Relational Database Design Theory (I)

March 24, 2023

Announcements

- Assignment (II) due: April 2, 2023
- The first quiz: April 7, 2023

Agenda

- Functional dependency theory (this lecture)
- NF's and decomposition algorithms (next lecture)

Functional Dependency Theory

Functional dependencies

Let $X = \{A_1, \ldots, A_n\}$ and $Y = \{B_1, \ldots, B_m\}$ be sets of attributes.

Definition [Functional dependency]

A functional dependency (FD) is of the form

$$X \rightarrow Y$$

that requires the attributes of X functionally determining the attributes Y.

In particular, a relation R satisfies $X \rightarrow Y$ if for every two tuples t_1 and t_2 of R

$$\wedge_{i=1}^{n} t_1[A_i] = t_2[A_i] \rightarrow \wedge_{j=1}^{m} t_1[B_j] = t_2[B_j].$$

- FD's are unique-value constraints.
- A FD $X \rightarrow Y$ holds on a relational schema R if every instance of R satisfies $X \rightarrow Y$.
- If $Y \subseteq X$, then $X \rightarrow Y$ is trivial.

Notation convention

- $A_1 \ldots A_n$ represents $\{A_1, \ldots, A_n\}$.
- Attributes: A, B, C, D, E
- Sets of attributes: X, Y, Z
- XY represents $\mathrm{X} \cup \mathrm{Y}$

FD example

sid	cid	cname	room	grade
123	AI-3613	Database	1-108	A+
223	AI-3613	Database	1-108	B+
123	CS-101	CS Intro.	3-325	A
334	CS-101	CS Intro.	3-325	A-
345	ICE-1404P	Database	2-203	A

Table: R(sid, cid, cname, room, grade)

- cid $\rightarrow$ cname
- cid $\rightarrow$ room
- cid $\rightarrow$ {cname, room}
- sid, cid $\rightarrow$ grade

Keys and FD's

Definition

A set X of attributes is a candidate key for relation R if

- X functionally determines all other attributes of R, i.e., X is a superkey.
- No proper subset of X functionally determines all other attributes of R.
  –That is, X is minimal.

A motivation to study FD's

sid	cid	cname	room	grade
123	AI-3613	Database	1-108	A+
223	AI-3613	Database	1-108	B+
123	CS-101	CS Intro.	3-325	A
334	CS-101	CS Intro.	3-325	A-
345	ICE-1404P	Database	2-203	A

Table: R(sid, cid, cname, room, grade)

- Data redundancy
- Update/insertion/deletion anomaly

Lossless join decomposition

Goal: Decompose R into R_1 and R_2 s.t.

$$R = R_1 \bowtie R_2$$

sid	cid	cname	room	grade
123	AI-3613	Database	1-108	A+
223	AI-3613	Database	1-108	B+
123	CS-101	CS Intro.	3-325	A
334	CS-101	CS Intro.	3-325	A-
345	ICE-1404P	Database	2-203	A

Table: R(sid, cid, cname, room, grade)

s_id	c_id	grade
123	AI-3613	A+
223	AI-3613	B+
123	CS-101	A
334	CS-101	A-
345	ICE-1404P	A

Table: R_1(sid, cid, grade)

c_id	cname	room
AI-3613	Database	1-108
CS-101	CS Intro.	3-325
ICE-1404P	Database	2-203

Table: R_2(cid, cname, room)

- $R_1 \cap R_2 = \{cid\}$.
- cid is a superkey of R_2, i.e., cid $\rightarrow \{cid, cname, room\}$.

Reasoning about FD's

Definition

- A set F of FD's logically implies a set G of FD's if every relation instance that satisfies all the FD's in F also satisfies all the FD's in G.
- F and G are equivalent if (i) F logically implies G and (ii) G logically implies F.

Example

- $\{A \to B\}$ logically implies $\{AC \to BC\}$.
- $\{A \to B, B \to C\}$ logically implies $\{A \to C\}$.
- $\{A \to B, B \to C\}$ is equivalent to $\{A \to B, B \to C, A \to C\}$.
- $\{A_1A_2 \to B_1B_2B_3\}$ is equivalent to $\{A_1A_2 \to B_1, A_1A_2 \to B_2, A_1A_2 \to B_3\}$.

Closure of attributes

Definition [Attribute closure]

Let X be a set of attributes and F be a set of FD's. The closure of X under F, written as X_F^+, is the set of all attributes B such that F logically implies $X \rightarrow B$.

- We omit the subscript F and write X^+ if F is clear from the context.
- To determine whether F logically implies $X \rightarrow Y$ it suffices to check whether $Y \subseteq X^+$.
- To see if X is a superkey of R, it suffices to check if X^+ contains all the attributes of R.

Computing attribute closure

Input: A set of attributes X and a set of FD's F
Output: X_F^+
1. $Z \leftarrow X$;
2. repeat
3. if ex. $X' \rightarrow Y'$ in F s.t. $X' \subseteq Z$ and $Y' \setminus Z \neq \emptyset$
4. then $Z \leftarrow Z \cup Y'$;
5. until (Z no longer changes);
6. return Z;

Figure: Computing attribute closure

- $F = \{A \rightarrow B, A \rightarrow C, CD \rightarrow E, CD \rightarrow K, B \rightarrow E\}$
- What is $\{A, D\}_F^+$?
- Is $\{A, D\}$ a superkey/candidate key?

Algorithm correctness

Correctness. $\widehat{X_F^+} = X_F^+$, where $\widehat{X_F^+}$ is the set of attributes computed by the algorithm.

- $\widehat{X_F^+} \subseteq X_F^+$. $X \subseteq X_F^+$ and by I.H. every new element introduced in line 4 is also in X_F^+.
- $X_F^+ \subseteq \widehat{X_F^+}$. Let B be an attribute not in $\widehat{X_F^+}$. It suffices to show that F cannot imply $X \to B$. That is, there is a table R s.t.(i) R satisfies F, and (ii) R does not satisfy $X \to B$.

  Let $\widehat{X_F^+} = \{A_1, A_2, \ldots, A_n\}$ and $\overline{\widehat{X_F^+}} = \{B_1, B_2, \ldots, B_m\}$. We define R as

A_1	A_2	...	A_n	B_1	B_2	...	B_m
1	1	...	1	1	1	...	1
1	1	...	1	0	0	...	0

  It should be clear that R does not satisfy $X \to B$. It remains to verify that R satisfies F.

  Claim. R satisfies F.

  We prove it by contraction. Let $X' \to Y'$ be an FD in F that R does not satisfy. By construction, we must have $X' \subseteq \{A_1, A_2, \ldots, A_n\}$ and $Y' \cap \{B_1, B_2, \ldots, B_m\} \neq \emptyset$.
  It follows that all the attributes in Y' should also be included in $\widehat{X_F^+}$ (lines 3-4).
  This contradicts to $Y' \cap \{B_1, \ldots, B_m\} \neq \emptyset$. □

Closure of FD's

Definition

The closure of F, denoted by F^+, is the set of all FD's logically implied by F.

Question. Given a set of FD's F, how to decide whether $X \rightarrow Y \in F^+$?

- Approach 1: compute X^+ and check whether $Y \subseteq X^+$.
- Approach 2: use Armstrong's axioms.

Armstrong's axioms

- Reflexivity: If $Y \subseteq X$, then $X \to Y$.
- Augmentation: If $X \to Y$, then $XZ \to YZ$.
- Transitivity: If $X \to Y$ and $Y \to Z$, then $X \to Z$.

Theorem (Armstrong '74). The Armstrong's axioms are both sound and complete.

- Soundness: Only correct FD's are derived.
- Completeness: Every FD in F^+ can be derived by using the axioms.

Motivation for canonical cover

- A set of FD's F defines a set of unique-value constraints.
- We want a minimal set F' of FD's to reduce constraint checking cost.
- F' should be equivalent to F to ensure correctness.

A canonical cover F_c of F is a minimal set of FD's equivalent to F.

Extraneous attributes

An attribute of a FD $X \to Y$ in FD is extraneous if we can remove it without changing F^+.

- An attribute $A \in X$ is extraneous and can be removed from the LHS of $X \to Y$ if

  $$F \text{ logically implies } (F \setminus \{X \to Y\}) \cup \{(X \setminus \{A\}) \to Y\}.$$

- Example. $F = \{AB \to C, A \to D, D \to C\}$
- An attribute $B \in Y$ is extraneous and can be removed from the RHS of $X \to Y$ if

  $$(F \setminus \{X \to Y\}) \cup \{X \to (Y \setminus \{B\})\} \text{ logically implies } F.$$

- Example. $F = \{A \to CD, D \to C\}$

Extraneous attributes (cont'd)

Let $X \to Y$ be a FD in F.

- $A \in X$ is extraneous and can be removed from the LHS of $X \to Y$ if

  $$F \text{ logically implies } (F \setminus \{X \to Y\}) \cup \{(X \setminus \{A\}) \to Y\}$$

  $$\iff Y \subseteq (X \setminus \{A\})^+_F.$$

- Example. $F = \{AB \to C, A \to D, D \to C\}$.
  B can be removed from $AB \to C$ since $\{A\}^+_F = \{A, C, D\}$ and $\{C\} \subseteq \{A\}^+_F$.

- $B \in Y$ is extraneous and can be removed from the RHS of $X \to Y$ if

  $$(F \setminus \{X \to Y\}) \cup \{X \to (Y \setminus \{B\})\} \text{ logically implies } F.$$

  $$\iff B \in X^+_{F'}, \text{ where } F' = (F \setminus \{X \to Y\}) \cup \{X \to (Y \setminus \{B\})\}.$$

- Example. $F = \{A \to CD, D \to C\}$.
  C can be removed from $A \to CD$ since $C \in \{A\}^+_{F'}$, where $F' = \{A \to D, D \to C\}$.

Canonical cover

Definition

A canonical cover F_c for F is a set of FD's equivalent to F such that

- No FD in F_c contains an extraneous attribute.
- Each LHS of a FD in F_c is unique.

Computing canonical cover

Input: A set F of FD's
Output: A canonical cover F_c of F

1. $F_c \leftarrow F$;
2. repeat
3. for each pair of FD's $X \rightarrow Y_1$ and $X \rightarrow Y_2$ in F_c do
4. replace them with $X \rightarrow Y_1Y_2$;
5. if ex. a FD in F_c with an extraneous attribute then
6. remove the extraneous attribute and update F_c;
7. until (F_c no longer changes)
8. return F_c;

Figure: Computing canonical cover

Canonical cover example

Let $F = \{A \rightarrow BC, B \rightarrow C, A \rightarrow B, AB \rightarrow C\}$.

- $F_c^0 = \{A \rightarrow BC, B \rightarrow C, AB \rightarrow C\}$
- $F_c^1 = \{A \rightarrow B, B \rightarrow C, AB \rightarrow C\}$
- $F_c^2 = \{A \rightarrow B, B \rightarrow C\}$

Let $F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow AB\}$.

- $F_c = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$.
- $F_c = \{A \rightarrow C, C \rightarrow B, B \rightarrow A\}$.
- $F_c = \{A \rightarrow C, B \rightarrow C, C \rightarrow AB\}$.