Esra Şekerci
2698215

IS 503 – Assignment 3, due: 10 May 2024

1) Consider the following relation and the set of functional dependencies (FDs):

R = (A, B, C, D, E, F, G)

F+ =
{ A → B,
BC → D,
E → AF,
BF → A,
CG → D,
CG → B,
FG → C,
AFG → B,
G → EF,
CD → B,
BCF → A,
AE → G }

a. Find the minimal cover of the FD set.

As stated in class, it's essential to follow three steps in the specified order,

- Put each FD in canonical form, i.e. there is only 1 attribute on the rhs
- Remove redundant attributes in the dependencies
- Remove redundant dependencies

Step 1:
A → B
BC →D
E → A
E → F
BF → A
CG → D
CG → B
FG → C
AFG → B
G → E
G → F
CD → B
BCF → A
AE → G

Step 2:
E → A
A → B
E → F
E → G (considering E → A, rather than using AE → G, **reflexive rule**)
G → C (considering G → F, rather than using FG → C, **reflexive rule**)
G → D (considering G → C, rather than using CG → D)

Step 3 | Minimal set:
E → A
E → F
A → B
E → G
G → C
G → D

b. Find the candidate key(s) of R.

Let's show how to find it step by step, using **Amstrong's Axioms**

A+ = {A, B}
B+ = {B}
C+ = {C}
D+ = {D}
E+ = {E, A, B, F, G, C, D}
F+ = {F}
G+ = {G, E, F, A, B, C, D}

Here,

E → A (given)
E → F (given)
Since A → B and A is element of E+ under F then E → B
AE → G again A is element of E+ under F then E → G
FG → C we know that F and G are elements of E+ so E → C
BC → D as B and C are now elements of E+ here we get E → D

G → E (given)
G → F (given)
E → A as G → E so G → A
A → B from uppermentioned facts G → B
FG → C so G → C
GC → D here G → C so G → D

Therefore, we have two minimal keys;
E+ = {A, B, C, D, E, F, G} and
G+ = {A, B, C, D, E, F, G}

2) Consider the following relation and the set of functional dependencies (FDs):

$$R = (A, B, C, D, E, G)$$

$$F+ = \{ AB \to C, C \to A, BC \to D, ACD \to B, D \to E, D \to G, BE \to C, CG \to B, CG \to D, CE \to A, CE \to G \}$$

a. Find the minimal cover of the FD set.

Step 1:All the dependencies are in **canonical form**.
Step 2: We now need to determine whether we have any **redundant attributes** in dependencies

$AB \to C$
$C \to A$
$BC \to D$
$ACD \to B$ turns into $CD \to B$ since $C \to A$ by pseudotransitivity
$D \to E$
$D \to G$
$BE \to C$
$CG \to B$
$CG \to D$
$CE \to A$ can be replaced with $C \to A$, so it is eliminated
$CE \to G$

C+ = {C, A}
D+ = {D, E, G}
CD+ = {C, D, A, B, E, G}
AB+ = {A, B, C, D, E, G}
BC+ = {B, C, D, E, G, A}
BE+ = {B, E, C, A, D, G}
CG+ = {C, G, D, E, A, B}
CE+ = {C, E, A, G, B, D}

Step 3: Remove **redundant FD** in F+.

$AB \rightarrow C$
$C \rightarrow A$
$BC \rightarrow D$
$CD \rightarrow B$
$D \rightarrow E$
$D \rightarrow G$
$BE \rightarrow C$
$CG \rightarrow B$
$CG \rightarrow D$
$CE \rightarrow G$

b) Find the minimal key/keys of R.

CD+ = {C, D, A, B, E, G}
AB+ = {A, B, C, D, E, G}
BC+ = {B, C, D, E, G, A}
BE+ = {B, E, C, A, D, G}
CG+ = {C, G, D, E, A, B}
CE+ = {C, E, A, G, B, D}

c) If R is not in BCNF, decompose it into a set of relations that satisfy BCNF.

Boyce-Codd Normal Form (BCNF) requires that every functional dependency $X \rightarrow A$ in a relation schema R satisfies the condition that X is a superkey of R. This ensures that non-superkey attributes cannot determine other attributes, disallowing dependencies of prime attributes on non-prime attributes. BCNF is a stricter form of normalization compared to third normal form (3NF) and helps maintain data integrity by avoiding certain anomalies.

Given that we couldn't identify a single minimal key, we might need to proceed with a more generalized approach to decompose relation R into a set of relations that satisfy Boyce-Codd Normal Form (BCNF).

3) Consider the following relation and its minimal cover:

R = (A, B, C, D, E, F, G, H, I, J)

F+ =
{ D→I,
E→G,
E→J,
H→B,
H→C,
AC→E,
CI→D,
CI→E,
GJ→A,
GJ→F }

a. Find the candidate key(s) of R.

- H do not appear on the right side, it must be in the key.

H+ = {H, B, C,}
HA+ = {HABCEG}
HB+ = {HBC}
HC+ = {HBC}
HD+ = {HDBCIEGJAF}
HE+ = {HEBCGJAF}
HF+ = {HFBC}
HG+ = {HGBC}
HI+ = {HIBCDEJGAF}
HJ+ = {HJBC}

Therefore candidate keys are HD+ and HI+.

b. Identify the best normal form that R satisfies.

Fortunately, the minimal FD set is provided prior to normalization. Now, we need to systematically verify our FDs for each normalization form, as follows:

1NF is indeed fundamental to defining a relation, as it lays the groundwork for subsequent normal forms. Ensuring atomic values for each attribute facilitates efficient data management and normalization processes. Moving on to 2NF, it enhances data organization by refining the principles established in 1NF. By introducing concepts such as functional dependencies and primary keys, 2NF aims to further eliminate redundancy and dependency issues in relational databases. Furthermore, in 2NF all attributes depend on the whole key.

Candidate keys for relation R are {HD, HI}, with prime attributes being {H, D, I} and non-prime attributes being {A, B, C, E, F, G, J}. In terms of functional dependencies (FDs), the left-hand side must be a proper subset of the candidate key, while the right-hand side must be a non-prime attribute.
Upon evaluation:

- For attribute H: As H can determine B and C, which are non-prime attributes, we conclude that relation R satisfies 2NF.
- For attribute D: Since D can determine I, a prime attribute, relation R does not satisfy 2NF. Further analysis for other partial dependencies is unnecessary.

Thus, the best normal form that relation R satisfies is 1NF. Given the failure to meet 2NF, there is no need to evaluate for 3NF and BCNF.

c. For each of the following decompositions, comment whether they are dependency-preserving and give a lossless join.

i. R1(ACEFGJ), R2(BCDHI), R3(CEGIJ)

R1(ACEFGJ,F1) F1 = { GJ → A, GJ → F, E → G, E → J, AC→E }
R2(BCDHI,F2) F2 = { D→I, H→B, H→C, CI→D }
R3(CEGIJ,F3) F3 = { E→G, E→J, CI→E }

The union of the functional dependencies (F1 ∪ F2 ∪ F3) yields the set
{D → I, E → G, E → J, H → B, H → C, AC → E, CI → E, GJ → A, GJ → F}.

The conclusion drawn is that Decomposition 1 is **dependency-preserving**.

	A	B	C	D	E	F	G	H	I	J
R1	a1	b1,2	a3	b1,4	a5	a6	a7	b1,8	b1,9	a10
R2	b2,1	a2	a3	a4	b2,5	b2,6	b2,7	a8	a9	b2,10
R3	b3,1	b2,2	a3	b3,4	a5	b3,6	a7	b3,8	a9	a10

The absence of a column solely containing "a" values indicates that Decomposition 1 does not yield a lossless join.

ii. R1(CDEI), R2(ACEFGJ), R3(BCGH)

R1(CDEI,F1) F1 = { D→I, CI→D, CI→E}
R2(ACEFGJ,F2)F2 = { E→G, E→J, AC→E, GJ→A, GJ→F }
R3(BCGH,F3)F3 = { H→B, H→C }

The union of the functional dependencies (F1 ∪ F2 ∪ F3) results in the set
{D → I, E → G, E → J, H → B, H → C, AC → E, CI → D, CI → E, GJ → A, GJ → F}.

The conclusion drawn is that Decomposition 2 is **dependency-preserving**.

	A	B	C	D	E	F	G	H	I	J
R1	b1,1	b1,2	a3	a4	a5	b1,6	b17	b1,8	a9	b1,10
R2	a1	b2,2	a3	b2,4	a5	a6	a7	b2,8	b2,9	a10
R3	b3,1	a2	a3	b3,4	b3,5	b3,6	a7	a8	b3,9	b3,10

The absence of a column containing exclusively "a" values indicates that Decomposition 2 does not ensure a lossless join.

iii. R1(BDH), R2(AEFGJ), R3(CDEHI)

R1(BDH, F1) F1 = { }
R2(AEFGJ, F2)F2 = { E→G, E→J, GJ→A, GJ→F }
R3(CDEHI, F3)F3 = { D→I, H→C, CI→D, CI→E }

F1 U F2 U F3 = { D→I, E → G, E → J, H→C, CI→D, CI→E, GJ → A, GJ → F }
(F1 U F2 U F3)+ = { D→I, E → G, E → J, H→C, CI→D, CI→E, GJ → A, GJ → F }

The conclusion drawn is that Decomposition 3 is **not dependency preserving**.

	A	B	C	D	E	F	G	H	I	J
R1	b1,1	a2	b1,3	a4	b1,5	b1,6	b17	a8	b1,9	b1,10
R2	a1	b2,2	b2,3	b2,4	a5	a6	a7	b2,8	b2,9	a10
R3	b3,1	b3,2	a3	a4	a5	b3,6	b3,7	a8	a9	b3,10

The absence of a column exclusively filled with "a" values indicates that Decomposition 3 does not result in a lossless join.