

Prática em Banco de Dados

Henry Costa Ungaro
Banco de Dados
UNIP

A partir do exemplo da tabela pessoa:

```
CREATE TABLE PESSOA
```

```
(
```

```
id_pessoa integer constraint id_pessoa_PK primary key,
```

```
nome varchar2(20),
```

```
cpf varchar2(14),
```

```
nascimento date
```

```
);
```

Com os dados:

```
INSERT INTO PESSOA VALUES (1,'Luis da Silva','11109865424','20-06-1994');
INSERT INTO PESSOA VALUES (2,'Maria Joaquina','87604465334','20-07-1981');
INSERT INTO PESSOA VALUES (3,'José da Silva','54279835424','08-03-1999');
INSERT INTO PESSOA VALUES (4,'João Henrique','283835424','28-08-1981');
INSERT INTO PESSOA VALUES (5,'Eliézio Mesquita','1087865424','28-02-1985');
INSERT INTO PESSOA VALUES (6,'Gustavo Souza','7464465334','17-12-1995');
INSERT INTO PESSOA VALUES (7,'Wesley Yamazack','54329835424','12-10-1997');
INSERT INTO PESSOA VALUES (8,'Edson Arantes','90874465334','02-04-2005');
INSERT INTO PESSOA VALUES (9,'Silvio Santos','54279835424','17-07-1955');
INSERT INTO PESSOA VALUES (10,'Zico Galinho','654838724','15-11-2011');
INSERT INTO PESSOA VALUES (11,'Irá Almeida','1087865424','07-09-2000');
INSERT INTO PESSOA VALUES (12,'Arnoldo Swaze Credo','798465334','25-05-1974');
INSERT INTO PESSOA VALUES (13,'Carlos Alberto','12329655424','03-01-1965');
```

- O operador LIKE é utilizado para buscar por uma determinada string dentro de um campo com valores textuais. Com ele podemos, por exemplo, buscar os registros cujo NOME inicia com uma determinada palavra, ou contém um certo texto.
- Para efetuar esse tipo de consulta, podemos utilizar também o caractere % para indicar um "coringa" (wildcard), ou seja, um texto qualquer que pode aparecer no campo. Sua sintaxe padrão é a seguinte:
- `SELECT colunas FROM tabela WHERE campo LIKE 'valor'`

- Nessa instrução, o "valor" pode ser informado de várias formas:
 - texto: Nesse caso, serão retornados todos os registros que contêm no campo buscado exatamente o "texto" informado no filtro. O funcionamento aqui é equivalente a utilizar o operador de igualdade (=);
 - %texto%: Serão retornados os registros que **contêm** no campo buscado o "texto" informado. Por exemplo, podemos buscar os nomes que contêm "Santos", ou que contêm uma sílaba ou letra específica. O registro com nome "Luis da Silva", por exemplo, contém o termo "da", então atenderia ao filtro '%da%';
 - %texto: Serão retornados os registros cujo valor do campo filtrado **termina com** o "texto" informado. O %, nesse caso, indica que pode haver qualquer valor no começo do campo, desde que ele termine com o "texto". Por exemplo, o registro com nome "Luis da Silva" atenderia ao filtro '%Silva';
 - texto%: Serão retornados os registros cujo valor do campo filtrado **começa com** o "texto" informado. Dessa vez, o % indica que após o "texto" pode haver qualquer valor. Por exemplo, o registro com nome "Luis da Silva", atenderia ao filtro 'Luis%'.

Considerando nossa tabela de exemplo, se executarmos a seguinte consulta, teremos como resultado os registros que aparecem :

```
SELECT * FROM PESSOA WHERE Nome LIKE 'J%';
```

ID_PESSOA	NOME	CPF	NASCIMENTO
4	João Henrique	283835424	28/08/81
3	José da Silva	54279835424	08/03/99

Além do %, existe ainda o símbolo especial underscore ou sublinhado (_), com o qual podemos efetuar filtros mais exatos sobre as colunas textuais, buscando por um termo em uma posição específica do texto. O underscore indica a quantidade de casas/caracteres antes ou depois do texto buscado. A seguir temos alguns exemplos de uso para que possamos compreender sua sintaxe:

- '_este': Filtra os registros que contém 1 caractere qualquer no começo e em seguida o termo 'este'. Por exemplo, seriam retornados registros contendo o valor 'teste', 'peste', 'veste';
- 'b_m': Filtra os registros que comecem com a letra "b", contenham 1 caractere em seguida, e depois a letra "m". Nesse caso, atenderiam a esse filtro, por exemplo, os valores "bom", "bem", "bPm", etc.
- '_u%': Filtra os registros cujo campo especificado comece com um caractere qualquer, em seguida contenha uma letra "u", e depois qualquer valor. Por exemplo, os valores "Luis da Silva" e "Gustavo" atenderiam a esse filtro.

O seguinte script pode ser executado sobre nossa tabela para que possamos visualizar o funcionamento do underscore na prática.

```
Select * from Pessoa Where Nome LIKE '_i%';
```

ID_PESSOA	NOME	CPF	NASCIMENTO
9	Silvio Santos	54279835424	17/07/55
10	Zico Galinho	654838724	15/11/11

O operador IN é utilizado quando desejamos consultar uma tabela, filtrando o valor de um de seus campos a partir de uma lista e possibilidades. Enquanto o operador de comparação de igualdade (=) avalia se os dois valores são iguais, o IN permite verificar se o valor de um campo se encontra em uma lista. Sua sintaxe é a seguinte:

```
SELECT campos FROM tabela WHERE campo IN (valor1, valor2, valor3);
```

```
SELECT * FROM Pessoa WHERE Id_Pessoa IN (2, 3, 7);
```

ID_PESSOA	NOME	CPF	NASCIMENTO
2	Maria Joaquina	87604465334	20/07/81
3	José da Silva	54279835424	08/03/99
7	Wesley Yamazack	54329835424	12/10/97

Esse operador é usado quando precisamos recuperar as linhas de uma tabela cujo valor de um campo encontra-se em um intervalo especificado.

```
SELECT campos FROM tabela WHERE campo BETWEEN  
inicio_intervalo AND fim_intervalo;
```

O seguinte script, cujo resultado é apresentado, retorna os registros da tabela PESSOA cuja data de nascimento esteja entre 01/01/1981 e 31/12/1990:

```
SELECT * FROM Pessoa WHERE Nascimento BETWEEN '01-01-1981' AND '31-12-1990'
```

ID_PESSOA	NOME	CPF	NASCIMENTO
2	Maria Joaquina	87604465334	20/07/81
4	João Henrique	283835424	28/08/81
5	Eliézio Mesquita	1087865424	28/02/85

OBRIGADO