ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Βάσεις Δεδομένων

4ο εξάμηνο
2022-2023

Εργασία

Δημήτρης Στυλιανού Π20004
Παναγιώτα Νικολάου Π20009

Περιεχόμενα

Εκφώνηση

Εισαγωγή

Έστω η ΒΔ ενός τοπικού ερασιτεχνικού ποδοσφαιρικού συλλόγου στην οποία διατηρούνται πληροφορίες συμμετοχής των μελών του σε διάφορες αθλητικές εκδηλώσεις. Οι πληροφορίες αφορούν τους παίκτες, τους προπονητές, τις ομάδες, τους αγώνες-πρόγραμμα αγώνων κλπ. Πιο συγκεκριμένα:

· **Παίκτες:** Για τους παίκτες διατηρούνται πληροφορίες οι οποίες αφορούν το όνομα, επώνυμο, την ομάδα στην οποία ανήκουν, τη θέση στην οποία παίζουν (center back, goal keeper, defender, Center Back, Sweeper/Libero, Right Back, Left Back, κλπ.). Επιπλέον για κάθε παίκτη διατηρούνται συνολικά στατιστικά του με τις κάρτες που έχει λάβει (κίτρινες, κόκκινες κλπ.) καθώς και το συνολικό αριθμό από γκολ που έχει βάλει, συνολικά λεπτά που ήταν ενεργός στον αγώνα κλπ. Το όνομα και το επώνυμο μπορούν να λάβουν μόνον έως 10 χαρακτήρες ελληνικά με πλήρη στίξη (τόνους, διαλυτικά, κλπ.). Δεν θα πρέπει να περιλαμβάνονται περισσότεροι από 11 παίκτες σε κάθε ομάδα. Υπάρχουν και μεταγραφές, κατά συνέπεια ένας παίκτης δεν ανήκει για όλα τα χρόνια στην ίδια ομάδα.

· **Προπονητές:** Προπονητές στο σύλλογο μπορούν να γίνουν μόνον παλιοί παίκτες του συλλόγου. Οπότε για τους προπονητές διατηρούνται όλες οι πληροφορίες όπως και για τους παίκτες επιπλέον της προπονητικής τους ιδιότητας στην όποια ομάδα του συλλόγου.

· **Ομάδες:** Για τις ομάδες διατηρούνται πληροφορίες οι οποίες αφορούν το όνομα τους, το γήπεδο της έδρας της, κάποια περιγραφή της ιστορίας τους, καθώς και διάφορα στατιστικά όπως: νίκες εντός/εκτός έδρας, ήττες εντός/εκτός έδρας, ισοπαλίες εντός/εκτός έδρας.

· **Αγώνες/πρόγραμμα αγώνων:** Για κάθε αγώνα διατηρούνται πληροφορίες όπως ποια είναι η γηπεδούχος και ποια η φιλοξενούμενη ομάδα, ποιο το σκορ της κάθε ομάδας, ποια η ημερομηνία που έγινε ο αγώνας. Επιπλέον θα πρέπει να γίνεται έλεγχος ώστε να μην προγραμματίζονται αγώνες με τις ίδιες ομάδες την ίδια μέρα. Για κάθε ομάδα θα πρέπει να υπάρχει διάστημα 10 ημερών μεταξύ των αγώνων της. Για κάθε αγώνα και για κάθε παίκτη διατηρούνται πληροφορίες όπως τα γκολ που μπήκαν, τα γκολ που ακυρώθηκαν, οι κάρτες (κόκκινες και κίτρινες) που δέχτηκε ένας παίκτης, τα πέναλτι, τα κόρνερ (και σε όλα αυτά, η χρονική στιγμή που συνέβησαν).

Ερώτημα 1 (40%). Σχεσιακή Βάση Δεδομένων

a. Με βάση τα παραπάνω στοιχεία, σχεδιάστε το σχεσιακό σχήμα της ΒΔ, υλοποιήστε το (εντολές CREATE TABLE) στο ΣΔΒΔ PostgreSQL και φορτώστε με δεδομένα τους πίνακες. Ενδεχομένως να χρειαστεί να υλοποιήσετε επιπλέον βοηθητικούς πίνακες, σε σχέση με αυτούς οι οποίοι περιγράφονται στην εισαγωγή. Επιπλέον, καλείστε να τεκμηριώσετε τους περιορισμούς ακεραιότητας των πινάκων (και να δηλώσετε τυχόν περιορισμούς που προκύπτουν από την εκφώνηση αλλά δεν μπορέσατε να υποστηρίξετε μέσα από τους περιορισμούς ακεραιότητας των πινάκων). Το παραδοτέο του υποερωτήματος είναι το σχεσιακό σχήμα της ΒΔ, οι εντολές CREATE TABLE και τα αρχεία τα οποία θα εισάγετε στους πίνακες. Οδηγία: για την ευκολότερη παραγωγή αληθοφανών δεδομένων προτείνεται να χρησιμοποιήσετε κάποιο εργαλείο παραγωγής δεδομένων (data generator) (π.χ. www.mockaroo.com, https://faker.readthedocs.io/en/master/, https://devskiller.github.io/jfairy/).

b. ~~Εφαρμόστε τη θεωρία της κανονικοποίησης πάνω στο σχεσιακό σχήμα της ΒΔ που σχεδιάσατε και ελέγξτε τον κάθε πίνακα εάν ακολουθεί την BCNF. Σε αντίθετη περίπτωση, αποσυνθέστε τους προβληματικούς πίνακες ώστε όλη η ΒΔ να είναι σε BCNF.~~ (Αφαιρείται από την εκφώνηση)

c. Πάνω στο τελικό σχήμα της ΒΔ υλοποιήστε 2 προβολές/όψεις (views):

· **Πρόγραμμα-αγώνων.** Μια προβολή που θα αφορά μια συγκεκριμένη ημερομηνία (π.χ. 30/5/2023) και θα περιλαμβάνει τις «δυναμικές» πληροφορίες των αγώνων εκείνης της ημέρας: τόπος διεξαγωγής αγώνα, χρόνος, ποιες ομάδες συμμετέχουν, ποιο το σκορ, ποιοι παίκτες από κάθε ομάδα (όνομα θέση, στο παιχνίδι, χρόνος συμμετοχής στο παιχνίδι, τις κάρτες που τυχόν χρεώθηκε, τα γκολ που έβαλε και πότε τα έβαλε).

· **Ετήσιο πρωτάθλημα αγώνων.** Μια προβολή που θα αφορά μια συγκεκριμένη αγωνιστική σεζόν (π.χ. 1/9/2022 – 30/6/2023) και θα περιλαμβάνει τις «στατικές» πληροφορίες των αγώνων εκείνου του διαστήματος: τόπος διεξαγωγής αγώνα, χρόνος, ποιες ομάδες συμμετέχουν, ποιο το σκορ μεταξύ τους, ποια ομάδα είναι εντός/εκτός έδρας.

Ερώτημα 2 (20%). Εκτελέστε τις παρακάτω ερωτήσεις (queries) στη ΒΔ (εντολές SELECT).

a) Ποιος είναι προπονητής μιας συγκεκριμένης ομάδας σε συγκεκριμένο αγώνα;

b) Τα γκολ, πέναλτι που έγιναν σε συγκεκριμένο αγώνα, ποια χρονική στιγμή και από ποιόν παίκτη.

c) Την αγωνιστική εικόνα ενός συγκεκριμένου παίκτη για μια αγωνιστική σεζόν: γκολ, πέναλτι, κάρτες, λεπτά αγώνα, θέση που έπαιξε.

d) Την αγωνιστική εικόνα μιας συγκεκριμένης ομάδας για μια αγωνιστική σεζόν: σε πόσους αγώνες συμμετείχε, σε πόσους ήταν γηπεδούχος και σε πόσους φιλοξενούμενη, πόσες ήττες /νίκες/ ισοπαλίες, πόσες φορές νίκησε/ έχασε/ έφερε ισοπαλία εντός/ εκτός έδρας.

Ερώτημα 3 (20%). Υλοποίηση triggers, cursors

a. Φτιάξτε έναν trigger ο οποίος κρατά/γεμίζει ένα πίνακα-ιστορικό. Όταν διαγράφονται με επιτυχία γραμμές από τον πίνακα ομάδες (π.χ. διαγράφονται όλες οι ομάδες οι οποίες δεν πέτυχαν καμία νίκη μέσα στο έτος) τότε οι διαγραμμένες γραμμές εισάγονται αυτόματα στον πίνακα ομάδες-υποβιβασμός-κατηγορίας.

b. Βρείτε για κάθε παίκτη ομαδοποιημένα ανά χρονικά διαστήματα και ανά ομάδα και ανά αγώνα τα: γκολ, πέναλτι, κάρτες, λεπτά αγώνα, θέση που έπαιξε. Χρησιμοποιείστε cursors ώστε να εμφανίσετε τις γραμμές σε ομάδες των 10.

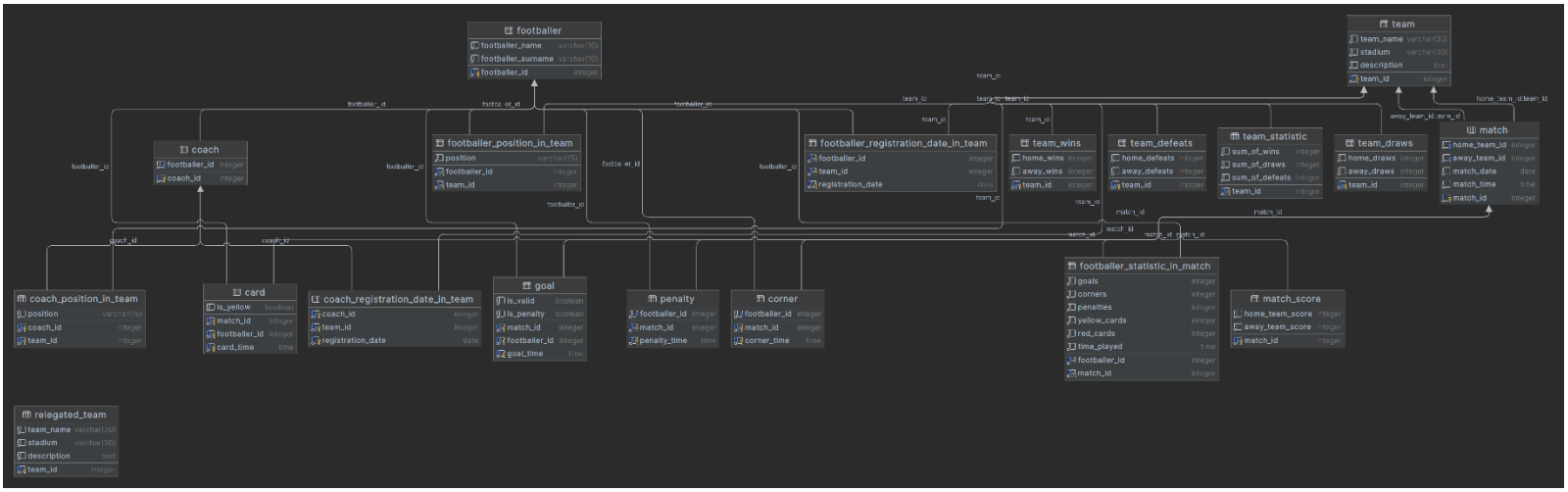
Ερώτημα 4 (20%). Σύνδεση ΒΔ με Application Programming Interface (API)

Υλοποιήστε προγραμματιστικά έναν client σε οποιαδήποτε γλώσσα προγραμματισμού γνωρίζετε (π.χ.

Python, Java, C) χρησιμοποιώντας την κατάλληλη βιβλιοθήκη σύνδεσης με την PostgreSQL (π.χ. psycopg2, JDBC, ODBC). Ο client θα συνδέεται στο ΣΔΒΔ της PostgreSQL, θα εκτελεί τα queries του Ερωτήματος 2, και θα εμφανίζει τα αποτελέσματα στον χρήστη (είτε σε terminal είτε με γραφικά).

Ερώτημα 1

Σχεσιακό σχήμα της Βάσης Δεδομένων



Εντολές CREATE TABLE

```
create table team
(
    team_id     integer     not null unique,
    team_name   varchar(30) not null,
    stadium     varchar(30) not null,
    description text        not null,
    primary key (team_id)
);

create table footballer
(
    footballer_id      integer     not null unique,
    footballer_name    varchar(10) not null check (footballer_name ~ '^[A-Ωα-
ωΆ-Ώά-ώΫϋΰΪϊΐ]+$'),
    footballer_surname varchar(10) not null check (footballer_surname ~ '^[A-
Ωα-ωΆ-Ώά-ώΫϋΰΪϊΐ]+$'),
    primary key (footballer_id)
);

create table footballer_registration_date_in_team
(
    footballer_id     integer not null,
```

```
    team_id           integer not null,
    registration_date date    not null,
    primary key (footballer_id, team_id, registration_date),
    foreign key (footballer_id) references footballer (footballer_id),
    foreign key (team_id) references team (team_id)
);

create table footballer_position_in_team
(
    footballer_id integer     not null,
    team_id       integer     not null,
    position      varchar(15) not null check (position in ('goalkeeper',
'defender', 'midfielder', 'forward')),
    primary key (footballer_id, team_id),
    foreign key (footballer_id) references footballer (footballer_id),
    foreign key (team_id) references team (team_id)
);

create table coach
(
    coach_id      integer not null unique,
    footballer_id integer not null,
    primary key (coach_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table coach_registration_date_in_team
(
    coach_id          integer not null,
    team_id           integer not null,
    registration_date date    not null,
    primary key (coach_id, team_id, registration_date),
    foreign key (coach_id) references coach (coach_id),
    foreign key (team_id) references team (team_id)

);

create table coach_position_in_team
(
    coach_id integer     not null,
    team_id  integer     not null,
    position varchar(15) not null check (position in
                                         ('head', 'assistant', 'goalkeeping',
'fitness', 'technical', 'scout',
                                          'youth')),
    primary key (coach_id, team_id),
    foreign key (coach_id) references coach (coach_id),
    foreign key (team_id) references team (team_id)
);

create table match
(
```

```
    match_id     integer not null unique,
    home_team_id integer not null,
    away_team_id integer not null,
    match_date   date    not null,
    match_time   time    not null,
    primary key (match_id),
    foreign key (home_team_id) references team (team_id),
    foreign key (away_team_id) references team (team_id)
);

create table match_score
(
    match_id        integer not null unique,
    home_team_score integer not null,
    away_team_score integer not null,
    primary key (match_id),
    foreign key (match_id) references match (match_id)
);

create table team_wins
(
    team_id   integer not null unique,
    home_wins integer not null,
    away_wins integer not null,
    primary key (team_id),
    foreign key (team_id) references team (team_id)
);

create table team_draws
(
    team_id    integer not null unique,
    home_draws integer not null,
    away_draws integer not null,
    primary key (team_id),
    foreign key (team_id) references team (team_id)
);

create table team_defeats
(
    team_id      integer not null unique,
    home_defeats integer not null,
    away_defeats integer not null,
    primary key (team_id),
    foreign key (team_id) references team (team_id)
);

create table team_statistic
(
    team_id        integer not null unique,
    sum_of_wins    integer not null,
    sum_of_draws   integer not null,
    sum_of_defeats integer not null,
```

```
    primary key (team_id),
    foreign key (team_id) references team (team_id)
);

create table goal
(
    match_id      integer not null,
    footballer_id integer not null,
    goal_time     time    not null,
    is_valid      boolean not null,
    is_penalty    boolean not null,
    primary key (match_id, footballer_id, goal_time),
    foreign key (match_id) references match (match_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table corner
(
    match_id      integer not null,
    footballer_id integer not null,
    corner_time   time    not null,
    primary key (match_id, corner_time),
    foreign key (match_id) references match (match_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table penalty
(
    match_id      integer not null,
    footballer_id integer not null,
    penalty_time  time    not null,
    primary key (match_id, penalty_time),
    foreign key (match_id) references match (match_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table card
(
    match_id      integer not null,
    footballer_id integer not null,
    card_time     time    not null,
    is_yellow     boolean not null,-- true -> yellow, false -> red
    primary key (match_id, footballer_id, card_time),
    foreign key (match_id) references match (match_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table footballer_statistic_in_match
(
    footballer_id integer not null,
    match_id      integer not null,
    goals         integer not null,
```

```
    corners       integer not null,
    penalties     integer not null,
    yellow_cards  integer not null,
    red_cards     integer not null,
    time_played   time    not null,
    primary key (match_id, footballer_id),
    foreign key (match_id) references match (match_id),
    foreign key (footballer_id) references footballer (footballer_id)
);

create table relegated_team
(
    team_id     integer     not null unique,
    team_name   varchar(30) not null,
    stadium     varchar(30) not null,
    description text        not null,
    primary key (team_id)
);
```

Προβολές/όψεις (views)

```
create or replace view match_statistic_for_a_specific_date as
with current_match as (select match_id,
                              home_team_id,
                              away_team_id,
                              match_date,
                              match_time
                       from footballclub_db.public.match
                       where match_date = '2023-05-05'),

     home_team as (select cm.match_id, t.team_name as home_team_name
                   from footballclub_db.public.team as t
                            inner join current_match as cm
                                       on t.team_id = cm.home_team_id),

     away_team as (select cm.match_id, t.team_name as away_team_name
                   from footballclub_db.public.team as t
                            inner join current_match as cm
                                       on t.team_id = cm.away_team_id),

     match_stadium as (select cm.match_id, t.stadium
                       from footballclub_db.public.team as t
                                inner join current_match as cm
                                           on t.team_id = cm.home_team_id),

     match_score as (select cm.match_id, ms.home_team_score,
ms.away_team_score
                     from footballclub_db.public.match_score as ms
                              inner join current_match as cm
                                         on ms.match_id = cm.match_id),
```

```
     footballer_stat as (select fsm.match_id,
                                fsm.footballer_id,
                                f.footballer_name,
                                f.footballer_surname,
                                f.position,
                                fsm.time_played,
                                fsm.yellow_cards,
                                fsm.red_cards
                         from (footballclub_db.public.footballer natural join
footballclub_db.public.footballer_position_in_team) as f
                                  inner join
(footballclub_db.public.footballer_statistic_in_match natural join
current_match) as fsm
                                                      on f.footballer_id =
fsm.footballer_id),

     goal as (select g.match_id,
                     g.footballer_id,
                     g.goal_time
              from footballclub_db.public.goal as g
                       natural join footballer_stat as fs
              where g.is_valid = true)

select cm.match_id,
       ht.home_team_name,
       at.away_team_name,
       ms.stadium,
       cm.match_date,
       cm.match_time,
       msc.home_team_score,
       msc.away_team_score,
       fs.footballer_id,
       fs.footballer_name,
       fs.footballer_surname,
       fs.position,
       fs.time_played,
       fs.yellow_cards,
       fs.red_cards,
       g.goal_time
from current_match as cm
         inner join home_team as ht
                    on cm.match_id = ht.match_id
         inner join away_team as at
                    on cm.match_id = at.match_id
         inner join match_stadium as ms
                    on cm.match_id = ms.match_id
         inner join match_score as msc
                    on cm.match_id = msc.match_id
         inner join footballer_stat as fs
                    on cm.match_id = fs.match_id
         full outer join goal as g
```

```
                        on cm.match_id = g.match_id and fs.footballer_id =
g.footballer_id;


create or replace view league_statistic_for_a_specific_season as
with current_match as (select match_id,
                              home_team_id,
                              away_team_id,
                              match_date,
                              match_time
                       from footballclub_db.public.match
                       where match_date >= '2023-01-01'
                         and match_date <= '2023-12-31'),

     home_team as (select cm.match_id, t.team_name as home_team_name
                   from footballclub_db.public.team as t
                            inner join current_match as cm
                                       on t.team_id = cm.home_team_id),

     away_team as (select cm.match_id, t.team_name as away_team_name
                   from footballclub_db.public.team as t
                            inner join current_match as cm
                                       on t.team_id = cm.away_team_id),

     match_stadium as (select cm.match_id, t.stadium
                       from footballclub_db.public.team as t
                                inner join current_match as cm
                                           on t.team_id = cm.home_team_id),

     match_score as (select cm.match_id, ms.home_team_score,
ms.away_team_score
                     from footballclub_db.public.match_score as ms
                              inner join current_match as cm
                                         on ms.match_id = cm.match_id)

select cm.match_id,
       ht.home_team_name,
       at.away_team_name,
       ms.stadium,
       cm.match_date,
       cm.match_time,
       msc.home_team_score,
       msc.away_team_score
from current_match as cm
         inner join home_team as ht
                    on cm.match_id = ht.match_id
         inner join away_team as at
                    on cm.match_id = at.match_id
         inner join match_stadium as ms
                    on cm.match_id = ms.match_id
         inner join match_score as msc
                    on cm.match_id = msc.match_id;
```

Ερώτημα 2

```
a)
query =
select t3.coach_id, t4.footballer_name, t4.footballer_surname
from (select home_team_id as home, away_team_id as away
      from match
      where match_id = %s
        and (home_team_id = %s or away_team_id = %s)) as t1
         inner join coach_registration_date_in_team as t2
                    on t2.team_id = %s and (t2.team_id = t1.home or t2.team_id
= t1.away)
         inner join coach as t3 on t3.coach_id = t2.coach_id
         inner join footballer as t4 on t4.footballer_id = t3.footballer_id;

b)
goal_query =
select t1.goal_time, t1.is_valid, t2.footballer_id, t2.footballer_name,
t2.footballer_surname
from goal as t1
         natural join footballer as t2
where t1.match_id = %s;

penalty_query =
select t1.penalty_time, t2.footballer_id, t2.footballer_name,
t2.footballer_surname
from penalty as t1
         natural join footballer as t2
where t1.match_id = %s;

c)
query =
select t1.goals, t1.penalties, t1.yellow_cards, t1.red_cards,
t1.time_played, t2.position
from footballer_statistic_in_match as t1
         natural join footballer_position_in_team as t2
where t1.footballer_id = %s;

d)
query =
select sum(t1.home_wins + t1.away_wins +
           t2.home_defeats + t2.away_defeats +
           t3.home_draws + t3.away_draws),
       sum(t1.home_wins + t2.home_defeats + t3.home_draws),
       sum(t1.away_wins + t2.away_defeats + t3.away_draws)
from team_wins as t1
         natural join team_defeats as t2
```

```
        natural join team_draws as t3
where team_id = %s;
```

Τα παραπάνω ερωτήματα (SQL queries) είναι γραμμένα σε python και βρίσκονται μέσα στο αρχείο "main.py". Όπου '%s' αντιστοιχεί στα δεδομένα εισόδου που ζητούνται από τον χρήστη.

Ερώτημα 3

```
a)
-- functions returning trigger
create or replace function team_relegation() returns trigger as
$$
begin
    insert into footballclub_db.public.relegated_team
    values (old.team_id, old.team_name, old.stadium, old.description);
    return old;
end;
$$ language plpgsql;
-- triggers creation
create or replace trigger team_relegation
    after delete
    on footballclub_db.public.team
    for each row
execute function team_relegation();
```

Ερώτημα 4

Για την εκτέλεση των queries του **ερωτήματος 2**, γράφτηκε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο θα βρείτε στον παρακάτω σύνδεσμο [dimitrisstyl7/databases_project_2023: Εργασία Βάσεων Δεδομένων 2023 / Databases Project 2023 (github.com).]

Παραδοχές

- Κάθε σεζόν διαρκεί έναν χρόνο.
- Για να γίνει προπονητής ένας παίκτης πρέπει απλά να ήταν παλιό μέλος του συλλόγου (πρέπει να έχει περάσει τουλάχιστον μια σεζόν).
- Ένας παίκτης μπορεί να έχει μόνο μια θέση σε μια ομάδα για την τρέχον σεζόν.
- Ένας προπονητής μπορεί να έχει μόνο μια θέση σε μια ομάδα για την τρέχον σεζόν.