ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΡΓΑΣΙΑ 2023

ΓΕΩΡΓΑΝΤΖΗ ΠΑΝΑΓΙΩΤΑ Π20039
ΔΟΣΙΑΔΗΣ ΙΩΑΝΝΗΣ Π20059
ΣΠΥΡΟΠΟΥΛΟΥ ΚΑΝΕΛΛΑ Π20179

Ερώτημα 1

α)

coaches

```
CREATE TABLE IF NOT EXISTS public.coaches
(
    coach_id integer NOT NULL DEFAULT
nextval('coaches_coach_id_seq'::regclass),
    first_name character varying(50) COLLATE pg_catalog."default"
NOT NULL,
    last_name character varying(50) COLLATE pg_catalog."default" NOT
NULL,
    team_id integer,
    coaching_position character varying(50) COLLATE
pg_catalog."default",
    CONSTRAINT coaches_pkey PRIMARY KEY (coach_id),
    CONSTRAINT coaches_team_id_fkey FOREIGN KEY (team_id)
        REFERENCES public.teams (team_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.coaches
   OWNER to postgres;
```

Εισαγωγή Προπονητών

```
\i /Users/gdosi/Downloads/coaches.sql
```

matches

```
CREATE TABLE IF NOT EXISTS public.matches
(
    match_id integer NOT NULL,
    home_team character varying(50) COLLATE pg_catalog."default",
    away_team character varying(50) COLLATE pg_catalog."default",
    home_score integer,
    away_score integer,
    match_date date,
    stadium character varying COLLATE pg_catalog."default",
    CONSTRAINT matches_pkey PRIMARY KEY (match_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.matches
    OWNER to postgres;
```

Εισαγωγή αγώνων

```
\i /Users/gdosi/Downloads/matches.sql
```

players

```
CREATE TABLE IF NOT EXISTS public.players
(
    player_id integer NOT NULL,
    first_name character varying(50) COLLATE pg_catalog."default"
NOT NULL,
    last_name character varying(50) COLLATE pg_catalog."default" NOT
NULL,
    team_id integer,
    "position" character varying(50) COLLATE pg_catalog."default",
    yellow_cards integer,
    red_cards integer,
    goals_scored integer,
    minutes_played integer,
    CONSTRAINT players_pkey PRIMARY KEY (player_id),
    CONSTRAINT team_id_fk FOREIGN KEY (team_id)
        REFERENCES public.teams (team_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.players
    OWNER to postgres;
```

Εισαγωγή παικτών
```
\i /Users/gdosi/Downloads/players.sql
```

teams

```
CREATE TABLE IF NOT EXISTS public.teams
(
    team_id integer NOT NULL DEFAULT
nextval('teams_team_id_seq'::regclass),
    name character varying(50) COLLATE pg_catalog."default" NOT
NULL,
    stadium character varying(50) COLLATE pg_catalog."default",
    history text COLLATE pg_catalog."default",
    home_wins integer,
    away_wins integer,
    home_losses integer,
    away_losses integer,
    home_draws integer,
    away_draws integer,
    CONSTRAINT teams_pkey PRIMARY KEY (team_id)
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.teams
    OWNER to postgres;
```

Εισαγωγή παικτών
```
\i /Users/gdosi/Downloads/teams.sql
```

matchstats

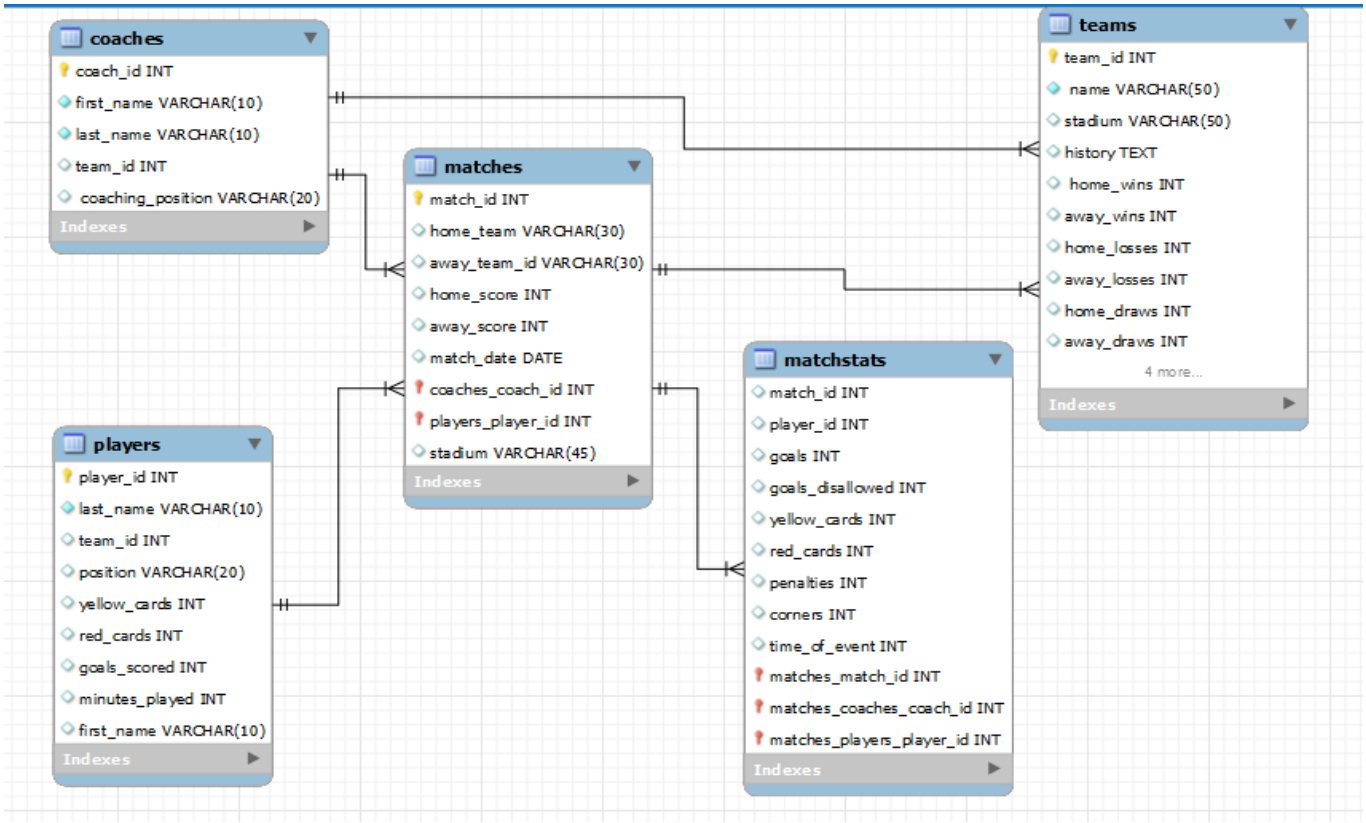
```
CREATE TABLE IF NOT EXISTS public.matchstats
(
    match_id integer,
    player_id integer,
    goals integer,
    goals_disallowed integer,
    yellow_cards integer,
    red_cards integer,
    penalties integer,
    corners integer,
    time_of_event timestamp without time zone,
    CONSTRAINT matchstats_match_id_fkey FOREIGN KEY (match_id)
        REFERENCES public.matches (match_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
```

```
        NOT VALID,
    CONSTRAINT matchstats_player_id_fkey FOREIGN KEY (player_id)
        REFERENCES public.players (player_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE IF EXISTS public.matchstats
    OWNER to postgres;
ALTER TABLE IF EXISTS public.matchstats
    OWNER to postgres;
```

Εισαγωγή στατιστικών αγώνων

```
\i /Users/gdosi/Downloads/matchstats.sql
```

Σχεσιακό σχήμα της ΒΔ



C. Πρόγραμμα-αγώνων

```
CREATE VIEW game_schedule AS
SELECT m.match_id, m.match_date, m.stadium, m.home_team,
```

```
m.away_team, m.home_score, m.away_score,
       p.player_id, p.first_name || ' ' || p.last_name AS
player_name, p.position, ms.goals, ms.goals_disallowed,
       ms.yellow_cards, ms.red_cards, ms.penalties, ms.corners,
ms.time_of_event
FROM matches m
LEFT JOIN matchstats ms ON m.match_id = ms.match_id
LEFT JOIN players p ON ms.player_id = p.player_id
WHERE m.match_date = '31-05-2023';
```

```
1  SELECT *
2  FROM game_schedule;
3
4
5
6
7
8
```

Data Output | Messages | Notifications

	match_id integer	match_date date	stadium character varying	home_team character varying	away_team character varying	home_score integer	away_score integer
1	6	2023-05-31	SKORPIOS ARENA	AKRATITOS	IKAROS	0	5
2	6	2023-05-31	SKORPIOS ARENA	AKRATITOS	IKAROS	0	5
3	685	2023-05-31	DIMOTIKO STADIO MEGARWN	ZOFRIA	AGIA ANNA	3	3

Ετήσιο πρωτάθλημα αγώνων

```
CREATE VIEW championship_schedule AS
SELECT m.match_id,
       m.home_team AS home_team_id,
       m.away_team AS away_team_id,
       m.home_score,
       m.away_score,
       m.match_date,
       m.stadium,
       t1.name AS home_team_name,
       t2.name AS away_team_name,
       t1.stadium AS home_team_stadium,
       t2.stadium AS away_team_stadium
FROM matches m
JOIN teams t1 ON CAST(m.home_team AS integer) = t1.team_id
JOIN teams t2 ON CAST(m.away_team AS integer) = t2.team_id
WHERE m.match_date BETWEEN '01-01-2023' AND '30-06-2023';
```

Ερώτημα 2

a) Ποιος είναι προπονητής μιας συγκεκριμένης ομάδας σε συγκεκριμένο

αγώνα;

```
SELECT
    c.first_name || ' ' || c.last_name AS coach_name
FROM
    public.matches m
    JOIN public.teams t ON m.home_team = t.name OR m.away_team =
t.name
    JOIN public.coaches c ON t.team_id = c.team_id
WHERE
    m.match_id = 17
    AND t.name = 'ZOFRIA';
```

```
1  SELECT
2      c.first_name || ' ' || c.last_name AS coach_name
3  FROM
4      public.matches m
5      JOIN public.teams t ON m.home_team = t.name OR m.away_team = t.name
6      JOIN public.coaches c ON t.team_id = c.team_id
7  WHERE
8      m.match_id = 17
9      AND t.name = 'ZOFRIA';
10
```

Data Output | Messages | Notifications

	coach_name text
1	Lefteris Papadimitriou
2	Ioannis Panagiotakis
3	Thanasis Papadopou...

β)Τα γκολ, πέναλτι που έγιναν σε συγκεκριμένο αγώνα, ποια χρονική στιγμή και από ποιον παίκτη;

```
SELECT ms.time_of_event, p.last_name, p.first_name
FROM matchstats ms
JOIN players p ON ms.player_id = p.player_id
WHERE ms.match_id = 233 AND (ms.goals > 0 OR ms.penalties > 0)
```

```sql
SELECT ms.time_of_event, p.last_name, p.first_name
FROM matchstats ms
JOIN players p ON ms.player_id = p.player_id
WHERE ms.match_id = 233 AND (ms.goals > 0 OR ms.penalties > 0)
```

Data Output | Messages | Notifications

	time_of_event integer	last_name character varying	first_name character varying
1	16	Sotiriou	Sotiris

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

```sql
SELECT ms.goals, ms.penalties, ms.yellow_cards, ms.red_cards,
p.minutes_played, p.position
FROM matches m
JOIN matchstats ms ON m.match_id = ms.match_id
JOIN players p ON p.player_id = ms.player_id
WHERE p.player_id = 50
```

```sql
SELECT ms.goals, ms.penalties, ms.yellow_cards, ms.red_cards, p.minutes_played, p
FROM matches m
JOIN matchstats ms ON m.match_id = ms.match_id
JOIN players p ON p.player_id = ms.player_id
WHERE p.player_id = 50
```

Data Output | Messages | Notifications

	goals integer	penalties integer	yellow_cards integer	red_cards integer	minutes_played integer	position character varying
1	0	1	0	0	1104	Offensive Midfielder

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.

```
SELECT
    COUNT(*) AS total_matches,
    SUM(CASE WHEN home_team = 'PANORAMA' THEN 1 ELSE 0 END) AS
home_matches,
    SUM(CASE WHEN away_team = 'PANORAMA' THEN 1 ELSE 0 END) AS
away_matches,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score >
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score >
home_score THEN 1
             ELSE 0 END) AS wins,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score <
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score <
home_score THEN 1
             ELSE 0 END) AS losses,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score =
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score =
home_score THEN 1
             ELSE 0 END) AS draws,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score >
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score >
home_score THEN 1
             ELSE 0 END) AS home_wins,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score <
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score <
home_score THEN 1
             ELSE 0 END) AS away_losses,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score =
away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score =
home_score THEN 1
             ELSE 0 END) AS draws,
    SUM(CASE WHEN home_team = 'PANORAMA' THEN home_score
             WHEN away_team = 'PANORAMA' THEN away_score
             ELSE 0 END) AS goals_scored,
    SUM(CASE WHEN home_team = 'PANORAMA' THEN away_score
             WHEN away_team = 'PANORAMA' THEN home_score
             ELSE 0 END) AS goals_conceded
FROM
    public.matches
WHERE
    ('PANORAMA' IN (home_team, away_team))
```

```
    AND match_date >= '01-01-2023'
    AND match_date <= '30-06-2023';
```

```
SELECT
    COUNT(*) AS total_matches,
    SUM(CASE WHEN home_team = 'PANORAMA' THEN 1 ELSE 0 END) AS home_matches,
    SUM(CASE WHEN away_team = 'PANORAMA' THEN 1 ELSE 0 END) AS away_matches,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score > away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score > home_score THEN 1
             ELSE 0 END) AS wins,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score < away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score < home_score THEN 1
             ELSE 0 END) AS losses,
    SUM(CASE WHEN home_team = 'PANORAMA' AND home_score = away_score THEN 1
             WHEN away_team = 'PANORAMA' AND away_score = home_score THEN 1
```

Data Output | Messages | Notifications

	total_matches bigint	home_matches bigint	away_matches bigint	wins bigint	losses bigint	draws bigint	home_wins bigint	away_losses bigint	draws bigint
1	16	8	8	9	7	0	9	7	0

Ερώτημα 3

A. Αρχικά δημιουργήσαμε τον πίνακα relegation_teams

```
CREATE TABLE IF NOT EXISTS relegation_teams
(
    team_id integer NOT NULL,
    name character varying(50) COLLATE pg_catalog."default" NOT
NULL,
    stadium character varying(50) COLLATE pg_catalog."default",
    history text COLLATE pg_catalog."default",
    home_wins integer,
    away_wins integer,
    home_losses integer,
    away_losses integer,
    home_draws integer,
    away_draws integer,
    CONSTRAINT relegation_teams_pkey PRIMARY KEY (team_id)
)
TABLESPACE pg_default;

ALTER TABLE IF EXISTS relegation_teams
    OWNER to postgres;
```

Και στη συνέχεια δημιουργήσαμε τον trigger

```
CREATE OR REPLACE FUNCTION teams_delete_trigger()
RETURNS TRIGGER AS $$
```

```
BEGIN
    -- Έλεγχος αν η διαγραφή ικανοποιεί τις συνθήκες
    IF OLD.home_wins = 0 AND OLD.away_wins = 0 THEN
        -- Εισαγωγή της διαγραμμένης γραμμής στον πίνακα ομάδων
υποβιβασμού
        INSERT INTO relegation_teams (team_id, name, stadium,
history, home_wins, away_wins, home_losses, away_losses,
home_draws, away_draws)
        VALUES (OLD.team_id, OLD.name, OLD.stadium,
OLD.history, OLD.home_wins, OLD.away_wins, OLD.home_losses,
OLD.away_losses, OLD.home_draws, OLD.away_draws);
    END IF;

    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

-- Δημιουργία του trigger
CREATE TRIGGER teams_delete_trigger
AFTER DELETE ON teams
FOR EACH ROW
EXECUTE FUNCTION teams_delete_trigger();
```

β.

```
-- Δημιουργία τύπου για τα αποτελέσματα του cursor
CREATE TYPE player_stats AS (
    first_name character varying(50),
    last_name character varying(50),
    match_date date,
    team_name character varying(50),
    goals integer,
    penalties integer,
    yellow_cards integer,
    red_cards integer,
    time_of_event integer,
    position character varying(50)
);

-- Δημιουργία συνάρτησης
CREATE OR REPLACE FUNCTION get_player_stats()
  RETURNS SETOF player_stats AS $$
DECLARE
  my_cursor CURSOR FOR
    SELECT
        p.first_name,
        p.last_name,
        m.match_date,
        t.name AS team_name,
        ms.goals,
        ms.penalties,
```

```
        ms.yellow_cards,
        ms.red_cards,
        ms.time_of_event,
        p.position
    FROM
        public.players p
        JOIN public.matchstats ms ON p.player_id = ms.player_id
        JOIN public.matches m ON ms.match_id = m.match_id
        JOIN public.teams t ON p.team_id = t.team_id
    ORDER BY
        p.player_id, m.match_date;
  my_record player_stats;
BEGIN
  OPEN my_cursor;
  LOOP
    FETCH my_cursor INTO my_record;
    EXIT WHEN NOT FOUND;
    RETURN NEXT my_record;
  END LOOP;
  CLOSE my_cursor;
  RETURN;
END;
$$ LANGUAGE plpgsql;
```

Query | Query History | Scratch Pad

```
1  SELECT * FROM get_player_stats();
2
3
4
5
6
```

Data Output | Messages | Notifications

	first_name character varying	last_name character varying	match_date date	team_name character varying	goals integer	penalties integer	yellow_cards integer	red_cards integer	time_o integer
1	Ioannis	Georgiou	2023-02-14	SKORPIOS FILIS	0	0	0	0	
2	Petros	Dimitriadis	2023-06-13	SKORPIOS FILIS	0	1	0	0	
3	Kiriakos	Kostopoulos	2023-04-26	SKORPIOS FILIS	0	1	0	0	
4	Stavros	Apostolou	2023-01-03	SKORPIOS FILIS	1	0	0	0	
5	Nikolas	Aggelis	2023-05-14	PANORAMA	0	0	0	0	
6	Petros	Kalantzis	2023-04-14	PANORAMA	0	0	0	1	
7	Antonis	Sotiropoulos	2023-06-12	PANORAMA	0	0	0	0	
8	Petros	Aggelis	2023-01-10	PANORAMA	1	0	0	0	
9	Nikos	Papakostas	2023-05-31	AKRATITOS	0	0	0	1	
10	Ioannis	Papageorgiou	2023-02-03	AKRATITOS	0	0	0	0	

ΕΡΩΤΗΜΑ 4

α)

```
import psycopg2

conn = psycopg2.connect(
    host="localhost", import psycopg2

conn = psycopg2.connect(
    host="localhost",
    database="football",
    user="postgres",
    password="2110"
)

def get_coaches_of_match(match_id, team_name):
    with conn.cursor() as cursor:
        query = "SELECT c.first_name || ' ' || c.last_name AS coach_name " \
                "FROM public.matches m " \
                "JOIN public.teams t ON m.home_team = t.name OR m.away_team = t.name " \
                "JOIN public.coaches c ON t.team_id = c.team_id " \
                "WHERE m.match_id = %s AND t.name = %s"
        cursor.execute(query, (match_id, team_name))
        results = cursor.fetchall()
        if results:
            coaches = [result[0] for result in results]
            return "Coaches: " + ", ".join(coaches)
        else:
            return "No coaches found for the given match."

match_id = 52
team_name = "PAOK MANDRAS"
result = get_coaches_of_match(match_id, team_name)
print(result)

conn.close()
    database="football",
    user="postgres",
    password="2110"
)

def get_coaches_of_match(match_id, team_name):
    with conn.cursor() as cursor:
        query = "SELECT c.first_name || ' ' || c.last_name AS coach_name " \
                "FROM public.matches m " \
                "JOIN public.teams t ON m.home_team = t.name OR m.away_team = t.name " \
                "JOIN public.coaches c ON t.team_id = c.team_id " \
                "WHERE m.match_id = %s AND t.name = %s"
        cursor.execute(query, (match_id, team_name))
```

```
        results = cursor.fetchall()
        if results:
            coaches = [result[0] for result in results]
            return "Coaches: " + ", ".join(coaches)
        else:
            return "No coaches found for the given match."

match_id = 52
team_name = "PAOK MANDRAS"
result = get_coaches_of_match(match_id, team_name)
print(result)

conn.close()
```

```
import psycopg2

conn = psycopg2.connect(
    host="localhost",
    database="football",
    user="postgres",
    password="2110"
)

def get_coaches_of_match(match_id, team_name):
    with conn.cursor() as cursor:
        query = "SELECT c.first_name || ' ' || c.last_name AS coach_name " \
                "FROM public.matches m " \
                "JOIN public.teams t ON m.home_team = t.name OR m.away_team = t.name " \
                "JOIN public.coaches c ON t.team_id = c.team_id " \
                "WHERE m.match_id = %s AND t.name = %s"
        cursor.execute(query, (match_id, team_name))
        results = cursor.fetchall()
        if results:
            coaches = [result[0] for result in results]
            return "Coaches: " + ", ".join(coaches)
        else:
            return "No coaches found for the given match."

match_id = 52
team_name = "PAOK MANDRAS"
result = get_coaches_of_match(match_id, team_name)
print(result)

conn.close()
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
$

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$ python connection.py
Coaches: Yiannis Raptis, Konstantinos Kostas, Kiriakos Aggelis
```

β)

```python
import psycopg2


host = 'localhost'
database = 'football'
user = 'postgres'
password = '2110'


connection = psycopg2.connect(host=host, database=database, user=user, password=password)


def get_goals_and_penalties(match_id):
    with connection.cursor() as cursor:
        query = "SELECT ms.time_of_event, p.last_name, p.first_name " \
                "FROM matchstats ms " \
                "JOIN players p ON ms.player_id = p.player_id " \
                "WHERE ms.match_id = %s AND (ms.goals > 0 OR ms.penalties > 0)"
        cursor.execute(query, (match_id,))
        result = cursor.fetchall()
        if result:
            return result
        else:
            return "No goals or penalties found for the given match."


match_id = 233
result = get_goals_and_penalties(match_id)
if isinstance(result, str):
    print(result)
else:
    for event in result:
        event_time = event[0]
        player_name = event[1]
        print(f"{event_time} - {player_name}")


connection.close()
```

```python
import psycopg2


host = 'localhost'
database = 'football'
user = 'postgres'
password = '2110'


connection = psycopg2.connect(host=host, database=database, user=user, password=password)


def get_goals_and_penalties(match_id):
    with connection.cursor() as cursor:
        query = "SELECT ms.time_of_event, p.last_name, p.first_name " \
                "FROM matchstats ms " \
                "JOIN players p ON ms.player_id = p.player_id " \
                "WHERE ms.match_id = %s AND (ms.goals > 0 OR ms.penalties > 0)"
        cursor.execute(query, (match_id,))
        result = cursor.fetchall()
        if result:
            return result
        else:
            return "No goals or penalties found for the given match."


match_id = 233
result = get_goals_and_penalties(match_id)
if isinstance(result, str):
    print(result)
else:
    for event in result:
        event_time = event[0]
        player_name = event[1]
        print(f"{event_time} - {player_name}")


connection.close()
```

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$ python connection2.py
16 - Sotiriou
```

γ)

```python
import psycopg2

conn = psycopg2.connect(database="football", user="postgres", password="2110", host="localhost", port="5432")
cur = conn.cursor()

player_id = 10

query = "SELECT ms.goals, ms.penalties, ms.yellow_cards, ms.red_cards, p.minutes_played, p.position " \
        "FROM matches m " \
        "JOIN matchstats ms ON m.match_id = ms.match_id " \
        "JOIN players p ON p.player_id = ms.player_id " \
        "WHERE p.player_id = %s"

cur.execute(query, (player_id,))

results = cur.fetchall()
for row in results:
    goals, penalties, yellow_cards, red_cards, minutes_played, position = row
    print(f"Γκολ: {goals}, Πέναλτι: {penalties}, Κίτρινες Κάρτες: {yellow_cards}, "
```

```
        f"Κόκκινες Κάρτες: {red_cards}, Λεπτά Αγώνα: {minutes_played}, Θέση: {position}")

cur.close()
conn.close()
```

```
import psycopg2

conn = psycopg2.connect(database="football", user="postgres", password="2110", host="localhost", port="5432")
cur = conn.cursor()

player_id = 10

query = "SELECT ms.goals, ms.penalties, ms.yellow_cards, ms.red_cards, p.minutes_played, p.position " \
        "FROM matches m " \
        "JOIN matchstats ms ON m.match_id = ms.match_id " \
        "JOIN players p ON p.player_id = ms.player_id " \
        "WHERE p.player_id = %s"

cur.execute(query, (player_id,))

results = cur.fetchall()
for row in results:
    goals, penalties, yellow_cards, red_cards, minutes_played, position = row
    print(f"Γκολ: {goals}, Πέναλτι: {penalties}, Κίτρινες Κάρτες: {yellow_cards}, "
          f"Κόκκινες Κάρτες: {red_cards}, Λεπτά Αγώνα: {minutes_played}, Θέση: {position}")

cur.close()
conn.close()
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$ python connection3.py
Γκολ: 0, Πέναλτι: 1, Κίτρινες Κάρτες: 0, Κόκκινες Κάρτες: 0, Λεπτά Αγώνα: 908, Θέση: Center Back
```

δ)

```
import psycopg2


conn = psycopg2.connect(
    host="localhost",
    database="football",
    user="postgres",
    password="2110"
)


desired_team = "ERMIS"
query = """
    SELECT
        home_wins AS home_wins,
        away_wins AS away_wins,
        home_losses AS home_losses,
        away_losses AS away_losses,
        home_draws AS home_draws,
        away_draws AS away_draws
    FROM
        teams
    WHERE
```

```
        name = %(team)s
"""


cursor = conn.cursor()
cursor.execute(query, {'team': desired_team})


result = cursor.fetchone()


print("Στατιστικά της ομάδας,",desired_team)
print("Νίκες Εντός:", result[0])
print("Νίκες Εκτός:", result[1])
print("Ήττες Εντός:", result[2])
print("Ήττες Εκτός:", result[3])
print("Ισοπαλίες Εντός:", result[4])
print("Ισοπαλίες Εκτός:", result[5])
print("Συνολικοί Αγώνες:", result[0]+result[1]+result[2]+result[3]+result[4]+result[5])
print("Αγώνες Εντός:", result[0]+ result[2]+result[4])
print("Αγώνες Εκτός:", result[1]+ result[3]+result[5])
print("Συνολικές Νίκες:", result[0]+result[1])
print("Συνολικές Ήττες:", result[2]+ result[3])
print("Συνολικές Ισοπαλίες:", result[4]+result[5])

cursor.close()
conn.close()
```

```python
import psycopg2


conn = psycopg2.connect(
    host="localhost",
    database="football",
    user="postgres",
    password="2110"
)


desired_team = "ERMIS"
query = """
    SELECT
        home_wins AS home_wins,
        away_wins AS away_wins,
        home_losses AS home_losses,
        away_losses AS away_losses,
        home_draws AS home_draws,
        away_draws AS away_draws
    FROM
        teams
    WHERE
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL

```
gdosi@DESKTOP-RB4EFDG MINGW64 ~
$

gdosi@DESKTOP-RB4EFDG MINGW64 ~
$ python connection4.py
Στατιστικά της ομάδας, ERMIS
Νίκες Εντός: 4
Νίκες Εκτός: 1
Ηττες Εντός: 3
Ηττες Εκτός: 6
Ισοπαλίες Εντός: 1
Ισοπαλίες Εκτός: 1
Συνολικοί Αγώνες: 16
Αγώνες Εντός: 8
Αγώνες Εκτός: 8
Συνολικές Νίκες: 5
Συνολικές Ηττες: 9
Συνολικές Ισοπαλίες: 2
```