CS50 - SECTION 8

11/06/18

POST-MORTEM ON PSET 7

QUICK REMINDERS

1. What's the difference between the client-side and server-side?

QUICK REMINDERS

1. What's the difference between the client-side and server-side?

Server-side deals with code run on the server, whereas client-side refers to code downloaded and run locally on the user's computer.

QUICK REMINDERS

1. What's the difference between the client-side and server-side?

For the purposes of validation, we used JS in PS7 to achieve client-side validation. JS wouldn't even let the user submit the form if they had entered incorrect form input.

You used Flask and Python to achieve client-side validation. An error was returned to the user if they submitted incorrect form input.

QUICK REMINDERS

1. What's the difference between the client-side and server-side?

Client-side validation is about immediate user feedback, whereas server-side validation is about data integrity.

QUICK REMINDERS

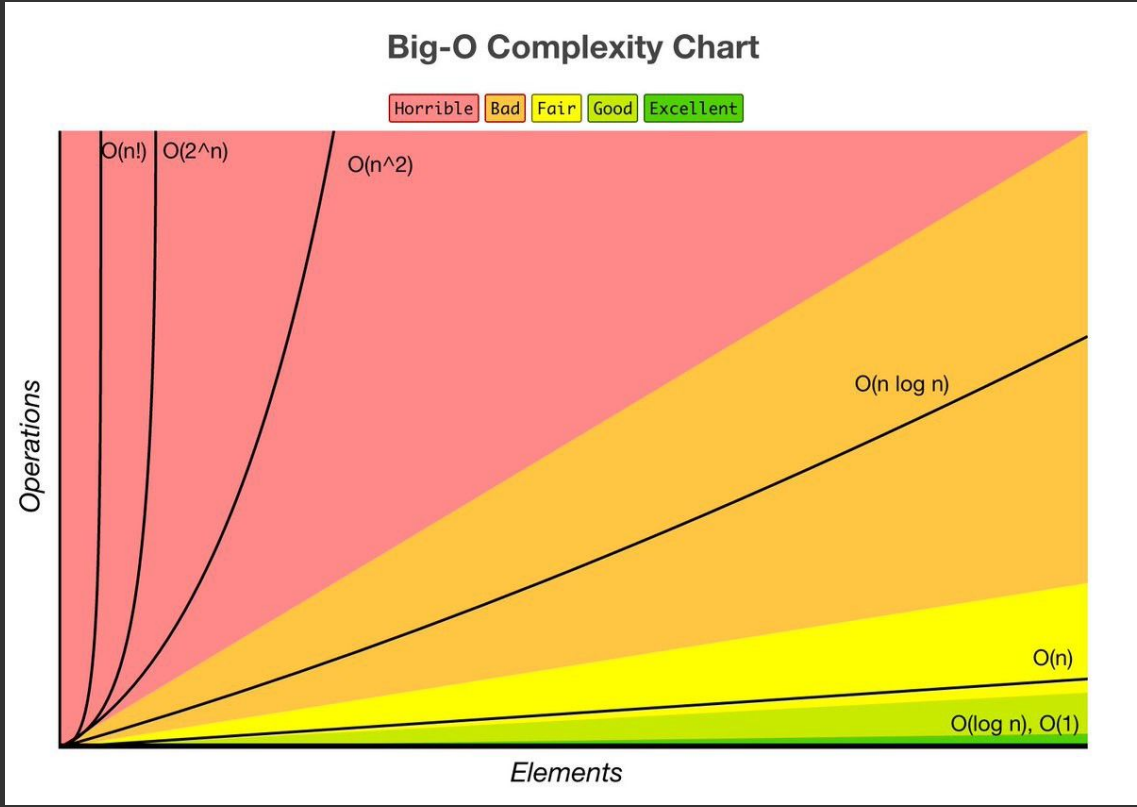
2. One quick way to improve the design of your programs is to identify common “bottlenecks” in processing speed.

QUICK REMINDERS

2. One quick way to improve the design of your programs is to identify common “bottlenecks” in processing speed.

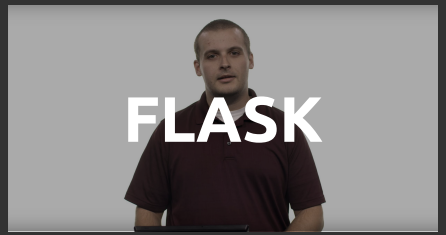
For example, in PS7, there were multiple ways to iterate over the `lines` of the input. It was possible to do this with one for loop instead of nested for loops. Nested for loops dramatically increase the running time of our program.

QUICK REMINDERS



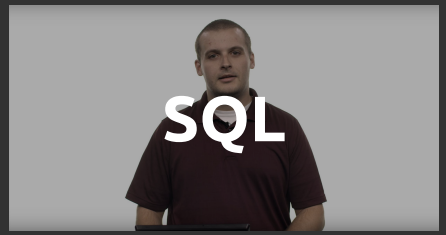
CONCEPTS DEEP-DIVE

“SHORTS” FOR THE WEEK



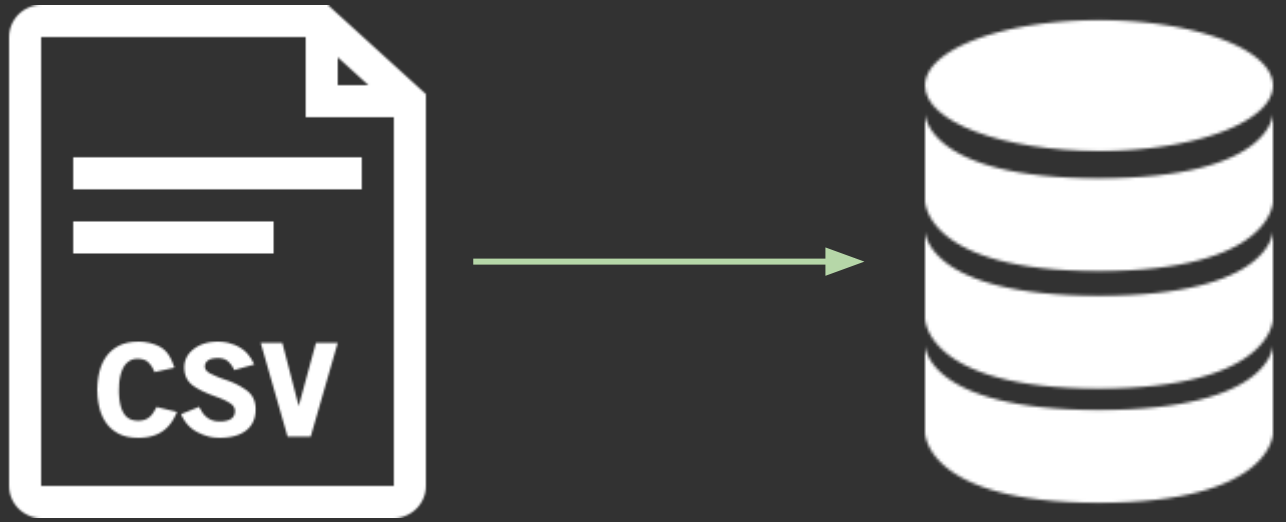
https://youtu.be/jOKx1JkRlho

https://youtu.be/xgyc_wOQt2Y

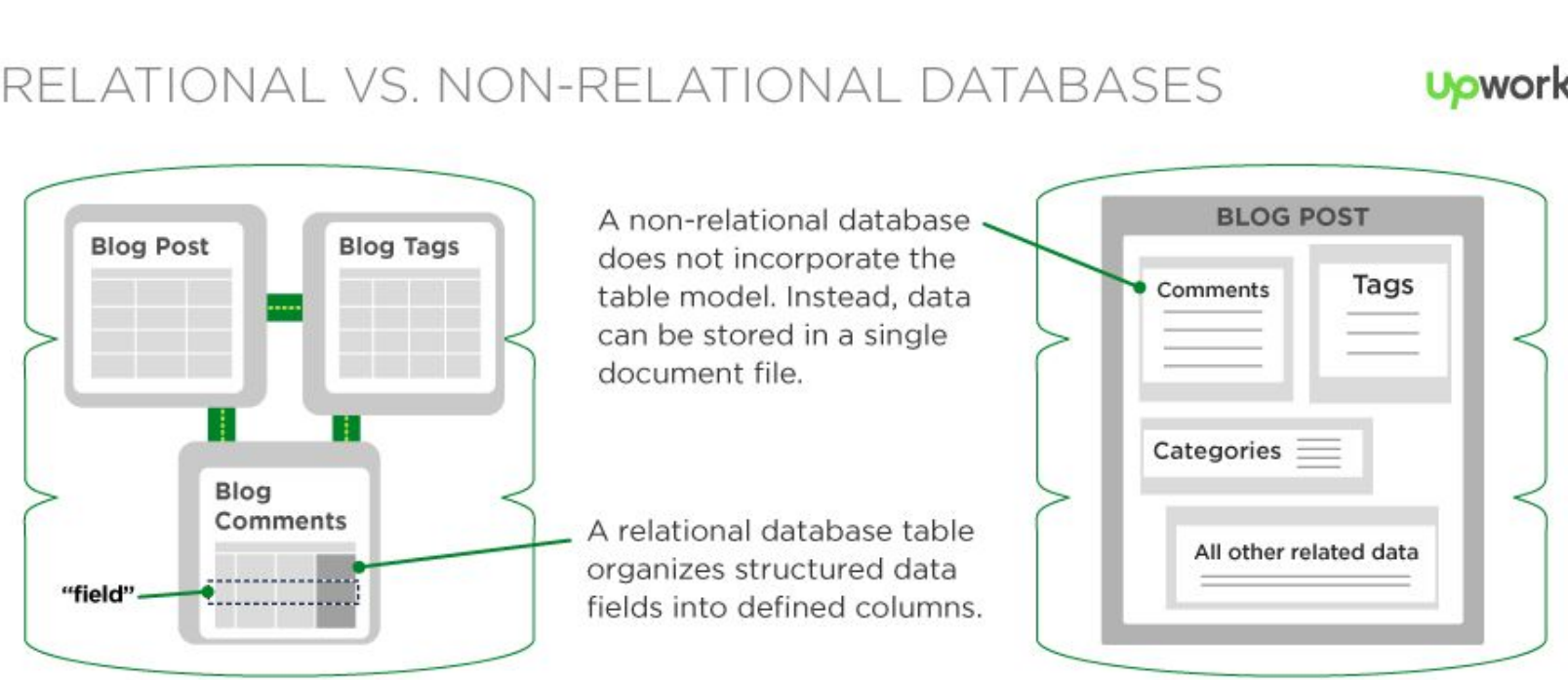


https://youtu.be/nfGiGSCEYRI

MOVING BEYOND THE CSV FILE



MOVING BEYOND THE CSV FILE



MOVING BEYOND THE CSV FILE

SQL

DATABASE DESIGN PRINCIPLES

- Why do we use types when designing our databases?

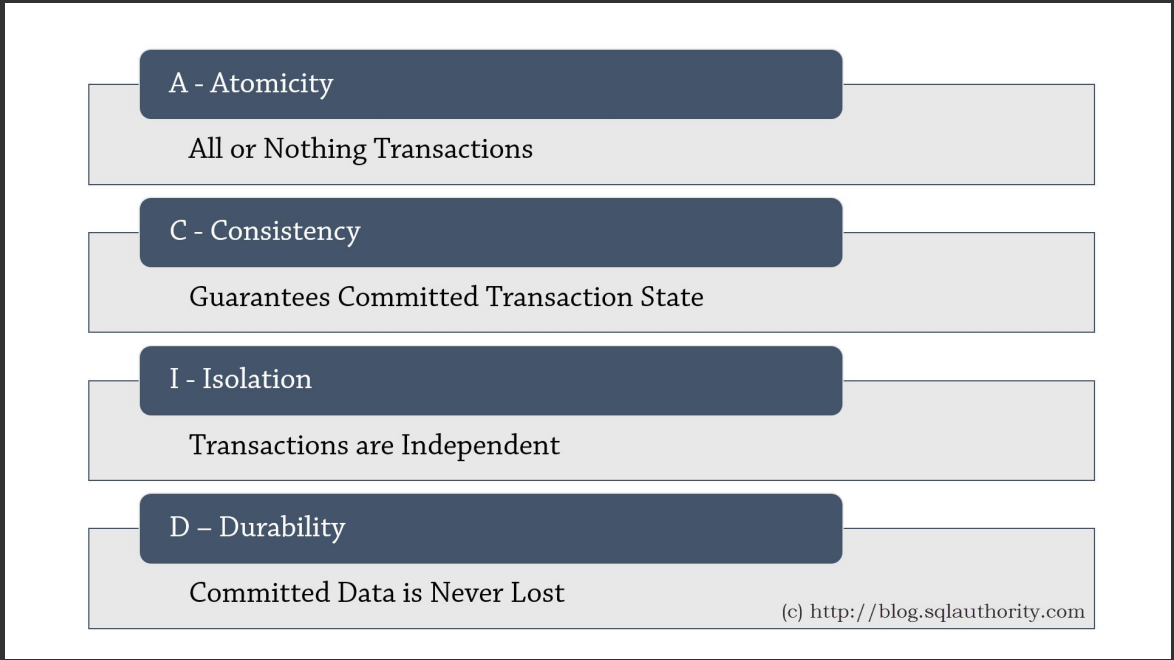
DATABASE DESIGN PRINCIPLES

- Why do we use types when designing our databases?

They allow us to increase the efficiency of data storage/retrieval AND they improve data integrity.

DATABASE DESIGN PRINCIPLES

- Why do we use types when designing our databases?



RELATIONAL DATABASE TERMINOLOGY

- **Database -** One discrete data system; The largest structure we'll be working with

RELATIONAL DATABASE TERMINOLOGY

- **Database -** One discrete data system; The largest structure we'll be working with
- **Table -** A set of data contained inside a database; Consists of columns and rows

RELATIONAL DATABASE TERMINOLOGY

- **Database -** One discrete data system; The largest structure we'll be working with
- **Table -** A set of data contained inside a database; Consists of columns and rows
- **Record -** One individual set of fields in the database (i.e. a single row)

RELATIONAL DATABASE TERMINOLOGY

- **Database -** One discrete data system; The largest structure we'll be working with
- **Table -** A set of data contained inside a database; Consists of columns and rows
- **Record -** One individual set of fields in the database (i.e. a single row)
- **Key -** A field in a table used to retrieve records (*more later*)

RELATIONAL DATABASE TERMINOLOGY

- **Database -** One discrete data system; The largest structure we'll be working with
- **Table -** A set of data contained inside a database; Consists of columns and rows
- **Record -** One individual set of fields in the database (i.e. a single row)
- **Key -** A field in a table used to retrieve records (*more later*)
- **Schema -** A representation of the underlying structure behind the database; Serves as a template of the database

KEYS AND SQL

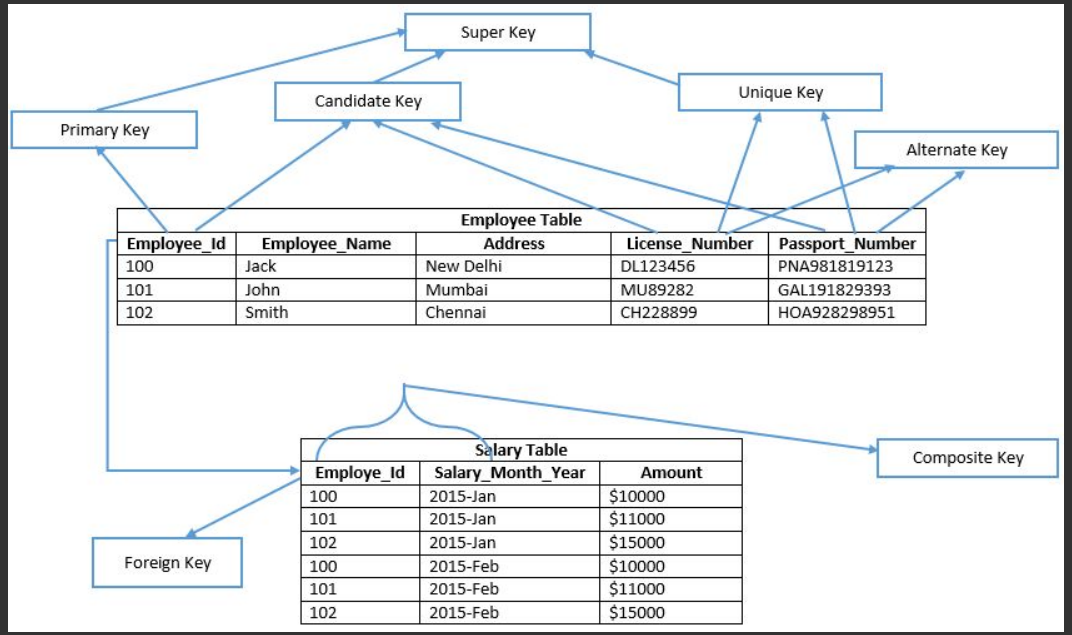
- Keys are just a way to retrieve records from our tables
- They should be unique *within* the table
  - Example: You have an `employee` table. What types of keys could you use to identify your records?

KEYS AND SQL

- Keys are just a way to retrieve records from our tables
- They should be unique *within* the table
  - Example: You have an `employee` table. What types of keys could you use to identify your records?
    - **An employee ID**
    - **Driver's license number**
    - **A full legal name (possibly—often causes conflicts!)**

KEYS AND SQL

- There are many different types of keys (we use **primary keys** in CS50):



Employee Table

Employee_Id	Employee_Name	Address	License_Number	Passport_Number
100	Jack	New Delhi	DL123456	PNA981819123
101	John	Mumbai	MU89282	GAL191829393
102	Smith	Chennai	CH228899	HOA928298951

Salary Table

Employe_Id	Salary_Month_Year	Amount
100	2015-Jan	$10000
101	2015-Jan	$11000
102	2015-Jan	$15000
100	2015-Feb	$10000
101	2015-Feb	$11000
102	2015-Feb	$15000

STRUCTURED QUERY LANGUAGE (SQL)

- We'll be working with SQL using the **CRUD** paradigm:
  - **Create** new databases, tables, and records
  - **Read** existing records from the database
  - **Update** records and database schema
  - **Delete** records or database schema

STRUCTURED QUERY LANGUAGE (SQL)

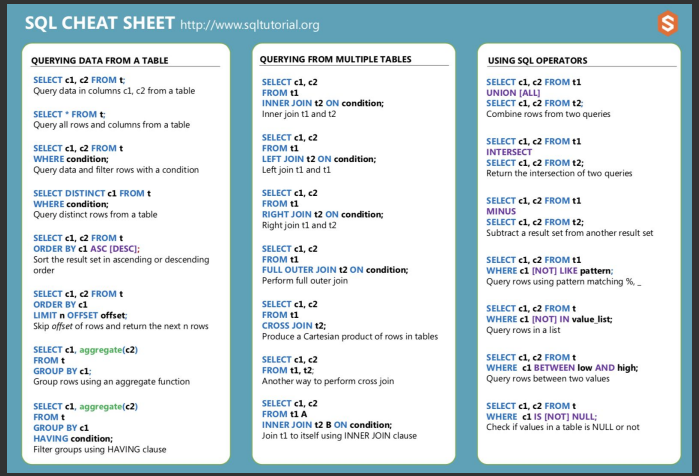
- The primary commands you'll be using in CS50 for PS8:
  - **UPDATE**: Updates records and database schema
  - **INSERT INTO**: Inserts values into a table
  - **SELECT**: Selects values from a table
  - **DELETE**: Deletes records from a table

STRUCTURED QUERY LANGUAGE (SQL)

- SQL is the underlying language that standardizes how we run statements to implement the CRUD paradigm
  - There exists many different **database engines** that allow us to execute SQL
  - We utilize SQLite in CS50, but an extremely popular alternative is MySQL
- We also have access to a tool called `phpliteadmin` to give us a graphical user interface (GUI) for running different SQL commands

STRUCTURED QUERY LANGUAGE (SQL)

SQL Cheat Sheet



SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

SELECT c1, c2 FROM t;
Query data in columns c1, c2 from a table

SELECT * FROM t;
Query all rows and columns from a table

SELECT c1, c2 FROM t
WHERE condition;
Query data and filter rows with a condition

SELECT DISTINCT c1 FROM t
WHERE condition;
Query distinct rows from a table

SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
Sort the result set in ascending or descending order

SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
Skip *offset* of rows and return the next n rows

SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
Group rows using an aggregate function

SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
Inner join t1 and t2

SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
Left join t1 and t1

SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
Right join t1 and t2

SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
Perform full outer join

SELECT c1, c2
FROM t1
CROSS JOIN t2;
Produce a Cartesian product of rows in tables

SELECT c1, c2
FROM t1, t2;
Another way to perform cross join

SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
Combine rows from two queries

SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
Return the intersection of two queries

SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
Subtract a result set from another result set

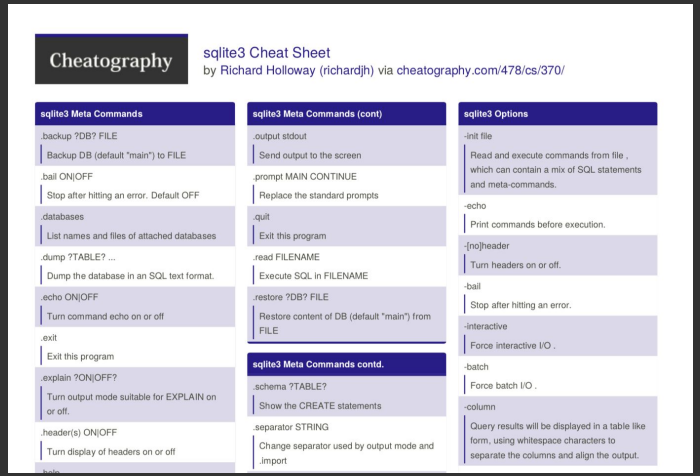
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
Query rows using pattern matching %, _

SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
Query rows in a list

SELECT c1, c2 FROM t
WHERE c1 BETWEEN low AND high;
Query rows between two values

SELECT c1, c2 FROM t
WHERE c1 IS [NOT] NULL;
Check if values in a table is NULL or not

sqlite3 Cheat Sheet



Cheatography

sqlite3 Cheat Sheet
by Richard Holloway (richardjh) via cheatography.com/478/cs/370/

sqlite3 Meta Commands

.backup ?DB? FILE
Backup DB (default "main") to FILE

.bail ON|OFF
Stop after hitting an error. Default OFF

.databases
List names and files of attached databases

.dump ?TABLE? ...
Dump the database in an SQL text format.

.echo ON|OFF
Turn command echo on or off

.exit
Exit this program

.explain ?ON|OFF?
Turn output mode suitable for EXPLAIN on or off.

.header(s) ON|OFF
Turn display of headers on or off

sqlite3 Meta Commands (cont)

.output stdout
Send output to the screen

.prompt MAIN CONTINUE
Replace the standard prompts

.quit
Exit this program

.read FILENAME
Execute SQL in FILENAME

.restore ?DB? FILE
Restore content of DB (default "main") from FILE

sqlite3 Meta Commands contd.

.schema ?TABLE?
Show the CREATE statements

.separator STRING
Change separator used by output mode and .import

sqlite3 Options

-init file
Read and execute commands from file , which can contain a mix of SQL statements and meta-commands.

-echo
Print commands before execution.

-[no]header
Turn headers on or off.

-bail
Stop after hitting an error.

-interactive
Force interactive I/O .

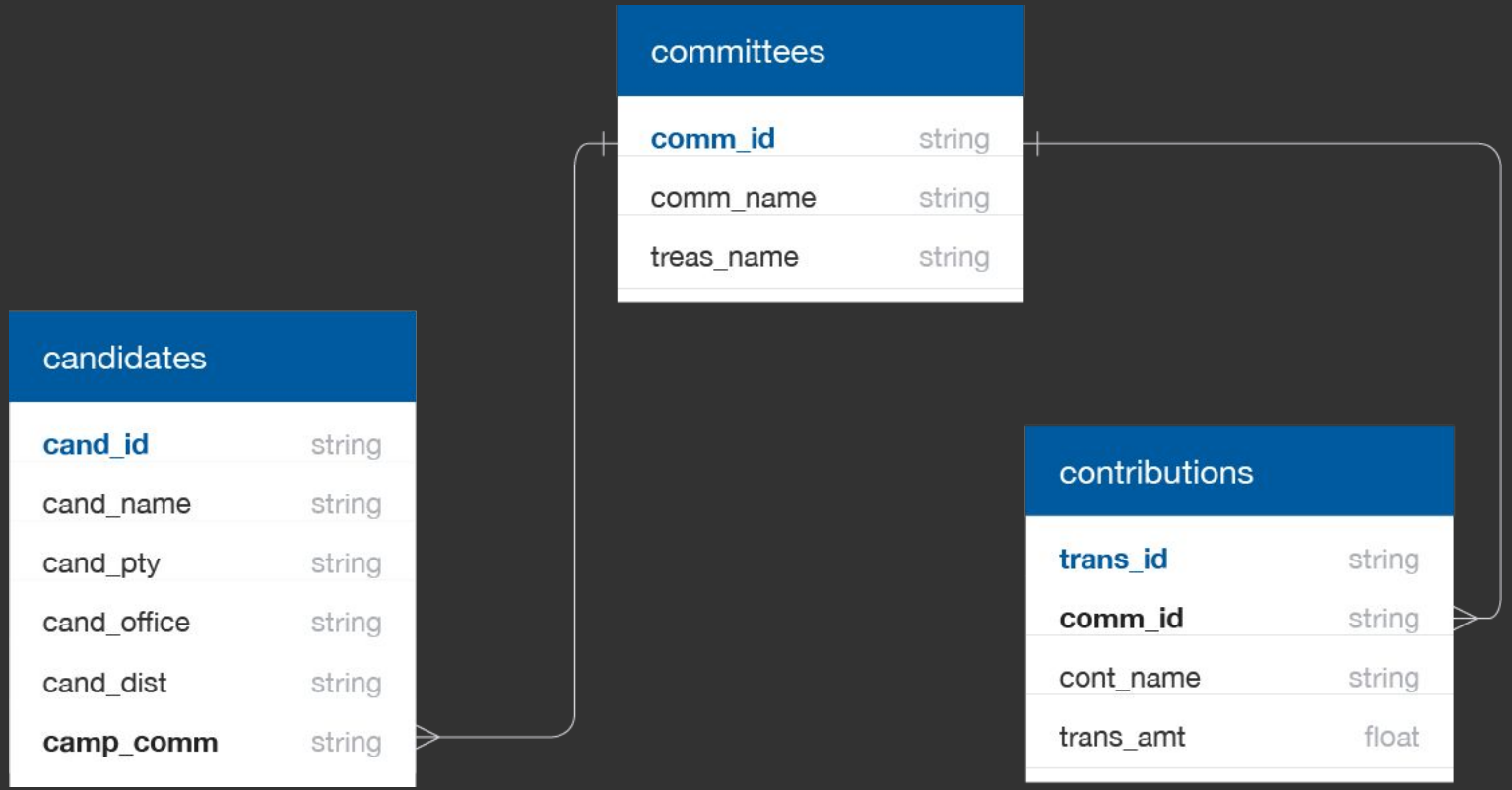
-batch
Force batch I/O .

-column
Query results will be displayed in a table like form, using whitespace characters to separate the columns and align the output.

COLLABORATIVE DEMO: CAMPAIGN FINANCE

http://bit.ly/2F5NpsN

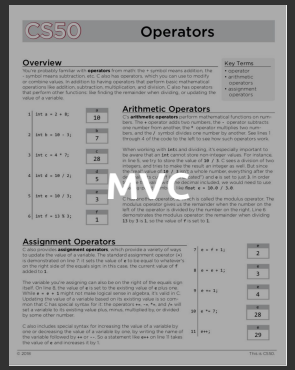
COLLABORATIVE DEMO: CAMPAIGN FINANCE



COLLABORATIVE DEMO: CAMPAIGN FINANCE SOLUTIONS

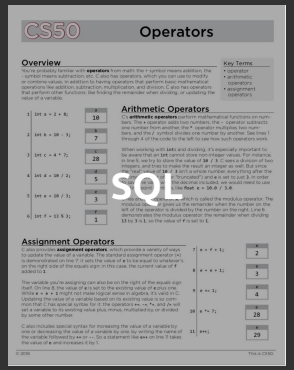
http://bit.ly/2APFJqw

REFERENCE SHEETS



MVC

https://www.dropbox.com/sh/5y662ey1hc4sde4/AACcfWq8pwZ_PkfN_AekbkXa/MVC.pdf?dl=0



SQL

https://www.dropbox.com/sh/5y662ey1hc4sde4/AACjqHN3NtSKk4ShsRDFd_Sja?dl=0&m=&preview=SQL.pdf

QUIZ REVIEW

NOTES ON THE QUIZ

- The Quiz will be released at noon this Saturday and due by **noon on Thursday 11/15**
- The Quiz takes most people 6-10 hours
  - It can be helpful to split up your working time into different large blocks
- It is very *conceptually-focused*—Unlike problem sets, you're not trying to write code or implement an algorithm
  - The focus is on the broad concepts of the class and what we've learned so far

TIPS FOR THE QUIZ

- Review lecture notes, my slides, reference sheets, and other conceptually-focused materials
- We've spent a lot of time discussing context and *why* in section—Apply that mindset to the quiz
  - Ask: Why do we use a certain algorithm? What role does this play in my code? Is this syntax or convention?
- Work out the Quiz problems on paper and type them up after the fact
  - Remember our whiteboarding strategy
  - If you're trying to come up with an algorithm, run through examples, **edge cases**, etc.

STRATEGIES

- **Between now and Saturday:**
  - Locate all the content materials you have and organize them
  - Review About the Quiz on CS50's webpage
  - Check out released problems (linked above)
- **Between Saturday and Tuesday:**
  - Try to complete an entire attempt of the Quiz—Skip problems if you get stuck and give yourself a lot of time to think through them
- **Between Tuesday and Thursday:**
  - Review your answers, check for mistakes/edge cases, and perfect your wording
  - Submit!

QUIZ REVIEW

We’ll now whatever questions you might have and spend time reviewing together.

REFLECTIONS

GOALS OF THE COURSE

1. Learn to think computationally and algorithmically.

GOALS OF THE COURSE

1. Learn to think computationally and algorithmically.
2. Improve your problem-solving skills—this will help you in every other course and in real-life too!

GOALS OF THE COURSE

1. Learn to think computationally and algorithmically.
2. Improve your problem-solving skills—this will help you in every other course and in real-life too!
3. Gain practical programming skills in a plethora of languages (C, Python, JavaScript, etc.)

GOALS OF THE COURSE

1. Learn to think computationally and algorithmically.
2. Improve your problem-solving skills—this will help you in every other course and in real-life too!
3. Gain practical programming skills in a plethora of languages (C, Python, JavaScript, etc.)
4. Familiarize yourself with best practices for software design.

GOALS OF THE COURSE

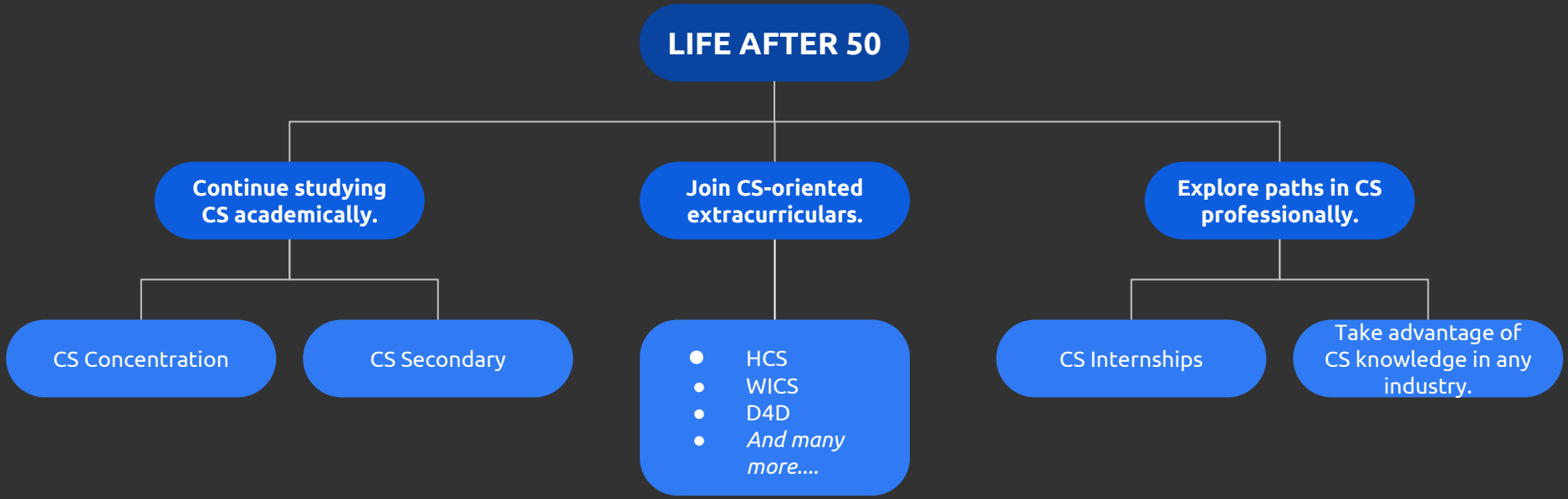
1. Learn to think computationally and algorithmically.
2. Improve your problem-solving skills—this will help you in every other course and in real-life too!
3. Gain practical programming skills in a plethora of languages (C, Python, JavaScript, etc.)
4. Familiarize yourself with best practices for software design.
5. Build a community with your peers that will stay with you after this course.

LIFE AFTER 50

WHAT REMAINS:

1. Finish `C$50 Finance`.
2. Complete the Quiz.
3. Complete your Final Project.
4. Attend the CS50 Fair.
5. Enjoy life after 50!

NEXT STEPS


LIFE AFTER 50
Continue studying CS academically.
Join CS-oriented extracurriculars.
Explore paths in CS professionally.
CS Concentration
CS Secondary
HCS
WICS
D4D
And many more....
CS Internships
Take advantage of CS knowledge in any industry.

NEXT STEPS—EXPLORING CS ACADEMICALLY

All the info you need and more at:

https://harvardcs.info/index.html

NEXT STEPS—CS CONCENTRATION

Category	Number of required courses	Basic Concentration	Honors Concentration
Mathematics	(Total: 2-4 courses)		
Preparation	0-2	Math 1a and/or Math 1b as needed	same
Linear algebra	1	Math 21b/22b/23a/25a/55a	same
Probability/statistics or Multivariable calculus	1	Stat 110 or Math 21a/22a/23b/23c/25b/55b	same
Basic Software	2	2 out of CS 50, CS 51, CS 61	same
Theory	2	CS 121 + CS 124 or other theory course	same
Technical Electives	4 (6 for honors)	4 courses from list that includes CS 20, CS 51-299, STAT 110, AM 106, AM 107, and others. (at most one CS 91r)	6 courses from same list
Breadth Requirement		**2** tech electives must be CS with penultimate digit 3-8	**3** tech electives must be CS with penultimate digit 3-8
Tutorial		At most one CS91r is tech elective	same
Thesis		Not required	Not required but recommended

NEXT STEPS—CS SECONDARY

- Requires four courses with numbers 100 or greater—CS50, CS51, and CS61 also count
- **You're ¼ the way there!**

NEXT STEPS—SPRING 2019

Consider taking **CS51: Introduction to Computer Science II**

SEMINAR—AN INTRODUCTION TO JQUERY

Time: Thursday, October 8, 4:30-5:30pm
Location: 67 Mt. Auburn St., the HSA building on the 4th floor

jQuery is helpful to solve the problem of writing client-side JavaScript code for web applications. It simplifies cumbersome JS, adds a variety of powerful dynamic features (e.g. animations), and is an industry-standard tool.

A FINAL NOTE

It's been a privilege to be your Teaching Fellow this semester.

I am a resource for you all now and *after* CS50:

- wadesilvestro@college.harvard.edu
- (863) 666-4003
- GET /wes HTTP/1.1

THANKS FOR A SPECTACULAR SECTION EXPERIENCE!