Comp 155
Introduction to Database Management

Objectives

Provide a general introduction to the field of database management by:

- Providing a brief history of database management
- Describing 4 basic types of database models
- Examining the relational model
- Introducing basic terminology
- Describing the advantages and disadvantages of database processing
- Examining SQL standards

A Brief History of Database Management

- Before the introduction of DBMS, all data were stored in individual files
  - File management systems
  - Needed a file description
  - Huge problem to maintain
- Idea was to take the definition of a file's content and structure out of individual programs, and store it together with the data in a database

- Modern databases emerged in the 1960s
- IBM developed the Generalized Update Access Method (GUAM) in 1964 for North American Rockwell, the prime contractor for the APOLLO project
- GUAM was made available for the general public under the name Data Language/I (DL/I) in 1966

- DL/I became the data management component for the Information Management System (IMS), which was the dominant DBMS for many years
- In the mid-1960s, General Electric developed Integrated Data Store (I-D-S)

- The COnference on DAta SYstems Languages (CODASYL) tackled the problem of providing standards for DBMSs in the late 1960s
- In 1971, the CODASYL standards were presented to the American National Standards Institute (ANSI)

- 1970 E.F. (Ted) Codd published landmark paper in which he applied concepts from relational algebra to problems of storing large amounts of data
- Throughout the 1970s, the relational model was the subject of intense research
- Initially there was a lot of resistance to the relational model

- Codd's article triggered a flurry of relational database research, including a major research project in IBM.
- Goal of the project, called System /R, was to prove the workability of the relational concept and to provide some experience in actually implementing a relational DBMS.
- Work began in the mid 1970s.

- In 1974 & 1975, the first phase of System /R produced a minimal prototype.
- In addition, the project included work on database query languages. One of these languages was called SEQUEL, an acronym for Structured English Query Language.
- In 1978 & 1979 implemented System /R in some client sites. Also implement the query language which changed to SQL (Structured Query Language) for legal reasons.
- Despite the name change, the SEQUEL pronunciation remained

- Publicity about System /R attracted the attention of a group of engineers in Menlo Park, California.
- Saw a chance for a commercial product.
- In 1977 formed a company called Relational Software to build a relational DBMS based on SQL.
- Named their product “Oracle”.
- Shipped it in 1979 and it was the first commercially available relational DBMS.
- Beat IBM’s first product to market by a full 2 years.
- Aggressively sold their product and eventually renamed the company after its flagship.

- University of California, Berkeley also researched relational databases in the mid-1970s.
- Developed Ingres
- Had a query language called QUEL.
- In 1980, several professors left Berkeley and founded Relational Technology, Inc which built a commercial version of Ingres.
- Ingres and Oracle were arch-rivals.
- Although Ingres was technologically superior in many areas, lost out to Oracle’s aggressive marketing.

- Meanwhile, IBM was turning System /R into a commercial product called SQL/DS.
- Began shipping in 1982.
- In 1983, introduced Database 2 (DB2). Shipped this in 1985.
- By May 1985, Oracle claimed to have over 1000 installations, Ingres claimed a similar number and SQL/DS plus DB2 also around 1000.

- Still a struggle against traditional databases, but query languages (e.g. SQL, QUEL) pushed them forward.
- Market forces boosted popularity of SQL in the late 1980s.
- Through early 1990s, steadily improving SQL implementations and dramatic improvements in processor speeds made SQL a practical solution for transaction processing applications.

- SQL’s supremacy in the database world has not gone unchallenged.
- Early 1990s “object databases” look as though they will take over as did object-oriented programming.
- Lots of venture capital-backed “object database” companies.
- Hoped to make relational databases and their vendors obsolete.

- SQL and relational databases so far have more than withstood the challenge.
- Total revenues for object-oriented databases are measured in hundreds of millions of dollars (at best).
- Total revenues for SQL and relational DBMS tools & services are measured in tens of billions of dollars of sales per year.

Other data models

Four types of data models

- Network
- Hierarchical
- Relational
- Object-oriented

- Oracle10g is object relational

Hierarchical Model

- Perceived by the user as a collection of hierarchies, or trees
- More restrictive structure than a network model
- GUAM, DL/I, and IMS are examples of DBMSs that conform to the hierarchical model

Hierarchical Database Structure

DEPARTMENT
↕
FACULTY
↕
STUDENT

Network Model

- Perceived by the user as a collection of record types and relationships between these record types
- I-D-S and other CODASYL systems are examples of DBMSs that conform to the network data model

Network Database Structure

PUBLISHER ↔ **BOOK** ↔ **BOOK-AUTHOR** ↔ **AUTHOR**

BOOK ↔ **BOOK-BRANCH** ↔ **BRANCH**

Object-Oriented Model

- According to Rao (1994), "The object-oriented database (OODB) paradigm is the combination of object-oriented programming language (OOPL) systems and persistent systems. The power of the OODB comes from the seamless treatment of both persistent data, as found in databases, and transient data, as found in executing programs."

Relational Model

- In his 1985 *Computerworld* article, Ted Codd presented 12 rules that a database must obey if it is to be considered truly relational.
- Are the semiofficial definition of a relational database
- Rules were theoretically based and so represent more of an ideal goal than a true definition

Codd's 12 Rules

1. Information rule. All information in a relational database is represented explicitly at the logical level and in exactly one way – by values in a table.
2. Guaranteed access rule. Each and every datum (atomic value) in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name.
3. Systematic treatment of NULL values. NULL values (distinct from an empty character string or string of blank characters and distinct from zero or any other number) are supported in a fully relational DBMS for representing missing information and inapplicable information in a systematic way, independent of the data type.
4. Dynamic online catalog based on the relational model. The database description is represented at the logical level in the same way as ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

5. Comprehensive data sublanguage rule. A relational system may support several languages and various modes of terminal use (e.g. fill-in-the-blanks mode). However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings, and that is comprehensive in supporting all of the following items:
    a. Data definition
    b. View definition
    c. Data manipulation (interactive and by program)
    d. Integrity constraints
    e. Authorization
    f. Transaction boundaries (begin, commit and rollback)
6. View updating rule. All views that are theoretically updateable are also updateable by the system.
7. High-level insert, update and delete. The capability of handling a base relation or derived relation as a single operand applies not only to the retrieval of data, but also to the insertion, update and deletion of data.
8. Physical data independence. Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representations or access methods.
9. Logical data independence. Application programs and terminal activities remain logically unimpaired whenever information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.
10. Integrity independence. Integrity constraints specific to a particular relational database must be definable in the relational data sublanguage and storable in the catalog, not in the application programs.
11. Distribution independence. A relational DBMS has distribution independence.
12. Nonsubversion rule. If a relational system has a low-level (single records at a time) language, that low level cannot be used to subvert or bypass the integrity rules and constraints expressed in the higher-level relational language (multiple records at a time).

Basic Terminology

Tables or Relations or Files

R1C1	R1C2	R1C3	R1C4	R1C5
R2C1	R2C2	R2C3	R2C4	R2C5
R3C1	R3C2	R3C3	R3C4	R3C5
R4C1	R4C2	R4C3	R4C4	R4C5
R5C1	R5C2	R5C3	R5C4	R5C5

Rows/Tuples/Records

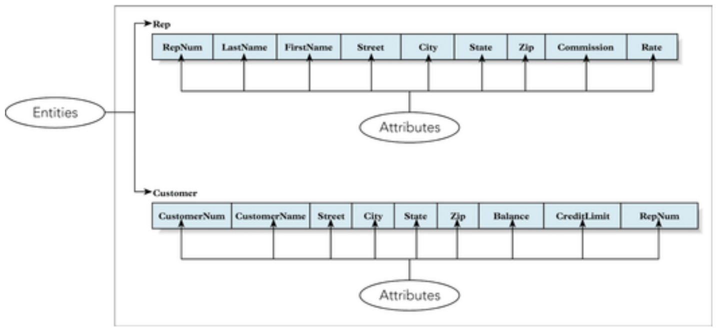
Columns/Attributes/Fields

Relational Model
In formal relational theory as defined by relational mathematics:
- A relation is a two-dimensional table (= file)
- A tuple is a row of the table (= record)
- An attribute is a column of the table (= field)

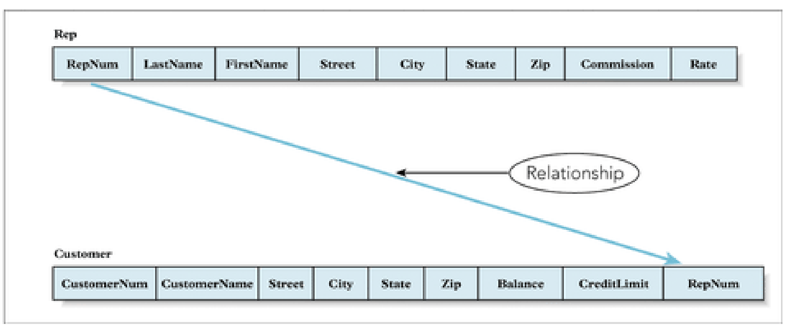
Entities, Attributes, and Relationships

- Entity – a person, place, thing, or event
- Attribute – a property of an entity
  - For the entity "Person," attributes could include eye color and height
- Relationship – an association between entities
  - Publishers are related to the books they publish, and a book is related to its publisher

Entities and Attributes



One-to-Many Relationship



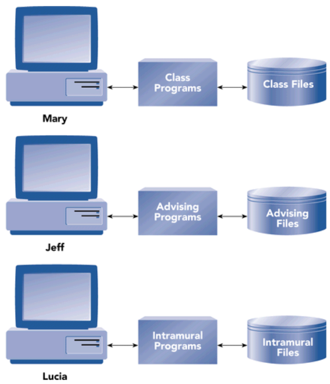
More Terminology

- A DBMS is a program or collection of programs whose function is to manage a database on behalf of the people who use it.
- Database design is the determination of the structure of the database.
- Data file – stores information on a single entity and the attributes of that entity
- Database – a structure that can store information about multiple types of entities, the attributes of these entities, and the relationships among the entities

Database Processing

- Database processing – the data to be processed are stored in a database and the data in the database are being manipulated by the DBMS
- Nondatabase approach
  - Duplication of data
  - Extremely difficult to fulfill requirements that involve data from more than one system

Nondatabase Approach

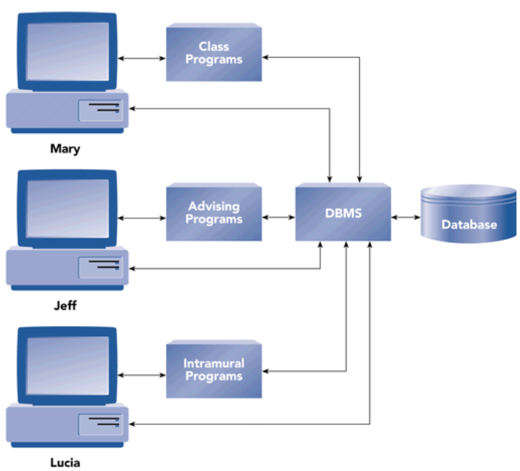


But you would like to know which classes a student is taking, the name of the student's advisor & in which intramural sports the student is participating

Database Processing

- Database approach
  - Common database managed by a DBMS
  - Each entity appears only once in the system, reducing the duplication of data
  - With all data being in a single database, it is possible to list all information concerning the entities

Database Approach



Advantages of Database Processing

- Getting more information from the same amount of data
- Sharing data
- Balancing conflicting requirements
- Controlling redundancy
- Facilitating consistency
- Improving integrity
- Expanding security
- Increasing productivity
- Providing data independence

Disadvantages of Database Processing

- Larger file size
- Increased complexity
- Greater impact of failure
- More difficult recovery

SQL Standards

- Development of standards for SQL has been one of the most important developments
- Official standards include:
- American National Standards Institute (ANSI)
- International Standards Organization (ISO)
- “De facto” standards include:
  - IBM’s DB2 product family
  - Oracle’s SQL dialect
- SQL-86 aka SQL-87
  - Largely based on DB2 SQL
  - Officially adopted as ANSI standard X3.135 in 1986
  - Adopted as an ISO standard in 1987

- SQL-89 or SQL1
  - Slightly revised and expanded from the 1986/7 standard.

- SQL-92 or SQL2
  - To address holes in the original standard, a new draft was prepared
  - Unlike the 1989 standard, specified features considerably beyond those in current commercial SQL products
  - Very controversial.
  - Finally approved in October 1992.
  - Represented a large step from SQL1, so proposed 3 levels of compliance:
    - Entry-Level
    - Intermediate-Level
    - Full

- SQL-99 or SQL3 or SQL:1999 ANSI/ISO
  - The 1992 standard was based on a purely relational model.
  - New standard extends the model to include object-oriented constructs.
  - Language became, object-relational

- SQL:2003 ANSI/ISO
  - Introduced XML support and fields with autogenerated values

- SQL:2006
  - Defines ways SQL can be used with XML
    - XQuery, the XML Query Language
- SQL:2008
  - Further language enhancements
    - TRUNCATE TABLE
    - ALTER COLUMN SET DATA TYPE “type”
    - Enhanced MERGE
    - INSTEAD OF type triggers

Summary

Provided a general introduction to the field of database management by:

- Providing a brief history of database management
- Describing 4 basic types of database models
- Examining the relational model
- Introducing basic terminology
- Describing the advantages and disadvantages of database processing
- Examining SQL standards.