Database Basics

CSE 100

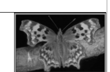
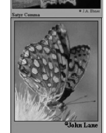
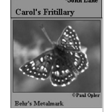
Much of the data people keep is not stored as text, but rather is organized in the form of tables. Knowing how that data is structured and becoming proficient at manipulating it is the key to finding things out.

FIT 100 Displaying Information Informally

- Tables are a common way to present information

Informal tables must be *regularized* to be database tables ...

North American Butterflies

Common Name	Scientific Name	Wingspan	Range	Larval Food	Adult Food	Photo
Satyr Comma	*Polygonia satyrus*	1 3/4 - 2 1/2	North America	Nettles	Tree sap	
Carol's Fritillary	*Speyeria carolae*	2 3/8 - 3 3/8	Clark County NV	Charleston Mountain Violet	Flower nectar	
Behr's Metalmark	*Apodemia virgulti*	3/4 -15/16	Southern California, Baja	Buckwheat	Flower nectar	

- A (relational) database is a set of tables -- tables describe *entities*

FIT 100 Terms of a Database Table

- Name the structural parts of database tables

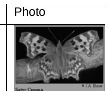
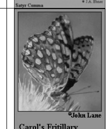
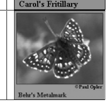
A DB key is a field(s) that's unique for each row

Attribute (Field)

Table name

Tuple (Row) (Records)

Butterfly

Name	Genus	Species	LWS	GWS	Range	L_Diet	A_Diet	Photo
Satyr Comma	*Polygonia*	*satyrus*	1.75	2.5	North America	Nettles	Tree sap	
Carol's Fritillary	*Speyeria*	*carolae*	2.375	3.375	Clark County NV	Charleston Mountain Violet	Flower nectar	
Behr's Metalmark	*Apodemia*	*virgulti*	0.75	0.9375	So. California, Baja	Buckwheat	Flower Nectar	

- Attributes have types -- Species is a string, Photo is jpg

FIT 100 Terminology

- The structure of a database is its *database schema*
- The schema specifies …
  - The list of tables forming the database
  - For each table, the fields of its records
  - For each field, its properties, i.e. data type, key or not key, default value, etc.

Butterfly

Name	String	
Genus	String	
Species	String	
LWS	Double	*The least wing span measurement*
GWS	Double	*The greatest wing span measurement*
Range	String	
L_Diet	String	*Larval Diet*
A_Diet	String	*Adult Diet*
Photo	JPEG	

Primary Key – Genus:Species

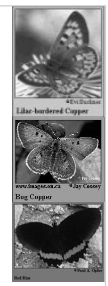
FIT 100

Instances

- A database as the word is normally used, i.e. tables with specific contents, is known as a database *instance* (of a data base schema)
- There can be many instances of a single database schema

Butterfly

Lilac-bordered Copper	*Lycaena nivalis*	1 - 1 3/8	Rockies & west	Douglas' knotweed	Flower nectar
Bog Copper	*Lycaena epixanthe*	7/8 - 1	Great Lakes & east	Cranberries	Raindrops
Red Rim	*Biblis hyperia*	2 - 3	Rio Grande Valley TX	Noseburn and spurge	Rotting fruit



FIT 100

Redundancy is Very Very Very Bad

- Database design is the process of setting up a database schema
- Not every design is good … avoid redundancy

  Information is redundant if it is stored in multiple places in a database
- Example: UW groups needing your home address
  - Administration -- to send tuition bills
  - Dean -- to send notification of being on "Dean's List"
  - IMA -- to send you back your locker deposit

Multiple copies of the same information can have different values in different locations, i.e. be *inconsistent*, its worse than no data

FIT 100 University DB Design

- Avoiding redundancy … keep specialized tables
- The Administration keeps permanent records

Personal Data

Student
Student_ID
First_Name
Middle_Name
Last_Name
Birthdate
Citizenship

Primary Key: Student_ID

Home_Base
Student_ID
Street_Address
City
State
Country
Postal Code

Primary Key: Student ID

Permanent Address

- Others keep their own tables

These tables have no redundancy

Dean's Data

Top_Scholar
Student_ID
Nick_Name
Major
GPA

Primary Key: Student_ID

Good_Sport
Student_ID
Locker Number
Deposit_Amt
Awards

Primary Key: Student_ID

IMA Data

FIT 100 Combining Tables

- When the Dean accesses data, combine lists

Dean Data

Student_ID	Nick_Name	Major	GPA
1006616	Candy	PoliSci	3.82
1007250	Jake T	Chem	3.79

Permanent Address

Student_ID	Street_Address	City	State	Country	Postal_Code
0998185	1 Rocky Rd	Burnaby	BC	Canada	V6N 4T4
1006616	61 Peekaboo St	Vail	CO	USA	71021
1006681	1011 Passing Ln	Ames	IA	USA	50502
1007250	4 Wheel Dr	Daytona	FL	USA	32330
1010043	302 Second Pl	New York	NY	USA	10011
…	…	…	…	…	…

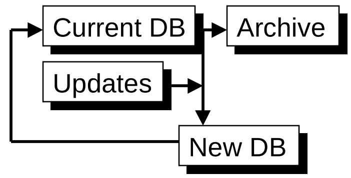
Student_ID	Nick_Name	Major	GPA	City	State	Country	Postal_Code
1006616	Candy	PoliSci	3.82	Vail	CO	USA	71021
1007250	Jake T	Chem	3.79	Daytona	FL	USA	32330
1010043	Broadway Joe	History	3.80	NewYork	NY	USA	10011
…	…	…	…	…	…	…	…

Concept: Arrange tables to avoid redundancy, join tables to provide the data users want to see

FIT 100

Organizing Business Data

- Companies, universities, government agencies, etc. have many database applications in common
  - Employee records
  - Payroll records
  - Customers/clients/students records
  - Products/services listings
  - …



- Databases, as described thus far, are completely adequate for representing and managing this data
- Database changes can be controlled
  - Changes can be made by “authorized” employees
  - Changes can be made periodically, in batches

FIT 100

Maintaining On-line Databases

- Many databases are only useful if they are on-line, i.e. can be changed interactively
  - Airline reservations
  - Credit card accounts, ATMs and other banking
  - Catalog merchandising
  - eCommerce
  - …
- On-line changes are called *transactions*
- The “transactions concept” not only makes interactive DBs possible, it is a good metaphor for other database processing as well

FIT 100 Transactions ...

- A transaction is a single operation (reference or change) to a database usually involving only one or a few records
  - Credit card purchase
  - ATM withdrawal of cash
  - Flight reservation
  - …
- Many transactions are taking place at once, typically
- Keeping the DB “correct” is a problem!

VISA processed 110,086,395 transactions on December 14, 1998, a 1-day world’s record

FIT 100 Correct Database?

- Because transactions take place simultaneously, there is a possibility that two computers can be making changes to the same data at the same instant, possibly corrupting it …
- Consider two transactions
  - T1: Deposit $100 into Account #12345
  - T2: Withdraw $100 from Account #12345
- When transactions are over, the balance *should be* unchanged
- But what if the transactions take place “at once”?

FIT 100 Tale Of Two Transactions

Balance: $500

- T1: Deposit $100 — T2: Withdraw $100
- `T1.1: Fetch balance for` `T2.1: Fetch balance for`
- `Acct #12345: $500.00` `Acct #12345: $500.00`
- `T2.2: Balance >= $100?`
- `T2.3: Yes, Bal = Bal - $100`
- Balance: $400 `T2.4: Set DB balance to $400`
- `T2.5: End transaction`
- `T1.2: Bal = Bal + $100`
- `T1.3: Set DB balance to $600` Balance: $600
- `T1.4: End transaction`

Time

- The depositor might be pleased, but the bank won't be!

FIT 100 Correctness

- The DB System must assure that every DB change happens as if the transactions happened one-at-a-time
- The one-at-a-time protocol solves the "problem" with T1 and T2:
  - T1 applied first, then T2: $500 --> $600 --> $500
  - T2 applied first, then T1: $500 --> $400 --> $500
- Transaction processing systems make sure such problems do not arise by "locking" the data (only one computer at a time can unlock the data)

"Two changes at once with unsavory results" can happen when multiple people work on the same document … use locking idea

FIT 100 Reliability

- What happens to the database when …
  - The power goes out
  - Someone spills coffee into the disk drives
  - The computer crashes with all the changes to the DB for the last three hours in its (volatile) RAM
  - A new employee accidentally deletes the payroll file before printing the pay checks?
  - A virus cleans off the corporate disks
  - A hacker infiltrates the enterprise and begins transferring funds to a Swiss bank account
  - A disgruntled employee deletes the retirement plans and stock option accounts for senior management
  - ...

These problems could happened to you, too

FIT 100 Basic Mechanisms

- Several techniques preserve the integrity of the data
  - Error detection/correction in the hardware
  - Passwords and authentication assist in verifying that the person(s) making changes are legitimate
  - Validation … verify that changes to the DB are “plausible”
  - “Commitment” … keep record in a safe place of all changes to the database, and then when it has been verified, make the actual change effective; deletions never actually result in the data being removed

Backup copies of a DB must be made regularly, and kept off-site

Do you back up your information?

FIT 100

Redundancy is Very Very Very Good

- To protect against computer crashes, disk failures, loss of power, etc. duplicate the hardware, disks, power sources, etc.
- The duplicate systems can compare answers as a means of detecting errors
- RAID systems are arrays of disks that contain "hot spares" and special data encodings to recover from disk failures
- By keeping a snapshot of the database and a record of all of the transactions, *it is possible in case of catastrophic disaster to reconstruct the database by applying all of the transactions to the old database*