

Discussion 10:

SQL

TA: **Jerry Chen**
Email: **jerry.c@berkeley.edu**
TA Website: **jerryjrchen.com/cs61a**

Agenda

1. Attendance
2. Announcements
3. SQL

Attendance

Sign in at bit.do/jerrydisc

OR

Come to me for check-in

Announcements

Scheme proj due tonight!

Hw 13 is due next Wed

No section, no in-person lab next week

Last section will be a little bit different...

Discussion 11 Preview



A problem has been detected and Windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

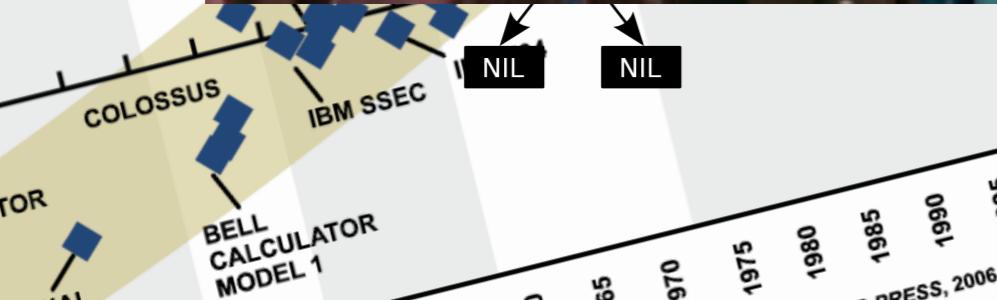
Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)

*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, Datestamp 3d6dd67c



Databases



Databases

Data — information about pretty much anything

A **database** is an ordered collection of data

Use **tables** to organize data

Databases show up everywhere!



SQL

Structured Query Language

(Pronounced "Ess Cue El" or "Sequel")

Used to manage data stored in a database

A **declarative** language — broadly speaking, tell it **what we want**, not how to do it

All "queries" (expressions) end in a semicolon ";"

SQL

The **select** statement creates tables

- Use the **union** command to join two select rows

The **create table** expression saves a table for later

Select

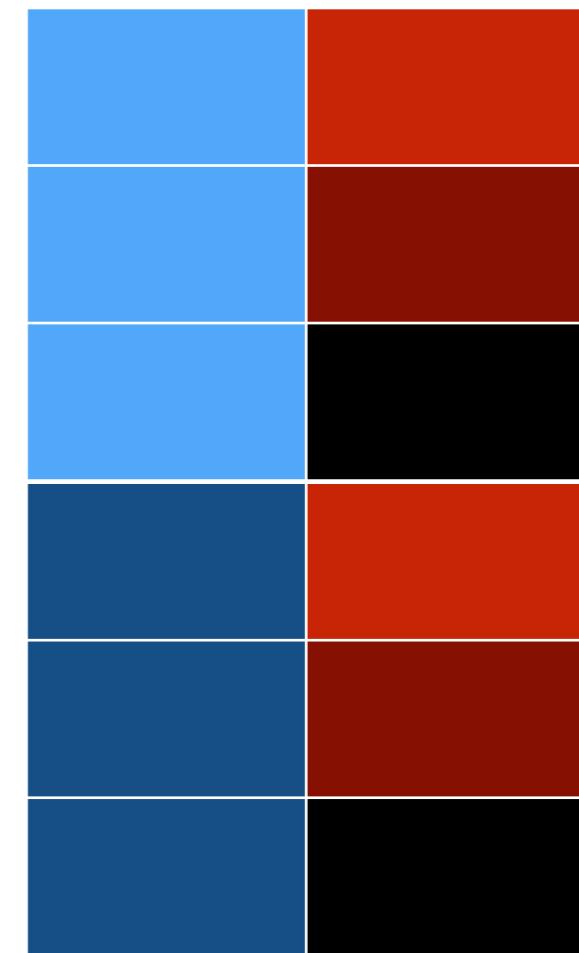
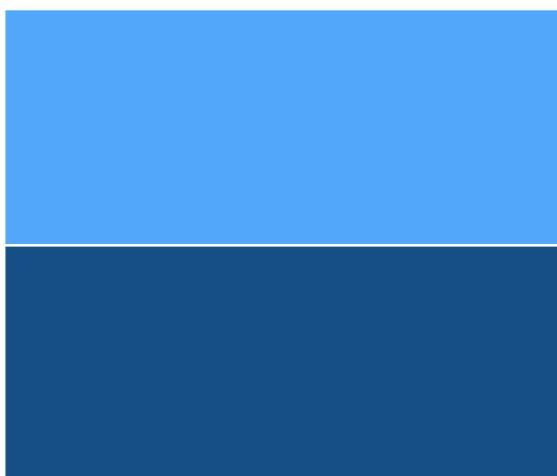
select doesn't have to start from scratch: can select
from an existing table to create a new one

Specify what columns to keep in your result!

Filter results using boolean expressions in the
where clause

Joins

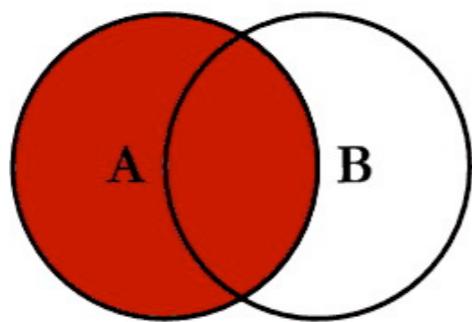
When we join two tables together, consider all possible pairings:



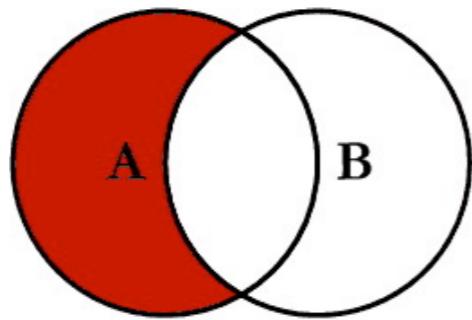
Joins

Of course, it gets more complicated (out of scope)

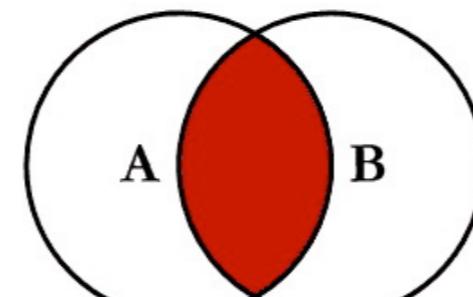
SQL JOINS



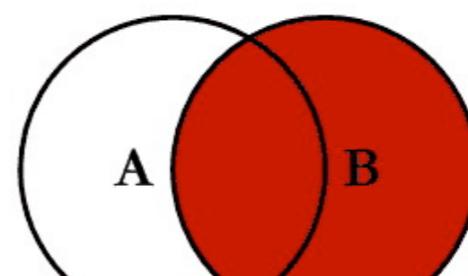
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



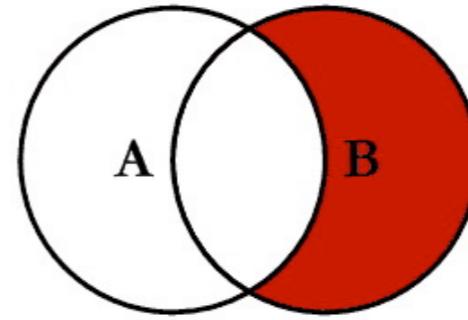
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



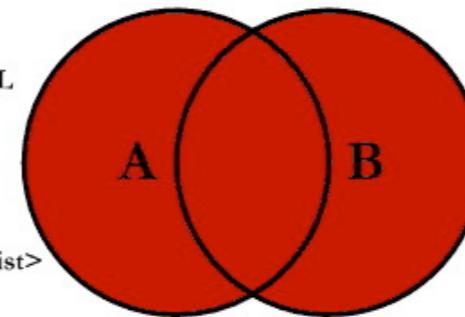
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



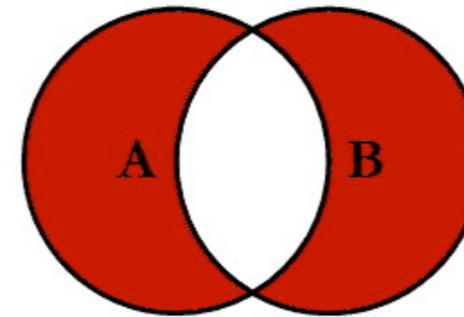
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

Joins

If we're joining with ourself (or a table with the same column names), we may require **aliasing**

Be wary of **duplicates** and **self-joins** (row joined to itself)!

- Solve by **enforcing ordering**

Recursive Select

Like regular recursion start off with a **base row** (base case)

Subsequent rows based off of previous rules
(recursive step)

Use filter (where) to determine when to stop

We usually use a local **with** table to create recursive tables