Solution Tutorial 03 :

Exercise 01 :

We have in the following functional dependencies C dependent only from A:

A→E, F, G, C ;

A, B→C, M ; $\neg 2NF$

B→I, J ;

E→G, H ;

J→K

So after the normalization to 2NF we have :

A→E, F, G, C ; $\neg 3NF$

A, B→M ;

B→I, J ;

E→G, H ;

J→K

Now the normalization to 3NF: *we have G dependent to E so we eliminate the transitivity.*

A→E, F, C ;

A, B→M ;

B→I, J ;

E→G, H ;

J→K

Now all the FD are in 3NF.

Exercise 02: Learn how to deduce transitivity between FDs:

Q1. *{x->y ; z->w} => xz -> yw oui x×z -> y×w by concatenation.*

Q2. *{xy -> z ; z -> x} => z ->y non*

Lastname, Firstname -> address

address -> Lastname but address -/> Firstname in a house can live many persons.

Q3. *{x -> y ; y ->z} => x -> yz yes by transitivity on a x-> y et y ->z so x-> z which means x -> y and z hence we deduce that x-> yz*

Q4. *{ x-> y ; w->z } et w⊇y => x->z non*

Counterexample :

Address -> Lastname

Lastname, Firstname-> age but address -/> age.

Q5. *{w -> y, x-> z} => wx -> y yes by concatenation wx -> y and z ce qui signifie que wx -> y and wx -> z*

Q6. *{x -> y} et y ⊇ z => x -> z yes x->y and y ⊇ z so y= z∪w such that w= y-z which means x->y we have x-> z ∪w so x -> z and x-> w.*

Which means x -> y so x -> z∪w hence x-> z and x -> w

Q7. *{x -> y, x ->w, wy-> z} => x-> z yes*

x-> y and x->w and we have wy->z which means x -> z.

Q8. *{x ,y -> z, y-> w} => x w ->z no counterexample :*

Lastname, N°mobil -> Firstname

N°mobil -> address mais nom, address -/> Firstname

Q9. *{x -> y, xy -> z} => x->z yes x->y and xy -> z by transitivity we obtain y from x so we can eliminate from x y ->z we obtain x -> z.*

Exercise 03 :

1. ***Universal schema:***

*R(*Book-ID, Title, Lname-A, Fname-A, Copy-ID, Nbre-pages, Code-Pub, Name-Pub, year, Code-depot, Name-depot, City, Date, Qte)

2. ***Normalization:***

*R1((*Book-ID, Title, Lname-A, Fname-A*)* $\neg 2NF$

*R2(*Book-ID, Copy-ID, Nbre-pages, Code-Pub, Name-Pub, year*)* $\neg 2NF$

*R3(*Book-ID, ,Copy-ID, Code-depot, Name-depot, City, Date, Qte*)* $\neg 2NF$

Normalization to 2NF

*R1((*Book-ID, Title)

R'_1*(*Book-ID, Lname-A, Fname-A*)*

*R2(*Book-ID, Copy-ID, Nbre-pages)

R'_2*(*Book-ID, Copy-ID, Code-Pub, year*)*

R''_2*(*Code-Pub, Name-Pub)

*R3(*Code-depot, Name-depot, City)

R'_3*(*Book-ID, ,Copy-ID, Code-depot, Date, Qte)

All relation now are in 2NF

Normalization to 3NF

All relation now are in 3NF

Renaming relations:

*Book ((*Book-ID, Title)

*Author (*Book-ID, Lname-A, Fname-A*)*

*Copy(*Book-ID, Copy-ID, Nbre-pages)

*Published (*Book-ID, Copy-ID, Code-Pub, year*)*

*Publisher (*Code-Pub, Name-Pub)

*Depot(*Code-depot, Name-depot, City)

*Stored (*Book-ID, ,Copy-ID, Code-depot, Date, Qte)

Exercise 04:

1. Universal schema:

R(Supp-ID ,Name –Supp, Adr-Supp, Prod-ID, Prod-Des, Prod-name, Weight, Unit-Price, Lname-Mer ,Fname-Mer, Adr-Mer, Date-order, QTE-purshased, Date-delivery, QTE-delivered) 1NF but $\neg 2NF$

2. Normalization:

R1(Supp-ID ,Name –Supp, Adr-Supp, Prod-ID) $\neg 2NF$

R2(Prod-ID, Prod-Des, Prod-name, Weight, Unit-Price,)

R3(Lname-Mer ,Fname-Mer, Adr-Mer)

R4(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-order, QTE-purshased)

R5(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-delivery, QTE-delivered)

Put it in 2NF:

R_1' *(Supp-ID ,Name –Supp, Adr-Supp)*

R_1'' *(Supp-ID , Prod-ID)*

R2(Prod-ID, Prod-Des, Prod-name, Weight, Unit-Price,)

R3(Lname-Mer ,Fname-Mer, Adr-Mer)

R4(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-order, QTE-purshased)

R5(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-delivery, QTE-delivered)

All relation in 2NF and in 3NF.

Renaming relation:

Supplier(Supp-ID ,Name –Supp, Adr-Supp)

Have(Supp-ID , Prod-ID)

Product(Prod-ID, Prod-Des, Prod-name, Weight, Unit-Price,)

Merchant(Lname-Mer ,Fname-Mer, Adr-Mer)

Order(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-order, QTE-purshased)

Delivery(Supp-ID, Lname-Mer ,Fname-Mer, Prod-ID, Date-delivery, QTE-delivered)

Exercise 05:

1. Universal schema:

R(Stud-ID ,Stud-Lname, Stud-Fname, Stud-Adr, Course-C, Title, Summary, Code-pgm , Name-pgm, Prof-ID, Prof-Lname, Prof-Fname, Rank, Prof-Adr) 1NF but $\neg 2NF$

2. Normalization:

Step01:

R1(Stud-ID ,Stud-Lname, Stud-Fname, Stud-Adr, code-pgm) $\neg 2NF$

R2(Prof-ID, Code-pgm , Name-pgm) $\neg 2NF$

R3(Stud-ID, Course-C, title, summary) $\neg 2NF$

R4(Prof-ID, Prof-Lname, Prof-Fname, Rank, Prof-Adr, Cource-C) $\neg 2NF$

Step02:

R_1' *(*Stud-ID ,Stud-Lname, Stud-Fname, Stud-Adr*) 2NF*

R_1'' *(Stud-ID, Code-pgm) 2NF*

R_2'*(Code-pgm , Name-pgm)*

R_2'' *(Prof-ID, Code-pgm)*

R_3'*(Course-C, title, summary)*

R_3'' *(Stud-ID, Course-C)*

R_4'*(Prof-ID, Prof-Lname, Prof-Fname, Rank, Prof-Adr)*

R_4'' *(Prof-ID, Cource-C)*

Step 03:

All relation in 3NF.

Renaming relation:

*Student(*Stud-ID ,Stud-Lname, Stud-Fname, Stud-Adr*) 3NF*

Choose(Stud-ID, Code-pgm) 3NF

Program(Code-pgm , Name-pgm) 3NF

Managed (Prof-ID, Code-pgm) 3NF

Course (Course-C, Title, Summary) 3NF

*Take (*Stud-ID, Course-C*) 3NF*

Professor(Prof-ID, Prof-Lname, Prof-Fname, Rank, Prof-Adr)3NF

Created (Course-C, Prof-ID) 3NF