

# Assignment 6: Decomposition and Normal Forms

Hien Tu - tun1

November 20, 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  $id \rightarrow user\_id$ . 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  $id \rightarrow date$ .
- We also have  $id, inv\_id \rightarrow inv\_confirmed$  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  $id, product \rightarrow p\_amount$  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  $product \rightarrow p\_price$  since each product has its own price. The price of the product wouldn't depend on the event that the product is brought 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  $id \rightarrow inv\_id, inv\_confirmed$ . 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 third 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  $I \rightarrow St, Si, Ss, Sd$  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.

We need to compute the minimal cover of the functional dependencies

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