COMP 3005: Database Management Systems – Assignment 3 – Andy Chia – 101111058

|  |  |
| --- | --- |
| 1. | *Consider the following proposed rule: If A → B and C → B, then A → C. Prove that this rule is not sound.*  This rule is not sound because it does not follow any of Armstrong’s Axioms to prove its validity. It might try to simulate the Transitivity rule but because “C → B” and not “B → C”, the proposed rule is not sound. |

|  |  |
| --- | --- |
| 2. | *Consider the following relation R = {A, B, C, D, E} and the following set of functional dependencies*  *F = {*  *A → BC*  *CD → E*  *B → D*  *E → A}*  *Compute B+. Is R in BCNF? If not, give a lossless decomposition of R into BCNF. Show your work for all previous questions.*  B­+ = BD \*Not a superkey  R is not in BCNF  (ABCDE)  (ABC) (ADE) \**A → BC*  (ABC) (BD) (ADE)  (ABC) (CDE) (BD) (ADE)  (ABC) (CDE) (BD) (ADE) (AE)  (ABC) (CDE) (BD) (AE) ----------- \*(ADE) can be removed because it can be reobtained |

|  |  |
| --- | --- |
| 3. | *Give a lossless, dependency-preserving decomposition into 3NF of schema R in Q2.*  A → BC  CD → E  B → D  E → A  With this we have: (ABC) (CDE) (BD) (AE)  There is no Extraenous attribute to remove  (ABC) (CDE) (BD) (EA) |

|  |  |
| --- | --- |
| 4. | *Assume the following decomposition of R in Q2: R1(A, B, C) and R2(C, D, E). Is this decomposition lossy or lossless? Why?*  This is lossless decomposition because either and hold. |

|  |  |  |
| --- | --- | --- |
| 5. |  | *Consider the following relation R(A, B, C, D, E, G) and the set of functional dependencies*  *F = {*  *A → BCD*  *BC → DE*  *B → D*  *D → A}* |
|  | (a) | *Compute B+:*   |  |  | | --- | --- | | B → D | (B, D) | | D → A | (A, B, D) | | A → BCD | (A, B, C, D) | | BC → DE | (A, B, C, D, E) | |
|  | (b) | *Prove (using Armstrong’s axioms) that AG is superkey.*  (AG)+  Result = AG   |  |  | | --- | --- | | A → BCD | (A, B, C, D, G) | | BC → DE | (A, B, C, D, E, G) |   The other functions are redundant in this case  (AG)+ = ABCDEG  AG is a superkey |
|  | (c) | *Compute Fc.*  *F = {*  *A → BCD*  *BC → DE*  *B → D*  *D → A}*  Nothing to combine since they are all unique.  D is extraneous in A → BCD because of B → D.  Fc = {A→BC, BC→DE, B→D; D→A}  D is extraneous in BC → DE Because of Armstrong’s Decomposition rule  BC → D \*this makes D in B → D and we can remove it  BC → **E**  Fc = {A→BC, BC→DE, D→A}  C is extraneous in BC→DE  Fc = {A→BC, B→DE, D→A} |
|  | (d) | *Give a 3NF decomposition of the given schema based on a canonical cover.*  (ABC) (BDE) (AD) (AG) |
|  | (e) | *Give a BCNF decomposition of the given schema based on F. Use the first functional dependency as the violator of the BCNF condition.*  (ABCDEG)  (ABC) (BCDEG)  (ABC) (BDEG) (AD) \*R2 links R0 with R1 dropping C in R1  (ABC) (BDE) (AD) (AG) |

|  |  |
| --- | --- |
| 6. | *Given the following set of functional dependencies:*  *A → BC*  *B → AC*  *C → AB*  *Show that it is possible to find more than one unique canonical cover for this set.*  A → B  A → C  B → A  B → C  C → A  C → B  Using Transitivity, we can remove some of the functions: A→B and B→A  A → C  B → C  C → AB |

|  |  |
| --- | --- |
| 7. | *Consider the schema R = (A, B, C, D, E, G) and the set F of functional dependencies:*  *A → BC*  *BD → E*  *CD → AB*  *Use the BCNF decomposition algorithm to find a BCNF decomposition of R. Start with A → BC. Explain your steps. Is this decomposition lossy or lossless? Is it dependency-preserving?*  BCNF:  (ABCDEG)  (ABC) (ABCDEG) \*R0 is moved out (A → BC)  (ABC) (BDE) (ABCDG) \*R1 is moved out (BD → E)  (ABC) (BDE) (ABCD) (ADG) \*R4 is created so that it can link G with any of the other functions  (ABC) (BDE) (AC) (AD) (ADGE) \*R21 and R22 split from R2 because B is recoverable with R1 & R2  This decomposition is Lossless  It is not dependency preserving: CD → AB |