

Biblioteca do Programador Harbour

HBMYSQL

Guia de referência

**Versão
2010.07.30**

Marcos Antonio Gambeta

Sumário

Licença	4
Donativos	4
Contato	4
Introdução	5
Exemplos	6
Usando as classes	6
Conectando com o servidor	6
Selecionando um banco de dados	6
Criando uma tabela	6
Apagando uma tabela	6
Executando uma consulta	6
Encerrando a conexão com o servidor	7
Verificando a conexão	7
Usando as funções	7
Conectando com o servidor	7
Criando um banco de dados	7
Selecionado um banco de dados	8
Apagando um banco de dados	8
Listando os bancos de dados	8
Criando uma tabela	8
Apagando uma tabela	9
Alterando uma tabela	9
Listando as tabelas	9
Inserindo dados na tabela	9
Alterando dados na tabela	9
Apagando dados da tabela	9
Consultando dados	10
Classes	10
TBColumnSQL	10
TBrowseSQL	10
TMySQLQuery	11
TMySQLRow	12
TMySQLServer	12
TMySQLTable	13
Funções	15
MYSQL_AFFECTED_ROWS	15
MYSQL_CLOSE	15
MYSQL_COMMIT	15
MYSQL_DATA_SEEK	16
MYSQL_ERROR	16
MYSQL_ESCAPE_STRING	16
MYSQL_ESCAPE_STRING_FROM_FILE	16
MYSQL_FETCH_FIELD	17
MYSQL_FETCH_ROW	17
MYSQL_FIELD_COUNT	17
MYSQL_FIELD_SEEK	18
MYSQL_FREE_RESULT	18
MYSQL_GET_HOST_INFO	18
MYSQL_GET_SERVER_INFO	19
MYSQL_GET_SERVER_VERSION	19

<u>MYSQL_LIST_DBS</u>	19
<u>MYSQL_LIST_FIELDS</u>	19
<u>MYSQL_LIST_TABLES</u>	20
<u>MYSQL_NUM_FIELDS</u>	20
<u>MYSQL_NUM_ROWS</u>	20
<u>MYSQL_QUERY</u>	21
<u>MYSQL_REAL_CONNECT</u>	21
<u>MYSQL_ROLLBACK</u>	21
<u>MYSQL_SELECT_DB</u>	22
<u>MYSQL_STORE_RESULT</u>	22
<u>MYSQL_USE_RESULT</u>	22
<u>Funções obsoletas</u>	23
<u>DATATOSQL</u>	23
<u>FILETOSQLBINARY</u>	23
<u>SQLAFFROWS</u>	23
<u>SQLAND</u>	23
<u>SQLCLOSE</u>	23
<u>SQLCOMMIT</u>	24
<u>SQLCONNECT</u>	24
<u>SQLDATAS</u>	24
<u>SQLFETCHF</u>	24
<u>SQLFETCHR</u>	24
<u>SQLFICOU</u>	24
<u>SQLFREER</u>	25
<u>SQLFSEEK</u>	25
<u>SQLGETERR</u>	25
<u>SQLHOSTINFO</u>	25
<u>SOLLISTDB</u>	25
<u>SOLLISTF</u>	26
<u>SOLLISTTBL</u>	26
<u>SQLNROWS</u>	26
<u>SQLNUMFI</u>	26
<u>SQLQUERY</u>	26
<u>SQLROLLBACK</u>	26
<u>SQLSELECTD</u>	27
<u>SQLSRVINFO</u>	27
<u>SQLSTORER</u>	27
<u>SQLUSERES</u>	27
<u>SQLVERSION</u>	27
<u>Ferramentas</u>	28
<u>DBF2MYSQL</u>	28

Licença

Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-nd/2.5/br/> ou envie uma carta para Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Donativos

Embora este guia seja distribuído livremente, espera-se daqueles que o adotarem como fonte de consulta que apoiem o projeto através de donativos. Este apoio ajudará na continuidade e na qualidade deste trabalho, permitindo um investimento maior de tempo no seu desenvolvimento.

Contato

Para entrar em contato com o autor, envie sua mensagem para o *email* abaixo:

marcosgambeta@gmail.com

Introdução

HBMYSQL é uma das bibliotecas disponíveis no projeto Harbour. Sua função é permitir a comunicação entre um programa desenvolvido em Harbour com o sistema gerenciador de banco de dados (SGDB) MySQL Server.

O objetivo deste guia não é ensinar SQL ou como usar o MySQL Server, mas sim documentar e demonstrar os recursos presentes na biblioteca. Aqui você encontrará informações úteis que lhe ajudarão a usar a biblioteca no desenvolvimento de sistemas.

As informações estão divididas nas seguintes seções:

[Exemplos](#)

Uma série de exemplos simples demonstrando como usar tanto as classes quanto as funções.

[Classes](#)

Relação das classes.

[Funções](#)

Relação das funções.

[Funções obsoletas](#)

Relação das funções obsoletas.

[Ferramentas](#)

Relação de ferramentas úteis.

Nota do autor: Este é um trabalho em progresso, de forma que não está completo ainda. Mas já apresenta informações úteis, sendo por isto disponibilizado publicamente.

Exemplos

Esta seção apresenta uma série de exemplos simples, demonstrando como usar as classes e funções da biblioteca *hbmysql*. A primeira parte apresenta exemplos relacionados com as classes e a segunda parte, exemplos relacionados com as funções.

Usando as classes

Conectando com o servidor

```
oServer := TMySQLServer():New("localhost", "root", "")
```

Selecionando um banco de dados

```
oServer.SelectDB("banco")
```

Criando uma tabela

```
aTabela := {;  
    {"campo1","c",10,0},;  
    {"campo2","n",10,2};  
};  
  
IF oServer.CreateTable("tabela", aTabela)  
    Alert("tabela criada com sucesso")  
ELSE  
    Alert(oServer.Error())  
ENDIF
```

Apagando uma tabela

```
IF !oServer.DeleteTable("tabela")  
    Alert(oServer.Error())  
ENDIF
```

Executando uma consulta

```
oQuery := oServer.Query("SELECT * FROM tabela")
```

Encerrando a conexão com o servidor

```
oServer:Destroy()
```

Verificando a conexão

```
IF oServer:NetErr()  
    Alert(oServer:Error())  
ENDIF
```

Usando as funções

Conectando com o servidor

A conexão com o servidor se faz através da função [mysql_real_connect](#). A função retorna um ponteiro e seu sucesso pode ser testado com a função *empty*. Este ponteiro deverá ser usado nas demais funções envolvendo a manipulação de dados.

```
pCon := mysql_real_connect("localhost", "root", "password")  
  
IF Empty(pCon)  
    ? "a conexão com o servidor falhou"  
ENDIF
```

Criando um banco de dados

Um banco de dados é criado com o comando CREATE DATABASE, sendo usada a função [mysql_query](#) para executar a sentença SQL. Se o retorno for 0, a criação do banco foi bem sucedida.

```
nRetorno := mysql_query(pCon, "CREATE DATABASE nome_do_banco_de_dados")  
  
IF nRetorno == 0  
    ? "banco de dados criado com sucesso"  
ENDIF
```

Você pode usar IF NOT EXISTS para não ocorrer um erro ao tentar criar um banco que já existe.

```
CREATE DATABASE IF NOT EXISTS nome_do_banco_de_dados
```

Selecionado um banco de dados

Estando o banco de dados criado, ele deve ser selecionado para se poder trabalhar com ele. Esta seleção do banco pode ser feita de duas formas: através de uma sentença SQL ou através da função [mysql_select_db](#). O sucesso da operação pode ser confirmado através do valor de retorno.

```
nRetorno := mysql_query(pCon, "USE nome_do_banco_de_dados")  
  
IF nRetorno == 0  
    ? "banco de dados selecionado com sucesso"  
ENDIF
```

```
nRetorno := mysql_select_db(pCon, "nome_do_banco_de_dados")  
  
IF nRetorno == 0  
    ? "banco de dados selecionado com sucesso"  
ENDIF
```

Apagando um banco de dados

Um banco de dados pode ser apagado através do comando DROP DATABASE.

```
nRetorno := mysql_query(pCon, "DROP DATABASE nome_do_banco_de_dados")  
  
IF nRetorno == 0  
    ? "banco de dados apagado com sucesso"  
ENDIF
```

Atenção!

Use este comando com cautela para não haver perda de dados. Uma vez que o banco de dados seja eliminado, você não poderá desfazer a operação.

Listando os bancos de dados

...

Criando uma tabela

A criação de tabelas é feita através de sentenças SQL. Visto que estas sentenças podem ser bem

longas, pode-se criar uma variável e dividir a criação do comando em etapas. O exemplo abaixo demonstra como criar uma tabela, de nome *agenda*, com os campos *codigo*, *nome* e *telefone*. O campo *codigo* é a chave primária, iniciando com 1 e sendo incrementado automaticamente.

```
cSQL := "CREATE TABLE agenda ("
cSQL += "'codigo' int(10) unsigned NOT NULL AUTO_INCREMENT,"
cSQL += "'nome' varchar(45) NOT NULL,"
cSQL += "'telefone' varchar(12) DEFAULT NULL,"
cSQL += "PRIMARY KEY ('codigo') ) "
cSQL += "ENGINE=InnoDB "
cSQL += "AUTO_INCREMENT=1 "
cSQL += "DEFAULT CHARSET=latin1"

nRetorno := mysql_query(pCon, cSQL)

IF nRetorno == 0
    ? "tabela criada com sucesso"
ENDIF
```

Apagando uma tabela

...

Alterando uma tabela

...

Listando as tabelas

...

Inserindo dados na tabela

...

Alterando dados na tabela

...

Apagando dados da tabela

...

Consultando dados

...

Classes

Nesta seção você encontra uma relação das classes disponíveis na biblioteca. Para cada classe, são apresentadas as seguintes informações: hierarquia, propriedades, métodos, biblioteca e código.

TBColumnSQL

Hierarquia

TBColumn → TBColumnSQL

Propriedades

oBrw

nFieldNum

Métodos

New(cHeading, bBlock, oBrw)

Block()

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tsqlbrw.prg

TBrowseSQL

Hierarquia

TBrowse → TBrowseSQL

Propriedades

oCurRow

oQuery

Métodos

New(nTop, nLeft, nBottom, nRight, oServer, oQuery, cTable)

EditField()

BrowseTable(lCanEdit, aExitKeys)

KeyboardHook(nKey)

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tsqlbrw.prg

TMySQLQuery

Hierarquia

TMySQLQuery

Propriedades

nSocket

nResultHandle

cQuery

nNumRows

nCurRow

lBof

lEof

lFieldAsData

aRow

nNumFields

aFieldStruct

lError

Métodos

New(nSocket, cQuery)

Destroy()

End()

Refresh()

GetRow(nRow)

Skip(nRows)

Bof()

Eof()

RecNo()

LastRec()

GoTop()

GoBottom()

GoTO(nRow)

FCount()

NetErr()

Error()

FieldName(nNum)

FieldPos(cFieldName)

FieldGet(cnField)

FieldLen(nNum)

FieldDec(nNum, lFormat)

FieldType(nNum)

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tmysql.prg

TMySQLRow

Hierarquia

TMySQLRow

Propiedades

aRow

aDirty

aOldValue

aOriValue

aFieldStruct

cTable

Métodos

New(aRow, aFStruct, cTableName)

FieldGet(cnField)

FieldPut(cnField, Value)

FieldName(nNum)

FieldPos(cFieldName)

FieldLen(nNum)

FieldDec(nNum, lFormat)

FieldType(nNum)

MakePrimaryKeyWhere()

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tmysql.prg

TMySQLServer

Hierarquia

TMySQLServer

Propiedades

nSocket

cServer

cDBName

cUser

cPassword

lError

cCreateQuery

Métodos

New(cServer, cUser, cPassword, nPort)

Destroy()
SelectDB(cDBName)
CreateTable(cTable, aStruct, cPrimaryKey, cUniqueKey, cAuto)
DeleteTable(cTable)
TableStruct(cTable)
CreateIndex(cName, cTable, aFNames, lUnique)
DeleteIndex(cName, cTable)
ListDBs()
ListTables()
Query(cQuery)
NetErr()
Error()
CreateDatabase(cDataBase)
Sql_Commit()
Sql_Rollback()
Sql_Version()

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tmysql.prg

TMySQLTable

Hierarquia

TMySQLQuery → TMySQLTable

Propiedades

cTable
aOldValue

Métodos

New(nSocket, cQuery, cTableName)
GetRow(nRow)
Skip(nRow)
GoTop()
GoBottom()
GoTo(nRow)
Update(oRow, lOldRecord, lRefresh)
Save()
Delete(oRow, lOldRecord, lRefresh)
Append(oRow, lRefresh)
GetBlankRow(lSetValues)
SetBlankRow()
Blank()
FieldPut(cnField, Value)
Refresh()
MakePrimaryKeyWhere()

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\tmysql.prg

Funções

Nesta seção você encontra uma relação das funções disponíveis na biblioteca. Para cada função, são apresentadas as seguintes informações: descrição curta, sintaxe, parâmetros, retorno, biblioteca e código.

MYSQL_AFFECTED_ROWS

Obtém o número de linhas afetadas na operação anterior.

Sintaxe

`mysql_affected_rows(pCon) → nRows`

Parâmetros

pCon : ponteiro para uma conexão

Retorno

número de linhas afetadas

Biblioteca

hbmysql

Código

`\harbour\contrib\hbmysql\mysql.c`

MYSQL_CLOSE

Fecha a conexão.

Sintaxe

`mysql_close(pCon) → NIL`

Parâmetros

pCon : ponteiro para uma conexão

Retorno

NIL

Biblioteca

hbmysql

Código

`\harbour\contrib\hbmysql\mysql.c`

MYSQL_COMMIT

Realiza um *commit* na transação corrente.

Sintaxe

`mysql_commit(pCon) → nResult`

Parâmetros

pCon : ponteiro para uma conexão

Retorno

Retorna 0 em caso de sucesso e diferente de 0 em caso de erro.

Biblioteca

hbmysql

Código

`\harbour\contrib\hbmysql\mysql.c`

MYSQL_DATA_SEEK

Posiciona na linha determinada no resultado de uma consulta.

Sintaxe

`mysql_data_seek(pResult, nRow)` → NIL

Parâmetros

`pResult` : ponteiro para o resultado de uma consulta

`nRow` : número da linha

Retorno

NIL

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_ERROR

Retorna a descrição do último erro ocorrido.

Sintaxe

`mysql_error(pCon)` → `cError`

Parâmetros

`pCon` : ponteiro para uma conexão

Retorno

descrição do erro

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_ESCAPE_STRING

Prepara uma *string* para ser usada numa consulta.

Sintaxe

`mysql_escape_string(cString)` → `cNewString`

Parâmetros

`cString` : string para ser processada

Retorno

string processada

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_ESCAPE_STRING_FROM_FILE

Prepara uma *string*, obtida de um arquivo, para ser usada numa consulta.

Sintaxe

`mysql_escape_string_from_file(cFileName)` → `cString`

Parâmetros

cFileName : nome do arquivo contendo string para ser processada

Retorno

string processada

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_FETCH_FIELD

Retorna informações sobre as colunas do resultado da consulta.

Sintaxe

mysql_fetch_field(pResult) → aField

Parâmetros

pResult : ponteiro para o resultado de uma consulta

Retorno

aField[1] = name // char

aField[2] = table // char

aField[3] = def // char

aField[4] = type // numeric

aField[5] = length // numeric

aField[6] = max_length // numeric

aField[7] = flags // numeric

aField[8] = decimals // numeric

Na primeira chamada, retorna informações sobre a primeira coluna. Nas chamadas subsequentes, retorna informações sobre as próximas colunas.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_FETCH_ROW

Retorna uma linha do resultado de uma consulta.

Sintaxe

mysql_fetch_row(pResult) → aRow

Parâmetros

pResult : ponteiro para o resultado de uma consulta

Retorno

array com os dados da linha

Deve ser chamada sucessivamente para obter as linhas da consulta.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_FIELD_COUNT

Retorna o número de colunas da consulta mais recente.

Sintaxe

mysql_field_count(pCon) → nFields

Parâmetros

pCon : ponteiro para uma conexão

Retorno

número de colunas (campos)

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_FIELD_SEEK

Posiciona o ponteiro na coluna especificada.

Sintaxe

mysql_field_seek(pResult, nIndex) → NIL

Parâmetros

pResult : ponteiro para o resultado de uma consulta

nIndex : índice da coluna

Retorno

NIL

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_FREE_RESULT

Libera a memória ocupada pelo resultado de uma consulta.

Sintaxe

mysql_free_result(pResult) → NIL

Parâmetros

pResult : ponteiro para o resultado de uma consulta

Retorno

NIL

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_GET_HOST_INFO

Obtém uma *string* descrevendo o tipo de conexão e o nome do servidor.

Sintaxe

mysql_get_host_info(pCon) → cHostInfo

Parâmetros

pCon : ponteiro para uma conexão

Retorno

descrição do tipo de conexão e o nome do servidor

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_GET_SERVER_INFO

Obtém uma *string* com a versão do servidor MySQL.

Sintaxe

mysql_get_server_info(pCon) → cServerInfo

Parâmetros

pCon : ponteiro para uma conexão

Retorno

versão do servidor MySQL

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_GET_SERVER_VERSION

Obtém a versão do servidor MySQL em forma de número.

Sintaxe

mysql_get_server_version(pCon) → nServerVersion

Parâmetros

pCon : ponteiro para uma conexão

Retorno

versão do servidor MySQL

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_LIST_DB

Obtém uma lista dos bancos de dados presentes no servidor MySQL.

Sintaxe

mysql_list_dbs(pCon) → aDBS

Parâmetros

pCon : ponteiro para uma conexão

Retorno

array com a lista de bancos de dados

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_LIST_FIELDS

Obtém um ponteiro para o resultado de uma consulta, contendo uma lista de campos.

Sintaxe

mysql_list_fields(pCon, cTableName) → pResult

Parâmetros

pCon : ponteiro para uma conexão

Retorno

ponteiro para o resultado de uma consulta

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_LIST_TABLES

Obtém uma lista de tabelas, conforme a expressão regular informada.

Sintaxe

mysql_list_tables(pCon, cWild) → aTables

Parâmetros

pCon : ponteiro para uma conexão

cWild : expressão regular

Retorno

array com uma lista de tabelas

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_NUM_FIELDS

Obtém o número de colunas no resultado de uma consulta.

Sintaxe

mysql_num_fields(pResult) → nNumFields

Parâmetros

pResult : ponteiro para o resultado de uma consulta

Retorno

número de colunas (campos)

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_NUM_ROWS

Obtém o número de linhas no resultado de uma consulta.

Sintaxe

mysql_num_rows(pResult) → nNumRows

Parâmetros

pResult : ponteiro para o resultado de uma consulta

Retorno

número de linhas

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_QUERY

Realiza uma consulta no banco de dados selecionado.

Sintaxe

mysql_query(pCon, cQuery) → nResult

Parâmetros

pCon : ponteiro para uma conexão

cQuery : comandos SQL para serem executados

Retorno

Retorna 0 em caso de sucesso e <> 0 em caso de erro.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_REAL_CONNECT

Realiza uma conexão com um servidor MySQL.

Sintaxe

mysql_real_connect(cHost, cUser, cPassword, nPort, nFlags) → pCon

Parâmetros

cHost : nome do servidor

cUser : nome do usuário

cPassword : senha do usuário

nPort : número da porta de comunicação

nFlags : flags

O valor *default* de nPort é MYSQL_PORT.

O valor *default* de nFlags é 0.

Retorno

um ponteiro válido se a conexão for bem sucedida.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_ROLLBACK

Desfaz a transação corrente.

Sintaxe

mysql_rollback(pCon) → nResult

Parâmetros

pCon : ponteiro para uma conexão

Retorno

Retorna 0 em caso de sucesso e <> 0 em caso de erro.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_SELECT_DB

Seleciona um banco de dados.

Sintaxe

mysql_select_db(pCon, cDatabase) → nResult

Parâmetros

pCon : ponteiro para uma conexão

Retorno

Retorna 0 em caso de sucesso e <> 0 em caso de erro.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

MYSQL_STORE_RESULT

Armazena o resultado de uma consulta.

Nota para o autor: confirmar a correta descrição para esta função.

Sintaxe

mysql_store_result(pCon) → pResult

Parâmetros

pCon : ponteiro para uma conexão

Retorno

ponteiro para o resultado de uma consulta

Biblioteca

hbmysql

Código

harbour\contrib\hbmysql\mysql.c

MYSQL_USE_RESULT

Usa o resultado de uma consulta.

Nota para o autor: confirmar a correta descrição para esta função.

Sintaxe

mysql_use_result(pCon) → pResult

Parâmetros

pCon : ponteiro para uma conexão

Retorno

ponteiro para o resultado de uma consulta

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysql.c

Funções obsoletas

Estes são os nomes originais das funções da biblioteca *hbmysql*. Estes nomes foram modificados para ficarem em sincronia com os nomes das funções da API do MySQL Server. Na sequência, você encontrará o nome original e o novo atual, facilitando os ajustes em seus programas que ainda utilizam os nomes obsoletos.

Com exceção da função [SQLAND](#), cada uma das funções abaixo chama a função equivalente com o nome correto. Estas funções obsoletas poderão deixar de existir em futuras versões do Harbour, de forma que a melhor solução é sua substituição pela função correta.

DATATOSQL

Veja a função [MYSQL_ESCAPE_STRING](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

FILETOSQLBINARY

Veja a função [MYSQL_ESCAPE_STRING_FROM_FILE](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLAFFROWS

Veja a função [MYSQL_AFFECTED_ROWS](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLAND

Veja a função `HB_BITAND`, no guia de funções do Harbour.

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLCLOSE

Veja a função [MYSQL_CLOSE](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLCOMMIT

Veja a função [MYSQL_COMMIT](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLCONNECT

Veja a função [MYSQL_REAL_CONNECT](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLDATAS

Veja a função [MYSQL_DATA_SEEK](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLFETCHF

Veja a função [MYSQL_FETCH_FIELD](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLFETCHR

Veja a função [MYSQL_FETCH_ROW](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLFICOU

Veja a função [MYSQL_FIELD_COUNT](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLFREER

Veja a função [MYSQL_FREE_RESULT](#).**Biblioteca**

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLFSEEK

Veja a função [MYSQL_FIELD_SEEK](#).**Biblioteca**

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLGETERR

Veja a função [MYSQL_ERROR](#).**Biblioteca**

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLHOSTINFO

Veja a função [MYSQL_GET_HOST_INFO](#).**Biblioteca**

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLLISTDB

Veja a função [MYSQL_LIST_DB](#).**Biblioteca**

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLLISTF

Veja a função [MYSQL_LIST_FIELDS](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLLISTTBL

Veja a função [MYSQL_LIST_TABLES](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLNROWS

Veja a função [MYSQL_NUM_ROWS](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLNUMFI

Veja a função [MYSQL_NUM_FIELDS](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLQUERY

Veja a função [MYSQL_QUERY](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLROLLBACK

Veja a função [MYSQL_ROLLBACK](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLSELECTD

Veja a função [MYSQL_SELECT_DB](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLSRVINFO

Veja a função [MYSQL_GET_SERVER_INFO](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLSTORER

Veja a função [MYSQL_STORE_RESULT](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLUSERES

Veja a função [MYSQL_USE_RESULT](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

SQLVERSION

Veja a função [MYSQL_GET_SERVER_VERSION](#).

Biblioteca

hbmysql

Código

\harbour\contrib\hbmysql\mysqlold.c

Ferramentas

Esta seção traz uma relação de ferramentas úteis no desenvolvimento.

Nota do autor: apenas uma ferramenta está listada, por enquanto. Outras ferramentas úteis serão adicionadas em futuras revisões deste guia.

DBF2MYSQL

DBF2MYSQL é uma ferramenta que importa, para uma tabela do MySQL Server, os dados de um arquivo .DBF.

Parâmetros:

-h <hostname>

Nome do servidor. Usa “localhost” caso não seja informado

-u <user>

Nome do usuário. Usa “root” caso não seja informado.

-p <password>

Senha. Usa “” (sem senha) caso não seja informado.

-d <database>

Nome do banco de dados.

-t <table>

Nome da tabela onde os dados serão importados.

-c

Indica que a tabela deverá ser apagada e criada novamente, antes dos dados serem importados.

-f <dbfname>

Nome do arquivo .dbf que será importado.

Todos os parâmetros, com exceção de -h, -u, -p e -c, são obrigatórios. Ou seja, os nomes do banco de dados, da tabela e do arquivo .dbf sempre deverão ser informados.

Exemplos de uso:

```
dbf2mysql -d dbteste -t tblagenda -f agenda.dbf
```

No servidor local, no banco de dados *dbteste*, importa os dados do arquivo *agenda.dbf* na tabela *tblagenda*, sem recriar a tabela.

```
dbf2mysql -d dbteste -t tblagenda -c -f agenda.dbf
```

No servidor local, no banco de dados *dbteste*, importa os dados do arquivo *agenda.dbf* na tabela

tblagenda, recriando a tabela.

Código: \harbour\contrib\hbmysql\utils\dbf2mysql.prg