Assignment 6: Decomposition and Normal Forms

Hien Tu - tun1

November 21, 2021

Part 1: The analysis of a quick-event wizard for a local community

1. Minimal cover of all realistic non-trivial functional dependencies:

   - Since each event is organized by a user, we know that if the event id (*id*) is the same, then the organizer (*user_id*) must be the same. So, we can have id $\longrightarrow$ user_id. The reverse would not hold since an organizer could organize many events.
   - Furthermore, each event happens in a day, so, we know that if the event id (*id*) is the same, the date that the event happens in (*date*) must be the same. So, we can have id $\longrightarrow$ date.
   - We also have id, inv_id $\longrightarrow$ inv_confirmed since each guest would determine what event to go to. We need both the event id (*id*) and the guest (*inv_id*) to determine if the invitation is confirmed (*inv_confirmed*).
   - Furthermore, we have id, product $\longrightarrow$ p_amount since we would only know how much to bring if we know what to bring and where we need to bring it to. Only the product wouldn't be able to determine the amount since different event could need different amount of the same product. For example, in the given table, event 1 needs 4 chips while event 2 only needs 2 chips.
   - We have product $\longrightarrow$ p_price since each product has its own price. The price of the product wouldn't depend on the event that the product is brough into. An example is that, in the given table,

the chips cost \$2 no matter if it is brought into event with id 1 or 2 and the cola would cost \$4 no matter if it is brought into event with id 1 or 2.

2. An example of a non-trivial dependency is id $\twoheadrightarrow$ inv_id, inv_confirmed. This would hold since, for example, in the given table, from the first row and the fourth row, we know that there would exist a row where inv_id and inv_confirmed are the same as the first row and the rest of the attributes (user_id, date, product, p_price, p_amount) are the same as the fourth row. This row is the thrid row. We also know that there would exist a row where inv_id and inv_confirmed are the same as the fourth row and the rest of the attributes (user_id, date, product, p_price, p_amount) are the same as the first row. This row is the second row.

Part 2: Refinement of an order-table for a cinema chain

We will use the short hand notation for each attribute for brevity.

3. The relational schema is not in 3NF since I $\longrightarrow$ St, Si, Ss, Sd would violate the 3NF property since $\{$St, Si, Ss, Sd$\} \not\subseteq$ I, I is not a (super)key and each attribute in $\{$St, Si, Ss, Sd$\} \setminus$ I = $\{$St, Si, Ss, Sd$\}$ is not part of a key.

   To decompose, first, we need to compute the minimal cover of the functional dependencies

   - We can use Decomposition on the first three functional dependencies to get
     $\{$I $\longrightarrow$ St; I $\longrightarrow$ Si; I $\longrightarrow$ Ss; I $\longrightarrow$ Sd; I $\longrightarrow$ Fi; I $\longrightarrow$ Fl; I $\longrightarrow$ Fs; I $\longrightarrow$ Ri; I $\longrightarrow$ Rs; Si $\longrightarrow$ Si; Si $\longrightarrow$ Ss; Si $\longrightarrow$ Sd; Ss $\longrightarrow$ Sd; Sd $\longrightarrow$ Ss; St, Ri $\longrightarrow$ Fi; Fi $\longrightarrow$ Fl; Fi, Si $\longrightarrow$ Fs; Ri $\longrightarrow$ Rs$\}$
   - From Reflexivity, since Si $\subseteq$ Si, we can get Si $\longrightarrow$ Si. So, we don't need to include Si $\longrightarrow$ Si in the minimal cover.
     By Transitivity on I $\longrightarrow$ Si and Si $\longrightarrow$ Ss, we can get I $\longrightarrow$ Ss. So, we don't need I $\longrightarrow$ Ss.
     By Transitivity on I $\longrightarrow$ Si and Si $\longrightarrow$ Sd, we can get I $\longrightarrow$ Sd.

So, we don't need I $\longrightarrow$ Sd.
By Union on I $\longrightarrow$ St and I $\longrightarrow$ Ri; we can get I $\longrightarrow$ St, Ri. By Transitivity on I $\longrightarrow$ St, Ri and St, Ri $\longrightarrow$ Fi; we can get I $\longrightarrow$ Fi. So we don't need I $\longrightarrow$ Fi.
By Transitivity on I $\longrightarrow$ Fi (that we just proved) and Fi $\longrightarrow$ Fl, we can get I $\longrightarrow$ Fl. So, we don't need I $\longrightarrow$ Fl.
By Union on I $\longrightarrow$ Fi and I $\longrightarrow$ Si; we can get I $\longrightarrow$ Fi, Si. By Transitivity on I $\longrightarrow$ Fi, Si and Fi, Si $\longrightarrow$ Fs, we can get I $\longrightarrow$ Fs. So, we don't need I $\longrightarrow$ Fs.
By Transitivity on Si $\longrightarrow$ Ss and Ss $\longrightarrow$ Sd, we can get Si $\longrightarrow$ Sd. So, we don't need Si $\longrightarrow$ Sd.
Thus, we would get
{I $\longrightarrow$ St; I $\longrightarrow$ Si; I $\longrightarrow$ Ri; Si $\longrightarrow$ Ss; Ss $\longrightarrow$ Sd; Sd $\longrightarrow$ Ss; St, Ri $\longrightarrow$ Fi; Fi $\longrightarrow$ Fl; Fi, Si $\longrightarrow$ Fs; Ri $\longrightarrow$ Rs}

- Notice that the set above is already minimal. So it is the minimal cover.

Starting the algorithm, we get know

- $result = \{\}$
- $cover$ = {I $\longrightarrow$ St; I $\longrightarrow$ Si; I $\longrightarrow$ Ri; Si $\longrightarrow$ Ss; Ss $\longrightarrow$ Sd; Sd $\longrightarrow$ Ss; St, Ri $\longrightarrow$ Fi; Fi $\longrightarrow$ Fl; Fi, Si $\longrightarrow$ Fs; Ri $\longrightarrow$ Rs}

First, we have I $\longrightarrow$ St

- Since we have I $\longrightarrow$ St, I $\longrightarrow$ Si, I $\longrightarrow$ Ri $\in$ *cover*, we get $B = \{St, Si, Ri\}$
- So, $result$ = {(I, St, Si, Ri)}

Next, we have Si $\longrightarrow$ Ss

- Since only Si $\longrightarrow$ Ss $\in$ *cover* that starts with Si, $B = \{Ss\}$
- So, $result$ = {(I, St, Si, Ri), (Si, Ss)}

Then, we have Ss $\longrightarrow$ Sd

- Since only Ss $\longrightarrow$ Sd $\in$ *cover* that starts with Ss, $B = \{Sd\}$
- So, $result$ = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd)}

Then, we have Sd $\longrightarrow$ Ss

- Since only Sd $\longrightarrow$ Ss $\in$ *cover* that starts with Sd, $B = \{Ss\}$
- So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss)}

Next, we have St, Ri $\longrightarrow$ Fi

- Since only St, Ri $\longrightarrow$ Fi $\in$ *cover* that starts with St, Ri; $B = \{Fi\}$
- So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss), (St, Ri, Fi)}

Next, we have Fi $\longrightarrow$ Fl

- Since only Fi $\longrightarrow$ Fl $\in$ *cover* that starts with Fi, $B = \{Fl\}$
- So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss), (St, Ri, Fi), (Fi, Fl)}

Next, we have Fi, Si $\longrightarrow$ Fs

- Since only Fi, Si $\longrightarrow$ Fs $\in$ *cover* that starts with Fi, Si; $B = \{Fs\}$
- So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss), (St, Ri, Fi), (Fi, Fl), (Fi, Si, Fs)}

Finally, we have Ri $\longrightarrow$ Rs

- Since only Ri $\longrightarrow$ Rs $\in$ *cover* that starts with Ri, $B = \{Fi\}$
- So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss), (St, Ri, Fi), (Fi, Fl), (Fi, Si, Fs), (Ri, Rs)}

Since *result* doesn't contain the key, which is "I, P, Rp", we need to add it to *result*. So, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (Sd, Ss), (St, Ri, Fi), (Fi, Fl), (Fi, Si, Fs), (Ri, Rs), (I, P, Rp)}
Since (Sd, Ss) $\subseteq$ (Ss, Sd), we will remove (Sd, Ss) from *result*. Thus, *result* = {(I, St, Si, Ri), (Si, Ss), (Ss, Sd), (St, Ri, Fi), (Fi, Fl), (Fi, Si, Fs), (Ri, Rs), (I, P, Rp)}

Minimal cover for each relational schema of the resulting decomposition

- (I, St, Si, Ri)
  $\Longrightarrow$ {I $\longrightarrow$ St; I $\longrightarrow$ Si, I $\longrightarrow$ Ri}

- (Si, Ss)
  $\Longrightarrow$ {Si $\longrightarrow$ Ss}
- (Ss, Sd)
  $\Longrightarrow$ {Ss $\longrightarrow$ Sd}
- (St, Ri, Fi)
  $\Longrightarrow$ {St, Ri $\longrightarrow$ Fi}
- (Fi, Fl)
  $\Longrightarrow$ {Fi $\longrightarrow$ Fl}
- (Fi, Si, Fs)
  $\Longrightarrow$ {Fi, Si $\longrightarrow$ Fs}
- (Ri, Rs)
  $\Longrightarrow$ {Ri $\longrightarrow$ Rs}

From the minimal covers of the relational schemas of the resulting decomposition, we can see that the decomposition is dependency-preserving. It is also proved in the lecture that Decompose-3NF is dependency-preserving.

Furthermore, it is proved in the lecture that Decompose-3NF is also lossless-join. Thus, this decomposition should be lossless-join as well.

4. The relational schema is not in BCNF since I $\longrightarrow$ St, Si, Ss, Sd $\in \mathfrak{S}^+$ but I is not a (super)key.

   DECOMPOSE-BCNF:

   Since I $\longrightarrow$ St, Si, Ss, Sd $\in \mathfrak{S}^+$ violates the BCNF constraint since I is not the (super)key, then

   - $\mathbf{R}_1 = \mathrm{I}^+ =$ {I, St, Si, Ss, Sd, Fi, Fl, Fs, Ri, Rs}
   - $\mathbf{R}_2 = \mathrm{I} \cup \mathrm{Z} =$ {I, P, Rp}
   - Note that {I, P, Rp} satisfies the BCNF constraint, so they would be part of the resulting decomposition.

   For $\mathbf{R}_1$, since Si $\longrightarrow$ Si, Ss, Sd $\in \mathfrak{S}_1^+$ violates the BCNF constraint since Si is not the (super)key, then

- $\mathbf{R}_{1,1} = (\text{Si})^+ = \{\text{Si, Ss, Sd}\}$
- $\mathbf{R}_{1,2} = \text{Si} \cup \text{Z} = \{\text{Si, I, St, Fi, Fl, Fs, Ri, Rs}\}$

For $\mathbf{R}_{1,1}$, since Ss $\longrightarrow$ Sd $\in \mathfrak{S}^+_{1,1}$ violates the BCNF constraint, then

- $\mathbf{R}_{1,1,1} = (\text{Ss})^+ = \{\text{Ss, Sd}\}$
- $\mathbf{R}_{1,1,2} = \text{Ss} \cup \text{Z} = \{\text{Ss, Si}\}$
- Note that {Ss, Sd} and {Ss, Si} satisfies BCNF, so they would be part of the resulting decomposition.

For $\mathbf{R}_{1,2}$, since St, Ri $\longrightarrow$ Fi $\in \mathfrak{S}^+_{1,2}$ violates the BCNF constraint, then

- $\mathbf{R}_{1,2,1} = \{\text{St, Ri}\}^+ = \{\text{St, Ri, Fi, Fl, Rs}\}$
- $\mathbf{R}_{1,2,2} = \{\text{St, Ri}\} \cup \text{Z} = \{\text{St, Ri, I, Fs, Si}\}$
- Note that {St, Ri, I, Fs, Si} satisfies BCNF, so they would be part of the resulting decomposition.

For $\mathbf{R}_{1,2,1}$, since Fi $\longrightarrow$ Fl $\in \mathfrak{S}^+_{1,2,1}$ violates the BCNF constraint, then

- $\mathbf{R}_{1,2,1,1} = (\text{Fi})^+ = \{\text{Fi, Fl}\}$
- $\mathbf{R}_{1,2,1,2} = \text{Fi} \cup \text{Z} = \{\text{Fi, St, Ri, Rs}\}$
- Note that {Fi, Fl} satisfies the BCNF constraint, so they would be part of the resulting decomposition.

For $\mathbf{R}_{1,2,1,2}$, since Ri $\longrightarrow$ Rs $\in \mathfrak{S}^+_{1,2,1,2}$ violates the BCNF constraint, then

- $\mathbf{R}_{1,2,1,2,1} = (\text{Ri})^+ = \{\text{Ri, Rs}\}$
- $\mathbf{R}_{1,2,1,2,2} = \text{Ri} \cup \text{Z} = \{\text{Ri, Fi, St}\}$
- Noe that {Ri, Rs} and {Ri, Fi, St} satisfies the BCNF constraint, so they would be part of the resulting decomposition.

All relational schemas in the resulting decomposition are
{I, P, Rp}, {Ss, Sd}, {Si, Ss}, {St, Ri, Fs, Si, I}, {Fi, Fl}, {Ri, Rs}, {Ri, Fi, St}.

Minimal cover for each relational schema

- $\{\text{I, P, Rp}\}$
  $\Longrightarrow\ \{\}$
- $\{\text{Ss, Sd}\}$
  $\Longrightarrow\ \{\text{Ss} \longrightarrow \text{Sd, Sd} \longrightarrow \text{Ss}\}$
- $\{\text{Si, Ss}\}$
  $\Longrightarrow\ \{\text{Si} \longrightarrow \text{Ss}\}$
- $\{\text{St, Ri, Fs, Si, I}\}$
  $\Longrightarrow\ \{\text{I} \longrightarrow \text{St, I} \longrightarrow \text{Ri, I} \longrightarrow \text{Si}\}$
- $\{\text{Fi, Fl}\}$
  $\Longrightarrow\ \{\text{Fi} \longrightarrow \text{Fl}\}$
- $\{\text{Ri, Rs}\}$
  $\Longrightarrow\ \{\text{Ri} \longrightarrow \text{Rs}\}$
- $\{\text{Ri, Fi, St}\}$
  $\Longrightarrow\ \{\text{St, Ri} \longrightarrow \text{Fi}\}$

This decomposition is not dependency-preserving since we lost the dependency Fi, Si $\longrightarrow$ Fs.

It is proved in the lecture that Decompose-BCNF is lossless-join. Thus, the decomposition is lossless-join.