University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT:4

NAME : SATYA PRAKASH SWAIN **UID : 23BCS11072**
BRANCH : BE-CSE **SECTION : KRG_1A**
SEMESTER : 5TH **SUBJECT : 23CSP-339**
SUBJECT NAME : ADBMS

1. AIM:-

To find Prime attributes, Non-Prime attributes, and the Highest Normal Form for the given relations using functional dependencies.

PROBLEM 1:-

Consider a relation R having attributes as R(ABCD), functional dependencies are given below:
AB->C, C->D, D->A

PROBLEM 2:-

Relation R(ABCDE) having functional dependencies as :
A->D, B->A, BC->D, AC->BE

PROBLEM 3:-

Consider a relation R having attributes as R(ABCDE), functional dependencies are given below:
B->A, A->C, BC->D, AC->BE

PROBLEM 4:-

Consider a relation R having attributes as R(ABCDEF), functional dependencies are given below:
A->BCD, BC->DE, B->D, D->A

PROBLEM 5:-

Designing a student database involves certain dependencies which are listed below:
X ->Y, WZ ->X ,WZ ->Y ,Y ->W ,Y ->X ,Y ->Z

PROBLEM 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

2.SOLUTION:-

Problem 1:-

Step 1: Find Candidate Keys

- Start with AB:
  AB = {A, B, C, D} (using AB→C, C→D, D→A)
  Hence, AB is a Candidate Key.

Step 2: Prime and Non-Prime Attributes

- Prime Attributes: A, B
- Non-Prime Attributes: C, D

Step 3: Highest Normal Form

- 1NF: Satisfied (atomic attributes assumed)
- 2NF: Satisfied (AB is the only candidate key, no partial dependency exists)
- 3NF: Not satisfied
- Highest Normal Form: 2NF

Problem 2:-

Step 1: Find Candidate Keys

- BC = {A, B, C, D, E} ⇒ BC is a candidate key
- AC = {A, B, C, D, E} ⇒ AC is also a candidate key

Hence, Candidate Keys = {BC, AC}

Step 2: Prime and Non-Prime Attributes

- Prime Attributes: A, B, C
- Non-Prime Attributes: D, E

Step 3: Highest Normal Form

- 1NF: Satisfied
- 2NF: Satisfied (no partial dependency with respect to both keys)
- 3NF: Satisfied (A→D allowed since A is prime)
  Highest Normal Form = 3NF

Problem 3:-

Step 1: Find Candidate Keys

- A:
  - Start with A ⇒ {A}
  - A → C ⇒ {A, C}
  - AC → BE ⇒ {A, C, B, E}
  - BC → D (B, C present ⇒ add D) ⇒ {A, B, C, D, E}
    A covers all attributes ⇒ A is a candidate key
- B:
  - Start with B ⇒ {B}
  - B → A ⇒ {A, B}
  - A → C ⇒ {A, B, C}

- BC → D ⇒ {A, B, C, D}
- AC → BE ⇒ {A, B, C, D, E}
  B^+ covers all attributes ⇒ B is a candidate key

Hence, Candidate Keys = {A, B}

Step 2: Prime and Non-Prime Attributes

- Prime Attributes: A, B
- Non-Prime Attributes: C, D, E

Step 3: Highest Normal Form

- 1NF: Satisfied (attributes are atomic)
- 2NF: Satisfied (A and B are single-attribute keys, no partial dependencies)
- 3NF: Satisfied (for each FD, LHS is either a key or prime attribute ⇒ no violation)
- BCNF:
  - B → A (B is a key)
  - A → C (A is a key)
  - BC → D (B is a key ⇒ BC is a superkey)
  - AC → BE (A is a key ⇒ AC is a superkey)
    BCNF is satisfied

Problem 4:-

Step 1: Finding Candidate Keys

1. AF:
   - Start with {A, F}
   - A → BCD ⇒ {A, B, C, D, F}
   - BC → DE ⇒ {A, B, C, D, E, F}
   - AF is a Candidate Key
2. BF:
   - Start with {B, F}
   - B → D ⇒ {B, D, F}
   - D → A ⇒ {A, B, D, F}
   - A → BCD ⇒ {A, B, C, D, F}
   - BC → DE ⇒ {A, B, C, D, E, F}
   - BF is a Candidate Key
3. DF:
   - Start with {D, F}
   - D → A ⇒ {A, D, F}
   - A → BCD ⇒ {A, B, C, D, F}
   - BC → DE ⇒ {A, B, C, D, E, F}
   - DF is a Candidate Key

Candidate Keys = {AF, BF, DF}

Step 2: Prime and Non-Prime Attributes

- Prime Attributes (appear in at least one Candidate Key):
  A, B, D, F

- Non-Prime Attributes (do not appear in any Candidate Key):
  C, E

Step 3: Highest Normal Form
- Candidate Keys are AF, BF, DF (all composite).
- Check Partial Dependencies:
  - A → BCD → A is a proper subset of AF.
  - RHS contains C (non-prime) → Partial Dependency exists
  - Violates 2NF

Hence, Highest Normal Form = 1NF

Problem 5:-

Step 1: Finding Candidate Keys
1. Y:
   - Start: {Y}
   - Y → W ⇒ {Y, W}
   - Y → X ⇒ {Y, W, X}
   - Y → Z ⇒ {Y, W, X, Z}
   - Y = {W, X, Y, Z} ⇒ Y is a Candidate Key
2. X:
   - Start: {X}
   - X → Y ⇒ {X, Y}
   - Y → W ⇒ {X, Y, W}
   - Y → Z ⇒ {X, Y, W, Z}
   - X= {W, X, Y, Z} ⇒ X is a Candidate Key
3. WZ:
   - Start: {W, Z}
   - WZ → X ⇒ {W, Z, X}
   - X → Y ⇒ {W, Z, X, Y}
   - Y → W, Z (already present) ⇒ closure complete
   - WZ = {W, X, Y, Z} ⇒ WZ is a Candidate Key

Candidate Keys = {Y, X, WZ}

Step 2: Prime and Non-Prime Attributes
- Prime Attributes: W, X, Y, Z
- Non-Prime Attributes: None

Step 3: Highest Normal Form
- All candidate keys are superkeys.
- In every functional dependency, the LHS is a superkey.
- No partial or transitive dependencies on non-prime attributes exist.

Highest Normal Form = BCNF

Problem 6:-

Step 1: Candidate Key

- By inspection, AF covers all attributes ⇒ AF is taken as the Candidate Key

Step 2: Prime and Non-Prime Attributes

- Prime Attributes: A, F
- Non-Prime Attributes: B, C, D, E

Step 3: Highest Normal Form

- 1NF: Satisfied (attributes are atomic)
- 2NF: Violated
  - Candidate Key = AF
  - $A \rightarrow BC$, $A \rightarrow D \Rightarrow$ Partial Dependency exists
- 3NF & BCNF: Not applicable as 2NF is violated

Highest Normal Form = 1NF

5.LEARNING OUTCOMES:-

1.Learn how to identify **candidate keys** for a given relation using functional dependencies.
2.Understand the difference between **prime and non-prime attributes**.
3.Analyze **functional dependencies** to determine **highest normal form** of a relation.
4.Identify **partial and transitive dependencies** in a relation.
5.Gain knowledge of how **functional dependencies affect table normalization** (1NF, 2NF, 3NF, BCNF).
6.Develop the ability to **decompose relations** based on functional dependencies to eliminate redundancy.
7.Understand the role of functional dependencies in designing **normalized relational tables**.
8.Learn to **apply closure of attribute sets** to find candidate keys and verify superkeys.