COMP 430
Intro. to Database Systems
Course overview
Databases central to modern life
Understanding the context of the course topic
Computing in the 1950’s

An exciting time:

• John Backus – FORTRAN, 1957
• John McCarthy – LISP, invented 1958, implemented 1962

But no one cared about data management.
CODASYL (1959)

First **CO**nference on **DA**ta **S**ystems and **L**anguages
IDENTIFICATION DIVISION.

PROGRAM-ID. SAMPLE.

AUTHOR. J.P.E. HODGSON.

DATE-WRITTEN. 4 February 2000

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE CONTROL.

SELECT STUDENT-FILE ASSIGN TO SYSIN
ORGANIZATION IS LINE SEQUENTIAL.

SELECT PRINT-FILE ASSIGN TO SYSOUT
ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.

FILE SECTION.

FD STUDENT-FILE

RECORD CONTAINS 43 CHARACTERS
DATA RECORD IS STUDENT-IN.

01 STUDENT-IN PIC X(43).

FD PRINT-FILE

RECORD CONTAINS 80 CHARACTERS
DATA RECORD IS PRINT-LINE.

01 PRINT-LINE PIC X(80).

WORKING-STORAGE SECTION.

01 DATA-REMAINS-SWITCH PIC X(2) VALUE SPACES.

01 RECORDS-WRITTEN PIC 99.

01 DETAIL-LINE.

05 FILLER PIC X(7) VALUE SPACES.

05 RECORD-IMAGE PIC X(43).

05 FILLER PIC X(30) VALUE SPACES.

01 SUMMARY-LINE.

05 FILLER PIC X(7) VALUE SPACES.

05 FILLER PIC X(17) VALUE 'Records were read'.

05 FILLER PIC X(53) VALUE SPACES.

Data arranged in records.
Data described declaratively, separate from “code”.

A sample program just to show the form.
The program copies its input to the output, and counts the number of records.
At the end this number is printed.
000330 PROCEDURE DIVISION.
000331
000340 PREPARE-SENIOR-REPORT.
000350     OPEN INPUT STUDENT-FILE
000360     OUTPUT PRINT-FILE.
000361     MOVE ZERO TO RECORDS-WRITTEN.
000370     READ STUDENT-FILE
000380     AT END MOVE 'NO' TO DATA-REMAINS-SWITCH
000390     END-READ.
000400     PERFORM PROCESS-RECORDS
000410     UNTIL DATA-REMAINS-SWITCH = 'NO'.
000411     PERFORM PRINT-SUMMARY.
000420     CLOSE STUDENT-FILE
000430     PRINT-FILE.
000440     STOP RUN.
000450
000460 PROCESS-RECORDS.
000470     MOVE STUDENT-IN TO RECORD-IMAGE.
000480     MOVE DETAIL-LINE TO PRINT-LINE.
000490     WRITE PRINT-LINE.
000500     ADD 1 TO RECORDS-WRITTEN.
000510     READ STUDENT-FILE
000520     AT END MOVE 'NO' TO DATA-REMAINS-SWITCH
000530     END-READ.
000540
000550 PRINT-SUMMARY.
000560     MOVE RECORDS-WRITTEN TO TOTAL-READ.
000570     MOVE SUMMARY-LINE TO PRINT-LINE.
000571     WRITE PRINT-LINE.
000572

Code processes a record at a time. English-like syntax. Keywords capitalized.
Information Management System (1966)

Developed by IBM for Saturn V

Used first serious data model – hierarchical
Hierarchical Data Model

- Department (DName, Supervisor, Address)
- Employee (EName, PercentEffort, Age, Salary)
- Project (PName, Director, Duration, Customer)
Hierarchical Data Model

Which employees supervised by Swarat work on Pliny?

DL/1 code:

Get unique Department (Supervisor = "Swarat")
Until failure do
    Get next within parent
    Until failure do
        Get next within parent (PName = "Pliny")
            ...
    Enddo
Enddo
Hierarchical Data Model – directional

Department (DName, Supervisor, Address)

Employee (EName, PercentEffort, Age, Salary)

Project (PName, Director, Duration, Customer)

Can search from Departments to Projects, but not the other way.
Hierarchical Data Model – redundancy

Assume:
- Employees can work on multiple projects.
- Projects can have multiple employees.

Leads to redundancy – Project info repeated for each of its employees.
CODASYL Network Data Model (1969)

Department (DName, Supervisor, Address)

Affiliation (PercentEffort)

Employee (EName, Age, Salary)

Works (Duration)

Project (PName, Director, Customer)
CODASYL Network Data Model (1969)

Which employees supervised by Swarat work on Pliny?

Search $\uparrow$ good when few employees on Pliny project.

Search $\downarrow$ good when few employees supervised by Swarat.

Code locks us into an access pattern.
While the Hierarchical vs. Network Data Model raged on...
Relational Model (1970)

Edgar F Codd:


Resulted in 1981 Turing Award
Relational Model

Department (DName, Supervisor, Address)

AffiliatedWith (DName, EName, PercentEffort)

Employee (EName, Age, Salary)

WorksFor (Ename, Pname, Duration)

Project (PName, Director, Customer)
Relational Model

Department (DName, Supervisor, Address)

AffiliatedWith (DName, EName, PercentEffort)

Employee (EName, Age, Salary)

WorksFor (Ename, Pname, Duration)

Project (PName, Director, Customer)

Which employees supervised by Swarat work on Pliny?

Declarative query:
\{ e.EName | Employee(e) and 
exists (w,a,d)(WorksFor(w) and 
AffiliatedWith(a) and 
Department(d) and 
e.EName = w.EName and 
w.PName = “Pliny” and 
a.EName = e.Ename and 
a.DName = d.DName and 
d.Supervisor = “Swarat”) \}
Movement towards Relational Model

• 1969: Published internally at IBM
• 1970: Published externally
• 1974: IBM’s System R project begins
• 1979: Oracle RDBMS to market – market cap: $0B to ~$160B now
• 1980: IBM RDBMS to market
• Dominates market and mindshare

Turing Awards: 1981 (Cobb), 2014 (Stonebraker)
Object-Oriented DBs (1980s)

Persistent objects

Didn’t catch on:
• 1980’s-style objects fit well into relational model
• Lost advantages of declarative queries

Idea has returned with more complex modern objects
XML DBs

HTML-like data tags
XML popular for data interchange

• Semi-structured data model – more flexible than relational
• Has declarative query language (XQuery)

• Similar to hierarchical model, with similar disadvantages
MapReduce (2008)

Not really a DB, but an algorithm framework for large data sets

- Uses first-order functions – fits well with modern PLs
- Flexible data format – no schemas
- No separate SQL language
- Slow, but easily parallelized
Spark, et al.

MapReduce + many other bulk data operations + some SQL ops

- OO library with first-order functions – fits well with modern PLs
- Effective caching in distributed RAM – avoids repeated I/O of traditional MapReduce
Course Pragmatics
Course schedule still being developed
Structure

Classes: lecture + activities

Some labs(?): help with technologies

Assignments: 8-10. Some individual, some group. 55% total

Exams: 3 in evenings 15% each
Technologies

- SQLite
- SQL Server
- Google Cloud
- Jupyter
- GitHub
Information

www.clear.rice.edu/~comp430/

- Go read Honor Code & late policies!

OWL-Space for assignments & grades

Piazza for discussion
Do before next class – Install software

• Python 2.7 – **Get latest version!** – [www.python.org](http://www.python.org)
  - Note that this includes an implementation of SQLite.
  - Install the ipython-sql package – [python-packaging-user-guide.readthedocs.org](http://python-packaging-user-guide.readthedocs.org)

• Jupyter – [jupyter.org](http://jupyter.org)
  - Change the default notebook directory – [jupyter-notebook.readthedocs.org](http://jupyter-notebook.readthedocs.org)
  - Make it convenient to run jupyter-notebook executable – e.g., a shortcut.

• Sample Jupyter notebook and SQLite database
  - 01a-introduction.ipynb, rainfall.db – See schedule on COMP 430 website.
  - Put these in the default notebook directory.

• Run jupyter-notebook to verify installation.