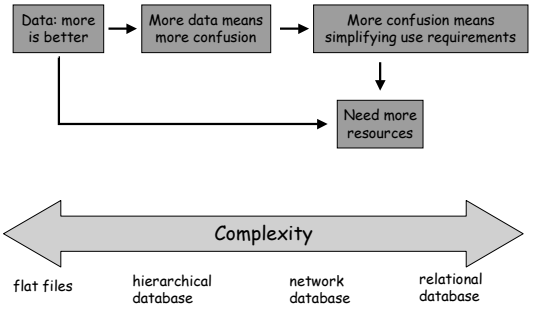
ICS 132: Organizational Information Systems

Information Management and Database Systems

information management

- organisations depend on information
  - about their own processes
  - about what's going on around them
  - the basis of *monitoring* and *planning*
    - the real world is too hard to keep track of
    - information abstracts and summarises it
      - brings the world into alignment with some model
      - denominate the work and treat the results like equations
    - equations represent the work
      - working with the equations tells you whether and how you need to address the work!

the factors at work



data, database, DBMS

- data
  - a big pile of bits
- a database
  - structured collection of data
  - organised according to predefined relations
    - paper documents?
    - contact list on my Pilot?
    - world wide web?
- why bother with a database?
  - need to maintain consistency
  - don't want to have to repeat information

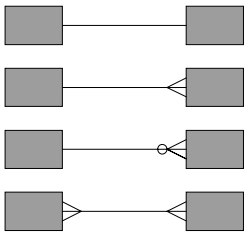
data, database, DBMS

- DBMS: Data Base Management System
  - set of programs to define, update, control databases
    - this is what we often mean when we say "database"
    - Sybase, Oracle, DB2, MySQL, Postgres…
  - DBMS responsibilities
    - layout out information on the disk, building indexes, getting from one piece of data to another
  - your responsibilities
    - modeling the information
    - describing the relations
    - creating queries

ER modeling

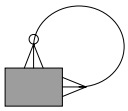
- identifying entities and the relationships between them
  - not unlike OO modelling, but entirely static
- types of relationships
  - one to one
  - one to many
  - optional one to many
  - many to many

ER modeling



ER modeling

- things to remember
  - the simplicity of ER is useful
    - ER is a communication tool – esp. with the participants
  - you're dealing with generic entities, not specific



the relational model

- most common (but not the only one)
- database is a set of tables
  - each table expresses a relation between data items
  - each row of the table is a record
  - each column is an attribute
- not just any table will do
  - for instance, we need a *key field*
    - a field (or set of fields) that uniquely identifies every record
  - other properties are enforced by *normalization*
    - iteratively refining the database format for efficiency

first normal form

- no repeating groups
  - essentially, normalise the record length

Title	Price	Author1	Author2	Author3
Where the Action Is	$30.00	Dourish		
Analyzing Social Settings	$31.95	Lofland	Lofland	
Compilers	$72.00	Aho	Sethi	Ullman

first normal form

- no repeating groups
  - essentially, normalise the record length

Title	Price	Author
Where the Action Is	$30.00	Dourish
Analyzing Social Settings	$31.95	Lofland
Compilers	$72.00	Aho
Compilers	$72.00	Sethi
Compilers	$72.00	Ullman

second normal form

- no non-key attributes depend on part of the key
  - essentially, break the data into many tables

Author	Title	Price	Email
Dourish	Where the Action Is	$30.00	jpd@ics.uci.edu
Baldi	Bioinformatics	$49.95	baldi@ics.uci.edu

second normal form

- no non-key attributes depend on part of the key
  - essentially, break the data into many tables

Author	Email
Dourish	jpd@ics.uci.edu
Baldi	baldi@ics.uci.edu

Author	Title	Price
Dourish	Where the Action Is	$30.00
Baldi	Informatics	$49.95

third normal form

- no attributes depend on other non-key attributes
  - again, break the data into many tables

Author	Title	Price	Purchaser	Date
Dourish	Where the Action Is	$30.00	Maria	12/21/00
Dourish	Where the Action Is	$30.00	Joe	1/1/01
Baldi	Bioinformatics	$49.95	Lisa	1/2/01

third normal form

- no attributes depend on other non-key attributes
  - again, break the data into many tables

Title	Purchaser	Date
Where the Action Is	Maria	12/21/00
Where the Action Is	Joe	1/1/01
Bioinformatics	Lisa	1/2/01

Author	Title	Price
Dourish	Where the Action Is	$30.00
Baldi	Informatics	$49.95

normalisation

- what's the point?

normalisation

- what's the point?
  - eliminate redundancy
  - eliminate opportunities for inconsistency

Author	Title	Price	Purchaser	StudentID
Dourish	Where the Action Is	$30.00	Maria	12/21/00
Dourish	Where the Action Is	$25.00	Joe	1/1/01
Baldi	Bioinformatics	$49.95	Lisa	1/2/01

the transaction model

- normalisation spreads data across multiple tables
  - single action requires many updates
    - a new customer placing a new order?
  - consistency is important
  - transactions group operations into logical units

the ACID properties

- Atomicity
- Consistency
- Independence
- Durability

getting it out again

- query languages
  - SQL is most common
    - "SELECT name,id FROM grades WHERE grade='A'";

getting it out again

- query languages
  - SQL is most common
    - "SELECT name,id FROM grades WHERE grade='A'";

Figure 18 SQL statement generated for complex query

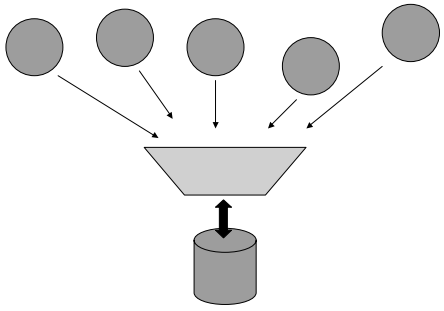
```
Filter String:
  (&(|(objectclass=PERSON)(objectclass=GROUP))(sn=SMITH)(!(member=*)))

SQL Statement:
  SELECT entry.EntryData,
       FROM ldap_entry as entry WHERE entry.EID in
       ( SELECT distinct ldap_entry.EID FROM ldap_entry,ldap_desc
       WHERE
       (!(ldap_entry.EID=ldap_desc.DEID AND ldap_desc.ACEID=?)
       AND
       ldap_entry.EID IN
       (((SELECT EID FROM OBJECTCLASS WHERE OBJECTCLASS = PERSON)
       UNION
       (SELECT EID FROM OBJECTCLASS WHERE OBJECTCLASS = GROUP))
       INTERSECT
       (SELECT EID FROM SN WHERE SN = SMITH)
       INTERSECT
       (SELECT EID FROM ldap_entry WHERE EID NOT IN
       (SELECT EID FROM MEMBER))))
```

getting it out again

SELECT DISTINCTROW HLLink.HLID, HLLink.HLCaseID, HLLink.HLRemarks, HLLink.HLCRCInit, CaseArchive.CaseStatusClass, CaseArchive.CaseStatusCategory, Organizations.OrgName, Organizations.OrgDept, Organizations.OrgAddress1, Organizations.OrgAddress2, Organizations.OrgCity, Organizations.OrgState, Organizations.OrgZIP, Organizations.OrgPhone, Organizations.OrgFax, Organizations.OrgCategory, People.PersonPrefix, People.PersonFirst, People.PersonMiddle, People.PersonLast, People.PersonSuffix, People.PersonTitle, People.PersonAddress1, People.PersonAddress2, People.PersonCity, People.PersonState, People.PersonZIP, People.PersonPhone, People.PersonPExt, People.PersonFax, People.PersonCategory, TimeTable.TTBillingID, TimeTable.TTTaskID, TimeTable.TTUser, TimeTable.TTStart, TimeTable.TTSeconds, TimeTable.TTAddMin, Cases.CaseReportingRegion, Cases.CaseReportingState, Cases.CaseTypeClass, Cases.CaseTypeCategory, Cases.CaseStatusClass, Cases.CaseStatusCategory, Cases.CaseDiagnosis, Cases.CaseIsOffLabel, Plans.PlanClass, Plans.PlanCategory, Plans.PlanTitle, Products.ProductName, Diagnoses.Diagnosis FROM Diagnoses RIGHT JOIN (Organizations RIGHT JOIN (TimeTable INNER JOIN ((Plans RIGHT JOIN (People RIGHT JOIN (OPLink AS PhysicianOPLink RIGHT JOIN (Products RIGHT JOIN (Cases RIGHT JOIN HLLink ON Cases.CaseID = HLLink.HLCaseID) ON Products.ProductID = Cases.CaseProductID) ON PhysicianOPLink.OPID = Cases.CasePhysicianOPID) ON People.PID = HLLink.HLPID) ON Plans.PlanID = Cases.CasePlanID) LEFT JOIN CaseArchive ON HLLink.HLID = CaseArchive.HLID) ON TimeTable.TimerID = HLLink.HLTimerID) ON Organizations.OID = HLLink.HLOID) ON Diagnoses.DiagAbbrev = Cases.CaseDiagnosis;

3-tier architecture



distributing databases

- managing information access needs
  - locality
  - performance
- three forms of distribution
  - distributing tables
  - distributing rows
  - replication
- two-phase commit
  - "can commit?"
  - "do commit!"

alternatives to relational

- object-oriented
  - hierarchical schemas
  - migrate code closer to data
- text databases
  - free-form indexing
  - less structure
    - but more useful for unanticipated queries
- geographical information systems
  - not a natural model for relational systems

organisational perspectives

- information all comes with a point of view
  - complete information is a myth; so what is left out?
- information models encode assumptions
  - about the state of the world or the objects modeled
  - example: US Army deployment
- normalisation distributes information
  - distributed locus of power and control

management concerns

- information quality
  - bad information is worse than none at all
    - it's easy to load a database with accurate information
    - it's harder to maintain the accuracy over time
    - distribution makes this worse
      - multiplicity of information, lack of "human access control"
- accessibility
  - the point of having the information is to use it
    - availability
    - admissability
    - but there's a down side…
      - once you have information, you may have to disclose it
      - security! (remember the risks, from last week)

summary

- key points:
  - information processing is about making the world tractable
    - amenable to summarisation, modeling & prediction
  - DBMS provides a framework for data management
    - regularised for efficiency, consistency & maintenance
  - think about where the database fits
    - technically
    - organisationally
    - politically

homework

- See the web site for details
  - two questions
    - exercise in transforming a database into 1NF, 2NF, 3NF
    - explore DNS as a distributed database
  - due at next Wednesday's lecture

what's coming up

- Friday
  - discussion section
  - homeworks back
- Monday
  - performance and competition
  - Alter chapter 6