DSA5104 Assm1 A0216941E

Date No.

1.1. $F = \{ J \to K, KX \to Z, ZY \to J \}$, $R = JKXYZ$, since there is no FD in set of F that determines XY, XY must be part of candidate keys. Test diff combinations next.

try: $[XYJ]^+ = XYJK$ ($\because J \to K$)
$XYJKZ$ ($\because KX \to Z$)
$\because [XYJ]^+ = R$, XYJ is candidate key.

try: $[XYK]^+ = XYKZ$ ($\because KX \to Z$)
$XYKZJ$ ($\because ZY \to J$)
$\because [XYK]^+ = R$, XYK is candidate key.

try: $[XYZ]^+ = XYZJ$ ($\because ZY \to J$)
$XYZJK$ ($\because J \to K$)
$\because [XYZ]^+ = R$, XYZ is candidate key.

$\therefore$ All candidate keys for R: XYJ, XYK, XYZ

1.2.1. R = ABCDEGH, X = ACEH. Find F_X.
the only FDs in F that start w/ attributes in X are, $\{AC \to B, E \to G\}$

① $AC \to B$: ACB
ACBD ($\because B \to D$)
ACBDE ($\because AD \to E$)
ACBDEG ($\because E \to G$, but G not in X)
$\therefore AC \to E$ is in F_X
② $E \to G$: EG (already terminal step).
③ no FDs containing H in X

$\therefore F_X = \{AC \to E\}$, $F_X^+ = \{AC \to E\}$ ($\because$ no other valid FDs)

1.2.2. $X = ACEH$, from 1.2.1, $F_X = F_X^+ = \{AC \to E\}$

for X to be in BCNF, ~~AC~~ ∵ $AC \to E$ is the only FD, then AC must be a superkey.

However, AC is NOT a ~~super~~ superkey, ∴ X is NOT in BCNF.

Decompose into BCNF: ① use the FD: $AC \to E$ as 1 sub relation. ⇒ $R_1 = ACE$

② remaining H ⇒ $R_2 = ACH$

- $R_1 = ACE$ ~~has~~ has $F_{R_1} = \{AC \to E\}$ where AC is superkey, ∴ ACE is in BCNF
- $R_2 = ACH$ has no non-trivial FD, ∴ ACH is in BCNF

∴ X is decomposed into ACE & ACH.

1.3.1. For R decomposed into [illegible] $R_1 = \{AB, BC, ABDE, EG\}$, consider the first FD in F: $AB \to C$, ∵ there are no decomposed relations in R_1 that contain attributes A, B and C, the first FD in F ~~is~~ is ~~not~~ NOT preserved after decomposition.

Since 1 FD in F is already not preserved, we can already consider this decomposition to ~~be~~ NOT be dependency-preserving.

1.3.2. We can check by attempting to reconstruct R by natural ~~joining~~ joining different combinations of decomposed relations, ensuring each ~~is~~ natural join is lossless thru the process.

We can check for lossless join by checking for

① common attribute(s) ② If those common attribute(s) is/are ~~can~~ candidate key for either one ~~sub relation~~ relation or both during natural join.

Taking $ABDE$ & EG, ① common attribute: E ② $E^+ = EG$ ~~..~~ ∴ E is candidate key of EG, Therefore $ABDEG$ decomp. into $ABDE$ & EG is ~~not~~ lossless	To get R from $ABDEG$, we are left with C, try: ~~natural~~ $ABDEG$ natural join BC. ① common attribute: B, ② $B^+ = BD$, ∴ B is NOT candidate key of $ABDEG$ & BC. Therefore, decomp. of R into $ABDEG$ & BC will ~~not~~ ~~be loss~~ NOT be lossless.

Following which, ∵ B is ~~not~~ also NOT candidate key for AB, as such we cannot ~~jo~~ natural join AB & BC w/o duplicates.
AB & BC both have no common attributes with EG.

As we have exhausted all possible options, we conclude that the decomposition of R is <u>NOT</u> lossless.

2.1. /Courses/Course[@CID="1234"]/Students/Student.

2.2.
```
for $c in /Courses/Course
  where $c/@CID = "1234"
  return $c/Students/Student
```

2.3. The query is NOT correct. The logic is good but it fails to "break" the inner loops when the where clause is activated. As such, for courses with 3 or more unique names, the same course will be returned multiple times, resulting in issue of duplicates.

2.4.
```
for $c in /Courses/Course
return <CourseCount>
          <CID> {$c/@CID} </CID>
          <TotalStudents> {fn:count($c/Students/Student)} </TotalStudents>
       </CourseCount>
```