Database Concepts

Anne Denton

Department of Computer Science
North Dakota State University

Outline

Table of Contents

Some Motivation

- Databases (almost) need no motivation
- Most businesses are supported by one or more databases
- Mentions of "Data" and "Information" ubiquitous

(Although not all such mentions are as informed as you should be by the end of the course, see link below:
`http://dilbert.com/strips/comic/1995-11-17/`)

References

- R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems," 7th Edition (or earlier), Pearson, 2016
  - Covers approximately the same material as the course, but some of the examples are outdated
- T. Connolly and C. Begg, "Database Systems: A Practical Approach to Design, Implementation, and Management," 6th Edition (or earlier), Pearson 2015
  - Modeling convention match the course most closely, but some topics are missing
- J. A. Hoffer, V. Ramesh, and H. Topi, "Modern Database Management," 12th Edition (or earlier), Prentice Hall, 2015
  - Focus on practical examples (modeling slightly different)
- H. Garcia-Molina, J. D. Ullman, and J. Widom, "Database Systems: The Complete Book," 2nd Edition, Pearson, 2009
  - Focus on the systems aspect

Table of Contents

Definition of a database

- Logically coherent collection of data
- Designed and built for a specific purpose
- Represents some aspect of the real world: miniworld or Universe of Discourse (UoD)

Question 1

According to this definition of a database as a

- Logically coherent collection of data
- Designed and built for a specific purpose
- Representing some aspect of the real world: miniworld or Universe of Discourse (UoD)

what is rather a database:

1. A handwritten address booklet
2. The internet

Why Care About This Definition?

- Technology is a means to an end and not a goal in itself
- A database is only as good as its design
- Note the lines are increasingly blurred
  - Some modern NoSQL databases provide very little structure and impose few constraints
  - Even on the internet, with its loose structure, database concepts prevail
- Examples of Database Concepts for the World-Wide Web
  - Google search uses structured representation of link connectivity, but not within a conventional database
  - The software that powers Wikipedia uses a database `http://www.mediawiki.org/wiki/Manual:Database_layout`

Table of Contents

Brainstorming Question

Take some time to think about what you consider good reasons to use a database system

Speed and Size

- According to conventional wisdom, database systems used for speed
  - **True**, they highly optimized
  - **But**, speed may conflict with data integrity expectations
- Large size of data as motivation for using a database
  - **True**, they are optimized to scale to very large tables
  - **But**, they are designed for conventional applications, not text or multimedia

Question 2 (Multiple answers can be correct)

Conventional database management systems are designed for speed, i.e. they

1. Typically return query results in about one millisecond
2. Very rarely take more than one second to return results regardless of data size

Speed Requirements

- Guaranteed access times, e.g. $<$ 1s all day, every day often important
- Recovery requirements prevent $\sim 1\mu s$ access: Need to write logs to disk

Question 3

To understand size requirements of typical databases, consider the following example. The catalog of a library of about the size of NDSUs (500,000 titles) and about 200 characters of information per book

1. Fits into the memory of a recent PC
2. Normally does not fit into memory but does fit on a standard hard drive
3. Does not fit on a standard hard drive but does fit onto a RAID system
4. Requires speciality database hardware

Size Expectations

Size requirements of a typical library database

- 500,000 titles
- approx. 200 characters per entry
- $\Rightarrow$ 100 Megabytes
- Note that databases can be of any size

Expectations of Database Management Systems

- Data quality
  - Redundancy control: no inconsistently duplicated data
  - Enforcing integrity and domain constraints
  - Depends on database design!
- Flexibility
  - Power of queries
  - Creating multiple views of data
- Data integrity
  - Concurrency control: Each transaction executed as if there were no others
  - Recovery from failures that leave disk unharmed

Data Quality Expectations

- Redundancy control: no inconsistently duplicated data
  - Early in the course we will discuss how to model the application and then translate the model so that that there will be no “uncontrolled redundancy"
  - Later we will use the mathematical framework of normalization to test our design
  - To satisfy redundancy control, a database has to be designed correctly for the miniworld!
- Enforcing integrity and domain constraints
  - Maintaining data quality relies on constraint checking of database systems
  - Making sure that constraints are appropriate is part of the design!
  - We will only discuss implicit constraints (such as key and domain constraints)

Question 4 (Multiple answers can be correct)

One good reason to use a database is data quality, i.e., a conventional database management system

1. Tests the correctness of data entries
2. Disallows poor designs
3. Offers constraint checking
4. Allows preventing uncontrolled redundancy

Flexibility Expectations

- Power of queries
  - Flexibility of querying was main motivation for relational databases
  - Efficient implementation allowed breakthrough
  - Database designer does not have to predict all possible queries!
  - Relational algebra forms the mathematical basis
  - SQL is the language used by nearly all relational database systems
  - Learning how to combine tables flexibly is a core topic of this course!
- Creating multiple views of data
  - A separate achievement of relational databases is the power of granting privileges
  - Users can be granted or revoked privileges at the level of individual tables and at the level of individual commands

Question 5 (Multiple answers can be correct)

One good reason to use a database is flexibility, i.e., a conventional database management system allows

1. Flexibly combining different types of information
2. Flexibly changing data definitions during use
3. Flexibly defining which user has access to what information

Data Integrity Expectations

- Concurrency control: Each transaction executed as if there were no others
- Recovery from failures that leave disk unharmed
- The underlying mechanism for achieving both concurrency control and recovery from failure is called “transaction processing"
- Any query or combination of queries happens in a transaction

Transaction Processing

- Transactions are expected to satisfy ACID requirements

  Atomicity Cannot be broken down into smaller units

  Consistency (Consistency Preservation) A database that is in a consistent state before the transaction will still be so after the transaction

  Isolation Any operation appears as if it was the only one executed at the time

  Durability A transaction that was “committed" will never be undone

- Even practitioners routinely talk about ACID transactions as a service of relational systems!
- We will discuss transaction processing later in the course

Question 6 (Multiple answers can be correct)

One good reason to use a database is data integrity, i.e., a conventional database management system ensures that

1. Two users can interact with the database at the same time without interfering with each other
2. Even if a customer suffers a power outage, there won't be remnants of half-completed transactions in the database
3. If a fire destroys the system that hosts the database, no data will be lost

Course content relevant to objectives

- Data quality
  - Entity-Relationship modeling
  - Relational modeling
  - Normalization
- Power of queries
  - Relational algebra
  - SQL
  - Web applications
  - (Query optimization)
- Data integrity
  - Transaction processing

Table of Contents

One-off Applications

Many problems with one-off applications

- Incompatibility of data that is exchanged between them
- Programming effort
- Don’t scale to large applications
- Difficult to maintain

Shared Files

Systems that were designed to use shared files attempted to address

- Redundancy control
  - Avoiding inconsistently replicated information
- Concurrency control
  - Avoid data corruption due to multiple users accessing the same data

But required reimplementation of these features.

Hierarchical databases and network databases

- Hierarchical databases prevalent up to the 1970s
- Most common legacy systems
- Difficult to query
- Don’t scale as well as relational databases

Table of Contents

Relational Database Management Systems (RDBMS)

- Standard for traditional databases
- Proposed by E.F. Codd 1970 (Turing Award 1981)
- De-facto standard since the 1980s
- Have been extraordinarily successful

Question 7

The first major type of database management system beyond just sharing files between applications was

1. Hierarchical
2. Relational

Object-Oriented Databases

- Integration of object-oriented concepts into databases
- Problem: Performance
- Object-relational features now common in relational databases
- Fully object-oriented databases only successful in niche markets

Question 8

Most current transactional databases (e.g., for banks, travel agencies) are

1. Hierarchical
2. Network Databases
3. Relational
4. Object-Oriented

Question 9 (Multiple answers can be correct)

Object-oriented databases

1. Are the typical choice in conjunction with object-oriented programming
2. Can be useful for specialized applications
3. Are substantially slower than relational databases
4. Influenced object-relational features that are common in modern relational databases

Extensible Markup Language (XML)

- Used since the late 1990s
- Designed for semi-structured data
- Plain text, i.e., humanly readable
- Standard for exchange of data between relational databases
  - JSON is a more compact alternative that has similar properties
- Inherently hierarchical structure makes querying difficult

Question 10 (Multiple answers can be correct)

Extendable Markup Language (XML)

1. Has largely replaced older relational systems
2. Is a standard for semi-structured data
3. Can be used for representing relational content if necessary

Data warehouses and online analytical processing (OLAP)

- Data mining / analytical processing requires read access to entire tables, conflicts with updates
- Data warehouses use snapshots of data combined from multiple operational databases
- Multi-dimensional data cube model
- Typically underlying relational model
- OLAP attempts to do both efficiently

Question 11 (Multiple answers can be correct)

Data warehouses that extract, transform, and load the content from multiple transactional databases

1. Are not used anymore because transactional database management systems have become more centralized
2. Can be faster to query because they don't require the concurrency control that slows down transactional database management systems
3. Were among the first to offer a hypercube data model that allows flexible querying of large fact tables

NoSQL

- Distributed relational databases problematic
- Web-based businesses require data to be distributed and fault-tolerant
- Various types of databases without full SQL support exist, examples:
  - Key-value
  - Column-oriented
  - Document-oriented
  - Graph database
- Google and Amazon at forefront

Question 12 (Multiple answers can be correct)

NoSQL databases

1. Are designed for very large very distributed data
2. Have taken over many application areas where relational databases used to be standard
3. Have more limited constraint checking and concurrency control guarantees than relational databases
4. There are multiple types of NoSQL databases that are based on highly disparate expectations

Question 13

A graph (network) can be represented in

1. A relational or a graph database
2. Only a graph database
3. Neither a relational nor a graph database

Table of Contents

Real-time and active database technology

- Specialized databases exist for time-critical applications
- Standard databases have limited ability to respond to external events through triggers

Multimedia databases

- Standard databases assume similarly sized fields
- Large data stored as binary large objects (BLOBs)
- Multimedia databases offer special processing capabilities

Full-text Databases

- Special text-search features
- "Information retrieval" is a large research and application area
- Full-text databases operate on largely unstructured data
- Data is commonly indexed

Geographic information systems (GIS)

- Geospatial data can be raster or vector data
- GIS have geo-spatial functionality
  - Intersections between regions
  - Converting between reference systems, projections, and representations
- Typically have underlying RDBMS
  - Traditionally data were sometimes stored in files
  - Concurrency control and recovery are better for RDBMS than files
- Google Maps and Google Earth are similar but lack some functionality

Question 14 (Multiple answers can be correct)

Which of the following types of data benefit from a system that has special functionality designed for that data type rather than only a general purpose database management system

1. Integers
2. Geographical information
3. Multimedia data
4. Calendar dates

Table of Contents

Relational Database Management System (RDBMS)

- General purpose software
  - Can be used in almost any application domain
  - Specialized databases only needed when additional functionality is needed
- Meta data stored in catalog and queried as data
  - System tables are maintained by the RDBMS
  - They behave much like user tables and can be queried the same way
  - Being able to metadata like user data is a design principle of RDBMSs

Question 15

A database system has a separate metadata repository for table names and attribute names and types

1. True
2. False

Question 16 (Multiple answers can be correct)

You are asked to develop a database for your city. Which of the following statements are appropriate?

1. You look for a database that offers transaction processing for municipal data
2. If you can find an application that is specific to your needs, it will likely have an RDBMS as backend
3. If geographic information is of interest, you may need a geographic information system as part of your application
4. You should use generic table names since application-specific data should be stored in the middleware

Database System

- Application-specific programs + RDBMS + stored database $\Rightarrow$ database system
- Application-specific programs and queries interface with RDBMS
- RDBMS retrieves data from stored database

Question 17

The terms "database system" and "database management system" are synonyms

1. True
2. False

Table of Contents

Concept of a Relation

- A relation is essentially a table (details later)
- A relation can represent an entity
  - Person
  - Object
  - Class
- Alternatively a relation can represent a relationship, e.g.
  - A person has an object
  - An object is a type of other object
  - A person supervises another person
  - A webpage links to another webpage

Question 18 (Multiple answers can be correct)

Relations in a relational database can represent

1. Objects
2. People
3. Relationships
4. Constraints

Example of Relations

Student

student_id	name
7	Brown
16	White

Section

call_number	dept	course_number
1234	CSci	213
5678	CSci	366

Enrollment

student_id	call_number
7	5678
16	5678

Support for Multiple User Views

- Different users need different, potentially redundant, views of the data
- The same information does not have to be stored multiple times
  - Avoiding uncontrolled redundancy
  - Assumes good design!
- Controlled redundancy can be very helpful
  - RDBMS offers “Views" that recompute or cache commonly used derived information
  - Support for this need is central to SQL

Allows Avoiding Inconsistencies

Information seen by student:

Student

student_id	name	major	gpa
7	Brown	Computer Science	3.4

Information seen by instructor of a specific course:

Student

name	student_id	grade
Smith	7	A

Problem of uncontrolled redundancy

- Combination of id and name should only have been stored once
- Can be achieved by proper modeling (Entity-Relationship and relational modeling)
- Can be identified through normalization
- The above listings should be views and that are constructed from stored information

Question 19

Relational databases

1. Avoid any kind of redundancy in data
2. Avoid uncontrolled redundancy

Table of Contents

Database administrators, DBA

- Oversee and manage resources
- Authorize access
- Coordinate and monitor use

Database Designers

- Identify data to be stored and structure to be chosen
- Communicate with future users!
- Develop views for different user groups

End Users

- **Parametric end users**, e.g. retail clerks, use canned transactions
- **Casual end users**, e.g. managers (different information every time)
- **Sophisticated end users**, e.g. engineers, business analysts (most complex requirements)
- **Stand-alone users**, use personal database systems, e.g. for tax declarations

System Analysts and Software Engineers

- **System analysts** determine needs of parametric end users
- **Software engineers** implement specifications for canned transactions

System Developers

- **Designers an implementers of RDBMS** work on the RDBMS software itself
- **Tool developers** design additional (optional) software packages that help in the database design and use. Tools may be developed and marketed by independent vendors.

Question 20 (Multiple answers can be correct)

Which of the following database roles are typically intended for computer scientists

1. Parametric user
2. Database administrator
3. Database designer
4. Tool developer