**1. Canonical Cover Algorithm (11 pts).**

Given R(ABCD),

A. Find a canonical cover for the following set F of functional dependencies.

A🡪BC

AB🡪D

C🡪AD

D🡪B

B is extraneous in AB🡪D because A🡪B (given)

Combining A🡪BC and A🡪D, we get **{A🡪BCD, C🡪AD, D🡪B}**

B is extraneous in A🡪BCD because D🡪B (given)

D is extraneous in A🡪CD because C🡪D (given)

**Final Ans: A**🡪**C**

**C**🡪**AD**

**D**🡪**B**

B. Find all candidate keys.

**Ans: A and C**

**2. Dependency Preservation (15 pts).**

For the universal relation R(w,x,y,z), consider the decomposition D consisting of R1(w,y,z) and R2(x,y), and the set F of functional dependencies { y🡪xz ; yz🡪w ; x🡪w }.

Recall that the projection of set of functional dependences G on relation Rx consists of every functional dependency in (G)+ that contains only attributes from Rx.

**a.** Compute the projection of F on R1.

**Ans: {y->z, yz->w}**

**b.** Compute the projection of F on R2.

**Ans: {y->x}**

**c.** Does the decomposition D preserve the set of dependencies F? Why or why not?

**Ans: No, x**🡪**w is not preserved (cannot be derived from union of {y**🡪**z, yz**🡪**w} and {y**🡪**x}**

**3. Nonadditive (Lossless) Join Property (24 pts).**

R = {ABCDEFGHIJ}

F = { AB -> C

A -> DE

B -> F

F -> GH

D -> IJ

}

Determine whether each decomposition into R1 and R2 has the lossless join property. Be sure to state why or why not.

**a.** R1(ABC) and R2(ABDEFGHIJ)

**Ans: Yes, intersection of R1 and R2 is AB, and AB is a superkey in R1.**

**b.** R1(ADE) and R2(EBCFGHIJ)

**Ans: No, R1 intersect R2 is E, and E is not a superkey in either R1 or R2**

**c.** R1(BFDIJ) and R2(ABCDEFGH)

**Ans: Yes, R1 intersect R2 is BFD. BFD is a superkey in R1.**

**d.** R1(BDFGHIJ) and R2(ABCDE)

**Ans: Yes, R1 intersect R2 is BD, and BD is a superkey in R1**

**4. Normalization and Decomposition (20 pts).**

R(ABCDEG)

FDs {AB=>C, AC=>B, AD=>E, B=>D, BC=>A, E=>G}

**a.** Identify candidate keys of R. Hint: There are three candidate keys.

**Ans: AB, AC, BC**

**b.** Construct a decomposition of R into relations in 2NF.

R1(ABCEG) R2(BD), with R1(B) as a foreign key referencing R2.

Note that this decomposition does not preserve AD->E (but you did not have to worry about dependency preservation issues in that question)

**c.** Construct a decomposition of R into relations in 3NF.

You can continue decomposition by splitting (E,G) into a separate table as a transitive functional dependency, which will give you 3NF.

R1(A, B, C, E)

R2(B, D)

R3(E, G)

With R1(B) as a foreign key referencing R2

With R1(E) as a foreign key referencing R3

It should be easy to verify that it is in 3NF. In this particular case, we did not achieve dependency preservation (AD->E has been lost), but in this particular question achieving dependency preservation is more difficult than what we discussed in class.

**d.** Are all of the relations in your decomposition in BCNF? Either explain why they are, or identify one that is not and explain why it is not. (Note that for a relation to be in BCNF, the determinants of all functional dependencies in the relation must be superkeys *of that relation* – not superkeys of the original universal relation.)

**Yes, because for all preserved FDs (everything but AD=>E), left hand side is a superkey in its respective relation.**