ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Systems@ETH zürich

Data Modelling and Databases (DMDB)
Spring Semester 2017
Lecturer(s): Gustavo Alonso, Ce Zhang
Assistant(s): Claude Barthels, Eleftherios Sidirourgos, Eliza Wszola, Ingo Müller, Kaan Kara, Renato Marroquín, Zsolt István

ETH Zurich
Systems Group
Date: April 28, 2017
Last update: August 16, 2017

Exercise 8: Functional Dependencies

Solution

The exercises marked with * will be discussed in the exercise session. You can solve the other exercises as practice, ask questions about them in the session, and hand them in for feedback. **All exercises may be relevant for the exam.**

Ask **Ingo (ingo.mueller@inf.ethz.ch)** for feedback on this week's exercise sheet or give it to the TA of your session (preferably stapled and with your e-mail address).

1 Functional Dependencies and Keys*

1.1 Order Relation

Consider the following relation:

Order (`Product_Id, Product_Name, Customer_Id, Customer_Name, Order_Date, Item_Price, Amount, VAT, Gross_Total, Net_Total`)

Assumptions:

- The sales tax (`VAT`) value can vary from product to product (e.g. 8% for books, 16% for luxury items).
- The gross total is the net total price plus the sales tax.
- Customer orders on the same day are combined. We only have one order per customer and per day.
- Properties do not change over time – everything is "write-once".

Questions:

1. Determine all functional dependencies of the relation *Order*.

Solution:

- $Product\_Id \rightarrow Product\_Name, Item\_Price, VAT$
- $Customer\_Id \rightarrow Customer\_Name$
- $Product\_Id, Customer\_Id, Order\_Date \rightarrow Amount$
- $Item\_Price, Amount \rightarrow Net\_Total$
- $Net\_Total, VAT \rightarrow Gross\_Total$

Note: If the product/customer names are unique, one could also consider the following functional dependencies:

- $Product\_Name \rightarrow Product\_Id$
- $Customer\_Name \rightarrow Customer\_Id$

Note: Because of the correlation between Net_Total, VAT and Gross_Total, one could also consider the following functional dependencies:

- $Gross\_Total, VAT \rightarrow Net\_Total$
- $Gross\_Total, Net\_Total \rightarrow VAT$

Note: Because of the correlation between Net_Total, Amount and Item_Price, one could also consider the following functional dependencies:

- $Net\_Total, Amount \rightarrow Item\_Price$
- $Net\_Total, Item\_Price \rightarrow Amount$

Note: More functional dependencies are in the closure of the ones given above, but have been omitted for sake of brevity.

2. Find all the candidate keys.

Solution:

- $\{Product\_Id, Customer\_Id, Date\}$

Note: If product/customer names are unique, we have the following additional key candidates:

- $\{Product\_Id, Customer\_Name, Date\}$
- $\{Product\_Name, Customer\_Name, Date\}$
- $\{Product\_Name, Customer\_Id, Date\}$

1.2 Student – Lecture Relation

Consider the following database:

`Student_Id`	`Student_Address`	`Lecture`	`Teaching_Assisant`
1234	Rämistrasse 72	Data Modelling and Databases	Bob
1280	Rennweg 19	Concepts of Concurrent Computation	Scott
1234	Rämistrasse 72	Visual Computing	Sarah
1299	Börsenstrasse 42	Concepts of Concurrent Computation	Benjamin
1356	Klusplatz 45	Concepts of Concurrent Computation	Benjamin

Note:

- Assume that there is exactly one teaching assistant assigned to each student for every course.

Questions:

1. Determine all functional dependencies of the relation above.

Solution:

- $Student\_Id \rightarrow Address$
- $Student\_Id, Course \rightarrow Teaching\_Assistant$

Note: Formally, it is not possible to deduce functional dependencies from a database as they must hold for *all* possible databases, not just the given one. In this exercise, the sample database serves as illustration of the assumed world.

2. Give an example of a super key and a candidate key.

Solution:

- The set of all the attributes is a trivial super key.
- $Student\_Id, Course$ is a candidate (i.e., minimal) key.

1.3 Closures and Keys

Given a relation R (A,B,C,D,E,F,G) with the following five functional dependencies F:

(1) $A \to BC$ (2) $E \to CF$ (3) $B \to E$ (4) $CD \to EF$ (5) $A \to G$

Questions:

1. Find the closure of A.

Solution:

- (6) $A \to B$ (because (1))
- (7) $A \to C$ (because (1))
- (8) $A \to E$ (transitive property (6) and (3))
- (9) $A \to CF$ (transitive property from (8) and (2))
- (10) $A \to G$ (because (5))

Therefore, $\{A\}+ = \{A, B, C, E, F, G\}$

2. Find a candidate key for R.

Solution: $\{A, D\}$

3. What is the closure of G?

Solution: $\{G\}+ = \{G\}$

1.4 Multiple Choice

Given a relation R (A,B,C,D,E,G) with the following eight functional dependencies F:

$$
\begin{array}{llll}
AB \to C & D \to EG & C \to A & BE \to C \\
BC \to D & CG \to BD & ACD \to B & CE \to AG
\end{array}
$$

For the following statements, decide whether they are true or false. For false statements, explain why you think that they are wrong.

Id	Statement	True	False	Explanation
1	The closure of BC is $\{A, D, E, G\}$		✓	
2	All attributes of R are in the closure of BC.	✓		
3	The closure of AC is $\{A, C\}$	✓		
4	ABC is a super key of R	✓		
5	ABC is a candidate key of R.		✓	
6	BC is the only candidate key of R.		✓	

Explanation:

1. The closure of BC includes the elements themselves. It is $\{A, B, C, D, E, G\}$.
2. From the previous question we know that the closure of BC is $\{A, B, C, D, E, G\}$.
3. $C \rightarrow A$ is the only functional dependency that can be used.
4. Since the closure of BC includes all elements, ABC is a super key.
5. ABC is not minimal, hence it is not a candidate key. BC is a candidate key.
6. There are other candidate keys, e.g., BD and CE.

2 Minimal Basis*

Find a minimal basis of the following sets of functional dependencies.

1. (1) $A \rightarrow BC$ (2) $B \rightarrow C$ (3) $A \rightarrow B$ (4) $AB \rightarrow C$

Solution:

- Reduction of the left side: $AB \rightarrow C$ can be reduced to $A \rightarrow C$ because $\{C\} \subseteq Closure(F, A) = (A), (AB), (ABC)$
- Reduction of the right side: $A \rightarrow BC$ can be reduced to $A \rightarrow B$ because $\{C\} \subseteq Closure(F \backslash \{A \rightarrow BC\} \cup \{A \rightarrow B\}, A) = (A), (AB), (ABC)$
- We now have twice the same functional dependency ($A \rightarrow B$). One can be removed.
- $A \rightarrow C$ can be reduced to $A \rightarrow \oslash$ because $\{C\} \subseteq Closure(F \backslash \{A \rightarrow C\}, A) = (A), (AB), (ABC)$
- As minimal basis remains: $\{A \rightarrow B,\ B \rightarrow C\}$

2. (1) $AB \to C$ (2) $C \to A$ (3) $BC \to D$ (4) $ACD \to B$
(5) $BE \to C$ (6) $CE \to FA$ (7) $CF \to BD$ (8) $D \to EF$

Solution:

- Reduction of the left side: $ACD \to B$ can be reduced to $CD \to B$ because $\{B\} \subseteq Closure(F, CD) = (CD), (CDEF), (CDEFB)$
- Reduction of the right side: $CD \to B$ can be reduced to $CD \to \emptyset$ because $\{B\} \subseteq Closure(F \backslash \{CD \to B\}, CD) = (CD), (CDEF), (CDEFB)$
- Reduction of the right side: $CE \to FA$ can be reduced to $CE \to F$ because $\{A\} \subseteq Closure(F \backslash \{CE \to FA\} \cup \{CE \to F\}, CE) = (CE), (CEF), (CEFA)$
- Reduction of the right side: $CF \to BD$ can be reduced to $CF \to B$ because $\{D\} \subseteq Closure(F \backslash \{CF \to BD\} \cup \{CF \to B\}, CF) = (CF), (CFB), (CFBD)$
- As minimal basis remains:

  $AB \to C,$ $C \to A,$ $BC \to D,$ $BE \to C,$
  $CE \to F,$ $CF \to B,$ $D \to EF$

- An alternative solution would be to right-reduce $CF \to BD$ instead of $CD \to B$

3 Decomposition*

3.1 Student – Lecture Relation

Consider the following relation S and its decomposition into S_1 and S_2.

Relation Name	Attributes
S	`Student_Id, Date_Enrolled, Course_Id, Room_Nr, Professor`
S_1	`Student_Id, Date_Enrolled, Course_Id`
S_2	`Date_Enrolled, Room_Nr, Professor`

Note:

- Assume that each course is always taught only by the same professor and takes place in the same room.

Questions:

1. Determine all functional dependencies of S.

Solution:

- $Course\_Id \to Room\_Nr, Professor$

- $Student\_Id, Course\_Id \rightarrow Date\_Enrolled$

Note: The functional dependencies of S also hold in S_1 and S_2.

2. Why is the decomposition of S to S_1 and S_2 a lossy decomposition? Prove it formally.

Solution: The decomposition of S into S_1 and S_2 is lossless iff $S = S_1 \cup S_2$ (which is given) and $S = S_1 \bowtie S_2$. We show by counter-example that the latter equality does not hold. Let S contain the following tuples:

Stud'_Id	Date_Enr'd	Course_Id	Room_Nr	Professor
12345	2016-02-03	252-0063-00L	CAB G 61	Alonso
54321	2016-02-03	252-0817-00L	NULL	Alonso, Mattern, Roscoe

Then $S_1 = \Pi_{\texttt{Student\_Id, Date\_Enrolled, Course\_Id}}(S)$ contains the following tuples:

Stud'_Id	Date_Enr'd	Course_Id
12345	2016-02-03	252-0063-00L
54321	2016-02-03	252-0817-00L

$S_2 = \Pi_{\texttt{Date\_Enrolled, Room\_Nr, Professor}}(S)$ contains the following:

Date_Enrolled	Room_Nr	Professor
2016-02-03	CAB G 61	Alonso
2016-02-03	NULL	Alonso, Mattern, Roscoe

and $S_1 \bowtie S_2$ contains the following:

Stud'_Id	Date_Enr'd	Course_Id	Room_Nr	Professor
12345	2016-02-03	252-0063-00L	CAB G 61	Alonso
54321	2016-02-03	252-0063-00L	CAB G 61	Alonso
12345	2016-02-03	252-0817-00L	NULL	Alonso, Mattern, Roscoe
54321	2016-02-03	252-0817-00L	NULL	Alonso, Mattern, Roscoe

This is different from S. $\square$

Note: Using the decomposition lemma is *not* correct. Its condition, $R_1 \cap R_2 \rightarrow R_i$, $i \in \{1, 2\}$, is a sufficient condition, but not a nessecary one, i.e., the lemma says that if the condition holds, then the decomposition is lossless, but it does not say anything about the other direction.

3. Show a lossless decomposition of the same relation, and prove that it is lossless.

Solution: S can be decomposed into:

- $S_1(Student\_Id, Date\_Enrolled, Course\_Id)$
- $S_2(Course\_Id, Room\_Nr, Professor)$

$S_1 \cap S_2 = Course\_Id$ and $Course\_Id \rightarrow R2$

3.2 Decomposition Lemma

Given a relation $R = R_1 \cup R_2$ and its decomposition into $R_1 = \prod_{R1}(R)$ and $R_2 = \prod_{R2}(R)$. Assuming $R_1 \cap R_2 \rightarrow R_1$, prove that $R = R_1 \bowtie R_2$.

Hints:

- Assume the elements of R_1 and R_2 have the forms (α, β) and (α, γ) with $\alpha = \{A_1, ..., A_i\}$, $\beta = \{B_1, ..., B_j\}$, and $\gamma = \{C_1, ..., C_k\}$, respectively.
- First prove $R \subseteq R_1 \bowtie R_2$
- Next, prove $R_1 \bowtie R_2 \subseteq R$

Solution: For a, b, c instances of α, β, γ.

Direction 1: $R \subseteq R_1 \bowtie R_2$

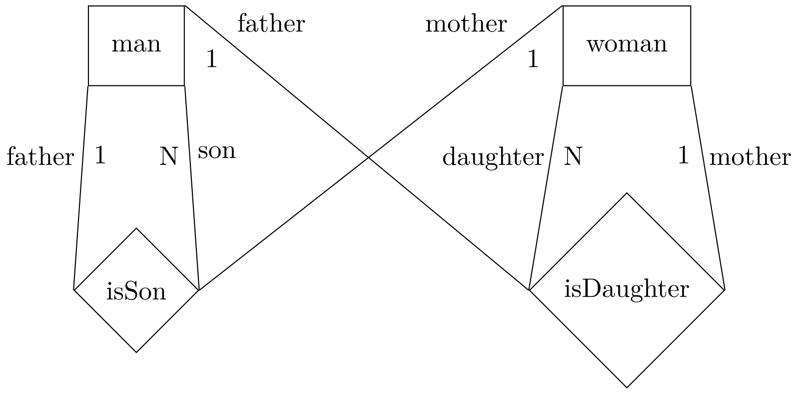
$$\begin{aligned} \text{If } (a,b,c) \in R &\Rightarrow (a,b) \in \textstyle\prod_{R_1}(R) = R_1 \text{ and } (a,c) \in \textstyle\prod_{R_2}(R) = R_2 \\ &\Rightarrow (a,b,c) \in R_1 \bowtie R_2 \end{aligned}$$

Direction 2: $R_1 \bowtie R_2 \subseteq R$

$$\begin{aligned} \text{If } (a,b,c) \in R_1 \bowtie R_2 &\Rightarrow (a,b) \in R_1 \text{ and } (a,c) \in R_2 \\ &\Rightarrow (a,b,c') \in R \text{ such that } \textstyle\prod_{R_1}((a,b,c')) = (a,b), \\ &\quad\;\; \text{and } (a,b',c) \in R \text{ such that } \textstyle\prod_{R_2}((a,b',c)) = (a,c) \\ \\ \alpha \rightarrow \beta &\Rightarrow b = b' \\ &\Rightarrow (a,b',c) = (a,b,c) \in R \end{aligned}$$

4 Functional Dependencies and ER*

Consider the following ER model:



1. Which functional dependencies are given *by the model*?

Solution:

mother, daughter → father
father, daughter → mother
mother, son → father
father, son → mother

2. Which additional functional dependencies would you add to represent the real world more accurately?

Solution:

daughter → mother, father
son → mother, father

3. Determine the minimal basis of all functional dependencies (including the ones you added).

Solution: With the right-reduction rule, the right side of each functional dependency of Part 1 can be made empty because it is already implied by a functional dependency of Part 2. The reduced functional dependency can then be removed and only the following remain:

daughter → mother, father
son → mother, father

4. Discuss what this means for the expressiveness of entity-relationship models.

Solution: As we have discussed in Exercise Sheets 1 and 2, ER cannot express everything we might want to express. The functionalities in ER allow us to express some contraints, but none fit the needs of the example in this exercise. However, we understand better what the functionalities express: If an entity has a functionality of "1", such as "mother", then this entity functionally depends on all other entities of the relationship, e.g., "father, son $\rightarrow$ mother". Futhermore, functional dependencies are a tool with which we can express constraints that we could not have with ER, such as "daughter $\rightarrow$ mother, father".