6. Further Normalization

Stéphane Bressan

February 4, 2015

This lecture is based on material by Professor Ling Tok Wang.

CS 4221: Database Design

The Relational Model

Ling Tok Wang
National University of Singapore

1

CS4221: The Relational Model

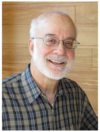
https://www.comp.nus.edu.sg/~lingtw/cs4221/rm.pdf

Content I

Content II

Readings

- Ronald Fagin, "Multivalued Dependencies and a New Normal Form for Relational Databases". ACM Transactions on Database Systems (TODS) Volume 2 Issue 3, 1977.
- David Maier, Alberto O. Mendelzon, and Yehoshua Sagiv, "Testing Implications of Data Dependencies". ACM Transactions on Database Systems (TODS) Volume 4 Issue 4, 1979.




Catalog		
Course	Lecturer	Text
Programming	{Tan CK, Lee SL}	{The Art of Programming, Java}
Maths	{Tan CK}	{Java}
...		

The Catalog relation is a nested relation.
It is in Non-First Normal Form (NF^2).

The indicated courses are taught by all of the indicated teachers, and use all the indicated text books.

The course determines the set of lecturers.
The course determines the set of texts.

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
Programming	Lee SL	Java
DS and Alg.	Tan CK	Java
...		

We transform the Catalog relation into First Normal Form (1NF). What anomalies?

The dependencies cannot be captured by functional dependencies. They are multi-valued dependencies.

Unlike functionl dependencies, multi-valued dependencies are relation sensitive.

Catalog			
Course	Lecturer	Text	Percentage
Programming	Tan CK	The Art of Programming	30
Programming	Tan CK	Java	40
Programming	Lee SL	The Art of Programming	90
Programming	Lee SL	Java	10
DS and Alg.	Tan CK	Java	100
...			

A teacher teaches course and uses a percentage of from a text book.

The previous multi-valued dependencies do not hold anymore.

Definition

An instance r of a relation schema R satisfies the multi-valued dependency σ: $X \twoheadrightarrow Y$, X multi-determines Y or Y is multi-dependent on X, with $X \subset R$, $Y \subset R$ and $X \cap Y = \emptyset$ if and only if , for $Z = R - (X \cup Y)$, two tuples of r agree on their X-value, then there exists a t-uple of r that agrees with the first tuple on the X- and Y-value and with the second on the Z-value.

$$(r \models \sigma)$$

$$\Leftrightarrow$$

$$(\forall t_1 \in r \ \forall t_2 \in r \ (t_1[X] = t_2[X] \Rightarrow$$

$$\exists t_3 \in r \ (t_3[X] = t_1[X] \wedge t_3[Y] = t_1[Y] \wedge t_3[Z] = t_2[Z])))$$

Each X-value in r is consistently associated with one set of Y-value in r.

Notice that the presence of two different t-uples with the same X-values generally implies the presence of two additional t-uples with the Y-values (when Z is not empty).

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Lee SL	Java
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
...		

$$\{Course\} \twoheadrightarrow \{Lecturer\}$$

We sometime use the following embedded MVD notation.

$$X \twoheadrightarrow Y \mid Z$$

It reads "X multi-determines Y independently of Z".

$$\pi_{X \cup Y \cup Z}(r) = \pi_{X \cup Y}(r) \bowtie \pi_{X \cup Z}(r)$$

Catalog			
Course	Lecturer	Text	Percentage
Programming	Tan CK	The Art of Programming	30
Programming	Tan CK	Java	40
Programming	Lee SL	The Art of Programming	90
Programming	Lee SL	Java	10
DS and Alg.	Tan CK	Java	100
...			

$$\{Course\} \not\twoheadrightarrow \{Lecturer\}$$

$$\{Course\} \twoheadrightarrow \{Lecturer\} \mid \{Text\}$$

Nothing can be done about this kind of embedded multi-valued dependencies ...

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
Programming	Lee SL	Java
DS and Alg.	Tan CK	Java
...		

$$\{\text{Course}\} \twoheadrightarrow \{\text{Teacher}\}$$

$$\{\text{Course}\} \twoheadrightarrow \{\text{Text}\}$$

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
Programming	Lee SL	Java
DS and Alg.	Tan CK	Java
...		

$$\{\text{Course}\} \twoheadrightarrow \{\text{Teacher}\} \mid \{\text{Text}\}$$

$$\{\text{Course}\} \twoheadrightarrow \{\text{Text}\} \mid \{\text{Teacher}\}$$

Definition

A multi-valued dependency $X \twoheadrightarrow Y$ is trivial if and only if

1. $Y = R - X$ or
2. $Y \subset X$.

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
Programming	Lee SL	Java
DS and Alg.	Tan CK	Java
...		

$$\{Text\} \twoheadrightarrow \{Course, Lecturer\}$$

Theorem

The Complementation inference rule is sound.
$\forall X \subset R \; \forall Y \subset R$

$$(X \twoheadrightarrow Y) \Rightarrow (X \twoheadrightarrow R - X - Y)$$

Theorem

The Augmentation inference rule is sound.
$\forall X \subset R \; \forall Y \subset R \; \forall V \subset R \; \forall W \subset R$

$$((X \twoheadrightarrow Y) \wedge (V \subset W)) \Rightarrow (X \cup W \twoheadrightarrow Y \cup V)$$

Theorem

The Transitivity inference rule is sound.

$\forall X \subset R \; \forall Y \subset R \; \forall Z \subset R$

$$((X \twoheadrightarrow Y) \wedge (Y \twoheadrightarrow Z)) \Rightarrow (X \twoheadrightarrow Z - Y)$$

Theorem

The Replication (Promotion) inference rule is sound.

$\forall X \subset R \; \forall Y \subset R$

$$(X \rightarrow Y) \Rightarrow (X \twoheadrightarrow Y)$$

Functional dependencies are a special case of multi-valued dependencies.

Theorem

The Coalescence inference rule is sound.

$\forall X \subset R\ \forall Y \subset R\ \forall Z \subset R\ \forall W \subset R$

$$(X \twoheadrightarrow Y) \land (W \rightarrow Z) \land (Z \subset Y) \land (W \cap Y = \emptyset)) \Rightarrow (W \rightarrow Z)$$

Theorem

Complementation, Augmentation, Transitivity, Replication and Coalescence, with the Armstrong Axioms form a sound and complete system for fucntional and multi-valued depenencies.

Theorem

The Multi-valued Union inference rule is sound.

$\forall X \subset R \; \forall Y \subset R \; \forall Z \subset R$

$$((X \twoheadrightarrow Y) \wedge (X \twoheadrightarrow Z)) \Rightarrow (X \twoheadrightarrow Y \cup Z))$$

Theorem

The *Multi-valued Intersection* *inference rule is* *sound*.

$\forall X \subset R \; \forall Y \subset R \; \forall Z \subset R$

$$((X \twoheadrightarrow Y) \wedge (X \twoheadrightarrow Z)) \Rightarrow (X \twoheadrightarrow Y \cap Z)$$

Theorem

The Multi-valued Difference inference rule is sound.

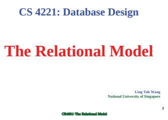
$\forall X \subset R \; \forall Y \subset R \; \forall Z \subset R$

$$((X \twoheadrightarrow Y) \wedge (X \twoheadrightarrow Z)) \Rightarrow (X \twoheadrightarrow Y - Z)$$

There is no decomposition rule.

$$\cancel{(X \twoheadrightarrow Y \cup Z\}) \Rightarrow (X \twoheadrightarrow Y)}$$

Try the examples pages 52 and 53 of the slides:



https://www.comp.nus.edu.sg/~lingtw/cs4221/rm.pdf

Theorem

Let $R = \{A, B\}$. R statisfies $\emptyset \twoheadrightarrow \{A\}$ if and only if, for all valid instances r of R, r is the Cartesian product of its projections on A and B.

$$r = \pi_A(r) \times \pi_B(r)$$

We also have $\emptyset \twoheadrightarrow \{B\}$.

As a special case, $\emptyset \rightarrow \{A\}$ means that the A-value is constant, or r is empty. Still $\emptyset \twoheadrightarrow \{B\}$ but not necessarily $\emptyset \rightarrow \{B\}$.

Definition

A relation R is in Fourth Normal Form (4NF) if and only if any non-trivial MVD $X \twoheadrightarrow Y$ holds in R implies X is a superkey of R.

Theorem

$$\mathit{4NF} \subset \mathit{BCNF}$$

$$\mathit{4NF} \neq \mathit{BCNF}$$

Catalog		
Course	Lecturer	Text
Programming	Tan CK	The Art of Programming
Programming	Tan CK	Java
Programming	Lee SL	The Art of Programming
Programming	Lee SL	Java
DS and Alg.	Tan CK	Java
...		

$$Course \twoheadrightarrow Lecturer$$

$$Course \twoheadrightarrow Text$$

Catalog_L	
Course	Lecturer
Programming	Tan CK
Programming	Lee SL
DS and Alg.	Tan CK
...	

Catalog_T	
Course	Text
Programming	The Art of Programming
Programming	Java
DS and Alg.	Java
...	

Theorem

A relation schema R satisfies the multi-valued dependency $X \twoheadrightarrow Y$ if and only if every valid instance of R is such that :

$$r = \pi_{X \cup Y}(r) \bowtie \pi_{X \cup \ (R-Y)}(r)$$

$R(X, Y, Z)$ is the join of its projections $R_1(X, Y)$ and $R_2(X, Z)$.

Decomposition into 4NF

If $X \twoheadrightarrow Y$ is a 4NF violation for relation R, we can decompose R using the same technique as for BCNF.

1. $X \cup Y$ is one of the decomposed relations.
2. All but $Y - X$ is the other.

Theorem

Any relation can be non-loss decomposed into an equivalent collection of 4NF relations.

Shortcomings

- The algorithm is not dependency preserving (no algorthm can be dependency preserving because there might not exists a lossless dependency preserving decomposition in Fourth Normal form. Why?).
- There may be several possible decompositions.
- It does not always find all the keys.
- Decomposition in BCNF may exists but not reachable by binary decomposition.

Another Method [by Ling Tok Wang]

1. Normalize the relation R into a set of 3NF and/or BCNF relations based on the given set of FDs.
2. For each relation not in 4NF, if all attributes belong to the same key and there exists non-trivial MVDs in the relation, then decompose the relation into 2 smaller relations (don't if you loose functional dependencies).

Let Σ be a set of functional and multi-valued dependencies on a relation schema R. The Chase is an algorithm that solves the decision problem of whether a functional or multi-valued dependency σ is satisfied by R with Σ.

$$(R \text{ with } \Sigma) \models \sigma?$$

Example 1

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

Example 2

$$R = \{A, B, C, D\}$$

$$\{\{A\} \twoheadrightarrow \{B\}, \{B\} \twoheadrightarrow \{C\}\} \models \{A\} \twoheadrightarrow \{C\}?$$

Example 3

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{C, D\} \rightarrow \{B\}\} \models \{A\} \rightarrow \{B\}?$$

Example 1

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

Create an instance r on the schema $\{A, B, C, D\}$ with two t-uples and distinct values for all attributes.

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2

Example 1 (Cont.)

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

We want to chase $\{A\} \rightarrow \{C\}$.
Make the A-values the same.

$$a_1 = a_2$$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2

Example 1 (Cont.)

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

Use $\{A\} \twoheadrightarrow \{B, C\}$. Create two new t-uples by copying the two t-uples that have the same A-value but swapping their B- and C-values. The multi-valued dependency generates t-uples. It is a t-uple generating dependency.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_2	d_1
a_1	b_1	c_1	d_2

Example 1 (Cont.)

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

Use $\{D\} \rightarrow \{C\}$. For each pair of t-uple with the same D-value, make their C-value the same.

$$c_1 = c_2$$

The functional dependency generates values. It is a values generating dependency.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_1	d_2

Example 1 (Cont.)

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{D\} \rightarrow \{C\}\} \models \{A\} \rightarrow \{C\}?$$

There is nothing else to do. We observe that r satisfies $\{A\} \rightarrow \{C\}$.

$$r \models \{A\} \rightarrow \{C\}$$

Therefore the answer is yes

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_1	d_2

Example 1 (Cont.)

r also satisfies $\{D\} \rightarrow \{A\}$ but this is a coincidence. We can only answer the question about $\{A\} \rightarrow \{C\}$.
Another chase is needed for $\{D\} \rightarrow \{A\}$. Do it!

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_1	d_2

Example 2

$$R = \{A, B, C, D\}$$

$$\{\{A\} \twoheadrightarrow \{B\}, \{B\} \twoheadrightarrow \{C\}\} \models \{A\} \twoheadrightarrow \{C\}?$$

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2

Example 2 (Cont.)

$$R = \{A, B, C, D\}$$

$$\{\{A\} \twoheadrightarrow \{B\}, \{B\} \twoheadrightarrow \{C\}\} \models \{A\} \twoheadrightarrow \{C\}?$$

We want to chase $\{A\} \twoheadrightarrow \{C\}$.
Make the A-values the same.

$$a_1 = a_2$$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2

Example 2 (Cont.)

$$R = \{A, B, C, D\}$$

$$\{\{A\} \twoheadrightarrow \{B\}, \{B\} \twoheadrightarrow \{C\}\} \models \{A\} \twoheadrightarrow \{C\}?$$

Use $\{A\} \twoheadrightarrow \{B\}$.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_2	d_2

Example 2 (Cont.)

$$\{\{A\} \twoheadrightarrow \{B\}, \{B\} \twoheadrightarrow \{C\}\} \models \{A\} \twoheadrightarrow \{C\}?$$

Use $\{B\} \twoheadrightarrow \{C\}$ (twice).

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_2	d_2
a_1	b_1	c_2	d_1
a_1	b_1	c_1	d_2
a_1	b_2	c_1	d_2
a_1	b_2	c_2	d_1

Example 2 (Cont.)

There is nothing else to do.

$$r \models \{A\} \twoheadrightarrow \{C\}$$

Therefore the answer is yes

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_1	d_1
a_1	b_1	c_2	d_2
a_1	b_1	c_2	d_1
a_1	b_1	c_1	d_2
a_1	b_2	c_1	d_2
a_1	b_2	c_2	d_1

Example 3

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{C, D\} \rightarrow \{B\}\} \models \{A\} \rightarrow \{B\}?$$

A	B	C	D
a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2

Example 3 (cont.)

$$\{\{A\} \twoheadrightarrow \{B, C\}, \{C, D\} \rightarrow \{B\}\} \models \{A\} \rightarrow \{B\}?$$

Use $\{A\} \twoheadrightarrow \{B, C\}$.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_2	d_1
a_1	b_1	c_1	d_2

Example 3 (cont.)

There is nothing else to do.

$$r \not\models \{A\} \rightarrow \{B\}$$

Therefore the answer is No

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_2	c_2	d_1
a_1	b_1	c_1	d_2

We have built a counter-example.

The Power of The Chase

What is surprising and powerful is that we can use The Chase to prove that a functional or multi-valued dependecy is satisfied!

Theorem

The Chase always builds a counter example if it exists and does not if it does not exists.

Setting The Chase

Let Σ be a set of functional and multi-valued dependencies on a relation schema R. Let σ be a be a functional and multi-valued dependency.

$$\sigma = X \rightarrow Y \text{ or } \sigma = X \twoheadrightarrow Y$$

1. Create a table r with schema R with two tuples with all different values.
2. For each $A \in X$, make the A-values the same (choosing new and different values for each A, though).

If R is not given, then use the attributes in Σ and σ.

Chasing The Chase

Repeat the following until you reach a fixed point (nothing changes):

1. For each functional dependency $Z \rightarrow V \in \Sigma$.
    1. If there are tuples in the table with same Z-value, then set their V-values to be the same.
2. For each multi-valued dependency $Z \twoheadrightarrow V \in \Sigma$.
    1. If there are two tuples in the table with same Z-value, then add two new tuples with all the same values and except for their V-values that are swapped.

Exit with:

$$r \models \sigma \quad \text{is equivalent to} \quad \Sigma \models \sigma$$

This means that you only need to check whether or not r satisfies the functional or multi-valued depedency σ that you were chasing.

Theorem

The Chase is sound and complete for σ.

$$r \models \sigma \quad \text{is equivalent to} \quad \Sigma \models \sigma$$

Theorem

The Chase always terminates.

How to use to check to check that a decompostion is lossless?

Summary

- How do we find non-trivial MVDs in a relation?
- MVDs are relation sensitive.
- If a relation is not in 4NF, then there is a non-loss decomposition of R into a set of 4NF relations. However, it may not cover all the given FDs.
- When we normalize relations involving only FDs, we must maintain (cover) all the non-trivial FDs. However, when we normalize relations to 4NF, we want to remove non-trivial MVDs.
- The Chase Algorithm for FD/MVD membership test.

Definition

A relation schema R satisfies a join dependency, $\bowtie [X_1, \cdots, X_n]$ if and only if every valid instance of R is such that :

$$r = \pi_{X_1}(r) \bowtie \cdots \bowtie \pi_{X_n}(n)$$

Read and self-study pages 66 to 76 of "CS 4221: Database Design The Relational Model" by Prof. Ling Tok Wang. These topics will will neither be covered nor examined. You will find related discussions in the articles and books given as complementary readings.

CS 4221: Database Design

The Relational Model

Ling Tok Wang
National University of Singapore

1

CS4221: The Relational Model

https://www.comp.nus.edu.sg/~lingtw/cs4221/rm.pdf