Database Basics
and
An Introduction to the
Relational Database Model

FIT 100

FIT 100 What is a Database?

- ❖ A collection of information/data that is important to us
  - ❑ Can be manual (a telephone book)
  - ❑ Nowadays, mostly electronic
    - ✢ Organized so a computer can quickly access and retrieve the desired information
- ❖ We create these collections so that the information/data stored in them can be found when needed
- ❖ The most fundamental unit of a database is a table

FIT 100 Why Know About Databases?

- ❖ They will one day take over the world! ☺
- ❖ Much of the data that is stored out in the world is in the form of tables. Tables are, or can be, highly structured
- ❖ All technologies-Internet, etc. are more powerful when they can connect to a database of information stored for a particular purpose.

FIT 100 Why Know About Databases?

- ❖ Knowing how the data is structured and becoming proficient at retrieving and manipulating it is a very powerful skill to have.

> “If spreadsheets are the ‘number crunchers’ of the digital world, databases are the real ‘information crunchers’. Databases excel at managing and manipulating structured information.”
>
> ~Rose Vines
> GeekGirls.com

So, DBs do all that…Why do I care?

- Databases are a way to store and process certain things and events in the world we consider important.
- Databases can hold:
  - The past
  - The present
  - A prediction of the future?
    - Computation on data can reveal relationships not previously identified, or predict results under certain conditions
- But in order for databases to accurately store and display that information, it must be structured
  - Remember, computers are exacting, discrete

Data Storage before Relational Databases– Flat Files

- Computing up until ~1980
  - File based data storage (file processing systems)
  - Grouped similar data together in separate flat files
  - Programs had to be written to :
    - Read data from a file on the disk
    - Process data (enter student grades, update employee salary, look up titles in book inventory)
    - Write data back to the file
- But there are problems with file based data storage
  - They deal directly with how we think

Summer

Problems with Flat Files and File-Processing Systems

- Need to write a program to get at data
  - Too hard to get to data
    - Time consuming and expensive
  - Too easy for anyone who writes the program to get to data
  - Data often duplicated
    - Many different pieces of data associated together in one file and then duplicated across files
  - Data dependency
    - If you change format of any of data in flat file system, then you are forced to change all programs that access the data

      Zip codes changing from 5 to 9 digits
  - Don't know which programs are using the data
    - Files full of data on a disk don't tell you who accesses them

Main Problem Is Redundant Data

Name	**Course**	**Room#**	**Instructor**
Paul Stevens	FIT 100	MGH 420	Whiteaker
Holly Eggelston	FIT 100	MGH 420	Whiteaker
Stephanie Wright	LIS 540	EE1 045	Boiko
Lisa Spagnolo	INFO 480	EE1 025	Whiteaker
Pam Green	FIT 100	MGH 420	Whiteaker
Thomas Nguyen	LIS 540	EE1 045	Boiko
Lisa Spagnolo	LIS 540	EE1 045	Boiko

…

- Redundant data leads to data anomalies
  - Integrity and update anomalies
- Bad data leads to bad decisions

Redundant Data, Redundant Data

- ❖ When storing the same data in multiple locations, the likelihood of inconsistency is very high.
- ❖ What is my real name?
  - ❑ Table 1: my name is Grace
  - ❑ Table 2: my name is Graciela
  - ❑ Table 3: my name is Grce
  - ❑ Table 4: my name is Grase

Multiple copies of the same information can have different values in different locations. Inconsistency of information is worse than no information

- ❖ Relational Database technology was developed to reduce/eliminate data redundancy in information stores that undergo constant updates

Relational Database Basics

- ❖ Information is stored in tables
  - ❑ Tables consist of columns (attributes) and rows (records)
  - ❑ Every row is unique and identified by a key (values in one or more columns)
- ❖ Each table only stores information about one thing or theme
- ❖ Relationships/associations are also stored
  - ❑ We'll see how in a minute

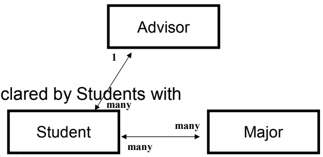
But, before you can have a database or tables…

- ❖ You first have to know what you need tables of:
  - ❑ Why are you creating the system?
- ❖ The best way to do that is to get a picture/model of all the things that should become tables:
  - ❑ What do you want to keep track of? What is important?
- ❖ The world can be reduced to 2 things:
  - ❑ Things or themes (people, items/objects, events, actions)
  - ❑ Relationships
    - ✣ How those things or themes relate to each other-or don't
- ❖ One way of deriving those pictures out of a user's need is to create a model to show things and their relationships

How Do You Model Real World Associations?

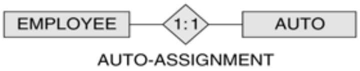
The Student Advisor Model

- ❖ There are Students with
  - ❑ Name
  - ❑ Address
- ❖ There are Advisors with
  - ❑ Name
  - ❑ Department
- ❖ There are Majors that are declared by Students with
  - ❑ Name
- ❖ There are relationships
  - ❑ Each Student has one Advisor
  - ❑ Each Advisor has many Student advisees
  - ❑ One Student can have many Majors
  - ❑ One Major can have many Students

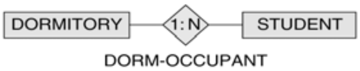


FIT 100 Kinds of Relationships

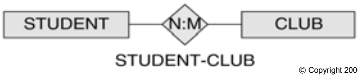
- One to One



- One to Many



- Many to Many



FIT 100 Relational Database Technology

- Developed by E.F. Codd in the early 70's
- Data is integrated
- Data duplication is reduced
  - Done by normalizing tables: only store information on one "thing" or "theme"
- Data is easy to understand

FIT 100 An Informal Table

Name	**Course**	**Room#**	**Instructor**
Paul Stevens	FIT 100	MGH 420	Whiteaker
Holly Eggelston	FIT 100	MGH 420	Whiteaker
Stephanie Wright	LIS 540	EE1 045	Boiko
Lisa Spagnolo	INFO 480	EE1 025	Whiteaker
Pam Green	FIT 100	MGH 420	Whiteaker
Thomas Nguyen	LIS 540	EE1 045	Boiko
Lisa Spagnolo	LIS 540	EE1 045	Boiko
…			

FIT 100 Relational Database Technology

- To normalize a table for use in a relational database
  - There should be one table for each type of object or "entity"
  - For the informal table just shown, each of the following would be given its own table structure:
    - Student table to list all students – one row for each student
    - Course table to list all courses – one row for each course
    - Instructor table to list all instructors – one row for each instructor
    - Class_Event table to show each student, instructor and course instance associated together at a certain place and time
  - Each row has a unique identifier: a key

Key Structure, Relationships and Implementation

- A *key* is a group of one or more attributes that uniquely identifies a row

- Keys are unique and not null
  - Primary keys are noted in table design by developers to uniquely identify a row. Primary means it has been selected instead of other possibilities to be the unique identifier for the row

  - Primary Key fields are indexed

Storing Relationships using Keys

- Modeling a view of the world is one thing, actually storing it in a database is another

- In Relational Database implementation, the rules are simple:
  - If the association/relationship to be stored is one-to-many (1:N), place the attribute identified as the primary key from the one table as an additional attribute in the many table.
    - This additional attribute is known as a foreign key

    - It is a key that is sitting in a table foreign to the table in which it is a primary key

Storing Relationships using Keys

- If the association/relationship to be stored is many-to-many (M:N), a new table structure must be created to hold the associations. This “bridge” table will have as foreign key attributes, the primary key of each table that is part of the relationship
    - The key for the bridge table then becomes either:
      - – The combination of all the foreign keys OR
      - – A new attribute will be added to added as a surrogate key

The Keys Are the Key to Relational DB’s

- Keys show how tables are related to one another

Student

SID	Name
1	Paul Stevens
2	Holly Eggelston
3	Stephanie Wright
4	Lisa Spagnolo
5	Pam Green
6	Thomas Nguyen

Course

CourseID	Name	Room
10	FIT 100	MGH 420
20	LIS 540	EE1 045
30	INFO 480	EE1 025

Class_Event

SID	CourseID	InstID
3	20	200
4	30	100
1	10	100
2	10	100
4	30	200

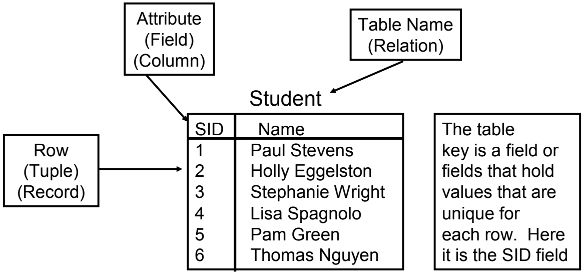
Instructor

InstID	Name
100	Whiteaker
200	Boiko

Note:
No Data Redundancies

FIT 100 Terms of a Database Table

- The structural parts of database tables are:



Student

SID	Name
1	Paul Stevens
2	Holly Eggelston
3	Stephanie Wright
4	Lisa Spagnolo
5	Pam Green
6	Thomas Nguyen

The table key is a field or fields that hold values that are unique for each row. Here it is the SID field

- Attributes have types – SID is an Integer, Name is a string

FIT 100 General Database 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.

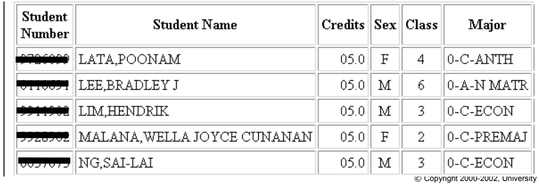
Student

SID	Integer	*student UW ID #*
FName	String	*student first name*
LName	String	*student last name*

Primary Key: SID

FIT 100 Instances

- When most people think “database”, they mean a particular instance: tables with specific contents
  - Known as a *database instance* (of a database schema)
    - Construction blueprints (a schema)
    - The actual home after construction (an instance of that schema)
- There can be many instances of a single database schema



Student Number	Student Name	Credits	Sex	Class	Major
[redacted]	LATA,POONAM	05.0	F	4	0-C-ANTH
[redacted]	LEE,BRADLEY J	05.0	M	6	0-A-N MATR
[redacted]	LIM,HENDRIK	05.0	M	3	0-C-ECON
[redacted]	MALANA,WELLA JOYCE CUNANAN	05.0	F	2	0-C-PREMAJ
[redacted]	NG,SAI-LAI	05.0	M	3	0-C-ECON

FIT 100 Again, Redundancy is Very, Very Bad

- Database design is the process of modeling a user’s view of some part of the world and setting up a database schema
- Not every design is good… there are a lot of databases out there that don’t avoid redundancy

Information is redundant if it is stored
in multiple places in a database

- Relational Databases minimize data redundancy by creating tables that store one thing or theme.
  - Associations among things are shown with keys