Advanced Topics in Databases

Practical Assignment

Aníbal Silva Rui Fernandes

June, 2022

Database

Some changed parameters:

- athleteid = firstname + lastname + birthdate + inc_id;
- clubid = code + nation + region;
- resultid = meetid + resultid;
- swimstyleid = distance + relaycount + stroke;
- eventid = meetid + eventid.

Database


annp_final
athlete
license character varying(255)
athleteid character varying(255)
firstname character varying(250)
lastname character varying(255)
birthdate date
gender character varying(255)
nation character varying(255)
annp_final
club
clubid character varying(255)
code character varying(255)
nation character varying(255)
region character varying(255)
name character varying(255)
annp_final
meet
meetid integer
city character varying(255)
name character varying(255)
course character varying(255)
nation character varying(255)
annp_final
swimstyle
swimstyleid character varying(255)
relaycount integer
stroke character varying(255)
distance integer
public
facts
meetid integer
clubid character varying(255)
athleteid character varying(255)
swimstyleid character varying(255)
club_nation character varying(255)
club_region character varying(255)
athlete_gender character varying(255)
athlete_nation character varying(255)
points integer
swimtime time without time zone
distance integer

Average Age

```
SELECT AVG(age(birthdate))
FROM annp_final.athlete;
```

- **Average Age:** 46 years, 6 months and 31 days.

Youngest Athlete

```
SELECT *
FROM annp_final.athlete
ORDER BY age(birthdate) ASC
LIMIT 1;
```

- **Name:** Ana Mónica Eloi
- **Gender:** Female
- **Birthdate:** 29/12/1996
- **Age:** 25 years

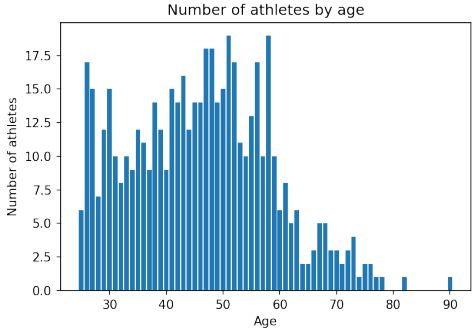
Oldest Athlete

```
SELECT *
FROM annp_final.athlete
ORDER BY age(birthdate) DESC
LIMIT 1;
```

- **Name:** Virgílio Zacarias Costa
- **Gender:** Male
- **Birthdate:** 21/07/1931
- **Age:** 90 years

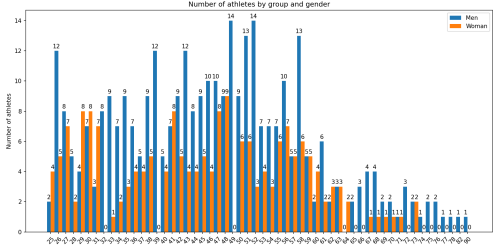
Athletes by Age

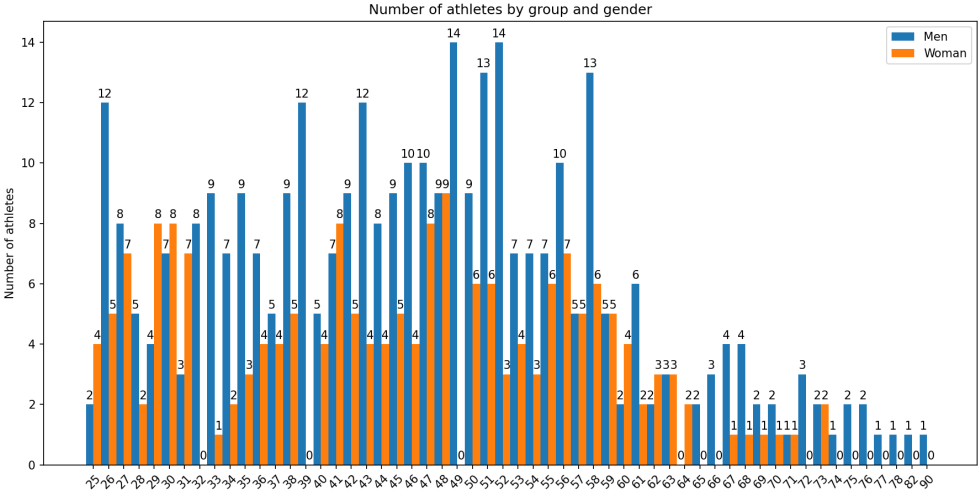
```
SELECT
    COUNT(*),
    EXTRACT(YEAR FROM age(birthdate)) AS age
FROM annp_final.athlete
GROUP BY age
ORDER BY age ASC;
```



Athletes by Age & Gender

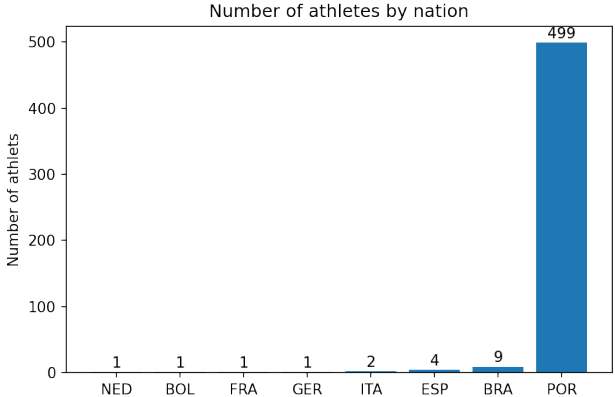
```
SELECT
    EXTRACT(YEAR FROM age(birthdate)) AS age,
    gender,
    COUNT(*)
FROM annp_final.athlete
GROUP BY (age, gender)
ORDER BY age ASC;
```




Number of athletes by group and gender
Number of athletes
Men
Woman
0
2
4
6
8
10
12
14
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 82 90

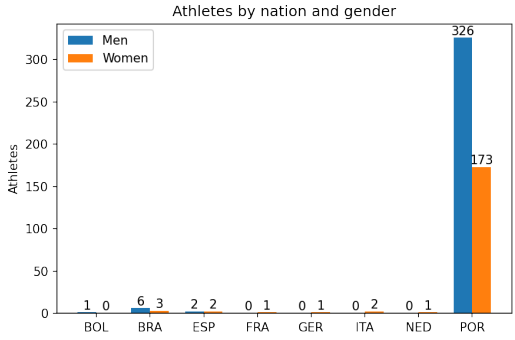
Athletes by Nation

```sql
SELECT
    nation,
    COUNT(*) AS nationCount
FROM annp_final.athlete
GROUP BY nation
ORDER BY nationCount ASC;
```

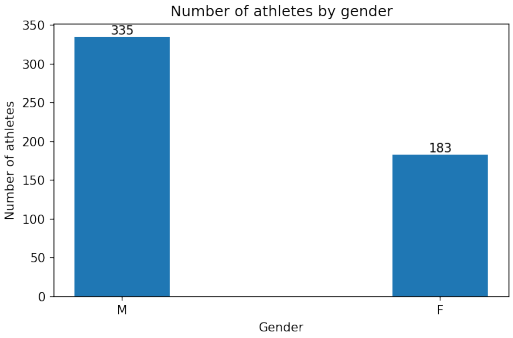


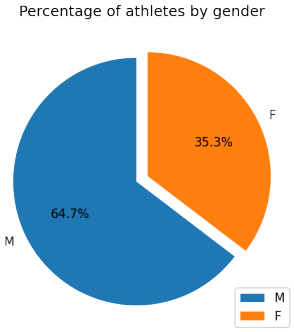
Athletes by Nation & Gender

```sql
SELECT
    nation,
    gender,
    COUNT(*)
FROM annp_final.athlete
GROUP BY CUBE (nation, gender)
ORDER BY nation, gender NULLS LAST;
```

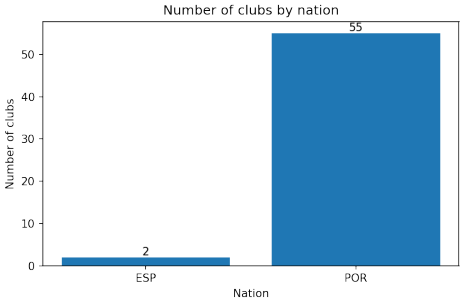


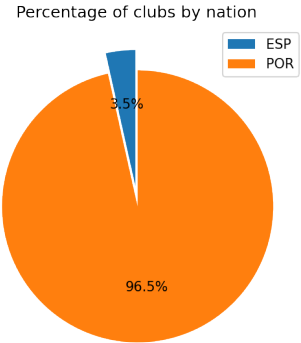
Athletes by Gender





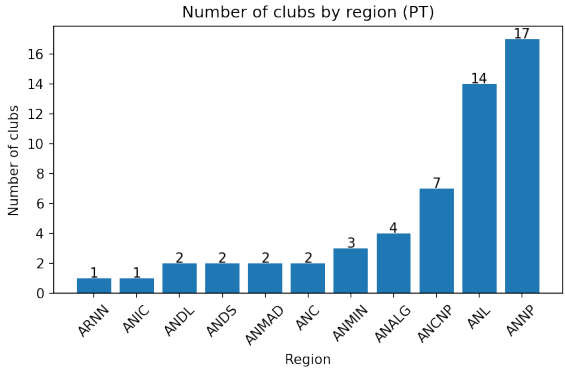
Clubs by Nation



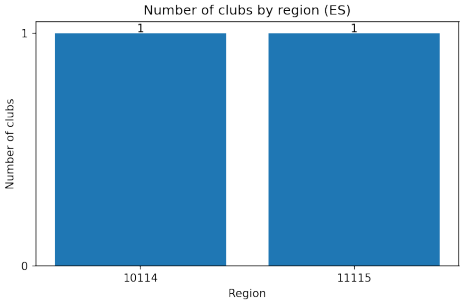


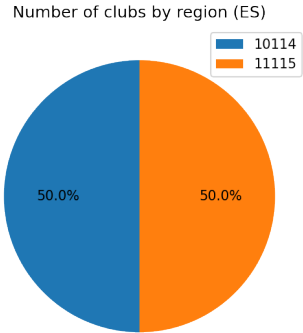
Clubs by Region (PT)

```sql
SELECT
    region,
    COUNT(*) AS regionCount
FROM annp_final.club
WHERE region SIMILAR TO '[A-Z]+'
GROUP BY region
ORDER BY regionCount ASC;
```



Clubs by Region (ES)

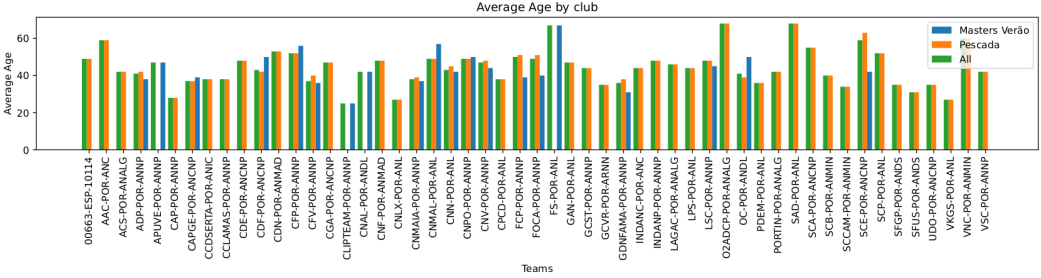




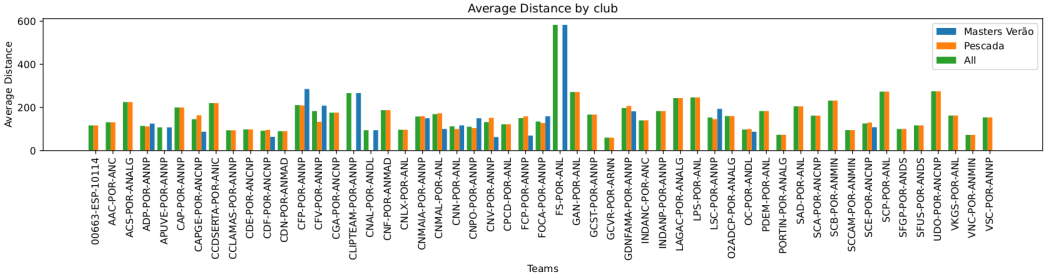
Club Fact statistics

```
    SELECT
    CASE GROUPING(af.clubid)
        WHEN 1 THEN 'all_clube'
        ELSE af.clubid
    END AS "Club",
    CASE GROUPING(af.meetid)
        WHEN 1 THEN 'all_meets'
        ELSE af.meetid
    END AS "Tournament",
    ROUND(AVG(points), 2) AS "Average Points",
    ROUND(AVG(distance), 2) AS "Average Distance",
    ROUND(AVG(swimtime), 2) AS "Average Swimtime",
    ROUND(AVG(birthdate)) AS "Average Age",
    COUNT(af.athleteid) AS "Total Athletes"
FROM (
    SELECT
        CAST(meetid AS VARCHAR(255)),
        f.athleteid,
        clubid,
        points,
        distance,
        ROUND(CAST(
              (EXTRACT(
                MICROSECONDS FROM swimtime))/10^6
                      AS NUMERIC), 2) AS swimtime,
        round(EXTRACT(
            EPOCH FROM
                CURRENT_TIMESTAMP - a.birthdate)/(365*24*60*60)) AS birthdate
    FROM facts f
    JOIN annp_final.athlete a ON a.athleteid = f.athleteid
    ) af
WHERE points IS NOT NULL
GROUP BY CUBE (af.clubid, af.meetid)
ORDER BY "Average Points" DESC;
```

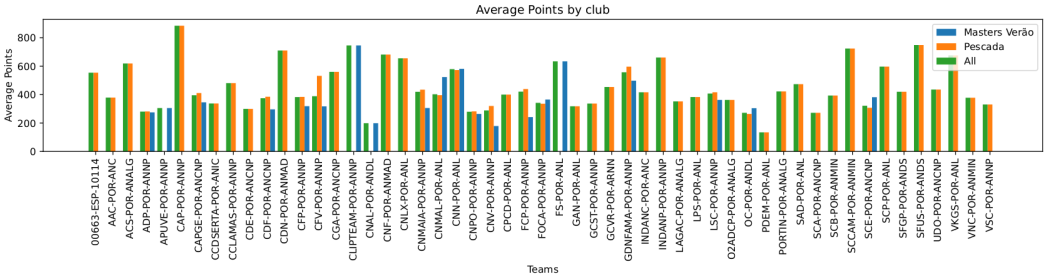
Overall Club Statistics I



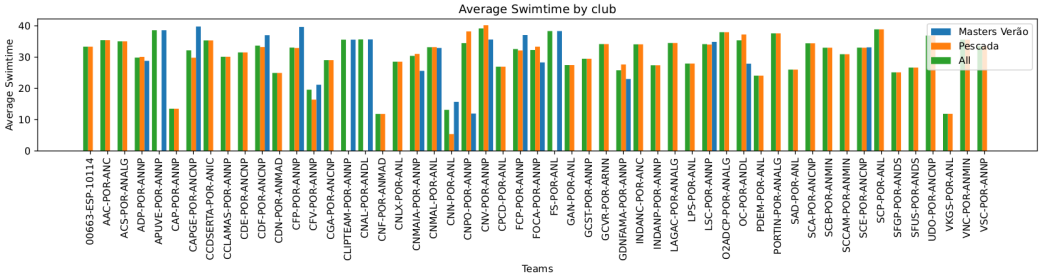
Overall Club Statistics II



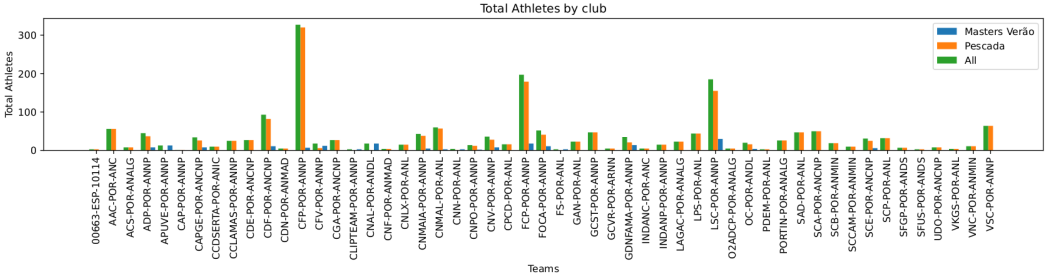
Overall Club Statistics III



Overall Club Statistics IV



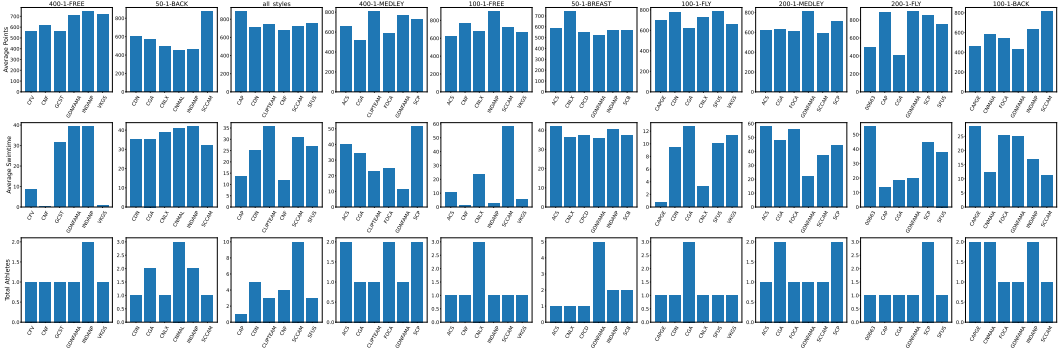
Overall Club Statistics V



Swim Style Statistics By Club

```
SELECT
    CASE GROUPING(c.code)
        WHEN 1 THEN 'all_clubs'
        ELSE c.code
    END AS "Team",
    CASE GROUPING(af.swimstyleid)
        WHEN 1 THEN 'all_styles'
        ELSE af.swimstyleid
    END AS "SwimStyle",
    ROUND(CAST(
                AVG(EXTRACT(
                        MICROSECONDS FROM af.swimtime))/10^6
                            AS NUMERIC), 2) as "Average Swimtime",
    round(AVG(af.points), 2) AS "Average Points",
    COUNT(af.athleteid) AS "Total Athletes"
    FROM facts af
    INNER JOIN annp_final.club c ON c.clubid = af.clubid
    WHERE af.points IS NOT NULL
    GROUP BY CUBE (af.swimstyleid, c.code)
```

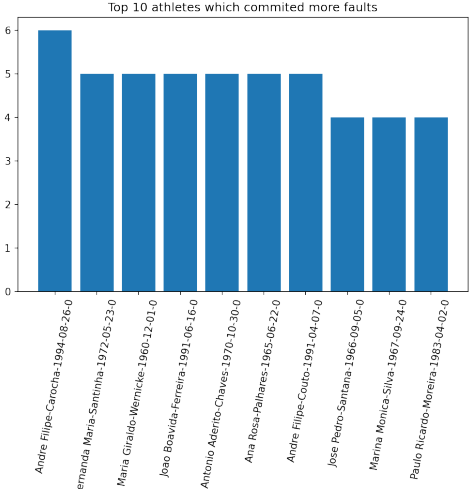
Swim Style Statistics



Athletes that commited faults

```
    WITH tmp AS (SELECT
        athleteid,
        CASE
                WHEN points IS NULL THEN 'fault'
                ELSE 'notfault'
        END AS fault
FROM facts)
SELECT
        athleteid,
        count(tmp.fault) AS "Number of Faults"
FROM tmp
WHERE fault = 'fault'
GROUP BY athleteid
ORDER BY "Number of Faults" DESC;
```

Athletes that commited faults



Athlete & Swim Style Statistics I

```
SELECT
CASE GROUPING(c.code)
    WHEN 1 THEN 'all_clubs'
    ELSE c.code
END AS "Team",
CASE GROUPING(af.swimstyleid)
    WHEN 1 THEN 'all_styles'
    ELSE af.swimstyleid
END AS "SwimStyle",
ROUND(CAST(
            AVG(EXTRACT(
                    MICROSECONDS FROM af.swimtime))/10^6
                        AS NUMERIC), 2) as "Average Swimtime",
round(AVG(af.points), 2) AS "Average Points",
COUNT(af.athleteid) AS "Total Athletes"
FROM facts af
INNER JOIN annp_final.club c ON c.clubid = af.clubid
WHERE af.points IS NOT NULL
GROUP BY CUBE (af.swimstyleid, c.code);
```

Athlete & Swim Style Statistics I