

Database Tutorial 2: Functional Dependencies and Normal Forms

2015-02-10

1. (9 points) Modeling and Design

flight code	airline	prime flight	operating airline	departure city	departure airport	destination city	destination airport	aircraft type	seats
SK111	SAS	SK111	SAS	Gothenburg	GOT	Frankfurt	FRA	B737	140
LH555	Lufthansa	SK111	SAS	Gothenburg	GOT	Frankfurt	FRA	B737	140
AF111	Air France	AF111	Air France	Gothenburg	GOT	Paris	CDG	A320	170
LH111	Lufthansa	LH111	Lufthansa	Frankfurt	FRA	Paris	CDG	A321	200
LH222	Lufthansa	LH222	Lufthansa	Frankfurt	FRA	Malta	MLA	A320	170
AF222	Air France	AF222	Air France	Paris	ORY	Malta	MLA	A320	170
AB222	Air Berlin	AB222	Air Berlin	Frankfurt	FRA	Munich	MUC	A320	170
KM111	Air Malta	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
LH333	Lufthansa	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
SK222	SAS	KM111	Air Malta	Munich	MUC	Malta	MLA	A319	140
AF333	Air France	AF333	Air France	Paris	CDG	Frankfurt	FRA	A320	170

We assume the following (slightly simplified) conventions for this domain:

- the “flight code” attribute determines all other attributes on a row
- the “prime flight” is the flight code used by the airline operating the flight; the “flight code” in the first column can thus belong to another airline that has a code sharing agreement with the operating airline
- the “prime flight” appears in the table as a “flight code” as well, having itself as prime flight
- each airport has a unique code
- every aircraft of the same type has the same number of seats

(It is a common practice that one and the same flight can be booked using different airlines. Each airline uses a different “flight code”, but the passengers end up in the same plane. The code used by the actual operating airline is called the “prime flight” code. For example, whether you book flight LH333 with Lufthansa or flight SK222 with SAS, you end up in the plane of Air Malta flight KM111.)

- (a) (3 points) Identify the functional dependencies and keys in the domain as described in Question 1. You must have some functional dependencies that are not superkeys. Consider the entire Table 1 as one relation. For functional dependencies, it is enough to list a base (a minimal set that implies all the others).

Definition *Functional Dependency:* For a tuple, if it agrees in attributes $A_1 \dots A_n$ and also has to agree on $B_1 \dots B_m$ we can write $A_1 \dots A_n \rightarrow B_1 \dots B_m$

Solution:

Key : *FlightCode* (optionally also, assuming another company uses just one code for sharing a flight, *primeFlight*, *airline*)

FDs :

FlightCode → *Airline*
FlightCode → *PrimeFlight*
PrimeFlight → *OperatingAirline*
PrimeFlight → *DepartureAirport*
PrimeFlight → *DestinationAirport*
PrimeFlight → *AircraftType*
DepartureAirport → *DepartureCity*
DestinationAirport → *DestinationCity*
AircraftType → *Seats*
 ... (Transitive closure)
 Working here but can be different in reality, also by transitivity trivial:
DepartureAirport, DestinationAirport, OperatingAirline → *PrimeFlight*
DepartureAirport, DestinationAirport, Airline → *FlightCode*

Sample Solution (Exam VT2015):

Functionaldependencies
flightCode → (all attributes) (enough to say airline and primeFlight)
departureAirport → *departureCity*
destinationAirport → *destinationCity*
aircraft → *seats*
primeFlight → all attributes except flightcode and airline
 optionally also:
primeFlightairline → (all attributes) (enough to write flightCode)
primeFlight → *operatingAirline*
keys :
flightCode
 optionally also (assuming another company uses just one code for sharing a flight)
primeFlight, airline

- (b) (4 points) Starting with Table 1 and the functional dependencies and keys in (2a), decompose the relation into BCNF (Boyce-Codd Normal Form). Show all intermediate steps.
 Notice: if you find out that the relation is already in BCNF, then you have done something wrong in (2a).

Definition A relation R is in Boyce-Codd Normal Form (BCNF) if, whenever a nontrivial FD $X \rightarrow A$ holds on R , X is a superkey of R .

- Remember: nontrivial means A is not part of X
- Remember: a superkey is any superset of a key (including the keys themselves).

Definition Algorithm to Decompose R to BCNF:

Given a relation R and FDs F .

1. Identify new FDs using the transitive rule, and add these to F .
2. Look among the FDs in F for a violation $X \rightarrow A$ of BCNF w.r.t. R .
3. Decompose R into two relations
 - One relation R_X containing all the attributes in X^+ .
 - The original relation R , except the values in X^+ that are not also in X (i.e. $R - X^+ + X$), and with a reference from X to X in R_X .
4. Repeat from 2 for the two new relations until there are no more violations.

Definition Algorithm to compute X^+ :

Given a set of FDs, F , and a set of attributes, X :

1. Start with $X^+ = X$.
2. For all FDs $Y \rightarrow B$ in F where Y is a subset of X^+ , add B to X^+ .
3. Repeat step 2 until there are no more FDs that apply.

Solution:

Decompose R :

$R = \{\text{FlightCode}, \text{Airline}, \text{PrimeFlight}, \text{OperatingAirline}, \text{DepartureAirport}, \text{DepartureCity}, \text{DestinationAirport}, \text{DestinationCity}, \text{AircraftType}, \text{Seats}\}$

Step 1: Transitive Rule

1. $\text{FlightCode} \rightarrow \text{Airline}$
2. $\text{FlightCode} \rightarrow \text{PrimeFlight}$
3. $\text{PrimeFlight} \rightarrow \text{OperatingAirline}$
4. $\text{PrimeFlight} \rightarrow \text{DepartureAirport}$
5. $\text{PrimeFlight} \rightarrow \text{DestinationAirport}$
6. $\text{PrimeFlight} \rightarrow \text{AircraftType}$
7. $\text{DepartureAirport} \rightarrow \text{DepartureCity}$
8. $\text{DestinationAirport} \rightarrow \text{DestinationCity}$
9. $\text{AircraftType} \rightarrow \text{Seats}$
10. $(2 \ \& \ 3) \Rightarrow \text{FlightCode} \rightarrow \text{OperatingAirline}$
11. $(2 \ \& \ 4) \Rightarrow \text{FlightCode} \rightarrow \text{DepartureAirport}$
12. $(11 \ \& \ 7) \Rightarrow \text{FlightCode} \rightarrow \text{DepartureCity}$

13. $(2 \ \& \ 5) \Rightarrow \textit{FlightCode} \rightarrow \textit{DestinationAirport}$

14. $(13 \ \& \ 8) \Rightarrow \textit{FlightCode} \rightarrow \textit{DestinationCity}$

15. $(2 \ \& \ 6) \Rightarrow \textit{FlightCode} \rightarrow \textit{AircraftType}$

16. $(15 \ \& \ 9) \Rightarrow \textit{FlightCode} \rightarrow \textit{Seats}$

17. $(4 \ \& \ 7) \Rightarrow \textit{PrimeFlight} \rightarrow \textit{DepartureCity}$

18. $(5 \ \& \ 8) \Rightarrow \textit{PrimeFlight} \rightarrow \textit{DestinationCity}$

19. $(6 \ \& \ 9) \Rightarrow \textit{PrimeFlight} \rightarrow \textit{Seats}$

Step 2: Find a violation

$\textit{PrimeFlight} \rightarrow \textit{OperatingAirline}$

Step 3.1: Compute $\textit{PrimeFlight}^+$

$\textit{PrimeFlight}^+ = \{\textit{PrimeFlight}\}$

$\textit{PrimeFlight}^+ = \textit{PrimeFlight}^+ \cup$

$\{\textit{OperatingAirline}, \textit{DepartureAirport}, \textit{DestinationAirport}, \textit{AircraftType}\}$

$\textit{PrimeFlight}^+ = \textit{PrimeFlight}^+ \cup \{\textit{DepartureCity}, \textit{DestinationCity}, \textit{Seats}\}$

Step 3.2: Split relation

$R1 = \textit{PrimeFlight}^+ = \{\textit{PrimeFlight}, \textit{OperatingAirline}, \textit{DepartureAirport}, \textit{DestinationAirport}, \textit{AircraftType}, \textit{DepartureCity}, \textit{DestinationCity}, \textit{Seats}\}$

With PrimeFlight the new key and following FDs (Subset of the original FDs still holding on new relation)

1. $\textit{PrimeFlight} \rightarrow \textit{OperatingAirline}$

2. $\textit{PrimeFlight} \rightarrow \textit{DepartureAirport}$

3. $\textit{PrimeFlight} \rightarrow \textit{DestinationAirport}$

4. $\textit{PrimeFlight} \rightarrow \textit{AircraftType}$

5. $\textit{DepartureAirport} \rightarrow \textit{DepartureCity}$

6. $\textit{DestinationAirport} \rightarrow \textit{DestinationCity}$

7. $\textit{AircraftType} \rightarrow \textit{Seats}$

8. $\textit{PrimeFlight} \rightarrow \textit{DepartureCity}$

9. $\textit{PrimeFlight} \rightarrow \textit{DestinationCity}$

10. $\textit{PrimeFlight} \rightarrow \textit{Seats}$

$R = R - \textit{PrimeFlight}^+ \cup \{\textit{PrimeFlight}\} = \{\textit{FlightCode}, \textit{Airline}, \textit{PrimeFlight}\}$

New reference: $\textit{PrimeFlight} \rightarrow R1.\textit{PrimeFlight}$

New FDs that still hold:

1. $\textit{FlightCode} \rightarrow \textit{Airline}$

2. $\textit{FlightCode} \rightarrow \textit{PrimeFlight}$

⇒ in BCNF

Now looking at R1

Step 2: Find a violation

$DepartureAirport \rightarrow DepartureCity$

Step 3.1: Compute $DepartureAirport^+$

$DepartureAirport^+ = \{DepartureAirport\} \cup \{DepartureCity\}$

Step 3.2: Split relation

$R2 = DepartureAirport^+ = \{\underline{DepartureAirport}, DepartureCity\}$

With $DepartureAirport$ as new Key and new FD

1. $DepartureAirport \rightarrow DepartureCity$

⇒ in BCNF

$R1 = R1 - DepartureAirport^+ \cup \{DepartureAirport\} = \{\underline{PrimeFlight}, OperatingAirline, DepartureAirport, DestinationAirport, AircraftType, DestinationCity, Seats\}$

New reference: $DepartureAirport \rightarrow R2.DestinationAirport$

New FDs that still hold:

1. $PrimeFlight \rightarrow OperatingAirline$
2. $PrimeFlight \rightarrow DepartureAirport$
3. $PrimeFlight \rightarrow DestinationAirport$
4. $PrimeFlight \rightarrow AircraftType$
5. $DestinationAirport \rightarrow DestinationCity$
6. $AircraftType \rightarrow Seats$
7. $PrimeFlight \rightarrow DestinationCity$
8. $PrimeFlight \rightarrow Seats$

Again R1

Step 2: Find a violation

$DestinationAirport \rightarrow DestinationCity$

Step 3.1: Compute $DestinationAirport^+$

$DestinationAirport^+ = \{DestinationAirport\} \cup \{DestinationCity\}$

Step 3.2: Split relation

$R3 = DestinationAirport^+ = \{\underline{DestinationAirport}, DestinationCity\}$

with $DestinationAirport$ new key and new FD

1. $DestinationAirport \rightarrow DestinationCity$

⇒ in BCNF

$R1 = R1 - DestinationAirport^+ \cup DestinationAirport = \{\underline{PrimeFlight}, OperatingAirline, DepartureAirport, DestinationAirport, AircraftType, Seats\}$

New reference: $DestinationAirport \rightarrow R3.DestinationAirport$ New FDs that still hold:

1. $PrimeFlight \rightarrow OperatingAirline$
2. $PrimeFlight \rightarrow DepartureAirport$
3. $PrimeFlight \rightarrow DestinationAirport$

4. $PrimeFlight \rightarrow AircraftType$

5. $AircraftType \rightarrow Seats$

6. $PrimeFlight \rightarrow Seats$

Again R1:

Step 2: Find a violation

$AircraftType \rightarrow Seats$

Step 3.1: Compute $AircraftType^+$

$AircraftType^+ = \{AircraftType\} \cup \{Seats\}$

Step 3.2: Split relation

$R4 = AircraftType^+ = \{AircraftType, Seats\}$

New key AircraftType and new FD:

1. $AircraftType \rightarrow Seats$

\Rightarrow in BCNF

$R1 = R1 - AircraftType^+ \cup \{AircraftType\} = \{PrimeFlight, OperatingAirline, DepartureAirport, DestinationAirport, AircraftType\}$

New reference: AircraftType \rightarrow R4.AircraftType

New FDs that still hold:

1. $PrimeFlight \rightarrow OperatingAirline$

2. $PrimeFlight \rightarrow DepartureAirport$

3. $PrimeFlight \rightarrow DestinationAirport$

4. $PrimeFlight \rightarrow AircraftType$

\Rightarrow in BCNF

Final result: $R = \{FlightCode, Airline, PrimeFlight\}$

PrimeFlight \rightarrow R1.PrimeFlight

$R1 = \{PrimeFlight, OperatingAirline, DepartureAirport, DestinationAirport, AircraftType\}$

DepartureAirport \rightarrow R2.DepartureAirport

DestinationAirport \rightarrow R3.DestinationAirport AircraftType \rightarrow R4.AircraftType

$R2 = \{DepartureAirport, DepartureCity\}$

$R3 = \{DestinationAirport, DestinationCity\}$

$R4 = \{AircraftType, Seats\}$

Sample Solution (Exam VT2015):

R1(aircraftType, seats)

R2(destinationAirport, destinationCity)

R3(departureAirport, departureCity)

primeFlight \rightarrow operatingAirline

brings another relation if considered

R4(flightCode, airline, primeFlight, operatingAirline, departureAirport, destinationAirport, aircraftType)

departureAirport \rightarrow R3.departureAirport

destinationAirport \rightarrow R2.destinationAirport

aircraftType \rightarrow R1.aircraftType

- (c) (2 points) Suppose you know the attributes of a relation and that it has no functional dependencies.
- Do you have enough information to bring it to BCNF. If yes, how? If no, why?

Solution:
Sample Solution (Exam VT2015):
 Yes, because it is already in BCNF.

- Do you have enough information to bring it to the Fourth Normal Form (4NF). If yes, how? If no, why?

Definition 4NF is a strengthening of BCNF to handle redundancy that comes from independence.

- An MVD $X \twoheadrightarrow Y$ is trivial for R if $Y \subseteq X$ and $X \cup Y = R$
- Non-trivial $X \twoheadrightarrow Y$ violates 4NF for a relation R if X is not a superkey. (Note that what is (or is not) a superkey is still determined by FDs only)

Definition Multivalued dependencies (MVD) are another kind of dependencies. An MVD $X \twoheadrightarrow Y$ says that X determines Y independently of all other attributes. An MVD $X \twoheadrightarrow Y$ is an assertion that if two tuples of a relation agree on all the attributes of X , then their components in the set of attributes Y may be swapped, and the result will be two tuples that are also in the relation.

Example:

Code	Room	Teacher
TDA357	VR	Niklas Broberg
TDA357	VR	Rogardt Haldal
TDA357	HC1	Niklas Broberg
TDA357	HC1	Rogardt Haldal

$code \twoheadrightarrow room$
 $code \twoheadrightarrow teacher$

Every FD is an MVD (but of course not the other way around)

Solution:
Example Solution (Exam VT2015): No. There can be multivalued dependencies that are not functional dependencies.

2. (10 points) Suppose we have relation $R(A, B, C, D, E, F, G)$ with keys $\{A, B, C\}$, $\{A, C, D\}$ and $\{A, C, G\}$, and functional dependencies $A \rightarrow E$, $\{A, B\} \rightarrow D$, $\{A, B, C\} \rightarrow F$, $\{A, B, C\} \rightarrow G$, $\{C, D\} \rightarrow G$, $E \rightarrow F$, $G \rightarrow B$.
- (a) i. (1 point) State, with reasons, which of the FDs listed above violate BCNF.

Solution:
Sample Solution (Exam HT2014):
 Only $A, B, C \rightarrow F$ and $A, B, C \rightarrow G$ fulfill definition of BCNF since their left sides are keys,
 the remaining FDs $A \rightarrow E$, $A, B \rightarrow D$, $A, B, C \rightarrow F$, $C, D \rightarrow G$, $E \rightarrow F$, $G \rightarrow B$ don't
 \Rightarrow not in BCNF

- ii. (4 points) Decompose relation R to BCNF. Show each step in the normalization process, and at each step indicate which functional dependency is being used. Indicate keys and references for the resulting relations.

Solution:

Sample Solution (Exam HT2014):

Decompose R on $A \rightarrow E$

$A^+ = A, E, F$

$R1(\underline{A}, E, F)$

$R2(A, B, C, D, G)$

Reference: $A \rightarrow R1.A$

Decompose R2 on $A, B \rightarrow D$

$A, B^+ = A, B, D$

$R21(\underline{A}, \underline{B}, D)$

Reference: $A \rightarrow R1.A$

$R22(A, B, C, G)$

Reference: $(A, B) \rightarrow R21(A, B)$

Decompose R1 on $E \rightarrow F$

$E^+ = E, F$

$R11(\underline{E}, F)$

$R12(E, \underline{A})$

Reference: $E \rightarrow R11.E$

Decompose R22 on $G \rightarrow B$ $G^+ = G, B$ $R221(\underline{G}, B)$ $R222(\underline{A}, \underline{C}, \underline{G})$ Reference: $G \rightarrow R221.G$

Update reference for R21: $A \rightarrow R12.A$

- (b) i. (1 point) Which attributes of R are prime?

Definition An attribute is prime in relation R if it is a member of any key of R.

Solution:

Sample Solution (Exam HT2014):

A, B, C, D, G are prime and E, F are not

- ii. (1 point) State, with reasons, which of the FDs listed above violate 3NF.

Definition 3NF is a weakening of BCNF. Non-trivial $X \rightarrow A$ violates 3NF for R if X is not a superkey of R, and A is not prime in R.

Solution:

$A \rightarrow E$: A is no superkey and E is not prime \Rightarrow Violates 3NF

$AB \rightarrow D$: AB is no superkey but D is prime \Rightarrow OK

$ABC \rightarrow F$: ABC is a superkey but F is not prime \Rightarrow OK

$ABC \rightarrow G$: ABC is a superkey and G is prime \Rightarrow OK

$CD \rightarrow G$: CD is no superkey but G is prime \Rightarrow OK

$E \rightarrow F$: E is no superkey and F is not prime \Rightarrow Violates 3NF

$G \rightarrow B$: G is no superkey but B is prime \Rightarrow OK

- iii. (3 points) Decompose relation R to 3NF.

Definition Algorithm to decompose R to 3NF: Given a relation R and a set of FDs F :

1. Compute the minimal basis of F .
Minimal basis means F , except remove $A \rightarrow C$ if you have $A \rightarrow B$ and $B \rightarrow C$.
2. Group together FDs with the same LHS.
3. For each group, create a relation with the LHS as the key.
4. If no relation contains a key of R , add one relation containing only a key of R .

Solution: Step 1: Minimal Basis

FDs are already minimal:

$A \rightarrow E$
 $A, B \rightarrow D$
 $A, B, C \rightarrow F$
 $A, B, C \rightarrow G$
 $C, D \rightarrow G$
 $E \rightarrow F$
 $G \rightarrow B$

Step 2: Group LHS

$A \rightarrow E$
 $E \rightarrow F$
 $G \rightarrow B$
 $A, B \rightarrow D$
 $C, D \rightarrow G$
 $A, B, C \rightarrow F$
 $A, B, C \rightarrow G$

Step 3: Create relations

see Example Solution

Sample Solution (Exam HT2014):

$R_1(\underline{A}, E)$
 $R_2(\underline{A}, B, D)$
 $R_3(\underline{A}, B, C, F, G)$
 $R_4(\underline{C}, D, G)$
 $R_5(\underline{E}, F)$
 $R_6(\underline{G}, B)$

3. (8 points)

- (a) (5 points) Give an example of a relation that is in BCNF (Boyce-Codd Normal Form) but not in 4NF (Fourth Normal Form). Show all the information that is needed: attributes, dependencies, keys, etc, clearly stating what the 4NF violations are, as well as an instance (a set of tuples).

Solution: Cmp. Course Book 106pp

name	street	city	title	year
C. Fisher	123 Maple St.	Hollywood	Star Wars	1977
C. Fisher	5 Locust Ln.	Malibu	Star Wars	1977
C. Fisher	123 Maple St.	Hollywood	Empire Strikes Back	1980
C. Fisher	5 Locust Ln.	Malibu	Empire Strikes Back	1980
C. Fisher	123 Maple St.	Hollywood	Return of the Jedi	1983
C. Fisher	5 Locust Ln.	Malibu	Return of the Jedi	1983

No BCNF violation (no non-trivial FDs at all)
 But: $name \twoheadrightarrow street, city$ is a MVD and a 4NF violation
 the MVD is not trivial ($\{street, city\} \not\subseteq \{name\}$) and name is no superkey in the relation

(b) (3 points) Transform your relation in 3a) to 4NF.

Definition The signature of a tuple, S , is the set of all its attributes, $\{A_1, \dots, A_n\}$. A relation R of signature S is a set of tuples with signature S . But we will sometimes also say “relation” when we mean the signature itself.

Definition Let X, Y, Z be disjoint subsets of a signature S such that $S = X \cup Y \cup Z$. Then Y has a multivalued dependency on X in R , written $X \twoheadrightarrow Y$, if for all tuples t, u in R , if $t.X = u.X$ then there is a tuple v in R such that $v.X = t.X$, $v.Y = t.Y$ and $v.Z = u.Z$
 An alternative notation is $X \twoheadrightarrow Y|Z$, emphasizing that Y is independent of Z .

If two tuples have the same value for X , different values for Y and different values for the Z attributes, then there must also exist tuples where the values of Y are exchanged, otherwise Y and Z are not independent!

If we have:

code	name	room	teacher
TDA357	Databases	VR	Niklas Broberg
TDA357	Databases	HC1	Rogardt Heldal

we also need to have

TDA357	Databases	VR	Rogardt Heldal
TDA357	Databases	HC1	Niklas Broberg

otherwise room and teacher would not be independent!

$Code \twoheadrightarrow Name$ There are four possible combinations of values for the attributes room and teacher, and the only possible value for the name attribute, Databases, appears in combination with all of them.

$Code \twoheadrightarrow Teacher$ There are two possible combinations of values for the attributes name and room, and all possible values of the attribute teacher appear with both of these combinations.

$Code \twoheadrightarrow Room$ There are two possible combinations of values for the attributes name and teacher, and all possible values of the attribute room appear with both of these combinations

Algorithm:

Consider a relation R with signature S and a set M of multivalued dependencies. R can be brought to 4NF by the following steps:

1. If R has no 4NF violations, return R
2. If R has a violating multivalued dependency $X \twoheadrightarrow Y$, decompose R to two relations
 - $R1$ with signature $X \cup \{Y\}$
 - $R2$ with signature $S - Y$
3. Apply the above steps to $R1$ and $R2$

Solution:

$name \twoheadrightarrow street, city$ is the 4NF violation

$R1 = \{name\} \cup \{street, city\} = \{name, street, city\}$

$R2 = \{name, street, city, film, year\} - \{street, city\} = \{name, year, film\}$

\Rightarrow no more 4NF violations