CS2102 Database Systems

SCHEMA REFINEMENT: FUNCTIONAL DEPENDENCIES

DB Schema = Relation Schemas + Constraints

- Data represented by schemas have application-dependent constraints relating to attribute values

MovieList Database

title	director	address	phone	Time
Schlinder's List	Spielberg	Holland	3355	1130
Saving Private Ryan	Spielberg	Holland	3355	1430
Noth by Northwest	Hitchcock	Orchard	1234	1400
The Godfather	Coppola	Orchard	1234	1700
Saving Private Ryan	Spielberg	Orchard	1234	2130

- Data constraints:
  - Each movie has one director
  - Each cinema has one phone number
  - Each cinema screens one movie at a time

Good and Bad Schema Design

- Problems with the MovieList database
  - Redundant storage: Some information is stored repeatedly
  - Insertion anomaly: Cannot store information about a new movie if the screening place and time are not known

MovieList Database

title	director	address	phone	Time
Schlinder's List	Spielberg	Holland	3355	1130
Saving Private Ryan	Spielberg	Holland	3355	1430
Noth by Northwest	Hitchcock	Orchard	1234	1400
The Godfather	Coppola	Orchard	1234	1700
Saving Private Ryan	Spielberg	Orchard	1234	2130

Good and Bad Schema Design

- Problems with the MovieList database:
  - Deletion anomaly: Lose information about cinema at Holland if we delete all movies directed by Spielberg
  - Update anomaly: Inconsistent updates may occur if the phone number of a cinema changes

MovieList Database

title	director	address	phone	Time
Schlinder's List	Spielberg	Holland	3355	1130
Saving Private Ryan	Spielberg	Holland	3355	1430
Noth by Northwest	Hitchcock	Orchard	1234	1400
The Godfather	Coppola	Orchard	1234	1700
Saving Private Ryan	Spielberg	Orchard	1234	2130

Good and Bad Schema Design

- Refine a bad schema by decomposing it into multiple good ones

Movie

title	director
Schlinder's List	Spielberg
Saving Private Ryan	Spielberg
Noth by Northwest	Hitchcock
The Godfather	Coppola

Screens

address	time	title
Holland	1130	Schlinder's List
Holland	1430	Saving Private Ryan
Orchard	1400	Noth by Northwest
Orchard	1700	The Godfather
Orchard	1430	Saving Private Ryan

Cinema

address	phone
Holland	3355
Orchard	1234

Good and Bad Schema Design

- Refined schema allows
  - Insertion of new movies without knowing their screening details
  - Deletion of movies without losing information about cinemas
  - Updating a single record to change a cinema's phone number

Schema Design Issues

- Two main problems:
  - How to determine whether a schema design is good or bad?
  - How to transform a bad design into a good one?
- Theory of functional dependencies provide a systematic approach to address these issues
- Introduced by E.F. Codd
  - *A relational model for large shared data banks*, in Communications of the ACM, Vol. 13, No. 6, 1970.

Functional Dependencies (FDs)

- Let X and Y be subsets of attributes of a relation R
- A functional dependency $X \rightarrow Y$ holds over R if and only if for any instance *r* of R, whenever two tuples t_1 and t_2 of *r* agree on the attributes X, they also agree on the attributes Y.

$$t_1.X = t_2.X \Rightarrow t_1.Y = t_2.Y$$

- We say that X functionally determines Y (or Y functionally depends on X)

Example

- MovieList (title, director, address, phone, time)

title	director	address	phone	Time
Schlinder's List	Spielberg	Holland	3355	1130
Saving Private Ryan	Spielberg	Holland	3355	1430
Noth by Northwest	Hitchcock	Orchard	1234	1400
The Godfather	Coppola	Orchard	1234	1700
Saving Private Ryan	Spielberg	Orchard	1234	2130

- Functional dependencies on MovieList:
  - title $\rightarrow$ director
  - address $\rightarrow$ phone
  - address, time $\rightarrow$ title

FDs Definitions

- Let *r* be a relation instance of relation schema *R*
- *r* **satisfies** FD $X \rightarrow Y$ if **for every pair of tuples** t_1 and t_2 in *r* such that $t_1.X = t_2.X$, it is also true that $t_1.Y = t_2.Y$
- An FD *f* **holds** on *R* if and only if for any relation instance *r* of *R*, *r* satisfies *f*

FDs Definitions

- *r* is said to violate an FD *f* if *r* does not satisfy *f*
- *r* is a legal instance of *R* if *r* satisfies all FDs that hold on *R*
- An FD X $\rightarrow$ Y is a trivial FD if Y $\subseteq$ X; otherwise it is a non-trivial FD

Example

- Consider relation schema Movie (title, director, producer)
- Let r be a legal relation instance of Movie

title	director	producer
Angela's Ashes	Parker	Williams
Saving Private Ryan	Spielberg	Williams
Noth by Northwest	Hitchcock	Harris
Schindler's List	Spielberg	Williams
Vertigo	Hitchcock	Harris

- FD producer → director does not hold on Movie
- r satisfies the FD director → producer
  - But we cannot conclude that director → producer holds on Movie
- Based on legal instances of R, we can tell which FDs do not hold on R, but we <u>cannot</u> deduce which non-trivial FDs hold on R!

Quiz

- Consider the relation instance r of schema R(A, B, C)

r

A	B	C
0	0	0
2	1	2
1	1	2
0	0	1

- List all non-trivial FDs that are satisfied by r

Reasoning about FDs

- Implication problem:
  - Given a set of FDs F that hold on R, and an FD f, does f also hold on R?
- Example in MovieList, we have FDs

$$
\begin{aligned}
F = \{ & \text{title} \rightarrow \text{director}, \\
& \text{address} \rightarrow \text{phone}, \\
& \{\text{address, time}\} \rightarrow \text{title} \}
\end{aligned}
$$

- F logically implies f if every relation instance r of R that satisfies the FDs in F also satisfies the FD f

Reasoning about FDs

- Let *F* and *G* denote sets of FDs, and *f* denote an FD
- *F* implies G if *F* implies *g* for each $g \in G$
- Closure of F, denoted by F^+, is the set of all FDs implied by F.
- Two sets of FDs, F and G, are equivalent, denoted by $F \equiv G$, if $F^+ = G^+$.

Axioms for FDs

- A collection of formal rules used to derive an FD from a set of FDs
- Armstrong's Axioms

  Let *X*, *Y*, *Z* denote sets of attributes over a relation schema R

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

Axioms for FDs

- Armstrong's Axioms are both sound and complete
  - Sound: Any derived FD is implied by F
  - Complete: All FDs in F^+ can be derived

Example

- Consider R(A, B, C, D, E) with FDs

$$F = \{A \rightarrow C, B \rightarrow C, CD \rightarrow E\}$$

  Show that F implies $AD \rightarrow E$

1. $A \rightarrow C$ (given)
2. $AD \rightarrow CD$ (augmentation with (1))
3. $CD \rightarrow E$ (given)
4. $AD \rightarrow E$ (transitivity with (2) and (3))

Additional Inference Rules

- Union:

  If $X \to Y$ and $X \to Z$, then $X \to YZ$

- Decomposition:

  If $X \to YZ$, then $X \to Y$ and $X \to Z$

Example

- Show that $\{A \rightarrow BCD\}$ is equivalent to

  $\{A \rightarrow B, A \rightarrow C, A \rightarrow D\}$

Let $F = \{A \rightarrow BCD\}$

Let $G = \{A \rightarrow B, A \rightarrow C, A \rightarrow D\}$

By the decomposition rule, we have

  F implies $A \rightarrow B, A \rightarrow C, A \rightarrow D$

Therefore, F implies G

By the union rule, we have

  $\{A \rightarrow B, A \rightarrow C\}$ implies $A \rightarrow BC$ and

  $\{A \rightarrow BC, A \rightarrow D\}$ implies $A \rightarrow BCD$

Therefore, G implies F

Hence, $F \equiv G$

Quiz

- Show that $\{A \to B, AB \to C, D \to AC, D \to E\}$ and $\{A \to BC, D \to AE\}$ are equivalent

Superkeys, Keys & Prime Attributes

- A set of attributes X is a superkey of schema R (with FDs F) if F implies $X \rightarrow R$
- A set of attributes X is a key of schema R if
  - X is a superkey, and
  - No proper subset of X is a superkey
- An attribute A in R is a prime attribute if A is contained in some key of R; otherwise, it is a nonprime attribute

Example

- Consider MovieList (title, director, address, phone, time) with FDs
  - {address, time} $\rightarrow$ title
  - address $\rightarrow$ phone
  - title $\rightarrow$ director
- {address, time} is the only key of MovieList
- {address, time} are the only prime attributes in MovieList
- Any superset of {address, time} is a superkey of MovieList

Example

- Consider R(A, B, C, D) with FDs

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

  Is AB a superkey?

1. $AB \rightarrow ABC$ (augmentation of $A \rightarrow B$ with AB)
2. $ABC \rightarrow ABCD$ (augmentation of $B \rightarrow D$ with ABC)
3. $AB \rightarrow ABCD$ (transitivity with (1) and (2))

Hence, AB is a superkey.

Closure of a Set of FDs

- Computing F^+ for a set of FDs F is not efficient as the size of F^+ could be exponentially large
- Consider the relation scheme R(A,B,C,D)

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

  $F+ = \{ \{A\} \rightarrow \{A\}, \{B\} \rightarrow \{B\}, \{C\} \rightarrow \{C\}, \{D\} \rightarrow \{D\}, \ldots,$
  $\{A\} \rightarrow \{B\}, \{A,B\} \rightarrow \{B\}, \{A,D\} \rightarrow \{B,D\}, \{A,C\} \rightarrow \{B,C\},$
  $\{A,C,D\} \rightarrow \{B,C,D\}, \{A\} \rightarrow \{A,B\}, \{A,B\} \rightarrow \{A,B\},$
  $\{A,D\} \rightarrow \{A,B,D\}, \{A,C\} \rightarrow \{A,B,C\}, \{A,C,D\} \rightarrow \{A,B,C,D\},$
  $\{B,C\} \rightarrow \{D\}, \ldots, \{A,C\} \rightarrow \{D\}, \ldots\}$

Attribute Closure

- More efficient to compute the closure of a set of attributes
- Let $X \subseteq R$ and F be a set of FDs that hold on R
- Closure of X (with respect to F), denoted by X^+, is the set of attributes that are functionally determined by X with respect to F

Computing Attribute Closure

Input: X, F

Output: X^+ w.r.t. F

Let $X_0 = X$ and $i = 0$

Repeat

$X_{i+1} = X_i \cup Z$ such that there is some FD $Y \rightarrow Z \in F$ and $Y \subseteq X_i$

Until when $X_{i+1} = X_i$

Return X_i

Example

- Given F = {AB → C, C → A, BC → D, ACD → B, D → EG, BE → C, CG → BD, CE → AG}, compute the closure of BD

i	X_i	FD used
0	BD	Given
1	BDEG	D → EG
2	BCDEG	BE → C
3	ABCDEG	CE → AG
4	ABCDEG	none

- Thus, BD^+ = ABCDEG

Example

- Given F = {A $\rightarrow$ C, B $\rightarrow$ C, CD $\rightarrow$ E}, show that F implies AD $\rightarrow$ E

i	**X_i**	**FD used**
0	AD	given
1	ACD	A $\rightarrow$ C
2	ACDE	CD $\rightarrow$ E
3	ACDE	none

- Thus, AD^+ = ACDE
- Since E $\in$ AD^+, therefore F implies AD $\rightarrow$ E

Equivalence of Sets of FDs

- We can use attribute closure to determine if two sets of FDs F and G are equivalent
- For each FD $X \rightarrow Y \in F$
  - Compute X+ with respect to G
  - $X \rightarrow Y \in G^{+}$ if $Y \subseteq X+$
- Do the same for each FD in G

Redundant Attributes in FDs

- An attribute $A \in X$ is redundant in the FD $X \rightarrow B$ if $(F - \{X \rightarrow B\} \cup \{X - A\} \rightarrow B\})$ is equivalent to F
- How to check if $A \in X$ is redundant in the FD $X \rightarrow B$?
  - Compute $(X - A)^+$ w.r.t. F
  - $A \in X$ is redundant in $X \rightarrow B$ if $B \in (X - A)^+$

Redundant Attributes in FDs

- What are the redundant attributes in $\{AB \rightarrow C, A \rightarrow B, B \rightarrow A\}$?

Redundant FDs

- An FD $f \in F$ is redundant if $(F - \{ f \})$ is equivalent to F
- How to check if an FD $X \rightarrow A$ is redundant in F ?
  - Compute X^+ w.r.t. $F - \{ X \rightarrow A \}$
  - $X \rightarrow A$ is redundant in F if $A \in X^+$

Redundant FDs

- What are the redundant FDs in

  $\{A \rightarrow B,\ A \rightarrow C,\ B \rightarrow A,\ B \rightarrow C,\ C \rightarrow A\}$?

Minimal Cover for FDs

- A set of FDs F is a minimal cover for a set of FDs G if and only if
  - Every FD in F is of the form $X \rightarrow A$ where X is a set of attributes, A is a single attribute and X has no redundant attributes
  - There are no redundant FDs in F
  - F and G are equivalent

Computing Minimal Cover

- Algorithm
  - Use decomposition rule to obtain FDs with one attribute on RHS
  - Remove redundant attributes from LHS of each FD
  - Remove redundant FDs
- Minimal covers may not be unique due to choice of redundant attributes/FDs

Example

- Let $F = \{ABCD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$. Find a minimal cover of F.

1. Decompose FDs
   - All FDs in F have a single attribute on the RHS
2. Eliminate redundant attributes
   - B in $ABCD \rightarrow E$ is redundant since $E \in ACD^+$ w.r.t. F
   - $F = \{ACD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$
   - D in $ACD \rightarrow E$ is redundant since $E \in AC^+$ w.r.t. F
   - $F = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$
   - There are no more redundant attributes in F

Example

$F = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$

3. Eliminate redundant FDs
   - $AC \rightarrow D$ is redundant since $D \in AC^+$ w.r.t. $F - \{AC \rightarrow D\}$
   - $F = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$
   - There are no more redundant FDs in F

A minimal cover of F is $\{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$

Next…

Schema Refinement: Decomposition