Databases

Lecture 6

Functional Dependencies. Normal Forms (III)

Sabina S. CS

* See recap lecture example with schema decomposition.

- R[A] - a relation
- F - a set of functional dependencies
- α – a subset of attributes

- problems

I. compute the closure of F: F^+

II. compute the closure of a set of attributes under a set of functional dependencies, e.g., the closure of α under F: α^+

III. compute the minimal cover for a set of dependencies

- R[A] - a relation
- F - a set of functional dependencies
- problems

I. compute the closure of F: F^+

- the set F^+ contains all the functional dependencies implied by F
- F implies a functional dependency *f* if *f* holds on every relation that satisfies F
- the following 3 rules can be repeatedly applied to compute F^+ (Armstrong's Axioms):
  - α, β, γ - subsets of attributes of A
  1. reflexivity: if $\beta \subseteq \alpha$, then $\alpha \to \beta$
  2. augmentation: if $\alpha \to \beta$, then $\alpha\gamma \to \beta\gamma$
  3. transitivity: if $\alpha \to \beta$ and $\beta \to \gamma$, then $\alpha \to \gamma$
- these rules are complete (they generate all dependencies in the closure) and sound (no erroneous functional dependencies can be derived)

- R[A] - a relation
- F - a set of functional dependencies
- problems

I. compute the closure of F: F^+

- the following rules can be derived from Armstrong's Axioms:

4. union: if $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$, then $\alpha \rightarrow \beta\gamma$

$$\left.\begin{array}{l} \alpha \rightarrow \beta \underset{\text{augmentation}}{=>} \alpha\alpha \rightarrow \alpha\beta \\ \\ \alpha \rightarrow \gamma \underset{\text{augmentation}}{=>} \alpha\beta \rightarrow \beta\gamma \end{array}\right\} \underset{\text{transitivity}}{=>} \quad \alpha \rightarrow \beta\gamma$$

- R[A] - a relation
- F - a set of functional dependencies
- problems

I. compute the closure of F: F^+

- the following rules can be derived from Armstrong's Axioms:

5. decomposition: if $\alpha \rightarrow \beta\gamma$, then $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$

$$\left.\begin{array}{l} \alpha \rightarrow \beta\gamma \\ \beta\gamma \rightarrow \beta \text{ (reflexivity)} \end{array}\right\} \underset{\text{transitivity}}{=>} \alpha \rightarrow \beta$$

($\alpha \rightarrow \gamma$ can similarly be shown to hold)

- R[A] - a relation
- F - a set of functional dependencies
- problems

I. compute the closure of F: F^+

- the following rules can be derived from Armstrong's Axioms:

6. pseudotransitivity: if $\alpha \to \beta$ and $\beta\gamma \to \delta$, then $\alpha\gamma \to \delta$

$$\left.\begin{array}{l} \alpha \to \beta => \alpha\gamma \to \beta\gamma \\ \beta\gamma \to \delta \end{array}\right\} \underset{\text{transitivity}}{=>} \quad \alpha\gamma \to \delta$$

- $\alpha, \beta, \gamma, \delta$ - subsets of attributes of A

- R[A] - a relation
- F - a set of functional dependencies
- α – a subset of attributes
- problems

II. compute the closure of a set of attributes under a set of functional dependencies

- determine the closure of α under F, denoted as α^+
- α^+ - the set of attributes that are functionally dependent on α under F

- R[A] - a relation
- F - a set of functional dependencies
- α – a subset of attributes
- problems

II. compute the closure of a set of attributes under a set of functional dependencies

- algorithm

```
closure := α;
repeat until there is no change:
   for every functional dependency β → γ in F
      if β ⊆ closure
         then closure := closure ∪ γ;
```

- $R[A]$ - a relation
- F - a set of functional dependencies
- problems

III. compute the minimal cover for a set of dependencies

Definition: F, G - two sets of functional dependencies; F and G are equivalent (notation $F \equiv G$) if $F^+ = G^+$.

- R[A] - a relation
- F - a set of functional dependencies
- problems

III. compute the minimal cover for a set of dependencies

Definition: F - set of functional dependencies; a minimal cover for F is a set of functional dependencies F_M that satisfies the following conditions:

1. $F_M \equiv F$
2. the right side of every dependency in F_M has a single attribute;
3. the left side of every dependency in F_M is irreducible (i.e., no attribute can be removed from the determinant of a dependency in F_M without changing F_M's closure);
4. no dependency *f* in F_M is redundant (no dependency can be discarded without changing F_M's closure).

* closure of a set of functional dependencies

P1. Let $R[ABCDEF]$ be a relational schema and S a set of functional dependencies over R, $S = \{A \to B, A \to C, CD \to E, CD \to F, D \to E\}$.

Show the following FDs are in S^+: $A \to BC, CD \to EF, AD \to E, AD \to F$.

$$\left.\begin{array}{l} A \to B \\ A \to C \end{array}\right\} \underset{\text{union}}{=>} A \to BC$$

$$\left.\begin{array}{l} CD \to E \\ CD \to F \end{array}\right\} \underset{\text{union}}{=>} CD \to EF$$

$$\left.\begin{array}{l} A \to C \underset{\text{augmentation}}{=>} AD \to CD \\ \qquad\qquad\qquad\qquad CD \to E \end{array}\right\} \underset{\text{transitivity}}{=>} AD \to E$$

* <u>closure of a set of functional dependencies</u>

P1. Let $R[ABCDEF]$ be a relational schema and S a set of functional dependencies over R, $S = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, CD \rightarrow F, D \rightarrow E\}$.

Show the following FDs are in S^+: $A \rightarrow BC, CD \rightarrow EF, AD \rightarrow E, AD \rightarrow F$.

$$\left.\begin{array}{l} A \rightarrow C \\ CD \rightarrow F \end{array}\right\} \Rightarrow AD \rightarrow F$$

pseudotransitivity

* closure of a set of attributes under a set of functional dependencies

P2. Let $R[ABCDEF]$ be a relational schema, S a set of functional dependencies over R and α a subset of attributes of the set of attributes of R, $S = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, CD \rightarrow F, D \rightarrow E\}$, $\alpha = \{A, D\}$. Compute α^+.

$\alpha^+ = \{A, D\}$

$A \rightarrow B$ => $\alpha^+ = \{A, B, D\}$

$A \rightarrow C$ => $\alpha^+ = \{A, B, C, D\}$

$CD \rightarrow E$ => $\alpha^+ = \{A, B, C, D, E\}$

$CD \rightarrow F$=> $\alpha^+ = \{A, B, C, D, E, F\}$

$D \rightarrow E$, E already in α^+

- iterate over all dependencies one more time, α^+ remains unchanged
- $\alpha^+ = \{A, B, C, D, E, F\}$

* minimal cover for a set of functional dependencies

P3. Let $R[ABCD]$ be a relational schema and S a set of functional dependencies over R, S={$A \to BC$, $B \to C$, $A \to B$, $AB \to C$, $AC \to D$}. Compute a minimal cover of S.

- decomposition: $A \to BC$ => $A \to B$, $A \to C$

=> $A \to B$

$A \to C$

$B \to C$

$\cancel{A \to B}$ - can be eliminated

$AB \to C$

$AC \to D$

* minimal cover for a set of functional dependencies

P3. Let $R[ABCD]$ be a relational schema and S a set of functional dependencies over R, S={$A \to BC$, $B \to C$, $A \to B$, $AB \to C$, $AC \to D$}. Compute a minimal cover of S.

- augmentation: $A \to C$ => $A \to AC$
- transitivity: $A \to AC$, $AC \to D$ => $A \to D$

  => C in $AC \to D$ is redundant

=> $A \to B$

$A \to C$

$B \to C$

$AB \to C$

$A \to D$

* minimal cover for a set of functional dependencies

P3. Let $R[ABCD]$ be a relational schema and S a set of functional dependencies over R, S={$A \to BC$, $B \to C$, $A \to B$, $AB \to C$, $AC \to D$}. Compute a minimal cover of S.

- augmentation: $A \to C$ => $AB \to CB$
- decomposition: $AB \to CB$ => $AB \to C$

  => can eliminate $AB \to C$

=> $A \to B$
$A \to C$
$B \to C$
$A \to D$

* <u>minimal cover for a set of functional dependencies</u>

P3. Let $R[ABCD]$ be a relational schema and S a set of functional dependencies over R, S={$A \to BC$, $B \to C$, $A \to B$, $AB \to C$, $AC \to D$}. Compute a minimal cover of S.

- transitivity: $A \to B$, $B \to C$ => $A \to C$

  => can eliminate $A \to C$

=> $A \to B$

$B \to C$

$A \to D$

Example 11. Consider relation DFM[Department, FacultyMembers, MeetingDates], with repeating attributes *FacultyMembers* and *MeetingDates*.

- a possible instance is given below:

Department	FacultyMembers	MeetingDates
Computer Science	FCS1 FCS2 … FCSm	DCS1 DCS2 … DCSn
Mathematics	FM1 FM2 … FMp	DM1 DM2 … DMq

- eliminate repeating attributes (such that the relation is at least in 1NF) - replace DFM by a relation DFM' in which *FacultyMember* and *MeetingDate* are scalar attributes:

Department	FacultyMember	MeetingDate
Computer Science	FCS1	DCS1
Computer Science	FCS1	DCS2
...	...	...
Computer Science	FCS1	DCSn
Computer Science	FCS2	DCS1
Computer Science	FCS2	DCS2
...	...	...
Mathematics	FM1	DM1
...	...	...
Mathematics	FMp	DMq

Department	FacultyMember	MeetingDate
Computer Science	FCS1	DCS1
Computer Science	FCS1	DCS2
…	…	…
Computer Science	FCS1	DCSn
Computer Science	FCS2	DCS1
Computer Science	FCS2	DCS2
…	…	…
Mathematics	FM1	DM1
…	…	…
Mathematics	FMp	DMq

- in this table, each faculty member has the same meeting dates
- therefore, when adding / changing / removing rows, additional checks must be carried out

- a simple functional dependency $\alpha \to \beta$ means, by definition, that every value u of α is associated with a unique value v for β

Definition. Let $R[A]$ be a relation with the set of attributes $A = \alpha \cup \beta \cup \gamma$. The multi-valued dependency $\alpha \rightrightarrows \beta$ (read α *multi-determines* β) is said to hold over R iff each value u of α is associated with a set of values v for β: $\beta(u) = \{v_1, v_2, \dots, v_n\}$, and this association holds regardless of the values of γ.

->

- obs. $\sigma_{\alpha=u}(R)$ produces a relation that contains the tuples of R where $\alpha = u$

- let $R[A]$ be a relation, $\alpha \rightrightarrows \beta$ a multi-valued dependency, and $A = \alpha \cup \beta \cup \gamma$, with γ a non-empty set
- the association among the values in $\beta(u)$ for β and the value u of α holds regardless of the values of γ (the context)
- i.e., these associations (between u and an element in $\beta(u)$) exist for any value w in γ:
  - $\forall w \in \Pi_\gamma\left(\sigma_{\alpha=u}(R)\right),\ \exists r_1, r_2, \dots, r_n$ such that $\Pi_\alpha(r_i) = u$, $\Pi_\beta(r_i) = v_i$, $\Pi_\gamma(r_i) = w$

- if $\alpha \rightrightarrows \beta$ and the following rows exist:

α	β	γ
u_1	v_1	w_1
u_1	v_2	w_2

then the following rows must exist as well:

α	β	γ
u_1	v_1	w_2
u_1	v_2	w_1

Property. Let $R[A]$ be a relation, $A = \alpha \cup \beta \cup \gamma$. If $\alpha \rightrightarrows \beta$, then $\alpha \rightrightarrows \gamma$.

Justification.

- Let u be a value of α in R.
- Let $\beta(u) = \Pi_\beta\left(\sigma_{\alpha=u}(R)\right)$, $\gamma(u) = \Pi_\gamma\left(\sigma_{\alpha=u}(R)\right)$ (the β and γ values in the tuples where $\alpha = u$).

Since $\alpha \rightrightarrows \beta \Rightarrow$

$\forall w \in \gamma(u), \forall v \in \beta(u), \exists r = (u, v, w)$, or

$\forall v \in \beta(u), \forall w \in \gamma(u), \exists r = (u, v, w)$,

therefore $\alpha \rightrightarrows \gamma$.

- for relation DFM' (in the previous example):

{Department} $\rightrightarrows$ *{FacultyMember}*, *{Department}* $\rightrightarrows$ *{MeetingDate}*

Definition. A relation R is in 4NF iff, for every multi-valued dependency $\alpha \rightrightarrows \beta$ that holds over R, one of the statements below is true:

- $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$, or
- α is a superkey.

- trivial multi-valued dependency $\alpha \rightrightarrows \beta$ in relation R: $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$

- if $R[\alpha, \beta, \gamma]$ and $\alpha \rightrightarrows \beta$ (non-trivial, α not a superkey), R is decomposed into the following relations:

$R_1[\alpha, \beta] = \Pi_{\alpha \cup \beta}(R)$

$R_2[\alpha,\gamma] = \Pi_{\alpha \cup \gamma}(R)$

- relation DFM' is decomposed into:
  DF [Department, FacultyMember]
  DM [Department, MeetingDate]

Example 12. Consider relation FaPrCo[FacultyMember, Program, Course], storing the programs and courses for different faculty members

- its key is {FacultyMember, Program, Course}
- this relation has no nontrivial functional dependencies or multi-valued dependencies, it’s in 4NF
- consider the following data in the relation:

Fa	Pr	Co
F1	P1	C2
F1	P2	C1
F2	P1	C1
F1	P1	C1

- the relation cannot be decomposed into 2 relations (via projection), because new data would be introduced through the join
- this claim can be justified by considering the three possible projections on two attributes:

FaPr	Fa	Pr
	F1	P1
	F1	P2
	F2	P1

FaCo	Fa	Co
	F1	C2
	F1	C1
	F2	C1

PrCo	Pr	Co
	P1	C2
	P2	C1
	P1	C1

- when evaluating $\mathrm{FaPr} * \mathrm{PrCo}$, the following data is obtained:

$\mathbf{R' = FaPr * PrCo}$	**Fa**	**Pr**	**Co**
	F1	P1	C2
	F1	P1	C1
	F1	P2	C1
	F2	P1	C2
	F2	P1	C1

- this result set contains an extra tuple, which didn't exist in the original relation
- the same is true for the other join combinations: $\mathrm{FaPr} * \mathrm{FaCo}$ and $\mathrm{PrCo} * \mathrm{FaCo}$

- when evaluating $\mathrm{R}' * \mathrm{FaCo}$ (i.e., $\mathrm{FaPr} * \mathrm{PrCo} * \mathrm{FaCo}$), the original relation FaPrCo is obtained
- conclusion: FaPrCo cannot be decomposed into 2 projections, but it can be decomposed into 3 projections, i.e., FaPrCo is *3-decomposable*:

$$\mathrm{FaPrCo} = \mathrm{FaPr} * \mathrm{PrCo} * \mathrm{FaCo}, \text{ or } \mathrm{FaPrCo} = *(\mathrm{FaPr}, \mathrm{PrCo}, \mathrm{FaCo})$$

- this conclusion (FaPrCo is 3-decomposable) is true for the data in the relation
- 3-decomposability can be specified as a constraint:

** if (F1, P1) ∈ FaPr and (F1, C1) ∈ FaCo and (P1, C1) ∈ PrCo then (F1, P1, C1) ∈ FaPrCo*

- this restriction can be expressed on FaPrCo (all legal instances must satisfy the constraint):

** if (F1, P1, C2) ∈ $FaPrCo$ and (F1, P2, C1) ∈ $FaPrCo$ and (F2, P1, C1) ∈ $FaPrCo$ then (F1, P1, C1) ∈ $FaPrCo$*

- consider the following relation instance:

Fa	Pr	Co
F1	P1	C2
F1	P2	C1

- if the previous restriction holds, then, if (F2, P1, C1) is added to the relation, (F1, P1, C1) must be also added:

Fa	Pr	Co
F1	P1	C2
F1	P2	C1
F2	**P1**	**C1**
F1	**P1**	**C1**

* what if (F1, P1, C1) is removed from the instance?

Definition. Let $R[A]$ be a relation and $R_i[\alpha_i]$, i=1,2, ...,m, the projections of R on α_i. R satisfies the join dependency $* \{\alpha_1, \alpha_2, \ldots, \alpha_\mathrm{m}\}$ iff $R = R_1 * R_2 * \cdots * R_m$.

- FaPrCo has a join dependency (FaPrCo = FaPr $*$ PrCo $*$ FaCo)

Definition. Relation R is in 5NF iff every non-trivial JD is implied by the candidate keys in R.

- JD $* \{\alpha_1, \alpha_2, \ldots, \alpha_\mathrm{m}\}$ on R is trivial iff at least one α_i is the set of all attributes of R.
- JD $* \{\alpha_1, \alpha_2, \ldots, \alpha_\mathrm{m}\}$ on R is implied by the candidate keys of R iff each α_i is a superkey in R.

=> FaPrCo not in 5NF

- decomposition: projections on FaPr, PrCo, FaCo

References

- [Ta13] ȚÂMBULEA, L., Curs Baze de date, Facultatea de Matematică și Informatică, UBB, 2013-2014
- [Ra02] RAMAKRISHNAN, R., GEHRKE, J., Database Management Systems (3rd Edition), McGraw-Hill, 2002
- [Da03] DATE, C.J., An Introduction to Database Systems (8th Edition), Addison-Wesley, 2003
- [Ga09] GARCIA-MOLINA, H., ULLMAN, J., WIDOM, J., Database Systems: The Complete Book (2nd Edition), Pearson Education, 2009
- [Ha96] HANSEN, G., HANSEN, J., Database Management And Design (2nd Edition), Prentice Hall, 1996
- [Ra02S] RAMAKRISHNAN, R., GEHRKE, J., Database Management Systems, Slides for the 3rd Edition, http://pages.cs.wisc.edu/~dbbook/openAccess/thirdEdition/slides/slides3ed.html
- [Si11] SILBERSCHATZ, A., KORTH, H., SUDARSHAN, S., Database System Concepts (6th Edition), McGraw-Hill, 2011
- [Si19S] SILBERSCHATZ, A., KORTH, H., SUDARSHAN, S., Database System Concepts, Slides for the 7th Edition, http://codex.cs.yale.edu/avi/db-book/
- [Ul11] ULLMAN, J., WIDOM, J., A First Course in Database Systems, http://infolab.stanford.edu/~ullman/fcdb.html