Experiment 1.4

Student Name: Divyansh Manhas
UID: 23BAI70239
Branch: BE-AIT-CSE
Section/Group: 23AML_KRG-1
Semester: 5^{th}
Date of Performance: 10 Sep 2025
Subject Name: ADBMS
Subject Code: 23CSP-333

1. **Problem Title:** Functional Dependency and Keys.

2. **Problem Description:** In relational databases, keys are defined using functional dependencies (FDs). A super key is any set of one or more attributes that can uniquely identify a tuple in a relation; formally, if a set of attributes X functionally determines all attributes of the relation (X → R), then X is a super key. Among super keys, the candidate keys are those that are minimal, meaning no proper subset of them can still uniquely determine all attributes of the relation. From the set of candidate keys, one is chosen as the primary key, which serves as the main identifier for tuples in the relation. When a key is made up of two or more attributes, it is called a composite key, and this occurs when the combination of attributes together functionally determines all other attributes of the relation, but no single attribute in that set can do so individually. Thus, super keys guarantee uniqueness, candidate keys are the minimal super keys, the primary key is the selected candidate key, and composite keys are keys formed by combining multiple attributes.

3. **Questions:**

   a. R(ABCD): -
      i. **AB->C**
      ii. **C->D**
      iii. **D->A**

AB+ = ABCD
BC+ = CBDA
DB+ = DBAC

Candidate keys = {AB, BC, DB} PA = {A, B, C, D} NPA = {}
Normalisation: 3NF because (X is a super key or candidate key OR Y is a prime attribute) (If all attributes come out to be prime – R is in 3NF).

b. R(ABCDE): -
   i. A->D
   ii. B->A
   iii. BC->D
   iv. AC->BE.

   Given: R (A, B, C, D, E); ***FD:*** A->D, B->A, BC->D, AC->BE.

   CA+ = DCABE
   CB+ = CBADE

   Candidate keys = {CA, CB} PA= {A, C, B} NPA= {D, E}
   Normalisation: 1NF

c. R(ABCDE)
   i. B->A
   ii. A->C
   iii. BC->D
   iv. AC->BE

   Given FD: B->A, A->C, BC->D, AC->BE.

   A+ = DCABE
   B+ = CBADE

   Candidate keys = {A, B} PA= {A, B} NPA= {C, D, E}
   Normalisation: BCNF or 3.5NF because (All the FD in X is super key or candidate key).

d. R(ABCDEF)
   i. A->BCD
   ii. BC->DE
   iii. B->D
   iv. D->A.

   Given: R (A, B, C, D, E, F);

   FB+ = BDACEF
   FD+ = DABCEF

   Candidate keys= {FB, FD} PA= {A, D, F, B} NPA= {C, E}
   Normalisation: 1NF because (X is a subset of candidate key AND Y is non-prime attribute) – then R is not in 2NF.

e. R(W, X, Y, Z): -
   i. X ->Y

ii. WZ ->X
iii. WZ ->Y
iv. Y ->W
v. Y ->X
vi. Y ->Z

Given: R (W, X, Y, Z); ***FD:*** X ->Y, WZ ->X, WZ ->Y, Y ->W,Y ->X, Y ->Z.

Since X -> Y so we can write WZ ->Y.

FD': X ->Y, WZ ->Y, Y ->W, Y ->X, Y ->Z.
X+ = YWXZ
Y+ = WXZY
WZ+ = WZYX

Candidate keys = {X, Y, WZ} PA= {X, Y, WZ} NPA= {}
Normalisation: BCNF because (All the FD in X is super key or candidate key).

f. R1(A, B, C, D, E, F)
   i. **A -> BC**
   ii. **D -> E**
   iii. **BC -> D**
   iv. **A -> D**

Given: R (A, B, C, D, E, F); ***FD:*** A -> BC, D -> E, BC -> D, A -> D**.**
AF+ = AFDBCE

Candidate keys = {AF} PA= {A, F} NPA= {B, C, D, E}
Normalisation: 1NF because (X is a subset of candidate key AND Y is non-prime attribute) – then R is not in 2NF.