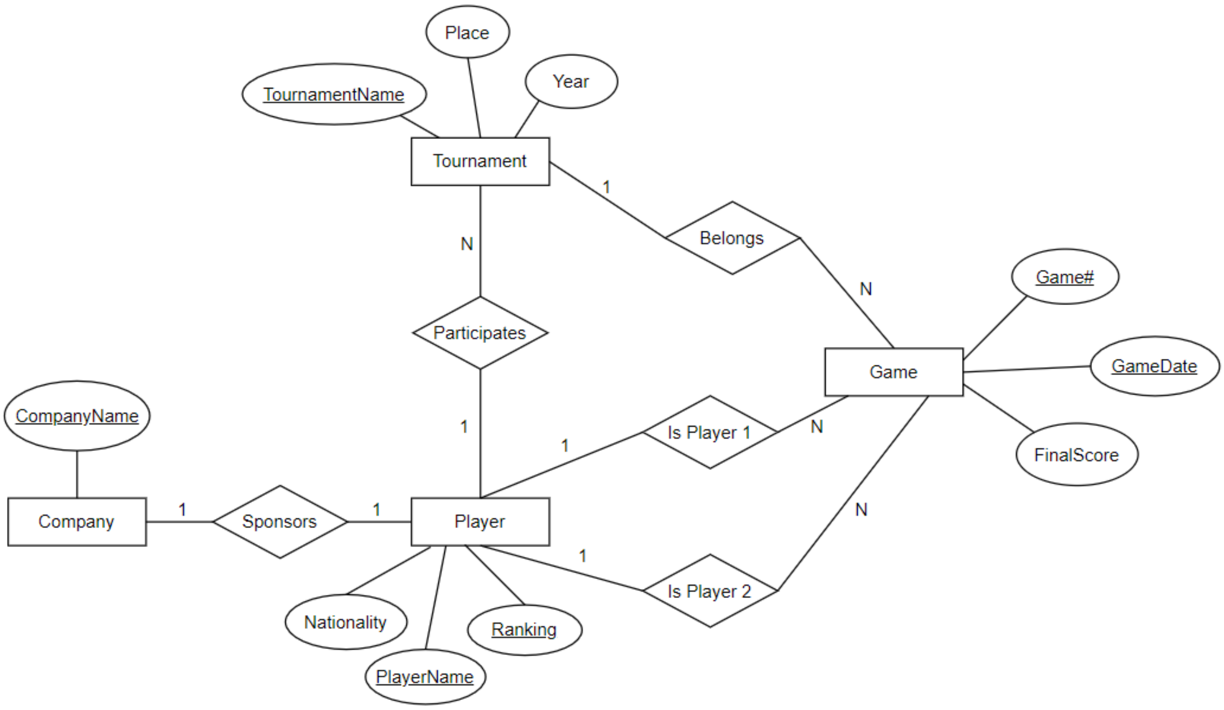
Ορφέας Φιλιππόπουλος el18082

Δέσποινα Γραμμένου el18061

Παπαρρηγόπουλος Θοδωρής el18040

Άσκηση 1:



Entities:

Tournament (name_of_tournament, place, date)

Game (game#, gamedate, final_score)

Player (name_of_player, nationality, ranking)

Business (name_of_business)

Relations:

Participates (name_of_tournament, name_of_player)

is_Player1 (name_of_player, game#, gamedate)

is_Player2 (name_of_player, game#, gamedate)

Belongs (<u>name_of_tournament, Game#, gamedate)</u>

Sponsors (<u>name_of_player, name_of_business)</u>

Άσκηση 2:

Q1) Αρχικά εκφράζουμε το pid αυτών που εργάζονται στην GOOGLE στην σχεσιακή άλγεβρα χρησιμοποιώντας natural join.

GoogleWorkers $\leftarrow \pi_{pid}$ ($\sigma_{companyname='Google'}$ (Person X Company))

Στη συνέχεια εκφράζουμε το pid και το sharenum αυτών που έχουν μετοχές στην εταιρεία Facebook και βρίσκουμε το τελικό αποτέλεσμα κάνοντας natural join μεταξύ αυτών που εργάζονται στην GOOGLE και έχουν μετοχές στην Facebook.

Facebookshares $\leftarrow \pi_{pid,sharenum}$($\sigma_{companyname='Facebook'}$(Company X Shares))

Final $\leftarrow \pi_{pid}$ ($\sigma_{sharenum>500}$(GoogleWorkers X Facebookshares))

b)

Θα βρούμε το pid των υπαλλήλων που οι managers τους έχουν μετοχές στην εταιρεία που δουλεύουν χρησιμοποιώντας καρτεσιανό γινόμενο.

Final $\leftarrow \pi_{Person.pid}$($\sigma_{Shares.pid=Person.managerid}$(Person $_{(Person.cid=Shares.cid)}$X Shares))

c)

Αρχικά θα εκφράσουμε το pid και το cid των υπαλλήλων που έχουν μετοχές σε τρείς εταιρείες και στη συνέχεια χρησιμοποιώντας καρτεσιανό γινόμενο μεταξύ αυτών και στη συνέχεια διαλέγοντας το pid να είναι ίδιο και το cid διαφορετικό προκύπτει το τελικό αποτέλεσμα.

Shares1 $\leftarrow \pi_{pid,cid}$(Shares)

Shares2 $\leftarrow \pi_{pid,cid}$(Shares)

Shares3 $\leftarrow \pi_{pid,cid}$(Shares)

Final $\leftarrow$ $\pi_{pid}(\sigma_{((Shares1.pid=Shares2.pid=Shares3.pid) \wedge (Shares1.cid <> Shares2.cid) \wedge (Shares1.cid <> Shares3.cid) \wedge (Shares2.cid <> Shares3.cid))}$ (Shares1 X Shares2 X Shares3))

d)Αρχικά κάνουμε project το cid των εταιρείων. Διαλέγουμε το pid και το cid των υπαλλήλων που έχουν μετοχές και χρησιμοποιώντας division προκύπτει το τελικό αποτέλεσμα.

Companies $\leftarrow \pi_{cid}$(Company)

Shares1 $\leftarrow \pi_{pid,cid}$(Shares)

Final $\leftarrow \pi_{pid}$(Shares1 / Companies)

Άσκηση 3:

A.

Q1.

```
SELECT storeid, sname
FROM store
WHERE (employee_number <= 100) or (city = "Αθήνα") ;
```

Q2.

```
SELECT store.sname
FROM
(
   SELECT *
   FROM
   (      SELECT *
           FROM Goods
```

```
        WHERE gname = “μολύβι”
    ) as pencils
    INNER JOIN Supply ON Supply.gid = pencils.gid
) as pencilsupply
INNER JOIN Store ON store.storeid = pencilsuppy.storeid;
```

Q3.

```
SELECT Store.sname, Store.city
FROM
(
        SELECT *
        FROM Supply as sx
        WHERE NOT EXISTS (
          (SELECT sy.gid FROM Supply as sy WHERE sy.storeid = “0808”)
          EXCEPT
          (SELECT sp.gid FROM Supply as sp WHERE sp.storeid = sx.storeid)
        )
) as div
INNER JOIN Store ON Store.storeid = div.storeid;
```

Q4.

```
SELECT Store.sname
FROM
(
        SELECT COUNT(gid), storeid
        FROM Supply
        GROUP BY storeid
        ORDER BY COUNT(gid) DESC
```

```sql
    LIMIT 5
) as top5
INNER JOIN Store ON Store.storeid = top5.storied;
```

Q5.

```sql
SELECT Store.city
FROM
(
    SELECT *
    FROM Supply
    INNER JOIN Goods ON Supply.gid = Goods.gid
) as res
INNER JOIN Store ON Store.storeid = res.storeid
GROUP BY Store.city
HAVING (MAX(res.price) > 200);
```

Q6.

```sql
SELECT ath.gid
FROM
(
    SELECT Supply.storeid, Supply.gid
    FROM Supply, Store
    WHERE Supply.storeid = Store.storeid AND Store.city = “Αθήνα”
) as ath
WHERE NOT EXISTS (
  (SELECT sy.storeid FROM ath as sy)
  EXCEPT
  (SELECT sp.storeid FROM ath as sp WHERE sp.gid = ath.gid)
```

);

Q7.

```
SELECT ath.gid
FROM
(
	SELECT Supply.storeid, Supply.gid
	FROM Supply, Store
	WHERE Supply.storeid = Store.storeid AND Store.city = “Αθήνα”
) as ath
WHERE ath.gid NOT IN
(
	SELECT Supply.gid
	FROM Supply, Store
	WHERE Supply.storeid = Store.storeid AND Store.city = “Πάτρα”
);
```

Άσκηση 4:

A. Θα χρησιμοποιήσουμε τα Functional dependencies για να βρούμε τα κλειδιά. Αρχικά βρίσκουμε τα attributes που δεν είναι στην δεξιά μεριά κάποιου functional dependency. Παρατηρούμε ότι αυτά που δεν βρίσκονται στην δεξιά μεριά είναι τα B,C. Στη συνέχεια βρίσκουμε τα attributes που εμφανίζονται στην δεξιά μεριά κάποιου functional dependency αλλά όχι στην αριστερή. Στην δική μας περίπτωση είναι το A. Άρα προκύπτουν τα εξής: LHS: A={B,C} και RHS: B = {A}. Θα υπολογίσουμε την κλειστότητα του {B,C}. Παρατηρούμε ότι η κλειστότητα του δίνει όλα τα χαρακτηριστικά του R οπότε είναι το μοναδικό candidate key.

B. Στα παρακάτω functional dependencies θα δούμε αν μπορούμε να ενώσουμε κάποιο functional dependency.

B $\rightarrow$ EA

EBC $\rightarrow$ D

BED $\rightarrow$ A

Βλέπουμε ότι δεν μπορούμε να ενώσουμε κάποιο functional dependency.Στη συνέχεια θα δούμε μήπως κάποιο χαρακτηριστικό είναι εξωτερικό. Το D στη σχέση BED → A είναι εξωτερικό καθώς από τη σχέση B → EA μπορούμε να παράγουμε το A και από τη σχέση EBC → D μπορούμε να παράγουμε το D.Οπότε τελικά θα είναι

F = (B → EA, EBC → D, BE → A)

Στην παραπάνω σχέση θα δούμε μήπως μπορούμε να ενώσουμε κάποιο functional dependency.Δεν μπορούμε να ενώσουμε. Οπότε θα ψάξουμε για extraneous attribute.Παρατηρούμε ότι το A στη σχέση B → EA είναι εξωτερικό καθώς αν αντικαταστήσουμε την παραπάνω σχέση στο F με την B → E θα δούμε ότι το B->E και το BE->A.Οπότε το A είναι εξωτερικό. Οπότε προκύπτει εκ νέου

F = (B → E, EBC → D, BE → A)

Βλέπουμε στην παραπάνω σχέση μήπως μπορούμε να ενώσουμε κάποιο functional dependency.Δεν μπορούμε να ενώσουμε. Οπότε θα ψάξουμε για extraneous attribute. Παρατηρούμε ότι το E στη σχέση BE → A είναι εξωτερικό. Το B → E και BE → A. Από τη στιγμή που το E παράγεται από το B,είναι εξωτερικό στη σχέση BE → A. Οπότε προκύπτει εκ νέου F = (B → E, EBC → D, B → A).

Τώρα ενώνουμε τις σχέσεις B → E και B → A σε μία: B → EA. Οπότε είναι F = (B → EA, EBC → D).Στη συνέχεια ελέγχουμε αν κάποιο attribute είναι εξωτερικό. Το B παράγει το E. Οπότε το E είναι εξωτερικό στην σχέση EBC → D.Οπότε είναι F = (B → EA, BC → D).

Στην παραπάνω μορφή της F δεν μπορούμε να ενώσουμε κάποιο functional dependency. Δεν υπάρχει κάποιο άλλο εξωτερικό attribute και κάθε αριστερή μεριά είναι μοναδική. Οπότε η κανονική κάλυψη της F είναι: F = (B → EA, BC → D).

Για να βρούμε την ελάχιστη κάλυψη θα σπάσουμε τις εξαρτήσεις που περιέχουν περισσότερα από ένα attributes στο δεξί μέρος. Οπότε προκύπτει

F = (B → E, B → A, BC → D)

C.

H R είναι σε 1NF. Δεν είναι σε 2NF καθώς ένα υποψήφιο κλειδί είναι το B,C και τα E,A εξαρτώνται από το B που είναι μέρος του candidate key(partial dependency).

D.

F = (B → EA, BC → D)

Για το functional dependency B → EA έχουμε το σχήμα R1 = (B,E,A)

Για το functional dependency BC → D έχουμε το σχήμα R2 = (B,C,D)

Θα ελέγξουμε αν κάποιο από τα R1,R2 περιέχει το candidate key,δηλαδή το BC.Παρατηρούμε ότι υπάρχει στο σχήμα R2.Οπότε δεν χρειάζεται να προσθέσουμε κάποιο άλλο σχήμα. Οπότε διασπάται σε R1 και R2,που είναι σε 3NF.

Άσκηση 5:

A. Ακολουθώντας τον αλγόριθμο που ακολουθήσαμε και στην τέταρτη άσκηση έχουμε ότι LHS: A = {B} και RHS: B = {D}. Παρατηρούμε ότι η κλειστότητα του B δεν μας δίνει όλα τα χαρακτηριστικά του R, επομένως για κάθε attribute x στο R – RHS = (A,B,C) κοιτάμε αν το BU{x} είναι candidate key.Παρατηρούμε ότι η κλειστότητα του (B,A) μας δίνει όλα τα χαρακτηριστικά του R οπότε είναι candidate key.Ακόμα η κλειστότητα του (B,C) μας δίνει όλα τα χαρακτηριστικά του R.Οπότε τα (B,A),(B,C) είναι candidate keys.

B.

Για να είναι μια σχέση σε BCNF μορφή θα πρέπει να είναι σε 3NF μορφή και για κάθε dependency A->B το A πρέπει να είναι super key.Βλέπουμε ότι η B->D παραβιάζει τις προϋποθέσεις του BCNF καθώς το B δεν είναι super key.Οπότε το R χωρίζεται σε R1 (ABC) και R2 (BD) με F1: AB → C, C → A και F2: B → D.

Το R2 είναι σε BCNF προφανώς.

Το R1 δεν είναι σε BCNF καθώς η σχέση C → A παραβιάζει τις προϋποθέσεις, καθώς το C δεν είναι superkey. Άρα το R1 χωρίζεται σε R3 (BC) και R4 (AC) με F3: $\emptyset$

και F4: C → A.

Τώρα τα R2,R3,R4 είναι όλα σε BCNF.

Παρατηρούμε πως χάνεται η εξάρτηση AB → C, μετά τη $2^{η}$ επανάληψη του αλγορίθμου.