

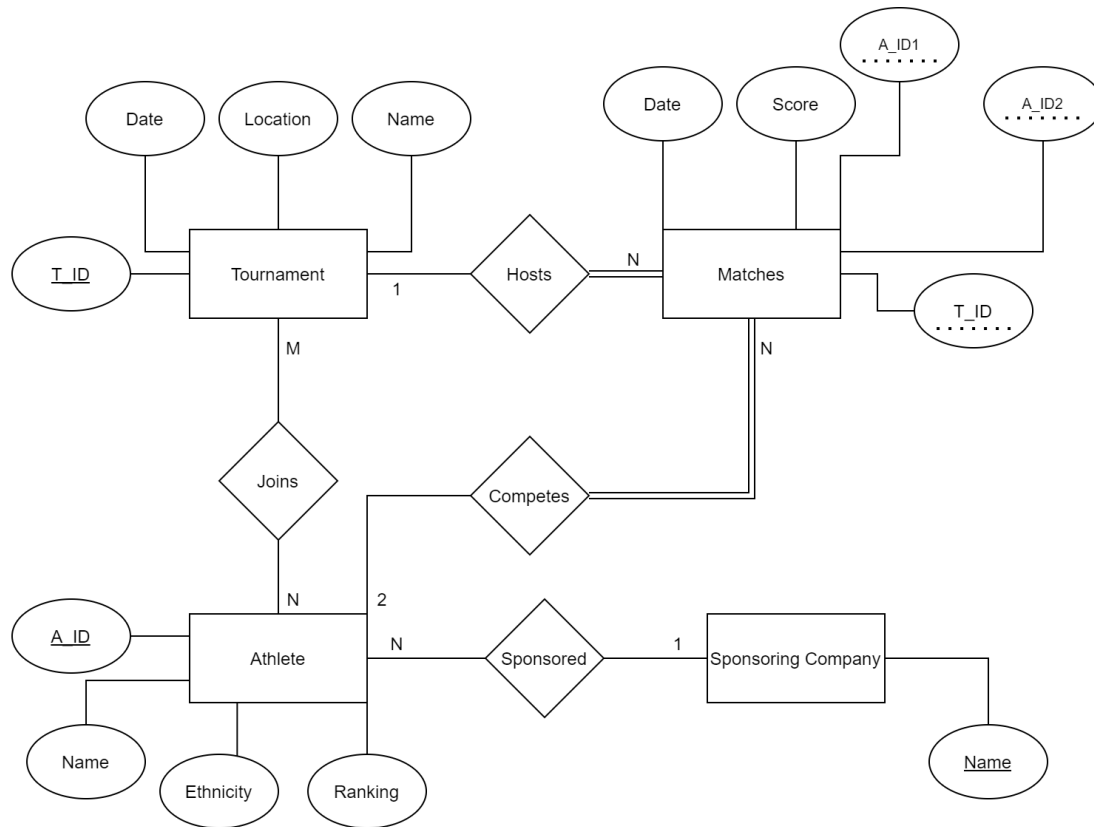
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Βάσεις Δεδομένων
Σειρά Ασκήσεων

Ομάδα 102
Αναστάσιος Λαγός - el13531
Κωνσταντίνος Βασιλάκης - el16504

Ασκηση 1

A.



To Entity matches έχει δυο Foreign Keys:

1) A_ID, το οποίο είναι composite και περιέχει τα IDs των δυο συμμετεχόντων.

2) T_ID, το οποίο είναι το ID του tournament.

Το τελευταίο είναι αναγκαίο διότι δυο αθλητές μπορούν να παίξουν στο ίδιο match αλλά σε διαφορετικό τουρνουά.

Ένας αθλητής μπορεί να γραφτεί σε πολλά tournaments και προφανώς ένα tournament μπορεί να έχει πολλούς αθλητές εγγεγραμμένους..

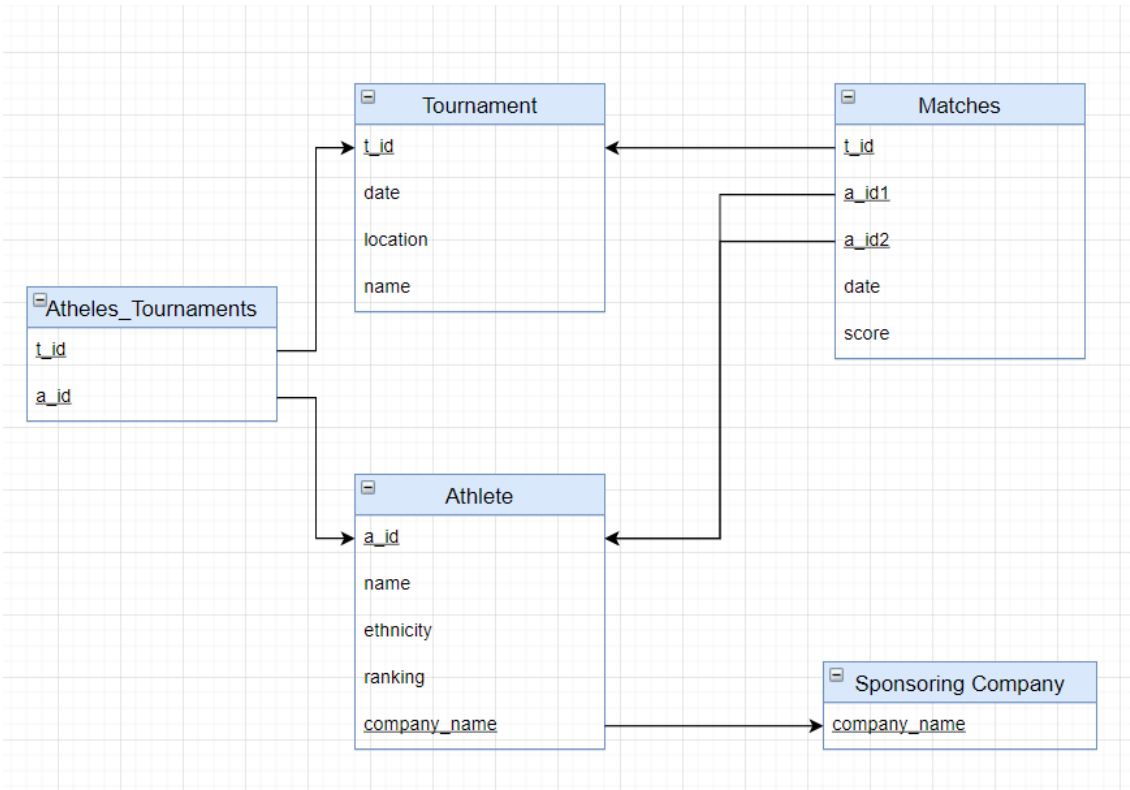
Θεωρούμε ότι σε ένα tournament μπορεί να μην γραφτεί κανείς για αυτό δεν έχουμε total participation μεταξύ Tournament-Joins.
Για τον ίδιο λόγο δεν έχουμε total participation μεταξύ Tournament-Hosts.

B.

Παρακάτω το σχεσιακό σχήμα του παραπάνω ER. Το matches έχει τρία foreign keys.

1. t_id. Περιέχει το id του τουρνουά στο οποίο παίρνει μέρος ο αγώνας.
2. a_id1. Περιέχει το id του πρώτου αθλητή που συμμετέχει στον αγώνα.
3. a_id2. Περιέχει το id του δεύτερου αθλητή που συμμετέχει στον αγώνα.

Η σχέση μεταξύ Athlete-Company είναι Many to One οπότε το foreign key ανήκει στο Athlete. Το Athlete - Tournament είναι Many to Many οπότε φτιάχνουμε έναν ξεχωριστό πίνακα με τους συνδυασμούς των foreign keys.



Ασκηση 2

Q1.

$$\begin{aligned}
 facebook_cid &\leftarrow \Pi_{cid}(\sigma_{companyname='Facebook'}(Company)) \\
 google_cid &\leftarrow \Pi_{cid}(\sigma_{companyname='Google'}(Company)) \\
 google_emp &\leftarrow \Pi_{pid}(\sigma_{cid=Google_cid}(Person)) \\
 \Pi_{pid}(\sigma_{pid=google_emp \wedge cid=facebook_cid \wedge sharenun > 500}(Shares))
 \end{aligned}$$

Q2.

$$\begin{aligned}
 &\rho_{(p_pid, p_cid)}(Person) \\
 join1 &\leftarrow (Person) \bowtie_{managerid=pid} (Shares) \\
 \Pi_{p_pid}(\sigma_{p_cid=cid}(join1))
 \end{aligned}$$

Q3.

$$\Pi(pid \quad g \quad (count(pid) > 3)(Shares))$$

Q4.

$$\Pi_{pid}(\sigma_{(cid = g \text{ } (count(cid))(Company))=(pid = g \text{ } (count(pid))(Shares))})$$

Ασκηση 3

A.

Q1. Βρίσκουμε τα stores με το πολύ 100 υπαλλήλους ή που βρίσκονται στην Αθήνα και επιστρέφουμε τα storeid και sname

```
SELECT storeid,sname FROM 'store'
WHERE employee_number<=100 and city='Αθήνα'
```

Q2. Βρίσκουμε τα stores που έχουν μολύβια και επιστρέφουμε το όνομα τους. Εδώ βρίσκουμε πρώτα τα goods μολύβια και κάνουμε δύο natural joins πρώτα με το supply στο gid και μετά με το store στο storeid.

```
SELECT store.sname FROM 'goods'
INNER JOIN supply ON goods.gid=supply.gid
INNER JOIN store ON supply.storeid=store.storeid
WHERE gname='Μολύβι'
```

Q3. Πρακτικά βρίσκουμε το sname και city από το μαγαζί με storeid = 0808. Αρχικά επιλέγουμε από το supply την γραμμή με storeid=0808 και κρατάμε το gid με ένα projection. Μετά με ένα division κρατάμε από το supply μόνο το storeid που αντιστοιχεί σε αυτό το gid και με ένα natural join στο Store παίρνουμε τα υπόλοιπα στοιχεία του store από τα οποία κρατάμε τα sname και city με ένα projection.

```
SELECT store.sname,store.city FROM 'supply'
INNER JOIN store ON store.storeid=supply.storeid
WHERE supply.storeid='1'
```

B.

Q4.

```
SELECT COUNT(*) as num_of_store_goods FROM 'store'
INNER JOIN supply ON store.storeid=supply.storeid
INNER JOIN goods ON supply.gid=goods.gid
GROUP BY store.storeid
ORDER BY num_of_store_goods DESC
LIMIT 5
```

Q5.

```
SELECT DISTINCT store.storeid FROM 'store'
INNER JOIN supply ON store.storeid=supply.storeid
INNER JOIN goods ON supply.gid=goods.gid
WHERE goods.price>200
```

Q6.

```
SELECT goods.*,COUNT(*) as store_count,
(SELECT COUNT(*) FROM 'store' WHERE store.city='Αθήνα') as total_stores
FROM 'store'
INNER JOIN supply ON store.storeid=supply.storeid
INNER JOIN goods ON supply.gid=goods.gid
WHERE store.city='Αθήνα'
GROUP BY goods.gid
HAVING store_count=total_stores
```

Q7.

```
SELECT DISTINCT goods1.gid FROM 'goods' as goods1
INNER JOIN supply ON supply.gid=goods1.gid
INNER JOIN store ON supply.storeid=store.storeid
WHERE store.city='Αθήνα' AND
NOT EXISTS
(SELECT DISTINCT goods2.gid FROM 'goods' as goods2
INNER JOIN supply ON supply.gid=goods2.gid
INNER JOIN store ON supply.storeid=store.storeid
WHERE store.city='Πάτρα' AND
goods1.gid=goods2.gid)
```

Ασκηση 4

A.

Βλέπουμε ότι τα B και C παρουσιάζονται μόνο στο αριστερό μέλος οπότε αυτά θα πρέπει να υπάρχουν στα υποψήφια κλειδιά. Παρατηρούμε ότι το closure του BC είναι

$$BC^+ = \{A, B, C, E\} = \{A, B, C, D, E\}$$

οπότε το BC είναι το μοναδικό υποψήφιο κλειδί.

B.

Έχουμε τις συναρτησιακές εξαρτήσεις

$$B \rightarrow EA, EBC \rightarrow D, BED \rightarrow A$$

Από αυτές μπορούμε να σπάσουμε το δεξιό μέλος στην πρώτη

$$B \rightarrow E, B \rightarrow A, EBC \rightarrow D, BED \rightarrow A$$

Η τέταρτη σχέση μπορεί να διαγραφεί αφού το A προκύπτει και από την δεύτερη σχέση

$$B \rightarrow E, B \rightarrow A, EBC \rightarrow D$$

Τέλος η 4η σχέση μπορεί να απλοποιηθεί αφαιρώντας το E αφού αυτό μπορεί να προκύψει από το B λόγω της πρώτης οπότε προκύπτει και η ελάχιστη κάλυψη

$$B \rightarrow E, B \rightarrow A, BC \rightarrow D$$

Μία κανονική κάλυψη μπορεί να δοθεί ενώνοντας τις δύο πρώτες σχέσεις

$$B \rightarrow EA, BC \rightarrow D$$

C.

Για να βρούμε την πιο αυστηρή κανονική μορφή χρησιμοποιούμε την απλοποιημένη έκφραση της ελάχιστης κάλυψης που βρήκαμε παραπάνω

$$B \rightarrow EA, BC \rightarrow D$$

Έχουμε ότι το υποψήφιο κλειδί είναι το BC και prime attributes = {B,C} , non-prime attributes = {A,D,E} ,

Ξεκινάμε από μία περισσότερο αυστηρή μορφή και πηγαίνουμε προς την λιγότερο.

BCNF δεν μπορεί να είναι γιατί πρέπει στο αριστερό μέλος να έχουμε μόνο super keys το οποίο δεν ικανοποιεί την πρώτη εξάρτηση

3NF δεν μπορεί να είναι γιατί πρέπει στο αριστερό μέρος να έχουμε super keys (ελέγχθηκε στο BCNF) ή στο δεξιό μέλος να έχουμε prime attribute που επίσης δεν ισχύει για την πρώτη σχέση

2NF δεν μπορεί να είναι γιατί η πρώτη σχέση αποτελεί μία μερική συσχέτιση αφού το αριστερό μέλος B είναι υποσύνολο του υποψήφιου κλειδιού BC και το δεξιο μέρος

είναι non prime attribute.

Οπότε η παραπάνω συσχέτιση είναι σε 1NF

D.

Ξεκινάμε βρίσκοντας το closure του B και δημιουργούμε ένα καινούριο υποσύνολο

$$R_1 = B^+ = \{B, E, A\}$$

Στο επόμενο υποσύνολο βάζουμε τα υπόλοιπα στοιχεία μαζί με το prime attribute B για να έχουμε lossless decomposition και επίσης αν μπορούμε να πάμε στο D αφού χρειάζεται και το B και το C. Οπότε

$$R_2 = \{B, C, D\}$$

Παρατηρούμε ότι

α) στο R_1 το B είναι super key του υποσυνόλου οπότε είναι και 3NF

β) στο R_2 το BC είναι super key του υποσυνόλου οπότε είναι και 3NF

Η παραπάνω αποσύνθεση εκτός από 3NF είναι και BCNF αφού στα αριστερά μέλη έχουμε μόνο super keys.

Ασκηση 5

1 - $A = B$

2 - $B = D$

3 - $A^+ \neq R$

4 - Closure of $R - B = \{ABC\}$

$$A \cup \{A\} = \{AB\}, \{AB\}^+ = ABCD = R$$

$$A \cup \{B\} = \{B\}, \{B\}^+ = BD \neq R$$

$$A \cup \{C\} = \{BC\}, \{BC\}^+ = ABCD = R$$

Attribute Closure

$$A^+ = A$$

$$B^+ = BD$$

$$C^+ = AC$$

$$D^+ = D$$

Τα CKs είναι

$$\{AB\}^+ = ABCD$$

$$\{BC\}^+ = ABCD$$

$AB \rightarrow C$ Είναι κλειδί οπότε δεν έχουμε παράβαση του BCNF.

$B \rightarrow D$ Εδώ δεν είναι κλειδί,

οπότε θα φτιάξουμε δυο σχέσεις τέτοιες ώστε, η πρώτη περιλαμβάνει attributes των FDs που παραβιάζουν το BCNF ενώ η δεύτερη τα αρχικά attributes εκτός το RHS του FD που παραβιάζει το BCNF:

$$(BC), (ABC)$$

Για το BD, το CK είναι το B.

Για το ABC, τα CKs είναι AB, BC. Για το AB έχουμε $AB \rightarrow C$ και επειδή το AB είναι κλειδί, ανήκει και στο BCNF.

$B \rightarrow D$, δεν υπάρχει κάπου.

$C \rightarrow A$ υπάρχει αλλά το C δεν είναι κλειδί οπότε αποσυνθέτουμε σε (BD)(CA)(BC).

Το FD (CA) το παίρνουμε όπως είναι ενώ τα άλλα δυο είναι χωρίς το A.

Το BD δεν άλλαξε οπότε είναι ακόμα στο BCNF.

Το CA έχει το C ως CK και του εφαρμόζεται το $C \rightarrow A$. Οπότε είναι σε BCNF.

Το BC έχει το BC ως CK οπότε είναι σε BCNF.

Τελικώς έχουμε (BD)(CA)(BC).

Οι εξαρτήσεις δεν διατηρήθηκαν.