

Operations on Databases



Tables are useful, but they become much more powerful when we can manipulate them to create new tables from existing tables. For that, we need special operations

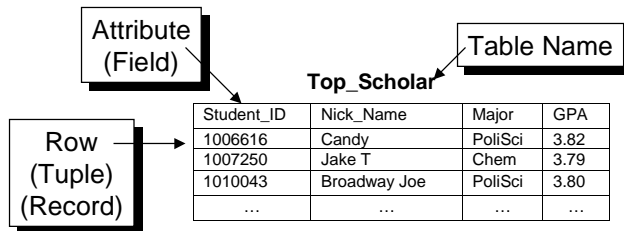
This entire lecture is concepts, implying that it should be entirely “green”.

© University of Washington, 2001



Recall database vocabulary ...

- ❖ A database is a set of tables
- ❖ The structure of a database is given by its schema
- ❖ An instance is the current contents of a database



- ❖ Fields have a type ... column items all have that type
- ❖ A database table stores information about entities

© University of Washington, 2001

**FIT
100**

Fun & Games with Schedule Schema

- ❖ A sports team's schedule follows a standard form

Schedule		
Game_No	Integer	Game sequence number
Opponent	String	The other team
Month	Integer	Month of game
Day	Integer	Day of game
Time	Time_Fmt	Start time
Home_Away	String	"Home" or "Away"
Key: Game_No		

The Game_No is not required as key, since Month:Day are a key

© University of Washington, 2001

**FIT
100**

Select

- ❖ Select picks a subset of records on some criterion
Select_from <table> **On** <T/F predicate>
- ❖ For example, to select the August home games

Select_from MsSchedule **On** Month=8 AND Home_Away = "Home"

Game_No	Opponent	Month	Day	Time	Home_Away
104	Chicago	8	1	01:30 PM	Home
105	Chicago	8	1	04:30 PM	Home
106	Toronto	8	11	01:30 PM	Home
107	Toronto	8	12	06:00 PM	Home
...	...	8	Home

SumVac = Select_from MsSchedule **On** Month=8 AND Home_Away = "Home"

© University of Washington, 2001

FIT 100

Project

- ❖ Project extracts columns from a table
Project <attribute_list> From <table>
- ❖ Picking out the essential information from SumVac

Project Opponent, Day, Time From SumVac

Game_No	Opponent	Month	Day	Time	Home_A
104	Chicago	8	1	01:30 PM	Home
105	Chicago	8	1	04:30 PM	Home

Opponent	Day	Time
Chicago	1	01:30 PM
Chicago	1	04:30 PM
Toronto	11	01:30 PM
Toronto	12	06:00 PM
...

MySchedule = Project Opponent, Day, Time From SumVac

FIT 100

Union

- ❖ Union combines two tables with like attributes
<table> + <table>

For example, include New York series in schedule

Opponent	Day	Time
Chicago	1	01:30 PM
Chicago	1	04:30 PM
Toronto	11	01:30 PM
Toronto	12	06:00 PM
...

Opponent	Day	Time
New York	6	01:30 PM
New York	7	04:30 PM
New York	8	01:30 PM
...

Opponent	Day	Time
Chicago	1	01:30 PM
Chicago	1	04:30 PM
Toronto	11	01:30 PM
Toronto	12	06:00 PM
New York	6	01:30 PM
New York	7	04:30 PM
New York	8	01:30 PM
...

MyExtraSchedule = MySchedule + NewYorkSeries

**FIT
100**

Difference

- ❖ Difference removes a table from a table with like attributes

$\langle \text{table} \rangle - \langle \text{table} \rangle$

If NightGames is table of Ms games after 5:00

$\text{MyDaySchedule} = \text{MySchedule} - \text{NightGames}$

Opponent	Day	Time
Chicago	1	01:30 PM
Chicago	1	04:30 PM
Toronto	11	01:30 PM
Toronto	12	06:00 PM
New York	6	01:30 PM
New York	7	04:30 PM
New York	8	01:30 PM
...

MySchedule

Opponent	Day	Time
Chicago	1	01:30 PM
Chicago	1	04:30 PM
Toronto	11	01:30 PM
New York	6	01:30 PM
New York	7	04:30 PM
New York	8	01:30 PM
...

MyDaySchedule

© University of Washington, 2001

**FIT
100**

Product

- ❖ For each row in first table, concatenate every row in second table

$\langle \text{table} \rangle \times \langle \text{table} \rangle$

- ❖ Product can be used to add new fields ...

Outcome	Won_Loss	String	<i>The result of the game</i>
---------	----------	--------	-------------------------------

- ❖ MySchedule x Outcome

Product's Column Rule:
If tableA has m columns and
tableB has n columns, then
tableA x tableB has m+n columns

Opponent	Day	Time	Won_Loss
Chicago	1	01:30 PM	
Chicago	1	04:30 PM	
Toronto	11	01:30 PM	
Toronto	12	06:00 PM	
New York	6	01:30 PM	
New York	7	04:30 PM	
New York	8	01:30 PM	
...	

© University of Washington, 2001

**FIT
100**

More Products

- ❖ Product creates a table of “all pairs”
- ❖ For example, let Spring be spring training schedule and Player be

Player

Name	String	Player name
Position	String	Primary position played
At_Bats	Integer	Number of times to bat in a game
Hits	Integer	Number of hits
Strike_Outs	Integer	Number of times struck out

Spring x Player has a row for each player in each game

Product's Row Rule:
If tableA has m rows and
tableB has n rows, then
tableA x tableB has mn rows

© University of Washington, 2001

**FIT
100**

The Rules Always Apply

- ❖ Visualizing a product ...



- ❖ The row and column rules always apply

Product's Column Rule:
If tableA has m columns and
tableB has n columns, then
tableA x tableB has m+n columns

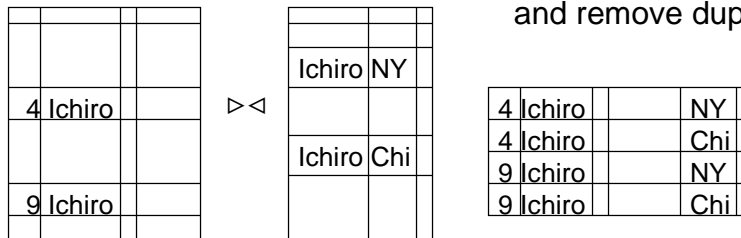
Product's Row Rule:
If tableA has m rows and
tableB has n rows, then
tableA x tableB has mn rows

© University of Washington, 2001

FIT 100

Join -- Product with a Match

- ❖ Join is a combination operation performing a product and a select
- ❖ The Join symbol is a “bow tie” \bowtie
- ❖ Natural Join ... suppose two tables have a the same attribute, then pair all rows of the two tables, but keep only those rows that match on the common attribute and remove duplicate



FIT 100

Other Joins

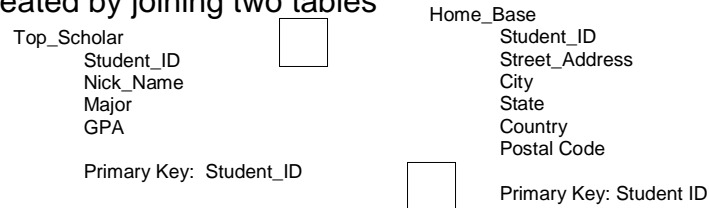
- ❖ Besides matching on the common attribute, other operations are allowed too: $>$, $>=$, $<$, $<=$, $<>$
- ❖ Join is very useful because it allows us to construct more complete database views from small tables

Permanent Address			Student_ID	Nick_Name	Dean Data	GPA
Student_ID	Street_Address	City	1006616	Candy		3.82
0998185	1 Rocky Rd	Burnaby	1007250	Jake T		3.79
1006616	61 Peekaboo St	Vail	1010043	Broadway Joe	PoliSci	3.80
1006681	1011 Passing Ln	Ames
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
...

Constructing Dean's View

- ❖ The Dean's View, which doesn't literally exist, is created by joining two tables



- ❖ The common field is student ID, so ...
- ❖ Top_Scholar \bowtie Home_base creates the result

Summary

- ❖ The five basic operations on tables are
 - ❑ Select -- pick out rows
 - ❑ Project -- pick out columns
 - ❑ Union -- combine two tables with like attributes
 - ❑ Difference -- remove items from one table based on another
 - ❑ Product -- form new table with all pairs from two tables
- ❖ Join is a powerful operation created from project/select
- ❖ Database operations allow the DB to be set up to avoid redundancy, but allow the data to be exhibited to users in whatever form they want