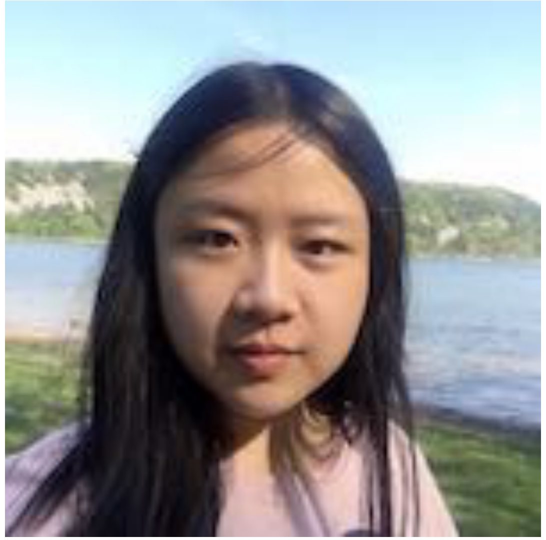
Lecture 8:
Design Theory III

Announcements

- Grades for PS1 on Canvas.
  - For grading questions: ***your best bet is Minzhen***



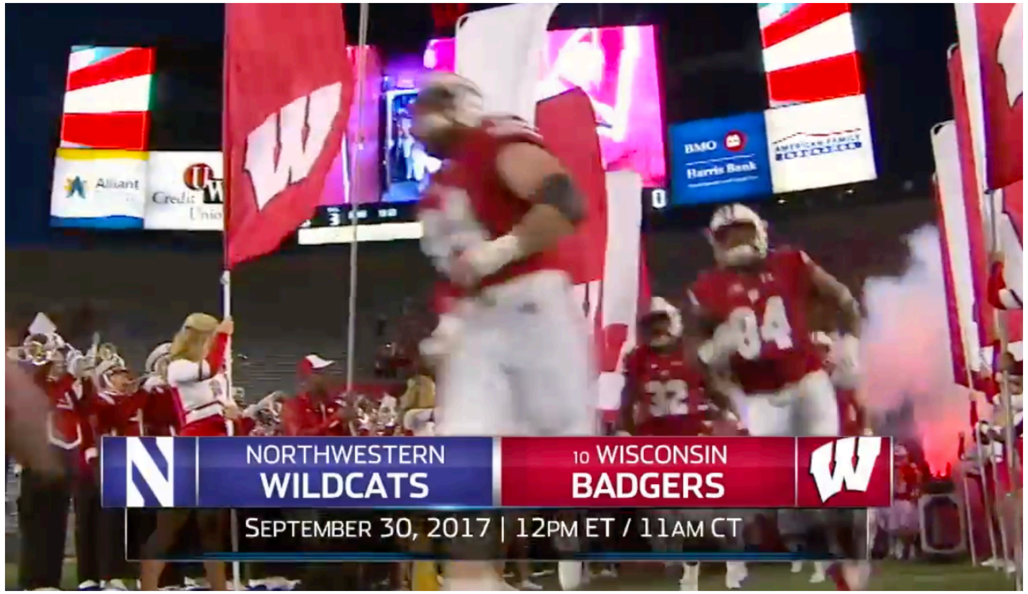
Minzhen is the real BOSS!

Announcements

- Grades for PS1 on Canvas.
  - For grading questions: ***your best bet is Minzhen (She is the real BOSS)***
- Project part 1 due next Wednesday 10/4 @ Midnight.
  - PUSH PUSH PUSH!
  - Discussion at the end of the lecture today

Announcements

- Grades for PS1 on Canvas.
  - For grading questions: ***your best bet is Minzhen (She is the real BOSS)***
- Project part 1 due next Wednesday 10/4 @ Midnight.
- PS2 is out! Due next Friday 10/6 @ Midnight.
  - MUCH EASIER! Focus on project!
  - Do PS2 while watching the game tomorrow!



Today's Lecture

1. 3rd – Normal Form

2. Multi-Value Dependencies (MVDs)
   - ACTIVITY

3. Project Part1 - Discussion

1. 3NF and Dependency Preservation

What you will learn about in this section

1. Recap: Dependency Preserving Decompositions

2. 3NF Definition

3. 3NF Decomposition

Boyce-Codd Normal Form

BCNF is a simple condition for removing anomalies from relations:

A relation R is <u>**in BCNF**</u> if:

if $\{A_1, \ldots, A_n\} \rightarrow B$ is a *non-trivial* FD in R

then $\{A_1, \ldots, A_n\}$ **is a superkey** for R

Equivalently: ∀ sets of attributes X, either ($X^+ = X$) or (X^+ = all attributes)

In other words: there are no “bad” FDs

Boyce-Codd Normal Form

BCNF is a simple condition for removing anomalies from relations:

A relation R is <u>**in BCNF**</u> if:

if $\{A_1, \ldots, A_n\} \rightarrow B$ is a *non-trivial* FD in R

then $\{A_1, \ldots, A_n\}$ **is a superkey** for R

Equivalently: ∀ sets of attributes X, either ($X^+ = X$) or (X^+ = all attributes)

In other words: there are no "bad" FDs

Dependency Preserving Decompositions

- Given **R** and a set of FDs *F*, we decompose **R** into **R1** and **R2**. Suppose:
  - **R1** has a set of FDs *F1*
  - **R2** has a set of FDs *F2*
  - *F1* and *F2* are computed from *F*

A decomposition is **dependency preserving** if by enforcing *F1* over **R1** and *F2* over **R2**, we can enforce *F* over **R**

Bad Example

Unit	Company
Galaga99	UW
Bingo	UW

Unit	Product
Galaga99	Databases
Bingo	Databases

No problem so far. All *local* FD's are satisfied.

{Unit} → {Company}

Unit	Company	Product
Galaga99	UW	Databases
Bingo	UW	Databases

Let's put all the data back into a single table again:

Violates the FD {Company,Product} → {Unit}!!

Possible Solutions

- Various ways to handle so that decompositions are all lossless / no FDs lost
  - For example **3NF**- stop short of full BCNF decompositions.

- Usually a tradeoff between redundancy / data anomalies and FD preservation…

BCNF still most common- with additional steps to keep track of lost FDs…

3NF Definition

A relation R is **in <u>3NF</u>** if:

If **$\{A_1, ..., A_n\}$** → **B** is a *non-trivial* FD in R

then $\{A_1, ..., A_n\}$ is a superkey for R **OR**

B is part of some key of R ***(prime attribute)***

BCNF implies 3NF. Why?

Why use 3NF?

- **Example**: **R**(A, B, C) with $A, B \longrightarrow C$ and $C \longrightarrow A$
  - is in 3NF. Why?
  - is not in BCNF. Why?

Compromise used when BCNF not achievable: *aim for BCNF and settle for 3NF*

Lossless-join and dependency preserving decomposition into a collection of 3NF relations is always possible!

3NF Decomposition

1. Apply the algorithm for BCNF decomposition until all relations are in 3NF (we can stop earlier than BCNF)
2. Compute a minimal basis *F'* of *F*
3. For each non-preserved FD $X \longrightarrow A$ in *F'*, add a new relation R(X, A)

This is not fair game; read textbook for minimal basis

You only need to remember that to get 3NF we stop short of full BCNF decompositions

3. MVDs

What you will learn about in this section

1. MVDs
2. ACTIVITY

Multi-Value Dependencies (MVDs)

- A multi-value dependency (MVD) is another type of dependency that could hold in our data, ***which is not captured by FDs***

- Formal definition:
  - Given a relation **R** having attribute set **A**, and two sets of attributes $\mathbf{X}, \boldsymbol{Y} \subseteq \boldsymbol{A}$
  - The ***multi-value dependency (MVD)*** $\boldsymbol{X} \twoheadrightarrow \boldsymbol{Y}$ holds on R if
  - ***for any tuples*** $\boldsymbol{t_1}, \boldsymbol{t_2} \in \boldsymbol{R}$ s.t. $\boldsymbol{t_1}[X] = \boldsymbol{t_2}[X]$, there exists a tuple $\boldsymbol{t_3}$ s.t.:
    - **$t_1[X] = t_2[X] = t_3[X]$**
    - **$t_1[Y] = t_3[Y]$**
    - **$t_2[A \setminus Y] = t_3[A \setminus Y]$**
      - *Where A \ B means "elements of set A not in set B"*

Multi-Value Dependencies (MVDs)

- One less formal, literal way to phrase the definition of an MVD:

- ***The MVD $X \twoheadrightarrow Y$*** holds on R if for any pair of tuples with the same X values, the "swapped" pair of tuples with the same X values, but the other permutations of Y and A\Y values, is also in R

Ex: X = {x}, Y = {y}:

x	y	z
1	0	1
1	1	0

For $X \twoheadrightarrow Y$ to hold must have…

x	y	z
1	0	1
1	1	0
1	0	0
1	1	1

Note the connection to a local *cross-product…*

Multi-Value Dependencies (MVDs)

- Another way to understand MVDs, in terms of *conditional independence:*

- ***The MVD*** $\boldsymbol{X} \twoheadrightarrow \boldsymbol{Y}$ holds on R if given X, Y is conditionally independent of A \ Y and vice versa…

Here, given x = 1, we know for ex. that:
y = 0 → z = 1

I.e. z is conditionally ***dependent*** on y given x

x	y	z
1	0	1
1	1	0

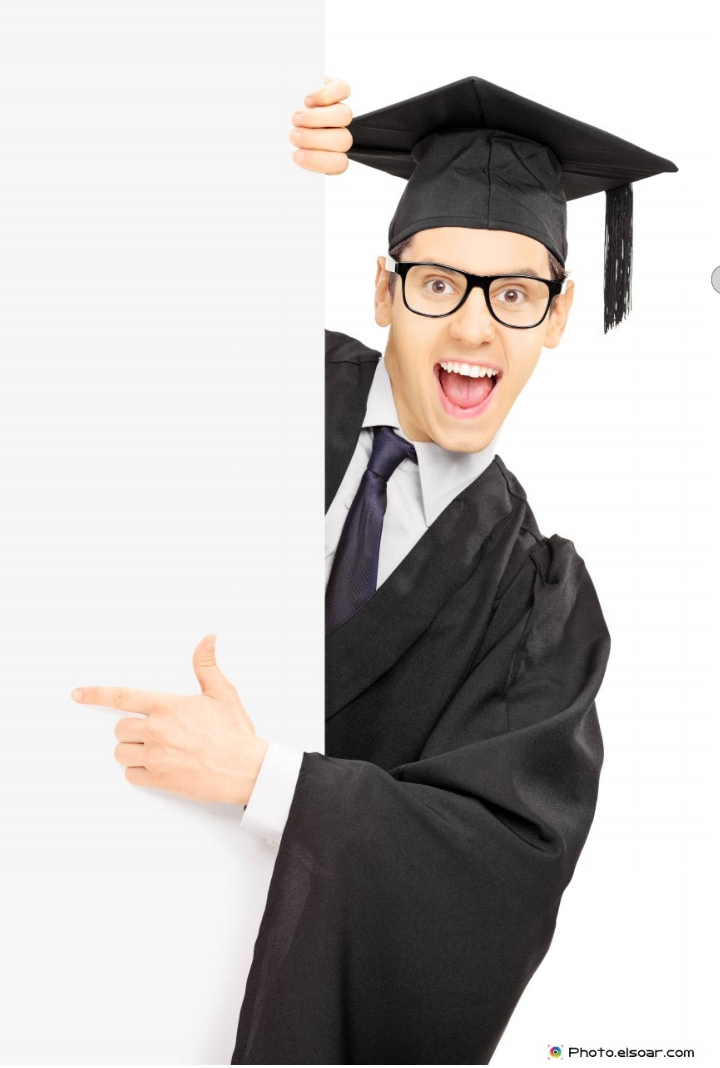
Here, this is not the case!

I.e. z is conditionally ***independent*** of y given x

x	y	z
1	0	1
1	1	0
1	0	0
1	1	1

Multiple Value Dependencies (MVDs)

A “real life” example…



Grad student thinks:

“Hmm… what is real life?? Watching a movie over the weekend?”

MVDs: Movie Theatre Example

Movie_theater	film_name	snack
UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
UWM 1	Star Trek: The Wrath of Kahn	Burrito
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
UWM 2	Star Wars: The Boba Fett Prequel	Ramen
UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

Are there any functional dependencies that might hold here?

No…

And yet it seems like there is some pattern / dependency…

MVDs: Movie Theatre Example

Movie_theater	film_name	snack
UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
UWM 1	Star Trek: The Wrath of Kahn	Burrito
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
UWM 2	Star Wars: The Boba Fett Prequel	Ramen
UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

For a given movie theatre…

MVDs: Movie Theatre Example

Movie_theater	film_name	snack
UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
UWM 1	Star Trek: The Wrath of Kahn	Burrito
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
UWM 2	Star Wars: The Boba Fett Prequel	Ramen
UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

For a given movie theatre…

Given a set of movies and snacks…

MVDs: Movie Theatre Example

Movie_theater	film_name	snack
UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
UWM 1	Star Trek: The Wrath of Kahn	Burrito
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
UWM 2	Star Wars: The Boba Fett Prequel	Ramen
UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

For a given movie theatre…

Given a set of movies and snacks…

Any movie / snack combination is possible!

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_1	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
	UWM 1	Star Trek: The Wrath of Kahn	Burrito
	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_2	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

More formally, we write $\{A\} \twoheadrightarrow \{B\}$ if for any tuples t_1,t_2 s.t. $t_1[A] = t_2[A]$

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_1	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
t_3	UWM 1	Star Trek: The Wrath of Kahn	Burrito
	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_2	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

More formally, we write $\{A\} \twoheadrightarrow \{B\}$ if for any tuples t_1,t_2 s.t. $t_1[A] = t_2[A]$ there is a tuple t_3 s.t.

- $t_3[A] = t_1[A]$

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_1	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
t_3	UWM 1	Star Trek: The Wrath of Kahn	Burrito
	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_2	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

More formally, we write **{A} ↠ {B}** if for any tuples t_1,t_2 s.t. $t_1[A] = t_2[A]$ there is a tuple t_3 s.t.

- $t_3[A] = t_1[A]$
- $t_3[B] = t_1[B]$

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_1	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
t_3	UWM 1	Star Trek: The Wrath of Kahn	Burrito
	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_2	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

More formally, we write $\{A\} \twoheadrightarrow \{B\}$ if for any tuples t_1,t_2 s.t. $t_1[A] = t_2[A]$ there is a tuple t_3 s.t.

- $t_3[A] = t_1[A]$
- $t_3[B] = t_1[B]$
- and $t_3[R\backslash B] = t_2[R\backslash B]$

Where R\B is "R minus B" i.e. the attributes of R not in B

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_2	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
	UWM 1	Star Trek: The Wrath of Kahn	Burrito
t_3	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_1	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

Note this also works!

Remember, an MVD holds over *a relation or an instance*, so defn. must hold for every applicable pair…

MVDs: Movie Theatre Example

	Movie_theater (A)	film_name (B)	Snack (C)
t_2	UWM 1	Star Trek: The Wrath of Kahn	Kale Chips
	UWM 1	Star Trek: The Wrath of Kahn	Burrito
t_3	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Kale Chips
t_1	UWM 1	Lord of the Rings: Concatenated & Extended Edition	Burrito
	UWM 2	Star Wars: The Boba Fett Prequel	Ramen
	UWM 2	Star Wars: The Boba Fett Prequel	Plain Pasta

This expresses a sort of dependency (= data redundancy) that we *can't* express with FDs

**Actually, it expresses *conditional independence* (between film and snack given movie theatre)!*

Connection to FDs

If $A \rightarrow B$ does $A \twoheadrightarrow B$?

Comments on MVDs

- MVDs have "rules" too!
  - **Experts**: Axiomatizable

- 4^{th} Normal Form is "non-trivial MVD"

- *For AI nerds*: MVD is conditional independence in graphical models!

See the MVDs IPython notebook
for more examples!

Summary

- Constraints allow one to reason about **redundancy** in the data

- Normal forms describe how to **remove** this redundancy by **decomposing** relations
  - Elegant—by representing data appropriately certain errors are essentially impossible
  - For FDs, BCNF is the normal form.

- A tradeoff for insert performance: 3NF

3. Project Part 1: Discussion

Going Once, Going Twice ...

Q: Is it a **relationship** or an **entity set**?

A: Should it be a **set** or a **multiset**? Do I need multiple instances of an element or one?

Going Once, Going Twice ...

Q: Is a User a Seller or a Buyer?

A: Think of what the current json schema says. “Note that a user may be a bidder in one auction and a seller in another. However, his Rating, Location, and Country information are the same wherever he appears in our data (which reflects a snapshot in time).

Going Once, Going Twice …

Q: Currently and Number_of_Bids?

A: Just follow the description ☺