

**Βάσεισ δεδομενων**

**σειρα ασκησεων**



May 26, 2021

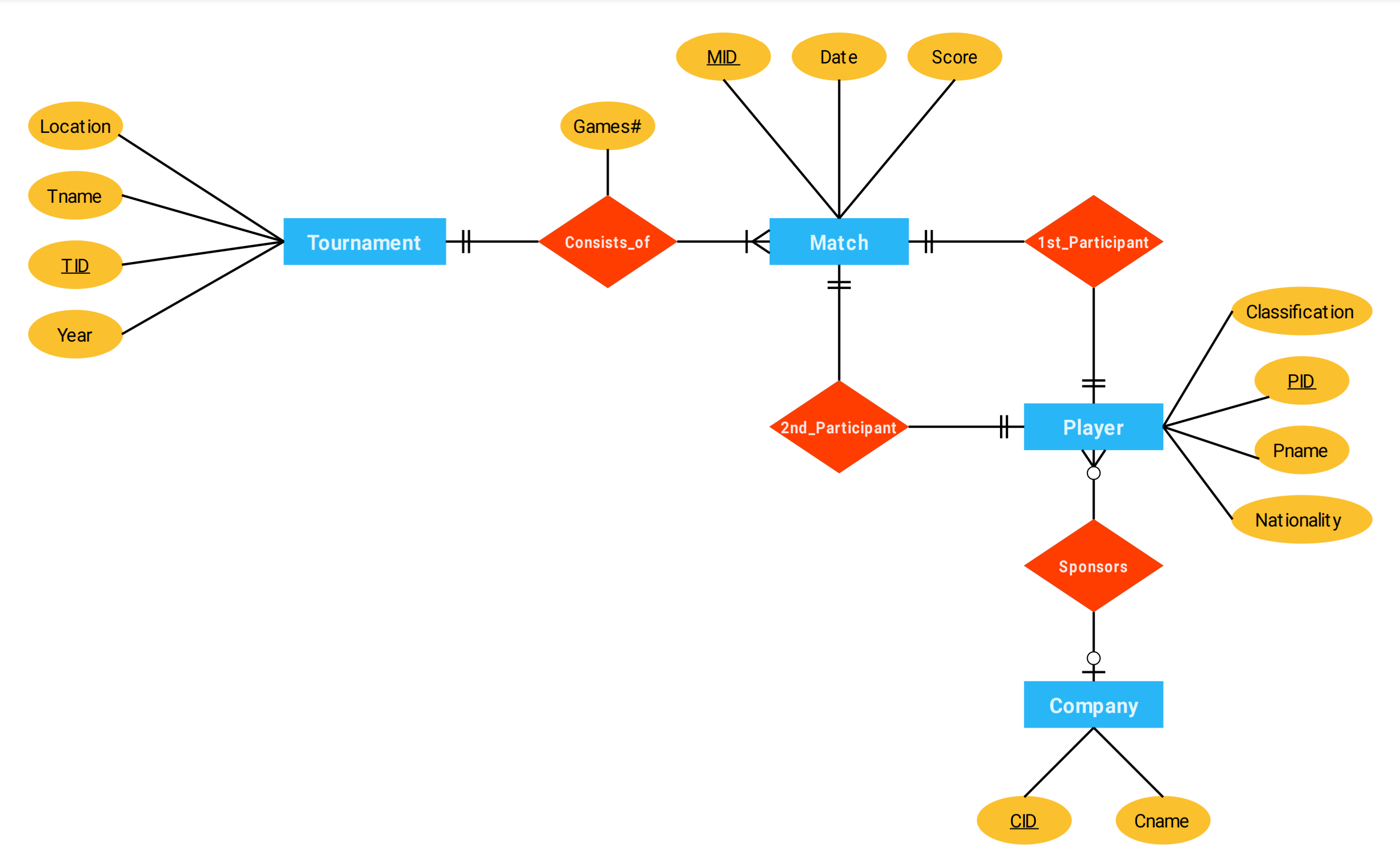
ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

ΚΡΙΣ ΚΟΥΤΣΗ – EL18905

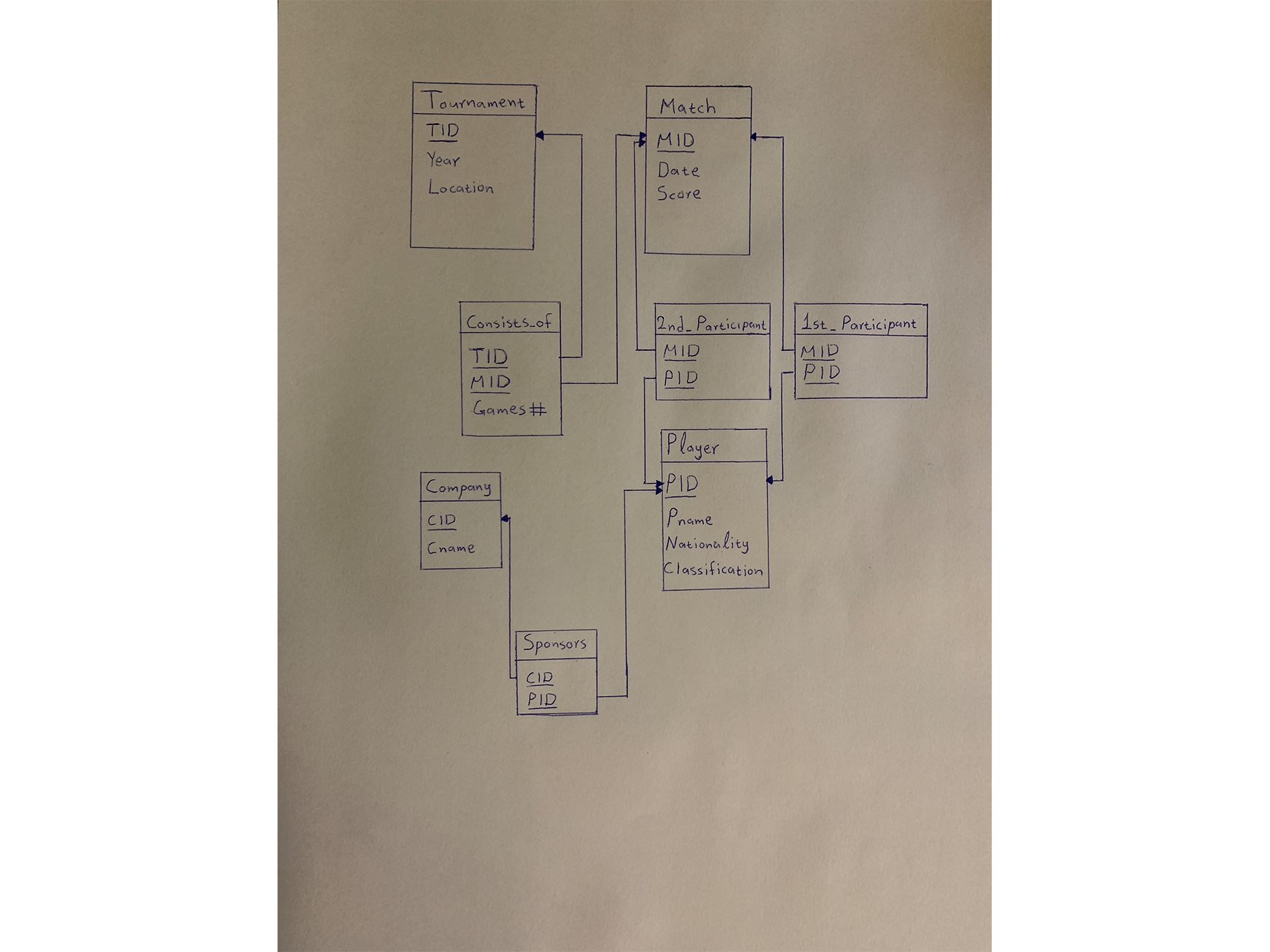
ΑΡΙΑΔΝΗ ΚΑΖΔΑΓΛΗ – EL18838

**Άσκηση 1**

**Α.**



**Β.**



**Άσκηση 2**

Q1.

Π*pid*(σ *companyname = “Google”* (σ *Company.cid = Person.cid* (Company × Person)))

Π*pid* (σ *companyname = “Facebook”* (σ *Company.cid = Shares.cid ^ sharenum > 500* (Company × Shares)))

Q2.

Π*pid* (σ *Person.cid = Shares.cid ^ Person.managerid = Shares.pid* (Person × Shares))

Q3.

shares\_1 <- (Π*pid,cid* (Shares))

shares\_2 <- (Π*pid,cid* (Shares))

shares\_3 <- (Π*pid,cid* (Shares))

Π*shares\_1.pid* (σ *shares\_1.cid <> shares\_3.cid* (shares\_1 ⋈((shares\_1.pid = shares\_2.pid) ^ (shares\_1.cid <> shares\_2.cid)) shares\_2

⋈((shares\_2.pid = shares\_3.pid) ^ (shares\_2.cid <> shares\_3.cid)) shares\_3))

Q4.

Π*pid* ((Π*pid,cid* Shares / Π*cid* Company) ⋈pid = pid Person)

**Άσκηση 3**

**A.**

Q1.

Βρες το id των καταστημάτων και τα ονόματά τους που έχουν το πολύ 100 εργαζόμενους ή βρίσκονται στην Αθήνα.

Q2.

Βρες τα ονόματα των καταστημάτων που προμηθεύουν μολύβια.

Q3.

Βρες τα ονόματα των καταστημάτων και την πόλη στην οποία βρίσκονται, που προμηθεύουν όλα τα εμπορεύματα που προμηθεύει το κατάστημα '0808'

**Β.**

Q1. **SELECT** storeid, sname

**FROM** Store

**WHERE** employee\_number <= 100 **OR** city = ‘Αθήνα’;

Q2. **SELECT** S.sname

**FROM** Store **AS** S, Goods **AS** G, Supply **AS** U

**WHERE** G.gname = ‘μολύβι’ **AND** G.gid = U.gid **AND** S.storeid = U.storeid;

Q3. **SELECT** sname, city

**FROM** Store, Supply

**WHERE** gid = **ALL** (**SELECT** gid

**FROM** Supply

**WHERE** storeid = ‘0808’);

Q4. **SELECT** sname

**FROM** Store

**WHERE** storeid **IN**(**SELECT** storeid, count(gid) **AS** count

**FROM** Supply

**GROUP** **BY** (storeid)

**ORDER** **BY** count **DESC**

**LIMIT** 5);

Q5. **SELECT** S.city

**FROM** Store **AS** S, Goods **AS** G, Supply **AS** A

**WHERE** A.gid = G.gid **AND** S.storeid = A.storeid **AND** 200 < G.price

**AND** G.price **IN**(**SELECT** MAX(price)

**FROM** Goods **AS** F, Store **AS** C, Supply **AS** U

**GROUP** **BY** (C.storeid)

**WHERE** U.gid = F.gid **AND** C.storeid = U.storeid);

Q6. **SELECT** gid

**FROM** Supply **AS** S1, Store **AS** S

**WHERE** S.city = ‘Αθήνα’ **AND** S1.storeid = S.stroreid **AND**

**NOT** **EXISTS** (**SELECT** **DISTINCT** gid

**FROM** Store, Supply

**WHERE** city = ‘Αθήνα’ **AND** Store.storeid = Supply.storeid)

**EXCEPT**

(**SELECT** gid

**FROM** Supply **AS** S2

**WHERE** S1.gid = S2.gid);

Q7. **SELECT** gid

**FROM** Supply, Store

**WHERE** city = ‘Αθήνα’ **AND** Store.storeid = Supply.storeid

**AND** gid **NOT** **IN** (**SELECT** gid

**FROM** Supply, Store

**WHERE** city = ‘Πάτρα’ **AND** Supply.storeid = Store.storeid);

**Άσκηση 4**

R(ABCDE), F={B -> EA, EBC -> D, BED -> A}

**A)**

a = {B, C} (attributes που δεν εμφανίζονται δεξιά)

B -> EA αρά έχουμε {A, B, C, E}

EBC -> D αρά έχουμε {A, B, C, D, E}

οπότε το {B, C} είναι το μοναδικό υποψήφιο κλειδί.

**B)**

BED -> A θα δούμε αν το Ε είναι εξωτερικό attribute έχουμε αρχικά:

{B, D}

Β -> ΕΑ άρα έχουμε {A, B, D, E} που είναι μέσα το A άρα το Ε είναι εξωτερικό attribute.

To set γίνεται {B -> EA, EBC -> D, BD -> A}

Αντίστοιχα το D είναι εξωτερικό attribute από το BD -> A γιατι αν το αφαιρέσουμε τότε θα έχουμε το Β -> Α. Από το Β -> ΕΑ καταλήγουμε στο Α πάλι.

To set γίνεται {B -> EA, EBC -> D, B -> A} που είναι ίδιο με το set {B -> EA, EBC -> D}

Το Ε είναι εξωτερικό στο EBC -> D οπότε καταλήγουμε στο set Fc = {B -> EA, BC -> D} που είναι η κανονική κάλυψη.

Άρα εύκολα βρίσκουμε την ελάχιστη κάλυψη, όπου η μόνη διαφορά της είναι πως δε γίνεται να έχουμε παραπάνω από 1 attribute στο δεξιό κομμάτι.

Η ελάχιστη κάλυψη έχει το set {B -> E, B -> A, BC -> D}

**C)**

Εξετάζουμε αρχικά αν το R ικανοποιεί την 3NF.

Το BC είναι superkey του R, οπότε η εξάρτηση BC -> ικανοποιεί τις προϋποθέσεις για 3NF.

Αντίθετα το B -> EA παραβιάζει τις προϋποθέσεις για 3NF (non-trivial, όχι superkey).

Έχουμε ως prime attributes τα B, C (BC candidate key).

Για να είναι το R σε 3NF θα πρέπει να ικανοποιείται η 3η συνθήκη.

Έχουμε B -> EA. Θα πρέπει κάθε attribute του συνόλου EA – B να είναι prime attribute. Δεν ισχύει, άρα το R δεν είναι σε 3NF, οπότε δεν θα ικανοποιεί ούτε τις πιο αυστηρές κανονικές μορφές.

Εξετάζουμε αν ικανοποιείται η 2NF. Δεν ικανοποιείται γιατί το B είναι proper subset του candidate key BC.

Συνεπώς θεωρούμε πως το R ικανοποιεί μόνο την 1NF, εφόσον τα attributes περιέχουν atomic values.

**D)**

Ελέγχουμε τα functional dependencies του Fc.

Το BC -> D δεν παραβιάζει την 3NF (BC superkey).

Το B -> EA παραβιάζει την 3NF γιατί το LHS δεν είναι superkey και το RHS δεν περιέχεται στο candidate key.

Οπότε καταλήγουμε στην ακόλουθη αποσύνθεση:

R1 = {ABE}

R2 = {BCD}

**Άσκηση 5**

R(ABCD), F = {AB -> C, B -> D, C -> A}

**A)**

a = {B} (attributes που δεν εμφανίζονται δεξιά)

B -> D άρα έχουμε {B, D}. Το Β δεν είναι superkey

b = {D} (attributes που βρίσκονται δεξιά αλλά όχι αριστερά)

Άρα πρέπει να δοκιμάσουμε τα {B, A} και {Β, C}

Ξεκινάμε με το {A, B}

AB -> C άρα έχουμε {A, B, C}

B -> D άρα έχουμε {A, B, C, D}

Το {A, B} είναι υποψήφιο κλειδί

Δοκιμάζουμε το {B, C}

C -> A άρα έχουμε {A, B, C}

B -> D άρα έχουμε {A, B, C, D}

Το {B, C} είναι υποψήφιο κλειδί

**B)**

Το F είναι ήδη σε minimal μορφή.

Το B -> D παραβιάζει τη BCNF αφού το LHS δεν είναι superkey.

Οπότε κάνουμε την παρακάτω αποσύνθεση:

R1 = {B, D}, F1 = {B -> D}

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

Το R1 είναι σε BCNF καθώς το B είναι superkey στο R1.

Όσον αφορά το R2, το C -> A παραβιάζει την BCNF αφού το LHS δεν είναι superkey.

Οπότε κάνουμε την παρακάτω αποσύνθεση:

R21 = {A, C}, F21 = {C -> A}

R22 = {B, C}

Το R21 είναι σε BCNF γιατί το C είναι superkey στο R21.

Το R22 είναι σε BCNF αφού κανένα dependency στο F δεν έχει όλα του τα attributes στο R22.

Οι εξαρτήσεις δε διατηρούνται γιατί δεν μπορούμε να υπολογίσουμε το AB -> C χωρίς να κάνουμε το R21⋈R22.