

Βάσεις Δεδομένων

Σειρά Ασκήσεων

Ακαδημαϊκό έτος 2020-2021

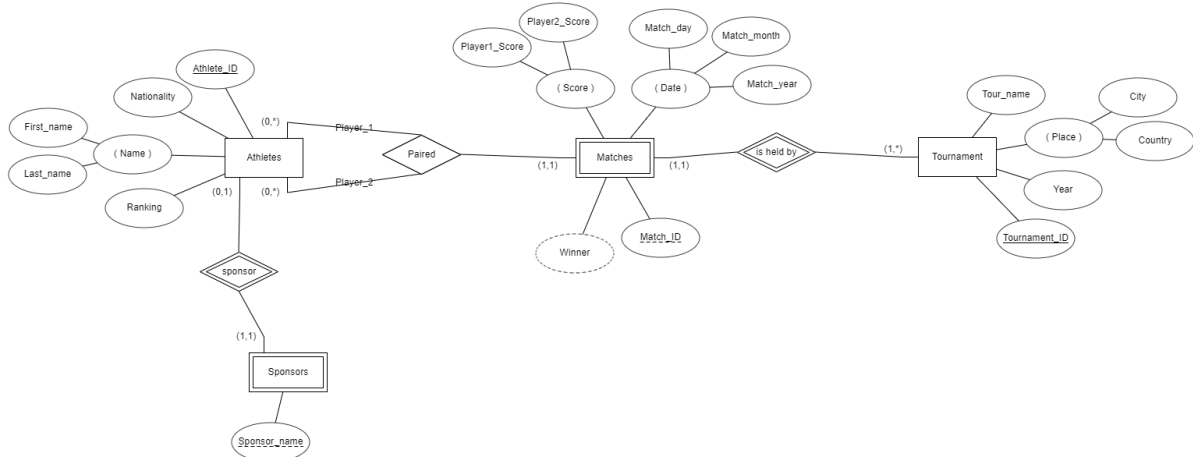
Χριστίνα Προεστάκη | AM : 03118877

Κωνσταντίνος Μορές | AM : 03118073

Αναστάσης Αγγλογάλλος | AM : 03118641

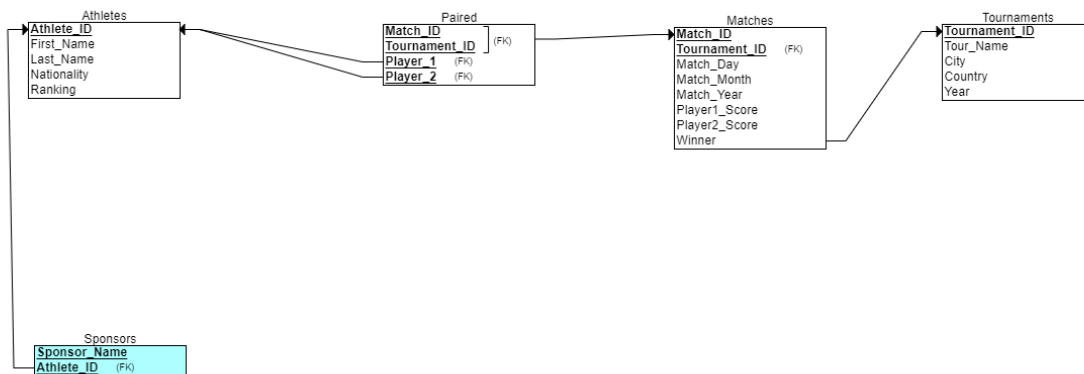
1η Άσκηση

A) Το ζητούμενο ER διάγραμμα:



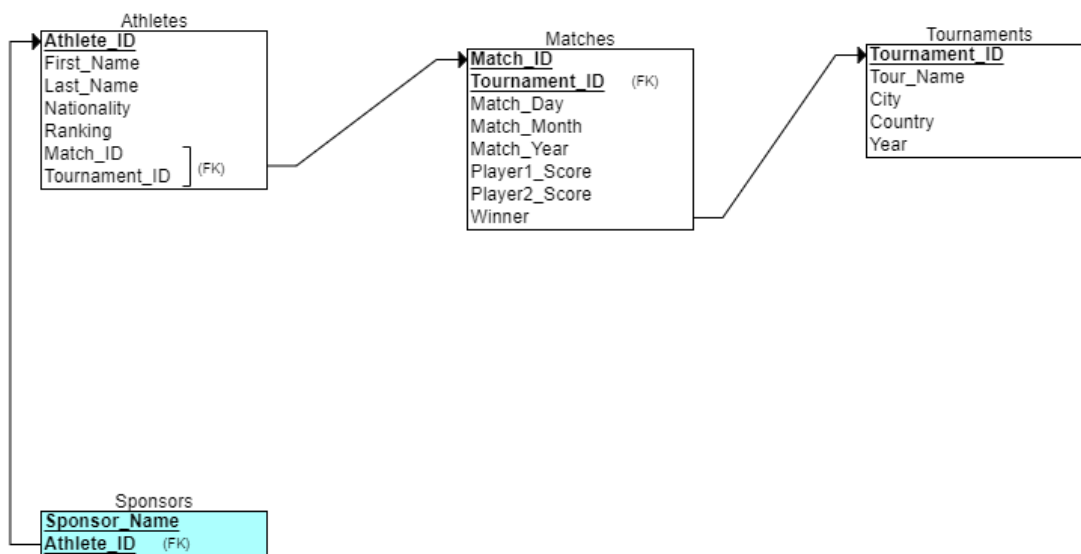
ΣΧΟΛΙΑ: Το Match_ID είναι μοναδικό μόνο για κάθε τουρνουά, γι αυτό και η οντότητα Matches είναι αδύναμη και εξαρτάται από την οντότητα Tournaments. Επίσης, θεωρήθηκε ότι κάποιος παίκτης της βάσης δεδομένων μπορεί να μην έχει συμμετάξει σε κάποιον αγώνα. Το attribute winner είναι derived και προκύπτει από το score. Επιπλέον, θεωρήθηκε ότι ένα τουρνουά πρέπει να έχει τουλάχιστον έναν αγώνα. Κάθε αθλητής έχει ένα μοναδικό Athlete_ID που τον χαρακτηρίζει. Αντίστοιχα κάθε τουρνουά έχει ένα Tournament_ID. Κάθε σπόνσορας πρέπει να χρηματοδοτεί έναν παίκτη. Τέλος, μέσω της σχέσης paired σχηματίζεται ένα ζευγάρι δύο αθλητών το οποίο συνδέεται με έναν συγκεκριμένο αγώνα.

B) Το σχεσιακό σχήμα που προκύπτει από το παραπάνω ER διάγραμμα:



ΣΧΟΛΙΑ: Οι σχέσεις “is held by” και “sponsor”, καθώς χαρακτηρίζουν αδύναμες οντότητες, μπορούν να εκφραστούν με ένα attribute στις αδύναμες οντότητες που θα περιέχει το primary key της δυνατής οντότητας που τις χαρακτηρίζουν. Έτσι το sponsors έχει FK το Athlete_ID και το Matches έχει FK το Tournament_ID.

Το παραπάνω σχεσιακό σχήμα μπορεί να απλοποιηθεί ως εξής:



ΠΡΟΣΟΧΗ: Τα attributes του table Athletes “Match_ID” και “Tournament_ID” μπορεί να έχουν τιμές null επειδή δεν έχουμε total participation στη σχέση paired. Αν αυτό είναι επιθυμητό να αποφευχθεί, προτείνεται η χρήση του προηγούμενου σχεσιακού σχήματος.

2η Άσκηση

Person(pid, cid, salary, managerid)

Company(cid, companyname, location)

Shares(pid, cid, sharenum)

Q1. Βρες το pid των υπαλλήλων της εταιρείας με Google που έχουν περισσότερες από 500 μετοχές της εταιρείας Facebook.

Απάντηση Q1:

$\pi_{pid}((Person) \bowtie (\sigma_{companyname='Google'}(Company)) \bowtie ((\sigma_{sharenum>500}(Shares)) \bowtie (\sigma_{companyname='Facebook'}(Company)))$

Q2. Βρες το pid των υπαλλήλων που οι managers τους έχουν μετοχές στην εταιρεία που δουλεύουν.

Απάντηση Q2:

Ο manager και ο υπάλληλος δουλεύουν στην ίδια εταιρεία.

Ο μάνατζερ είναι και αυτός υπάλληλος στην εταιρεία. Οπότε το pid του είναι το managerid για τον υπάλληλο που διοικεί. Άρα:

$Manager \leftarrow (\sigma_{pid=managerid}(Person))$

$\pi_{pid}(Person \bowtie_{pid=managerid} Manager \bowtie_{cid=cid \wedge sharenum>0} Shares)$

Q3. Βρες το pid των υπαλλήλων που έχουν μετοχές από τουλάχιστον 3 εταιρείες.

Απάντηση Q3:

$\pi_{pid}(Person \bowtie_{Gcount(cid \geq 3)}(Shares))$

Q4. Βρες τους υπαλλήλους που έχουν μετοχές όλων των εταιρειών.

Απάντηση Q4:

$\pi_{pid,cid}(Shares / \pi_{cid}(Company))$

3η Άσκηση

Θεωρούμε το παρακάτω σχήμα βάσης που κρατά πληροφορίες σχετικές με τα προϊόντα που προμηθεύουν διάφορα μαγαζιά:

Store(storeid, sname, employee_number, city)

Goods(gid, gname, price)

Supply(storeid, gid)

A.

Q1. $\pi_{\text{storeid, sname}}(\sigma_{\text{employee_number} \leq 100 \text{ or city} = \text{"Αθήνα"}}(\text{Store}))$

Η παραπάνω έκφραση δηλώνει την επιλογή:

- των εγγραφών employee_number οι οποίες είναι μικρότερες ή ίσες από 100

Ή

- έχουν ως city την "Αθήνα".

Έπειτα από αυτές τις εγγραφές που πληρούν τα πάνω κριτήρια κρατάμε τις ιδιότητες τους storeid και sname.

SQL:

select storeid, sname

from Store

where employee_number <= 100 **or** city = "Αθήνα"

Q2. $\pi_{\text{sname}}(((\sigma_{\text{gname} = \text{"μολύβι"}}(\text{Goods}) \bowtie \text{Supply}) \bowtie \text{Store}))$

Η παραπάνω έκφραση δηλώνει την προβολή των ονομάτων(sname) των καταστημάτων(Store) τα οποία παρέχουν(Supply) προϊόν(Good) με το όνομα(gname) "μολύβι".

SQL:

select sname

from Store, Supply

where Store.storeid=Supply.storeid

in (**select** gid

from Goods, Supply

where gname="μολύβι" **and** Goods.gid=Supply.gid))

Q3. $\pi_{\text{sname, city}}((\text{Supply} / \pi_{\text{gid}}(\sigma_{\text{storeid} = \text{"0808"}}(\text{Supply}))) \bowtie \text{Store})$

Η παραπάνω εκφραση δηλώνει την προβολή του ονόματος(sname) και της πόλης(city) του καταστήματος(Store) τα οποία περιέχουν απόθεμα(Supply) τέτοιο

ώστε οι κωδικοί προϊόντων τους gid να εμπεριέχονται στο σύνολο κωδικών προϊόντων που βρίσκονται μέσα στο απόθεμα του καταστήματος(storeid='0808').

SQL:

```
select sname, city  
from Store, Supply  
where Store.storeid=0808 and Store.gid=Supply.gid
```

Intersect

```
select sname, city  
from Store, Supply  
where Store.gid=Supply.gid
```

B.

Σχήμα Βάσης:

Store(storeid, sname, employee_number, city)

Goods(gid, gname, price)

Supply(storeid, gid)

Q4. Βρείτε τα ονόματα των 5 μαγαζιών με τα περισσότερα προϊόντα.

```
Select top 5 Store.name  
from (  
    select St.sname , count(Su.gid)  
    from Store as St , Supply as Su  
    where St.storeid = Su.storeid  
    order by count(Su.gid)  
    ) as Store
```

Q5. Βρείτε τις πόλεις για τις οποίες το ακριβότερο προϊόν που πωλείται σε κάποιο μαγαζί κοστίζει περισσότερο από 200 ευρώ.

//”Εστω και ένα προϊόν να κοστίζει πάνω από 200 ευρώ τότε και το ακριβότερο θα πληρεί την συνθήκη.

```
select distinct St.sname  
from Store as St , Goods as G , Supply as Su  
where Su.storeid=St.storeid and Su.gid=G.gid and G.price>200
```

Q6. Βρείτε τα ids των προϊόντων που τα βρίσκει κανείς σε όλα τα μαγαζιά της Αθήνας.

```
select distinct Su.gid
```

from Supply **as** Su , Store **as** St
where St.sname **like** '%Athens%' **and** Su.storeid=St.storeid

Q7. Βρείτε όλα τα ids των προϊόντων που τα βρίσκει κανείς στην Αθήνα αλλά όχι στην Πάτρα.

```
//Δεν χρειάζεται select distinct καθώς το gid είναι κύριο κλειδί του Goods
select Su.gid                                     //Προϊόντα που βρίσκει κάποιος στην Αθήνα
from Supply as Su , Store as St
where St.sname like '%Athens%' and su.storeid=st.storeid
except                                           //αφαίρεση
select Su.gid
from Supply as Su , Store as St
where St.sname like '%Patra%' and su.storeid=st.storeid
                                           //Προϊόντα που βρίσκει κάποιος στην Αθήνα
```

4η Άσκηση

A) Ακολουθούμε τον αλγόριθμο για την εύρεση των υποψήφιων κλειδιών.

- A = BC in superkey
- B = A not in superkey

A+ = BC

= BEAC (B -> EA)

= BEACD (EBC -> D)

= R

Επομένως το BC είναι μοναδικό κλειδί.

B)

Μοναδικό γνώρισμα στο δεξί μέρος:

- B -> E
- B -> A
- EBC -> D
- BED -> A

Αντικατάσταση για το αριστερό μέρος:

- B -> E

- $B \rightarrow A$
- $EBC \rightarrow D \Rightarrow BC \rightarrow D$
- $BED \rightarrow A \Rightarrow BD \rightarrow A$

Εξετάζουμε αλλαγές στην κλειστότητα και αφαιρούμε. Προκύπτει:

Ελάχιστη Κάλυψη:

- **$B \rightarrow E$**
- **$B \rightarrow A$**
- **$BC \rightarrow D$**

Μοναδικότητα στο αριστερό μέρος για να προκύψει η κανονική κάλυψη:

- **$BC \rightarrow D$**
- **$B \rightarrow EA$**

C) Η R είναι 1NF εφόσον το κλειδί είναι το BC και τα EA εξαρτώνται από το B.

D) Ξεκινώντας από την κανονική κάλυψη:

- **$BC \rightarrow D$**
- **$B \rightarrow EA$**

Προκύπτουν:

(BCD)

(BEA)

Όπου BC: κλειδί

Η μορφή 3NF είναι η παραπάνω.

5η Άσκηση

A)

$A = \{B\}$

$B = \{D\}$

$A^+ =$

$BD (B \rightarrow D) \text{ not } R$

$R - D = ABC$

$AB^+ = ABC (AB \rightarrow C)$

$ABCD (B \rightarrow D)$

Άρα το **AB** είναι υποψήφιο κλειδί.

$CB^+ =$

$CBDA (C \rightarrow A, B \rightarrow D)$

Επομένως και το **CB** είναι υποψήφιο κλειδί.

B)

Πρέπει να σπάσουμε εφόσον έχουμε εξαρτήσεις που το αριστερό μέρος δεν είναι κλειδί.

Οι $B \rightarrow D$ και $C \rightarrow A$

- **$R1 = BD \rightarrow BCNF$, επειδή το **B** είναι κλειδί ($B \rightarrow D$)**
- **$R2 = BAC$**

Η σχέση $C \rightarrow A$ παραβιάζεται στην $R2$ εφόσον δε δίνει το **B**, επομένως το **C** δεν είναι κλειδί και πρέπει να σπάσω τη σχέση.

- **$R3 = CA \rightarrow BCNF$, επειδή το **C** είναι κλειδί ($C \rightarrow A$)**
- **$R4 = BC \rightarrow BCNF$ εφόσον το **BC** είναι υποψήφιο κλειδί**

Το αποτέλεσμα της αποσύνθεσης είναι **R1, R3, R4**.