DM505 Database Design and Programming
DM576 Database Systems

Panagiotis Tampakis

ptampakis@imada.sdu.dk

Functional Dependencies

Functional Dependencies

- $X \rightarrow Y$ is an assertion about a relation R that whenever two tuples of R agree on all the attributes of X, then they must also agree on all attributes in set Y
  - Say "$X \rightarrow Y$ holds in R"
  - Convention: …, X, Y, Z represent sets of attributes; A, B, C,… represent single attributes
  - Convention: no set formers in sets of attributes, just ABC, rather than $\{A,B,C\}$

Functional Dependencies

- Example

title	*year*	*length*	*genre*	*studioName*	*starName*
Star Wars	1977	124	SciFi	Fox	Carrie Fisher
Star Wars	1977	124	SciFi	Fox	Mark Hamill
Star Wars	1977	124	SciFi	Fox	Harrison Ford
Gone With the Wind	1939	231	drama	MGM	Vivien Leigh
Wayne's World	1992	95	comedy	Paramount	Dana Carvey
Wayne's World	1992	95	comedy	Paramount	Mike Meyers

title year → length genre studioName **TRUE**

title year → starName **FALSE**

Splitting Right Sides of FDs

- $X \rightarrow A_1 A_2 \ldots A_n$ holds for R exactly when each of $X \rightarrow A_1$, $X \rightarrow A_2$,..., $X \rightarrow A_n$ hold for R
- Example: $A \rightarrow BC$ is equivalent to $A \rightarrow B$ and $A \rightarrow C$
- There is no splitting rule for left sides

Example: FDs

Drinkers(name, addr, beersLiked, manf, favBeer)

- Reasonable FDs to assert:
  1. name → addr favBeer
     - Note: this FD is the same as name → addr and name → favBeer
  2. beersLiked → manf

Example: Possible Data

name	addr	beersLiked	manf	favBeer
Peter	Campusvej	Odense Cl.	Albani	Erdinger W.
Peter	Campusvej	Erdinger W.	Erdinger	Erdinger W.
Lars	NULL	Odense Cl.	Albani	Odense Cl.

Because name → addr

Because name → favBeer

Because beersLiked → manf

Keys of Relations

- *K* is a *superkey* for relation *R* if *K* functionally determines all of *R*
- *K* is a *key* for *R* if *K* is a superkey, but no proper subset of *K* is a superkey

Example: Superkey

Drinkers(name, addr, beersLiked, manf, favBeer)

- {name, beersLiked} is a superkey because together these attributes determine all the other attributes
  - name → addr favBeer
  - beersLiked → manf

Example: Key

- {name, beersLiked} is a key because neither {name} nor {beersLiked} is a superkey
  - name doesn't → manf
  - beersLiked doesn't → addr
- There are no other keys, but lots of superkeys
  - Any superset of {name, beersLiked}

Where Do Keys Come From?

1. Just assert a key *K*
   - The only FDs are $K \rightarrow A$ for all attributes *A*
2. Assert FDs and deduce the keys by systematic exploration

More FDs From "Physics"

- Example:
  "no two courses can meet in the same room at the same time" tells us:
  - hour room → course

Reasoning About FDs

- We are given FDs $X_1 \rightarrow A_1$, $X_2 \rightarrow A_2$,..., $X_n \rightarrow A_n$, and we want to know whether an FD $Y \rightarrow B$ must hold in any relation that satisfies the given FDs
  - Splitting/Combining Rule

Example 3.5: In Example 3.1 the set of FD's:

```
title year → length
title year → genre
title year → studioName
```

is equivalent to the single FD:

```
title year → length genre studioName
```

that we asserted there. □

Reasoning About FDs

- Transitive Rule
  - Example: If $A \rightarrow B$ and $B \rightarrow C$ hold, surely $A \rightarrow C$ holds, even if we don't say so
- Important for design of good relation schemas

Reasoning About FDs

- Given two sets of FD's *S* and *T*:
  - *S* and *T* are *equivalent*
    - if the set of relation instances satisfying *S* is exactly the same as the set of relation instances satisfying *T.*
    - *E.g.* $A \rightarrow B$, $A \rightarrow C$ and $A \rightarrow D$ are equivalent with $A \rightarrow B\ C\ D$.

S	T
$A \rightarrow B$ $A \rightarrow C$ $A \rightarrow D$	$A \rightarrow B\ C\ D$

Reasoning About FDs

- Given two sets of FD's *S* and *T*:
  - *S follows* from *T*
    - if every relation instance that satisfies all the FD's in *T* also satisfies all the FD's in *S.*
    - *E.g.* $A \rightarrow C$ follows from $A \rightarrow B$ and $B \rightarrow C$.

S	T
$A \rightarrow C$	$A \rightarrow B$ $B \rightarrow C$

Trivial FD's

- A constraint of any kind on a relation is said to be *trivial*
  - if it holds for every instance of the relation, regardless of what other constraints are assumed.
- A trivial FD has a right side that is a subset of its left side
  - E.g. $A\ B \rightarrow A$

Closure

- Given
  - a set of FD's *S*, a set of attributes $A=\{A_1,A_2,\ldots,A_n\}$ and a set of attributes $B=\{B_1,B_2,\ldots,B_n\}$
- The *closure* of attributes *A*, denoted A^+ is *B*
  - *Iff* every relation that satisfies all the FD's in set *S* also satisfies $A_1A_2\ldots A_n \rightarrow B$.
    - In simple words, we want to find all attributes B that are determined by attributes $A_1A_2\ldots A_n$

Closure

- Algorithm
  1. If necessary, split the FD's of S, so each FD in S has a single attribute on the right.
  2. Initialize $A^+ = A$, i.e., $A^+ = \{A_1, A_2, \ldots, A_n\}$
  3. Repeatedly expand A^+ and add new attributes by using FD's that follow from A^+
  4. Until there are no more attributes to add to A^+

Closure

- Example
  - Let us consider a relation with attributes *A, B, C, D, E* and *F.*
  - Suppose that this relation has the FD's
    - $AB \rightarrow C$,
    - $BC \rightarrow AD$,
    - $D \rightarrow E$,
    - $CF \rightarrow B$.
  - Find the closure of $\{A,B\}^+$

Closure

- Example
  - $\{A,B\}^+ = \{A,B\} = \{A,B,C\} = \{A,B,C,D\} = \{A,B,C,D,E\}$

Closure

- Usage
  - By computing the closure of any set of attributes,
    - we can test whether any given FD $A_1A_2...A_n \rightarrow B$ follows from a set of FD's *S.*
    - We can identify all candidate keys and superkeys of a relation

Exercise

$R(A, B, C, D)$ with FD's $AB \rightarrow C$, $BC \rightarrow D$, $CD \rightarrow A$, and $AD \rightarrow B$.

a) What are all the nontrivial FD's that follow from the given FD's? You should restrict yourself to FD's with single attributes on the right side.

b) What are all the keys of R?

c) What are all the superkeys for R that are not keys?

Closing Sets of FD's

- The problem:
  - Which FD's we use to represent the full set of FD's for a relation?
- Given a set of FD's *S* (the FD's that hold in a given relation)
  - any set of FD's equivalent to *S* is said to be a *basis* for *S.*
  - we shall limit ourselves to only *bases* whose FD's have singleton right sides.

Closing Sets of FD's

- A *minimal basis* for a relation is a basis *B* that satisfies three conditions:
  1. All the FD's in *B* have singleton right sides.
  2. If any FD is removed from *B,* the result is no longer a basis.
  3. If for any FD in *B* we remove one or more attributes from the left side of *F,* the result is no longer a basis.

Closing Sets of FD's

- Example
  - R(A,B,C)
  - A→B, A→C, B→A, B→C, C→A, C→B
  - AB→C, AC→B, BC→A
  - A→BC
  - A→A

Projection

- Suppose we have a relation *R* with set of FD's *S*,
  - Suppose we want to project *R* by computing $R_1 = \pi_L(R)$, for some list of attributes of *R*
  - What FD's hold in R_1?
  - All FD's that:
    - Follow from *S*, and
    - Involve only attributes of R_1

Projection

- Algorithm
  1. Initialize $T = \emptyset$
  2. For each set of attributes X that is a subset of the attributes of R_1,
     1. compute X^+ w.r.t the set of FD's S
     2. add to T all non-trivial FD's of X^+
        - Keep only FD's whose right side belong to R_1
  3. Calculate the *minimal basis* of T.

Example: Projecting FDs

- *ABC* with FDs $A \rightarrow B$ and $B \rightarrow C$
  Project onto *AC:*
  - $A^+ = ABC$; yields $A \rightarrow B$, $A \rightarrow C$
  - $C^+ = C$; yields nothing
  - $AC^+ = ABC$; see A^+
- Resulting FDs: $A \rightarrow B$, $A \rightarrow C$

Exercise

Let us consider the following relation R(A,B,C,D) with FD's:

- A→B
- B→C
- C→D

Find the FD's that hold for $\pi_{A, C, D}(R)$

Summary 6

More things you should know:

- Functional Dependencies
- Key, Superkey
- Closure
- Projecting FDs