

# **ETEC – Cidade do Livro**

## **Banco de Dados II**

### **Aula 2**



**Prof. Me. Marco Aurelio M. Antunes**

# Operadores Especiais

```
create table pessoa(  
id_pessoa      int      identity,  
nome           varchar(40) not null,  
fone           varchar(20) null,  
email          varchar(40) null,  
dtnasc         datetime  not null,  
salario        decimal(7,2) not null  
primary key(id_pessoa)  
);
```

```
insert into pessoa values ('Getulina',NULL,NULL,'30/06/2000',1200);  
insert into pessoa values ('Geovalda','3262-8890',NULL,'01/02/2000',1900);  
Insert into pessoa values ('Graça','99798-1234','graça@uol.com.br','10/12/1990',5200);  
insert into pessoa values ('Natalina','3261-1234',NULL,'31/10/2019',1990);  
insert into pessoa values ('Uvina','99988-1234','uvina@uol.com.br','03/11/1995',900);  
insert into pessoa values ('Rosalina','97878-8787','rosa@uol.com.br','03/11/1989',9000);  
select * from pessoa;
```

# Operadores Especiais

São operadores utilizados para determinar melhor as linhas que queremos filtrar.

## IS NULL

Comando utilizado para saber se o conteúdo da coluna foi ou não inicializado. As colunas que não tem valor inicializado são nulas.

-- mostra se o campo telefone não foi inicializado

select nome, fone

from pessoa

where fone is null;

## IS NOT NULL

Comando utilizado para saber se o conteúdo da coluna foi inicializado.

-- mostra se o campo telefone foi inicializado

select nome, fone

from pessoa

where fone is not null;



# BETWEEN

Operador utilizado para determinar um intervalo de busca de forma simplificada.

-- Mostra pessoas que fazem nascidas em um intervalo de tempo

```
select nome, dtnasc  
from pessoa  
where dtnasc between '01/01/1990' and '01/01/2000';
```

-- Mostra pessoas em um intervalo entre salario

```
select nome, salario  
from pessoa  
where salario between 1900 and 2000;
```

-- Mostra pessoas fora de um intervalo de salario

```
select nome, salario  
from pessoa  
where salario not between 1900 and 2000;
```

# LIKE

Operador compara cadeias de caracteres, utilizando padrões de comparação para um ou mais caracteres.

**%** substitui um ou mais caracteres – coringa

**\_** substitui um caracter

Utilizando a combinação desses caracteres especiais pode se conseguir uma variedade grande de expressões:

LIKE 'A%' -- todas as palavras que iniciem com a letra A

LIKE '%A' -- todas as palavras que terminem com a letra A

LIKE '%A%' -- todas as palavras que tenham A em qualquer posição

LIKE 'A\_' -- string de 2 caracteres que tenham a primeira letra A e o segundo caractere seja qualquer outro

LIKE '\_A' -- string de 2 caracteres cujo primeiro caractere seja qualquer um e a última letra seja A

--Mostra pessoas que o nome comece com G

```
select * from pessoa  
where nome like 'G%';
```

-- Mostra pessoas que tenham a letra E na segunda posição da palavra

```
select * from pessoa  
where nome like '_e%';
```

-- pessoas que tenham a letra N no inicio e a letra T na terceira posição

```
select * from pessoa  
where nome like 'N_t%';
```

-- Mostra pessoas que tenham a letra N em algum lugar do nome

```
select * from pessoa  
where nome like '%n%';
```

-- Mostra pessoas com salario iniciado com 1

```
select nome, salario  
from pessoa  
where salario like '1%';
```



# IN

Operador que compara o valor de uma coluna com um conjunto de valores.

-- Mostra pessoas que constem na lista

```
select * from pessoa  
where id_pessoa in(1,3,5);
```

-- Mostra pessoas que constem na lista

```
select * from pessoa  
where nome in('Graça','Geovalda');
```

## NOT BETWEEN / NOT LIKE / NOT IN

Operadores que invertem o resultado da expressão, negando o resultado da condição, mostrando o resultado inverso.

-- Mostra pessoas que fazem nascidas fora de um intervalo de tempo

```
select nome, dtnasc  
from pessoa  
where dtnasc not between '01/01/1990' and '31/12/2000';
```

-- Mostra pessoas que o nome não comece com G

```
select * from pessoa  
where nome not like 'G%';
```

-- Mostra pessoas que não constem na lista

```
select * from pessoa  
where id_pessoa not in(1,3,5);
```





# Exercícios

- 1 - Identificar algum registro campo NULL
- 2 - Identificar os demais registros do mesmo campo NOT NULL
- 3 - Listar médicos com salario entre 1000 e 2000
- 4 - Utilizando like mostrar médicos
  - NOME começa com U
  - NOME começa com UVINA
  - NOME com a segunda letra V
  - NOME com a letra A em qualquer posição
- 5 – Listar médicos com código 1, 4 ou 8.