Functional Dependency

8.Functional Dependency

A "good" database schema should not lead to *update anomalies*.

- update anomalies,
- functional dependencies,
- Armstrong Axioms,
- closures.

8.1 Update Anomalies

- Redundancy in a database means storing a piece of data more than once.
- Redundancy is often useful for efficiency and semantic reasons, but creates the potential for consistency problems.
- A poor *redundancy control* may cause update anomalies.
- Consider the example relation below (adapted from "An Introduction to Database Systems" by Desai):

STUDENTS

Name	Course	Phone_no	Major	Prof	Grade
Jones	353	$\boxed{237\text{-}4539}$	Comp Sci	Smith	A
Ng	329	427-7390	Chemistry	Turner	B
Jones	328	$\boxed{237\text{-}4539}$	Comp Sci	Clark	B
Martin	456	388-5183	Physics	James	A
Dulles	293	371-6259	Decision Sci	Cook	C
Duke	491	823-7293	Mathematics	Lamb	B
Duke	356	823-7293	Mathematics	Bond	UN
Jones	492	$\boxed{237\text{- }4539}$	Comp Sci	Cross	UN
~~Baxter~~	~~379~~	~~839-0827~~	~~English~~	~~Broes~~	~~C~~

- *Modification anomalies*: e.g. Jones's phone number appears 3 times. When a phone number is changed, it must be changed in all 3 places, or the data will be inconsistent.

Insertion anomalies:

• If Jones enrolls in another course, and a different phone number is entered, again the data will be inconsistent.

• Also, if the only way that the association between course and professor is stored in this relation, we can only enter the association when someone enrolls in the course.

Deletion anomalies: If the last student in a course is deleted, the association between professor and course is lost.

8.2 Functional dependencies

A function *f* from S_1 to S_2 has the property

$$if\ x, y \in S_1 \text{ and } x = y, \text{then } f(x) = f(y).$$

A generalization of keys to avoid design flaws violating the above rule.

Let X and Y be sets of attributes in R.

X (*functionally*) determines *Y* , $X \rightarrow Y$, iff $t_1[X] = t_2[X]$ implies $t_1[Y] = t_2[Y]$.

i.e., f (t(X)) = t [Y]

We also say $X \rightarrow Y$ is a *functional* dependency, and that Y is *functionally* dependent on X.

X is called the *left side*, Y the *right side* of the dependency.

Examples:

• For every Name, there is a unique Phone_no and Major, assume Name is unique

• For every Course, there is a unique Prof

• For every Name and Course, there is a unique Grade

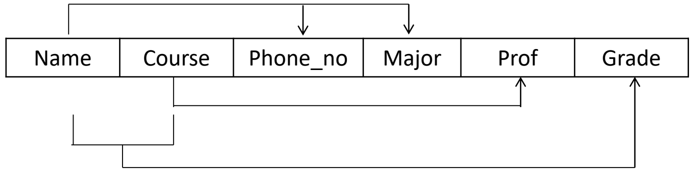
- In this example:

$$\{Name\} \rightarrow \{Phone\_no\ , Major\}$$

$$\{Course\} \rightarrow \{Prof\}$$

$$\{Name\ , Course\} \rightarrow \{Grade\}$$

- We can also show these in a diagram like this one:



- Notice that other FD's follow from these:

$$\{Name\} \rightarrow \{Major\}$$

$$\{Course\ , Grade\} \rightarrow \{Prof\ , Grade\}$$

- Let F be a set of FD's.

Definition 1: $X \to Y$ is inferred from F (or that F infers $X \to Y$), written in

$$F \models X \to Y$$

- if any relation instance satisfying F must also satisfy $X \to Y$.
- Impossible to list every relation to verify if $X \to Y$ is inferred from F.
- A set ρ of derivation rules are required, such that:

  a $X \to Y$ is inferred from F according to Definition 1 iff it can be derived using ρ.

8.3 Armstrong's axioms (1974)

Notation: If X and Y are sets of attributes, we write XY for their union.

e.g. X = {A, B}, Y = {B, C}, XY = {A, B, C}

- F1 (Reflexivity) If $X \supseteq Y$ then $X \to Y$.
- F2 (Augmentation) $\{X \to Y\} \models XZ \to YZ$.
- F3 (Transitivity) $\{X \to Y, Y \to Z\} \models X \to Z$.

- F4 (Additivity) $\{X \to Y, X \to Z\} \models X \to YZ$.
- F5 (Projectivity) $\{X \to YZ\} \models X \to Y$.
- F6 (Pseudotransitivity)

  $\{X \to Y, YZ \to W\} \models XZ \to W$.

Example: Given F = $\{A \to B, A \to C, BC \to D\}$, derive $A \to D$:

1. $A \to B$ (given)
2. $A \to C$ (given)
3. $A \to BC$ (by F4, from 1 and 2)
4. $BC \to D$ (given)
5. $A \to D$ (by F3, from 3 and 4)

- F4 (Additivity) $\{X \to Y, X \to Z\} \models X \to YZ$.
- F5 (Projectivity) $\{X \to YZ\} \models X \to Y$.
- F6 (Pseudotransitivity)

  $\{X \to Y, YZ \to W\} \models XZ \to W$.

In fact, F4, F5, and F6 can be derived from F1-F3.

Example: Prove $\{X \to Y, X \to Z\} \models X \to YZ$.

1) $X \to Y$ is given.

2) $XX \to XY$ (by F2); that is, $X \to XY$

3) $X \to Z$ is given.

4) $XY \to YZ$ (by F2)

5) $X \to YZ$ (by F3, 2) and 4))

We can prove that Armstrong's axioms are sound and complete:

- Sound: if F derives $A \to B$ by using Armstrong's axioms, then F |= $A \to B$ by Definition 1.
- Complete: if F |= $M \to N$ by Definition 1, then F derives $M \to N$ by using Armstrong's axioms.

8.4 Algorithm to Check a FD

Given F, how do we check if $X \to Y$ is in F^+?

F^+ denotes the smallest set of FD's that

- contains F, and
- is *closed* under Armstrong's axioms.

F^+ is the *closure* of F.

F = { $A \to B$, $B \to C$, $A \to C$ }

F⁺ = {AB -> A, AB -> B, AB -> C, AC -> A, AC -> B, AC -> C, ABC -> A, ABC -> B, ABC -> C, AB -> AB, AB -> BC, AB -> AC, …….}

F^+ always has an exponential size regarding |F|.

- Too expensive to compute F^+ to verify a membership.
- Instead we can compute the *closure* X^+ of X under F, X^+ is the largest set of attributes functionally determined by X.

It can be proven (using additivity) that

S1: $X^+ = \cup_{\forall X \to A \in F^+} A.$

S2: $X \to Y \in F^+$ iff (if and only if) $Y \subseteq X^+$.

$F = \{ A \rightarrow B, BC \rightarrow D, A \rightarrow C \}$, compute $\{A\}^+$

1st scan of F:

$X^+ := \{A\}$

$X^+ := \{A, B\}$

$X^+ := \{A, B, C\}$

2nd scan of F:

$X^+ := \{A, B, C, D \}$

3rd scan of F: no change, therefore the algorithm terminates.

$\{A\}^+ := \{A, B, C, D \}$

- Algorithm to compute X^+:

$X^+ := X$;
change := true;
while change do
    begin
    change := false;
    for each FD $W \rightarrow Z$ in F do
        begin
        if ($W \subseteq X^+$) and ($\not\subseteq$ X^+) then do
            begin
            $X^+ := X^+ \cup Z$;
            change := true;
            end
        end
    end

8.5 Algorithm to Compute a Candidate Key

- Given a relational schema R and a set F of functional dependencies on R.
- A key X of R must have the property that $X^+ = R$.

- **Algorithm to compute a candidate key**

Step 1: Assign X a superkey in F.

Step 2: Iteratively remove attributes from X while retaining the property $X^+ = R$ till no reduction on X.

The remaining X is a key.

$R = \{A, B, C, D\}$ and $F = \{ A \rightarrow B, BC \rightarrow D, A \rightarrow C \}$

$X = \{A, B, C\}$ if the left hand side of F is a super key.

A cannot be removed because $\{BC\}^+ = \{B, C, D\} \neq R$

B can be removed because $\{AC\}^+ = \{A, B, C, D\} = R$

$\Rightarrow$ $X = \{ A, C\}$

C can be further removed because $\{A\}^+ = \{A, B, C, D\}$

$\Rightarrow$ $X = \{A\}$