Content

- Related to **intellectual property**
  - form of creative endeavour that can be protected through a trademark, patent, copyright, industrial design, or integrated circuit topography
- Varies by industry

Content

- Organizations may store
  - Data
  - Documents
  - Spreadsheets
  - Presentations
  - web pages
  - text from blogs and discussion boards
  - graphics,
  - video files
  - audio files …

Organizing Content

- Content management challenge
  - processing and storing the right content
  - getting the right content to the right person in the right format at the right time

Content Management

- Management of content data
  - database management systems (DBMS)
    - Effectively & Efficiently storing & processing data
- Presentation of content
  - content management system (CMS)
    - Organizing documents
    - Seek out documents & organize access

Spreadsheet vs Database

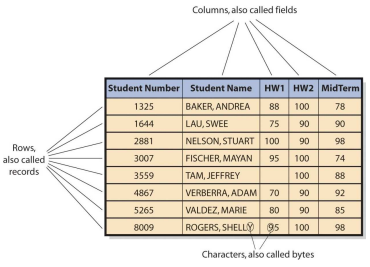
- Keeping track of things
  - Spreadsheets
    - Keep lists of single concept
  - Databases
    - Keep lists that involve multiple themes

Relational Database

- Conceptually simple
- Easy to understand
- Relationships aren't predefined
- Database can evolve as required
- Relationships are implied in the data

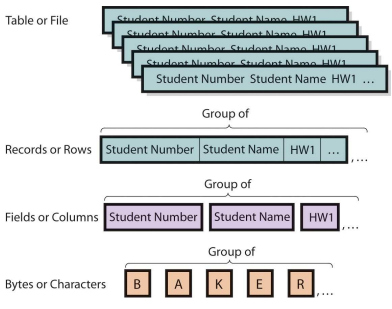
A Database

- Self-describing collection of integrated records



Student Number	Student Name	HW1	HW2	MidTerm
1325	BAKER, ANDREA	88	100	78
1644	LAU, SWEE	75	90	90
2881	NELSON, STUART	100	90	98
3007	FISCHER, MAYAN	95	100	74
3559	TAM, JEFFREY		100	88
4867	VERBERRA, ADAM	70	90	92
5265	VALDEZ, MARIE	80	90	85
8009	ROGERS, SHELLY	95	100	98

Hierarchy of data elements



Components of a Database

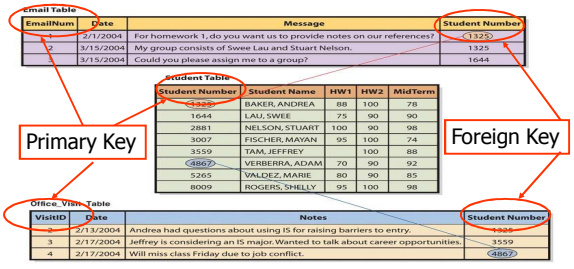
Tables or Files
+
Relationships among Rows in Tables
+
Metadata
=
Database

description of the structure of database

Relationships Among Records

- Primary Keys
  - Column(s) that identify unique row in table
  - Each table has a key
- Foreign keys
  - Primary Keys from a different table
- Relational database
  - Databases using tables, primary keys & foreign keys

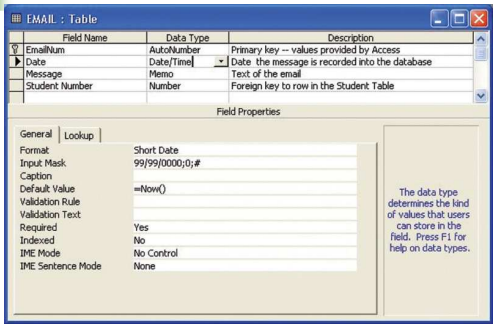
Relationships Among Records



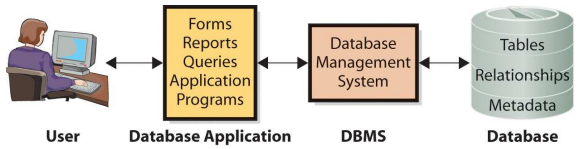
Metadata

- Databases are self-describing
  - Contain description of its content
- Metadata
  - Data that describes data
  - Makes databases more useful
  - Makes databases easier to use

Metadata (Access)



Components of a Database Application System



Database Management System (DBMS)

- Program
  - Creates database
  - Processes database
  - Administers database
- Usually licensed from vendors
  - Microsoft: Access, SQL server
  - Oracle: Oracle
  - IBM: DB2
  - Open Source: MySQL

They are Different!

- Database
  - Collection of tables, relationships & metadata
- DBMS
  - Software program

DBMS - Function

- Creation
  - Tables
  - Relationships in databases

DBMS - Function

- Process database
- Applications use DBMS
  - Read data
  - Insert data
  - Modify data
  - Delete data

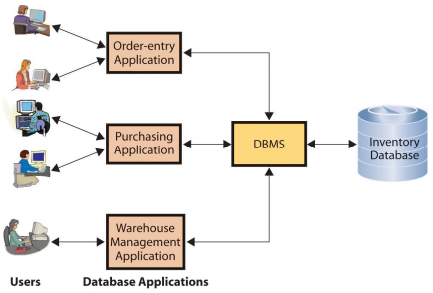
DBMS - Function

- Administration
  - Security (levels)
    - Access to the database
  - Backup
  - Improve performance
  - Removal of data no longer needed

Database Application

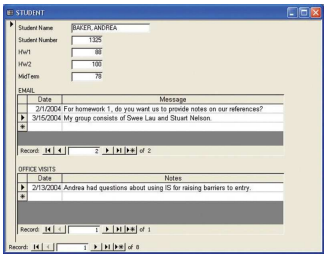
- Collection
  - Forms
  - Reports
  - Queries
  - Application programs that process a database

Database Applications



Forms

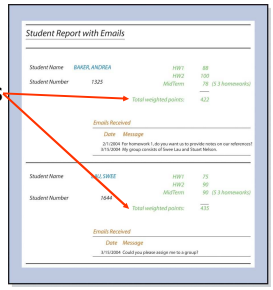
- Read data
- Insert data
- Modify data
- Delete data



Example: Data entry form

Reports

- Shows data in structured context
- Able to compute values

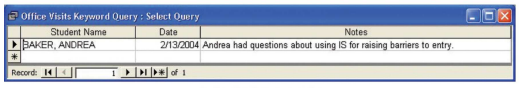


Example: Report

Queries

- Comprehensive & robust method to query data

a. Form used to enter phrase for search



b. Results of query operation

Queries

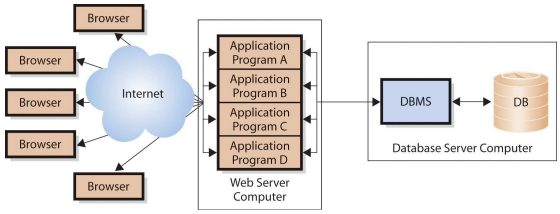
- Structured Query Language (SQL)
  - International standard for processing database

```
INSERT INTO Student
    ([Student Number], [Student Name], HW1, HW2, MidTerm)
    Values
    (10000, "Franklin Benjamin", 90, 95, 100)
```

Database Application Programs

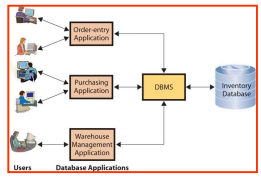
- Forms, reports & queries
  - Standard functions
- Application programs
  - Process logic specific to business need
  - Enables database processing over Internet
    - Intermediary between Web server & database
    - Responds to events (Reads, inserts, modifies, deletes data)

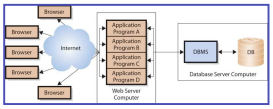
Application Programs on a Web Server



Multi-User Processing

- Problems
  - Lost-update problem
  - Locking used to coordinate activities of multiple users





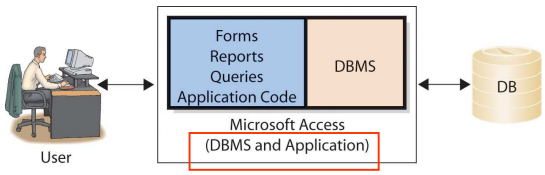
Enterprise DBMS

- Process organizational & workgroup databases
- Large Databases
- Support many, many users
- Examples: DB2, SQL Server, Oracle

Personal DBMS

- Designed for smaller, simpler database applications
- Supports fewer than 100 users
- Examples: Access, dBase, FoxPro, Paradox, R:Base

Personal Database System



Relational Database Design

- Designer creates table for every entity
- Key Attribute is primary key of table
- Attributes of entity become columns
- Tables normalized to single theme
- Represent relationships between tables
  - Add foreign key to one or more tables

Normalization

- Process
  - Converts table into two or more tables
  - Changes from poorly structured to well-structured
- Data integrity problems
  - Different names for the same entity
  - Produces incorrect and inconsistent information
  - Resolve by eliminating duplicated data
- Normalized tables
  - Eliminate data duplication
  - Slower to process
  - Every table has single topic

Poorly Designed Tables

Employee

Name	HireDate	Email	DeptNo	DeptName
Jones	Feb 1, 2002	Jones@ourcompany.com	100	Accounting
Smith	Dec 3, 2004	Smith@ourcompany.com	200	Marketing
Chau	March 7, 2004	Chau@ourcompany.com	100	Accounting
Greene	July 17, 2003	Greene@ourcompany.com	100	Accounting

a. Table Before Update

Employee

Name	HireDate	Email	DeptNo	DeptName
Jones	Feb 1, 2002	Jones@ourcompany.com	100	Accounting and Finance
Smith	Dec 3, 2004	Smith@ourcompany.com	200	Marketing
Chau	March 7, 2004	Chau@ourcompany.com	100	Accounting and Finance
Greene	July 17, 2003	Greene@ourcompany.com	100	Accounting

b. Table with Incomplete Update

Two Normalized Tables

Employee

Name	HireDate	Email	DeptNo
Jones	Feb 1, 2002	Jones@ourcompany.com	100
Smith	Dec 3, 2004	Smith@ourcompany.com	200
Chau	March 7, 2004	Chau@ourcompany.com	100
Greene	July 17, 2003	Greene@ourcompany.com	100

Department

DeptNo	DeptName
100	Accounting
200	Marketing
300	Information Systems

Database Elements

- Storing data:
  - An **entity** is anything about which the organization wishes to store as data.
  - Example: At Westen, one entity would be the student.

STUDENTS				
Student ID	**Last Name**	**First Name**	**Phone Number**	**Birth Date**
333-33-3333	Simpson	Alice	333-3333	10/11/84
111-11-1111	Sanders	Ned	444-4444	11/24/86
123-45-6789	Moore	Artie	555-5555	04/20/85

Database Elements

- Information about **attributes** of an entity stored in **fields**
- Example: (e.g., the student number & birth date)

STUDENTS				
Student ID	**Last Name**	**First Name**	**Phone Number**	**Birth Date**
333-33-3333	Simpson	Alice	333-3333	10/11/84
111-11-1111	Sanders	Ned	444-4444	11/24/86
123-45-6789	Moore	Artie	555-5555	04/20/85

Database Elements

- All the fields containing data about one entity form a **record**
- Example: one student

STUDENTS				
Student ID	**Last Name**	**First Name**	**Phone Number**	**Birth Date**
333-33-3333	Simpson	Alice	333-3333	10/11/84
111-11-1111	Sanders	Ned	444-4444	11/24/86
123-45-6789	Moore	Artie	555-5555	04/20/85

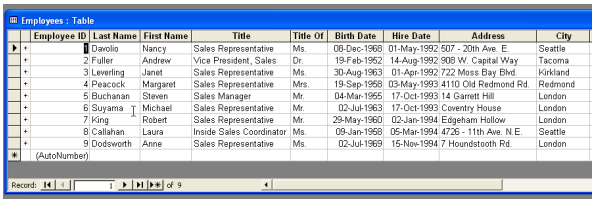
Database Elements

- A set of all related records forms a **file** (e.g., student file)
- Example: If Western only had three students and five fields for each student, then the entire file would be depicted below.

STUDENTS				
Student ID	**Last Name**	**First Name**	**Phone Number**	**Birth Date**
333-33-3333	Simpson	Alice	333-3333	10/11/84
111-11-1111	Sanders	Ned	444-4444	11/24/86
123-45-6789	Moore	Artie	555-5555	04/20/85

Database elements

- **Tables** in a relational database consist of rows & columns
  - Rows represent the instances of the entity
  - Columns represent the attributes
  - Every table **should** have a key attribute (primary key).



Relational Database

- Data model based on a relation
  - represented as a 2-D table
- Entities & Relationships in an E-R Model become tables in the Relational Model
- Rows in a table constitute the instances of the entity or relationship
- Attributes of the entities and relationships in an E-R Model are the columns in the tables
- Table that stores a relationship inherits the key attributes from all of the entities involved in the relationship

STUDENT Entity

- Assume that the attributes of the STUDENT entity are
  - Student Number (ST-NUM)
  - Name (NAME)
  - Address (ADDRESS)

STUDENT Entity Table

ST-NUM	NAME	ADDRESS
250078563	Rhonda Odanski	65-927 Richmond St.
250004423	Peter Chen	1848 Downes Crt.
250016788	Susanne Ferber	39 Danielle Cr.
250012745	Rick Mattatall	87 Dillabough St.

COURSE Entity

- Assume that the attributes of the COURSE entity are
  - Internal Course Identification number (COURSE-ID)
  - Course name (NAME)
  - Course number (NUMBER)
  - Term offered (TERM)

COURSE Entity Table

COURSE-ID	NAME	NUMBER	TERM
6701901	COMP SCI	1033	A
6701402	COMP SCI	1033	B
6701203	COMP SCI	1032	A
6701604	COMP SCI	1032	B

TAKES Relationship

- The TAKES relationship exists between the STUDENT and COURSE entities (STUDENT takes COURSE)
- The key from the STUDENT relation (ST-NUM) and the key from the COURSE relation (COURSE-ID) are the key to the TAKES relation

TAKES Relationship Table

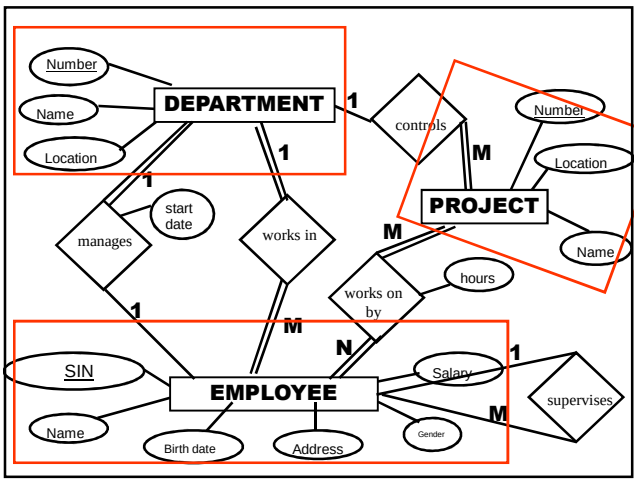
ST-NUM	COURSE-ID
250016788	6701402
250004423	6701901
250012745	6701901
250004423	6701604
250078563	6701402
250016788	6701203
250012745	6701604

TAKES Relationship

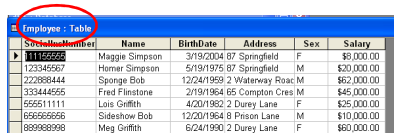
- The TAKES relationship captures the course registration of the students
- From the example
  - Rhonda is taking COMP SCI 1033B
  - Peter is taking COMP SCI 1033A & COMP SCI 1032B
  - Susanne is taking Comp SCI 1032A & COMP SCI 1033B
  - Rick is taking Comp SCI 1033A and COMP SCI 1032B

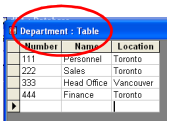
Converting an E-R Diagram into a Relational Database

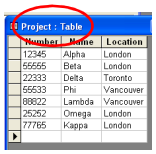
- **Step 1**:
  - For each entity in the diagram:
    - Create a table with the same name as the entity.
    - Each attribute becomes a column.
    - Key attribute of the entity becomes the primary key of the table



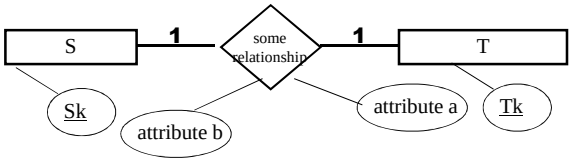
Step 1, Example



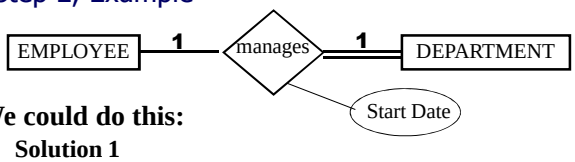




- **Step 2**:
  - For each 1:1 relationship in your model:
    - Identity the two participating tables (entities), say S and T and their respective key attributes, say Sk and Tk.
    - **Choose one**, say S, & make Sk the *foreign key* in table T.
    - Table T now has a new column, which holds Sk values.
    - If there are any attributes on the relationship, move them into table T as new columns.

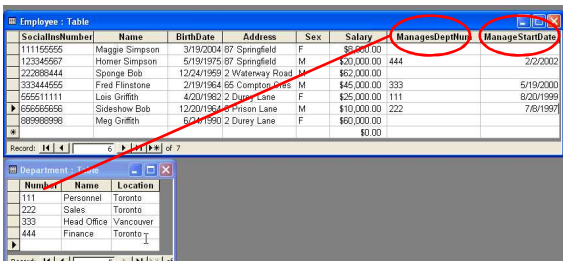


Step 2, Example



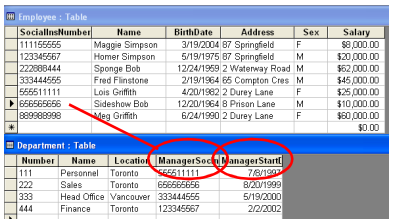
We could do this:

Solution 1



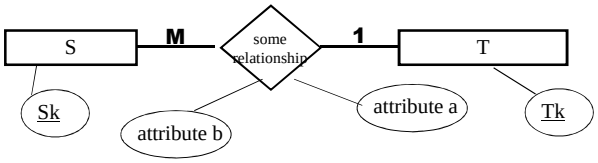
OR we could do this:

Solution 2

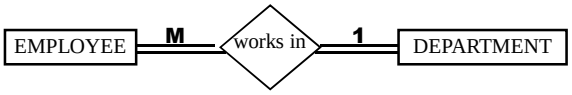


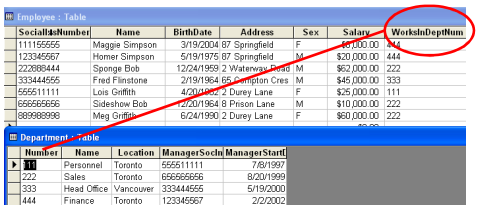
Which solution is better and why?

- **Step 3:**
  - For each 1:M relationship in your model:
    - Identity the two participating tables (entities), say S and T and their respective key attributes, say Sk and Tk.
    - Assume that S is the table on the M (*many*) side and that T is the table on the 1 (*one*) side.
    - Take the key from the *one* side, Tk and make it be a foreign key in the *many* (S) table.
    - Table S has a new column which hold Tk values.
    - If there are any attributes on the relationship, they would also become new columns in table S.

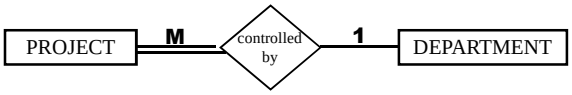


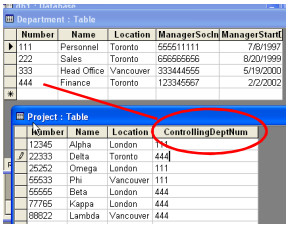
Step 3, Example



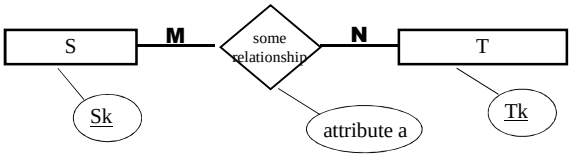


Also have 1:M relationships:

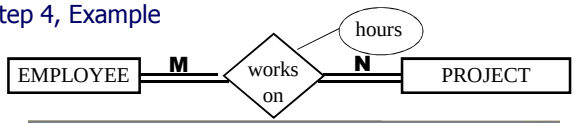


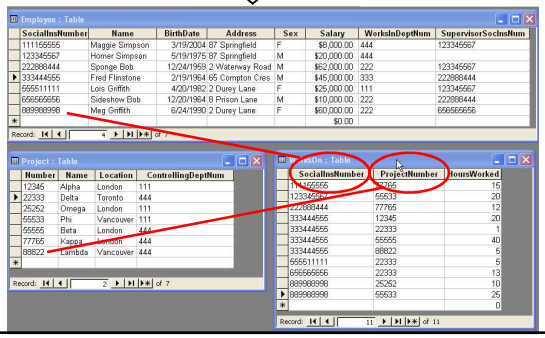


- **Step 4:**
  - For each M:N relationship in your model:
    - Identity the two participating tables (entities), say S and T and their respective key attributes, say Sk and Tk.
    - Create a new table, called R.
    - R's attributes(columns) should be Sk, Tk and any other attributes on the relationship.
    - The primary key of R will be the combination of Sk and Tk.



Step 4, Example



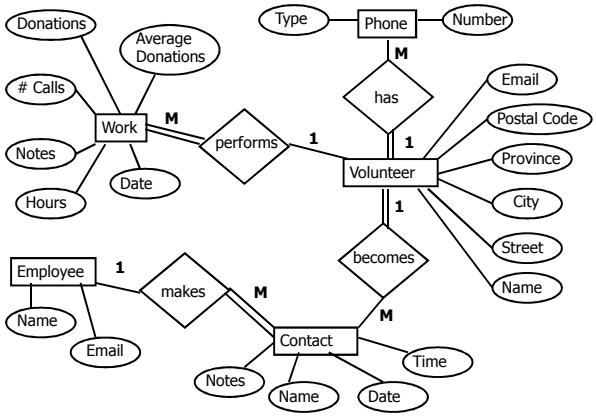


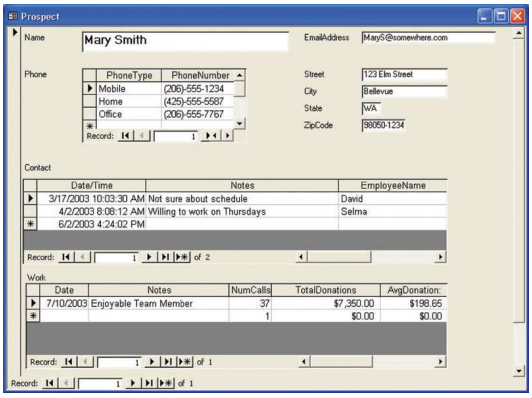
Your Role…

- Decide on
  - What data should be contained
  - How records are related to each other
- Review & Correct data model
  - Ensure it is an accurate view

Who Will Volunteer?

- Consultant creates data model
  - Based on interviews with users
- Data model reviewed and approved
- Database tables constructed
  - Primary and foreign keys selected
    - Based on interviews
- Microsoft Access database created
  - Relationships indicated
  - Forms and reports constructed





Things To Consider:

- No order or sequence in the table
  - Order of the rows does not matter
  - Order of the columns does not matter
- Primary keys and foreign keys, show the relationships between the instances
- Relational databases are conceptually simple and easy to understand
  - Simply need to understand tables, rows & columns
- Steps 1 to 4 work for mapping to any DBMS