

Database Systems - Homework 3 - Functional Dependencies & Normal Forms

Yosef Goren & Yonatan Kahalani

June 22, 2023

Contents

1	2
1.1	2
1.2	2
1.3	2
1.4	2
1.5	3
2 Scheme Analysis	3
2.1	3
2.2	3
2.3	4
2.4	4
3 Scheme Analysis	4
3.1	4
3.2	5
3.3	5
3.4	6
4 Variations on Armstrong's Axioms	6
4.1	6
4.2	6
5 MongoDB	7
5.1	7
5.2	7
5.3	7
5.4	8

1

1.1

X is not in $BCNF$.

Proof.

From transitivity and reflexivity, $CG \rightarrow H$.

Note CG is not a superkey in R_3 since D is not implied by any FD since it only appears as the left operand.

Thus $CG \rightarrow D$ is not satisfied and $R_3 \notin BCNF \Rightarrow X \notin BCNF$.

1.2

No.

Proof.

- e only appears as the right operand.
- e is not prime.
- Any properkey does not contain e .
- c is not a superkey in R_2 .
- $c \rightarrow e$ violates the properties of $3NF$.

1.3

By running the algorithm seen in class we get to:

A	B	C	D	E	G	H
	b	c	d			
		c		e		h
		c	d		g	h

 $\Rightarrow_{C \rightarrow E}$

A	B	C	D	E	G	H
	b	c	d	e		
		c		e		h
		c	d	e	g	h

And thus no more steps can be deduced and via the correctness of the algorithm - the decomposition does not preserve information.

1.4

Let

$$R := \{(a_1, b, c, d, e, g_1, a_1), (a_2, b, c, d, e, g_2, a_2)\}$$

Note that $R \models F$, additionally $(a_2, b, c, d, e, g_1, a_2) \in R_1 \bowtie R_2 \bowtie R_3$, thus the decomposition does not preserve information.

1.5

First we apply the confusing algorithm seen in the tutorial:

- $Z_F := \{A, H\}$
- On inspecting R_1 :
 $\{A, H\} \cup (\{A, H\} \cap (\{A, H\} \cap \{B, C, D\})^+ \cap \{B, C, D\}) = \{A, H\}$
- On inspecting R_2 :
 $B \notin R_2 \Rightarrow B \notin Z_F$
- On inspecting R_3 :
 $B \notin R_3 \Rightarrow B \notin Z_F$

Thus $B \notin Z_F \Rightarrow$ Dependencies are not preserved.

2 Scheme Analysis

2.1

The keys:

- $K_1 := EG$.
 By applying the algorithm we see:
 Note that A can be derived from $BCDEG$ by the FD $G \rightarrow ABCD$ and the Decomposition property.
 By applying the algorithm for finding keys 3 more times, we get rid of BCD . And we are left with EG .
 Note that $E \not\rightarrow G$ and $G \not\rightarrow E$ - hence K_1 is a proper-key.
- $K_2 := ABCD$.
 By the rule $BD \rightarrow G$ we get rid of G , and by the rule $AC \rightarrow D$ we get rid of D .
 Thus we are left with K_2 while not more attributes can be removed - so by the correctness of the algorithm - K_2 is a proper-key.

2.2

Let the decomposition be: $X = \{R_1(A, C, G), R_2(B, D, E), R_3(B, D, G)\}$.

Running the algorithm:

A	B	C	D	E	G		A	B	C	D	E	G	
a		c			g		a		c			g	
	b		d	e		$\Rightarrow_{BD \rightarrow G}$		b		d	e	g	$\Rightarrow_{G \rightarrow ABCD}$
	b		d		g			b		d		g	

A	B	C	D	E	G
a		c			g
a	b	c	d	e	g
	b		d		g

Thus by the correctness of the algorithm, the decomposition is information preserving.

2.3

The claim is correct.

Proof.

Let $K := A_{i_1}A_{i_2}\dots A_{i_k}$ s.t. K is the left operand in f ,
and so $\forall j \in \{1\dots k\} : i_j \in \{1\dots n\}$.

Now we show that K is a super-key:

Let $j \in \{1\dots n\} : A_j \notin K$.

A_j must be the right operand of f , Thus by Decomposition we can derive that $K \rightarrow A_j$. Thus $K^+ = R$.

Thus F satisfies the BCNF property and from a theorem from the lectures, this is sufficient to show that $(R, F) \in BCNF$.

2.4

The claim is incorrect.

Example.

Let $F := A_1 \rightarrow A_2A_3$.

Each A_i appears exactly once. But by the definition of a minimal cover - on the right side of each operand in the cover - there must be exactly one attribute.

3 Scheme Analysis

3.1

The key is $K := ADEG$.

Now we show for each attribute in R that it can be derived from K .

- H :
 - $G \rightarrow H$ (from F)
 - $K \rightarrow KH$ (Augmentation)
 - $KH \rightarrow H$ (Reflexivity)
 - $K \rightarrow H$ (Transitivity)
- B :
 - $G \rightarrow B$ (from F)
 - $K \rightarrow KB$ (Augmentation)
 - $KB \rightarrow B$ (Reflexivity)
 - $K \rightarrow B$ (Transitivity)

- C :
 - $E \rightarrow CH$ (from F)
 - $CH \rightarrow C$ (Reflexivity)
 - $E \rightarrow C$ (from F)
 - $K \rightarrow KC$ (Augmentation)
 - $KC \rightarrow C$ (Reflexivity)
 - $K \rightarrow C$ (Transitivity)

3.2

1. The claim is correct.

Proof.

Direction ' \Rightarrow '

- (a) $B \rightarrow G$ (from F)
- (b) $BG \rightarrow G$ (Augmentation)
- (c) $BG \rightarrow E$ (from F)
- (d) $G \rightarrow H$ (from F)
- (e) $BG \rightarrow BH$ (Augmentation)
- (f) $BH \rightarrow H$ (Reflexivity)
- (g) $BG \rightarrow H$ (e, f, Transitivity)
- (h) $BH \rightarrow EGH$ (b,c,g, Union)

Direction ' \Leftarrow ' Let $A_i \notin F$.

$\Rightarrow A_i$ cannot be derived from the left operand of f

\Rightarrow The left operand of f is not a super-key

$\Rightarrow (R, F) \notin BCNF$

2. The claim is incorrect.

Example.

Let $R := \{(a_1, b_1, c_1, d_1, e_1, g_1, h_1), (a_2, b_2, c_1, d_1, e_2, g_2, h_1)\}$.

Note that $R \models F$, while $R \not\models (CH \rightarrow AE)$ thus, from the soundness theorem - we can deduce there is no proof.

3.3

A	B	C	D	E	G	H
a	b				g	
	b	c		e		
			d	e		h

 $\Rightarrow_{B \rightarrow G}$

A	B	C	D	E	G	H
a	b				g	
	b	c		e	g	
			d	e		h

 $\Rightarrow_{BG \rightarrow E}$

A	B	C	D	E	G	H
a	b			e	g	
	b	c		e	g	
			d	e		h

 $\Rightarrow_{E \rightarrow CH}$

A	B	C	D	E	G	H
a	b	c		e	g	h
	b	c		e	g	h
		c	d	e		h

 $\Rightarrow_{C \rightarrow D}$

A	B	C	D	E	G	H
a	b	c	d	e	g	h
	b	c	d	e	g	h
		c	d	e		h

3.4

	X_2	X_3
BCNF	✓	✗
3NF	✓	✗

Both ✗'s can be demonstrated by $C \rightarrow D$.

4 Variations on Armstrong's Axioms

4.1

Yes. Since this system has less of freedom than the armstrong system, any proof generated by the reduced system is also a proof in the armstrong system. In particular - our system is satisfied by the correctness of the armstrong system.

4.2

No.

For example, let $F := \{A \rightarrow B, A \rightarrow C\}$.

Lemma 1 *In our reduced axiom system - regarding F , for any rule $X \rightarrow Y$: The number of attributes in the right operand is never larger than the number of attributes in the left operand. We denote this as $|X| \geq |Y|$.*

Lemma Proof.

By induction on the number of lines in a formal proof the rule.

Base: By reflexivity or FD from F .

Step: Reflexivity or FD like in the Base case. Transitivity:

We have already shown $W \rightarrow Y$, and that $Z \rightarrow W$. We now show $Z \rightarrow Y$. From the induction hypothesis - $|W| \geq |Y| \wedge |Z| \geq |W|$ Thus $|Z| \geq |Y|$.

Due to the lemma, in the reduced system, $A \rightarrow BC$ cannot be implied by F . While union is provable in the armstrong system (which means $A \rightarrow BC$ can be implied by F in the armstring system).

5 MongoDB

5.1

```
db.s.distinct("university", {seniority_year: 2023})
```

5.2

```
db.SeniorStaff.aggregate([
  {
    $match: {
      seniority_year: { $gt: 2013 }
    }
  },
  {
    $group: {
      _id: "$university",
      staff_members: {
        $push: {
          staff_member_name: "$staff_member_name",
          staff_member_id: "$staff_member_id"
        }
      }
    }
  }
])
```

5.3

```
db.SeniorStaff.aggregate([
  {
    $group: {
      _id: {
        university: "$university",
        estimated_retirement_year: "$estimated_retirement_year"
      },
      total_current_salary: { $sum: "$salary" },
      total_retirement_salary: { $sum: "$estimated_retirement_salary" }
    }
  },
  {
    $group: {
      _id: "$_id.university",
      retirement_years: {
        $push: {
          estimated_retirement_year: "$_id.estimated_retirement_year",
          salary_delta: { $subtract: ["$total_current_salary", "$total_retirement_salary"] }
        }
      }
    }
  },
  {
    $sort: {_id: 1}
  }
])
```

5.4

```
db.s.mapReduce(
  function() {
    emit(this.university, [this.estimated_retirement_year, this.salary, this.estimated_retirement_salary]);
  },
  function(university, staff_memebers) {
    expensive_staff_memebers_count = 0;
    for(memebr_details in staff_memebers){
      retire_year = memebr_details[0];
      cur_salary = memebr_details[1];
      retire_salary = memebr_details[2];
      if(retire_salary*1.5 <= cur_salary){
        expensive_staff_memebers_count++;
        if(expensive_staff_memebers_count > 20){
          break;
        }
      }
    }
    if(expensive_staff_memebers_count > 0){
      return 1;
    } else {
      return 0;
    }
  },
  {
    query: {estimated_retirement_year: {$gte:2025, $lte:2030}},
    out: "expensive_staff_unis"
  }
)
```