Advanced Management of Data

Advanced Database Applications

Weaknesses of RDBMS

An extended Data model

Standard Database Applications

- traditional business applications, such as
  - order processing
  - inventory control
  - banking
  - airline reservations
  - customer relationship management (CRM)
  - …

Advanced Database Applications

- computer-aided design (CAD)
- computer-aided manufacturing (CAM)
- computer-aided software engineering (CASE)
- network management systems
- office information systems (OIS) and multimedia systems
- digital publishing
- geographic information systems (GIS)
- scientific and medical applications
- expert systems
- interactive and dynamic Web sites

Advanced Database Applications

Computer-aided Design (CAD)

Databases have to store mechanical and electrical design data.

Challenges:

- design data is characterized by a large number of types, each with a small number of instances.
- designs may be very large, with many interdependent subsystems
- designs may evolve through time
- updates are far-reaching because of topological or functional relationships, and tolerances
- many design alternatives may be considered for each component
- Cooperative Engineering: Many staff members may be involved with the design, and may work in parallel on multiple versions of a large design.

Advanced Database Applications

Computer-aided manufacturing (CAM)

Databases store similar data to a CAD system, in addition to production-related data.

Challenges:

- hierarchical organization of data
- large volumes of data
- complex relationships between data
- realtime access and navigation through the data may be required
- response to changing conditions necessary, which may be described by rules or based on complex historical data

Advanced Database Applications

Computer-aided software engineering (CASE)

Databases store data relating to the stages of the software development lifecycle:

- planning
- requirements collection and analysis
- design
- implementation
- testing, maintenance, and documentation

Challenges:

- designs may be extremely large
- cooperative engineering is the norm

Advanced Database Applications

Network management systems

coordinate the delivery of communication services across a computer network, such as

- network path management
- problem management
- network planning

Challenges:

- handling of complex data
- real-time performance
- continuous operation

Advanced Database Applications

Office information systems (OIS) and multimedia systems

Databases store data relating to the computer control of information in a business, including electronic mail, documents, invoices, …

Challenges:

- need to handle a wider range of data types other than names, addresses, dates, and money, such as free-form text, photographs, diagrams, and audio and video sequences
- documents may have a specific structure imposed on them, perhaps described using a markup language such as HTML, or XML.
- Documents may be shared among many users using systems such as electronic mail and bulletin-boards based on Internet technology.

Advanced Database Applications

Digital Publishing

The publishing industry is undergoing profound changes in business practices.

Books, journals, papers, and articles are stored electronically and delivered over high-speed networks to consumers.

As with office information systems, digital publishing is being extended to handle multimedia documents consisting of text, audio, image, and video data and animation.

In some cases, the amount of information available to be put online is enormous, in the order of petabytes (10^{15} bytes).

Advanced Database Applications

Geographic information systems (GIS)

Databases store various types of spatial and temporal information, such as that used in land management and underwater exploration.

Challenges

- immense size of data
- complex searches
- multi-user support with very heavy volumes of information requests

Example

- EOS (Earth Observing System) is a collection of satellites launched by NASA in the 1990s to gather information that support scientists concerned with long-term trends regarding the earth's atmosphere, oceans, and land
- EOS data will be integrated with other data sources and will be stored in EOSDIS (EOS Data and Information System)

Advanced Database Applications

Scientific and medical applications

- have to store complex data representing systems such as molecular models for synthetic chemical compounds and genetic material

Expert systems

- may store knowledge and rule bases for AI applications.

Interactive and dynamic Websites

- e.g. web-shops with big online catalogs, customized 3D-renderings of products, …

Weaknesses of RDBMS

- poor representation of real-world entities
- semantic overloading
- poor support for integrity and enterprise constraints
- homogeneous data structure
- limited operations
- difficulty handling recursive queries
- impedance mismatch
- other problems associated with concurrency, schema changes and poor navigational access

Weaknesses of RDBMS

Poor representation of real-world entities

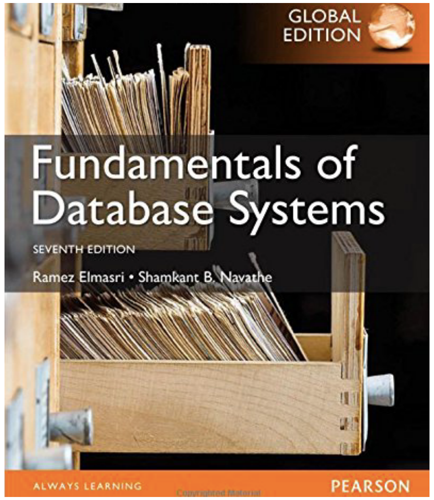
The process of deriving relations from a conceptual database often leads to the creation of relations that do not directly correspond to modeled entity types.

Cause

- complex entities consists of other entities
- entities are described by multi-valued attributes

Result

- fragmentation of a entity type into many relations
- is inefficient and leads to many joins during query processing



Example (1)

Imagine to create a simplified model of scientific documents / books.

Each book has a title, a publisher, may have several authors and may be described by several keywords.

In the relational model we would have to derive three relations to realize multivalued attributes.

To view books as a whole, we need to join all relations.

Books

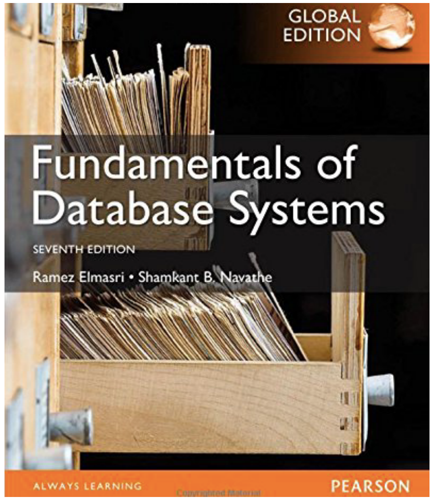
Title	Publisher
Fundamentals of Database Systems	Pearson

Authors

Title	Author
Fundamentals of Database Systems	Elmasri
Fundamentals of Database Systems	Navathe

Keywords

Title	Keyword
Fundamentals of Database Systems	Database
Fundamentals of Database Systems	Data Mining
Fundamentals of Database Systems	SQL



Example (2)

Even the joined relation does not provide a natural look on books and shows many redundancies

Result relation after Joining *Books*, *Authors* and *Keywords*			
<u>Title</u>	Publisher	<u>Author</u>	<u>Keyword</u>
Fundamentals of Database Systems	Pearson	Elmasri	Database
Fundamentals of Database Systems	Pearson	Elmasri	Data Mining
Fundamentals of Database Systems	Pearson	Elmasri	SQL
Fundamentals of Database Systems	Pearson	Navathe	Database
Fundamentals of Database Systems	Pearson	Navathe	Data Mining
Fundamentals of Database Systems	Pearson	Navathe	SQL

Weaknesses of RDBMS

Semantic Overloading

The relational model uses only one construct for representing data and relationships between data: the relation.

There is no mechanism to distinguish between entities and relationships, or to distinguish between different kinds of relationship that exist between entities.

➔ the relational model is said to be semantically overloaded.

Weaknesses of RDBMS

Poor support for integrity and general constraints

Integrity refers to the validity and consistency of stored data.

Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate.

Many commercial systems do not fully support these constraints and it is necessary to build them into the applications. This is dangerous and can lead to duplication of effort and inconsistencies.

There is no support for general constraints in the relational model, which again means that they have to be built into the DBMS or the application.

Weaknesses of RDBMS

Homogeneous data structure

The relational model assumes both horizontal and vertical homogeneity.

- Horizontal homogeneity means that each tuple of a relation must be composed of the same attributes
- Vertical homogeneity means that the values in a particular column of a relation must all come from the same domain

The first normal form requires that the intersection of a row and column must be an atomic value.

Drawbacks of Homogeneity

- too restrictive for many (more complex) “real-world” objects
- leads to unnatural and inefficient joins

Weaknesses of RDBMS

Limited Operations

The relational model has only a fixed set of operations, such as set and tuple-oriented operations, operations that are provided in the SQL specification.

Since (at least older versions of) SQL is not computationally complete, new operations cannot be specified.

This is too restrictive to model the behavior of many real-world objects.

Difficulty handling recursive queries

Atomicity of data means that repeating groups are not allowed in the relational model.

As a result, it is extremely difficult to handle recursive queries, that is, queries about relationships that a relation has with itself (directly or indirectly).

Example

staffNo	managerstaffNo
S005	S004
S004	S003
S003	S002
S002	S001
S001	NULL

commons.wikimedia.org

Consider the shown simplified *Staff* relation, which stores staff numbers and the corresponding manager's staff number.

How do we find all the managers who directly or indirectly manage staff member S005?

Analogously, how can we find the number of matryoshkas, which we have to open, in order to „free" a certain matryoshka.

Weaknesses of RDBMS

Impedance Mismatch

Older versions of SQL lack computational completeness and could not be used to develop more complex database applications.

To overcome this problem, SQL can be embedded in a procedural programming language.

However, this approach produces an impedance mismatch, because different programming paradigms are mixed:

- SQL is a declarative language that handles many rows of data at a time
- common high-level languages (such as C) are procedural languages that can handle only one row of data at a time

Since the type systems are different, type checking is prevented and necessary conversions require a lot of programming effort and runtime resources.

Weaknesses of RDBMS

Transactions

Transactions in business processing are generally short-lived and the concurrency control primitives and protocols such as two-phase locking are not particularly suited for long-duration transactions, which are more common for complex design objects.

Schema Changes

Database administrators must intervene to change database structures, and typically programs that access these structures must be modified to adjust to the new structures. These are slow and cumbersome processes even with current technologies.

As a result, most organizations are locked into their existing database structures. Even if they are willing and able to change the way they do business to meet new requirements, they are unable to make these changes, because they cannot afford the time and expense required to modify their information systems.

To meet the requirement for increased flexibility, we need a system that caters for natural schema evolution.

Weaknesses of RDBMS

Accessing records in databases can be done

- through declarative statements with selection based on one or more predicates - which is called associative access
- based on movement between individual records - which is called navigational access

RDBMSs

- use content-based associative access
- are poor at navigational access

Please note

Navigational access is important for many of the complex applications we discussed.

Alternatives

Object-Oriented DBMSs (OODBMS)?

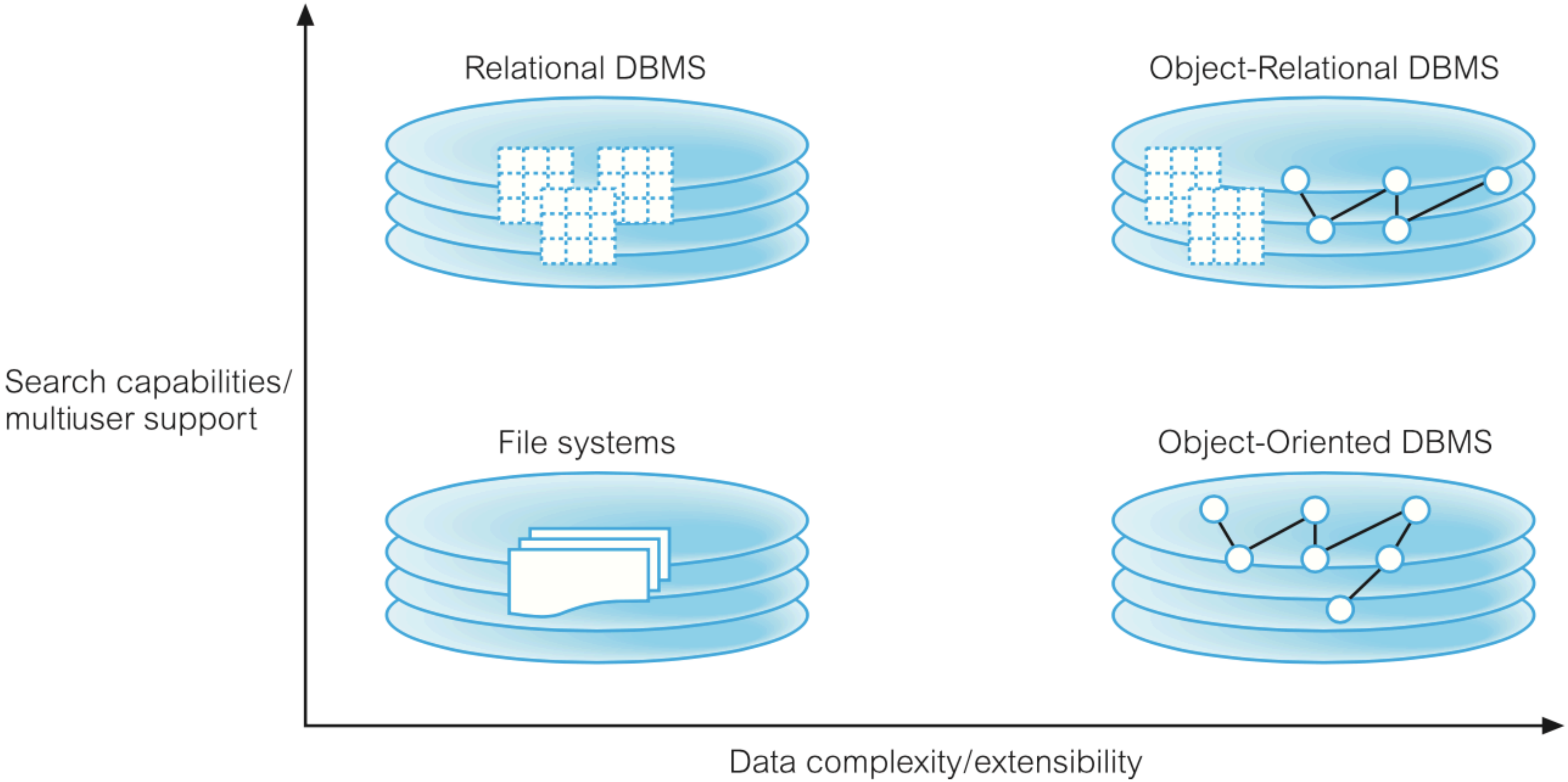
Most advanced database applications use object-oriented features, which should be supported directly by a DBMS, such as

- a user-extensible type system
- encapsulation
- inheritance
- polymorphism and dynamic binding of methods
- complex objects including non-first normal form objects, and object identity

Extending RDBMs?

Many of the addressed weaknesses of RDMBSs are not inherent to the relational model and could be addressed by extending RDBMs.

View of the database world



[Connolly & Begg]

Object-Relational Database Systems (ORDBMS)

Business View

Relational DBMSs dominate the database industry.

Hence, relational DBMSs should be extended with required features of advanced applications and remain compatible with existing applications

Extended Data Model?

Unfortunately, there is no single extended relational model but all incorporate some concept of object ➔ notion Object-Relational Database Systems (ORDBMS)

Implementations

Three of the leading RDBMS vendors - Oracle, Microsoft, and IBM - have all extended their systems into ORDBMSs, although the functionality provided by each is slightly different.

Non First Normal Form (NF^2)

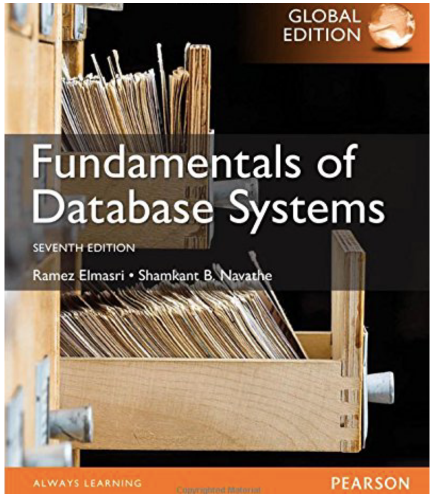
The First Normal Form (1NF) is the fundamental requirement for the relational model and states that values of attributes have to be atomic.

Simultaneously, 1NF is responsible for several drawbacks of the relational model
➔ an extended data model should not require 1NF

Objects of the Non First Normal Form are defined as follows:

- Every atomic value (such as integer, float, string) is an object.
- If $a_1, a_2, \dots, a_n$ are distinct attribute names and $o_1, o_2, \dots, o_n$ are objects, then $(a_1{:}o_1,\ a_2{:}o_2,\ \dots,\ a_n{:}o_n)$ is a tuple object.
- If $o_1, o_2, \dots, o_n$ are objects, then $\{o_1,\ o_2,\ \dots,\ o_n\}$ is a set object.

Each tuple of a NF^2 relation can contain such objects.



NF²-Example

<table>
<tr><th colspan="4">Books as a NF² relation</th></tr>
<tr><th><u>Title</u></th><th>Publisher</th><th><u>Author</u></th><th><u>Keyword</u></th></tr>
<tr><td rowspan="6">Fundamentals of Database Systems</td><td rowspan="6">Pearson</td><td rowspan="3">Elmasri</td><td rowspan="2">Database</td></tr>
<tr></tr>
<tr><td rowspan="2">Data Mining</td></tr>
<tr><td rowspan="3">Navathe</td></tr>
<tr><td rowspan="2">SQL</td></tr>
<tr></tr>
</table>

- Books as NF² relation:
```
( Title:      Fundamentals of Database Systems,
  Publisher:  Pearson,
  Author:     {Elmasri, Navathe},
  Keyword:    {Database, Data Mining, SQL} )
```
- Books as NF² relation realizes an intuitive representation of one book as one tuple
- book data does not need to be distributed over several relations
- there is no redundancy, such as in a joined relation

NF^2-Operations (1)

Requirement

To remain compatible with the relational data model we need operations to transform a relation in first normal form into a NF^2 relation and vice versa.

Nest

Creates a set of values from one or more attributes if the values of the remaining attributes are identical.

Unnest

Inverse operation

Nest - Example (1)

A	B	C	D	E
x	c	v	s	r
b	e	o	i	u
v	y	k	l	p
x	c	g	h	j
v	y	t	g	x
a	c	v	s	y
u	c	v	s	z

Nest CDE

A	B	CDE$
x	c	{(v,s,r),(g,h,j)}
b	e	(o,i,u)
v	y	{(k,l,p),(t,g,x)}
a	c	(v,s,y)
u	c	(v,s,z)

A	B	C	D	E
x	c	v	s	r
b	e	o	i	u
v	y	k	l	p
x	c	g	h	j
v	y	t	g	x
a	c	v	s	y
u	c	v	s	z

Nest AE

AE$	B	C	D
{(x,r),(a,y),(u,z)}	c	v	s
(b,u)	e	o	i
(v,p)	y	k	l
(x,j)	c	g	h
(v,x)	y	t	g

A	B	C	D	E
x	c	v	s	r
b	e	o	i	u
v	y	k	l	p
x	c	g	h	j
v	y	t	g	x
a	c	v	s	y
u	c	v	s	z

Nest ABC

ABC$	D	E
(x,c,v)	s	r
(b,e,o)	i	u
(v,y,k)	l	p
(x,c,g)	h	j
(v,y,t)	g	x
(a,c,v)	s	y
(u,c,v)	s	z

Nest - Example (2)

Result relation after Joining *Books*, *Authors* and *Keywords*			
Title	Publisher	Author	Keyword
Fundamentals of Database Systems	Pearson	Elmasri	Database
Fundamentals of Database Systems	Pearson	Elmasri	Data Mining
Fundamentals of Database Systems	Pearson	Elmasri	SQL
Fundamentals of Database Systems	Pearson	Navathe	Database
Fundamentals of Database Systems	Pearson	Navathe	Data Mining
Fundamentals of Database Systems	Pearson	Navathe	SQL

Nest *Keyword*

Nest *Author*

Unnest *Keyword$*

Unnest *Author$*

<table>
<tr><th colspan="4">Books as NF² relation</th></tr>
<tr><th>Title</th><th>Publisher</th><th>Author$</th><th>Keyword$</th></tr>
<tr><td rowspan="3">Fundamentals of Database Systems</td><td rowspan="3">Pearson</td><td rowspan="2">Elmasri</td><td>Database</td></tr>
<tr><td>Data Mining</td></tr>
<tr><td>Navathe</td><td>SQL</td></tr>
</table>

NF^2 - Operations (2)

Comparison

Between set-valued attributes only set operations are possible, such as a test of subset inclusion.

Question

How to associate tuples using Cartesian product or Natural Join based on set-valued attributes?

Set Identity?

To use the set identity is an intuitive idea, but does not work since too few tuple qualify for the result.

Intersection Join

If there is an nonempty intersecting set between qualifying attributes, tuples are associated.

Example - Intersection Join

NF² Relation R

AE$	B	C	D
{(x,r),(a,y),(u,z)}	c	v	s
(b,u)	e	o	i
(v,p)	y	k	l
{(x,j),(v,x)}	c	g	h

NF² Relation S

AE$	F
(a,y)	a
(u,b)	b
{(v,x),(x,j)}	c
(x,r)	d

Intersection Join between R and S

AE$	B	C	D	F
(a,y)	c	v	s	a
(x,r)	c	v	s	d
{(x,j),(v,x)}	c	g	h	c

Unnest AE$

A	B	C	D	E	F
a	c	v	s	y	a
v	c	g	h	x	c
x	c	g	h	j	c
x	c	v	s	r	d