

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος «ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (4Ο ΕΞ.)»

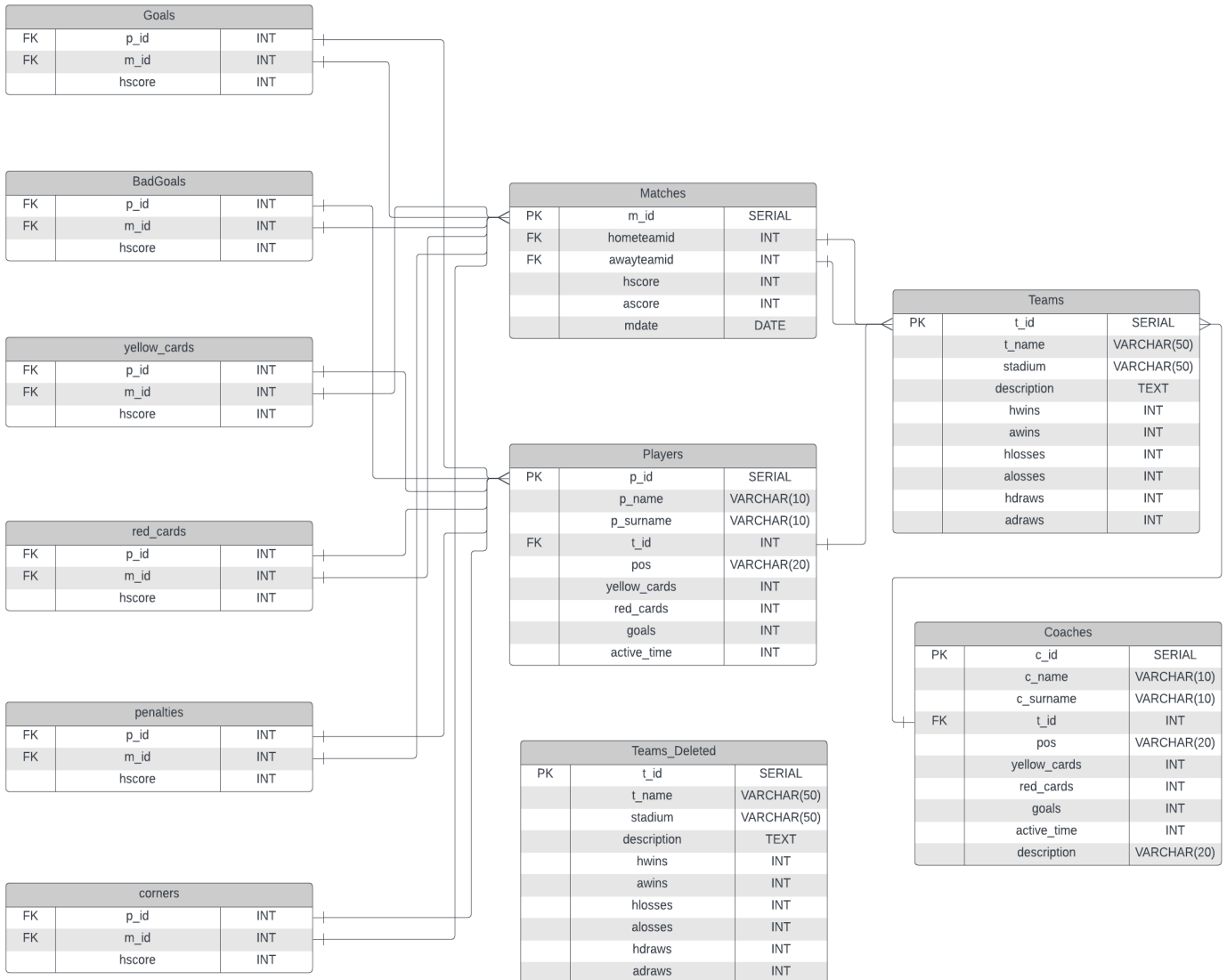
Τελική Εργασία Μαθήματος	ΤΕΛΙΚΟ ΠΑΡΑΔΟΤΕΟ ΜΑΘΗΜΑΤΟΣ
Όνομα φοιτητή – Αρ. Μητρώου	Λεωνίδας Πάστρας (Π20155)
	Άγγελος Μανουσάκης (Π20120)
	Φοίβος-Γεώργιος Καστανάς (Π21224)
Ημερομηνία παράδοσης	30/6/2023

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ:

Ερώτημα 1.a .....	3
Ερώτημα 1.c .....	9
Ερώτημα 2.a .....	10
Ερώτημα 2.b .....	11
Ερώτημα 2.c .....	12
Ερώτημα 2.d .....	13
Ερώτημα 3.a .....	13
Ερώτημα 3.b .....	15
Ερώτημα 4 .....	16

## Ερώτημα 1.α:

Το σχεσιακό σχήμα της ΒΔ είναι:



Η IsRested() ελέγχει ότι αν για μια ομάδα υπάρχει διάστημα 10 ημερών μεταξύ των αγώνων της.

```
CREATE FUNCTION IsRested(team_id INT, match_date DATE) RETURNS INT AS  
$$  
SELECT COUNT(team_id)  
FROM Matches  
WHERE (team_id = hometeamid OR team_id = awayteamid)  
AND (mdate BETWEEN match_date - INTERVAL '10 days' AND match_date + INTERVAL '10  
days');  
$$  
LANGUAGE SQL;  
  
CREATE FUNCTION NumOfPlayers(team_id int) RETURNS int AS '  
SELECT count(t_id) AS num_of_players FROM Players WHERE t_id=team_id;  
' LANGUAGE SQL;
```

Εντολή που φτιάχνει τον πίνακα 'Ομάδες':

```
CREATE TABLE Teams (  
  t_id SERIAL PRIMARY KEY,  
  t_name VARCHAR(50),  
  stadium VARCHAR(50),  
  description TEXT,  
  hwins INT,  
  awins INT,  
  hlosses INT,  
  alosses INT,  
  hdraws INT,  
  adraws INT  
);
```

Εντολή που φτιάχνει τον πίνακα 'Παίκτες':

```
CREATE TABLE Players (  
  p_id SERIAL PRIMARY KEY,  
  p_name VARCHAR(10),  
  p_surname VARCHAR(10),  
  t_id INT REFERENCES Teams(t_id) CHECK (numofplayers(t_id) < 11),  
  pos VARCHAR(20),  
  yellow_cards INT,  
  red_cards INT,  
  goals INT,  
  active_time INT  
);
```

Εντολή που φτιάχνει τον πίνακα 'Προπονητές':

```
CREATE TABLE Coaches (  
  c_id SERIAL PRIMARY KEY,  
  c_name VARCHAR(10),  
  c_surname VARCHAR(10),  
  t_id INT REFERENCES Teams(t_id),  
  pos VARCHAR(20),  
  yellow_cards INT,  
  red_cards INT,  
  goals INT,  
  active_time INT,  
  description VARCHAR(20)  
);
```

Εντολή που φτιάχνει τον πίνακα 'Αγώνες':

```
CREATE TABLE Matches (
```

```
m_id SERIAL PRIMARY KEY,  
hometeamid INT REFERENCES Teams (t_id),  
awayteamid INT REFERENCES Teams (t_id),  
hscore INT,  
ascore INT,  
mdate DATE,  
CONSTRAINT different_teams CHECK (hometeamid != awayteamid),  
CONSTRAINT hteam_rested CHECK (IsRested(hometeamid,mdate) < 1),  
CONSTRAINT ateam_rested CHECK (IsRested(awayteamid,mdate) < 1)  
);
```

```
CREATE TABLE Goals(  
p_id INT REFERENCES Players (p_id),  
m_id INT REFERENCES Matches (m_id),  
time_stamp int  
);
```

```
CREATE TABLE BadGoals(  
p_id INT REFERENCES Players (p_id),  
m_id INT REFERENCES Matches (m_id),  
time_stamp int  
);
```

```
CREATE TABLE yellow_cards(  
p_id INT REFERENCES Players (p_id),  
m_id INT REFERENCES Matches (m_id),  
time_stamp int  
);
```

```
CREATE TABLE red_cards(  
p_id INT REFERENCES Players (p_id),  
m_id INT REFERENCES Matches (m_id),
```

```
time_stamp int  
);
```

```
CREATE TABLE penalties(  
  p_id INT REFERENCES Players (p_id),  
  m_id INT REFERENCES Matches (m_id),  
  time_stamp int  
);
```

```
CREATE TABLE corners(  
  p_id INT REFERENCES Players (p_id),  
  m_id INT REFERENCES Matches (m_id),  
  time_stamp int  
);
```

## Οι περιορισμοί ακεραιότητας είναι:

### Πίνακας Players:

**PRIMARY KEY (p\_id):** Ορίζει το πεδίο p\_id ως πρωτεύον κλειδί, δηλαδή μοναδικό αναγνωριστικό για κάθε εγγραφή στον πίνακα.

**FOREIGN KEY (t\_id) REFERENCES Teams(t\_id):** Ορίζει το πεδίο t\_id ως ξένο κλειδί που αναφέρεται στον πίνακα Teams και συνδέει την καταχώριση του παίκτη με την ομάδα στην οποία ανήκει.

**CONSTRAINT CHECK (numofplayers(t\_id) < 11):** Επιβάλλει τον περιορισμό ότι σε μια ομάδα μπορούν να υπάρξουν μέχρι 11 παίκτες.

### Πίνακας Coaches:

**PRIMARY KEY (c\_id):** Ορίζει το πεδίο c\_id ως πρωτεύον κλειδί, μοναδικό για κάθε εγγραφή στον πίνακα.

**FOREIGN KEY (t\_id) REFERENCES Teams(t\_id):** Ορίζει το πεδίο t\_id ως ξένο κλειδί που αναφέρεται στον πίνακα Teams και συνδέει την καταχώριση του προπονητή με την ομάδα στην οποία ανήκει.

#### Πίνακας Teams:

**PRIMARY KEY (t\_id):** Ορίζει το πεδίο t\_id ως πρωτεύον κλειδί, μοναδικό για κάθε εγγραφή στον πίνακα.

#### Πίνακας Matches:

**PRIMARY KEY (m\_id):** Ορίζει το πεδίο m\_id ως πρωτεύον κλειδί, μοναδικό για κάθε εγγραφή στον πίνακα.

**FOREIGN KEY (hometeamid) REFERENCES Teams (t\_id):** Ορίζει το πεδίο hometeamid ως ξένο κλειδί που αναφέρεται στον πίνακα Teams και αντιπροσωπεύει την ομάδα που παίζει στον εντός έδρας αγώνα.

**FOREIGN KEY (awayteamid) REFERENCES Teams (t\_id):** Ορίζει το πεδίο awayteamid ως ξένο κλειδί που αναφέρεται στον πίνακα Teams και αντιπροσωπεύει την ομάδα που παίζει στον εκτός έδρας αγώνα.

**CONSTRAINT different\_teams CHECK (hometeamid != awayteamid):** Επιβάλλει τον περιορισμό ότι η ομάδα του σπιτιού και η ομάδα των εκτός έδρας αγώνων πρέπει να είναι διαφορετικές.

**CONSTRAINT CHECK hteam\_rested CHECK (IsRested(hometeamid,mdate) < 1):** Επιβάλλει τον περιορισμό ότι πρέπει η ομάδα να έχει κeno 10 ημερων πριν μπορεσει να παιξει.

**CONSTRAINT CHECK ateam\_rested CHECK (IsRested(awayteamid,mdate) < 1):** Επιβάλλει τον περιορισμό ότι πρέπει η ομάδα να έχει κeno 10 ημερων πριν μπορεσει να παιξει.

#### Πίνακας Goals:

**FOREIGN KEY (p\_id) REFERENCES Players (p\_id):** Ορίζει το πεδίο p\_id ως ξένο κλειδί που αναφέρεται στον πίνακα Players και συνδέει την καταχώριση με τον αντίστοιχο παίκτη.

**FOREIGN KEY (m\_id) REFERENCES Matches (m\_id):** Ορίζει το πεδίο m\_id ως ξένο κλειδί που αναφέρεται στον πίνακα Matches και συνδέει την καταχώριση με τον αντίστοιχο αγώνα.

Αντίστοιχα για τους πίνακες: BadGoals, Yellow\_cards, Red\_cards, Corners, Penalties

## Ερώτημα 1.c:



Η προβολή που δείχνει το πρόγραμμα αγώνων είναι:

```
CREATE VIEW Matches_Schedule AS
SELECT
    (SELECT stadium FROM Teams WHERE Matches.hometeamid=Teams.t_id) AS
    Match_Location,
    '90' AS Duration,
    Teams.t_name AS Team_Name,
    Matches.hscores AS Host_team_score,
    Matches.ascores AS away_team_score,
    Players.p_name AS players_names,
    Players.p_surname AS players_surnames,
    Players.pos AS Match_Position,
    (SELECT count() FROM yellow_cards WHERE yellow_cards.p_id=Players.p_id AND
    yellow_cards.m_id=Matches.m_id) AS yellow_cards,
    (SELECT count() FROM red_cards WHERE red_cards.p_id=Players.p_id AND
    red_cards.m_id=Matches.m_id) AS red_cards,
    Goals.time_stamp AS goal_time
FROM Matches
JOIN Teams ON Matches.hometeamid=Teams.t_id OR Matches.awayteamid=Teams.t_id
JOIN Players ON Teams.t_id=Players.t_id
LEFT JOIN Goals ON Goals.p_id=Players.p_id AND Goals.m_id=Matches.m_id
LEFT JOIN yellow_cards ON yellow_cards.p_id=Players.p_id AND
yellow_cards.m_id=Matches.m_id
LEFT JOIN red_cards ON red_cards.p_id=Players.p_id AND red_cards.m_id=Matches.m_id
WHERE Matches.mdate='2022-02-01'
ORDER BY Team_Name;
```

Η προβολή που δείχνει το αιτησιο πρωταθλημα αγωνων ειναι:

```
CREATE VIEW championship_schedule AS
SELECT DISTINCT
```

```
(SELECT stadium FROM Teams WHERE Matches.hometeamid=Teams.t_id) AS
Match_Location,
'90' AS duration,
home_team.t_name AS home_Team,
away_team.t_name AS away_team,
hscore AS home_team_score,
ascore AS away_team_Score
FROM Matches
JOIN Teams home_team ON Matches.hometeamid=home_team.t_id
JOIN Teams away_team ON Matches.awayteamid=away_team.t_id
WHERE Matches.mdate BETWEEN '2022/9/1' AND '2023/06/30'
ORDER BY Match_Location;
```

## Ερώτημα 2.α:

```
SELECT Coaches.c_name, Coaches.c_surname
FROM Matches
JOIN Teams ON (Matches.hometeamid = Teams.t_id OR Matches.awayteamid = Teams.t_id)
JOIN Coaches ON Teams.t_id = Coaches.t_id
WHERE Matches.m_id = 2
AND Teams.t_id = 1;
```

## Ερώτημα 2.β:

Τα γκολ που έγιναν σε συγκεκριμένο αγώνα, ποια χρονική στιγμή και από ποιόν παίκτη:

```
SELECT
Players.p_name,
```

```
Players.p_surname,  
Goals.time_stamp  
FROM  
Goals  
INNER JOIN Players ON Goals.p_id = Players.p_id  
INNER JOIN Matches ON Goals.m_id = Matches.m_id  
WHERE  
Matches.m_id = [ΙΔΑΓΩΝΑ];
```

Τα πέναλτι που έγιναν σε συγκεκριμένο αγώνα, ποια χρονική στιγμή και από ποιόν παίκτη:

```
SELECT  
Players.p_name,  
Players.p_surname,  
penalties.time_stamp  
FROM  
penalties  
INNER JOIN Players ON penalties.p_id = Players.p_id  
INNER JOIN Matches ON penalties.m_id = Matches.m_id  
WHERE  
Matches.m_id = [ΙΔΑΓΩΝΑ];
```

## Ερώτημα 2.c:

Η αγωνιστική εικόνα ενός συγκεκριμένου παίκτη για μια αγωνιστική σεζόν:  
γκολ, πέναλτι, κάρτες, λεπτά αγώνα, θέση που έπαιξε (με την σειρά):

```
SELECT COUNT(*) AS goals FROM Goals
```

```
JOIN Matches ON Goals.m_id = Matches.m_id
WHERE Goals.p_id = <player_id>
AND Matches.mdate BETWEEN <start_date> AND <end_date>;
```

```
SELECT COUNT(*) AS penalties FROM Penalties
JOIN Matches ON Penalties.m_id = Matches.m_id
WHERE Penalties.p_id = <player_id>
AND Matches.mdate BETWEEN <start_date> AND <end_date>;
```

```
SELECT COUNT(*) AS yellow_cards FROM yellow_cards
JOIN Matches ON yellow_cards.m_id = Matches.m_id
WHERE yellow_cards.p_id = <player_id>
AND Matches.mdate BETWEEN <start_date> AND <end_date>;
```

```
SELECT COUNT(*) AS red_cards FROM red_cards
JOIN Matches ON red_cards.m_id = Matches.m_id
WHERE red_cards.p_id = <player_id>
AND Matches.mdate BETWEEN <start_date> AND <end_date>;
```

```
SELECT COUNT(*) * 90 FROM Matches
JOIN Teams ON Matches.hometeamid=Teams.t_id OR Matches.awayteamid=Teams.t_id
JOIN Players ON Teams.t_id=Players.t_id
WHERE Players.p_id = {}
AND Matches.mdate BETWEEN <start_date> AND <end_date>;
```

```
SELECT pos FROM Players
WHERE p_id = <player_id>;
```

**Ερώτημα 2.d:**

```

SELECT COUNT(*) AS total_matches,
    SUM(CASE WHEN hometeamid=team_id THEN 1 ELSE 0 END)AS home_matches,
    SUM(CASE WHEN awayteamid=team_id THEN 1 ELSE 0 END)AS away_matches,
    SUM(CASE WHEN hometeamid=team_id AND hscore > ascore THEN 1 ELSE 0 END)AS home_wins,
    SUM(CASE WHEN hometeamid=team_id AND hscore < ascore THEN 1 ELSE 0 END)AS home_losses,
    SUM(CASE WHEN hometeamid=team_id AND hscore = ascore THEN 1 ELSE 0 END)AS home_draws,
    SUM(CASE WHEN awayteamid=team_id AND hscore < ascore THEN 1 ELSE 0 END)AS away_wins,
    SUM(CASE WHEN awayteamid=team_id AND hscore > ascore THEN 1 ELSE 0 END)AS away_losses,
    SUM(CASE WHEN awayteamid=team_id AND hscore = ascore THEN 1 ELSE 0 END)AS away_draws
FROM Matches
WHERE (awayteamid = team_id AND hometeamid = team_id)
AND mdate BETWEEN <start_date> AND <end_date>
AND (hscore IS NOT NULL or ascore IS NOT NULL);

```

### Ερώτημα 3.α:

Ο βοηθητικός πίνακας:

```

CREATE TABLE Teams_Deleted (
    t_id INT,
    t_name VARCHAR(50),
    stadium VARCHAR(50),
    description TEXT,
    hwins INT,
    awins INT,
    hlosses INT,
    alosses INT,
    hdraws INT,
    adraws INT
);

```

To trigger:

```

CREATE FUNCTION trigger_fun() RETURNS trigger AS $trigger_fun$

```

```
BEGIN
    INSERT INTO Teams_Deleted (t_id, t_name, stadium, description, hwins, awins, hlosses,
alosses, hdraws, adraws)
        VALUES (OLD.t_id, OLD.t_name, OLD.stadium, OLD.description, OLD.hwins, OLD.awins,
OLD.hlosses, OLD.alosses, OLD.hdraws, OLD.adraws);
    RETURN OLD;
END;
$trigger_fun$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_fun AFTER DELETE ON teams
    FOR EACH ROW EXECUTE FUNCTION trigger_fun();
```

**Ερώτημα 3.b:**

```

BEGIN;
DECLARE show_stats_cursor CURSOR WITH HOLD FOR
SELECT
    Matches.mdate AS match_date,
    players.p_name AS player_name,
    players.p_surname AS player_surname,
    Players.pos AS Match_Position,
    Goals.time_stamp AS goal_time,
    penalties.time_stamp AS penalty_time,
    (SELECT count() FROM yellow_cards WHERE yellow_cards.p_id=Players.p_id AND
yellow_cards.m_id=Matches.m_id) AS yellow_cards,
    (SELECT count() FROM red_cards WHERE red_cards.p_id=Players.p_id AND
red_cards.m_id=Matches.m_id) AS red_cards
FROM players
LEFT JOIN Teams ON Teams.t_id=Players.t_id
LEFT JOIN Matches ON Matches.hometeamid=Teams.t_id OR Matches.awayteamid=Teams.t_id
LEFT JOIN Goals ON Goals.p_id=Players.p_id AND Goals.m_id=Matches.m_id
LEFT JOIN penalties ON penalties.p_id=Players.p_id AND penalties.m_id=Matches.m_id
LEFT JOIN yellow_cards ON yellow_cards.p_id=Players.p_id AND
yellow_cards.m_id=Matches.m_id
LEFT JOIN red_cards ON red_cards.p_id=Players.p_id AND red_cards.m_id=Matches.m_id
ORDER BY Matches.mdate;
COMMIT;

FETCH 10 FROM test_cur;

```

## Ερώτημα 4:

Η λύση στο ερώτημα 4 βρίσκεται στο συνοδευτικό αρχείο python το οποίο ονομάζεται main.py