

ROTEIRO DE AULA PRÁTICA

TÍTULO DA AULA PRÁTICA: Condição e Estrutura de Repetição

Objetivos

- Entender na prática o funcionamento da condição e estrutura de repetição no banco de dados.
- Capacitar a resolução de problemas que requerem o uso de decisão e estrutura de repetição no banco de dados.
- Uso de decisão e estrutura de repetição em casos simples como prova de conceito.

Função IFF

A função IFF() é uma estrutura condicional que retorna um valor se a condição for verdadeira e outro valor se a condição for falsa. A função está presente a partir da versão SQL Server 2012.

Segue a sua sintaxe:

```
IFF(condicao, valor_caso_verdadeiro, valor_caso_falso)
```

Onde:

- condicao: Consiste no valor a ser testado.
- valor_caso_verdadeiro: Consiste no valor de retorno caso a condição seja verdadeira.
- valor_caso_falso: Consiste no valor de retorno caso a condição seja falsa.

Exemplo de uma condição que verifica se 100 é maior que 1000, retorna “verdadeiro” ou “falso”.

```
SELECT IFF(100>1000, 'verdadeiro', 'falso')
```

Exemplo de uma condição que verifica se um texto é igual a outro.

```
SELECT IFF('amarelo'='yellow', 'igual', 'diferente') as comparador
```

Exemplo de uma condição que verifica se existe email numa coluna da tabela de cliente.

```
SELECT id, nome, IFF(email is null, 'não há email', 'há email')  
FROM cliente
```

Condição

Quando é necessário executar um código de forma condicional com base na avaliação de uma expressão, é possível utilizar a lógica IF.

Segue a sua sintaxe:

```
IF expressao_booleana
    {declaracao_sql | bloco_declaracao}
[ ELSE
    {declaracao_sql | bloco_declaracao}
]
```

É possível criar IF's aninhados, um dentro de outro.

Um exemplo é testar se uma variável tem determinado valor, antes de executar a sua impressão.

```
DECLARE @variavel INT = 1;
IF @variavel = 1
    PRINT '@variavel = 1';
PRINT 'Fim de execução!';
```

É possível também criar um bloco de execução dentro da condicional, com os comandos BEGIN e END.

```
DECLARE @contador int = 1;
IF @contador = 1
BEGIN
    PRINT 'Esta instrução será executada apenas quando';
    PRINT 'o contador for igual a 1';
END
```

Estrutura Case When

É uma boa prática de programação utilizar os operadores CASE-WHEN quando é necessário várias condições utilizando a mesma variável na condicional.

Segue a sua sintaxe:

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

Por exemplo, para criar um ranking dos clientes que mais comprem no sistema de ecommerce da Figura 1.

```

select
  c.nome,
  c.email,
  case
    when sum(p.total) < 100 then 'Inativo'
    when sum(p.total) < 1000 then 'Ativo'
    when sum(p.total) < 10000 then 'Prospect'
    when sum(p.total) < 100000 then 'VIP'
  else
    'Parceiro'
  end as classificacao_cliente
from cliente c, pedido p
where c.id = p.id_cliente
group by c.nome, c.email

```



Figura 1: Diagrama de entidade e relacionamento de um sistema para *ecommerce*.

Estrutura de Repetição

É possível usar a estrutura de repetição WHILE no código T-SQL.

Segue a sintaxe básica do WHILE.

```

WHILE expressao_booleana
{ sql_statement | statement_block | BREAK | CONTINUE }

```

Enquanto a expressão booleana for verdadeira, a estrutura de repetição continuará executando, a nova expressão será reavaliada e uma nova iteração será executada. Caso a expressão booleana já inicie com valor falso, nenhuma iteração é executada.

Use o comando BREAK para sair programaticamente de dentro da iteração antes da expressão for novamente validade.

Use o comando CONTINUE para continuar o processamento da iteração para validação da próxima expressão.

Um exemplo simples de utilização do WHILE é criar números de 0 a 100.000 armazenar em uma tabela.

```
DECLARE @contador INT = 0;
CREATE TABLE tabela (numero int);
WHILE @contador < 100000
BEGIN
    INSERT INTO tabela VALUES (@contador);
    SET @contador += 1;
END
SELECT * FROM tabela;
```

Uma variável @contador foi declarada como inteira e instanciada com valor zero. Uma tabela foi criada com apenas uma coluna numérica para armazenar os valores a serem criados. Enquanto o contador for menor que 10.000, então o procedimento dentro do bloco BEGIN e END será executado. Os procedimentos envolvem a inclusão de um registro na tabela e o incremento do contador. Por fim, uma seleção dos registros na tabela é executada.

Um exemplo de uso do comando CONTINUE pode ser para separar números pares e inserir apenas os números ímpares.

```
DECLARE @contador INT = 0;
CREATE TABLE tabela (numero int);
WHILE @contador < 100000
BEGIN
    SET @contador += 1;

    IF @contador % 2 = 0
        CONTINUE;
    ELSE
        INSERT INTO tabela VALUES (@contador);
END
SELECT * FROM tabela;
```

Exercícios

1. Utilizando o DER da Figura 1:
 - a) Crie uma tabela e insira números pares de 2 a 100.000.
 - b) Crie uma tabela e insira os números de 1 a 20.000 e a informação se o número é par ou ímpar ("IMPAR", "PAR").
 - c) Crie uma tabela e insira os valores de uma sala de aula ("Sala 101", "Sala 102", ..., "Sala 192"). Use estrutura de repetição WHILE.

Referência Bibliográfica

FEITOSA, M. P. **Fundamentos de Banco de Dados.** Uma abordagem prático-didática. ed. 1, São Paulo: Independente, 2013.

Trabalhando com Whiles e IFs no T-SQL. Disponível em: <
<https://www.devmedia.com.br/trabalhando-com-whiles-e-ifs-no-t-sql/31372>>.

Acessado em: 22 fevereiro. 2023.