

Database Basics and An Introduction to the Relational Database Model



© Copyright 2000-2001, University of Washington



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

© Copyright 2000-2001, University of Washington



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.

© Copyright 2000-2001, University of Washington



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

© Copyright 2000-2001, University of Washington

FIT 100 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

© Copyright 2000-2001, University of Washington

FIT 100 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

© Copyright 2000-2001, University of Washington

FIT 100 Problems with Flat Files and File-Processing Systems

- ❖ Data often duplicated
 - ❑ Many different pieces of data associated together in one file and then duplicated across files
- ❖ Data separated and isolated
 - ❑ Different programs had to be written to access each file, but you had to know where the files were
- ❖ Application program dependent
 - ❑ A change to the file format would break the program
- ❖ Incompatible data files
- ❖ Too easy to reach data and too hard to reach data
- ❖ Difficult to understand

© Copyright 2000-2001, University of Washington

FIT 100 Main Problem Is Redundant Data

<u>Name</u>	<u>Course</u>	<u>Room#</u>	<u>Instructor</u>
Paul Stevens	FIT 100	MGH 420	Whiteaker
Holly Eggeston	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

© Copyright 2000-2001, University of Washington

FIT 100 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

© Copyright 2000-2001, University of Washington

FIT 100 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 (on 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

© Copyright 2000-2001, University of Washington

FIT 100 But, before you can have a database or tables...

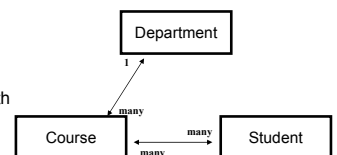
- ❖ You first have to know what you need tables of:
 - ❑ Why are you creating the system?
- ❖ The best way to do that is to get a picture/model of all the things that should become tables:
 - ❑ What do you want to keep track of? What is important?
- ❖ The world can be reduced to 2 things:
 - ❑ Things 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

© Copyright 2000-2001, University of Washington

FIT 100 How Do You Model Real World Relationships?

The Student Course Model

- ❖ There are Students with
 - ❑ Name
 - ❑ Address
- ❖ There are Courses with
 - ❑ Name
 - ❑ Number
- ❖ There are Departments with
 - ❑ Name
 - ❑ Location
- ❖ There are relationships
 - ❑ Each Course has one Department
 - ❑ Each Department has many Courses
 - ❑ One Student can have many Courses
 - ❑ One Course can have many Students



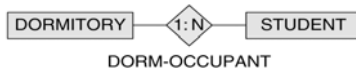
© Copyright 2000-2001, University of Washington

FIT 100 Kinds of Relationships

- ❖ One to One



- ❖ One to Many



- ❖ Many to Many



© Copyright 2000-2001, University of Washington

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

© Copyright 2000-2001, University of Washington

FIT 100 An Informal Table

<u>Name</u>	<u>Course</u>	<u>Room#</u>	<u>Instructor</u>
Paul Stevens	FIT 100	MGH 420	Whiteaker
Holly Eggeston	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

...

© Copyright 2000-2001, University of Washington

FIT 100 Relational Database Technology

- ❖ To normalize a table for use in a 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
 - + Primary key (PK)
 - + Primary key is never null and cannot be a duplicate

© Copyright 2000-2001, University of Washington

FIT 100

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
 - ❑ Candidate key fields are also often indexed if they are important for retrieval

© Copyright 2000-2001, University of Washington

FIT 100

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 relationship to be stored is 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
 - ❑ If the relationship to be stored is 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

© Copyright 2000-2001, University of Washington

FIT 100

The Keys Are the Key to Relational DB's

- ❖ Keys show how tables are related to one another

Student		Course		
SID	Name	CourseID	Name	Room
1	Paul Stevens	10	FIT 100	MGH 420
2	Holly Eggeston	20	LIS 540	EE1 045
3	Stephanie Wright	30	INFO 480	EE1 025
4	Lisa Spagnolo			
5	Pam Green			
6	Thomas Nguyen			

Class_Event			Instructor	
SID	CourseID	InstID	InstID	Name
3	20	200	100	Whiteaker
4	30	100	200	Boiko
1	10	100		
2	10	100		
4	30	200		

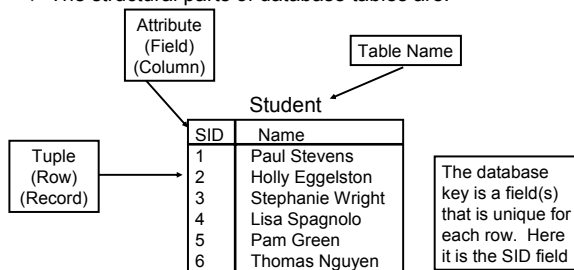
Note:
No Data Redundancies

© Copyright 2000-2001, University of Washington

FIT 100

Terms of a Database Table

- ❖ The structural parts of database tables are:



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

© Copyright 2000-2001, University of Washington

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	<i>student UW ID #</i>
FName	String	<i>student first name</i>
LName	String	<i>student last name</i>
Primary Key: SID		

© Copyright 2000-2001, University of Washington

FIT 100 Instances

- ❖ A database as the word is normally used: tables with specific contents
 - ❑ Known as a *database instance* (of a database schema)
 - ❑ Construction blueprints (a schema)
 - ❑ The actual home after construction (an instance)
- ❖ There can be many instances of a single database schema

Student Number	Student Name	Credits	Sex	Class	Major
0000000	LATA, POONAM	05.0	F	4	0-C-ANTH
0010000	LEE, BRADLEY J	05.0	M	6	0-A-N MATR
0020000	LIM, HENDRIK	05.0	M	3	0-C-ECON
0030000	MALANA, WELLA JOYCE CUNANAN	05.0	F	2	0-C-PREMAJ
0050000	NG, SAI-LAI	05.0	M	3	0-C-ECON

© Copyright 2000-2001, University of Washington

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.
 - ❑ Relationships among things are shown with keys

© Copyright 2000-2001, University of Washington