

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

SQL – Structure Query Language Linguagem Estruturada de Consulta

Exemplos de UPDATE, INSERT, DELETE, e SELECT:

SUMÁRIO

1* COMANDO UPDATE * ALTERAÇÃO de registro(s) *	1
2* COMANDO DELETE * EXCLUSÃO de registro(s) *	3
3* COMANDO INSERT * Inserção de registro nos TABLES *	4
4* COMANDO SELECT * SELEÇÃO/FILTRO de registro(s) *	5

1* COMANDO UPDATE * ALTERAÇÃO de registro(s) *

```
/* Altera ESTOQUE todos os produtos
com ESTOQUE NEGATIVO, para estoque IGUAL A ZERO */
UPDATE PRODUTO
  SET QTDE_PROD=0, QTD_FRAC_PRO=0
  WHERE QTDE_PROD<0;
```

```
/* Alterando preço, Aumentando o preço de todos produtos em 5% */
UPDATE PRODUTO
  SET PRE_PROD=PRE_PROD*1.05
```

```
/* Alterando o NOME DO PRODUTO 35 e seu PREÇO */
UPDATE PRODUTO
  SET NOM_PRODUTO = 'Nome Novo' , PRE_PRODUTO = 4.5
  WHERE COD_PRODUTO = 35
```

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

/* Alterando os dados do cliente 32/1 */

Update CLIENTE**Set**

NOME = 'Francisco Xavier' ,

ENDereco = 'Rua dos Poderes, 777' ,

BAIRRO = 'Centro' ,

CEP = '37890-000'

WhereCOD_CLI = 32 **and**

LOJA_CLI= 1

Código de ALTERAÇÃO dos dados do CLIENTE, via programação EM DELPHI :With F_Dm_Pedidos.ZQRY_Livre.SQL do
Begin

Clear;

//

// Código de ALTERAÇÃO na table de CLIENTES ;

Append('Update CLIENTE');

Append('Set');

Append('NOME = ' + QuotedStr(Ed_NOME2.Text) + ' , ');

Append('ENDERECO = ' + QuotedStr(Ed_ENDERECO2.Text) + ' , ');

Append('BAIRRO = ' + QuotedStr(Ed_BAIRRO2.Text) + ' , ');

Append('CEP = ' + QuotedStr(Ed_CEP2.Text));

Append('Where');

Append('COD_CLI='+F_Dm_Pedidos.Qry_Clientes.FieldName('COD_CLI').asString);

Append(' and ');

Append('LOJA_CLI='+F_Dm_Pedidos.Qry_Clientes.FieldName('LOJA_CLI').asString);

// Tenta ALTERAR o registro ;

Try

Showmessage('ALTERAÇÃO : Veja a query : '+#13+#13+

'"+F_Dm_Pedidos.ZQRY_Livre.SQL.Text+"');

F_Dm_Pedidos.ZQRY_Livre.ExecSQL; // Trava o registro . (destrava no COMMIT)

Except On E:Exception do

Begin

Application.MessageBox(Pchar(

'Algum usuário está "TRAVANDO" o REGISTRO ATUAL'+#13+#13+

'ERRO : "' + E.Message + ' " ;'),

'Impossível ALTERAR', MB_Ok);

//

// F_Dm_Dados.ZConnection.Rollback; => Não precisa do ROLLBACK ,

// pois SE DEU EXCEÇÃO, não CHEGOU A "ABRIR transação"

EXIT; // Sai de uma "possível procedure"

End;

End; // Fim do TRY ... EXCEPT ... END ;

//

// GRAVA FISICAMENTE, finaliza transação, e destrava o registro

F_Dm_Pedidos.ZConnection.Commit;

// Após alterar, DÁ UM "REFRESH" na table, para mostrar dados alterados ;

F_Dm_Pedidos.Qry_Clientes.Refresh;

End;

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

2* COMANDO DELETE * EXCLUSÃO de registro(s) ***** CUIDADO ; Sem a cláusula WHERE, todos registros serão apagados ****

```
/* APAGANDO todos CLIENTES da tabela . CUIDADO !!! */
DELETE FROM CLIENTE
```

```
/* APAGANDO o produto que tem CÓD.=23 */
DELETE FROM PRODUTO
WHERE (COD_PRODUTO=23);
```

```
/* APAGANDO o produto com estoque NEGATIVO */
DELETE FROM PRODUTO
WHERE QTDE_PROD<0;
```

```
/* APAGANDO o produto com Cód.Grupo=2 */
DELETE FROM PRODUTO
WHERE COD_GRUPO=2;
```

```
/* APAGANDO o CLIENTE com (cod_cli=32 e Loja_Cli=1) */
Delete From CLIENTE
Where
COD_CLI= 32
and
LOJA_CLI= 1
```

Código de EXCLUSÃO do registro do CLIENTE, via programação EM DELPHI :

```
With F_Dm_Pedidos.ZQRY_Livre.SQL do
Begin
    Clear;
    //
    // Código de EXCLUSÃO na table de CLIENTES ;
    Append('Delete From CLIENTE');
    Append('Where' );
    Append(' COD_CLI='+F_Dm_Pedidos.Qry_Clientes.FieldByName('COD_CLI').asString );
    Append(' and ' );
    Append(' LOJA_CLI='+F_Dm_Pedidos.Qry_Clientes.FieldByName('LOJA_CLI').asString );
    //
    // Tenta EXCLUIR o registro ;
    Try
        Showmessage('Veja a query : '+#13+#13+
            ' "'+F_Dm_Pedidos.ZQRY_Livre.SQL.Text+'"');
        // EXIT;
        F_Dm_Pedidos.ZQRY_Livre.ExecSQL; // Trava o registro . (destrava no COMMIT)
    Except On E:Exception do
        Begin
            Application.MessageBox( Pchar(
                'Algum usuário está "TRAVANDO" o REGISTRO ATUAL'+#13+#13+
                'ERRO : "'+ E.Message +' " ;' ),
                'Impossível EXCLUIR', MB_Ok);
            //
            // F_Dm_Dados.ZConnection.Rollback; => Não precisa do ROLLBACK ,
            // pois SE DEU EXCEÇÃO, não CHEGOU A "ABRIR transação"
            EXIT; // Sai da procedure ;
        End;
    End;
    //
    // GRAVA FISICAMENTE, finaliza transação, e destrava o registro
    F_Dm_Pedidos.ZConnection.Commit;
End;
Close; // Sai do FORM , após excluir o CLIENTE ATUAL ;
```

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

3* COMANDO INSERT * Inserção de registro nos TABLES *

/* Insere produto, na tabela de produtos, enviando cód.produto */

```
INSERT INTO Produto
( Cod_Pro, Nom_Pro, Pre_Venda, Pre_Compra, Qtd_Estoq )
VALUES
( 35 , 'PIA', 300.50 , 220.35 , 10 )
```

/* Insere produto, SEM CÓDIGO ,na tabela de produtos

Ao inserir SEM CÓDIGO do PRODUTO,

=> Se o CÓD.PRODUTO utiliza o campo auto-incremento, assumirá o valor do "generator".

Senão, assumirá NULO e dará um erro (exceção), por cod.cli ser NOT NULL e CHAVE PRIMARIA */

```
INSERT INTO Produto
( Nom_Pro, Pre_Venda, Pre_Compra, Qtd_Estoq )
VALUES
( 'PIA', 300.50 , 220.35 , 10 )
```

INSERT INTO CLIENTE

```
( COD_CLI, LOJA_CLI , NOME, ENDERECO, BAIRRO, CEP)
VALUES
( null , 1 , 'José dos Reis' , 'Rua do José, 35' , 'CENTRO', '37890-000' )
```

Código de INCLUSÃO do registro do CLIENTE, via programação EM DELPHI :

With F_Dm_Pedidos.ZQRY_Livre.SQL do

Begin

```
Clear;
//
// Código de INSERÇÃO na table de CLIENTES ;
Append('Insert into CLIENTE');
Append('( COD_CLI, LOJA_CLI , NOME, ENDERECO, BAIRRO, CEP)' );
Append('Values' );
//
// QUOTEDSTR(X) => "Envolve" a variável "X", com aspas simples => X tem que ser STRING ;
Append( '( null , 1 , ' );
Append( QuotedStr(Ed_Nome2.Text) + ',' );
Append( QuotedStr(Ed_Endereco2.Text) + ',' );
Append( QuotedStr(Ed_Bairro2.Text) + ',' );
Append( QuotedStr(Ed_Cep2.Text) );
Append( ')' );
//
// TENTA INCLUIR o registro DIGITADO ;
Try
    Showmessage('Veja a query : '+#13+#13+
        '''+F_Dm_Pedidos.ZQRY_Livre.SQL.Text+''');
    F_Dm_Pedidos.ZQRY_Livre.ExecSQL; // Trava o registro . (destrava no COMMIT)
Except On E:Exception do
    Begin
        Application.MessageBox( Pchar(
            'Algum usuário está "TRAVANDO" o REGISTRO ATUAL'+#13+#13+
            'ERRO : '+ E.Message + ' ' ),
            'Impossível INCLUIR', MB_Ok);
        //
        // F_Dm_Dados.ZConnection.Rollback; => Não precisa do ROLLBACK ,
        // pois SE DEU EXCEÇÃO, não CHEGOU A "ABRIR transação"
        Result := False; EXIT;
    End;
End;
//
// GRAVA FISICAMENTE, finaliza transação, e destrava o registro
F_Dm_Pedidos.ZConnection.Commit;
```

End;

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

4* COMANDO SELECT * SELEÇÃO/FILTRO de registro(s) ***Onde procurar mais detalhes SOBRE SELECT :**

* Para mais detalhes, vá no arquivo de 850 dicas de DELPHI e , na guia PESQUISAR, procure por AVG. Lá encontrará uma APOSTILA de SELECT, com vários exemplos ;

SINTAXE BÁSICA :

```
Select [campos]
From [Tabela(s)]
Where [condição]
```

```
/* Mostra todos campos da tabela de produtos */
```

```
SELECT * FROM PRODUTO
```

```
/* Mostra campo NOM_PRO, e PRE_VENDA da tabela de produtos,
somente para produtos do GRUPO=7 */
```

```
SELECT NOM_PRO, PRE_VENDA
FROM PRODUTO
WHERE GRUPO=7
```

```
/* Mostra todos CAMPOS da tabela de ITENS da venda número 51 */
```

```
SELECT * FROM ITENS
WHERE COD_VENDA=51
```

```
/* Mostra qtde de registros da tabela de produtos */
```

```
SELECT COUNT(*) FROM PRODUTO
```

```
/* Mostra qtde de produtos CADASTRADOS, do grupo=3 */
```

```
SELECT COUNT(*) FROM PRODUTO
WHERE grupo=3
```

Funções Agregadas (ou de Agrupamento)

Função	Retorno
(avg(n))	média do valor n, ignorando nulos
count(expr)	vezes que o número da expr avalia para algo não nulo
max(expr)	maior valor da expr
min(expr)	menor valor da expr
sum(n)	soma dos valores de n, ignorando nulos

```
/* Mostra o total DO CAMPO QTE_PRO, ou seja, o total do estoque , de todos produtos cadastrados
*/
```

```
SELECT SUM( QTE_PRO ) FROM PRODUTO
```

```
/* Mostra a MÉDIA do CAMPO PRE_PRO, ou seja,
A MÉDIA de PREÇO, de todos produtos cadastrados */
```

```
SELECT AVG( PRE_PRO ) FROM PRODUTO
```

```
/* Média de preço dos produtos do GRUPO 3 */
```

```
SELECT AVG( PRE_PRO ) FROM PRODUTO
WHERE COD_GRUPO=3
```

```
/* MENOR PREÇO de preço dos produtos do GRUPO 5 */
```

```
SELECT MIN( PRE_PRO ) FROM PRODUTO
WHERE COD_GRUPO=5
```

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

CLÁUSULA LIKE e "%" → Permite pesquisar por PARTE de um campo, com auxílio do "%"

```
/* Seleciona todos clientes com nome iniciado com "ANT" */
SELECT NOME FROM CLIENTE
WHERE NOME LIKE 'ANT%'
```

```
/* Seleciona todos clientes com a palavra "MARIA" */
SELECT * FROM CLIENTE
WHERE NOME LIKE '%MARIA%'
```

CLÁUSULA "GROUP BY" → Condições AGRUPADAS

```
/* Seleciona a MÉDIA salarial DOS CLIENTES , SEPARADOS POR LOJA */
SELECT LOJA_CLI, AVG( SALARIO )
From CLIENTE
Group by LOJA_CLI
```

CLÁUSULA "ESPECIAL" → Having + Group By

A cláusula "HAVING" pode ser utilizada para especificar quais grupos deverão ser exibidos, portanto restringindo-os.

```
/* Seleciona a MÉDIA salarial DOS CLIENTES ,
SEPARADOS POR LOJA
→ FILTRADOS SOMENTE PELA LOJA 2 (FILTRADOS COM WHERE, AÍ NÃO É POR GRUPO) */
SELECT LOJA_CLI, AVG( SALARIO )
From CLIENTE
Where LOJA_CLI=3
Group by LOJA_CLI

/* Seleciona a MÉDIA salarial DOS CLIENTES , SEPARADOS POR LOJA,
FILTRANDO COM HAVING, (HAVING => porque neste caso, é uma condição "agrupada")
SOMENTE OS SALÁRIO > 300,00 (HAVING) */
SELECT LOJA_CLI, AVG( SALARIO )
From CLIENTE
Where LOJA_CLI=3
Group by LOJA_CLI
Having AVG( SALARIO )>300
```

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

O Select "super-complexo" abaixo foi criada com o IBEASY,
Opção FERRAMENTAS → MODO QBE (Query Builder Expert)

```
SELECT /* Seleccione os campos ... */
"PEDIDOS"."COD_PEDIDO" as "Pedido", /* Cód.pedido de pedidos, mostrando "Pedido" */
"PEDIDOS"."DATA_PEDIDO" as "Data", /* Data.pedido de pedidos, mostrando "Data" */
"CLIENTE"."NOME" as "Cliente", /* Nome/cliente, mostrando "Cliente" */
Sum(QTDE_ITEN*VALOR_ITEN ) as "Total do Pedido",
"PEDIDOS"."VALOR_PAGO" as "Valor já Pago"

FROM "ITEN_PEDIDO", "CLIENTE", "PEDIDOS" /* das tabelas ... */

WHERE /* Enquanto ... */
"PEDIDOS"."LOJA_CLI"="CLIENTE"."LOJA_CLI" /* Loja Cli de Pedidos = Loja Cli de Clientes */
AND "PEDIDOS"."COD_CLI"="CLIENTE"."COD_CLI" /* Cod Cli de Pedidos = Cod Cli de Clientes */
AND "ITEN_PEDIDO"."COD_PED"="PEDIDOS"."COD_PEDIDO" /* Cod Ped de Iten_Pedido = Cod Ped de Pedidos */

GROUP BY /* Agrupada por CÓD.PEDIDO, DATA DO PEDIDO, NOME DO CLIENTE e VALOR PAGO */
"PEDIDOS"."COD_PEDIDO",
"PEDIDOS"."DATA_PEDIDO",
"CLIENTE"."NOME",
"PEDIDOS"."VALOR_PAGO"
```

(VEJA A PRÓXIMA PÁGINA)

UPDATE, INSERT DELETE e SELECT

Autor : Carlos Antônio Ferreira da Silva. carlosmuzambinho@yahoo.com.br

Veja o que o **IBEASY** facilmente nos proporciona :**Guia RESULTADOS :**

Consulta por Exemplo

QBE

CLIENTE
FUNCIONARIOS
ITEN_PEDIDO
PEDIDOS
PRODUTOS

CLIENTE

- COD_CLI
- LOJA_CLI
- NOME
- ENDERECO
- BAIRRO
- CEP

PEDIDOS

- COD_PEDIDO
- DATA_PEDIDO
- VALOR_PAGO
- COD_CLI
- LOJA_CLI
- COD_FUN

ITEN_PEDIDO

- COD_ITEM
- COD_PRO
- COD_PED
- NOM_PRODUTO
- QTDE_ITEM
- VALOR_ITEM

Pedido	Data	Cliente	Total do Pedido	Valor já Pago
3	03/11/2005	José da Silva Xavi	42,9	Null
5	03/11/2005	Maria da Silvax	22	Null

Campos Resultados SQL

Arquivo QBE corrente: D:\Meus documentos - Carlos\Curso PDD EAFM\Curso - Delphi\Pedidos - Firebird\Pedidos\Clientes e seus Pedidos.QBE

Sair

Guia CAMPOS :

Campo	COD_PEDIDO:"Pedido"	DATA_PEDIDO:"Data"	NOME:"Cliente"	QTDE_ITEM*VALOR_ITEM : "Total do Pedido"	VALOR_PAGO:"Valor já
Tabela	PEDIDOS	PEDIDOS	CLIENTE		PEDIDOS
Exibe	Sim	Sim	Sim	Sim	Sim
Restrições					
Pesquisa (Sc					
Função				Sum	
Grupo	Sim	Sim	Sim	Não	<input checked="" type="checkbox"/>

Campos Resultados SQL

Observação :

SELECTS complexos como o acima, devem ser gerados com aplicativos como **IBEASY**, que facilitam a vida do programador.

O **IBEASY**, na parte de "geração de Selects de modo visual", e na parte de IMPRESSÃO de DER (Diagrama Entidade-Relacionamento) é muito bom.

IBEXPERT versão Personal (free) :

Para alterar/criar Tables, Stored Procedures, Triggers, aconselho o **IBEXPERT versão Personal (free)**, por trazer detalhadamente os SCRIPTS ;

Como "chegar lá ." : **IBEASY**, menu FERRAMENTAS → MODO QBE (Query Builder Expert)