**COMP 353 : Assignment 2**

**Presented to :**

Professor Khaled Jababo

**Presented by :**

Jordan Hubscher, ID : 27019696

Brandon Mendez, ID : 27003606

Anthony Mendez, ID : 26998380

**1)** Prove or disprove the following statements. To prove a statement is true, give a formal argument (in cases involving implications among FD’s, use Armstrong’s Axioms). To disprove a statement, give a counterexample:

**A)** {A B, C AB} {C B}

Augmentation: A B gives AB BB. This can be reduced to ABB.

Pseudo-transitivity: Since ABB and C B

Then this is **True**.

**B)** {KL M, L N} {KN M}

**K L M N**

**R1** 1 0 0 1

**R2**  1 1 1 1

The above table does not satisfy the FD's therefore the answer is **False.**

**C)** {A C, BD A, C D} {AB CD}

Augmentation: A C gives AB CB. This can be reduced to AB C. Because we have AB C we also have AB D since C D.

Union: gives AB CD

Then this is **True**.

**D)** A relation scheme with only three attributes is necessarily at least in 3NF.

This statement is **False.**

If:

R = {A,B,C }

F = { A B, B C }

Taking the second FD into consideration

1- BC is not trivial

2- B is not a superkey

3- C is not part of any of the candidate keys either

Therefore with these three conditions, it is not in 3NF.

**2)** Suppose you are given a relation scheme R = {A, B, C, D}. For each of the following sets of functional dependencies, assuming those are the only dependencies that hold for R, do the following:

i. Identify the candidate key(s) for R.

ii. State whether or not the proposed decomposition of R into smaller relations is a good decomposition, briefly explaining why or why not.

1. {BC, DA}: decompose into BC and AD

i) The candidate key is BD.

ii) This is a bad decomposition because it is lossy. The join of BC and AD is the Cartesian Product which could be much bigger than ABCD

2. {ABC, CA, CD}: decompose into ACD and BC

i) The candidate keys are AB and BC

ii) The decomposition is lossless since ACD ∩ BC (which is C) → ACD however it is not dependency preserving since the dependency AB → C is not preserved.

3. {ABC, CAD}: decompose into ABC and AD

i) The candidate keys are A and C.

ii) Since A and C are already candidate keys it makes no sense to decompose because it is already in BCNF. It is also not dependency preserving therefore it is bad.

4. {AB, BC, CD}: decompose into AB and ACD

i) The candidate key is A.

ii) This is a lossless decomposition since A is a key however it is not dependency preserving B → C is not preserved.

5. {AB, BC, CD}: decompose into AB, AD and CD

i) The candidate key is A

ii) This decomposition is lossless since A is the key but it is not dependency preserving (consider BC) therefore it is not the best decomposition.

**3)**

You are given a relation scheme **R** = {B, N, S, T, A, R, C} where **B** = Building, **N** = Door Number, **S** = Street, **T** = Type, **A** = Architect, **R** = Subcontractor and **C** = Class. Constraints between the attributes can be expressed in the form of the following functional dependencies:

**F** = {ABT, AB, RC, NSBT}

**a)** Find all candidate keys of **F**. Prove that these are the only keys.

By observation, one can see that the attributes A, R, N and S are never on the RHS of any FD therefore they must form part of at least one key. First however, we check if they themselves are a key.

ARNS+ = ARNSBTC

This is equivalent to R therefore ARNS is our minimum candidate key and therefore our only key.

**b)** Derive a canonical cover for **F** in a systematic manner.

**Step 1**: Put the FD's in the simple form.

F = {ABT, AB, RC, NSB, NST}

**Step 2**: Minimize the LHS of each FD

ABT, Can we remove B?

A+ = ABT

**Yes** we can remove B as T is an element of A+

ABT, Can we remove A?

B+ = B

**No,** we cannot remove B as T is not an element of B+

NSB, can we remove S?

N+ = N

**No**, we cannot remove S

NSB, can we remove N?

S+ = S

**No,** we cannot remove N

The same applies to NST as it did for NSB

The new functional dependency becomes

F2 = {AT, AB, RC, NSB, NST}

**Step 3**: Delete redundant FD's

Consider: AT

A+ = AB ; therefore not redundant

Consider: AB

A+ = AT ; therefore not redundant

Consider: RC

R+ = R ; therefore not redundant

Consider: NSB

NS+ = NST ; therefore not redundant

Consider: NST

NS+ = NSB ; therefore not redundant

Therefore, our canonical cover is:

FC = {AT, AB, RC, NSB, NST}

**c)** Does a set of FD’s have a unique canonical cover? Why?

A set of FD's does not have a unique canonical cover because the canonical cover depends on the order of the FD's that are considered to be left and right reduced.

**d)** Decompose **R** into a set of 3NF schemes which preserve all dependencies in **F** and form a lossless join.

From the above part, we get the following connections:

R1 = (A,B) with A B

R2 = (R,C) with R C

R3 = (N,S,B,T) with NS BT

The candidate key ANRS is not a subset of any of the three relations therefore R4 = (A, R, N, S) with the FD: ARNS+ ARNSmust be added to make it lossless.

**4)**

Consider the relation scheme R = {A, B, C, D, E, G, H} with the following FD’s: F = {CD -> E, B -> ACE, DH -> B, A -> D}

**a)** Find a 3NF decomposition of R that is both lossless and dependency-preserving. Show all your work and justify the various steps taken.

GH+ = GH (Not Key)

GHA+ = GHADBCE (Key)

GHB+ = GHBACED (Key)

GHC+ = GHC (Not Key)

GHD+ = GHDBACE (Key)

Candidate Keys: {GHA, GHB, GHD}

Canonical Cover:

**Step 1**: Put Into Simple Form

F = {CD -> E, B -> A, B -> C, B -> E, DH -> B, A -> D}

**Step 2**: Eliminate unnecessary attributes from LHS

CD -> E C+ = C

D+ = D

No change

DH -> B D+ = D

H+ = H

No change

Fc = {CD -> E, B -> A, B -> C, B -> E, DH -> B, A -> D}

**Relations:**

R1 = {CDE} F1 = {CD -> E}

R2 = {AB} F2 = {B -> A}

R3 = {BC} F3 = {B -> C}

R4 = {BE} F4 = {B -> E}

R5 = {BDH} F5 = {DH -> B}

R6 = {AD} F6 = {A -> D}

This decomposition is Dependency Preserving, but not Lossless because none of the Candidate Keys can be found in the above relations. In order to make it Lossless, we must add a single Candidate Key as a relation

R7 = {AGH} F7 = {}