Outline

- Announcements
- Databases (cont’d)
- Algorithms and Protocols
- Student Presentations
- Akamai
Announcements I

- Database Assignment due 12/2 (submit electronically)
- Business paper - due 12/2 (last day of instruction)
Announcements II

- **Student Presentations** next week?
  - ??

- **Reading:**
  - Chapter 10 of Messerschmitt (Reader 1)
  - American Airline Case Study (Reader 2)
  - Chapter 1 on Networking

- **2nd Database tutorial**
  - Friday, Dec. 2, 3:00 p.m., BE109
Student Presentations

- Rachel Karagianes - Artificial Skin
- Eleonor Concepción - Galaxy Hotel System
The Relational Model

EMPLOYERS

<table>
<thead>
<tr>
<th>EMPL_ID</th>
<th>EMPL_NAME</th>
<th>EMPL_POSITION</th>
<th>DEPT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Alice</td>
<td>Manager</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
<td>Programmer</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
<td>Manager</td>
<td>2</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
<td>Accountant</td>
<td>2</td>
</tr>
</tbody>
</table>

DEPARTMENTS

<table>
<thead>
<tr>
<th>DEPT_ID</th>
<th>DEPARTMENT</th>
<th>DEPT. ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>2</td>
<td>Finance</td>
<td>New York</td>
</tr>
</tbody>
</table>
Each operation results in a new table

“PROJECT”

“SELECT”
Database Operations

<table>
<thead>
<tr>
<th>EMPLOYERS</th>
<th>DEPARTMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMPL_ID</td>
<td>DEPT. ID</td>
</tr>
<tr>
<td>100</td>
<td>Alice</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
</tr>
</tbody>
</table>

JOIN

<table>
<thead>
<tr>
<th>EMPL_ID</th>
<th>EMPL_NAME</th>
<th>EMPL_POSITION</th>
<th>DEPT.</th>
<th>ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Alice</td>
<td>Manager</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>101</td>
<td>Bob</td>
<td>Programmer</td>
<td>IT</td>
<td>San Jose</td>
</tr>
<tr>
<td>102</td>
<td>Chris</td>
<td>Manager</td>
<td>Finance</td>
<td>New York</td>
</tr>
<tr>
<td>103</td>
<td>David</td>
<td>Accountant</td>
<td>Finance</td>
<td>New York</td>
</tr>
</tbody>
</table>
Application Logic and Tables

Application logic (programming language) ➔ SQL ➔ Database
Databases & OLTP

- Click to add text
Recall - Two capabilities

- **Aggregation**: accessing multiple databases
- **Sharing**: two or more applications accessing the same databases
Example - Travel Agency

Travelocity.com

CheapTickets.com

Hotels

Cars

Airtickets

What can go wrong?
Example - Travel Agency

Travelocity.com

CheapTickets.com

Hotels

Cars

Airtickets

A resource might be unavailable
Two applications might try to access & update the same resource concurrently.
An application or a host might crash before the completion of the transaction
A customer’s transaction should be completed in its entirety, or aborted
Transaction Processing

“The coordination of multiple resources and the shared access to common resources in a systematic and consistent way”

Examples?

- Financial applications (stock market, ATMs)
- Reservations (travel, theatre)
- Manufacturing (inventory, purchasing, billing)
- Etc...
Online Transaction Processing (OLTP)

- Transaction Processing for networked applications

- 4 Important Properties of transactions: ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability
The ACID properties

- **Atomicity**
  - All transaction components should either complete together (commit) or abort
  - E.g. All reservations (airline, hotel, car) should be grouped as a single transaction that either commits, or aborts

- **Consistency**
  - A transaction must leave the system in a consistent state at the end of the transaction, or else abort
  - E.g. Either a consistent set of reservations has been made, or none

- **Isolation**
  - Concurrent transactions are allowed only if they don’t interfere with each other
  - Two travel agents can concurrently access the same database if the reservations are for different dates/places

- **Durability**
  - A transaction leaves the resources in a permanent state after it commits
Structure of a Transaction

Durable
Starting State

Actions to be performed

Successful completion

Durable, Consistent
End State

Abort

Rollback
OLTP

- Simplifies application development

- Enables protection and integrity of mission-critical data in a transparent way
  - for the end user
  - for the application developer
Monopoly players protocol

Player 1

Player 2

One-turn algorithm

This is a protocol interaction diagram
Application and infrastructure

The application defines its own application-level protocols.

Internally, the network uses protocols to implement the services it provides.
Example:

- **HHC Server**
- **Wireless Link**
- **HHC**
- **HEADQUARTERS**
  - **Airline Dataserver**
  - **Airline Intranet**

Diagram showing an airport setting with an airline server and wireless link to an HHC device.
Layered Protocols Example

- HHC Server Application
  - Windows OS
  - Break Messages into Packets
  - Networking Infrastructure

- HHC Server

- Application Level Protocol
  - Request Pass. Data

- HHC Application
  - Palm OS
  - Networking Infrastructure

- Send Packets
  - Acknowledge Packet
  - Link Level Protocol

- Send Pass. Data As Message
Example: HTTP (Hyper Text Transfer Protocol)

1. User activates URL
2. HTTP request
3. HTTP response (embedded document)
4. Browser displays document (if HTML) or invokes “helper application”
Example

Path from □ to □ is (R,D,D,D,R,R,R,R)

Is (R,D,D,D,R,R,R,R) an address?
No! -- not an address, because it depends on starting point.
Example

Address of \( (6,5) \)

Route from can be inferred
Program

Program

- Precise description of an algorithm in a formal language that is called programming language
- Actions are applied to data
Formulation in a language

- **Natural language**
  - No strict syntactic rules
  - Great density and semantic capability

- **Formal language**
  - Strict syntax and semantics

- **Programming language**
  - Formal language in which computations can be described
  - Executable by an electronic computer
Can we solve all problems?

Collatz Conjecture (Ulam):

```plaintext
while x!=1 do
    if (x is even) then x=x/2
    else x=3*x+1
```

Example:

```
7 → 22 → 11 → 34 → 17 → 52 → 26 → 13 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1
```

Given *any* arbitrary number \( x \), will the program terminate?

Open problem!
Translation of programs

Source Code
\((in\ a\ programming\ language)\)

\[\Downarrow\]

Compiler

Input \(\rightarrow\) Executable program \(\rightarrow\) Output
\((machine\ language)\)
Akamai Freeflow

Web Page

Large Company

Text....

Local Office or ISP

Akamai Server

INTERNET

NSP 1

NSP 2

Web Server
Quiz 4  (total 10 pts)

- What is SQL?
- How long (in bits) is an IP address?
- Akamai is famous for what?