

*David M. Keil*  
*Framingham State University*

## 3. Introduction to database management

1. Entities and tables
2. Searching, sorting, summarizing data
3. Database design
4. DB management in business environments

*Reading: Evans et al, Ch. 11; Handouts*

David Keil Introduction to Information Technology 1/12 1

## Inquiry

- What is worth knowing about databases, for non-geeks?
- What is the principle for organizing data?

David Keil Introduction to Information Technology 1/12 2

## Objectives

- 3a. Recognize or use simple database concepts
- 3b. Design a relational database using entity-relationship concepts
- 3c. Design simple database queries
- 3d. Group and summarize data to show patterns

## Database management

- Supports retrievable arrangement of data
- *Examples:* library catalog, student records, many business records
- Want to separate the database management software from the data design, enabling data design on site
- Likewise *query design* is possible on site
- Simple DB management can be done with MS Word tables, Excel spreadsheets
- Effective database and query design requires *analytical thinking*: breaking a problem into parts

## Motivation

- Consistency and non-redundancy of data require more than a list format
- Certain information is easiest to manage in *tables*
- End users retrieve information from these tables using queries
- Today DB experts and users also *design* tables
- Use of general-purpose *database management systems* (DBMSs) is growing

## 1. Entities and tables

- An *entity* (aka *class*) is a category of *instances* (*objects*) – things, people, places, transactions
- *Table*: stores data about instances of entities
- An entity has *attributes* (table columns) and *relationships*
- One instance of an entity is called a *tuple*, *record*, or *row*
- Example: A *student* has an *ID*, a *name*, and a *Year-of-graduation*

## Tables

- Store information for lookup
- Database tables include *metadata* (headers), may be empty
- Entities have *attributes* (fields), which have names, types, values
- *Example*: student table (*name, ID, email, major*)
- Row representing instance of entity is a *tuple* (record)
- *Sets* of tuples (relations) are unordered, by definition
- *Primary key* is unique identifier of a tuple

## DB concepts

<b>DB term</b>	<b>Math term</b>
Table, class, entity	relation
Record, row, object	tuple
Column, attribute	

- A relation is a *set* of tuples

## Database views

- *View*: A logical (vs. physical) table, created by an operation on a table
- Some views are a *window* into a table
- *Selection*: specification of a set of rows chosen by some criterion (e.g., all contacts with salary over \$55,000)
- *Projection*: specification of a set of columns (e.g., name and salary)

## Other operations on tables

- *Union*: Combines two compatible tables (*Example*: all students in either of two sections of a course)
- *Difference* (*Example*: Students who are not CS majors)
- *Product*: Creates combined table (*Example*: students  $\times$  courses), used to create smaller *join* relation
- *Join*: A *product*, restricted to matching values for common fields

### 3. Searching, sorting, summarizing

- *Query* (filter): generate selected view
- *Subtotal*: sum of numeric values in a field of a view
- *Sorting*: any table may be sorted on any attribute
- *Pivot tables*: show data correlations between independent and dependent variables

### **Example of table: List of professional contacts**

- Example file: *contacts.xlsx*
- Schema: (*Name, ID, Salary, Title, Company*)
- Search (*Find*)
- Sort on any field (*Data / Sort*), ascending or descending

## Queries

- *Query*: specification of a view
- Query languages
  - Query by example (enter value in a field)
  - Structured Query Language (SQL)
- *Examples*:
  - Select all student records where major is Business Administration
  - Select all employees where salary > 25000 and salary < 50000
  - Select name, salary from employees

## Querying (searching) in Excel

- In Excel, *filters* enable selection of rows of a table according to values of attributes
- In Excel, select column headings. Then *Data / Filter* places tabs at tops of columns
- *Example*: with *listings.xls*, to select all home listings with more than 2 bedrooms:
  - Click triangle beside “BDRM”
  - Choose “Custom”
  - Choose “Is greater than” and “2”
- Multiple conditions may be specified using “AND” and “OR” operators
- Can find duplicates using a filter

## Sorting in Excel and MS Word

- Columns of Excel spreadsheets and of tables, or lists of paragraphs, in Word documents may be sorted
- Sorts may be ascending or descending
- Word: use “A→Z” button
- Excel: use *Data* tab

## Subtotals and pivot tables

- In Excel, *Data / Subtotals* permits automatic insertion of subtotal information, e.g., total salary for EMC employees
- Use of subtotals and pivot tables enables higher-level data analysis than use of cell data alone
- A *pivot table* in Excel enables application of functions, e.g., *sum*, to all of a given field of all records that match by two criteria
- Summary data is also enabled, e.g., total salaries for all TJX vice presidents

## Correlations of variables

- A *pivot table* correlates the values of independent and dependent variables
- The columns used at the edges of the pivot table are the independent variables
- The column used in the body of the table is the dependent variable
- The pivot table shows how the dependent variable is affected by the values of the independent variables

## 3. Database design

- Integrity principle: no two records may represent the same real-world object
- This principle is enforced using *primary keys* and using a method called *entity-relationship design*
- A more general integrity principle is that any fact may be represented only once in a database

## Entity-relationship design

- The database design process may include consideration of *relationships* among *entities* stored as tables
- *Example:* Where *Student* and *Course* are entities, the real-world relationship “*Student registers for Course*” may guide design
- In that case, a course-registration table may be designed to store records whose attributes might include *Student ID*, *Course ID*, and *Date*

## Entity example: Course registration

- A student has an *ID*, a *name*, and a *Year-of-graduation*
- Primary key is *ID*, created to be unique
- A course section has a *course-ID*, *section-ID*, *name*, and *instructor*
- Entity-relationship diagram:



- A *Registrations* table could embody the relationship “registers for” between the *Students* and *Course-sections* entities

## Joining tables

- *Join*: A view that combines data from two or more tables where records have coinciding field values (e.g. course number)
- *Example*:
  - A *course* table lists course numbers, course titles
  - A *course-registration* table lists student IDs and course numbers of the courses the student registered for
  - A transcript will access both tables

## Join views

- Suppose a business maintains tables of *customers*, *products for sale*, and *transaction details* (instances of orders of a single item)
- Each transaction detail record includes customer ID and transaction detail identification, e.g., date, item #, qty., price
- To show all data about a single customer order, a *join* is performed, yielding the customer information plus all detail data for that transaction (i.e., a bill of sale)

## Information integrity in DBs

- *Rule:* A *fact* should appear only *once* in a database
- *Example:* Non-duplication assured by use of **primary key** prevents anomalies such as deleting only one record and leaving others for the same supposedly deleted entity
- *Example:* A student's address should appear with the student's record, not in every record of a student registering for a course. What if address changes?

## Enforcing integrity

### *Guidelines:*

- Each record has an attribute known to be unique
- Dependencies are reflected in design.  
*Example:* If Student name in a course registration depends on student ID, the course registration record contains the course ID but not the student name
- Many-to-one and many-to-many relationships are implemented using *key* references to participants in relationship

#### 4. Database management in business environments

- Excel works for simple tables
- A *database management system*, such as Access or Oracle, is specialized for database operations
- Web-based database access uses *middleware* to make database info available to the server in responding to client requests

#### Other database tools

- *Web pages* may access and even update database files for customer service
- *Forms* enable easy entry of data into a relation
- *Reports* format DB information for distribution
- *Graphical design tools* enable study of linked relationships among tables
- Integrity rules assure that a database remains internally consistent; e.g., no fact should appear more than once in a database

## Database file formats

- Specialized (e.g., email address books)
- Tables in word-processor or spreadsheet documents
- General-purpose DBMS software (Access, Oracle, Ingres)
- XML (Extended Markup Language), which uses HTML-like tags to denote field names, as *semantic* information
- Data for one DB may be stored at distributed locations

## Data analysis

- *Decision support systems* attempt to predict future sales, profits, costs, based on current data and alternative scenarios
- *Data mining* uses statistical methods to find unseen relationships in data
- *Data warehouses* collect data from multiple sources for analysis
- *Data analysis* is used to profile individuals for market and other purposes

## Data warehouses

- *Data warehouse*: large-scale archival storage of all or a business's data
- *Sources*: internal, external, sources, customers, web visitors
- *Scope*: enterprise
- *Data marts* organize access to data by department

## ISs for management of business

All use databases. *Categories*:

- *Office support systems*: day-to-day task and communication support; e.g., MS Office
- *Transaction processing systems*: e.g., course registrations, purchases
- *Management information systems*: detail, summary, and exception reports for managers
- *Decision support systems*: work with models and knowledge bases to provide basis for decisions
- Enterprise resource planning systems tie together multiple data sources and business processes

## Data mining

- Gathers and analyzes very large quantities of data
- Chief goal: to analyze customer behavior and attributes
- Analysis and gathering methods:
  - Classification (e.g., good/bad risk)
  - Estimation (e.g., residence patterns of low-risk persons)
  - Association rules (e.g., products bought together)
  - Clustering, description, visualization

## Database concepts

attribute	query
column	query by example
data design	query language
database	record
database manager	redundancy
entity	relation
entity-relationship modeling	relationship
fields	row
filter	<i>select</i> operation
index	SQL
key	table
logical assertion	tuple
metadata	unique identifier
pivot table	view
primary key	

## References

C. Breuning. Handout on using database features of Excel.

A. Evans, K. Martin, M. A. Poatsy. *Go! Technology in Action*, 6<sup>th</sup> ed. Prentice Hall, 2010.

J. Parsons, D. Oja. *Computer Concepts*, 9<sup>th</sup> ed. Thomson, 2007.