# Experiment – 4

| **Student Name:** Yash Sharma | **UID:** 23BCS14110 |
|---|---|
| **Branch:** BE-CSE | **Section/Group:** KRG-1 (B) |
| **Semester:** $5^{th}$ | **Date:** 09-09-25 |
| **Subject Name:** ADBMS | **Subject Code:** 23CSP-333 |

## Aim:

**Q1. 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 sets of prime and non-prime attributes.

**Q2. 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 sets of prime and non-prime attributes.

**Q3. 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 sets of prime and non-prime attributes.

**Q4. 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 sets of prime and non-prime attributes.

**Q5. 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.

**Q6. Debix Pvt Ltd needs to maintain a database with dependent attributes ABCDEF. These attributes are functionally dependent on each other, for which the functional dependency set F is 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 attributes.

## Objective:

**Q1:**
To analyse functional dependencies of relation R(ABCD) and determine candidate keys, along with the classification of prime and non-prime attributes.

**Q2:**
To evaluate the given FDs in relation R(ABCDE) and identify all possible candidate keys, prime, and non-prime attributes.

**Q3:**
To apply the closure method on functional dependencies of R(ABCDE) for finding candidate keys and distinguishing prime from non-prime attributes.

**Q4:**
To determine candidate keys of R(ABCDEF) by analysing given dependencies and classify attributes as prime or non-prime.

**Q5:**
To minimize the functional dependency set by eliminating redundant FDs for efficient design of the student database system.

**Q6:**
To identify the candidate keys, prime/non-prime attributes, and the highest normal form of relation R1(ABCDEF) using the given FD set.

## Answer:

### Q1:

**Relation:** R(A, B, C, D)
**FDs:** AB → C, C → D, D → A
**Closures / reasoning (brief):**

- $AB^+$ = {A, B} → C (from AB → C) → D (from C → D) → A (from D → A). So, $AB^+$ = {A, B, C, D} ⇒ **AB is a key**.
- $C^+$ = {C} → D → A ⇒ {A, C, D} (missing B) → not a key.
- $BC^+$ = {B, C} → D (C→D) → A (D→A) ⇒ {A, B, C, D} ⇒ **BC is a key**.
- $BD^+$ = {B, D} → A (D→A) and then AB→C ⇒ {A, B, C, D} ⇒ **BD is a key**.
- No single attribute alone gives all attributes.

**Candidate keys:** {AB, BC, BD}
**Prime attributes:** attributes that appear in any candidate key = {A, B, C, D} (all)
**Non-prime attributes:** ∅

### Q2:

**Relation:** R(A, B, C, D, E)
**FDs:** A → D, B → A, BC → D, AC → B E
**Closures / reasoning (brief):**

- $AC^+$: AC → B,E (given). With B we get A (already) and A → D gives D. So $AC^+$ = {A,B,C,D,E} ⇒ **AC is a key**.
- $BC^+$: BC → D (given). B → A gives A, then AC → B,E gives E (and B). So $BC^+$ = {A,B,C,D,E} ⇒ **BC is a key**.
- Check minimality: A, B, C individually are not keys; AC and BC are minimal.

**Candidate keys:** {AC, BC}
**Prime attributes:** {A, B, C}
**Non-prime attributes:** {D, E}

### Q3:

**Relation:** R(A, B, C, D, E)
**FDs:** B → A, A → C, BC → D, AC → B E
**Closures / reasoning (brief):**

- $B^+$: B → A → C; with A,C we get AC → B,E → gives E; BC → D (with B,C) gives D. So $B^+$ = {A,B,C,D,E} ⇒ **B is a key**.
- $A^+$: A → C; AC → B,E gives B and E; BC → D gives D. So $A^+$ = {A,B,C,D,E} ⇒ **A is a key**.

**Candidate keys:** {A, B} (both are single-attribute keys)

**Prime attributes:** {A, B}
**Non-prime attributes:** {C, D, E}

**Q4:**

**Relation:** R(A, B, C, D, E, F)
**FDs:** A → B C D, BC → D E, B → D, D → A
**Closures / reasoning (brief):**

- $A^+$: A → B,C,D. From BC → D,E (we have B,C) get E. So $A^+$ = {A,B,C,D,E} (missing F).
- $B^+$: B → D → A → B,C,D and then BC→E gives E ⇒ $B^+$ = {A,B,C,D,E} (missing F).
- $D^+$: D → A → B,C,D and BC→E gives E ⇒ $D^+$ = {A,B,C,D,E} (missing F).
  Thus any of A, B, or D together with F will give all attributes.
- $AF^+$: A gives {A,B,C,D,E} + F ⇒ all ⇒ **AF is a key**.
- $BF^+$: B gives {A,B,C,D,E} + F ⇒ all ⇒ **BF is a key**.
- $DF^+$: D gives {A,B,C,D,E} + F ⇒ all ⇒ **DF is a key**.

No smaller combination without F is a key.
**Candidate keys:** {AF, BF, DF}
**Prime attributes:** {A, B, D, F}
**Non-prime attributes:** {C, E}

**Q5:**

**Given FDs:**
X → Y
WZ → X
WZ → Y
Y → W
Y → X
Y → Z
**Goal:** remove redundant FDs (find a minimal cover).
**Step 1 — RHS already singletons.**
**Step 2 — test redundancy / implication (brief):**

- From Y → W and Y → Z we get Y → WZ. With WZ → X, Y → X follows. So Y → X is implied by Y→W, Y→Z, WZ→X ⇒ **Y → X is redundant**.
- From WZ → X and X → Y we get WZ → Y. So WZ → Y is implied by WZ→X and X→Y ⇒ **WZ → Y is redundant**.
- After removing those, remaining FDs are necessary (none is derivable from the others).

**Minimal (non-redundant) cover:**
X -> Y
WZ -> X
Y -> W
Y -> Z
(Optionally combine last two as Y -> WZ.)
**Final answer:** The redundant FDs are removed; the minimal cover is shown above.

**Q6:**

**Relation:** R1(A, B, C, D, E, F)
**FDs (F):** A → B C, D → E, BC → D, A → D
**Assumptions:** All attributes atomic.
**Step 1 — candidate key(s):**

- $A^+$: A → B,C and A → D (given). From BC → D we already have D; D → E gives E. So $A^+$ = {A, B, C, D, E} (missing F). A alone does not reach F.
- $AF^+$: A gives B,C,D,E and plus F gives all attributes ⇒ $AF^+$ = {A,B,C,D,E,F} ⇒ **AF is a key**.

No FD derives A from other attributes, so every key must include A. F is not derivable, so AF is minimal. Therefore **AF is the only candidate key**.
**Prime attributes:** attributes that appear in any candidate key = {A, F}
**Non-prime attributes:** {B, C, D, E}
**Step 2 — highest normal form:**

- Relation is in **1NF** (attributes atomic).
- Candidate key is composite (AF). There are FDs with a proper subset of the key on the LHS:
  - A → B C and A → D are dependencies from **A**, which is a proper subset of the key AF, to non-prime attributes (B, C, D, E). These are **partial dependencies** on part of a candidate key ⇒ **violates 2NF**.
- Therefore the highest normal form is **1NF**.