CS 377: Database Systems

Homework #3 Solutions

1. **Functional Dependencies** (10 + 4 points):

   Consider the following relation:

Tuple #	A	B	C
1	10	b1	c1
2	10	b2	c2
3	11	b4	c1
4	12	b3	c4
5	13	b1	c1
6	14	b3	c4

   (a) Given the above database content, which of the following functional dependencies **may hold** in the above relation. If the functional dependency is invalid, explain why by specifying the tuples that cause the violation.

   i. A $\rightarrow$ B
   ii. B $\rightarrow$ C
   iii. C $\rightarrow$ B
   iv. B $\rightarrow$ A
   v. C $\rightarrow$ A

   (b) Does the above relation have a potential candidate key that does not include all attributes in the relation? If it does, what is it? If it does not, why not?

(**ANSWER**)

(a) i. CANNOT HOLD. Tuples 1 and 2 have the same value for attribute A but different values for B.

ii. MAY HOLD. For the tuples that have the same value for attribute B (1 and 5, 4 and 6), the same values are present for C.

iii. CANNOT HOLD. Tuples 1,3, and 5 have the same value for attribute C (c1), but different values for B (b1 and b4).

iv. CANNOT HOLD. Tuples 1 and 6 have the same value for attribute B (b1), but different values of A (10, 13).

v. CANNOT HOLD. Tuples 1 and 5 have the same value for attribute C (c1), but different values of A (10, 13).

(b) Potential keys are (A, B) and (A, C). (B,C) is not a potential key because of the tuples: (10, b1, c1) and (13, b1, c1).

2. **Closures & Keys** (10 + 3 points): Consider a relation:

   `R(A, B, C, D, E)`

   with the following dependencies:

   - A → B, C
   - C, D → E
   - B → D
   - E → A

   (a) Compute the closures of all the functional dependencies

   (b) List all the candidate keys for R

   (**ANSWER**)

   (a)
   - $\{A\}^+ = \{A, B, C, D, E\}$
   - $\{C, D\}^+ = \{A, B, C, D, E\}$
   - $\{B\}^+ = \{D, B\}$
   - $\{E\}^+ = \{A, B, C, D, E\}$

   (b) From the previous part, we've identified at least three superkeys. SInce A and E only have a single attribute, they must be keys. Next we look at $\{C, D\}^+$. We find the closures of both $\{C\}^+$ and $\{D\}^+$, and note that they do not contain all the attributes. Thus (C, D) must also be key. The remaining one is $\{B\}^+$, which is missing A, C, E. Since A and E are keys, we try C. Since $\{B, C\}^+ = \{A, B, C, D, E\}$, we have found another key. Thus the candidate keys are (A), (B, C), (C, D), and (E).

3. **Dynamite Database - BCNF Normalization** (5 + 5 + 15): Youre designing a database for an online gaming service named Dynamite. The database should hold customer information, game information and sales. Consider the game sales relation with a schema and functional dependencies as follows:

   `R(saleID, saleTime, gameTitle, gamePublisher, publisherCutPercent, quantity, price,customerID, address, creditCardNo)`

   - gameTitle → price
   - gameTitle → gamePublisher
   - gamePublisher → publisherCutPercent
   - customerID → address
   - customerID → creditCardNo
   - saleID → saleTime, gameTitle, quantity, price, customerID

   (a) What are the key(s) of the relation?

   (b) Which of these functional dependencies violate BCNF?

(c) Decompose the relation to obtain a lossless decomposition of R that are in BCNF. Make sure it is clear what the keys are for each relation.

(**ANSWER**)

(a) Compute the base set which is saleID. Note that saleID is a key, so we are done.

(b) Check all the functional dependencies.

- gameTitle → price is a violation as gameTitle is not a key.
- gameTitle → gamePublisher is a violation as gameTitle is not a key.
- gamePublisher → publisherCutPercent is a violation as gamePublisher is not a key.
- customerID → address is a violation as customerID is not a key.
- customerID → creditCardNo is a violation as customerID is not a key.
- saleID → saleTime, gameTitle, quantity, price, customerID is okay as saleID is a key.

(c) We will use the previous part to decompose our relation.

- gameTitle → price is a violation and $\{\text{gameTitle}\}^+$ = gameTitle, price, gamePublisher, publisherCutPercent. We create a new relation R1(gameTitle, price, gamePublisher, publisherCutPercent) and R2(saleID, gameTitle, saleTime, quantity, customerID, address, creditCardNo).
- Check functional dependencies against R1. Note that gamePublisher → publisherCutPercent is a violation as both are non-key attributes. Thus we want to break it up again by finding $\{\text{gamePublisher}\}^+$ = gamePublisher, publisherCutPercent. This yields R11(gamePublisher, publisherCutPercent) and R12(gameTitle, price, gamePublisher).
- Check R11 and R12 for BCNF. It seems okay with the other functional dependencies.
- Check R2. customerID → address is a violation and $\{\text{customerID}\}^+$ = customerID, address, creditCardNo. We create a new relation R21(customerID, address, creditCardNo) and R22(saleID, gameTitle, customerID, saleTime, quantity).
- We check the FDs against $R21$ and $R22$ for BCNF violation. None, so we are done.

Thus the resulting decomposition and relations are

- R11(gamePublisher, publisherCutPercent)
- R12(gameTitle, price, gamePublisher)
- R21(customerID, address, creditCardNo)
- R22(saleID, gameTitle, customerID, saleTime, quantity)

4. **3NF & BCNF** (10 + 8 + 10 + 15 + 5 points):

Consider the following relation:

```
R(A, B, C, D, E, F, G, H)
```

and the following dependencies:

- B → C, D
- B, F → H

- C → A, G
- C, E, H → F
- C, H → B

(a) What are the key(s) of the relation?

(b) Which of these functional dependencies violate 3NF? What about BCNF?

(c) Decompose the relation to obtain a lossless decomposition of R that are in 3NF. Make sure it is clear what the keys are for each relation.

(d) Decompose the relation to obtain a lossless decomposition of R that are in BCNF. Make sure it is clear what the keys are for each relation.

(e) Is the resulting decomposition functional dependency-preserving? Explain why it does or does not.

(**ANSWER**)

(a) We will illustrate the first heuristic to figure out the keys. Find the closure of each of the FDs.

- $\{B\}^+$ = B, C, D, A, G
- $\{B, F\}^+$ = B, C, D, A, G, H, F
- $\{C\}^+$ = C, A, G
- $\{C, E, H\}^+$ = C, E, H, F, A, G, B, D
- $\{C, H\}^+$ = C, H, B, A, G, D

Second step is to add / subtract based on the closure of each set. So for the first one $\{B\}^+$, we are missing E, F, and H.

- Add E: $\{B, E\}^+$ = B, C, D, A, G, E
- Add F: $\{B, F\}^+$ = B, C, D, A, G, H, F
- Add H: $\{B, H\}^+$ = B, C, D, A, G, H
- Add E, F: $\{B, E, F\}^+$ = B, C, D, A, G, H, F, E which is a key!
- Add E, H: $\{B, E, H\}^+$ = B, C, D, A, G, E, H, F which is also a key!

We can skip the second closure set (covered by the previous one) and move onto the third one $\{C\}^+$, which is missing B, D, E, F, H.

- Add B: $\{C, B\}^+$ = C, A, G, B, D
- Add D: $\{C, D\}^+$ = C, A, G, D
- Add E: $\{C, E\}^+$ = C, A, G, E
- Add F: $\{C, F\}^+$ = C, A, G, F
- Add H: $\{C, H\}^+$ = C, H, B, A, G, D
- Add B, E: $\{C, B, E\}^+$ = C, A, G, B, D, E
- Add B, F: $\{C, B, F\}^+$ = C, A, G, B, D, F, H
- Add D, E: $\{C, D, E\}^+$ = C, A, G, D, E
- Add D, F: $\{C, D, F\}^+$ = C, A, G, D, F, B, H
- Add D, H: $\{C, D, H\}^+$ = C, A, G, D, H, B

- Add E, F: $\{C, E, F\}^+ = C, A, G, E, F$
- Add E, H: $\{C, E, H\}^+ = C, E, H, F, A, G, B, H$ which means this is a key!
- Add F, H: $\{C, F, H\}^+ = C, H, B, A, G, D, F$
- Add B, E, F: $\{C, B, E, F\}^+$ which would be a superkey from (B,E,F), so we ignore this one.
- Add D, E, F: $\{C, D, E, F\}^+ = C, A, G, D, E, F$
- Add D, E, H: $\{C, D, E, H\}^+$ which would be a superkey from (C, E, H) so skip.

Since the fourth one was proved to be a key, we can skip this and goto the last one. Note that since we enumerated C, H and the various ones, we can also skip this as well.

Thus we found 3 keys, (B, E, F), (B, E, H), and (C, E, H).

(b) BCNF requires that we check all the functional dependencies.

- D is not a prime attribute and B is a part of a key so it violates 3NF and BCNF.
- B, F is a part of a key, but not a key itself, so it violates BCNF. Note this one does not violate 3NF as H is a prime attribute.
- A and G are not prime attributes and C is a part of a key. Thus it violates both 3NF and BCNF.
- (C, E, H) is okay since it is a key.
- C, H, is also part of a key while B is a prime attribute and thus violates BCNF but not 3NF.

(c) We will use the previous part (b) to decompose our relation.

- $B \to C, D$ is a violation and $\{B\}^+ = B, C, D, A, G$ so we obtain two relations R1(A, $\underline{B}$, C, D, G) and R2(B, E, F, H)
- Check the FDs against R1. $C \to A, G$ is a violation of BCNF form. Thus we split R1 into two further relations: R11($\underline{C}$, A, G) and R12 ($\underline{B}$, C, D).
- Check R11 against the FDs, note that they are okay, so R11 is BCNF.
- Check R12 against the FDs, and since they are also okay, R12 is also BCNF.
- Check R2 against the FDs and no FDs are bad. So done!

Thus, the resulting decomposition is: R11($\underline{C}$, A, G), R12 ($\underline{B}$, C, D), R2(B, E, F, H) with keys (B, E, F) and (B, E, H).

(d) We will use the previous part (b) to decompose our relation.

- $B \to C, D$ is a violation and $\{B\}^+ = B, C, D, A, G$ so we obtain two relations R1(A, $\underline{B}$, C, D, G) and R2(B, E, F, H)
- Check the FDs against R1. $C \to A, G$ is a violation of BCNF form. Thus we split R1 into two further relations: R11($\underline{C}$, A, G) and R12 ($\underline{B}$, C, D).
- Check R11 against the FDs, note that they are okay, so R11 is BCNF.
- Check R12 against the FDs, and since they are also okay, R12 is also BCNF.
- Check R2 against the FDs and note that $B, F \to H$ is a violation of BCNF. Decompose R2 into two further relations, R21($\underline{B}$, $\underline{F}$, H) and R22($\underline{B}$, $\underline{E}$, $\underline{F}$).
- Check R21 against the FDs, note that they are okay, so R21 is BCNF.
- Check R22 against the FDs, note that they are okay, so R22 is BCNF.

Thus, the resulting decomposition is: R11(C, A, G), R12 (B, C, D), R21(B, F, H) and R22(B, E, F).

(e) Note that our BCNF decomposition from above does not preserve all the functional dependencies. For the first three functional dependencies, there is a relation that includes all of the functional dependencies attributes, ensuring that the functional dependency is preserved. However, at least one functional dependency is not preserved: C, E, H $\rightarrow$ F. Note that we can construct a valid instance of the relations that when joined does not preserve the functional dependency. For example:

B	C	D
b	1	c
f	1	g

C	A	G
1	a	d

E	B
2	b
2	f

B	F	H
b	4	3
f	5	3

The natural join of the four tables yields the following:

A	B	C	D	E	F	G	H
a	b	1	c	2	4	d	3
a	f	1	g	2	5	d	3

Note that in the resulting table, the relation violates the functional dependency C, E, H $\rightarrow$ F and C, H $\rightarrow$ B.