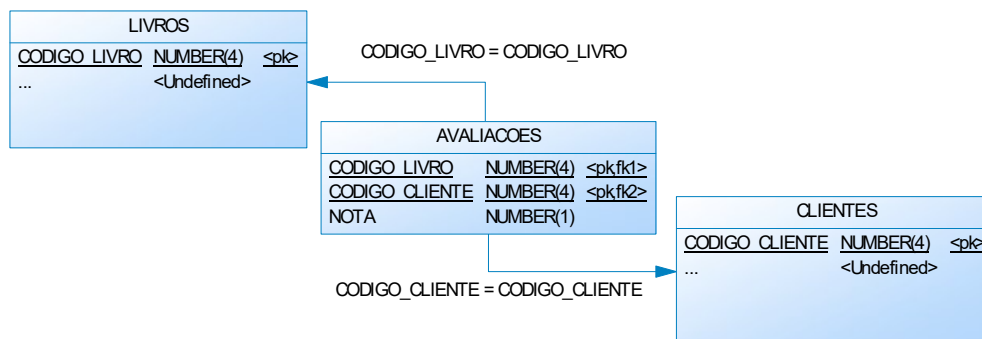


## Bases de Dados 2018/2019

### Exercícios da 8ª Aula (SQL):

1. Crie a tabela **AUTORES2**, análoga à tabela **AUTORES** contendo as seguintes restrições:  
→ Os nomes devem ser em maiúsculas;  
→ A idade tem que ser um valor positivo menor 200.
2. Pretende-se que passe a ser possível aos clientes da livraria avaliarem os livros que esta disponibiliza. A avaliação consistirá na atribuição de um número de estrelas compreendido entre 1 (livro muito mau) e 5 (livro muito bom). Crie a nova tabela **AVALIACOES** que permita registar esta informação, de acordo com o diagrama abaixo.



3. Faça uma cópia da tabela **LIVROS** para uma nova tabela **LIVROS\_BACKUP**. Use o comando **CREATE TABLE** com subconsulta. Na cópia de valores transforme os géneros dos livros para maiúsculas. Note que como está a produzir alterações sobre algumas colunas irá ter que usar pseudónimos para a tabela **LIVROS\_BACKUP** possuir colunas com nomes válidos.
4. Como ao criar a nova tabela **LIVROS\_BACKUP** as restrições não são copiadas altere a tabela de modo a incluir as restrições que existiam na tabela **LIVROS** original. Use o comando **ALTER TABLE**.
5. Insira, na tabela **LIVROS\_BACKUP**, o novo livro do Sérgio Sousa, cujo título é 'Informática para todos' e que foi editado hoje pela FCA. O livro custa 24€ e tem 430 páginas, tendo-lhe sido atribuído o ISBN 132434 e o código 51. Ainda não foi vendida qualquer unidade deste livro.
6. Insira na tabela **AUTORES2** um novo autor 'José de Magalhães' com o código 45 e contribuinte 77665544.
7. Apague, da tabela **LIVROS\_BACKUP**, o livro de código 51.

8. Actualize, na tabela **LIVROS\_BACKUP**, os preços dos livros de aventuras de modo a que estes sofram um aumento em 10%.
9. Apague da tabela **LIVROS\_BACKUP**, os livros de informática que tem um preço inferior à média dos preços de informática.
10. Faça uma réplica da **AUTORES** usando o comando *CREATE TABLE* com subconsulta, chamando à nova tabela **AUTORES\_BACKUP**.
11. Adicione à tabela **AUTORES\_BACKUP** a coluna **NLIVROS** do tipo *NUMBER* que irá conter informação sobre a quantidade de livros escritos por esse autor. A nova coluna é de preenchimento obrigatório, deve conter, por defeito, o valor 0 (zero) e deve permitir armazenar apenas valores compreendidos entre 0 e 150.
12. Actualize o campo **NLIVROS** adicionado no ponto anterior, com a informação do número de livros que cada autor escreveu.
13. Actualize o campo **GENERO\_PREFERIDO** da tabela **AUTORES\_BACKUP** de modo a referir-se ao género de livros que o autor mais escreveu.
14. Crie uma réplica da tabela **EDITORAS**, usando o comando *CREATE TABLE* com subconsulta, chamando à nova tabela **EDITORAS\_BACKUP**.
15. Apague da base de dados a tabela **EDITORAS\_BACKUP**.

### Exercícios Propostos

- a) Acrescente uma coluna **EDICAO**, do tipo *NUMBER(7)*, a **LIVROS\_BACKUP**.
- b) Actualize o atributo **UNIDADES\_VENDIDAS** da tabela de **LIVROS\_BACKUP** com a informação referente às vendas registadas pela livraria.
- c) Actualize o atributo **TOTAL\_VENDA** da tabela **VENDAS\_BACKUP** com a informação referente ao produto da quantidade de livros da venda (**QUANTIDADE**) pelo preço de venda do livro (**PRECO\_UNITARIO**).
- d) Crie a tabela **VENDAS\_BACKUP**, réplica da tabela **VENDAS**. Use o comando *CREATE TABLE* com subconsulta.
- e) Remova, da tabela **VENDAS\_BACKUP**, todas as vendas efectuadas no mês de Janeiro.
- f) Remova, da tabela **VENDAS\_BACKUP**, todas as vendas referentes ao livro mais caro (**PRECO\_TABELA**) do autor Sérgio Sousa.

```
INSERT INTO tabela VALUES (...)  
INSERT INTO tabela (col1 , col2 ) VALUES (21, 23)  
INSERT INTO tabela (...) SELECT ...
```

```
UPDATE tabela SET COL1 = VAL1, COL2=VAL2, ...WHERE ...  
UPDATE tabela SET (COL1,COL2) = (subconsulta) WHERE ...
```

```
DELETE FROM tabela WHERE ...
```

```
CREATE TABLE tabela (  
    col1 TIPO_DADOS [DEFAULT valor] [RESTRICAO1 ... ],  
    col2 TIPO_DADOS [DEFAULT valor] [RESTRICAO2 ... ],  
    [restrição1_tabela],  
    [restrição2_tabela]);
```

#### **RESTRIÇÕES**

PRIMARY KEY ,

UNIQUE

CHECK ( )

NOT NULL

FOREIGN KEY campo REFERENCES tabela(campo)

**Ex.:**

```
[CONSTRAINT nome_a_dar] PRIMARY KEY,  
CONSTRAINT nome_a_dar PRIMARY KEY(COLUNAS),
```

#### **NOTA :**

COLUNAS DO TIPO LONG E LONG RAW NÃO PODEM SER CHAVES PRIMARIAS

```
CREATE TABLE tabela AS SELECT ...
```

```
ALTER TABLE tabela
```

```
    ADD (coluna TIPO [DEFAULT valor] [RESTRICÇÃO])
```

```
    MODIFY (coluna TIPO [DEFAULT valor])
```

```
    DROP COLUMN coluna
```

```
    ADD CONSTRAINT nome CONSTRAINT [ENABLE | DISABLE]
```

```
    DROP CONSTRAINT nome_da_constraint
```

```
    ENABLE | DISABLE CONSTRAINT nome_constraint
```

```
DROP TABLE tabela [CASCADE CONSTRAINTS]
```