Week 1-2

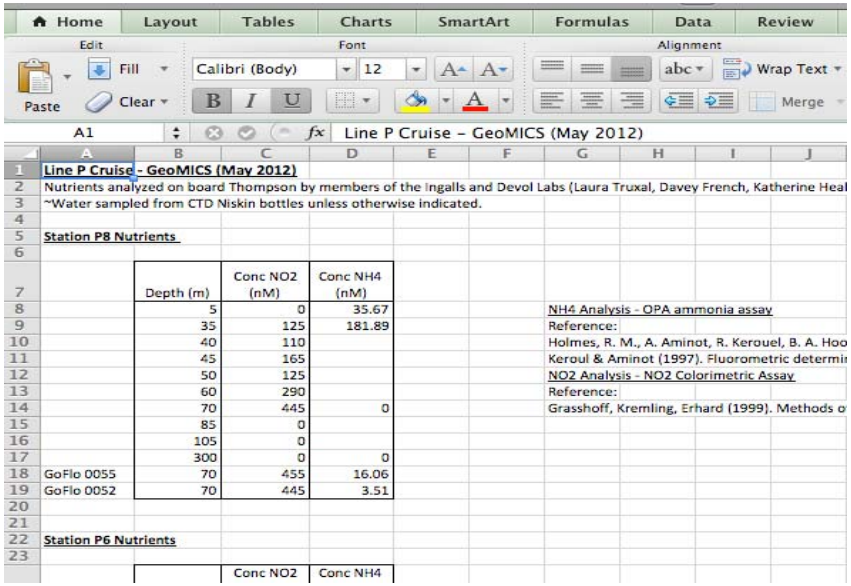
Data Models

Big Data

Prof. Hwanjo Yu
POSTECH

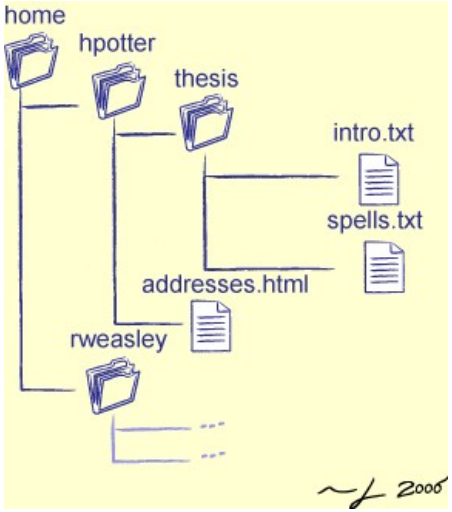
How do we store data?

How do we store data?



	A	B	C	D	E	F	G	H	I	J
1	Line P Cruise - GeoMICS (May 2012)									
2	Nutrients analyzed on board Thompson by members of the Ingalls and Devol Labs (Laura Truxal, Davey French, Katherine Heal									
3	~Water sampled from CTD Niskin bottles unless otherwise indicated.									
4										
5	Station P8 Nutrients									
6										
7		Depth (m)	Conc NO2 (nM)	Conc NH4 (nM)						
8		5	0	35.67			NH4 Analysis - OPA ammonia assay			
9		35	125	181.89			Reference:			
10		40	110				Holmes, R. M., A. Aminot, R. Kerouel, B. A. Hoo			
11		45	165				Keroul & Aminot (1997). Fluorometric determi			
12		50	125				NO2 Analysis - NO2 Colorimetric Assay			
13		60	290				Reference:			
14		70	445	0			Grasshoff, Kremling, Erhard (1999). Methods o			
15		85	0							
16		105	0							
17		300	0	0						
18	GoFlo 0055	70	455	16.06						
19	GoFlo 0052	70	445	3.51						
20										
21										
22	Station P6 Nutrients									
23										
			Conc NO2	Conc NH4						

What is the *data model*?



ANNOTATIONSUMMARY- C O MBINEDORFANNOTATION16_Phaeo_genome

###query	length	COG hit #1	e-value #1	identity #1	score #1	hit length #1	description #1
chr_4[480001-580000].287	4500						
chr_4[560001-660000].1	3556						
chr_9[400001-500000].503	4211	COG4547	2.00E-04	19	44.6	620	Cobalamin biosynthesis protein
chr_9[320001-420000].548	2833	COG5406	2.00E-04	38	43.9	1001	Nucleosome binding factor SPN
chr_27[320001-404298].20	3991	COG4547	5.00E-05	18	46.2	620	Cobalamin biosynthesis protein
chr_26[320001-420000].378	3963	COG5099	5.00E-05	17	46.2	777	RNA-binding protein of the Puf
chr_26[400001-441226].196	2949	COG5099	2.00E-04	17	43.9	777	RNA-binding protein of the Puf
chr_24[160001-260000].65	3542						
chr_5[720001-820000].339	3141	COG5099	4.00E-09	20	59.3	777	RNA-binding protein of the Puf
chr_9[160001-260000].243	3002	COG5077	1.00E-25	26	114	1089	Ubiquitin carboxyl-terminal hyd
chr_12[720001-820000].86	2895	COG5032	2.00E-09	30	60.5	2105	Phosphatidylinositol kinase and
chr_12[800001-900000].109	1463	COG5032	1.00E-09	30	60.1	2105	Phosphatidylinositol kinase and
chr_11[1-100000].70	2886						
chr_11[80001-180000].100	1523						

What is a data model?

Three Components

1. Structures
2. Constraints
3. Operations

Three components of data model

1. **Structures**
   - rows and columns?
   - nodes and edges?
   - key-value pairs?
   - a sequence of bytes?
2. **Constraints**
   - all rows must have the same number of columns
   - all values in one column must have the same type
   - a child cannot have two parents
3. **Operations**
   - find the value of key x
   - find the rows where column “lastname” is “Jordan”
   - get the next N bytes

What is a database?

A collection of information organized
to afford efficient retrieval

http://www.usg.edu/galileo/skills/unit04/primer04_01.phtml

Why would I want a database?

What problem do they solve?

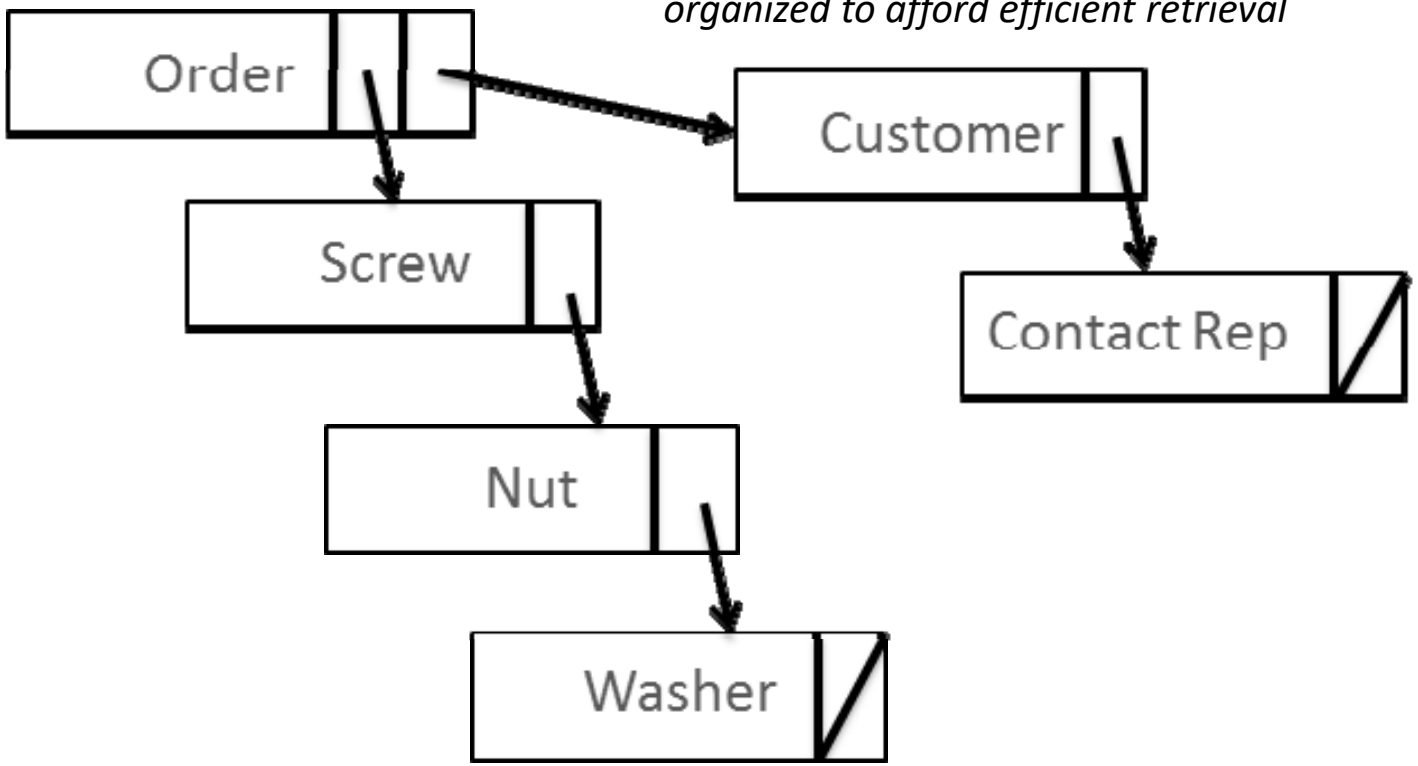
1. **Sharing**
   - Support concurrent access by multiple readers and writers
2. **Data Model Enforcement**
   - Make sure all applications see clean, organized data
3. **Scale**
   - Work with datasets too large to fit in memory
4. **Flexibility**
   - Use the data in new, unanticipated ways

Questions to consider

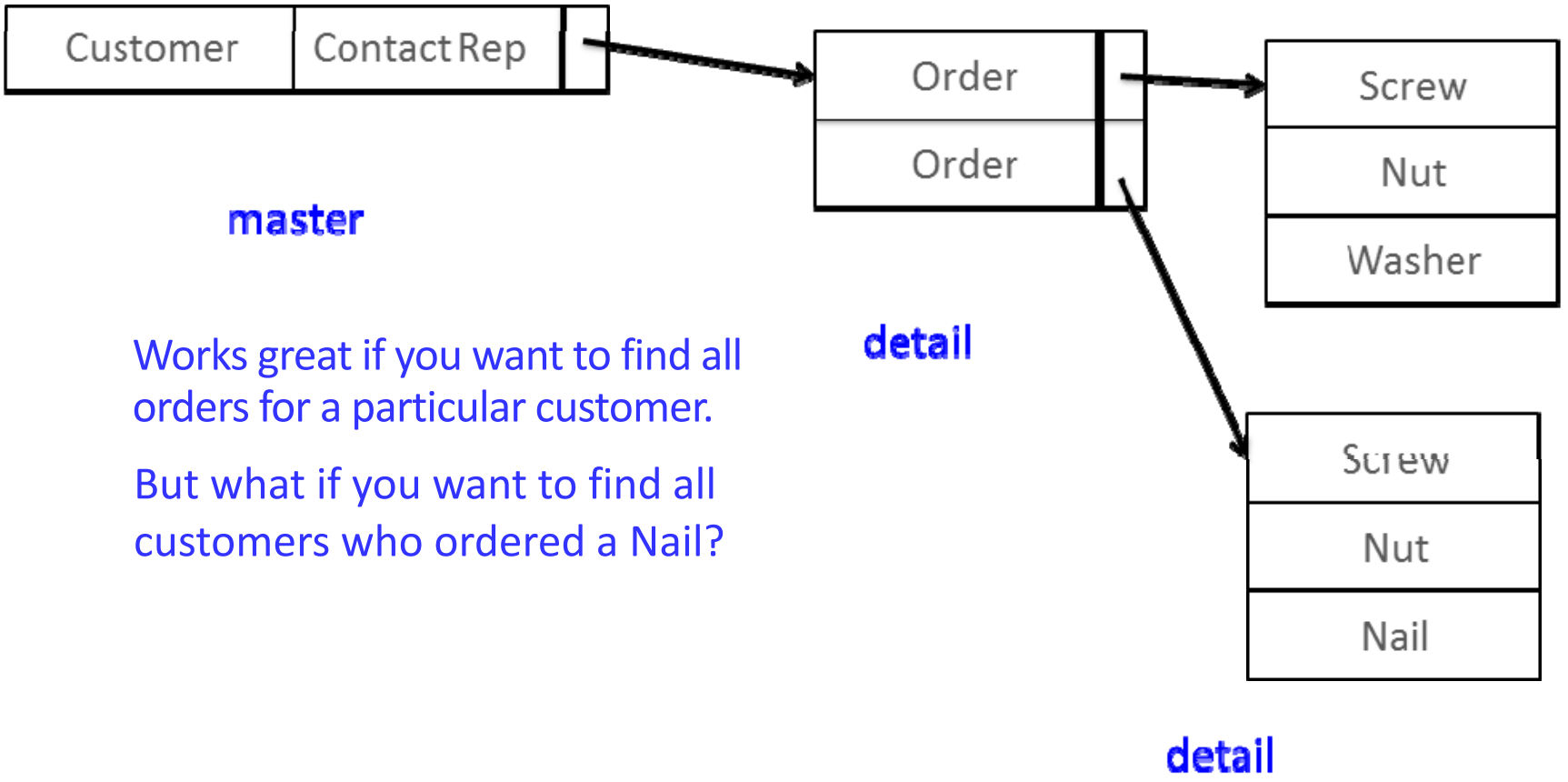
- How is the data physically organized on disk?
- What kinds of queries are efficiently supported by this organization, and what kinds are not?
- How hard is it to update the data, or add new data?
- What happens when I encounter new queries that I didn't anticipate? Do I reorganize the data? How hard is that?

Historical example: Network databases

Database: A collection of information organized to afford efficient retrieval



Historical example: Hierarchical databases



Works great if you want to find all orders for a particular customer.

But what if you want to find all customers who ordered a Nail?

Relational databases (Codd 1970)

"Relational Database Management Systems were invented to let you use one set of data in multiple ways, including ways that are unforeseen at the time the database is built and the 1st applications are written."

(Curt Monash, analyst/blogger)

Relational databases (Codd 1970)

- Everything is a table
- Every row in a table has the same columns
- Relationships are implicit: no pointers
- Processing is equivalent for
  - “find names registered for CSE344”
  - “find courses that Jane registered”

Course	Student Id
CSE 344	223…
CSE 344	244…
CSE 514	255..
CSE 514	244…

Student Id	Student Name
223…	Jane
244…	Joe
255..	Susan

Relational databases (Codd 1970)

Course	Student Id
CSE 344	223…
CSE 344	244…
CSE 514	255..
CSE 514	244…

Student Id	Student Name
223…	Jane
244…	Joe
255..	Susan

- ***Row: record, tuple, instance, object, …***
- ***Column: attribute, field, dimension, feature, …***

Data type and representation

- Record
  - Relational records
  - Data matrix, e.g. numerical matrix, crosstabs
  - Text documents, e.g. term-frequency vector
  - Transaction data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: maps
  - Image data, video data

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Attribute type

- Can be categorized
  - ***Nominal (or Categorical)***, e.g. Type of car, Color name
  - ***Binary*****,** e.g. Gender, Whether to have car or not
  - ***Ordinal***, e.g. Grade
  - ***Numerical***, e.g. Height, Temperature

    or
  - ***Discrete***, e.g. Integer
  - ***Continuous***, e.g. Real

Relational database history

Pre-Relational: if your data changed, your application broke.

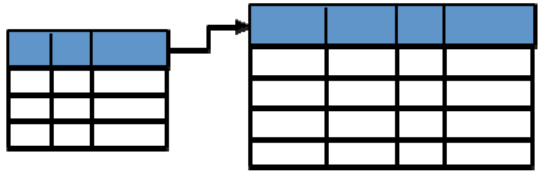
Early RDBMS were buggy and slow (and often reviled), but required only 5% of the application code.

"*Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed.*"

-- Codd 1979

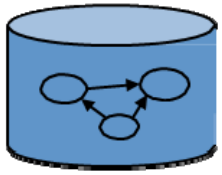
Key Ideas: Programs that manipulate tabular data exhibit an algebraic structure allowing reasoning and manipulation independently of physical data representation

Key idea: “Physical data independence”



relations

physical data independence

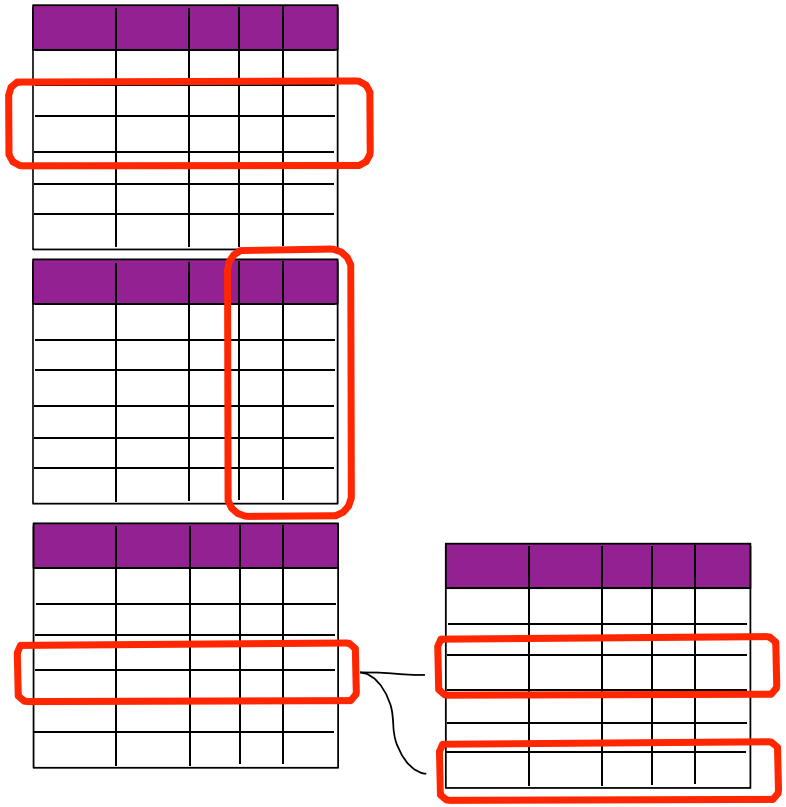


files and pointers

```
SELECT  seq
  FROM  ncbi_sequences
 WHERE  seq = 'GATTACGATATTA';
```

```
f = fopen ('table_file');
fseek (10030440);
while (True) {
    fread (&buf, 1, 8192, f);
    if (buf == GATTACGATATTA){
        . . .
```

Key idea: An algebra of tables



select

project

join

Other operators: aggregate, union, difference, cross product

Key idea: Algebraic optimization

$N = ((z*2)+((z*3)+0))/1$

Algebraic Laws:

1. (+) identity: $x+0 = x$
2. (/) identity: $x/1 = 1$
3. (*) distributes: $(n*x+n*y) = n*(x+y)$
4. (*) commutes: $x*y = y*x$

Apply rules 1, 3, 4, 2:

$N = (2+3)*z$

two operations instead of five, no division operator

Same idea works with the Relational Algebra!

Equivalent logical expressions; different costs

$$\sigma_{p=knows}(R) \bowtie_{o=s} (\sigma_{p=holdsAccount}(R) \bowtie_{o=s} \sigma_{p=accountHompage}(R))$$

right associative

$$(\sigma_{p=knows}(R) \bowtie_{o=s} \sigma_{p=holdsAccount}(R)) \bowtie_{o=s} \sigma_{p=accountHompage}(R)$$

left associative

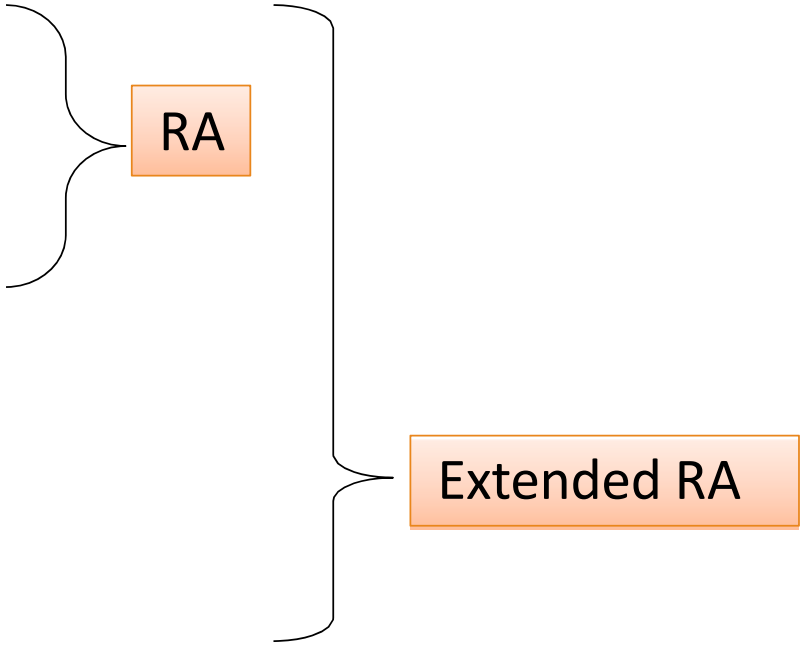
$$\sigma_{p1=knows\ \&\ p2=holdsAccount\ \&p3=accountHompage}(R \times R \times R)$$

cross product

Relational algebra operators

- Union ∪, intersection ∩, difference -
- Selection s
- Projection Π
- Join ⋈

- Duplicate elimination d
- Grouping and aggregation g
- Sorting t



Sets vs. Bags

- Sets: {a,b,c}, {a,d,e,f}, { }, . . .
- Bags: {a, a, b, c}, {b, b, b, b, b}, . . .

- Relational Algebra has two semantics:
  - Set semantics = standard Relational Algebra
  - Bag semantics = extended Relational Algebra
- Rule of thumb:
  - Every paper will assume set semantics
  - Every implementation will assume bag semantics
- Note that ordering is not specified in both set and bag semantics