



Announcements

Please Turn in Project 2D
Project 3 will be posted this afternoon



Unannounced Quiz 3

Please stow all papers, books, phones, computers, pdas, etc.

Take out 1 sheet of paper, print your name, student ID and section #



Unannounced Quiz 3

Answer all questions. In databases...

1. "Tuple" is the same as (a) row, (b) record, (c) table, (d) attribute; *list all that apply*
2. "Field" is the same as (a) column, (b) row, (c) attribute, (d) table; *list all that apply*
3. Give the five fundamental operations on tables for relational databases



Thinking of Databases

Databases are organized on two levels: 'physical' is how the data is stored, 'logical' is how it's viewed



Avoiding Redundancy

Redundancy is bad because it can lead to inconsistent data ... very bad!

- Keep only one copy of any data ... does that make it right???
- Rather than repeating data, reference it in the places where it is needed
 - Keep data in its own table
 - Save its key wherever it is needed

When users want the data, get it using its key!



Physical Database

Physical databases store data in the "best" way -- no redundancy, ...

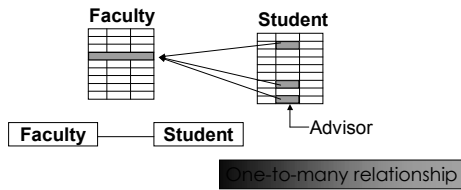
- Expect many tables of "simple" entities
- "Physical" means that the data is actually stored on the disk -- contrast with logical DBs that are "virtual tables"
- Physical databases are designed "for the computer" not for the user
- The "physical schema" gives table definitions and the relationships



Relationships

The table data entries are not just text & numbers, but they have meaning

- Relationships spell out that meaning



Kinds of Relationships

One-to-One



One-to-Many



Many-to-Many



Name relationships by their meaning



Logical Databases

Users want & need different information

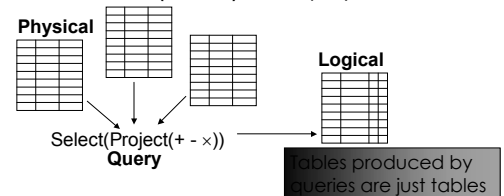
- Different tasks require different information
- Different authority levels, e.g. need to know
- Customizing to users means everyone sees exactly what they need to see

- * A view is a user's customized database
- * Views are virtual, built on-the-fly from the physical database and not kept
 - Data is always current
 - Custom structure can be very adaptable



Queries

Queries are commands (using the 5 table operations) that create logical database (views) from physical



SQL

The structured query language is the industry standard query language

"Structured" means the queries have a standard form

Common clauses --

```

SELECT <fields desired> ← like Project!
FROM <list of tables>
INNER JOIN <table> ON <conditions>
WHERE <criteria>

```

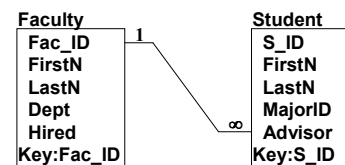
SQL is not case sensitive



Sample Database

Define a university DB schema

- ER Diagram **Faculty** — **Student**
- Specifying a 1-to-many relationship





Sample SQL Queries

Typical: `SELECT <attrs> FROM <tables>`

```
SELECT Student.FirstN, Student.LastN,
       Student.MajorID
FROM Student
WHERE Student.S_ID= 0206125;
```

```
SELECT Student.FirstN, Student.LastN
FROM Student
WHERE MajorID=14;
```



Join Example

Find the students of a given professor

```
SELECT Student.FirstN, Student.LastN,
       Faculty.LastN
FROM Student INNER JOIN Faculty
ON Student.Advisor = Faculty.Fac_ID
```

- Notice that selection comes from the combined (by Inner Join) table



DB Design Paradigm

Guidelines for good databases:

- Build physical DB to avoid redundancy, etc
- Each physical table represents 1 entity
- Expect that no physical table gives any user their exact view
- To build view, build a query that ...
Joins tables together into a 'super' table
Trims out only the items the user wants

These guidelines are not an algorithm,
but they usually produce good results



Project 3: SLAMA

Design DB to support a small mythical
WA town's police department ...

- "San Lucas Arrest Monitoring Application"
- The DB keeps track of suspects, police, and arrests. It monitors the blood alcohol and drug testing process while protecting the privacy of the citizens
- You will create the DB in MS Access, including tables, queries, forms and reports



Defining Tables

The physical database is a collection
of (3-4) tables

Field Name	Data Type	
Suspect ID	AutoNumber	unique identification number for suspects
first	Text	suspect's first name
middle	Text	suspect's middle name(s)
last	Text	suspect's last name
sex	Text	single character field M or F for male or female
birth	Date/Time	birthdate
street	Text	street address up to city
city	Text	city
state	Text	two letter state/province code; xx=not north
zip	Text	postal code



Filling Tables

Things are slow in San Lucas, so you
will have to generate data

badge#	rank	first	middle	last	street	city	state	zip	sex
22	Sergeant	Thomas	Edward	Richmont	2222 Mistena F	Bremerton	WA	98201	581-22-3353
233	Patrolman	Brenda	Lee	Harrison	4 Wheel Dr.	Bremerton	WA	98201	255-14-5441
234	Patrolman	Brendon	Leigh	Harris	2 Sans Road	Bremerton	WA	98202	033-13-1100

badge#	first	middle	last	sex	birth	street	city	state	zip
1	George	Herbert	Walker	M	2/6/1930	6 Yale Drive	New Haven	CT	06502
2	William	Clinton	Jefferson	M	6/30/1945	1230 E 13th St	Seattle	WA	98103
3	Barbara	Alex	Chance	F	9/16/1955	13 Surf Drive	Hollis	CA	90122
4	Albert	James	Austin	M	10/2/1961	32 Binary Street	Bremerton	WA	98201

arr. #	suspect badge#	arr. date	charge#	mirrands	mNotes	inventory	invenSign	breath	alqph	walk
2	3	2/27/2003	DUI	0		mg, brown belt	SR	Fail	Fail	Wooded
3	1	2/27/2003	Rowdy	0		Laptop computer	SR			



Defining Queries

The logical database (views) remove data from tables for users

```

qryBooking - Select Query
SELECT [tblArrest].[arr_ID], [tblOfficer].[first] AS tblOfficer_first, [tblOfficer].[last] AS
tblOfficer_last, [tblSuspect].[first] AS tblSuspect_first, [tblSuspect].[last] AS
tblSuspect_last, [tblArrest].[arr_date], [tblArrest].[charge], [tblArrest].[breath],
[tblArrest].[alpha], [tblArrest].[walk], [tblArrest].[stand], [tblArrest].[tothers],
[tblArrest].[alcohol], [tblArrest].[drug], [tblArrest].[Notes], [tblArrest].[tracking]
FROM tblSuspect INNER JOIN [tblOfficer] INNER JOIN tblArrest ON
[tblOfficer].[badge]=[tblArrest].[badge] ON
[tblSuspect].[suspect_ID]=[tblArrest].[suspectno];

```

We can write SQL directly, or we can use Query By Example



GUIs For The Views

The GUIs (forms) for users must be neat and attractive, with a slogan

Record: 1 of 3



Tracking Number

To protect privacy, lab samples are not coded with suspect's name

Record: 1 of 2

