

# Projecto de Bases de Dados (CC2005) - parte 2

## 1. Elementos do grupo

### Grupo nº 24

Nº mecanográfico	Nome
202209875	Marta Sofia Vieira Longo
202205101	Sara Catarina Vieira Táboas

## 2. Ajustes ao modelo da BD

O modelo da BD descrito na primeira parte foi sujeito a alguns ajustes:

1. Foram criados ID's para todas as tabelas baseados na ocorrência de diferentes valores numa determinada coluna ou numa combinação de colunas (coluna(s) que se pretendia(m) identificar).
2. A base de dados foi dividida vários ficheiros em CSV correspondentes às futuras tabelas em SQL, e em cada ficheiro foram eliminadas as linhas com valores de ID duplicados.

## 3. Povoamento de tabelas

**1)** A base de dados foi povoada através da importação ficheiros CSV;

**2)** O povoamento foi realizado a partir da criação anterior de vários ficheiros CSV, a partir do ficheiro extraído. Foi utilizada a biblioteca Pandas para separar as colunas do conjunto de dados relevantes para cada tabela e fazer as alterações necessárias. O povoamento da base de dados foi então conseguido através da criação de tabelas (tabelas do modelo relacional), cada uma delas gerada pela importação do ficheiro CSV correspondente.

3)

Nome da tabela	Nº de entradas
Atletas	186960
Categorias	765
Equipas	1231
Eventos	52
Modalidades	66
Participacoes	271116

## 4. Aplicação Python

1)

“Endpoint”	Funcionalidade
/	Página de entrada
/general-statistics/ (1)	Algumas estatísticas da base de dados
/athletes/	Lista todas as entradas da tabela Atletas, apresentado, para cada atleta, o seu ID e o nome do atleta correspondente
/athletes/<int: id>/	Lista as informações (idade, sexo, altura e peso) do atleta com ID=id. Permite acesso às participações do atleta especificado
/athletes/<int: id>/athletes-participations (2)	Lista todas as participações do atleta com ID=id, referindo a categoria, ano, cidade e época, e a medalha conseguida na participação.
/athletes/search/<expr>/ (3)	Lista todos os atletas cujo nome contém a expressão ‘expr’
/teams/	Lista todas as entradas da tabela Equipas, apresentando, para cada equipa, o seu ID, a sua designação e também o seu código olímpico
/teams/<int: id>/	Lista as informações da equipa com ID=id (nome, ID e código olímpico) e todos os membros que concorreram por essa equipa
/teams/search/<expr> (3)	Lista todas as equipas cuja designação contém a expressão ‘expr’

/games/	Lista todas as entradas da tabela Eventos, apresentando, para cada evento, o seu ID e o ano/época
/games/<int: id>/	Lista as informações do evento com ID=id, número de atletas que nele participaram, apresentando também o top 3 dos atletas mais premiados nesse evento, bem como o número de medalhas
/games/search/<expr> (3)	Lista as informações de todos os jogos cujo ano, cidade ou época contém a expressão 'expr'
/sports/	Lista todas as entradas da tabela Modalidades, apresentando para cada modalidade, o seu ID e a sua designação
/sports/<int: id>	Lista todas as categorias da modalidade com ID=id
/sports/search/<expr> (3)	Lista todas as modalidades cuja designação contém a expressão 'expr'
/categories/	Lista todas as entradas da tabela Categorias, apresentando, para cada categoria, o seu ID e a sua designação
/categories/<int: id>/	Lista os resultados da categoria com ID=id por ano, apresentando o ID e o nome dos atletas vencedores de medalhas e a respetiva medalha
/categories/search/<expr> (3)	Lista o nome de todas as categorias cuja designação contém a expressão 'expr'

## 2) Consultas SQL

### (1) /general-statistics/

1.

```
SELECT * FROM
  (SELECT COUNT(*) n_atletas FROM Atletas)
  JOIN
  (SELECT COUNT(*) n_modalidades FROM Modalidades)
  JOIN
  (SELECT COUNT(*) n_categorias FROM Categorias)
  JOIN
  (SELECT COUNT(*) n_eventos FROM Eventos)
  JOIN
  (SELECT COUNT(*) n_equipas FROM Equipas)
  (SELECT COUNT(*) atletas_mulheres FROM Atletas WHERE sex = 'F')
  JOIN
  (SELECT COUNT(*) atletas_homens FROM Atletas WHERE sex = 'M');
```

Seleciona todos os dados da BD e conta todas as entradas das tabelas Atletas, Modalidades, Categorias, Eventos e Equipas. Conta, também, os atletas femininas e masculinos.

2.

```
SELECT COUNT(*) AS count_medals, a.name, a.idAtletas, e.team
FROM Atletas a
JOIN Participacoes p ON a.idAtletas = p.idAtletas
JOIN Equipas e ON e.idEquipas = a.idEquipas
WHERE (p.medal = 'Gold' OR p.medal = 'Silver' OR p.medal = 'Bronze')
GROUP BY a.name
ORDER BY count_medals DESC
LIMIT 10;
```

Seleciona o nome dos 10 atletas com maior número de medalhas acumuladas, o seu ID, a equipa a que pertencem e conta o número de medalhas ganhas por cada atleta. A seleção é feita a partir da junção das tabelas Atletas, Participações e Equipas e a lista está ordenada decrescentemente pelo número medalhas de cada atleta.

3.

```
SELECT e.team, COUNT(*) AS count_medals, e.idEquipas
FROM Equipas e
JOIN Atletas a ON a.idEquipas = e.idEquipas
JOIN Participacoes p ON p.idAtletas = a.idAtletas
WHERE (p.medal = 'Gold' OR p.medal = 'Silver' OR p.medal = 'Bronze')
GROUP BY e.team
ORDER BY count_medals DESC
LIMIT 10;
```

Seleciona a designação e o ID das 10 equipas com maior número de medalhas acumuladas e conta o número de medalhas ganhas por cada equipa. A seleção é feita a partir da junção das tabelas Equipas, Atletas e Participações e a lista está ordenada decrescentemente pelo número de medalhas de cada equipa.

4.

```
SELECT e.year, e.season, e.city, COUNT(*) AS number_athletes, e.idEventos,
COUNT (CASE WHEN a.sex = 'F' THEN 1 END) AS number_female_athletes,
COUNT (CASE WHEN a.sex = 'M' THEN 1 END) AS number_male_athletes
FROM Eventos e
JOIN Participacoes p ON e.idEventos = p.idEventos
JOIN Atletas a ON p.idAtletas = a.idAtletas
GROUP BY e.year, e.season, e.city
ORDER BY number_athletes DESC
LIMIT 10;
```

Seleciona o ID, ano, cidade e época dos 10 eventos que tiveram maior número de participantes e conta o número de atletas total, o número de atletas femininas e o número de atletas masculinos que participaram em cada evento. A seleção é feita a partir das tabelas Eventos, Participações e Atletas e a lista está ordenada decrescentemente pelo número total de participantes de cada evento.

(2) /athletes/<int: id>/athletes-participations

5.

```
SELECT a.name, e.city, e.season, e.year, c.event, p.medal, e.idEventos,
c.idCategorias
FROM Eventos e
JOIN Participacoes p ON e.idEventos = p.idEventos
JOIN Categorias c ON p.idCategorias = c.idCategorias
JOIN Atletas a ON p.idAtletas = a.idAtletas
WHERE a.name LIKE (
    SELECT name
    FROM Atletas
    WHERE idAtletas = :id
);
```

Seleciona, para cada participação do atleta com ID = id, a categoria em que concorreu, o ano, cidade e época do evento em que participou e a medalha conseguida. A seleção é feita a partir da junção das tabelas Eventos, Participações, Categorias e Atletas.

(3) /athletes/search/<expr>/

5.

```
SELECT name, idAtletas
FROM Atletas
WHERE name LIKE ?
GROUP BY name
```

Seleciona, a partir da tabela Atletas, todos os nomes de todos os atletas distintos e os seus respetivos ID 's cujo nome contenha a expressão representada por '?'.

As restantes consultas SQL correspondentes aos *endpoints* identificados com (3) na tabela acima, são semelhantes a esta, com a única diferença de utilizar o ID e o atributo correspondente à designação da tabela a que se acede.