FIT100

Test Your Tech

JavaScript is:

A. The earliest known writing by Java Man.
B. Programming language for Web pages.
C. Instructions in the Starbucks bag on how to brew good coffee.

1

FIT100

Test Your Tech

JavaScript is:

A. The earliest known writing by Java Man.
B. Programming language for Web pages.
C. Instructions in the Starbucks bag on how to brew good coffee.

2

FIT100

Announcements

- No quizzes for rest of quarter
- No final

FIT100

Announcements

- Project 2B
  - Submit to Catalyst Collect-It by tonight before 10pm
  - Finishing up:
    - Turn in what you have
    - If something isn't working, put in comments
    - We're grading for *effort*, not for "perfect"

FIT100

Announcements

- Labs 13 and 14
  - Submit to Catalyst Collect-It by Tuesday, 12/11/07, before 10pm

FIT100

Announcements

- Extra-Credit Labs
  - Labs 10. 12, 15, and address-munging
    - Address-munging will be posted this weekend
  - Worth up to 25 points extra credit
- Submit to Catalyst Collect-It by Tuesday, 12/11/07, before 10pm

FIT100

Announcements

- Extra-credit papers:
  - ∗ Worth up to 25 extra-credit points
  - ∗ See course listserv archives for topics
  - ∗ New topic: Describe how the FIT course could be made more relevant for your community
    - Community = major, age group, ethnicity, heritage, identity, technical background
- Submit to Catalyst Collect-It by Tuesday, 12/11/07, before 10pm

FIT100

Announcements

- Project 3
  - ∗ We'll go over it in lecture today
  - ∗ Because you don't have as much time as usual,
    - I've created a database to get you started
    - Download it from the course calendar
    - Two of the three tables are already done
    - One query is already written
- Submit to Catalyst Collect-It by Tuesday, 12/11/07, before 10pm

FIT100

Announcements

- Questions about due dates?

FIT100

Designing Databases

Designing a database requires a "needs analysis"

©2004, Lawrence Snyder

FIT100

Standard Process

There are guidelines (no algorithm is possible) for creating a database

- Needs Analysis
- First cut at a physical DB solution
- Refinement of first cut … assess/improve
- Define relationships, and create tables
- Formulate the logical DB solution
- Refinement of the logical database
- Create the queries and GUIs
- Assess

FIT100

Needs Analysis

"Needs analysis" -- study the activity to determine what kind of DB is needed

- Identify who will create information, who will use it
- Find out the information gathered
- Find out what information is needed to conduct the activity
- Find out when the information is created, when it is needed, and how long it must be saved

“First Cut” --Initial Design

Using Entity-Relationship diagrams, create the best physical DB

- Design tables for the information created
- Limit tables to simple entities
- Worry about redundancy
- Assess -- does it make sense?

The design process is iterative … assess, improve, refine

Building First Physical DB

Build a version of the physical DB and create a few records to test work

- The design is fluid -- don’t invest much time in building sample files
- See if it is possible to follow the creation & use of the information through the system
- Eliminating poor designs now saves time

In designing a physical DB you will use relationships … specify them when the design is stable

Logical database

To formulate the views users want, a new needs analysis may be needed

- Who are the users to be supported by DB
- What information does each person need to see, what information do they enter?
- Using ER diagrams, formulate the tables needed for each user’s view
- Assess and refine

Building Logical DBs

With the design in hand, formulate tables using this strategy :

- Catalog what tables will provide the information for a given view table
- Create a supertable (probably using join) containing all data in the view table
- Decide which fields are needed and in what order, and “trim” to that
- Formulate an SQL query for the table

Implementation

A skeleton implementation is built to test out the design, then proceed…

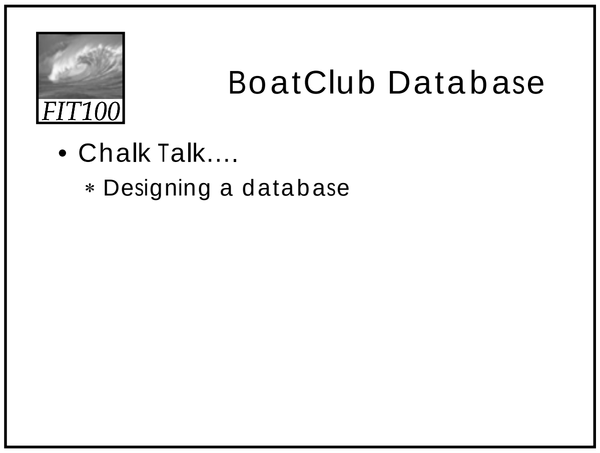
- Try out the full system to assess how it works
- Teach it to the users, and let them try
- Revise and retest if necessary
- Enhance and “bullet proof”

When the system is “deployed”, run both systems briefly to assure nothing falls through the cracks

BoatClub Database

Project 3 illustrates the ideas of database design … we will try it

- It is much easier to design “on paper” than with computer software … so work out the whole design before picking up the mouse



BoatClub Database

- Chalk Talk….
  - Designing a database