Introduction to Database Systems

Introduction to Database Systems

- So, what is a database, anyway?
- An integrated, self-describing collection of data about related sets of things and the relationships among them

If you burned down all our plants, and we just kept our people and our information files, we should soon be as strong as ever.

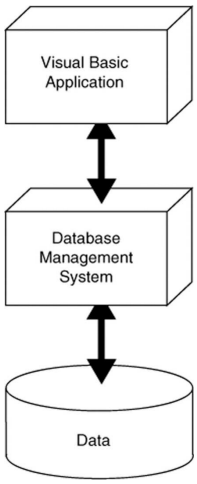
Thomas Watson, Jr. Former chairman of IBM

Visual Basic and Database Management Systems

- Simple text files as shown in chapter 9 are:
  - Fine for small amounts of data
  - But impractical for large amounts of data
- Businesses must maintain huge amounts of data
  - A *database management system (DBMS)* is the typical solution to the data needs of business
  - Designed to store, retrieve, & manipulate data
- Visual Basic can communicate with several DBMS
  - Tells DBMS what data to retrieve or manipulate

Layered Approach to Using a DBMS

- Applications that work with a DBMS use a layered approach
  - VB application is topmost layer
  - VB sends instructions to next layer, the DBMS
  - DBMS works directly with data
- Programmer need not understand the physical structure of the data
  - Just need to know how to interact with the database



Why not just use the file system?

Day8-21.txt

```
8,drive to work
9,teach class
10, …
```

Day8-22.txt

```
8,drive to work
9,eat donut
10, …
```

…

Could write programs to operate on this text file data.

File Storage Problems

- Sharing data
- Same data may be duplicated many times
- Need to write custom programs to manipulate the data (e.g search, print)
- As file systems become more complex, managing files gets more difficult
- Making changes in existing file structures is important and difficult.
- Security, data integrity (redundancy, inconsistency, anomalies) features are difficult to implement and are lacking.

File Storage Problems - Dependence

- *Structural Dependence*: A change in the file's structure requires the modification of all programs using that file.
- *Data Dependence*: A change in any file's data characteristics requires changes of all data access programs.

Solution: DBMS

- Logically related data are stored in a single data repository.
- The database represents a change in the way end user data are stored, accessed, and managed efficiently.
- Easier to eliminate most of the file system's data inconsistency, data anomalies, and data structural dependency problems.
- Store data structures and relationships
- DBMS takes care of defining all the required access paths.

Disadvantages of DBMS

- Cost of software and implementation
- Higher cost of processing routine batches
- Increase magnitude of potential disaster
- Lack of database technical capability

Relational Database Model

- Introduced in the 60's and 70's and is the most common type of DBMS today
- Data elements stored in simple tables (related)
- General structure good for many problems
- Easy to understand, modify, maintain

Examples: MySQL, Access, Oracle, SQL Server

- We will focus on relational databases using Microsoft Access in our course

The Relational Model

- Views entities as two-dimensional tables
  - Records are rows
  - Attributes (fields) are columns
- Tables can be linked
- Supports one-to-many, many-to-many, and one-to-one relationships

Terminology

- *Database*: a collection of interrelated tables
- *Table*: a logical grouping of related data
  - A category of people, places, or things
  - For example, employees or departments
  - Organized into rows and columns
- *Field*: an individual piece of data pertaining to an item, an employee name for instance
- *Record*: the complete data about a single item such as all information about an employee
  - A record is a row of a table

Database Table

- Each table has a *primary key*
  - Uniquely identifies that row of the table
  - Emp_Id is the primary key in this example
  - Serves as an index to quickly retrieve the record
- Columns are also called *fields* or *attributes*
- Each column has a particular data type

Emp_Id	First_Name	Last_Name	Department
001234	Ignacio	Fleta	Accounting
002000	Christian	Martin	Computer Support
002122	Orville	Gibson	Human Resources
003400	Ben	Smith	Accounting
003780	Allison	Chong	Computer Support

Row (Record)

Column

Field

VB and SQL Server Data Types

- VB data types must match table data types
- SQL Server and VB have similar data types

SQL Type	Usage	Visual Basic Type
Bit	True/false values	Boolean
DateTime	Dates and times	Date, DateTime
Decimal, Money	Financial values	Decimal
Float	Real-number values	Double
Int	Integer values	Integer
Smallint	Integers -32,768 to 32,767	Short
Varchar(n)	Variable length strings	String
Text	Strings more than 8000 char	String

Choosing Column Names

- Define a column for each piece of data
- Allow plenty of space for text fields
- Avoid using spaces in column names
- For the members of an organization:

Column Name	Type	Remarks
Member_ID	int	Primary key
First_Name	varchar(40)	
Last_Name	varchar(40)	
Phone	varchar(30)	
Email	varchar(50)	
Date_Joined	smalldatetime	Date only, no time values
Meeings_Attended	smallint	
Officer	Yes/No	True/False values

Issues with Redundant Data

- Database design minimizes redundant data
- In the following employee table:

ID	First_Name	Last_Name	Department
001234	Ignacio	Fleta	Accounting
002000	Christian	Martin	Computer Support
002122	Orville	Gibson	Human Resources
00300	Jose	Ramirez	Research & Devel
003400	Ben	Smith	Accounting
003780	Allison	Chong	Computer Support

- Same dept name appears multiple times
  - Requires additional storage space
  - Causes problems if misspelled
  - What if a department needs to be renamed?

Eliminating Redundant Data

- Create a department table

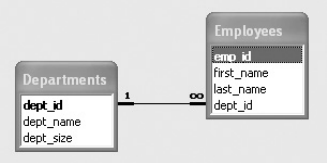
Dept_ID	Dept_Name	Num_Employees
1	Human Resources	10
2	Accounting	5
3	Computer Support	30
4	Research & Development	15

- Reference department table in employee table

ID	First_Name	Last_Name	Dept_ID
001234	Ignacio	Fleta	2
002000	Christian	Martin	3
002122	Orville	Gibson	1
003000	Jose	Ramirez	4
003400	Ben	Smith	2
003780	Allison	Chong	3

One-to-Many Relationships

- The previous changes created a *one-to-many relationship*
  - Every employee has one and only one dept
  - Every department has many employees
  - DeptID in department table is a *primary key*
  - DeptID in employee table is a *foreign key*
- One-to-many relationship exists when primary key of one table is specified as a field of another table



Normalization

- The previous example illustrated a technique used to make complex databases more efficient called Normalization
- Break one large table into several smaller tables
  - Eliminates all repeating groups in records
  - Eliminates redundant data
- Another example…

Redundant Data

Student ID#	Student Name	Campus Address	Major	Phone	Course ID	Course Title	Instructor Name	Instructor Location	Instructor Phone	Term	Grade
A121	Joy Egbert	100 N. State Street	MIS	555-7771	MIS 350	Intro. MIS	Van Deventer	T240C	555-2222	F'98	A
A121	Joy Egbert	100 N. State Street	MIS	555-7771	MIS 372	Database	Hann	T240F	555-2224	F'98	B
A121	Joy Egbert	100 N. State Street	MIS	555-7771	MIS 375	Elec. Comm.	Chatterjee	T240D	555-2228	F'98	B +
A121	Joy Egbert	100 N. State Street	MIS	555-7771	MIS 448	Strategic MIS	Chatterjee	T240D	555-2228	F'98	A –
A121	Joy Egbert	100 N. State Street	MIS	555-7771	MIS 474	Telecomm	Gilson	T240E	555-2226	F'98	C +
A123	Larry Mueller	123 S. State Street	MIS	555-1235	MIS 350	Intro. MIS	Van Deventer	T240C	555-2222	F'98	A
A123	Larry Mueller	123 S. State Street	MIS	555-1235	MIS 372	Database	Hann	T240F	555-2224	F'98	B –
A123	Larry Mueller	123 S. State Street	MIS	555-1235	MIS 375	Elec. Comm.	Chatterjee	T240D	555-2228	F'98	A –
A123	Larry Mueller	123 S. State Street	MIS	555-1235	MIS 448	Strategic MIS	Chatterjee	T240D	555-2228	F'98	C +
A124	Mike Guon	125 S. Elm	MGT	555-2214	MIS 350	Intro. MIS	Van Deventer	T240C	555-2222	F'98	A –
A124	Mike Guon	125 S. Elm	MGT	555-2214	MIS 372	Database	Hann	T240F	555-2224	F'98	A –
A124	Mike Guon	125 S. Elm	MGT	555-2214	MIS 375	Elec. Comm.	Chatterjee	T240D	555-2228	F'98	B +
A124	Mike Guon	125 S. Elm	MGT	555-2214	MIS 474	Telecomm	Gilson	T240E	555-2226	F'98	B
A126	Jackie Judson	224 S. Sixth Street	MKT	555-1245	MIS 350	Intro. MIS	Van Deventer	T240C	555-2222	F'98	A
A126	Jackie Judson	224 S. Sixth Street	MKT	555-1245	MIS 372	Database	Hann	T240F	555-2224	F'98	B +
A126	Jackie Judson	224 S. Sixth Street	MKT	555-1245	MIS 375	Elec. Comm.	Chatterjee	T240D	555-2228	F'98	B +
A126	Jackie Judson	224 S. Sixth Street	MKT	555-1245	MIS 474	Telecomm	Gilson	T240E	555-2226	F'98	A –
...	...	...	...	...	...	...	...	...	...	...	...

Normalized Data

Student Table

Student ID#	Student Name	Campus Address	Major	Phone
A121	Joy Egbert	100 N. State Street	MIS	555-7771
A123	Larry Mueller	123 S. State Street	MIS	555-1235
A124	Mike Guon	125 S. Elm	MGT	555-2214
A126	Jackie Judson	224 S. Sixth Street	MKT	555-1245
...	...	...	...	...

Class Table

Course ID	Course Title
MIS 350	Intro. MIS
MIS 372	Database
MIS 375	Elec. Comm.
MIS 448	Strategic MIS
MIS 474	Telecomm
...	...

Teaching Assignment

Course ID	Term	Instructor Name
MIS 350	F'98	Van Deventer
MIS 372	F'98	Hann
MIS 375	F'98	Chatterjee
MIS 448	F'98	Chatterjee
MIS 474	F'98	Gilson
...	...	...

Instructor Table

Instructor Name	Instructor Location	Instructor Phone
Chatterjee	T240D	555-2228
Gilson	T240E	555-2226
Hann	T240F	555-2224
Valacich	T240D	555-2223
Van Deventer	T240C	555-2222

Enrolled Table

Student ID#	Course ID	Term	Grade
A121	MIS 350	F'98	A
A121	MIS 372	F'98	B
A121	MIS 375	F'98	B +
A121	MIS 448	F'98	A –
A121	MIS 474	F'98	C +
A123	MIS 350	F'98	A
A123	MIS 372	F'98	B –
A123	MIS 375	F'98	A –
A123	MIS 448	F'98	C +
A124	MIS 350	F'98	A –
A124	MIS 372	F'98	A –
A124	MIS 375	F'98	B +
A124	MIS 474	F'98	B
A126	MIS 350	F'98	A
A126	MIS 372	F'98	B +
A126	MIS 375	F'98	B +
A126	MIS 474	F'98	A –
...	...	...	...

Associations

- Relationships among the entities in the data structures
- Three types
  - One-to-one
  - One-to-many
  - Many-to-many
- Relationships set by placing primary key from one table as foreign key in another
  - Creates “acceptable” redundancy

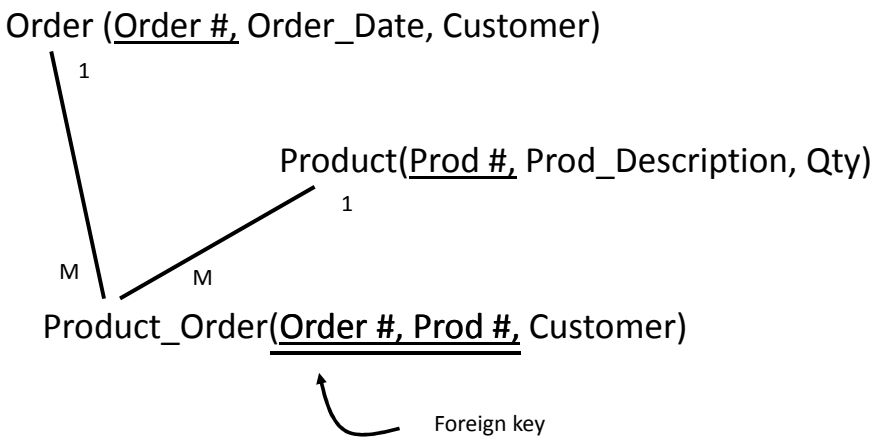
Association Examples

One-to-one (1:1)

One-to-many (1:M)

Many-to-many (N:M)

Associations



Microsoft Access is Unique

- Provides DBMS functions
  - Not "industrial-strength", designed for:
    - Individuals
    - Small workgroups
  - External application programs work with Access
    - We will use VB.NET and the Structured Query Language (SQL) later
- Provides built-in tools for reporting and for application development
  - Forms
  - Reports
  - Code modules using Visual Basic for Applications (VBA) which is similar to VB.NET but not the same
- Provides flexibility
  - Small, simple all-in-one environment
  - Data can be easily transferred to full-fledged DBMS

Introduction to Access

- Sample databases
  - Northwind
    - Included with every version of Access since 2.0
- Demonstration of Access
  - Startup
  - Create tables
  - Link table relationships
  - Create queries/reports
  - In this class you won't have to create your own databases, but I'll give you a database to work with

Access 2007 Example

Student ID	Last Name	First Name	DOB	Address
1	Mock	Kenrick	4-18-1968	123 Somewhere Ave
2	Cue	Barbie	3-21-1970	567 A Street
3	Obama	Barack	8-04-1961	123 Somewhere Ave

Access 2007 Example

CS 111 Table

Student ID	Grade
1	A
2	B
3	B

CS 201 Table

Student ID	Grade
1	B
2	A
3	C