Discussion 6

Normalization

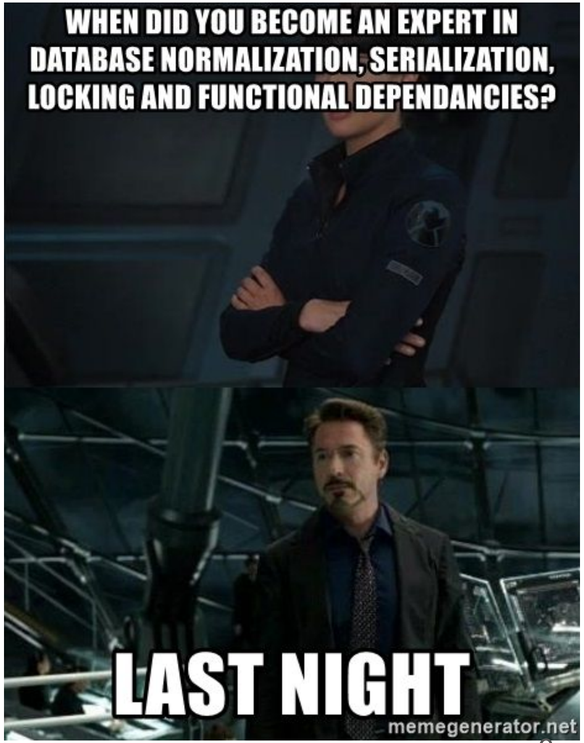
EECS 484

Logistics

- Project 2 deadline extended - due on Saturday, Oct. 8!
- HW3 due Oct 13
- Project 1 handwritten part graded - accepting regrade requests by Oct 13
- Midterm Exam Oct 21 (Friday), 7:30 - 9:30 EST
  - One double sided 8.5 x 11 **handwritten** cheat sheet is allowed
  - No discussion next week (13th, 14th) or the week after (20th, 21st)
  - Review session Oct 14 (Friday) 2:30, DOW 3150
  - Review session Oct 19 (Wednesday) 3:00 - 6:00, Zoom, recorded
  - Practice exams are available on Canvas
    - The solution file may have minor mistakes. Ask on Piazza.

Normalization Theory

- There are a lot of different ways we can represent data in a relation
  - Want to reduce redundancy and maximize integrity
- How do we make good tables
  - Start with a nice ER diagram and make tables
    - Great if starting from scratch
  - Take existing tables and decompose them into smaller tables
    - Will do this using functional dependencies and normal forms



Functional Dependency

- A → B
  - A functionally determines B
  - "If you know A, then you know B"
    - No other options
- F is a set of FDs for a relation
- F+ = closure of F = set of all FDs we can derive from F
- Example
  - F = ID→ Name, (ID→Title, Title→Manager)

    ID → manager

    F+ also includes, by transitivity, ID→Manager
    - Notice, ID allows me to functionally determine all columns

ID	Name	Title	Manager
1	John	Clerk	Alice
2	Jane	Clerk	Alice
3	Alice	Manager	Bob
4	Bob	Owner	Bob

Armstrong's Axioms

- Rules of inference for functional dependencies
  - Repeatedly applying them allows us to generate F+
- Reflexivity
  - If y ⊆ x then x→y   AB→A   x, y, z are set of attributes
  - Trivial self dependence
- Augmentation
  - If x→y then xz→yz for any z
- Transitivity
  - If x→y and y→z then x→z
- Derived axioms
  - Union: If x→y and x→z then x→yz
  - Decomposition: if x→yz then x→y and x→z

Relation Decomposition

A	B	C
1	2	3
4	2	5

A	B
1	2
4	2

$\bowtie$

B	C
2	3
2	5

1 2 3 ✓

1 2 5 ✗ junk.

- Let's split up larger relations into smaller relations
  - More relations means that data is more spread out
    - Will need joins to connect them back
    - **Lossless join** = can reconstruct data of original relation from decomposed relations
    - **Dependency preserving** = do we still have all the same dependencies
    - Lossless join is required while dependency preserving is nice to have
  - Notationally:
    - Decompose X into Y and Z
    - Lossless join: $(Y \cap Z \to Y)$ or $(Y \cap Z \to Z)$
      - Attributes common to Y and Z contains a key for Y or Z
    - Dependency preserving: $F+ = (F_Y \cup F_Z)+$
      - F_Y is the set of FDs in F+ that only involve attributes in Y (same for F_Z)
      - Original dependencies are preserved in the decomposed relations
- We'll use normal forms to help us decompose

Third Normal Form (3NF)

attributes that don't belong to any candidate key

- Builds off First and Second Normal Forms
  - 1NF = Each column contains a single value and each row must be unique
  - 2NF = 1NF + Attributes not a part of the primary key are fully dependent on primary key
  - 3NF = 2NF + No transitive functional dependencies on non-prime attributes
- Relation R is in 3NF form if and only if for all dependencies of the form $X \rightarrow A$ in F+ **(where X is a subset of attributes of R and A is a single attribute of R)**, the following hold true:
  - $A \subseteq X$ (trivial dependence) or $A \to A$
  - X is a super key or candidate key + 0/more attributes.
  - A is some part of a minimal key in R

CK: AB

F: C→B

- Superkey is a set of attributes that can unique identify a row in a table
  - Can be larger than candidate key (non-minimal)
- Can always decompose R into a collection of 3NF relations that satisfies the lossless join and dependency preserving properties F: C→AB not violated

① ② extract single attribute on RHS: F+: C→A, C→B

Boyce-Codd Normal Form (BCNF)

BCNF ⇒ 3NF

- Relation R is in BCNF form if and only if for all dependencies of the form $X \rightarrow A$ in F+ (where X is a subset of attributes of R and A is a single attribute of R)
  - $A \subseteq X$ (trivial dependence) or
  - X is a super key
- Missing the "part of a minimal key" category from 3NF
- Stronger than 3NF
  - If relation is BCNF then it is 3NF
    - Not the other way around though
- To decompose R into BCNF, start with unnormalized relation
  - If $X \rightarrow Y$ violates BCNF decompose into R-Y and XY
    - Repeat for all $X \rightarrow Y$ ⇒ that violates BCNF
  - Satisfies lossless join property, but not always dependency preserving

Examples

Question 1

- Consider a relation R=(A,B,C,D,E)
    - Dependencies as follows
    - A→B
    - A→C
    - A→D
    - BE→D
    - BE→AC
    - C→E
- What are candidate keys here?
    - Remember a candidate key is a **minimal** set of columns that allow us to **uniquely** define the relation
- Is R 3NF?
    - If so, is R BCNF?

Relation R with FDs F is in 3NF if, for all $X \rightarrow A$ in F^+

- $A \subseteq X$ (trivial dependency) or
- X is a super key or
- A is part of some (minimal) key for R **(prime attribute)**

X: subset of attributes

A: single attribute

Question 1

BE → DAC

BE → ABCDE

- Consider a relation R=(A,B,C,D,E)
  - Dependencies as follows
  - A→B
  - A→C
  - A→D
  - BE→D
  - BE→AC
  - C→E

Relation R with FDs F is in 3NF if, for all X → A in F^+

- $A \subseteq X$ (trivial dependency) or
- X is a super key or
- A is part of some (minimal) key for R **(prime attribute)**

X: subset of attributes

A: single attribute

- What are candidate keys here?
  - Remember a candidate key is a minimal set of columns that allow us to uniquely define the relation Keys founds = {A}

    Repetitively union A→B, A→C and A→D to get A→BCD

    Augment A on both sides to get A→ABCD

    Use transitivity on A→C and C→E to get A→E

    Union A→ABCD and A→E to get A→ABCDE

Question 1

- Consider a relation R=(A,B,C,D,E)
    - Dependencies as follows
    - A→B
    - A→C
    - A→D
    - BE→D
    - BE→AC
    - C→E

Relation R with FDs F is in 3NF if, for all X → A in F^+

- $A \subseteq X$ (trivial dependency) or
- X is a super key or
- A is part of some (minimal) key for R **(prime attribute)**

X: subset of attributes
A: single attribute

- What are candidate keys here?
    - Remember a candidate key is a minimal set of columns that allow us to uniquely define the relation

Keys founds = {A, BE} ⇒ given these check if any attributes derive candidate keys

Union BE→D and BE→AC to get BE→ACD
Augment BE on both sides to get BE→ABCDE

∵ C→E ⇒ BC→BE

Question 1

- Consider a relation R=(A,B,C,D,E)
  - Dependencies as follows
  - A→B
  - A→C
  - A→D
  - BE→D
  - BE→AC
  - C→E

Relation R with FDs F is in 3NF if, for all $X \rightarrow A$ in F^+

X: subset of attributes
A: single attribute

- $A \subseteq X$ (trivial dependency) or
- X is a super key or
- A is part of some (minimal) key for R (prime attribute)

- What are candidate keys here?
  - Remember a candidate key is a minimal set of columns that allow us to uniquely define the relation

Keys founds = {A, BE, BC}

Augment B on both sides of C→E to get BC→BE

Since we already know BE is a key

Use transitivity on BC→BE and BE→ABCDE to get BC→ABCDE

① don't consider super set of these

② First look at RHS of arrow

⇒ if attribute not occur on RHS

⇒ must be in candidate key

③ look at LHS, if some attribute doesn't occur on LHS, it shouldn't be a candidate key

Question 1

- Consider a relation R=(A,B,C,D,E)
  - Dependencies as follows
  - $A \rightarrow B$
  - $A \rightarrow C$
  - $A \rightarrow D$
  - $C \rightarrow E$
  - $BE \rightarrow D$
  - $BE \rightarrow AC$

Relation R with FDs F is in 3NF if, for all $X \rightarrow A$ in F^+

- $A \subseteq X$ (trivial dependency) or
- X is a super key or
- A is part of some (minimal) key for R **(prime attribute)**

X: subset of attributes

A: single attribute

- FDs in F^+ = {$A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $C \rightarrow E$, $BE \rightarrow D$, $BE \rightarrow A$, $BE \rightarrow C$, ...}
- Candidate keys = {A, BC, BE}
- Is R 3NF?
  - Yes

Question 1

- Consider a relation R=(A,B,C,D,E)
  - Dependencies as follows
  - $A \rightarrow B$
  - $A \rightarrow C$
  - $A \rightarrow D$
  - $C \rightarrow E$
  - $BE \rightarrow D$
  - $BE \rightarrow AC$

Rel. R with FDs F is in BCNF if, for all $X \rightarrow A$ in F^+

- $A \subseteq X$ (trivial FD), or
- X is a super key

(X:subset of attributes
A: single attribute)

- FDs in F^+ = {$A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $C \rightarrow E$, $BE \rightarrow D$, $BE \rightarrow A$, $BE \rightarrow C$, ……}
- Candidate keys = {A, BC, BE}
- Is R BCNF?
  - No, $C \rightarrow E$ violates both constraints

Question 2

- Relation R = (A,B,C,D,E)
  - Dependencies as follows
  - $A \to B$
  - $A \to C$
  - $A \to D$
  - $C \to E$
  - $BE \to D$
  - $BE \to AC$
- Decompose into BCNF

High-Level Algorithm

Input: a relation R with FDs F

1. Identify if any FDs violate BCNF (How?)
   - If $X \to Y$ violates BCNF, decompose R into **R - Y and XY**
2. Repeat for every $X \to Y$ that violates BCNF.

Output: a collection of relations that are in BCNF

CE ABCD.

$C \to E$

a key for smaller relation

$\Rightarrow$ BCNF

$A \to BCD$

superkey

Question 2

- Relation R = (A,B,C,D,E)
  - Dependencies as follows
  - $A \rightarrow B$
  - $A \rightarrow C$
  - $A \rightarrow D$
  - $C \rightarrow E$
  - $BE \rightarrow D$
  - $BE \rightarrow AC$
- Decompose into BCNF
  - We know $C \rightarrow E$ violates constraint
  - R1={A,B,C,D} R2={C,E}
  - Is this lossless?
    - Yes, $R1 \cap R2 = C$ which is a key for R2 ($C \rightarrow E$)

High-Level Algorithm

Input: a relation R with FDs F

1. Identify if any FDs violate BCNF (How?)
   - If $X \rightarrow Y$ violates BCNF, decompose R into **R - Y and XY**
2. Repeat for every $X \rightarrow Y$ that violates BCNF.

Output: a collection of relations that are in BCNF

Relation R, FDs F; Decomposed to X, Y

- Test: lossless-join w.r.t. F if and only if F^+ contains:

$$X \cap Y \rightarrow X, \quad \text{or} \quad X \cap Y \rightarrow Y$$

i.e. attributes common to X and Y contain a key for either X or Y

Question 2

- Relation R = (A,B,C,D,E)
  - Dependencies as follows
  - $A \rightarrow B$
  - $A \rightarrow C$
  - $A \rightarrow D$
  - $C \rightarrow E$
  - $BE \rightarrow D$
  - $BE \rightarrow AC$
- Decompose into BCNF
  - R1={A,B,C,D} R2={C,E}
  - Is this dependency preserving?
    - F1 contains $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, F2 contains $C \rightarrow E$
    - $(F1 \cup F2)+$ contains $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, $A \rightarrow E$, $C \rightarrow E$
    - Not equal to original F+, so not dependency preserving

Informally: We don't want the original FDs to span two tables.

R has a dependency-preserving decomposition to X, Y

if $F^+ = (F_x \cup F_y)^+$

Question 2

- Relation R = (A,B,C,D,E)
  - Dependencies as follows
  - $A \to B$
  - $A \to C$
  - $A \to D$
  - $C \to E$
  - $BE \to D$
  - $BE \to AC$

Rel. R with FDs F is in BCNF if, for all $X \to A$ in F^+

- $A \subseteq X$ (trivial FD), or
- X is a super key

X:subset of attributes
A: single attribute

- Decompose into BCNF
  - R1={A,B,C,D} R2={C,E}
  - R1 and R2 are both in BCNF form
    - You can check this
    - Note: No way of testing if collection of relations are in BCNF, only one relation at a time
  - Since R1 and R2 don't preserve dependencies, we could leave R as a 3NF relation

Get started on HW3!