Assignment 4

Student Name: Harshit Gupta **UID: 23BCS10228**
Branch: CSE **Section/Group: KRG 3-A**
Semester: 5th **Date of Performance:11/09/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

1. **Consider a relation R having attributes as R(ABCD), functional dependencies are given below:**

   AB->C, C->D, D->A

Identify the set of candidate keys possible in relation R. List all the set of prime and non -prime attributes.

I. Closure

We find the closure of potential candidate keys to see which can determine all attributes (A, B, C, D).

- (AB)+:
  - Start: AB
  - Using AB->C: ABC
  - Using C->D: ABCD
  - (AB)+ = ABCD
- (B)+:
  - Start: B. No FD has only B on the left side. (B)+ = B
- (C)+:
  - Start: C
  - Using C->D: CD
  - Using D->A: ACD
  - (C)+ = ACD (but missing B)
- (BC)+:
  - Start: BC
  - Using C->D: BCD
  - Using D->A: ABCD
  - (BC)+ = ABCD
- (BD)+:
  - Start: BD
  - Using D->A: ABD
  - Using AB->C: ABCD
  - (BD)+ = ABCD

II. Candidate Key(s)

From the closures, the minimal sets that can determine all attributes are: **AB, BC, and BD**.

III. Prime and Non-Prime Attributes

- **Prime Attributes:** Attributes that are part of any candidate key (A, B, C, D).
- **Non-Prime Attributes:** There are none. All attributes are prime.

IV. **Normal Form (NF) and Why?**

- **1NF:** Yes, as all attributes are atomic.
- **2NF:** Yes. There are no non-prime attributes, so partial dependencies cannot exist.
- **3NF:** Yes. Since all attributes are prime, no non-prime attribute is transitively dependent on a key (the definition of 3NF is satisfied).
- **BCNF: No.** The relation is **not in BCNF**. The definition of BCNF requires that for every non-trivial functional dependency X -> Y, X must be a superkey. We have the FD C -> D. C is not a superkey (as we saw, (C)+ = ACD, not ABCD). Similarly, D -> A violates BCNF as D is not a superkey.
- **Conclusion:** The highest normal form is **3NF**.

---

2. Relation R(ABCDE) having functional dependencies as :

A->D, B->A, BC->D, AC->BE

Identify the set of candidate keys possible in relation R. List all the set of prime and non-prime attributes.

I. **Closure &** II. **Candidate Key(s)**
Let's find a minimal superkey.

- (B)+:
  - Start: B
  - Using B->A: AB
  - Using A->D: ABD (Missing C and E)
- (C)+: C (No FDs for just C)
- (BC)+:
  - Start: BC
  - Using B->A: ABC
  - Using A->D: ABCD
  - Using BC->D: ABCD (same)
  - (BC)+ = ABCD (Missing E)

We need to find a key that includes E. Let's try (AC)+:

- Start: AC
- Using AC->BE: ACBE
- Using A->D: ACBED
- (AC)+ = ABCDE

Is AC minimal? Can we find a smaller key?

- (C)+ = C (fails)
- (A)+:
  - Start: A
  - Using A->D: AD (fails)
    So, AC is minimal.

Let's check if (BC)+ can be extended. We already have (BC)+ = ABCD. We need an FD to get E. There is no FD with just BC or its closure on the left that gives E. Therefore, BC is not a key.
Candidate Key: AC

III. Prime and Non-Prime Attributes

- **Prime Attributes:** Parts of the candidate key (A, C).
- **Non-Prime Attributes:** The remaining attributes (B, D, E).

IV. Normal Form (NF) and Why?

- **1NF:** Yes.
- **2NF: No.** There is a partial dependency. The FD A -> D is a problem. A is a subset of the candidate key AC, and it determines a non-prime attribute D. This is a partial dependency, which violates 2NF.
- **Conclusion:** The highest normal form is **1NF**.

---

3. Consider a relation R having attributes as R(ABCDE), functional dependencies are given below:

B->A, A->C, BC->D, AC->BE

Identify the set of candidate keys possible in relation R. List all the set of prime and non-prime attributes.

I. Closure & II. Candidate Key(s)

Let's find a minimal superkey.

- (B)+:
  - Start: B
  - Using B->A: AB
  - Using A->C: ABC
  - (B)+ = ABC (Missing D and E)

We need to get D and E. Let's try the obvious choice based on the FDs: (BC)+

- Start: BC
- Using B->A: ABC
- Using BC->D: ABCD
- Using A->C: ABCD (same)
- Using AC->BE: ABCD + BE = ABCDE
- (BC)+ = ABCDE

Is BC minimal?

- (B)+ = ABC (not all)
- (C)+ = C (not all)
  So, BC is minimal.

Let's check if B is a candidate key by itself? From above, (B)+ = ABC, so no.
Candidate Key: BC

III. **Prime and Non-Prime Attributes**

- **Prime Attributes:** Parts of the candidate key (B, C).
- **Non-Prime Attributes:** The remaining attributes (A, D, E).

IV. Normal Form (NF) and Why?

- **1NF:** Yes.
- **2NF: No.** There is a partial dependency. The FD B -> A is a problem. B is a subset of the candidate key BC, and it determines a non-prime attribute A. This violates 2NF.
- **Conclusion:** The highest normal form is **1NF**.

---

4. Consider a relation R having attributes as R(ABCDEF), functional dependencies are given below:

A->BCD, BC->DE, B->D, D->A

Identify the set of candidate keys possible in relation R. List all the set of prime and non prime attributes.

I. **Closure** & II. **Candidate Key(s)**
Let's find the closure of single attributes to find a key.

- (A)+:
    - Start: A
    - Using A->BCD: ABCD
    - Using B->D: ABCD (same)
    - Using BC->DE: ABCDE
    - Using D->A: ABCDE (A is already present)
    - (A)+ = ABCDE (Missing F)
- (B)+:
    - Start: B
    - Using B->D: BD
    - Using D->A: ABD
    - Using A->BCD: ABCD (same)
    - (B)+ = ABCD (Missing E and F)
- (D)+:
    - Start: D
    - Using D->A: AD
    - Using A->BCD: ABCD
    - (D)+ = ABCD (Missing E and F)

- (F)+: F

None of the single attributes are keys. Let's try A with F: (AF)+ = (A)+ + F = ABCDEF
Is AF minimal? Check (F)+ = F, so F must be part of any key. Check if A is necessary: (A)+ = ABCDE (missing F), so yes, A is needed.
Check (BF)+:

- Start: BF
- Using B->D: BDF
- Using D->A: ABDF
- Using A->BCD: ABCDF
- Using BC->DE: ABCDEF
  (BF)+ = ABCDEF. Is this minimal?
- (B)+ = ABCD (not all)
- (F)+ = F (not all)
  So, BF is a candidate key.

Check (DF)+:

- Start: DF
- Using D->A: ADF
- Using A->BCD: ABCDF
- Using BC->DE: ABCDEF
  (DF)+ = ABCDEF. It is also minimal.
  Candidate Keys: A, BF, DF (You could also find others like AF, but BF and DF are minimal).

III. Prime and Non-Prime Attributes

- **Prime Attributes:** Parts of any candidate key (A, B, D, F).
- **Non-Prime Attributes:** The remaining attributes (C, E).

IV. Normal Form (NF) and Why?

- **1NF:** Yes.
- **2NF:** Check for partial dependencies. The non-prime attributes are C and E.
  - C is determined by A (A->BCD). A is a candidate key itself, so this is a full functional dependency, not a partial one.
  - E is determined by BC (BC->DE). BC is not a superkey (is BC a candidate key? No, our candidate keys are A, BF, DF. BC is not a subset of any candidate key? B is prime, C is non-prime. This is not a partial dependency.
  - So, it seems to be in 2NF.
- **3NF:** Check for transitive dependencies. Is a non-prime attribute dependent on another non-prime?
  - Look at B->D. D is a prime attribute (it's part of candidate keys). A non-prime attribute (C or E) is not dependent on another non-prime. The dependencies are between prime attributes or from prime to non-prime.
  - So, it seems to be in 3NF.
- **BCNF: No.** Check if the left side of every FD is a superkey.
  - B->D: B is not a superkey. ((B)+ = ABCD, which is missing E and F). This violates BCNF.
  - D->A: D is not a superkey. ((D)+ = ABCD, missing E and F). This also violates BCNF.
- **Conclusion:** The highest normal form is **3NF**.

5. Designing a student database involves certain dependencies which are listed below:

- **X ->Y**
- **WZ ->X**
- **WZ ->Y**
- **Y ->W**
- **Y ->X**
- **Y ->Z**

I. **Closure &** II. **Candidate Key(s)**

Let's find a minimal superkey.

- (Y)+:
  - Start: Y
  - Using Y->W: WY
  - Using Y->X: WXY
  - Using Y->Z: WXYZ
  - (Y)+ = WXYZ

Is Y minimal? Yes, because no smaller set can work.

- (WZ)+:
  - Start: WZ
  - Using WZ->X: WXZ
  - Using WZ->Y: WXYZ
  - (WZ)+ = WXYZ
    WZ is also a candidate key.
    Candidate Keys: Y and WZ

III. **Prime and Non-Prime Attributes**

- **Prime Attributes:** Parts of the candidate keys (Y, W, Z).
- **Non-Prime Attributes:** There is only one other attribute: X. So, **X** is non-prime.

IV. **Normal Form (NF) and Why?**

- **1NF:** Yes (assumed).
- **2NF: Yes.** There is only one non-prime attribute (X). For the key Y, it's a single attribute key, so no partial dependency is possible. For the key WZ, the non-prime attribute X is dependent on the full key WZ (from WZ->X), so no partial dependency.
- **3NF: Yes.** The only non-prime attribute is X, and it is directly dependent on the candidate keys (Y->X and WZ->X). There is no transitive dependency.
- **BCNF: No.** Check all FDs. The FD X -> Y is a problem. X is a non-prime attribute, and it determines Y, which is a prime attribute. But for BCNF, the left side of *every* FD must be a superkey. X is not a superkey. This violates BCNF.
- **Conclusion:** The highest normal form is **3NF**.

6. Debix Pvt Ltd needs to maintain database having dependent attributes ABCDEF. These attributes are functionally dependent on each other for which functionally dependency set F given as:

{A -> BC, D -> E, BC -> D, A -> D} Consider a universal relation R1(A, B, C, D, E, F) with functional dependency set F, also all attributes are simple and take atomic values only. Find the highest normal form along with the candidate keys with prime and non-prime attribute.

I. **Closure** & II. **Candidate Key(s)**

Let's find a minimal superkey. Notice F is not on the right side of any FD, so it must be part of every candidate key.

- (A)+:
    - Start: A
    - Using A->BC: ABC
    - Using A->D: ABCD
    - Using BC->D: ABCD (same)
    - Using D->E: ABCDE
    - (A)+ = ABCDE (Missing F)

Therefore, A alone is not a key. Let's try (AF)+:

- (AF)+ = (A)+ + F = ABCDEF
  Is AF minimal? Check if A is necessary: (A)+ = ABCDE (missing F). Check if F is necessary: (F)+ = F. So both are needed.
  Is there a smaller key? Could A be replaced? No. So AF is a candidate key.
  We could also have other keys like (ABF)+, (ACF)+, etc., but they are not minimal since AF is sufficient.
  Candidate Key: AF

III. **Prime and Non-Prime Attributes**

- **Prime Attributes:** Parts of the candidate key (A, F).
- **Non-Prime Attributes:** The remaining attributes (B, C, D, E).

IV. **Normal Form (NF) and Why?**

- **1NF:** Yes (as stated in the problem).
- **2NF: No.** There are partial dependencies. The candidate key is AF. Look at the FD A->BC. A is a proper subset of the key, and it determines the non-prime attributes B and C. This is a partial dependency, which violates 2NF. Similarly, A->D is a partial dependency.
- **Conclusion:** The highest normal form is **1NF**.