

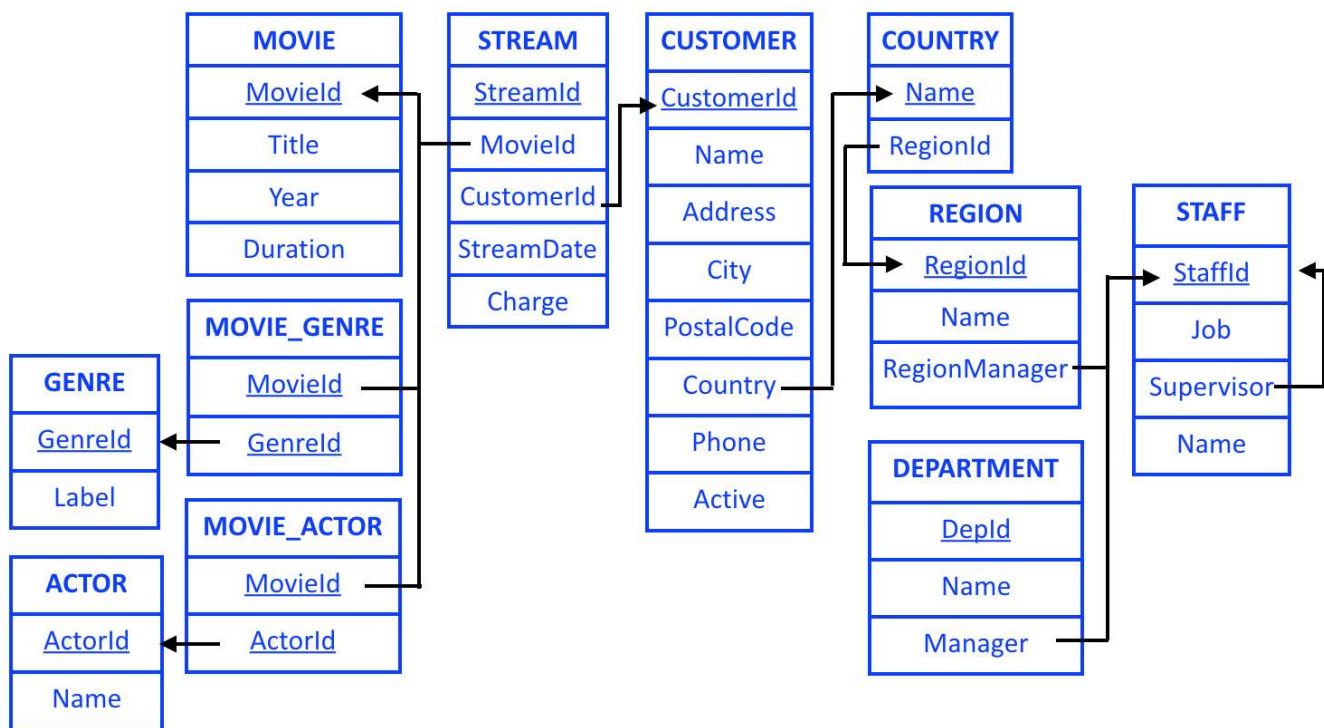
Bases de Dados - Aulas práticas - Ficha 11

Objectivos: Treino para o teste no Mooshak.



BD de referência

Iremos continuar usar a BD MovieStream.



SQL

[mstream.sql](#)

1

Escreva uma instrução **SELECT** para obter os registos correspondentes à seguinte sequência de operações em álgebra relacional para a BD MovieStream:

$$R_0 \leftarrow \text{CUSTOMER} \bowtie_{\text{CUSTOMER.CustomerId} = \text{STREAM.CustomerId}} \text{STREAM}$$

$$R_1 \leftarrow \sigma_{\text{Country} = 'India'}(R_0)$$

$$R_2 \leftarrow \text{Name } \mathcal{F}_N = \text{COUNT(StreamId), SUM(Charge)}(R_1)$$

$$R_3 \leftarrow \sigma_{N \leq 5}(R_2)$$

Resultado esperado (fragmento; a ordem é irrelevante, não precisa de usar ORDER BY):

```
+-----+-----+
| Alicia Mills      | 2 | 15.50 |
| Allen Butterfield | 4 | 32.25 |
| Amy Lopez        | 4 | 28.50 |
| Anita Morales    | 0 | NULL  |
| Beatrice Arnold  | 4 | 34.50 |
...
| Samuel Marlow    | 5 | 39.50 |
| Sherri Rhodes    | 0 | NULL  |
| Steven Curley    | 3 | 20.25 |
| Tim Cary         | 4 | 30.00 |
| Tonya Chapman    | 2 | 13.25 |
+-----+-----+
28 rows in set (0.01 sec)
```

2

Use uma instrução **SELECT** para obter o título do filme, a altura do "stream", e o valor cobrado para "streams" feitos em **Dezembro de 2017** de filmes classificados com o género '**Thriller**'.

Os resultados devem ser ordenados primeiro por título de filme, e depois (para "streams" do mesmo filme) de forma decrescente pela altura do stream.

Resultado esperado (fragmento):

```
+-----+-----+-----+
| Title                                | StreamDate          | Charge |
+-----+-----+-----+
| A History of Violence                | 2017-12-05 00:55:00 | 5.50   |
| Argo                                | 2017-12-30 14:04:00 | 5.50   |
| Blade Runner                         | 2017-12-05 10:46:00 | 6.25   |
| Captain Phillips                     | 2017-12-27 11:21:00 | 9.50   |
| Captain Phillips                     | 2017-12-17 00:11:00 | 5.00   |
...
| War of the Worlds                    | 2017-12-31 14:10:00 | 8.25   |
| White House Down                     | 2017-12-24 17:42:00 | 5.50   |
| White House Down                     | 2017-12-14 03:40:00 | 7.00   |
+-----+-----+-----+
44 rows in set
```

3

Use uma instrução **SELECT** para obter o título de todos os filmes que estejam classificados (pelo menos) com os géneros **'Action'** e **'Comedy'** (note que um filme pode ser classificado com vários géneros via **MOVIE_GENRE**.)

Os resultados devem ser ordenados pelo título de filme.

Resultado esperado (fragmento):

```

+-----+
| Title                                     |
+-----+
| 2 Guns                                  |
| 21 Jump Street                         |
| 22 Jump Street                         |
| Ant-Man                                |
| Austin Powers in Goldmember            |
| ...                                     |
| The Other Guys                         |
| The World's End                       |
| This Means War                         |
| Tropic Thunder                         |
| True Lies                             |
+-----+
45 rows in set

```

4

Use uma instrução **SELECT** para obter os título dos **20** filmes mais vistos por clientes oriundos da região **'Asia'** e correspondente nº de "streams" feitos.

Os resultados devem ser ordenados primeiro de forma decrescente pelo número de "streams", e depois (para filmes com igual nº de streams) pelo título do filme.

Resultado esperado (fragmento):

```
+-----+-----+
```

Traffic	16
Grindhouse	15
Star Wars: Episode VII - The Force Awakens	15
...	
The Passion of the Christ	12
The Terminator	12
X-Men	12

```
+-----+-----+
```

20 rows in set

5

Use uma instrução **UPDATE** para colocar o valor cobrado a **0 (Charge = 0)** para todos os "streams" associados a:

- clientes oriundos de qualquer região **EXCEPTO 'America'**;
- e filmes com uma duração igual ou inferior a **120** minutos classificados com o género **'Biography'**

Resultado esperado (em termos do nº de registos actualizados)

```
Query OK, 111 rows affected
```

6

Use uma instrução **DELETE** para remover todos os registos em **MOVIE_ACTOR** (participação de actores em filmes) associados a filmes classificados com o género **'Action'** e que não tenham tido nenhum "stream".

Resultado esperado (em termos do nº de registos removidos)

```
Query OK, 156 rows affected
```

7

Crie uma vista ("view") chamada **REGION_DATA** com campos **Name**, **Manager**, **Countries** e **Customers** em correspondência aos seguintes dados por região: o nome da região, o nome do "region manager", o número de países na região, e o nº de clientes de países nessa região.

Os dados da vista não precisam de estar ordenados por nenhum critério (não precisa de usar **ORDER BY**).

Para criar a vista use código com a seguinte forma:

```
DROP VIEW REGION_DATA;  
  
CREATE VIEW REGION_DATA(Name, Manager, Countries, Customers)  
AS  
...  
;
```

Resultado esperado para a consulta `SELECT * FROM REGION_DATA` após criada a "view" (fragmento):

```

+-----+-----+-----+-----+
| Name          | Manager          | Countries | Clients |
+-----+-----+-----+-----+
| Africa        | Felícia Antunes  | 21        | 63       |
...
| Europe        | Gastão Pinto     | 31        | 125      |
| Other countries | Gastão Pinto     | 10        | 11       |
+-----+-----+-----+-----+
5 rows in set

```

Nota: Irá obter como resposta **Compile Time Error** se os nomes dos campos da vista (Name, Manager, Countries, Customers) não corresponderem aos esperados.

8

Escreva uma instrução **SELECT** para obter os nomes de pares de actores que tenham actuado ambos no mesmo filme, o correspondente nº de filmes (distintos) em que participaram os dois, e o nº de streams feitos para esses filmes tais que o nº de streams é igual ou superior a **40**. O nome do 1º actor deve preceder alfabeticamente o nome do segundo actor.

Os resultados devem ser ordenados por: **(1)** nº de streams de forma decrescente; **(2)** nº de filmes de forma decrescente, **(3)** pelo nome do 1º actor; e finalmente **(4)** pelo nome do 2º actor.

Resultado esperado (fragmento):

```

+-----+-----+-----+-----+
| Helena Bonham Carter | Johnny Depp      | 5 | 100 |
| Daniel Radcliffe     | Rupert Grint     | 8 | 75  |
| Daniel Radcliffe     | Emma Watson      | 7 | 67  |
| Emma Watson          | Rupert Grint     | 7 | 67  |
| Cameron Diaz         | Eddie Murphy     | 4 | 61  |
...
| Ioan Gruffudd        | Jessica Alba     | 2 | 41  |
| Ioan Gruffudd        | Michael Chiklis  | 2 | 41  |
| Jessica Alba         | Michael Chiklis  | 2 | 41  |
| Ewan McGregor       | Natalie Portman  | 3 | 40  |
| Ben Stiller          | Owen Wilson      | 2 | 40  |
+-----+-----+-----+-----+
41 rows in set (0.23 sec)

```

9

Use uma instrução **SELECT** para obter para todas as combinações de nomes de países na região '**Other Countries**' e géneros de filmes, e para cada uma dessas combinações país-género o nº de streams feitos por clientes no país de filmes com esse género.

Os resultados deverão ser ordenados primeiro por nome de país e depois por género de filme.

Resultado esperado (fragmento):

```
+-----+-----+-----+
| Anguilla          | Action   | 1 |
| Anguilla          | Adventure| 0 |
| Anguilla          | Animation| 0 |
| Anguilla          | Biography| 0 |
...
| Anguilla          | Western  | 0 |
| French Polynesia  | Action   | 16|
| French Polynesia  | Adventure| 15|
...
| Virgin Islands    | Western  | 0 |
+-----+-----+-----+
210 rows in set ...
```

10

Suponha que a empresa MovieStream pretende atribuir um bônus de Natal a cada empregado identificado por **staff_id** em determinado ano **year** de acordo com a seguinte fórmula base:

$$n * (s + r + 1)$$

onde:

- **n** é o nº de streams do mês de Dezembro do ano **year**;
- **s** é o nº de funcionários supervisionados por **staff_id**;
- **r** é o nº de regiões geridas por **staff_id**;

Depois de calculado o valor pela fórmula descrita, este deve ser acertado tendo em conta tectos mínimos e máximos com valores de **300** e **2000** respectivamente, i.e., o valor de bônus deve ser acertado para **300** se **value < 300** e para **2000** se **value > 2000**, onde **value** é o valor obtido pela fórmula base.

Defina uma "**stored function**" chamada **bonus** para fazer este cálculo, onde pode assumir que o valor do parâmetro **staff_id** corresponde a uma entrada existente na tabela **STAFF**. Use a seguinte estrutura para o seu código:

```

DROP FUNCTION IF EXISTS bonus;
DELIMITER $

CREATE FUNCTION bonus(staff_id INT, year INT)
RETURNS INT
BEGIN
    DECLARE value INT;
    ... a completar ...
    RETURN value;
END$

DELIMITER ;

```

Se a "stored function" estiver correctamente definida então a seguinte consulta:

```

SELECT StaffId, bonus(StaffId,2017), bonus(StaffId,2018)
FROM STAFF ORDER BY StaffId;

```

deverá retornar algo como (fragmento):

```

+-----+-----+-----+
| StaffId | bonus(StaffId,2017) | bonus(StaffId,2018) |
+-----+-----+-----+
|      1 |          1290 |          2000 |
|      2 |           774 |          1779 |
|      3 |           300 |           593 |
...
|      9 |           516 |          1186 |
...
|     14 |         1032 |          2000 |
...
|     17 |           774 |          1779 |
+-----+-----+-----+
17 rows in set ...

```