EXPERIMENT - 4

Student Name: Umang Kaushik **UID: 23BCS10712**
Branch: BE-CSE **Section/Group: KRG_1-B**
Semester: 5th **Date of Performance: 09/09/2025**
Subject Name: ADBMS **Subject Code: 23CSP-333**

Question 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 and find highest normal form.

Solution: Candidate Key Derivation:

- Compute closures to find minimal keys: (AB)+ ={A,B,C,D}
  (BC)+ = {B, C, D, A}
  (BD)+ = {B, D, A, C}
  (A)+ = {A} → A does not give B or C directly.
  (C)+ = {C, D, A} (C→D, D→A) — missing B.
  (D)+ = {D, A, C} (D→A, A no new C except via AB) — missing B.
- Minimal sets whose closure is all attributes are AB, BC, BD.

Keys:

Candidate Keys = {AB, BC, BD}

Attributes:

Prime Attributes = {A, B, C, D} Non-Prime Attributes = {} (none)

Normalization:

BCNF:

• AB → C: AB is a candidate key → OK.
• C → D: C is not a superkey → violation.
• D → A: D is not a superkey → violation.
⇒ Not in BCNF.

3NF

• AB → C: LHS is key → OK.
• C → D: D is prime (every attribute is prime) → OK.
• D → A: A is prime → OK.
⇒ All FDs satisfy 3NF conditions.
Relation is in 3NF.

Highest Normal Form = 3NF

Question 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 and find highest normal form.

Solution: Candidate Key Derivation:

- Compute closures to find minimal keys:
  (A)+ = {A, D} (from A→D) — missing B, C, E.
  (B)+ = {B, A, D} (B→A, A→D) — missing C, E.
  (C)+ = {C} — gives nothing else alone.
  (AC)+ = {A, C, B, E, D} (AC→BE gives B, E; B→A already; A→D) = ABCDE. (BC)+ = {B, C, A, D, E} (B→A, AC→BE or BC→D then AC→BE) = ABCDE.
  (AB)+ = {A, B, D} (from B→A, A→D) — missing C, E.
- Minimal sets whose closure is all attributes are AC and BC.

Keys:

Candidate Keys = {AC, BC}

Attributes:

Prime Attributes = {A, B, C} Non-Prime Attributes = {D, E}

Normalization:

BCNF:

• A → D: A is not a key → violation.
• B → A: B is not a key → violation.
• BC → D: BC is a candidate key → OK.
• AC → BE: AC is a candidate key → OK.
⇒ Not in BCNF.

3NF: For each FD, check LHS is key or RHS attributes are prime:
• A → D: A not a key and D is non-prime → violation.
• B → A: B not a key but A is prime → OK.
BC → D: LHS is key → OK.
• AC → BE: LHS is key → OK.
⇒ Not in 3NF (because of A→D).

2NF: Check partial dependencies on part of any candidate key (non-prime depending on part of a key):

Candidate keys: AC and BC. Non-prime attributes are {D, E}.
A → D : A is a proper subset of the key AC and determines non-prime D → partial dependency → violation.
⇒ Not in 2NF.

1NF: Attributes are atomic → satisfies 1NF.

Highest Normal Form = 1NF

Question 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 and find highest normal form.

Solution: Candidate Key Derivation:

(A)+ = {A, C} (from A→C); from AC→BE get B,E; with B and C, BC→D gives D → so (A)+ = {A, B, C, D, E}.

(B)+ = {B, A} (from B→A); then A→C gives C; AC→BE gives E; BC→D gives D → so (B)+ = {A, B, C, D, E}.

(C)+ = {C}

(D)+ = {D}

(E)+ = {E}

Keys:

Candidate Keys = {A, B}

Attributes:

Prime Attributes = {A, B}
Non-Prime Attributes = {C, D, E}

Normalization:

BCNF:

• B → A : B is a candidate key → OK.
• A → C : A is a candidate key → OK.
• BC → D : BC contains B (a key), so BC is a superkey → OK.
• AC → BE : AC contains A (a key), so AC is a superkey → OK.
⇒ All FDs have superkey LHS → Relation is in BCNF.

- ## 3NF:

Since BCNF holds, 3NF is also satisfied.

- ## 2NF:

Candidate keys are single attributes, so there are no partial dependencies on a part of a composite key → satisfies 2NF.

- ## 1NF:

Attributes are atomic → satisfies 1NF.

Highest Normal Form = BCNF

Question 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 and find highest normal form.

Solution: Candidate Key Derivation:

- Attribute F never appears on the RHS of any dependency, so it must be included in every candidate key.
- Compute closures (with F included):
  • (AF)+: A→BCD gives {A, B, C, D}; with BC→DE we add E → {A, B, C, D, E}; including F → (AF)+ = {A, B, C, D, E, F}.
  • (BF)+: B→D, D→A, then A→BCD gives {A, B, C, D}; with BC→DE we get E → {A, B, C, D, E}; including F → (BF)+ = {A, B, C, D, E, F}.
  • (DF)+: D→A, then A→BCD gives {A, B, C, D}; with BC→DE we get E → {A, B, C, D, E}; including F → (DF)+ = {A, B, C, D, E, F}.
  • (CF)+ = {C, F} (C alone doesn't generate others) — not a key.
  • (EF)+ = {E, F} — not a key.
- Minimal keys are {AF}, {BF}, {DF}.

Keys:

Candidate Keys = {AF, BF, DF}
Attributes:

Prime Attributes = {A, B, D, F} Non-Prime Attributes = {C, E}

Normalization:

BCNF:

- A→BCD: A is not a superkey → violation.
- BC→DE: BC is not a superkey → violation.
- B→D: B is not a superkey → violation.
- D→A: D is not a superkey → violation.

⇒ Not in BCNF.

3NF:

For each FD, either LHS is a key or RHS is prime:

- A→BCD: A not a key, RHS contains non-prime C,E → violation.
- BC→DE: BC not a key, RHS contains non-prime E → violation.
- B→D: D is prime → OK.
- D→A: A is prime → OK.

⇒ Not in 3NF.

- ### 2NF:

  Candidate keys are {AF, BF, DF}. Non-prime attributes = {C, E}.

  - A→C: A is part of key AF and determines non-prime C → partial dependency → violation.

  ⇒ Not in 2NF.

- **1NF:** Attributes are atomic → satisfied.

Highest Normal Form = 1NF

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

- **X → Y**
- **WZ → X**
- **WZ → Y**
- **Y → W**
- **Y → X**
- **Y → Z**

The task here is to remove all the redundant FDs for efficient working of the student database management system.

Solution: We are given the relation R(W, X, Y, Z) with functional dependencies. Our aim is to find and remove the redundant dependencies

Write the FDs again -

1. X → Y
2. WZ → X
3. WZ → Y
4. Y → W
5. Y → X
6. Y → Z

Check redundancy one by one -

- Check FD (3): WZ → Y
  From (2) WZ → X and (1) X → Y, we can derive WZ → Y. So, FD (3) is redundant.

- Check FD (5): Y → X
  From (6) Y → Z and (4) Y → W, we already have (W,Z). Now, (W,Z) → X (from FD 2).
  Hence, from Y we can derive W and Z, then (WZ → X), so Y → X is also redundant.

Final minimal cover
The essential dependencies are:

1. X → Y
2. WZ → X
3. Y → W
4. Y → Z

After removing redundant dependencies, the minimal set of functional dependencies is:

- X → Y
- WZ → X
- Y → W
- Y → Z

This is the minimal cover of the given FDs, and hence these will be used for efficient working of the student database management system.

Question 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.

Solution: Candidate Key Derivation:

- Attribute F never appears on the RHS of any dependency, so it must be included in every candidate key.
  Compute closures (with F included):
- (AF)+ :
  A → BC → {A,B,C}

  BC → D → {A,B,C,D}

  D → E → {A,B,C,D,E}

  Add F → (AF)+ = {A,B,C,D,E,F}
- (BF)+ :
  Start with {B,F}. No FD gives A. Missing A → can't reach all attributes.
  ⇒ Not a key.
- (CF)+ :
  Start with {C,F}. No FD gives A. Missing A → not a key.
- (DF)+ :
  D → E → {D,E,F}. Still missing A,B,C → not a key.
- (EF)+ :
  Start with {E,F}. No FD gives A. Missing A,B,C,D → not a key. Thus the only minimal key = {AF}.

  Keys:

  Candidate Keys = {AF}

Attributes:

- Prime Attributes = {A, F}
- Non-Prime Attributes = {B, C, D, E}

Normalization:

BCNF:

A → BC : A not a superkey → violation. A → D : A not a superkey → violation. BC → D : BC not a superkey → violation. D → E : D not a superkey → violation.
⇒ Not in BCNF

3NF:

A → BC : A not a key, RHS has non-prime (B,C) → violation. A → D : A not a key, D non-prime → violation.
BC → D : BC not a key, D non-prime → violation. D → E : D not a key, E non-prime → violation.
⇒ Not in 3NF

2NF:

Candidate key = {AF}.
A → BC : A is part of candidate key and determines non-prime attributes → partial dependency → violation.
A → D : Same partial dependency → violation.
⇒ Not in 2NF

1NF:

All attributes are atomic → satisfied.

Highest Normal Form = 1NF