18 April 2003 (aged 79) Williams Island, Aventura, Florida, USA

The Third Database Revolution

- Google and Hadoop
- By 2005, Google was by far the biggest website in the world
- In 2003, Google revealed details of the distributed file system GFS that formed a foundation for its storage architecture and in 2004 it revealed details of the distributed parallel processing algorithm *MapReduce*, which was used to create World Wide Web indexes. In 2006, Google revealed details about its *BigTable* distributed structured database.
- This concept formed the basis for the *Hadoop* project.



"Three platforms" model corresponds to three waves of database technology

3rd Platform

Cloud Social Big Data Mobile
Internet of Things

Relational Database
NoSQL
NewSQL
Big Data platforms

2nd Platform

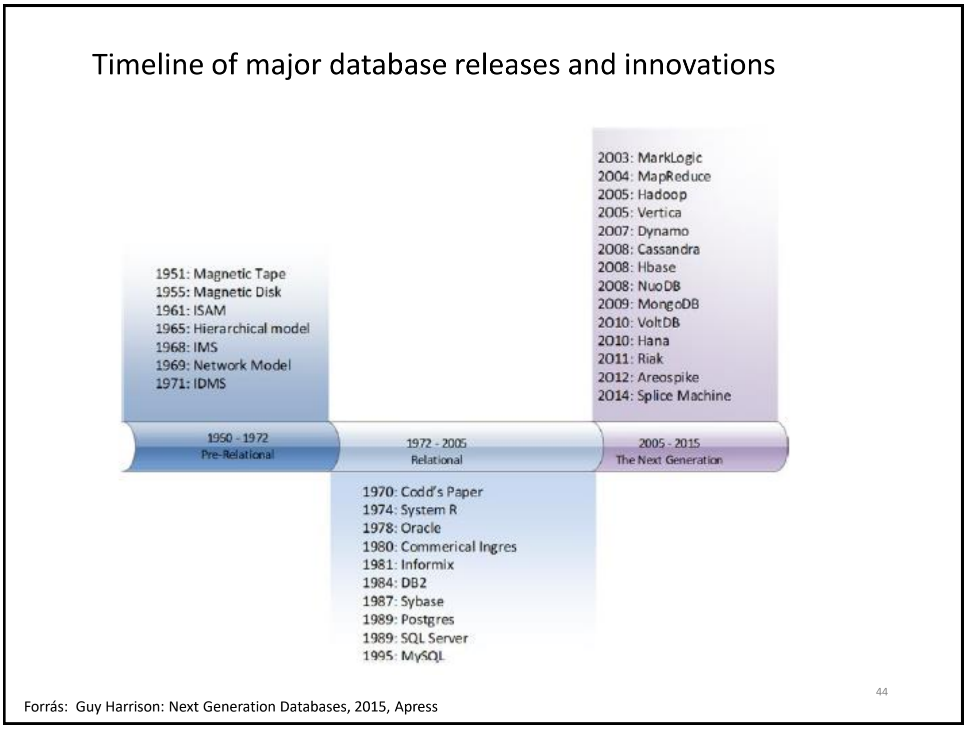
Client-Server Web 1.0

Relational Database

1st Platform

Mainframes
Minicomputers

Hierarchical Database
Network Database
ISAM files



Timeline of major database releases and innovations

1950 - 1972
Pre-Relational

1951: Magnetic Tape
1955: Magnetic Disk
1961: ISAM
1965: Hierarchical model
1968: IMS
1969: Network Model
1971: IDMS

1972 - 2005
Relational

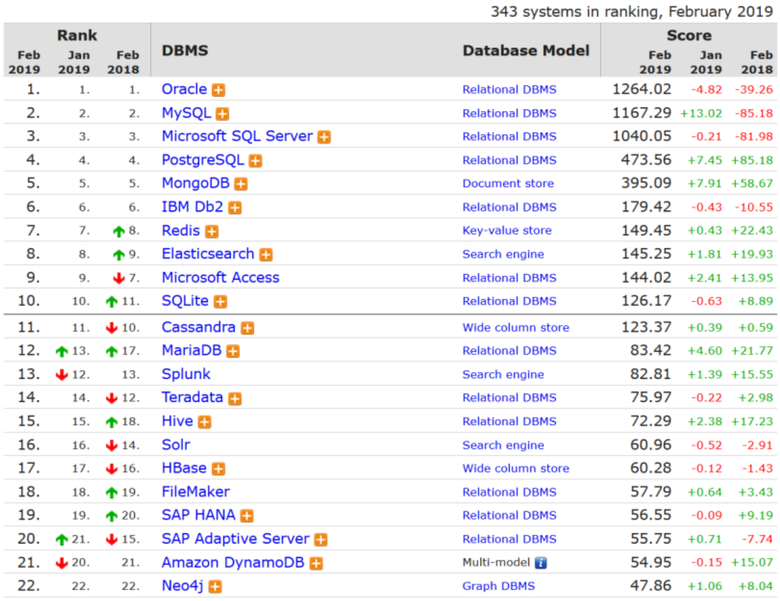
1970: Codd's Paper
1974: System R
1978: Oracle
1980: Commerical Ingres
1981: Informix
1984: DB2
1987: Sybase
1989: Postgres
1989: SQL Server
1995: MySQL

2005 - 2015
The Next Generation

2003: MarkLogic
2004: MapReduce
2005: Hadoop
2005: Vertica
2007: Dynamo
2008: Cassandra
2008: Hbase
2008: NuoDB
2009: MongoDB
2010: VoltDB
2010: Hana
2011: Riak
2012: Areospike
2014: Splice Machine

Forrás: Guy Harrison: Next Generation Databases, 2015, Apress

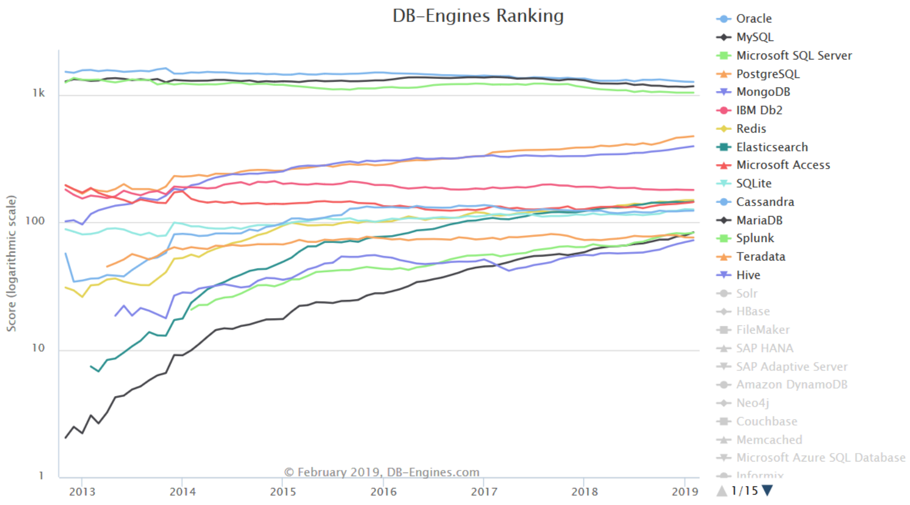
DB-Engines Ranking

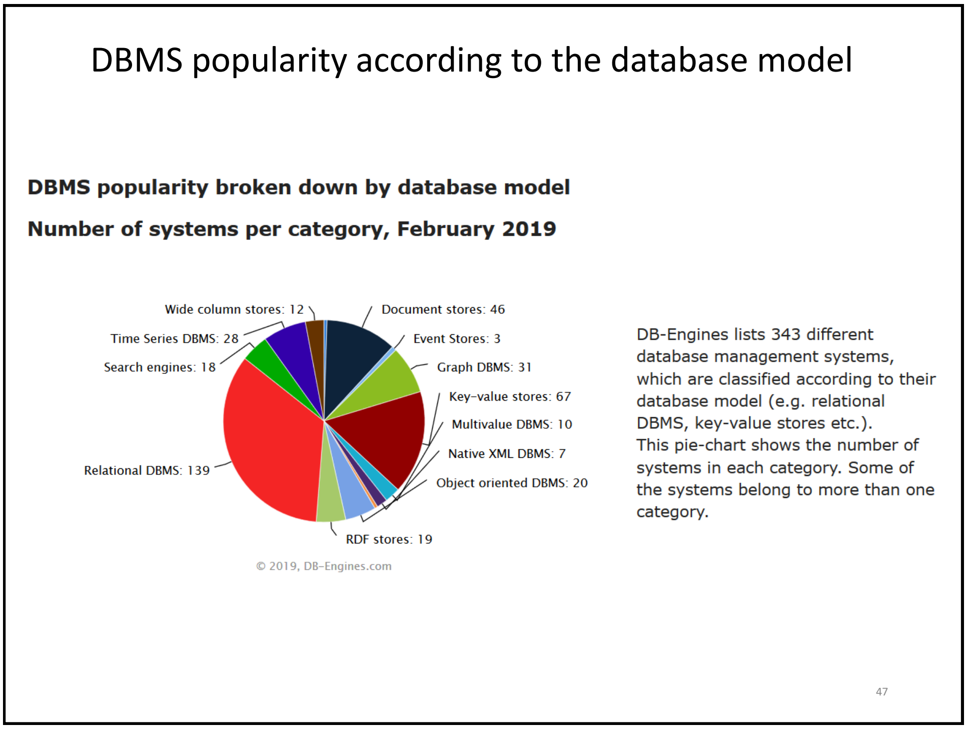


343 systems in ranking, February 2019

Rank			DBMS	Database Model	Score		
Feb 2019	Jan 2019	Feb 2018			Feb 2019	Jan 2019	Feb 2018
1.	1.	1.	Oracle ⊞	Relational DBMS	1264.02	-4.82	-39.26
2.	2.	2.	MySQL ⊞	Relational DBMS	1167.29	+13.02	-85.18
3.	3.	3.	Microsoft SQL Server ⊞	Relational DBMS	1040.05	-0.21	-81.98
4.	4.	4.	PostgreSQL ⊞	Relational DBMS	473.56	+7.45	+85.18
5.	5.	5.	MongoDB ⊞	Document store	395.09	+7.91	+58.67
6.	6.	6.	IBM Db2 ⊞	Relational DBMS	179.42	-0.43	-10.55
7.	7.	↑ 8.	Redis ⊞	Key-value store	149.45	+0.43	+22.43
8.	8.	↑ 9.	Elasticsearch ⊞	Search engine	145.25	+1.81	+19.93
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	144.02	+2.41	+13.95
10.	10.	↑ 11.	SQLite ⊞	Relational DBMS	126.17	-0.63	+8.89
11.	11.	↓ 10.	Cassandra ⊞	Wide column store	123.37	+0.39	+0.59
12.	↑ 13.	↑ 17.	MariaDB ⊞	Relational DBMS	83.42	+4.60	+21.77
13.	↓ 12.	13.	Splunk	Search engine	82.81	+1.39	+15.55
14.	14.	↓ 12.	Teradata ⊞	Relational DBMS	75.97	-0.22	+2.98
15.	15.	↑ 18.	Hive ⊞	Relational DBMS	72.29	+2.38	+17.23
16.	16.	↓ 14.	Solr	Search engine	60.96	-0.52	-2.91
17.	17.	↓ 16.	HBase ⊞	Wide column store	60.28	-0.12	-1.43
18.	18.	↑ 19.	FileMaker	Relational DBMS	57.79	+0.64	+3.43
19.	19.	↑ 20.	SAP HANA ⊞	Relational DBMS	56.55	-0.09	+9.19
20.	↑ 21.	↓ 15.	SAP Adaptive Server ⊞	Relational DBMS	55.75	+0.71	-7.74
21.	↓ 20.	21.	Amazon DynamoDB ⊞	Multi-model 🛈	54.95	-0.15	+15.07
22.	22.	22.	Neo4j ⊞	Graph DBMS	47.86	+1.06	+8.04

Source: db-engines.com





DBMS popularity according to the database model

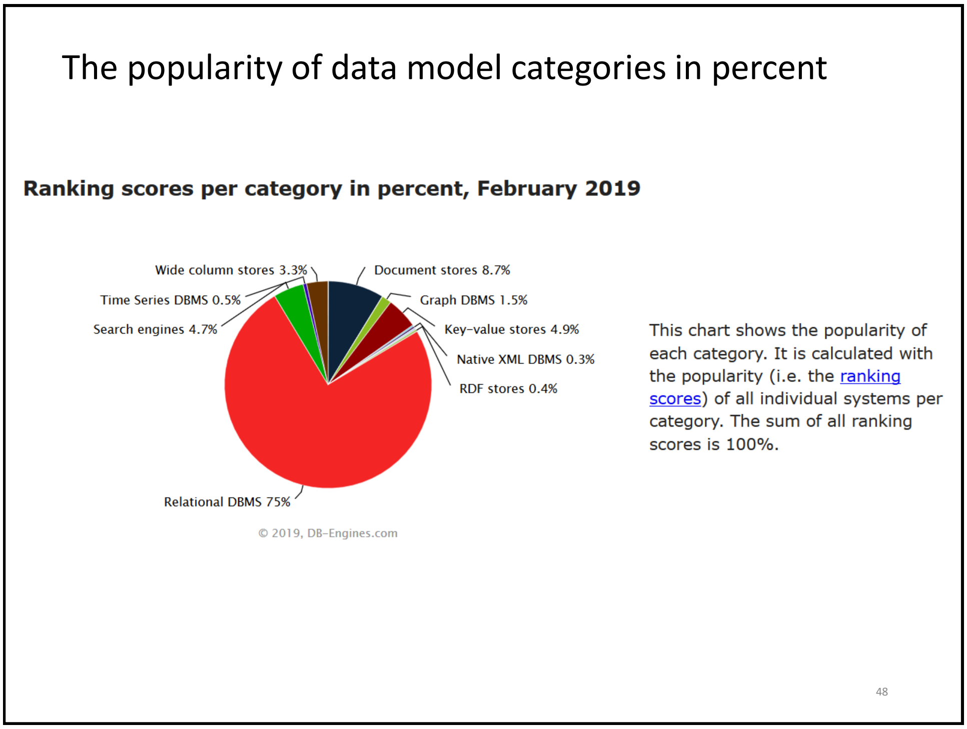
DBMS popularity broken down by database model

Number of systems per category, February 2019

© 2019, DB-Engines.com

DB-Engines lists 343 different database management systems, which are classified according to their database model (e.g. relational DBMS, key-value stores etc.). This pie-chart shows the number of systems in each category. Some of the systems belong to more than one category.

47



The popularity of data model categories in percent

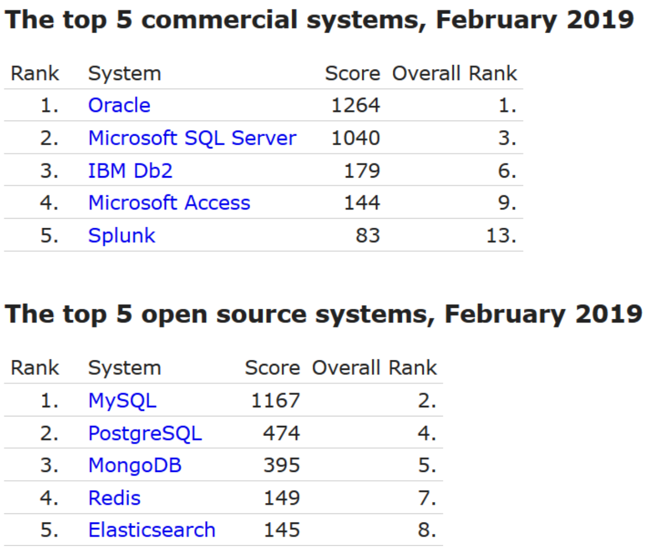
Ranking scores per category in percent, February 2019

© 2019, DB-Engines.com

This chart shows the popularity of each category. It is calculated with the popularity (i.e. the ranking scores) of all individual systems per category. The sum of all ranking scores is 100%.

48

Popularity rank of commercial and open source systems



The top 5 commercial systems, February 2019

Rank	System	Score	Overall Rank
1.	Oracle	1264	1.
2.	Microsoft SQL Server	1040	3.
3.	IBM Db2	179	6.
4.	Microsoft Access	144	9.
5.	Splunk	83	13.

The top 5 open source systems, February 2019

Rank	System	Score	Overall Rank
1.	MySQL	1167	2.
2.	PostgreSQL	474	4.
3.	MongoDB	395	5.
4.	Redis	149	7.
5.	Elasticsearch	145	8.

Thank you for your attention!