

*Information Systems*  
*An Overview of Database Management*

Nikolaj Popov

Research Institute for Symbolic Computation  
Johannes Kepler University of Linz, Austria  
`popov@risc.uni-linz.ac.at`

# Outline

Introduction

What Are Database Systems?

What Is a Database?

Why to Use Database?

Data Independence

Brief Overview of Systems

Summary

# Outline

Introduction

What Are Database Systems?

What Is a Database?

Why to Use Database?

Data Independence

Brief Overview of Systems

Summary

# Introduction

Goal of if this lecture:

- ▶ Explain what a database system is and why database systems are desirable.
- ▶ Briefly discuss the difference between relational systems and others.

# Introduction

- ▶ **Database system**: a computerized record-keeping system.

# Introduction

- ▶ **Database system**: a computerized record-keeping system.
- ▶ **Database**: a repository or a container for a collection of computerized data files.

# Introduction

- ▶ **Database system**: a computerized record-keeping system.
- ▶ **Database**: a repository or a container for a collection of computerized data files.
- ▶ Operations on databases:
  - ▶ Adding new files to the database
  - ▶ Inserting data into existing files
  - ▶ Retrieving data from existing files
  - ▶ Deleting data from existing files
  - ▶ Changing data in existing files
  - ▶ Removing existing files from the database
  - ▶ etc.

# Introduction

## Example (The Wine Cellar Database)

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
6	Chardonnay	Simi	2000	4	2000
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumè Blanc	Ch. St. Jean	2002	4	2003
43	Cab. Sauv.	Windsor	1995	12	2004
51	Pinot Noir	Fetzer	1997	3	2004
58	Merlot	Clos du Bois	1998	9	2004

File CELLAR



# Introduction

## Example (The Wine Cellar Database)

<u>BIN#</u>	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
6	Chardonnay	Simi	2000	4	2000
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumè Blanc	Ch. St. Jean	2002	4	2003
43	Cab. Sauv.	Windsor	1995	12	2004
51	Pinot Noir	Fetzer	1997	3	2004
58	Merlot	Clos du Bois	1998	9	2004

File CELLAR

## Retrieval:

```
SELECT    WINE, BIN#, PRODUCER
FROM      CELLAR
WHERE     READY = 2004 ;
```

WINE	<u>BIN#</u>	PRODUCER
Cab. Sauv.	43	Windsor
Pinot Noir	51	Fetzer
Merlot	58	Clos du Bois

# Introduction

## Example (The Wine Cellar Database)

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
6	Chardonnay	Simi	2000	4	2000
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumè Blanc	Ch. St. Jean	2002	4	2003
43	Cab. Sauv.	Windsor	1995	12	2004
51	Pinot Noir	Fetzer	1997	3	2004
58	Merlot	Clos du Bois	1998	9	2004

File CELLAR

### Inserting new data:

```
INSERT  
INTO      CELLAR ( BIN#, WINE, PRODUCER, YEAR, BOTTLES, READY )  
VALUES    ( 53, 'Pinot Noir', 'Saintsbury', 2001, 6, 2005 ) ;
```

# Introduction

## Example (The Wine Cellar Database)

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
6	Chardonnay	Simi	2000	4	2000
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumè Blanc	Ch. St. Jean	2002	4	2003
43	Cab. Sauv.	Windsor	1995	12	2004
51	Pinot Noir	Fetzer	1997	3	2004
58	Merlot	Clos du Bois	1998	9	2004

File CELLAR

### Deleting existing data:

```
DELETE
FROM   CELLAR
WHERE  BIN# = 2 ;
```

# Introduction

## Example (The Wine Cellar Database)

BIN#	WINE	PRODUCER	YEAR	BOTTLES	READY
2	Chardonnay	Buena Vista	2001	1	2003
6	Chardonnay	Simi	2000	4	2000
12	Joh. Riesling	Jekel	2002	1	2003
21	Fumè Blanc	Ch. St. Jean	2002	4	2003
43	Cab. Sauv.	Windsor	1995	12	2004
51	Pinot Noir	Fetzer	1997	3	2004
58	Merlot	Clos du Bois	1998	9	2004

File CELLAR

### Changing existing data:

```
UPDATE CELLAR
SET BOTTLES = 4
WHERE BIN# = 43 ;
```

# Introduction

- ▶ SELECT, INSERT, DELETE, UPDATE are called **statements**, **commands**, or **operators**.
- ▶ In the previous example they are expressed in a language called **SQL**.
- ▶ The Term **update** sometimes refers to the three operators: INSERT, DELETE, UPDATE. Do not confuse!

# Introduction

- ▶ SELECT, INSERT, DELETE, UPDATE are called **statements**, **commands**, or **operators**.
- ▶ In the previous example they are expressed in a language called **SQL**.
- ▶ The Term **update** sometimes refers to the three operators: INSERT, DELETE, UPDATE. Do not confuse!
- ▶ Terminology. The same things are referred differently in different contexts:
  - ▶ Files, records, fields (when talking about database systems in general).
  - ▶ Tables, rows, columns (when talking about SQL systems).
  - ▶ Relations, tuples, attributes (in more formal discussions).

# Introduction

- ▶ In the CELLAR table the columns WINE and PRODUCER contain character-string data.
- ▶ All other columns contain integer data.
- ▶ However, columns may contain data of arbitrary complexity.

# Introduction

- ▶ In the CELLAR table the columns WINE and PRODUCER contain character-string data.
- ▶ All other columns contain integer data.
- ▶ However, columns may contain data of arbitrary complexity.
- ▶ We might extend the CELLAR table to include additional columns:
  - ▶ LABEL (photo of the bottle label).
  - ▶ REVIEW (review text from some wine magazine).
  - ▶ MAP (showing where the wine comes from).
  - ▶ AUDIO (recording containing our own tasting notes).
  - ▶ etc.



# Introduction

- ▶ In the CELLAR table the columns WINE and PRODUCER contain character-string data.
- ▶ All other columns contain integer data.
- ▶ However, columns may contain data of arbitrary complexity.
- ▶ We might extend the CELLAR table to include additional columns:
  - ▶ LABEL (photo of the bottle label).
  - ▶ REVIEW (review text from some wine magazine).
  - ▶ MAP (showing where the wine comes from).
  - ▶ AUDIO (recording containing our own tasting notes).
  - ▶ etc.
- ▶ Column data types.

# Introduction

- ▶ Column BIN# constitutes the **primary key** for the table CELLAR.
- ▶ Meaning: No two CELLAR rows ever contain the same BIN# value.
- ▶ We use underlining to indicate primary key columns.

# Outline

Introduction

**What Are Database Systems?**

What Is a Database?

Why to Use Database?

Data Independence

Brief Overview of Systems

Summary

# Database Systems

- ▶ Database system: computerized record-keeping system.
- ▶ Four major components:
  - ▶ data,
  - ▶ hardware,
  - ▶ software,
  - ▶ users.

# Data

- ▶ Database systems are available on machines of different size and power.
- ▶ Systems on large machines tend to be **multi-user**, on smaller machines – **single-user**.
- ▶ Multi-user systems: many users can access the database at the same time.
- ▶ Single-user systems: at most one user can access the database at the same time.
- ▶ The distinction is largely irrelevant as far as most of the users are concerned.
- ▶ Special problems of multi-user systems mainly are internal to the systems.
- ▶ Data in the system can be stored in a single database, or can be split across several databases.

# Data

- ▶ Data in the database is in general both **integrated** and **shared**.
- ▶ Integrated database:
  - ▶ unification of several distinct files,
  - ▶ any redundancy among those files partly or wholly eliminated.
- ▶ Shared database:
  - ▶ sharing among different users,
  - ▶ different users can access the same data, maybe at the same time.

# Data

## Example (Integrated Database)

- ▶ Database containing an EMPLOYEE file and an ENROLLMENT file.
- ▶ The EMPLOYEE file contains data about employee names, addresses, salaries, etc:

NAME	ADDRESS	DEPARTMENT	SALARY	...
------	---------	------------	--------	-----

- ▶ The ENROLLMENT file contains data about the enrollment of employees in training courses:

NAME	COURSE	...
------	--------	-----

# Data

## Example (Integrated Database)

- ▶ Database containing an EMPLOYEE file and an ENROLLMENT file.
- ▶ The EMPLOYEE file contains data about employee names, addresses, salaries, etc:

NAME	ADDRESS	DEPARTMENT	SALARY	...
------	---------	------------	--------	-----

- ▶ The ENROLLMENT file contains data about the enrollment of employees in training courses:

NAME	COURSE	...
------	--------	-----

- ▶ Assume the courses administration needs to know the department for each enrolled student.
- ▶ No need to include this information in the ENROLLMENT file. Can be discovered in the EMPLOYEE file.



# Data

## In integrated and shared databases

- ▶ any given user is concerned with a small portion of the total database,
- ▶ different users' portions will overlap in various ways,
- ▶ even if two users share the same portion of the database, their views might be different.

# Hardware

Hardware components of a database system:

- ▶ The secondary storage volumes, together with the associated I/O devices, device controllers, etc.
- ▶ The hardware processor(s) and associated main memory.

Not considered in this course.

# Software

- ▶ The **Database management system (DBMS)**: a layer of software between the physical database and the users.
- ▶ DBMS
  - ▶ handles all requests to the database,
  - ▶ shields users from hardware-level details,
  - ▶ is the most important software component of the system.
- ▶ Other software components: utilities, application development tools, design aids, transaction manager, etc.

Sometimes people use the term *database* instead of *DBMS*. Do not confuse!

# Users

Three classes of users:

- ▶ Application programmers:
- ▶ End users:
- ▶ Database administrator.

# Users

Three classes of users:

- ▶ **Application programmers:** Write applications in some programming language, which then access the database by issuing a request (SQL statement) to the DBMS.
- ▶ **End users:**
  
- ▶ **Database administrator.**

# Users

Three classes of users:

- ▶ **Application programmers:** Write applications in some programming language, which then access the database by issuing a request (SQL statement) to the DBMS.
- ▶ **End users:** Access the database interactively, via online application or using a system interface.
  
- ▶ **Database administrator.**

# Users

Three classes of users:

- ▶ **Application programmers:** Write applications in some programming language, which then access the database by issuing a request (SQL statement) to the DBMS.
- ▶ **End users:** Access the database interactively, via online application or using a system interface.
  - ▶ Most systems include at least one built-in application, query language processor.
  - ▶ Most systems provide additional built-in interfaces, to help end users choose items from a menu or fill in a form, in contrast of issuing explicit database requests: menu- or forms-driven interfaces vs command-driven interfaces.
- ▶ **Database administrator.**

# Outline

Introduction

What Are Database Systems?

**What Is a Database?**

Why to Use Database?

Data Independence

Brief Overview of Systems

Summary



# Persistent Data

- ▶ The data in a database **persists** because
  - ▶ once it has been accepted by the DBMS for entry into the database,
  - ▶ it can subsequently be removed from the database only by some explicit request, not a mere side effect.

# Persistent Data

- ▶ The data in a database **persists** because
  - ▶ once it has been accepted by the DBMS for entry into the database,
  - ▶ it can subsequently be removed from the database only by some explicit request, not a mere side effect.

## Database

A database is a collection of persistent data that is used by the application systems of some given enterprise.

# Entities and Relationships

## Example

Manufacturing company records information about:

- ▶ its *projects*,
- ▶ the *parts* that are used in those projects,
- ▶ the *suppliers* who supply parts,
- ▶ the *warehouses* where the parts are stored,
- ▶ the *employees* who work in the projects,
- ▶ etc.

Projects, parts, suppliers, warehouses, employees: basic entities.

# Entities and Relationships

## Example (Cont.)

In addition to basic entities, the company keeps information about **relationships** linking those basic entities together:

- ▶ each supplier *supplies* certain parts,
- ▶ each part is *supplied* by some supplier,
- ▶ parts are *used* in projects,
- ▶ projects *use* parts,
- ▶ etc.

# Entities and Relationships

## Example (Cont.)

In addition to basic entities, the company keeps information about **relationships** linking those basic entities together:

- ▶ each supplier *supplies* certain parts,
- ▶ each part is *supplied* by some supplier,
- ▶ parts are *used* in projects,
- ▶ projects *use* parts,
- ▶ etc.

Binary (and bidirectional) relationships.

- ▶ Ternary relationship: each supplier supplies certain parts to certain projects.
- ▶ Not equivalent to three binary relationships: supplier supplies parts, parts are used in projects, and projects are supplied by suppliers. (Why?)

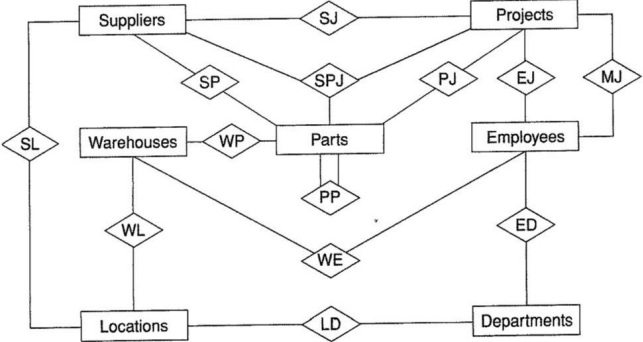
# Entities and Relationships

Important:

- ▶ Relationships are just as much a part of the data as are the basic entities.
- ▶ They must be represented in the database, like the basic entities.
- ▶ A relationship can be regarded as an entity in its own right.

# Entities and Relationships

Entity/Relationship (E/R) diagram from the previous example:



Representation:

- ▶ Entities by rectangles.
- ▶ Relationships by diamonds and connecting lines.

# Properties

- ▶ Entities (relationships included) can be regarded as having **properties**.
- ▶ Properties correspond to the information we wish to record about entities.
- ▶ Examples of properties: weight of a part, priority of a project, location of a supplier, plan of a warehouse, etc.



# Data and Data Models

Another view what data and databases are:

- ▶ **Data:** given facts from which additional facts can be inferred (by DBMS responding to a request).
- ▶ Logically, given facts correspond to true propositions.
- ▶ **Database:** collection of true propositions.

# Data and Data Models

SQL products are based on a the [relational model of data](#).

In the relational model

- ▶ data is represented by means of rows in tables,
- ▶ rows are interpreted as true propositions,
- ▶ operators are provided for operating on rows,
- ▶ operators support the process of inferring additional true propositions from the given ones.

# Data and Data Models

## Data Model

- ▶ An abstract, self-contained, logical definition of the objects, operators, etc. that together constitute the abstract machine with which users interact.
- ▶ The objects allow us to model the structure of data.
- ▶ The operators allow us to model its behavior.

### Model vs Implementation:

- ▶ Model is what the users have to know about,
- ▶ Implementation is what the users do not have to know about.

# Outline

Introduction

What Are Database Systems?

What Is a Database?

**Why to Use Database?**

Data Independence

Brief Overview of Systems

Summary

# Why to Use Database

The advantages of a database system over paper-based methods of bookkeeping:

- ▶ Compactness: No need in paper files.
- ▶ Speed: Machine is faster in retrieval than a human.
- ▶ Less drudgery: Mechanical tasks (maintaining files etc.) are better done by machines.
- ▶ Currency: Up-to-date information is available on demand at any time.
- ▶ Protection: The data can be better protected against unintentional loss and unlawful access.

# Why to Use Database

The advantages of a database system over paper-based methods of bookkeeping:

- ▶ Compactness: No need in paper files.
- ▶ Speed: Machine is faster in retrieval than a human.
- ▶ Less drudgery: Mechanical tasks (maintaining files etc.) are better done by machines.
- ▶ Currency: Up-to-date information is available on demand at any time.
- ▶ Protection: The data can be better protected against unintentional loss and unlawful access.

One more advantage in a multi-user environment:

- ▶ The database system provides the enterprize with centralized control of its data.

# Data Administration and Database Administration

- ▶ Data administrator
- ▶ Database administrator (DBA)

# Data Administration and Database Administration

- ▶ Data administrator
  - ▶ A person who has the central responsibility for the data.
  - ▶ Senior manager, not a technician (although familiar with the database system capabilities at a technical level).
  - ▶ Decides what data should be stored, establishes policies for maintaining and dealing with data.
- ▶ Database administrator (DBA)



# Data Administration and Database Administration

- ▶ Data administrator
  - ▶ A person who has the central responsibility for the data.
  - ▶ Senior manager, not a technician (although familiar with the database system capabilities at a technical level).
  - ▶ Decides what data should be stored, establishes policies for maintaining and dealing with data.
- ▶ Database administrator (DBA)
  - ▶ A technical person responsible for implementing data administrator's decisions.
  - ▶ An IT specialist.
  - ▶ Creates databases, puts in place the technical controls needed to enforce data administrator's policy decisions.
  - ▶ May have a staff of programmers and technical assistants.

# Advantages of Centralized Control

- ▶ Redundancy can be reduced.
- ▶ The data can be shared.
- ▶ Inconsistency can be avoided (to some extent).
- ▶ Transition support can be provided.
- ▶ Integrity can be maintained.
- ▶ Security can be enforced.
- ▶ Conflicting requirements can be balanced.
- ▶ Standards can be enforced.
- ▶ Data independence can be provided.

# Outline

Introduction

What Are Database Systems?

What Is a Database?

Why to Use Database?

**Data Independence**

Brief Overview of Systems

Summary

# Data Independence

- ▶ Two kinds of data independence: Physical and logical.
- ▶ Only physical data independence in this lecture.

# Data Dependence

- ▶ An application is **data-dependent**, if the physical representation of the data and (physical) access techniques can not be changed without affecting the application.
- ▶ Extremely undesirable property.

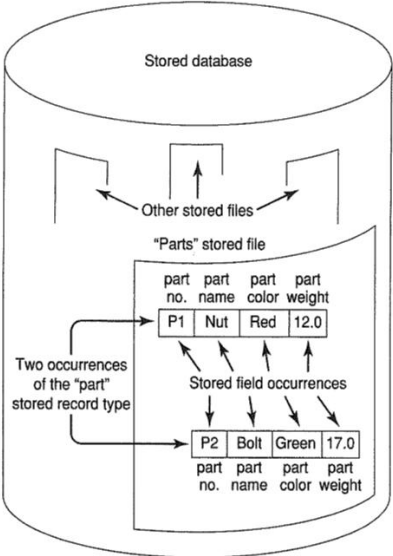
# Data Independence

- ▶ **Data independence**: the immunity of applications to change in physical representation and access techniques.
- ▶ What kind of changes we wish applications to be immune to?

# Data Independence

- ▶ **Stored fields:** smallest unit of stored data.
  - ▶ The database will contain many **occurrences** of each of several **types** of stored field.
  - ▶ Example: a database containing information about different kinds of parts might include a stored field type “part number”, and one occurrence of that stored field for each kind of part.
- ▶ **Stored record:** collection of stored fields.
  - ▶ A stored record **occurrence** consists of related stored field occurrences.
  - ▶ Database might contain many occurrences of stored record **type**.
- ▶ **Stored file:** collection of all currently existing occurrences of one type of stored record.

# Data Independence





# Data Independence

- ▶ In database systems the DBA might change the stored representation of data—stored fields, records, and files.
- ▶ Data as seen by applications does not change.

# Data Independence

Some aspects of the stores representation that might be subject to change:

- ▶ Representation of numerical data.
- ▶ Representation of character data.
- ▶ Units for numeric data.
- ▶ Data coding.
- ▶ Data materialization.
- ▶ Structure of stored records.
- ▶ Structure of stored files.

# Data Independence

- ▶ Database should be able to grow without impairing existing applications.
- ▶ Data independence is one of the reasons to separate data model from data implementation.

# Outline

Introduction

What Are Database Systems?

What Is a Database?

Why to Use Database?

Data Independence

**Brief Overview of Systems**

Summary

# Relational Systems and Others

- ▶ Relational system is a system in which
  - ▶ the data is perceived by the user as tables,
  - ▶ the operators available to the user derive “new” tables from “old” ones.
- ▶ Relation is basically a mathematical term for a table.
- ▶ Other systems:
  - ▶ Inverted list systems.
  - ▶ Hierarchical systems.
  - ▶ Network systems.
  - ▶ Object and object-relational systems.
  - ▶ Multi-dimensional systems.
  - ▶ Logic-based (deductive) systems.
  - ▶ Semistructured systems.

# Outline

Introduction

What Are Database Systems?

What Is a Database?

Why to Use Database?

Data Independence

Brief Overview of Systems

Summary

# Summary

- ▶ **Database system** can be thought as a computerized record-keeping system.

# Summary

- ▶ **Database system** can be thought as a computerized record-keeping system.
- ▶ It involves the **data** (stored in the **database**), **hardware**, **software** (in particular, **DBMS**) and **users**.



# Summary

- ▶ **Database system** can be thought as a computerized record-keeping system.
- ▶ It involves the **data** (stored in the **database**), **hardware**, **software** (in particular, **DBMS**) and **users**.
- ▶ Users can be divided into **application programmers**, **end users**, and the **DBA**.

# Summary

- ▶ **Database system** can be thought as a computerized record-keeping system.
- ▶ It involves the **data** (stored in the **database**), **hardware**, **software** (in particular, **DBMS**) and **users**.
- ▶ Users can be divided into **application programmers**, **end users**, and the **DBA**.
- ▶ DBA is responsible for administering the database and the database system with policies established by **DA**.

# Summary

- ▶ Databases are **integrated** and **shared**.

# Summary

- ▶ Databases are **integrated** and **shared**.
- ▶ They are used to store **persistent data** representing **entities** and **relationships** among entities.

# Summary

- ▶ Databases are **integrated** and **shared**.
- ▶ They are used to store **persistent data** representing **entities** and **relationships** among entities.
- ▶ One of the most important benefit of database systems is **data independence**.

# Summary

- ▶ Databases are **integrated** and **shared**.
- ▶ They are used to store **persistent data** representing **entities** and **relationships** among entities.
- ▶ One of the most important benefit of database systems is **data independence**.
- ▶ **Data independence** requires a sharp distinction between the data model and its implementation.

# Summary

- ▶ Databases are **integrated** and **shared**.
- ▶ They are used to store **persistent data** representing **entities** and **relationships** among entities.
- ▶ One of the most important benefit of database systems is **data independence**.
- ▶ **Data independence** requires a sharp distinction between the data model and its implementation.
- ▶ Relational systems are based on the **relational model**.