Design Theory: Functional Dependencies and Normal Forms, Part I

Instructor: Shel Finkelstein

Reference:
A First Course in Database Systems,
3rd edition, Chapter 3

Important Notices

- Final Exam is on Wednesday, March 22, noon-3pm in our usual classroom.
  - Final from Fall 2016 has been posted on Piazza (Resources→Exams). Answers to that Final will be posted during last week of classes.
- Lab4 assignment was posted on Monday, Feb 27.
  - Due by Sunday, March 12, 11:59pm (2 weeks).
  - Lab4 focusses on material in Lecture 10 (Application Programming), including JDBC and Stored Procedures/Functions.
  - If you don't attend Lectures and Labs, you probably will find Lab4 difficult.
- Gradiance Assignment #4 is due by Tuesday, March 7, 11:59pm.

Database Schema Design

- So far, we have learned database query languages:
  - SQL, Relational Algebra

- How can you tell whether a given database schema is “good” or “bad”?

- Design theory:
  - A set of design principles that allows one to determine what constitutes a “good” or “bad” database schema design.
  - A set of algorithms for modifying a “bad” design to a “better” one.

Example

- If we know that rank determines the salary scale, which is a better design? Why?

- Employees(<u>eid</u>, name, addr, rank, salary_scale)

OR

- Employees(<u>eid</u>, name, addr, rank)
  Salary_Table(<u>rank</u>, salary_scale)

Lots of Duplicate Information

eid	name	addr	rank	salary_scale
34-133	Jane	Elm St.	6	70-90
33-112	Hugh	Pine St.	3	30-40
26-002	Gary	Elm St.	4	35-50
51-994	Ann	South St.	4	35-50
45-990	Jim	Main St.	6	70-90
98-762	Paul	Walnut St.	4	35-50

- Lots of duplicate information
  - Employees who have the same rank have the same salary scale.

Update Anomaly

eid	name	addr	rank	salary_scale
34-133	Jane	Elm St.	6	70-90
33-112	Hugh	Pine St.	3	30-40
26-002	Gary	Elm St.	4	35-50
51-994	Ann	South St.	4	35-50
45-990	Jim	Main St.	6	70-90
98-762	Paul	Walnut St.	4	35-50

- Update anomaly
  - If one copy of salary scale is changed, then all copies of that salary scale (of the same rank) have to be changed.

Insertion Anomaly

eid	name	addr	rank	salary_scale
34-133	Jane	Elm St.	6	70-90
33-112	Hugh	Pine St.	3	30-40
26-002	Gary	Elm St.	4	35-50
51-994	Ann	South St.	4	35-50
45-990	Jim	Main St.	6	70-90
98-762	Paul	Walnut St.	4	35-50

- Insertion anomaly
  - How can we store a new rank and salary scale information if currently, no employee has that rank?
  - Use NULLS?

Deletion Anomaly

eid	name	addr	rank	salary_scale
34-133	Jane	Elm St.	6	70-90
33-112	Hugh	Pine St.	3	30-40
26-002	Gary	Elm St.	4	35-50
51-994	Ann	South St.	4	35-50
45-990	Jim	Main St.	6	70-90
98-762	Paul	Walnut St.	4	35-50

- Deletion anomaly
  - If Hugh is deleted, how can we retain the rank and salary scale information?
  - Is using NULL a good choice?
    - (Why not?)

So What Would Be a Good Schema Design for this Example?

- salary_scale is dependent only on rank
  - Hence associating employee information such as name, addr with salary_scale causes redundancy.

- Based on the constraints given, we would like to refine the schema so that such redundancies **cannot** occur.

- Note however, that sometimes database designers **may choose** to live with redundancy in order to improve query performance.
  - Ultimately, a good design is depends on the query workload.
  - But understanding anomalies and how to deal with them is still important.

Functional Dependencies

- The information that rank determines salary_scale is a type of integrity constraint known as a *functional dependency (FD).*

- Functional dependencies can help us detect anomalies that may exist in a given schema.

- The FD "rank → salary_scale" suggests that

  Employees(eid, name, addr, rank, salary_scale)

  should be *decomposed* into two relations:

  Employees(eid, name, addr, rank)

  Salary_Table(rank, salary_scale).

Meaning of an FD

- We have seen a kind of functional dependency before.
- Keys:
  - Emp(<u>ssn</u>, name, addr)
  - If two tuples agree on the ssn value, then they must also agree on the name and address values. (ssn $\rightarrow$ name, addr).

- Let **R** be a relation schema. A *functional dependency (FD)* is an integrity constraint of the form:

  $X \rightarrow Y$ (read as "X *determines* Y or X *functionally determines* Y")

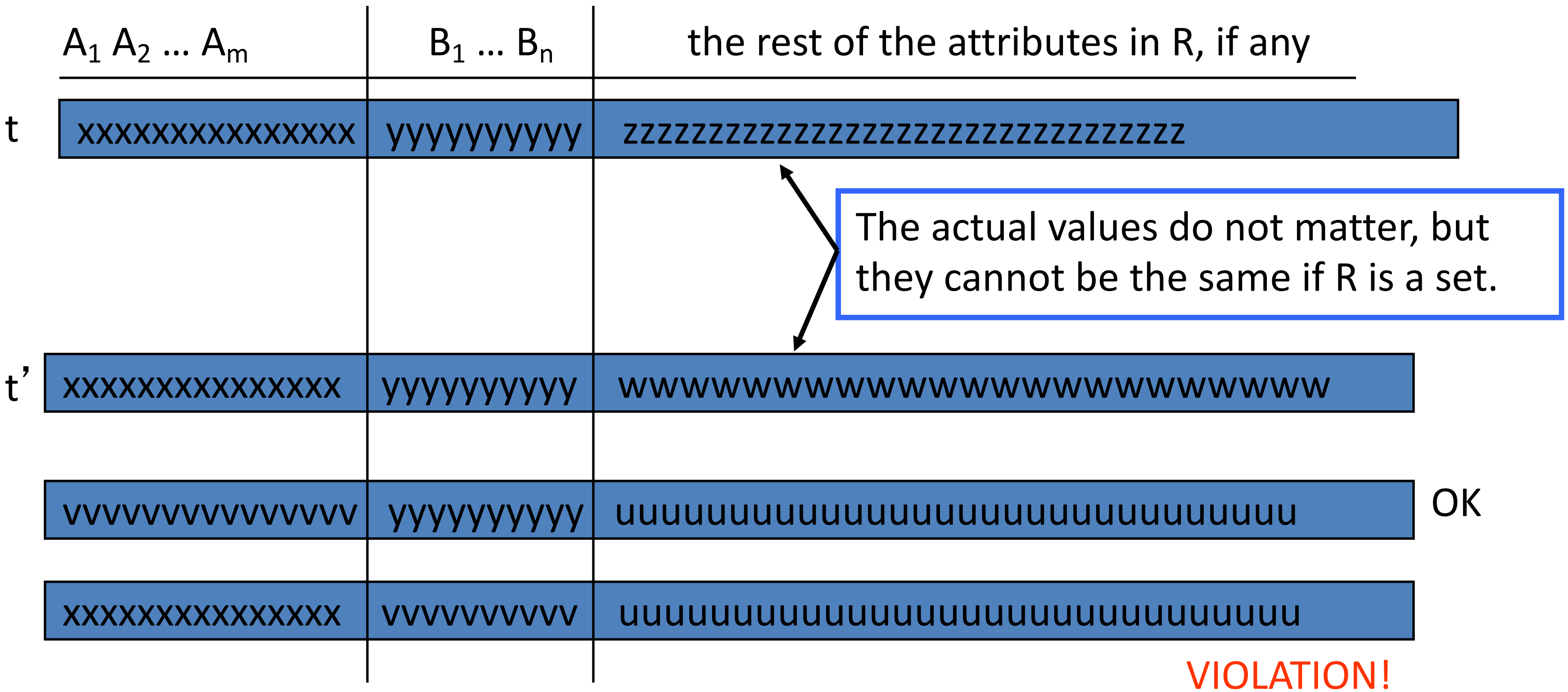
  where X and Y are non-empty subsets of attributes of **R**.
- A relation instance r of **R** *satisfies* the FD $X \rightarrow Y$ if

  for every pair of tuples t and t' in r, if $\underbrace{t[X]}$ = t'[X], then t[Y] = t'[Y]

Denotes the X value(s) of tuple t, i.e., project t on the attributes in X.

Illustration of the Semantics of an FD

- Relation schema R with the FD $A_1, \ldots, A_m \rightarrow B_1, \ldots, B_n$ where $\{ A_1, \ldots, A_m, B_1, \ldots, B_n \} \subseteq$ attributes(R).



More on Meaning of an FD

- Relation R satisfies $X \longrightarrow Y$
  - Pick any two (not necessarily distinct) tuples t and t' of an instance r of R. If t and t' agree on the X attributes, then they must also agree on the Y attributes.
  - The above must hold for *every possible instance* r of R.

- An FD is a statement about *all possible legal instances* of a schema. We cannot look at an instance (or even a set of instances) to determine which FDs hold.
  - Looking at an instance may help us determine that some FDs are <u>not</u> satisfied.

Reasoning about FDs

R(A,B,C,D,E)

Suppose $A \longrightarrow C$ and $C \longrightarrow E$. Is it also true that $A \longrightarrow E$?

In other words, suppose an instance r satisfies $A \longrightarrow C$ and $C \longrightarrow E$, is it true that r must also satisfy $A \longrightarrow E$?

YES

Proof: ?

Implication of FDs

- We say that a set $\mathcal{F}$ of FDs *implies* an FD F if for every instance r that satisfies $\mathcal{F}$, it must also be true that r satisfies F.
- Notation: $\mathcal{F} \vDash$ F

- Note that just finding some instance(s) r such that r satisfies $\mathcal{F}$ and r also satisfies F is not sufficient to prove that $\mathcal{F} \vDash$ F.

- How can we determine whether or not $\mathcal{F}$ implies F?

Armstrong's Axioms

- Use Armstrong's Axioms to determine whether or not $\mathcal{F} \vDash F$.
- Let X, Y, and Z denote sets of attributes over a relation schema R.
- **Reflexivity**: If $Y \subseteq X$, then $X \longrightarrow Y$.

    ssn, name $\longrightarrow$ name

  - FDs in this category are called *trivial FDs.*
- **Augmentation**: If $X \longrightarrow Y$, then $XZ \longrightarrow YZ$ for any set Z of attributes.

    ssn, name, addr $\longrightarrow$ name addr
- **Transitivity**: If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$.

    If ssn $\longrightarrow$ rank and rank $\longrightarrow$ sal_scale

    Then ssn $\longrightarrow$ sal_scale.

Union and Decomposition Rules

- **Union**: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
- **Decomposition**: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.

- Union and Decomposition rules are not essential. In other words, they can be derived using Armstrong's axioms.

- Derivation of the Union rule:

  (to fill in)

Union and Decomposition Rules

- **Union**: If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.
- **Decomposition**: If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.

- Union and Decomposition rules are not essential. In other words, they can be derived using Armstrong's axioms.

- Derivation of the Union rule:

  Since $X \rightarrow Z$, we get $XY \rightarrow YZ$ (augmentation)

  Since $X \rightarrow Y$, we get $X \rightarrow XY$ (augmentation)

  Therefore, $X \rightarrow YZ$ (transitivity)

Additional Rules

- Derivation of the Decomposition rule:

- We use the notation $\mathcal{F} \vdash F$ to mean that *F can be derived from* $\mathcal{F}$ *using Armstrong's axioms*.

Additional Rules

- Derivation of the Decomposition rule:

  $X \longrightarrow YZ$ (given)

  $YZ \longrightarrow Y$ (reflexivity)

  $YZ \longrightarrow Z$ (reflexivity)

  Therefore, $X \longrightarrow Y$ and $X \longrightarrow Z$ (transitivity).

- We use the notation $\mathcal{F} \vdash F$ to mean that *F can be derived from $\mathcal{F}$ using Armstrong's axioms*.
  - That's a lot of words, so we'll sometimes just read this as:
    "$\mathcal{F}$ generates F".
  - What was the meaning of $\mathcal{F} \vDash F$ ($\mathcal{F}$ implies F)?

Pseudo-Transitivity Rule

- **Pseudo-Transitivity**: If X $\rightarrow$ Y and WY $\rightarrow$ Z, then XW $\rightarrow$ Z.

- Can you derive this rule using Armstrong's axioms?

- Derivation of the Pseudo-Transitivity rule:

  (to fill in)

Pseudo-Transitivity Rule

- **Pseudo-Transitivity**: If $X \rightarrow Y$ and $WY \rightarrow Z$, then $XW \rightarrow Z$.

- Can you derive this rule using Armstrong's axioms?

- Derivation of the Pseudo-Transitivity rule:

  X -> Y and WY -> Z

  XW -> WY (augmentation)

  WY -> Z (given)

  Therefore XW -> Z (transitivity)

Completeness of Armstrong's Axioms

- **Completeness**: If a set $\mathcal{F}$ of FDs implies F, then F can be derived from $\mathcal{F}$ by applying Armstrong's axioms.
  - If $\mathcal{F} \models$ F, then $\mathcal{F} \vdash$ F.

    semantic syntactic

  - If $\mathcal{F}$ <u>implies</u> F, then one can prove F from $\mathcal{F}$ using Armstrong's axioms (i.e., $\mathcal{F}$ <u>generates</u> F).

For those familiar with Mathematical Logic:

- $\mathcal{F} \models$ F is "model-theoretic"
- $\mathcal{F} \vdash$ F is "proof-theoretic"

Soundness of Armstrong's Axioms

- **Soundness**: If F can be derived from a set of FDs $\mathcal{F}$ through Armstrong's axioms, then $\mathcal{F}$ implies F.
  - If $\mathcal{F} \vdash F$, then $\mathcal{F} \vDash F$.
    - That is, if $\mathcal{F}$ generates F, then $\mathcal{F}$ implies F.
  - Handwaving proof: If one can generate F from $\mathcal{F}$ using Armstrong's axioms, then surely $\mathcal{F}$ implies F. (<u>Why?</u>)

- With Completeness and Soundness, we know that
  $\mathcal{F} \vdash F$ if and only if $\mathcal{F} \vDash F$
  In other words, Armstrong's axioms generate precisely *all* the FDs that must hold under $\mathcal{F}$ (all the axioms that $\mathcal{F}$ implies).

- Great! But how can we decide whether or not $\mathcal{F}$ implies F?

Closure of a Set of FDs $\mathcal{F}$

- Let $\mathcal{F}^+$ denote the set of all FDs implied by a given set $\mathcal{F}$ of FDs.
  - Also called the closure of $\mathcal{F}$.
- To decide whether $\mathcal{F}$ implies F, first compute $\mathcal{F}^+$, then see whether F is a member of $\mathcal{F}^+$.
- Example: Compute $\mathcal{F}^+$ for the set { A $\rightarrow$ B, B $\rightarrow$ C} of FDs.
- Trivial FDs
  - A $\rightarrow$ A, B $\rightarrow$ B, C $\rightarrow$ C, AB $\rightarrow$ A, AB $\rightarrow$ B, BC $\rightarrow$ B, BC $\rightarrow$ C, AC $\rightarrow$ A, AC $\rightarrow$ C, ABC $\rightarrow$ A, ABC $\rightarrow$ B, ABC $\rightarrow$ C, ABC $\rightarrow$ AB , ABC $\rightarrow$ AC , ABC $\rightarrow$ BC, ABC $\rightarrow$ ABC
- Augmentation and transitivity (non-trivial FDs)
  - AC $\rightarrow$ B, AB $\rightarrow$ C
- Transitivity
  - A $\rightarrow$ C

Expensive and tedious!
Let’s find a better way.

<u>Attribute</u> Closure Algorithm

- Let X be a set of attributes and $\mathcal{F}$ be a set of FDs. *The attribute closure X^+ with respect to $\mathcal{F}$* is the set of all attributes A such that $X \rightarrow A$ is derivable from $\mathcal{F}$.
  - That is, all the attributes A such that $\mathcal{F} \vdash X \rightarrow A$

  <u>Input</u>: A set X of attributes and a set $\mathcal{F}$ of FDs.

  <u>Output</u>: X^+

  Closure = X; // initialize Closure to equal the set X
  repeat until no change in Closure {
   if there is an FD $U \rightarrow V$ in $\mathcal{F}$ such that $U \subseteq$ Closure,
   then Closure = Closure $\cup$ V;
  }
  return Closure;

If A $\in$ Closure (that is, if $A \in X^+$) , then $X \rightarrow A$.

More strongly, $\mathcal{F} \vdash X \rightarrow A$ if and only $A \in X^+$

FD Example 1 using Attribute Closure

- $\mathcal{F}$ = { A $\rightarrow$ B, B $\rightarrow$ C }.
- Question: Does A $\rightarrow$ C?
- Compute A^+

- Closure = { A }
- Closure = { A, B } (due to A $\rightarrow$ B)
- Closure = { A, B, C } (due to B $\rightarrow$ C)
- Closure = { A, B, C }
  - no change, stop
- Therefore A^+ = {A, B, C }
- Since C $\in$ A^+, answer YES.

FD Example 2 using Attribute Closure

- $\mathcal{F}$ = { AB $\rightarrow$ E, B $\rightarrow$ AC, BE $\rightarrow$ C }
- Question: Does BC $\rightarrow$ E?
- Compute BC^+

- Closure = { B, C }
- Closure = { A, B, C } (due to B $\rightarrow$ AC)
- Closure = { A, B, C, E } (due to AB $\rightarrow$ E)
- Closure = { A, B, C, E } (due to BE $\rightarrow$ C)
  – No change, so stop.
- Therefore BC^+ = {A,B,C,E}
- Since E $\in BC^+$, answer YES.

Algorithm for FDs
… and also for Keys/Superkeys

- To determine if an FD $X \rightarrow Y$ is implied by $\mathcal{F}$, compute X^+ and check if $Y \subseteq X^+$.
- Notice that Attribute Closure is less expensive to compute than $\mathcal{F}^+$.

- Algorithm can be modified to compute candidate keys. How?
  - Compute the closure of a single attribute in X^+. Then compute the closure of 2 attributes, 3 attributes and so on.
  - If the closure of a set of attributes contains all attributes of the relation, then it is a *superkey*.
  - If no proper subset of those attributes has a closure that contains all attributes of the relation, then it is a *key*.

Correctness of Algorithm

- Is it correct?

  Prove that the algorithm indeed computes X^+.

  - Show that for any attribute $A \in X^+$, it is the case that $X \rightarrow A$ is derivable from $\mathcal{F}$.
  - Show if $X \rightarrow A$ is derivable from $\mathcal{F}$, then it must be that $A \in X^+$.

Proof of Correctness

<u>Claim</u>: If $A \in X^+$, then $\mathcal{F} \vdash X \to A$.

Proof: By induction on the number of iterations in the attribute closure algorithm.

(to fill in)

Soundness and Completeness of the Attribute Closure Algorithm

- Soundness: Hence, if A in X^+, then $\mathcal{F} \vdash X \rightarrow A$. By the soundness of Armstrong's axioms, it follows that $\mathcal{F} \models F$.
- Is it true that if $\mathcal{F} \models F$, where F is the FD $X \rightarrow A$, then $A \in X^+$?

- Completeness.
- Claim: If that if $\mathcal{F} \models F$, where F is the FD $X \rightarrow A$, then we must have that $A \in X^+$.

Proof of Completeness

- Towards a contradiction, suppose $\mathcal{F} \vDash X \rightarrow A$ and $A \notin X^+$. Construct an instance I as follows. Define a relation r with attributes that consists of all attributes of $\mathcal{F}$. Create and instance r with two tuples t_1 and t_2 as follows:
  - Apply the attribute closure algorithm and set $t_1[X^+]=t_2[X^+]=1$. Then, insert 1 for every other attribute in t_1. Then, insert 0 for every other attribute in t_2.

- We first show that $r \vDash \mathcal{F}$. For every $U \rightarrow V \in \mathcal{F}$, either $t_1[U] \neq t_2[U]$ or $t_1[U]=t_2[U]$.
- In the former case, r trivially satisfies $U \rightarrow V$. For the latter case, if $t_1[U]=t_2[U]$ then we have $U \subseteq X^+$. Hence, it follows that the attribute closure algorithm would have added V to X^+. So $t_1[UV]=t_2[UV]$.
- Hence, $r \vDash \mathcal{F}$. However, since $A \notin X^+$, it follows that $r \nvDash X \rightarrow A$ thus contradicting our assumption that $\mathcal{F} \vDash X \rightarrow A$. (That is, r is a counter example to $\mathcal{F} \vDash X \rightarrow A$.)

Practice Homework 6

1. Let R(A,B,C,D,E) be a relation schema and let $\mathcal{F}$ = { AB $\rightarrow$ E, B $\rightarrow$ AC, BE $\rightarrow$ C } be a set of FDs that hold over R.

   a. Prove that $\mathcal{F} \vDash$ B $\rightarrow$ E using Armstrong's axioms.
   b. Compute the closure of B. That is, compute B^+.
   c. Give a key for R. Justify why your answer is a key for R.
   d. Show an example relation that satisfy $\mathcal{F}$.
   e. Show an example relation that does not satisfy $\mathcal{F}$.

2. Let R(A,B,C,D,E) be a relation schema and let $\mathcal{F}$ = { A $\rightarrow$ C, B $\rightarrow$ AE, B $\rightarrow$ D, BD $\rightarrow$ C } be a set of FDs that hold over R.

   a. Show that B $\rightarrow$ CD using Armstrong's axioms.
   b. Show a relation of R such that R satisfies $\mathcal{F}$ but R does not satisfy A $\rightarrow$ D.
   c. Is AB a key for R?