Analysis and Normal Forms 1

Lecture 8

Wednesday - Sep 27, 2023

Housekeeping

1. No homework dues this week!
2. Project deliverable 4 - posted and available
3. Quiz and quiz stats - Very nice!
4. Peer reviews - still some issues.
5. Changes to schedule (Version 4, see table below)
6. Discuss Quarto

Module	Week	Date	Day	Lectures/Quizzes	Deliverables/Notes
Normal forms	6	9/27	Wed	MTG10: L8 (Analysis and Normal Forms 1)	
Normal forms	7	10/2	Mon	MTG11: L9 (Analysis and Normal Forms 2)	
Normal forms	7	10/4	Wed	MTG12: L10 (Analysis and Normal Forms 3)	
Normal forms	8	10/9	Mon	MTG13: L11 (Quiz review session)	
Normal	8	10/11	Wed	MTG14: Quiz 3 today (Analysis and	

Housekeeping

Project deliverable 4

Project - deliverable 4 - Phase 1 submission (due week 8)

Due Oct 15 by 11:59pm **Points** 50 **Available** until Oct 19 at 11:59pm

1

Semester Project
Deliverable 4 - Database Design
Last updated: 9/26/2023

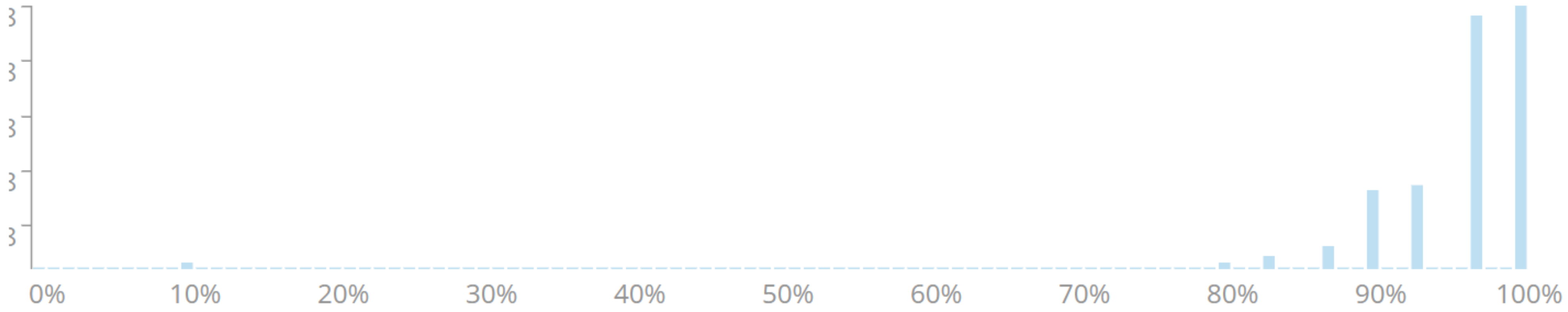
Each team will prepare a report and video documenting their domain, challenges, and proposed database solution. The quarto document will be rendered to a HTML document within a supplied GITHUB repository.

Quiz and quiz stats

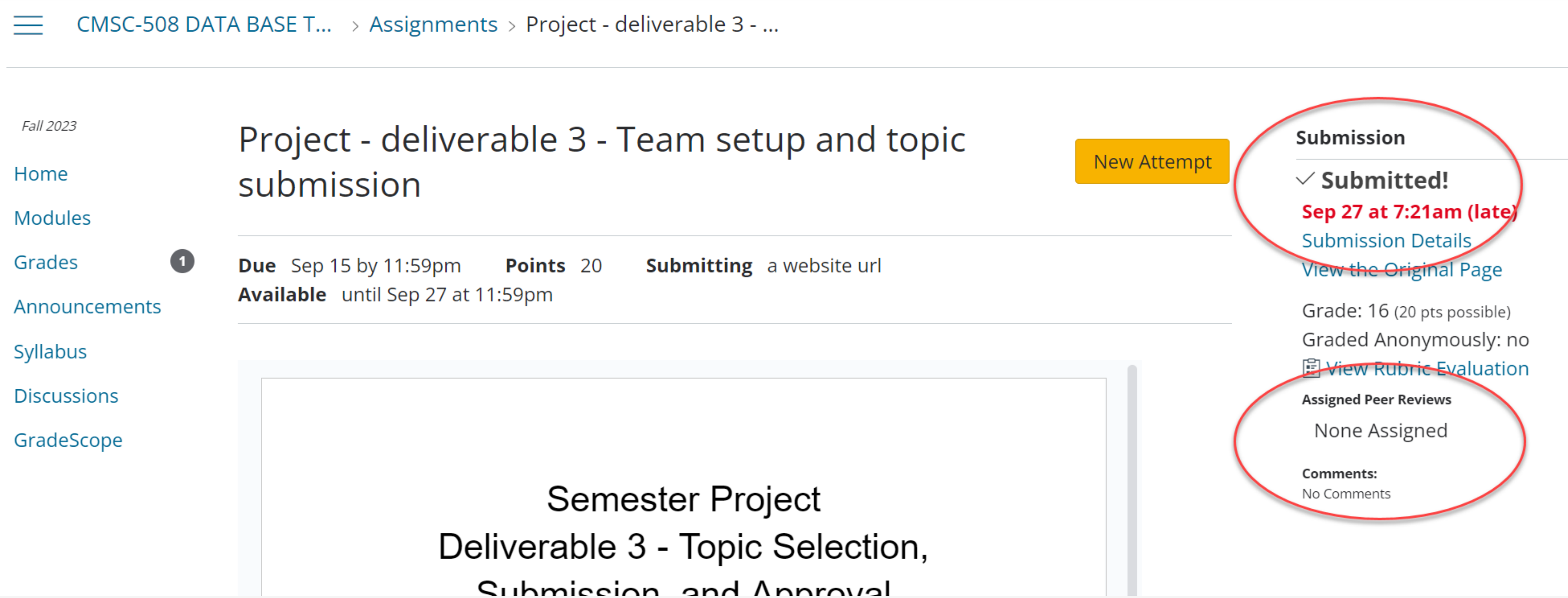
Quiz Summary

Section Filter ▾ | Student Analysis | Item Analysis

Average Score	High Score	Low Score	Standard Deviation	Average Time
95%	100%	10%	2.58	16:35



Peer reviews



Changes to the schedule

CMSC-508 DATA BASE THEORY > Pages > Intro - Course Schedule

es
s 1
ncements
us
sions
Scope

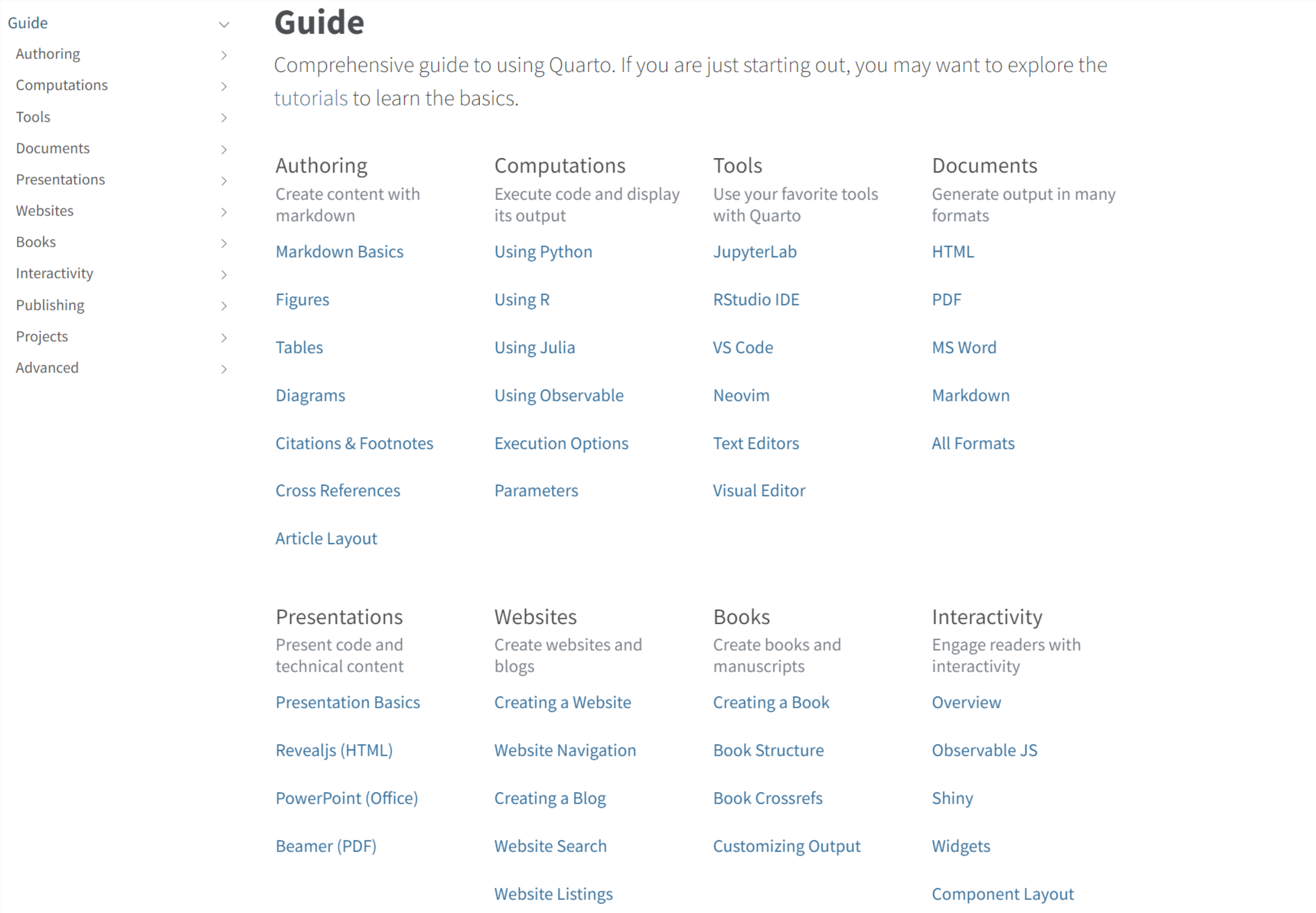
Intro - Course Schedule

CMSC508-FA2023-Calendar

ER Models	3	9/4	Mon		University closed: Labor day
ER Models	3	9/6	Wed	MTG4: Quiz 1 today (Entity-relation models)	
Relational Alg.	4	9/11	Mon	MTG5: L4 (DDL / DML / SQLLite / MySQL)	
Relational Alg.	4	9/12	Tue		HW3 due (ER Models 2 - extended)
Relational Alg.	4	9/13	Wed	MTG6: L5 (Relational models)	
Relational Alg.	4	9/15	Fri		PRJ3 due (Topic proposal video)
Relational Alg.	5	9/18	Mon	MTG7: L6 (Relational Algebra 1)	
Relational Alg.	5	9/20	Wed	MTG8: L7 (Relational Algebra 2)	
Relational Alg.	5	9/24	Sun		HW4 due (Relational Algebra Study Guide)
Normal forms	6	9/25	Mon	MTG9: Quiz 2 today (Relational Algebra)	
Normal forms	6	9/27	Wed	MTG10: L8 (Analysis and Normal Forms 1)	
Normal forms	7	10/2	Mon	MTG11: L9 (Analysis and Normal Forms 2)	
Normal forms	7	10/4	Wed	MTG12: L10 (Analysis and Normal Forms 3)	
Normal forms	7	10/8	Sun		HW5 due (Analysis and Normal Forms)
Normal forms	8	10/9	Mon	MTG13: L11 (Quiz review session)	
Normal forms	8	10/11	Wed	MTG14: Quiz 3 today (Analysis and Normal Forms)	
Normal forms	8	10/15	Sun		PRJ4 due (Phase 1 submission)
Intro to SQL	9	10/16	Mon	MTG15: L12 (Intro to SQL)	

Discuss Quarto

- What is working? What isn't working? What is confusing?



Guide
- Authoring
- Computations
- Tools
- Documents
- Presentations
- Websites
- Books
- Interactivity
- Publishing
- Projects
- Advanced

Guide

Comprehensive guide to using Quarto. If you are just starting out, you may want to explore the tutorials to learn the basics.

Authoring
Create content with markdown

Markdown Basics
Figures
Tables
Diagrams
Citations & Footnotes
Cross References
Article Layout

Computations
Execute code and display its output

Using Python
Using R
Using Julia
Using Observable
Execution Options
Parameters

Tools
Use your favorite tools with Quarto

JupyterLab
RStudio IDE
VS Code
Neovim
Text Editors
Visual Editor

Documents
Generate output in many formats

HTML
PDF
MS Word
Markdown
All Formats

Presentations
Present code and technical content

Presentation Basics
Revealjs (HTML)
PowerPoint (Office)
Beamer (PDF)

Websites
Create websites and blogs

Creating a Website
Website Navigation
Creating a Blog
Website Search
Website Listings

Books
Create books and manuscripts

Creating a Book
Book Structure
Book Crossrefs
Customizing Output

Interactivity
Engage readers with interactivity

Overview
Observable JS
Shiny
Widgets
Component Layout

Analysis and Normal forms

Relational database design to date

1. Identify main entities and relationships
   - Chen diagrams
2. Populate entities with attributes
   - Chen diagrams
   - Crows foot diagrams
3. Define the cardinality and participation of the relationships
   - Crows foot diagrams
4. Refine and simplify the model
   - Relational model notation
5. Translate the model to relational schemas
   - Define keys
   - Define queries using relational algebra

Are we there yet?

NO!!!! Then what else?

In the beginning … there was a google form …

Sample google form

This example form shows how data are collected and dropped into a google sheet for subsequent population into a database.

This form is automatically collecting emails for Virginia Commonwealth University users. Change settings

Enter Course and title
(e.g., CMSC508 - Database Theory)

Short answer text

Enter Instructor first and last name
(e.g., John Leonard)

Short answer text

Enter Instructor language favorites
(e.g., Perl, Python, C++, C, COBOL, FORTRAN, SQL, etc.)

Short answer text

… and then there were data, but …

RID	Course	Instructor	Languages
1	CMSC508 Databases	John Leonard	SQL, Python, Perl
2	CMSC508 Databases	Alberto Cano	SQL, Python, C++
3	CMSC475 UI/UX design	John Leonar	Javascript, Python
4	CMSC441 Capstone	Bob Dahlberg	COBOL, FORTRAN
5	CMSC320 Data Structures	Sarah Adams	C++, Java, Python
6	CMSC210 Software Design	Michael Turner	Java, C#
7	CMSC515 Computer Vision	Emily Parker	Python, MATLAB
8	CMSC430 Web Development	Jessica Clark	HTML, CSS, JavaScript
9	CMSC610 Machine Learning	Albert Cano	Python, R

But the data were ill-formed and unusable

- Each row represents a response
- Can relational algebra be used?
- Which instructors know SQL?
- Who teaches CMSC508?
- Is there redundant data?
- Are there typos?

How do we describe what we see?

How do we fix it?

How do we know when it’s fixed?

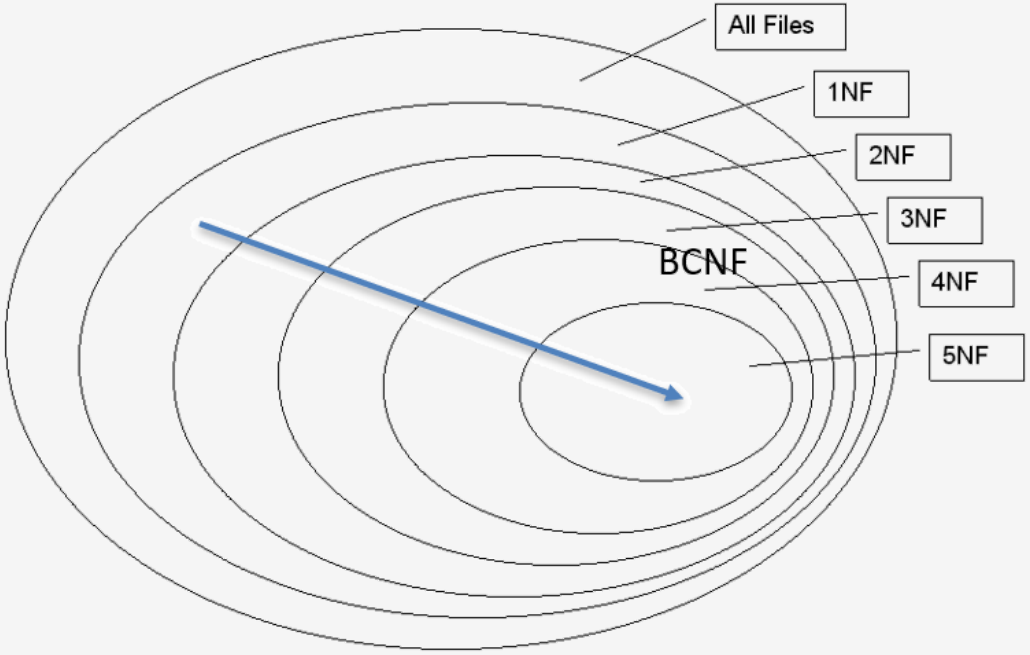
Normal forms

Database normalization is the process of reorganizing the relations to minimize data redundancy.

Normalization involves breaking down a table into less redundant and smaller tables without losing information by using functional dependencies.

The objective is to isolate data to minimize duplicates and so modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys and joins.

Escalating through the different normal forms removes more and more redundancy.



Normal forms

The usual suspects

First Normal Form - 1NF
A relation is in 1NF if and only if the domain of each attribute contains only atomic (indivisible) values and the value of each attribute contains only a single value from that domain.

Second Normal Form - 2NF
A relation is in 2NF if and only if it is in 1NF and all non-prime attributes (attributes not part of any candidate key) are fully functionally dependent on the entire candidate key.

Third Normal Form - 3NF
A relation is in 3NF if and only if it is in 2NF, and it has no transitive dependencies.

Boyce-Codd Normal Form - BCNF
A relation is in BCNF if and only if it is in 1NF, and for every non-trivial functional dependency A -> B, A is a superkey.

Crazy talk

Fourth Normal Form - 4NF
A relation is in 4NF if and only if it is in BCNF, and it has no multi-valued dependencies.

Fifth Normal Form - 5NF
A relation is in 5NF if and only if it is in 4NF, and it avoids join dependencies.

Sixth Normal Form - 6NF
A relation is in 6NF if and only if it is in 5NF, and it further eliminates all join dependencies and assures that every join dependency can be enforced by the superkeys of the relation.

Seventh Normal Form - 7NF
A relation is in 7NF if and only if it is in 6NF, and it eliminates all combinatorial join dependencies, ensuring that every possible join dependency is addressed.

Let's get normal - Our starting position

RID	Course	Instructor	Languages
1	CMSC508 Databases	John Leonard	SQL, Python, Perl
2	CMSC508 Databases	Alberto Cano	SQL, Python, C++
3	CMSC475 UI/UX design	John Leonar	Javascript, Python
4	CMSC441 Capstone	Bob Dahlberg	COBOL, FORTRAN
5	CMSC320 Data Structures	Sarah Adams	C++, Java, Python
6	CMSC210 Software Design	Michael Turner	Java, C#
7	CMSC515 Computer Vision	Emily Parker	Python, MATLAB
8	CMSC430 Web Development	Jessica Clark	HTML, CSS, JavaScript
9	CMSC610 Machine Learning	Albert Cano	Python, R

First Normal Form

A relation is in 1NF if and only if the domain of each attribute contains only atomic (indivisible) values and the value of each attribute contains only a single value from that domain.

- No multi-valued attributes. No arrays/lists in a cell.
- NOT REQUIRED BUT YOU SHOULD: break composite values

OK - let's do it.

First, we remove multi-valued attributes by giving *Languages* their own row

Let's get normal - First normal form

RID	Course	Instructor	Language
1	CMSC508 Databases	John Leonard	SQL
1	CMSC508 Databases	John Leonard	Python
1	CMSC508 Databases	John Leonard	Perl
2	CMSC508 Databases	Alberto Cano	SQL
2	CMSC508 Databases	Alberto Cano	Python
2	CMSC508 Databases	Alberto Cano	C++
3	CMSC475 UI/UX design	John Leonard	Javascript
3	CMSC475 UI/UX design	John Leonard	Python
4	CMSC441 Capstone	Bob Dahlberg	COBOL
4	CMSC441 Capstone	Bob Dahlberg	FORTRAN
5	CMSC320 Data Structures	Sarah Adams	C++
5	CMSC320 Data Structures	Sarah Adams	Java
5	CMSC320 Data Structures	Sarah Adams	Python
6	CMSC210 Software Design	Michael Turner	Java
6	CMSC210 Software Design	Michael Turner	C#
7	CMSC515 Computer Vision	Emily Parker	Python
7	CMSC515 Computer Vision	Emily Parker	MATLAB
8	CMSC430 Web Development	Jessica Clark	HTML
8	CMSC430 Web Development	Jessica Clark	CSS
8	CMSC430 Web Development	Jessica Clark	JavaScript

First Normal Form

A relation is in 1NF if and only if the domain of each attribute contains only atomic (indivisible) values and the value of each attribute contains only a single value from that domain.

- No multi-valued attributes. No arrays/lists in a cell.
- NOT REQUIRED BUT YOU SHOULD: break composite values

Don't stop now - let's keep going!
Now let's separate the composite attributes *Course* and *Instructor* into components.

Let’s get normal - First Normal Form

RID	Course Code	Course Name	First	Last	Language
1	CMSC508	Databases	John	Leonard	SQL
1	CMSC508	Databases	John	Leonard	Python
1	CMSC508	Databases	John	Leonard	Perl
2	CMSC508	Databases	Alberto	Cano	SQL
2	CMSC508	Databases	Alberto	Cano	Python
2	CMSC508	Databases	Alberto	Cano	C++
3	CMSC475	UI/UX design	John	Leonard	Javascript
3	CMSC475	UI/UX design	John	Leonard	Python
4	CMSC441	Capstone	Bob	Dahlberg	COBOL
4	CMSC441	Capstone	Bob	Dahlberg	FORTRAN
5	CMSC320	Data Structures	Sarah	Adams	C++
5	CMSC320	Data Structures	Sarah	Adams	Java
5	CMSC320	Data Structures	Sarah	Adams	Python
6	CMSC210	Software Design	Michael	Turner	Java
6	CMSC210	Software Design	Michael	Turner	C#
7	CMSC515	Computer Vision	Emily	Parker	Python
7	CMSC515	Computer Vision	Emily	Parker	MATLAB
8	CMSC430	Web Development	Jessica	Clark	HTML
8	CMSC430	Web Development	Jessica	Clark	CSS
8	CMSC430	Web Development	Jessica	Clark	JavaScript

First Normal Form

A relation is in 1NF if and only if the domain of each attribute contains only atomic (indivisible) values and the value of each attribute contains only a single value from that domain.

- No multi-valued attributes. No arrays/lists in a cell.
- NOT REQUIRED BUT YOU SHOULD: break composite values

Where do we go from here?

a. How many entities do we have?
b. Can you identify redundancies?
c. What is primary/candidate key? Why?
d. What are functional dependencies?
e. How would you remove redundancies?

Goal - get to Boyce Codd Normal Form (3.5NF)

RID	Course Code	Course Name	First	Last	Language
1	CMSC508	Databases	John	Leonard	SQL
1	CMSC508	Databases	John	Leonard	Python
1	CMSC508	Databases	John	Leonard	Perl
2	CMSC508	Databases	Alberto	Cano	SQL
2	CMSC508	Databases	Alberto	Cano	Python
2	CMSC508	Databases	Alberto	Cano	C++
3	CMSC475	UI/UX design	John	Leonard	Javascript
3	CMSC475	UI/UX design	John	Leonard	Python
4	CMSC441	Capstone	Bob	Dahlberg	COBOL
4	CMSC441	Capstone	Bob	Dahlberg	FORTRAN
5	CMSC320	Data Structures	Sarah	Adams	C++
5	CMSC320	Data Structures	Sarah	Adams	Java
5	CMSC320	Data Structures	Sarah	Adams	Python
6	CMSC210	Software Design	Michael	Turner	Java
6	CMSC210	Software Design	Michael	Turner	C#
7	CMSC515	Computer Vision	Emily	Parker	Python
7	CMSC515	Computer Vision	Emily	Parker	MATLAB
8	CMSC430	Web Development	Jessica	Clark	HTML
8	CMSC430	Web Development	Jessica	Clark	CSS
8	CMSC430	Web Development	Jessica	Clark	JavaScript

BCNF or 3.5NF

A relation is in BCNF if and only if it is in 1NF, and for every non-trivial functional dependency A -> B, A is a superkey.

Keys
Uniquely define entire rows.

Functional Dependencies
Uniquely defines relationships within rows.

BCNF
If you have a relationship within a row, it better be with a key!

The key, the whole key and nothing but the key, so help me Codd!

How do we get there from here?

We need to decompose the table replacing duplicate data with foreign keys. This will remove redundancy at the expense of creating new tables with joins.

Here is there - a fully BCNF schema

RID	Course Code	First	Last	Language
1	CMSC508	John	Leonard	SQL
1	CMSC508	John	Leonard	Python
1	CMSC508	John	Leonard	Perl
2	CMSC508	Alberto	Cano	SQL
2	CMSC508	Alberto	Cano	Python
2	CMSC508	Alberto	Cano	C++
3	CMSC475	John	Leonard	Javascript
3	CMSC475	John	Leonard	Python
4	CMSC441	Bob	Dahlberg	COBOL
4	CMSC441	Bob	Dahlberg	FORTRAN
5	CMSC320	Sarah	Adams	C++
5	CMSC320	Sarah	Adams	Java
5	CMSC320	Sarah	Adams	Python
6	CMSC210	Michael	Turner	Java
6	CMSC210	Michael	Turner	C#
7	CMSC515	Emily	Parker	Python
7	CMSC515	Emily	Parker	MATLAB
8	CMSC430	Jessica	Clark	HTML
8	CMSC430	Jessica	Clark	CSS
8	CMSC430	Jessica	Clark	JavaScript

Course Code	Course Name
CMSC210	Software Design
CMSC320	Data Structures
CMSC430	Web Development
CMSC441	Capstone
CMSC475	UI/UX design
CMSC508	Databases
CMSC515	Computer Vision
CMSC610	Machine Learning

First	Last
Alberto	Cano
Bob	Dahlberg
Sarah	Adams
John	Leonard
Michael	Turner
Emily	Parker
Jessica	Clark

Language
SQL
Python
Perl
C++
Javascript
COBOL
FORTRAN
Java
C#
MATLAB
HTML
CSS
R

RID
1
2
3
4
5
6
7
8
9

Try 2 - towards BCNF

RID	CID	Course Code	Course Name	PID	First	Last	LID	Language
1	1	CMSC508	Databases	1	John	Leonard	1	SQL
1	1	CMSC508	Databases	1	John	Leonard	2	Python
1	1	CMSC508	Databases	1	John	Leonard	3	Perl
2	1	CMSC508	Databases	2	Alberto	Cano	1	SQL
2	1	CMSC508	Databases	2	Alberto	Cano	2	Python
2	1	CMSC508	Databases	2	Alberto	Cano	4	C++
3	2	CMSC475	UI/UX design	1	John	Leonard	5	Javascript
3	2	CMSC475	UI/UX design	1	John	Leonard	2	Python
4	3	CMSC441	Capstone	3	Bob	Dahlberg	6	COBOL
4	3	CMSC441	Capstone	3	Bob	Dahlberg	7	FORTRAN
5	4	CMSC320	Data Structures	4	Sarah	Adams	4	C++
5	4	CMSC320	Data Structures	4	Sarah	Adams	8	Java
5	4	CMSC320	Data Structures	4	Sarah	Adams	2	Python
6	5	CMSC210	Software Design	5	Michael	Turner	8	Java
6	5	CMSC210	Software Design	5	Michael	Turner	9	C#
7	6	CMSC515	Computer Vision	6	Emily	Parker	2	Python
7	6	CMSC515	Computer Vision	6	Emily	Parker	10	MATLAB
8	7	CMSC430	Web Development	7	Jessica	Clark	11	HTML
8	7	CMSC430	Web Development	7	Jessica	Clark	12	CSS
8	7	CMSC430	Web Development	7	Jessica	Clark	5	JavaScript

Try 2 - Final BCNF solution

CID	Course Code	Course Name
1	CMSC508	Databases
2	CMSC475	UI/UX design
3	CMSC441	Capstone
4	CMSC320	Data Structures
5	CMSC210	Software Design
6	CMSC515	Computer Vision
7	CMSC430	Web Development
8	CMSC610	Machine Learning

PID	First	Last
1	John	Leonard
2	Alberto	Cano
3	Bob	Dahlberg
4	Sarah	Adams
5	Michael	Turner
6	Emily	Parker

LID	Language
1	SQL
2	Python
3	Perl
4	C++
5	Javascript
6	COBOL
7	FORTRAN
8	Java
9	C#
10	MATLAB
11	HTML
12	CSS
13	R

RID
1
2
3
4
5
6
7
8
9

RID	CID	PID	LID
1	1	1	1
1	1	1	2
1	1	1	3
2	1	2	1
2	1	2	2
2	1	2	4
3	2	1	5
3	2	1	2
4	3	3	6
4	3	3	7
5	4	4	4
5	4	4	8
5	4	4	2
6	5	5	8
6	5	5	9
7	6	6	2
7	6	6	10
8	7	7	11
8	7	7	12
8	7	7	5

Discussion

- The attributes in each relation depend on the primary key, the whole key and nothing else.
- No duplicates in each table.
- We can reconstruct the original table

$$Original = Join \bowtie Crse \bowtie Inst \bowtie Lang \bowtie Resp$$

Queries

Who knows SQL?

$$\pi_{(First,Last)}(\theta_{(Language=SQL)} Join \bowtie Lang \bowtie Inst)$$

What courses are taught by python programmers?

$$\pi_{CourseName}(\theta_{(Language=python)} Join \bowtie Lang \bowtie Inst \bowtie Crse)$$

What languages are known by instructors of CMSC508?

$$\pi_{Language}(\theta_{(CourseCode=CMSC508)} Join \bowtie Lang \bowtie Inst \bowtie Crse)$$

This CS - we need an algorithm!

The preceding examples show how an original table can be decomposed to minimize redundancy and improve integrity.

This manual, *by inspection* approach is not acceptable for computer science!

So of course, we can develop a notation and implement an algoritm to analyze relations.

This algorithm can be used to clean up original tables, or verify existing tables are BCNF.

We'll be spending the next bunch of lectures working with these algorithms.

Housekeeping

1. No homework dues this week!
2. Project deliverable 4 - posted and available
3. Quiz and quiz stats - Very nice!
4. Peer reviews - still some issues.
5. Changes to schedule (Version 4, see table below)
6. Discuss Quarto

Module	Week	Date	Day	Lectures/Quizzes	Deliverables/Notes
Normal forms	6	9/27	Wed	MTG10: L8 (Analysis and Normal Forms 1)	
Normal forms	7	10/2	Mon	MTG11: L9 (Analysis and Normal Forms 2)	
Normal forms	7	10/4	Wed	MTG12: L10 (Analysis and Normal Forms 3)	
Normal forms	8	10/9	Mon	MTG13: L11 (Quiz review session)	
Normal	8	10/11	Wed	MTG14: Quiz 3 today (Analysis and	