2 Schema Normalization

2.1 Identify the candidate key(s) for R1 & R2 and explain briefly why.

R1(A, B, C, D). Functional dependencies (FDs):

1. AB → C
2. AB → D
3. C → A
4. D → B

The superkeys for R1 are:

- ABCD (all attributes)
- BCD (since C → A for ABCD)
- ACD (since D → B for ABCD)
- ABD (since AB → C for ABCD)
- ABC (since AB → D for ABCD)
- AB (since AB → C for ABC *or* AB → D for ABD (both superkeys listed above))
- CD (since C → A for ACD *or* D → B for BCD (both superkeys listed above))

All but the last two superkeys can be reduced into AB and CD, so these are R1's candidate keys – they determine R1's other attributes uniquely.

R2(A, B, C, D, E). FDs:

1. A → BD
2. B → D
3. AB → C
4. E → A

The superkeys for R2 are:

- ABCDE (all attributes)
- ACE (since A → BD for ABCDE)
- ABCE (since B → D for ABCDE)
- ABDE (since AB → C for ABCDE)
- BCDE (since E → A for ABCDE)
- CE (since E → A for ACE)
- ABE (since AB → C for ABCE)
- AE (since A → BD for ABDE)
- E (since E → A for AE)
- ...

Only E is a candidate key for R2 (i.e., only it determines R2's other attributes uniquely but can not be reduced further).

Since E → A → BD and AB → C, E is sufficient for all permutations of the five attributes. It is also necessary, since it is not uniquely determined by any other attribute(s) (it is not on the right-hand side of any FD).

2.2 Derive a minimal cover for R1 and R2 with their FDs.

The minimal cover for R1 is:

- C → A (FD 3)
- D → B (FD 4)

(CD would be the key.)

The minimal cover for R2 is:

- E → A (FD 4)
- A → B (FD 2 with reduced right-hand side)
- A → D (FD 2 with reduced right-hand side)
- A → C (FD 3 with reduced left-hand side)

2.3 Identify whether the relations satisfy BCNF and whether they satisfy 3NF. Explain briefly why (or why not).

A relation is in BCNF if for every X → Y in the set of implied FDs:

- Y in X; or
- X is a superkey.

BCNF implies 3NF, but the relationship is also in 3NF if Y is part of one of its keys.

R1

AB is a superkey, so the first two FDs meet the BCNF requirements.

The story about the last two FDs is similar. C and D lead to A and B respectively. C and D are not superkeys and A and B are not "in" the former, so R1 is not in BCNF. A and B are both part of *one of* R1's candidate keys (namely AB), so the relation is in 3NF.

R2

R2 is not in BCNF, since the left-hand side of FD1 (A → BD) is not a superkey and does not "contain" the right-hand side.

It is also not in 3NF, since BD is not part of a key (i.e. *the* key, E) either.

2.4 If a relation is not in BCNF, derive a lossless join decomposition into BCNF and explain whether they preserve the dependencies (if they do not, indicate which FDs are not preserved).

R1

AB is already superkey, so the first two FDs do not violate BCNF requirements.

C → A does, since C is not a superkey. Create relations R1.1(ABD) and R1.2(AC).

Now in R1.1 D → B violates, so decompose again for R1.1(AB) R1.2(AC) R1.3(BD).

Now all FD left-hand sides are superkeys of the relations they apply to – the decomposition into BCFN is complete without loss of information.

However, this does not preserve dependencies. E.g. FD1, AB → C, can no longer be checked without joining R1.1 and R1.2.

R2

If we decompose into R2.1(AE) and R2.2(BCD), at least R2.1 (on which only E → A applies) does not violate BCNF.

However, the left-hand side of B → D is not a superkey of R2.2, so the next decomposition is R2.1(AE) R2.2(BD) R2.3(ABC).

In R2.3, only one FD applies; the left-hand side of AB → C is a superkey, so the lossless decomposition is complete.

This decomposition does not preserve dependencies either. E.g. A → BD can not be checked without joining R2.1 and R2.2.