

Advanced Topics in Databases

Practical Assignment

Aníbal Silva Rui Fernandes

June, 2022

Database

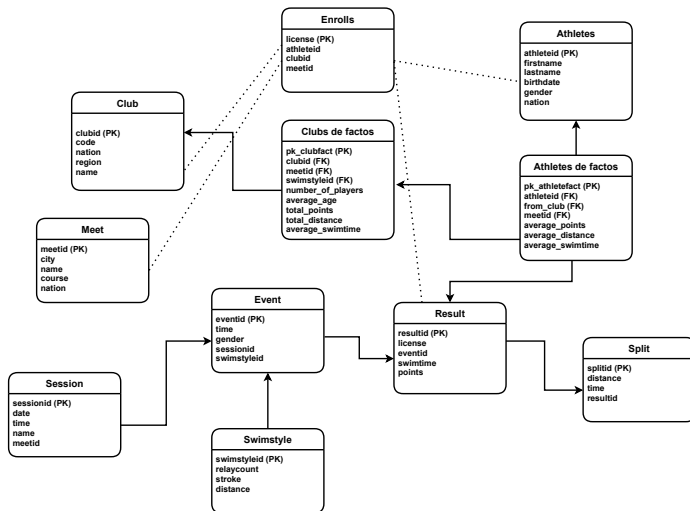
Some changed parameters:

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

PKs in fact tables:

- ▶ *Clubs de factos*, `pk_clubfact = clubid + meetid + swimstyleid;`
- ▶ *Athlete de factos*, `pk_athletefact = athleteid + from_club + meetid.`

Database



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

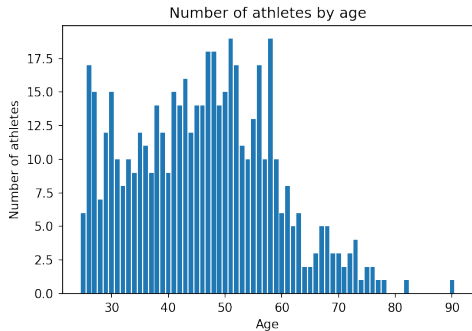
Oldest Athlete

```
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

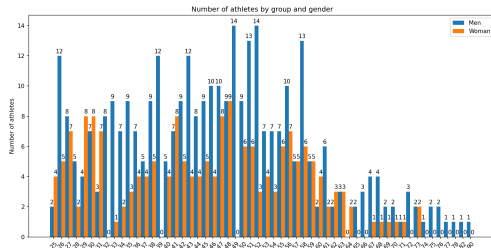
Athletes by Age

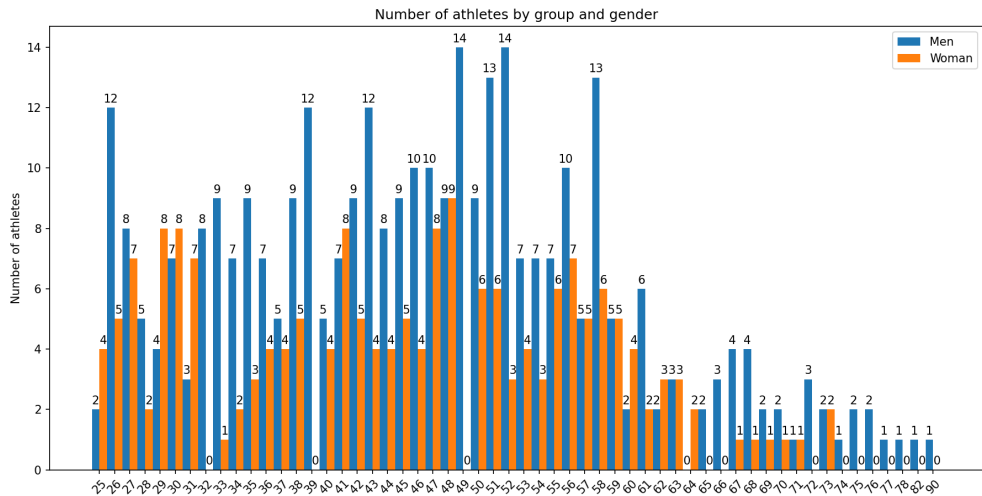
```
SELECT
    COUNT(*),
    EXTRACT(YEAR FROM age(birthdate)) AS age
FROM annp_final.athlete
GROUP BY age
ORDER BY age ASC;
```



Athletes by Age & Gender

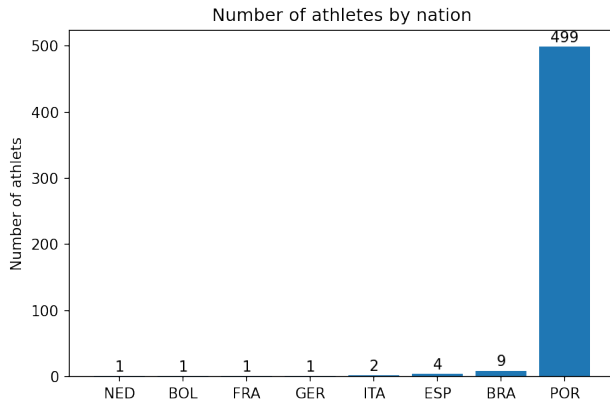
```
SELECT
    EXTRACT(YEAR FROM age(birthdate)) AS age,
    gender,
    COUNT(*)
FROM annp_final.athlete
GROUP BY (age, gender)
ORDER BY age ASC;
```





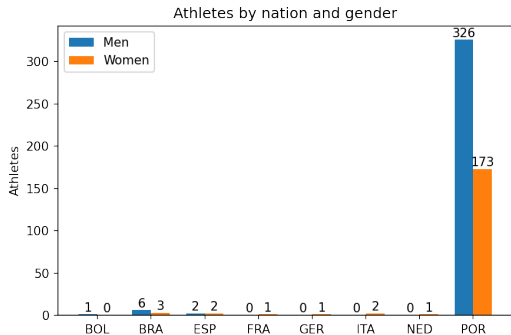
Athletes by Nation

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

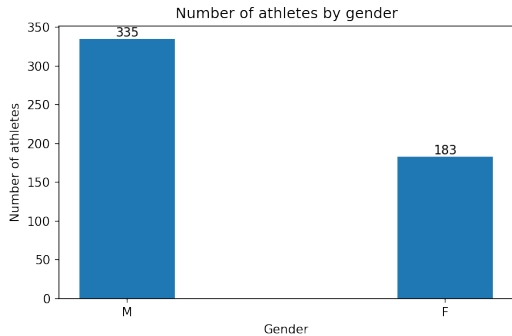


Athletes by Nation & Gender

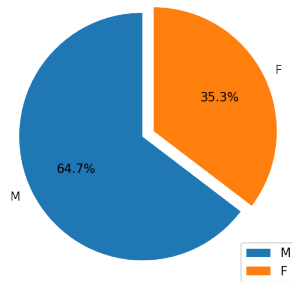
```
SELECT
    nation,
    gender,
    COUNT(*)
FROM annp_final.athlete
GROUP BY CUBE (nation, gender)
ORDER BY nation, gender NULLS LAST;
```



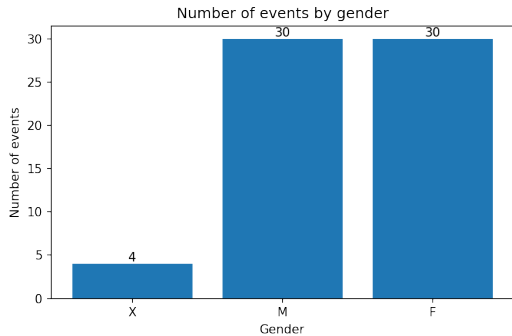
Athletes by Gender



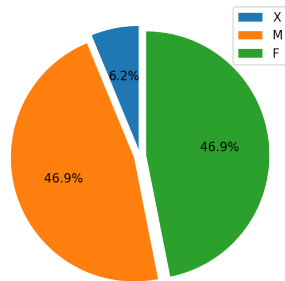
Percentage of athletes by gender



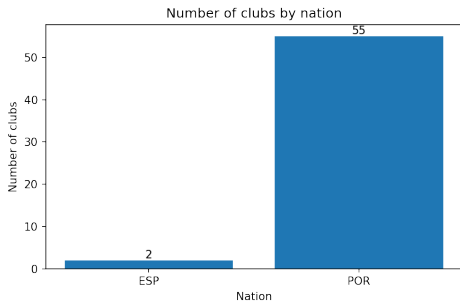
Events by Gender



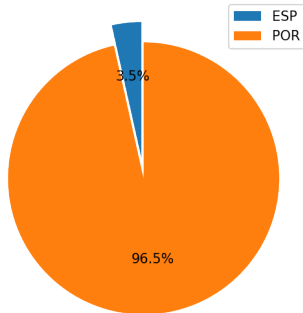
Percentage of events by gender



Clubs by Nation

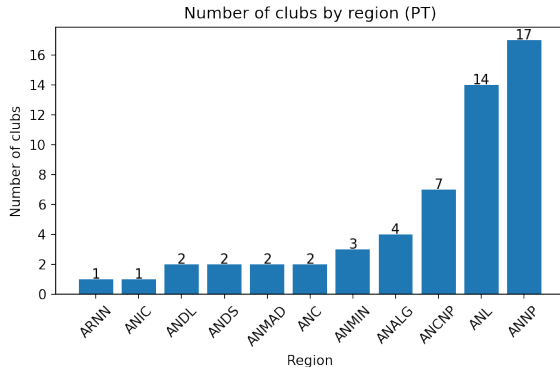


Percentage of clubs by nation

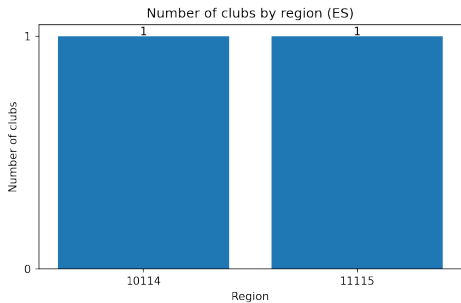


Clubs by Region (PT)

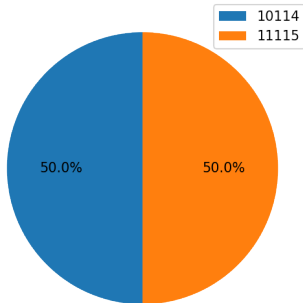
```
SELECT
    region,
    COUNT(*) AS regionCount
FROM annp_final.club
WHERE region SIMILAR TO '[A-Z]+'
GROUP BY region
ORDER BY regionCount ASC;
```



Clubs by Region (ES)



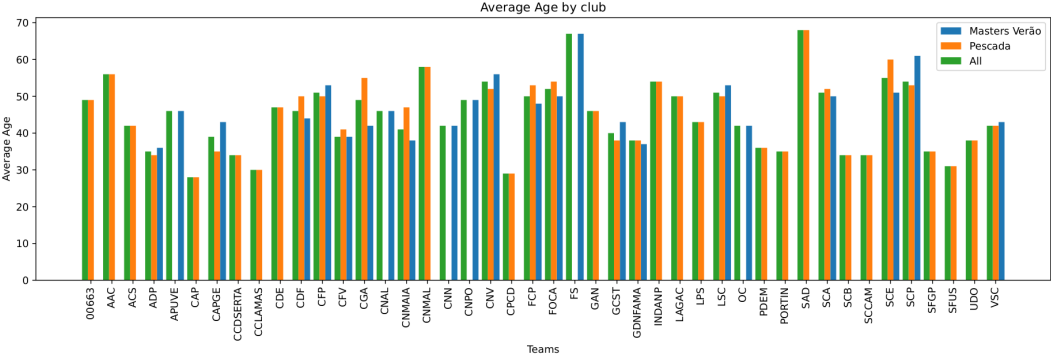
Number of clubs by region (ES)



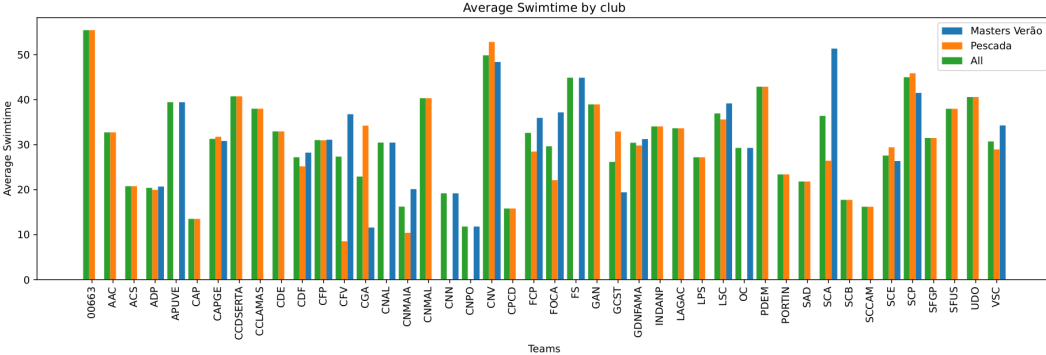
Club fact table statistics

```
SELECT
CASE GROUPING(cd.meetid)
  WHEN 1 THEN 'all_meets'
  ELSE cd.meetid
END AS "Tournament",
CASE GROUPING(c.code)
  WHEN 1 THEN 'all_clubs'
  ELSE c.code
END AS "Team",
ROUND(AVG(average_age), 0) AS "Average Age",
ROUND(AVG(average_swimtime), 2) AS "Average Swimtime",
ROUND(SUM(total_points)) AS "Total Points",
ROUND(SUM(number_of_players)) AS "Total Players"
FROM (
  SELECT CAST(meetid AS VARCHAR(255)),
         clubid,
         average_age,
         total_points,
         average_swimtime,
         number_of_players
  FROM annp_final.club_defacto) cd
JOIN annp_final.club c ON c.clubid = cd.clubid
GROUP BY CUBE (cd.meetid, c.code);
```

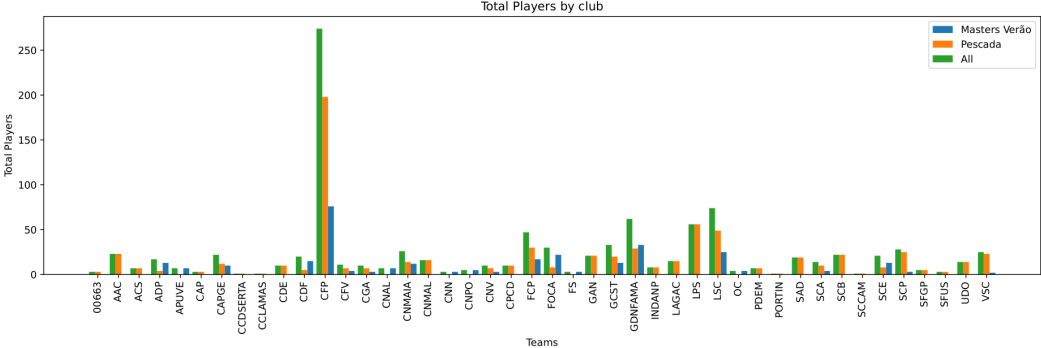
Overall Club Statistics I



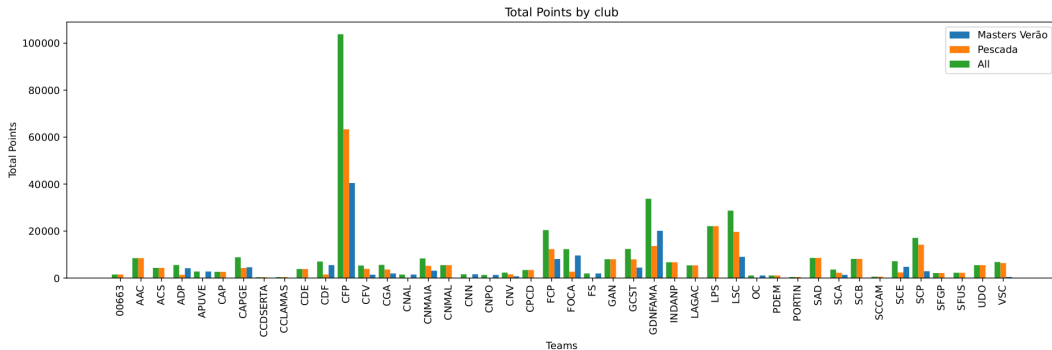
Overall Club Statistics II



Overall Club Statistics III



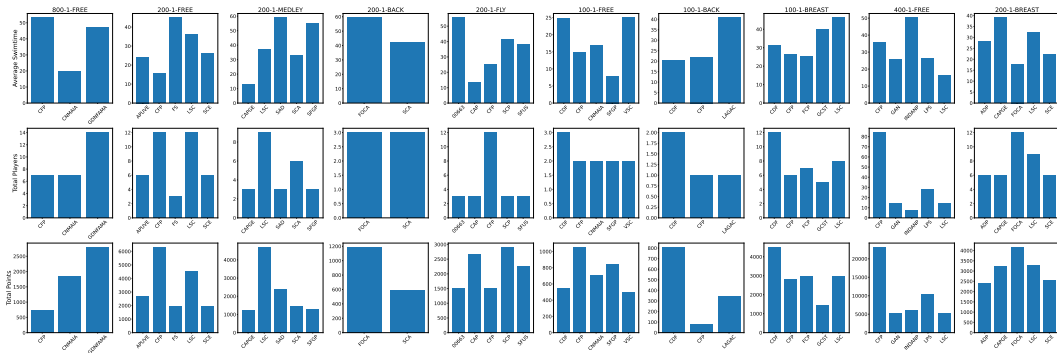
Overall Club Statistics IV



Swim Style Statistics By Club

```
SELECT
  CASE GROUPING(c.code)
    WHEN 1 THEN 'all_clubs'
    ELSE c.code
  END AS "Team",
  CASE GROUPING(cd.swimstyleid)
    WHEN 1 THEN 'all_styles'
    ELSE cd.swimstyleid
  END AS "SwimStyle",
  ROUND(AVG(average_swimtime), 2) AS "Average Swimtime",
  ROUND(SUM(total_points)) AS "Total Points",
  ROUND(SUM(number_of_players)) AS "Total Players"
FROM annp_final.club_defacto cd
JOIN annp_final.club c on c.clubid = cd.clubid
GROUP BY CUBE (c.code, cd.swimstyleid)
ORDER BY "Total Points" DESC
```

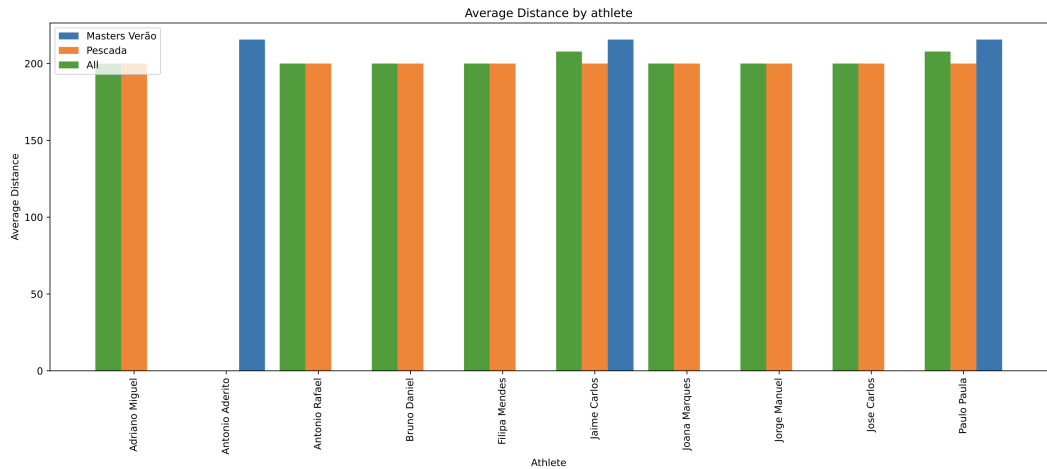
Swim Style Statistics



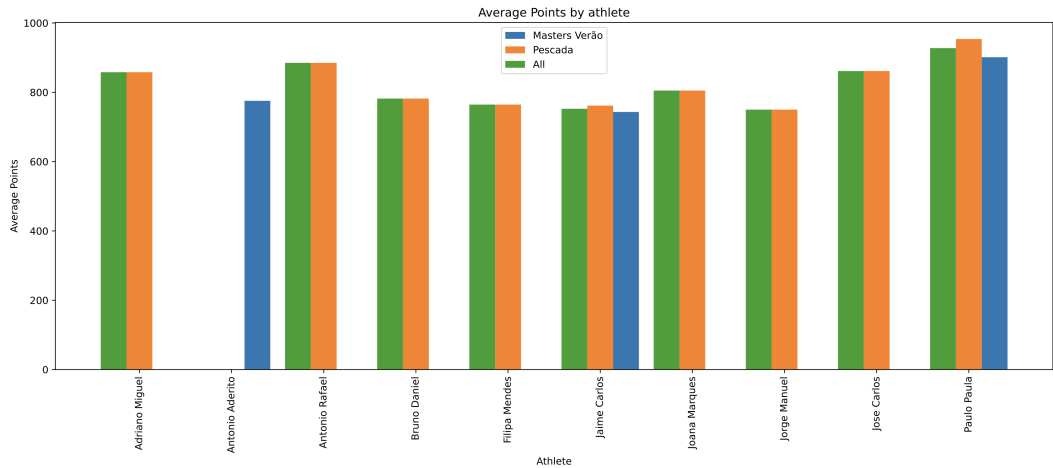
Athlete Statistics

```
SELECT
  CASE GROUPING(a.firstname)
    WHEN 1 THEN 'all_players'
    ELSE a.firstname
  END AS "Athletes",
  CASE GROUPING(af.meetid)
    WHEN 1 THEN 'all_meets'
    ELSE af.meetid
  END AS "Tournament",
  ROUND(AVG(average_points), 2) AS "Average Points",
  ROUND(AVG(average_distance), 2) AS "Average Distance",
  ROUND(AVG(average_swimtime), 2) AS "Average Swimtime"
FROM (
  SELECT
    athleteid,
    CAST(meetid as VARCHAR(255)),
    average_points,
    average_distance,
    average_swimtime
  FROM annp_final.athlete_defacto) af
JOIN annp_final.athlete a ON a.athleteid = af.athleteid
GROUP BY CUBE (a.firstname, af.meetid)
ORDER BY "Average Points" DESC;
```


Athlete Statistics I



Athlete Statistics II



Athlete Statistics III

