

MySQL

Rubens Prates

GUIA DE CONSULTA RÁPIDA

Rubens Prates

MySQL

www.novateceditora.com.br

novatec

novatec

Guia de Consulta Rápida

MySQL

Rubens Prates

Novatec Editora

www.novateceditora.com.br

Guia de Consulta Rápida **MySQL** de Rubens Prates

Copyright©2000 da Novatec Editora Ltda.

Todos os direitos reservados. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Revisão de texto: Sonia Maike

ISBN: 85-85184-78-7

Novatec Editora Ltda.

Rua Cons. Moreira de Barros 1084 Conj. 01

02018-012 São Paulo - SP Brasil

Fone: (0xx11) 6959-6529

Fax: (0xx11) 6950-8869

E-mail: novatec@novateceditora.com.br

Site: www.novateceditora.com.br

Comandos SQL	5
Operadores	26
Operadores Aritméticos	26
Operadores Bitwise	26
Operadores Lógicos	26
Operadores de Comparação	26
Operador Cast (BINARY)	27
Operador LIKE	27
Operador REGEXP	28
Funções	29
Funções de Agregação	29
Funções de Comparação	30
Função CASE	31
Funções Numéricas	32
Funções de String	34
Funções de Data e Hora	39
Miscelânea	44
Comentários no código SQL	46
Principais Programas do MySQL	47
Opções de Programas	48
Opções padrão de linha de comando	48
Arquivo de Opções	50
Variáveis de Ambiente	51
Servidor MySQL	52
Programas Clientes	56
isamchk e myisamchk	56
myisampack e pack_isam	58
mysql	60
mysqlaccess	62
mysqladmin	63
mysqlbug	65
mysqldump	66
mysqlimport	69
mysqlshow	71
Tipos de Dados	72
Tipos de Colunas	72
Tipos de Dados do MySQL	72
Literais	77
Valores NULL	78
Controle de Acesso a BD	79

Sumário

Nomes de Identificadores	82
Nomes de Identificadores	82
Uso de maiúsculas e minúsculas	82
Palavras Reservadas do MySQL	83
Funções PHP	84
Informações Gerais sobre o MySQL	88
Política de Licenças do MySQL	88
Número de versão do MySQL	88
Política de atualização de versões do MySQL	88
Diretórios default do MySQL	89
Diferenças nas estruturas dos diretórios	89
Informações Adicionais	90
Versão do MySQL utilizada no Guia	90
Bibliografia sobre MySQL	90
Notação utilizada neste Guia	90
Comentários e Sugestões	90
Índice	91

Comandos SQL

ALTER TABLE

Altera a estrutura de uma tabela.

ALTER [IGNORE] TABLE *tabela* ação [...]

Parâmetro	Significado
IGNORE	Utiliza somente a primeira linha quando encontrar linhas com duplicidade em uma chave UNIQUE, as outras linhas serão eliminadas. Se a opção não for especificada, será emitida uma mensagem de erro e nenhuma alteração será feita na tabela.
<i>tabela</i>	Nome da tabela a ser alterada.
<i>ação</i>	Ação ou ações a serem efetuadas na tabela.

Ações do comando ALTER TABLE

ADD COLUMN

Adiciona uma coluna à tabela.

**ALTER [IGNORE] TABLE *tabela*
ADD [COLUMN] *decl_coluna* [*local_coluna*]**

Parâmetro	Significado
<i>decl_coluna</i>	Declaração da coluna; mesmo formato usado no comando CREATE TABLE.
<i>local_coluna</i> ^[3.22]	Localização da coluna na tabela. Se não especificada, a coluna será a última da tabela.
FIRST	Adiciona a nova coluna na primeira posição da tabela.
AFTER	Adiciona a nova coluna após a coluna especificada.

ADD INDEX

Adiciona um índice à tabela.

**ALTER [IGNORE] TABLE *tabela*
ADD INDEX [*índice*] (*coluna*,...)**

ADD PRIMARY KEY

Adiciona uma chave primária à tabela.

**ALTER [IGNORE] TABLE *tabela*
ADD PRIMARY KEY (*coluna*,...)**

ADD UNIQUE

Adiciona um índice sem duplicidade à tabela.

**ALTER [IGNORE] TABLE *tabela*
ADD UNIQUE [*índice*] (*coluna*,...)**

ALTER COLUMN

Altera (SET) ou elimina (DROP) o valor default de uma coluna.

```
ALTER [ IGNORE ] TABLE tabela
    ALTER [ COLUMN ] coluna
    { SET | DROP } DEFAULT valor
```

CHANGE COLUMN

Altera o nome e/ou a declaração de uma coluna.

```
ALTER [ IGNORE ] TABLE tabela
    CHANGE [ COLUMN ] coluna decl_coluna
```

Parâmetro	Significado
<i>coluna</i>	Nome atual da coluna.
<i>decl_coluna</i>	Declaração da coluna; mesmo formato usado no comando CREATE TABLE.

DROP COLUMN

Remove uma coluna da tabela.

```
ALTER [ IGNORE ] TABLE tabela
    DROP [ COLUMN ] coluna
```

DROP INDEX

Remove um índice da tabela.

```
ALTER [ IGNORE ] TABLE tabela
    DROP INDEX índice
```

DROP PRIMARY KEY

Remove a chave primária da tabela.

```
ALTER [IGNORE] TABLE tabela
    DROP PRIMARY KEY
```

MODIFY COLUMN ^[3.22.16]

Altera a declaração de uma coluna. Similar a **CHANGE COLUMN**, exceto que não permite alterar o nome da coluna.

```
ALTER [IGNORE] TABLE tabela
    MODIFY [ COLUMN ] decl_coluna
```

Parâmetro	Significado
<i>decl_coluna</i>	Declaração da coluna; mesmo formato usado no comando CREATE TABLE.

RENAME

Altera o nome de uma tabela.

```
ALTER [ IGNORE ] TABLE tabela
    RENAME [ AS ] novo_nome
```

CHECK TABLE

Verifica a existência de erros na estrutura das tabelas. Somente com tabelas MyISAM. Similar a `myisamchk -m tabela`. Retorna uma tabela com as seguintes colunas:

Coluna	Significado
Table	Nome da tabela.
Op	Sempre "check".
Msg_type	Um dos status: error, info ou warning.
Msg_text	Mensagem de erro.

CHECK TABLE *tabela* [, *tabela...*]

CREATE DATABASE

Cria um banco de dados. No MySQL os bancos de dados são implementados como diretórios contendo arquivos que correspondem a tabelas do banco de dados. Como ainda não existem tabelas no banco de dados criado, este comando somente cria um subdiretório no diretório de dados do MySQL. Ocorrerá um erro se o banco de dados já existir ou se você não tiver o privilégio apropriado.

CREATE DATABASE *nomebd*

CREATE FUNCTION

Adiciona ao MySQL (na tabela *func* do banco de dados *mysql*) uma função definida pelo usuário, que estende o MySQL e funciona da mesma forma que uma função nativa (built-in) do MySQL, tais como ABS() e CONCAT(). Para mais detalhes consulte a ajuda on-line do MySQL.

CREATE [AGGREGATE] FUNCTION *nome*
RETURNS { STRING | REAL | INTEGER }
SONAME *nome_library*

Parâmetro	Descrição
AGGREGATE ^[3.23.5]	Indica que trata-se de uma função AGGREGATE. Essas funções funcionam exatamente como uma função de agregação nativa do MySQL, tais como SUM() ou COUNT().
FUNCTION	Especifica o nome pela qual será feita referência à função.
RETURNS	Indica o tipo de dado retornado pela função.
SONAME	Path da biblioteca que contém o código executável ou a função.

CREATE INDEX ^[3.22]

Adiciona um índice a uma tabela.

CREATE [UNIQUE] INDEX *índice* ON *tabela*
(*coluna*,...)

Parâmetro	Significado
<i>índice</i>	Nome do índice a ser criado.
UNIQUE	Não permite duplicidade.
ON	Especifica o nome da tabela.
<i>coluna</i>	Nome da coluna ou colunas usadas para formar o índice. No caso de várias colunas, os valores do índice serão formados pela concatenação das colunas.

CREATE TABLE

Cria uma tabela no banco de dados corrente.

CREATE [TEMPORARY] TABLE
[IF NOT EXISTS] *tabela* [(*definição*,...)]
[*opções*]
[{ IGNORE | REPLACE } *comando_select*]

definição:

```
{  
  declaração_coluna  
  | PRIMARY KEY (coluna_índice,...)  
  | KEY [índice] (coluna_índice,...)  
  | INDEX [índice] (coluna_índice,...)  
  | UNIQUE [ INDEX ] [índice] (coluna_índice,...)  
}
```

declaração_coluna:

```
coluna tipo  
[ NOT NULL | NULL ]  
[ DEFAULT valor ]  
[ AUTO_INCREMENT [ = valorinicial ] ]  
[ PRIMARY KEY ]
```

Parâmetro	Significado
TEMPORARY ^[3.23]	Cria uma tabela temporária, visível somente para a conexão corrente, a qual será removida quando a conexão for finalizada.
IF NOT EXISTS ^[3.23]	Cria a tabela somente se não existir uma tabela com o mesmo nome.
<i>tabela</i>	Nome da tabela a ser criada. A partir da versão 3.22 o nome da tabela pode ser especificado como: <i>nomebd.tabela</i> .
<i>definição</i>	Lista das colunas e índices a serem criados.

<i>declaração_coluna</i>	Declaração da coluna. É opcional se a tabela for criada com o resultado de um comando SELECT.
<i>coluna</i>	Nome da coluna.
<i>tipo</i>	Tipo de dados da coluna. Ver Tipos de Dados na página 72.
NOT NULL	Não permite valores NULL na coluna.
NULL	Permite valores NULL na coluna (default).
DEFAULT	Atribui o valor default à coluna, quando uma nova linha for criada sem especificar o valor da coluna.
AUTO_INCREMENT	Tipo especial de coluna numérica que pode ser atualizada automaticamente. Quando um valor NULL (ou 0) for inserido na coluna, o MySQL o substituirá automaticamente pelo próximo número na seqüência da coluna. O primeiro valor da coluna pode ser atribuído na opção AUTO_INCREMENT (página 10), sendo que o default é 1. É permitida apenas uma coluna AUTO_INCREMENT por tabela e a coluna deve ser UNIQUE ou PRIMARY KEY. Veja também o comando SET INSERT_ID na página 22. Para mais detalhes consulte o manual on-line do MySQL.
PRIMARY KEY	Indica que a coluna é a PRIMARY KEY (chave primária da tabela).
PRIMARY KEY <i>coluna_índice</i>	Identifica a PRIMARY KEY da tabela. Coluna a ser utilizada no índice. Para colunas CHAR e VARCHAR pode ser especificado o número de caracteres da coluna, usando a sintaxe: <i>coluna(tamanho)</i> . Para colunas BLOB e TEXT o tamanho deve ser sempre especificado.
KEY	Sinônimo de INDEX .
INDEX	Cria um índice. Podem existir até 16 índices por tabela.
UNIQUE	Cria um índice sem duplicidade. Ocorrerá um erro ao se adicionar uma nova linha que coincida com uma linha existente.
IGNORE REPLACE	Define se os valores duplicados em um índice UNIQUE devem ser ignorados (IGNORE) ou substituídos (REPLACE). Se esta opção não for especificada, ocorrerá um erro e os registros remanescentes serão ignorados.
<i>comando_select</i> ^[3,23]	Cria a tabela a partir dos registros retornados pelo comando SELECT.

opções ^[3.23]

AUTO_INCREMENT = <i>valor</i>	Especifica o valor inicial de AUTO_INCREMENT para a tabela (somente para tabelas MyISAM).
AVG_ROW_LENGTH = <i>n</i>	Tamanho médio aproximado das linhas na tabela. Usado para determinar o tamanho máximo de arquivo para tabelas com colunas BLOB ou TEXT que podem ultrapassar o limite de 4GB.
CHECKSUM = {0 1}	Mantém (1) ou não (0) um checksum para cada linha da tabela. Torna a atualização da tabela um pouco mais lenta, mas facilita a procura por erros na tabela (somente para MyISAM).
COMMENT = " <i>string</i> "	Comentário da tabela (até 60 caracteres).
DELAY_KEY_WRITE = {0 1}	Atualiza o cache do índice da tabela ocasionalmente (1) ou a cada inserção (0) (somente para tabelas MyISAM).
MAX_ROWS = <i>n</i>	Número máximo de linhas previsto para a tabela.
MIN_ROWS = <i>n</i>	Número mínimo de linhas previsto para a tabela.
PACK_KEYS = {0 1}	Se definida como 1, compacta os blocos de índices numa porcentagem maior que a usual, tornando as operações de atualização mais demoradas e as de leitura mais rápidas. Se definida como 0, somente os índices com chaves CHAR e VARCHAR, com 8 ou mais caracteres, são compactados (somente para tabelas MyISAM e ISAM).
PASSWORD = " <i>senha</i> "	Especifica uma senha para a encriptação do arquivo de descrição da tabela (.frm). Somente para versões especiais do MySQL; não tem efeito na versão padrão.
ROW_FORMAT = { default dynamic static compressed }	Define como as linhas deverão ser armazenadas (no futuro).
TYPE = <i>formato</i>	Formato de armazenamento da tabela.
ISAM	Formato original usado pelo MySQL.
MYISAM	Formato melhor que ISAM e deve substituí-lo no futuro (default).
HEAP	Formato para tabelas temporárias armazenadas na memória.

DELETE

Elimina linhas de uma tabela.

DELETE [**LOW_PRIORITY**] **FROM** *tabela*
[**WHERE** *condição*] [**LIMIT** *n*]

Parâmetro	Descrição
LOW_PRIORITY ^[3.22.5]	Retarda a execução do comando até que nenhum outro programa cliente esteja lendo a tabela.
WHERE	Especifica a condição que determina quais linhas devem ser eliminadas. Se a cláusula WHERE não for especificada, todas as linhas da tabela serão eliminadas.
LIMIT ^[3.22.7]	Limita o número de linhas a serem eliminadas.

DESCRIBE

Fornece informações sobre colunas de uma tabela. Similar ao comando **SHOW COLUMNS FROM** *tabela*.

DESC[RIBE] *tabela coluna*

Parâmetro	Descrição
<i>tabela</i>	Nome da tabela.
<i>coluna</i>	Nome da coluna. Podem ser especificados os caracteres curinga '%' e '_'.

DROP DATABASE

Elimina um banco de dados. Um banco de dados é representado por um diretório. Se existirem arquivos nesse diretório não relacionados com o banco de dados, eles não serão eliminados, nem o diretório será eliminado.

DROP DATABASE [**IF EXISTS**] *nomebd*

Parâmetro	Descrição
IF EXISTS ^[3.22.2]	Evita a ocorrência de erro se o banco de dados não existir ou se você não tiver os privilégios necessários.

DROP FUNCTION

Remove uma função previamente adicionada com o comando **CREATE FUNCTION**.

DROP FUNCTION *nome*

DROP INDEX ^[3.22]

Elimina um índice de uma tabela. Equivale ao comando **ALTER TABLE DROP INDEX**.

DROP INDEX *índice* **ON** *tabela*

DROP TABLE

Elimina uma ou mais tabelas. Todos os dados e definições das tabelas serão eliminados.

DROP TABLE [IF EXISTS] *tabela* [, ...]

Parâmetro	Descrição
IF EXISTS ^[3.22.2]	Evita a ocorrência de erro devido a tabelas inexistentes.

EXPLAIN

EXPLAIN *tabela*

Lista as colunas de uma tabela. Similar ao comando SHOW COLUMNS FROM *tabela*.

EXPLAIN SELECT *comando*SELECT

Exibe informações detalhadas sobre como o MySQL executará um comando SELECT. Permite visualizar onde as chaves não estão sendo usadas de maneira eficiente.

FLUSH ^[3.22.9]

Limpa ou reinicia processos internos utilizados pelo MySQL.

FLUSH *opção* [,...]

Opção	Descrição
HOSTS	Limpa o cache que armazena informações de hostnames para clientes.
LOGS	Fecha e reabre os arquivos padrão de log.
PRIVILEGES	Recarrega os privilégios das tabelas de permissões do MySQL.
STATUS ^[3.22.11]	Reinicializa as variáveis de status que indicam o estado corrente do servidor.
TABLES	Fecha todas as tabelas abertas e grava em disco os dados do cache.

GRANT [3.22.11]

Atribui privilégios de acesso a usuários do MySQL.

```
GRANT privilégio [(colunas)]
    [, privilégio [(colunas)] ...]
ON { *.* | * | bd.* | bd.tabela | tabela }
TO usuário@host [ IDENTIFIED BY 'senha' ] [, ...]
[ WITH GRANT OPTION ]
```

Parâmetro	Descrição
<i>privilégio</i>	Privilégio a ser atribuído.
ALL [PRIVILEGES]	Todos os privilégios.
ALTER	Alterar tabelas e índices.
CREATE	Criar bancos de dados e tabelas.
DELETE	Eliminar linhas de tabelas.
DROP	Eliminar bancos de dados e tabelas.
FILE	Ler e gravar arquivos no servidor.
INDEX	Criar e eliminar índices.
INSERT	Inserir linhas em tabelas.
PROCESS	Visualizar e eliminar quaisquer threads.
RELOAD	Recarregar as tabelas internas do MySQL (Ver comando FLUSH).
SELECT	Ler dados de tabelas.
SHUTDOWN	Desativar o servidor.
UPDATE	Alterar linhas em tabelas.
USAGE	Sem privilégios.
<i>colunas</i>	Nomes das colunas às quais os privilégios devem ser aplicados.
ON	Especifica o nível de privilégio.
<i>*.*</i>	Atribui os privilégios a todas as tabelas em todos os bancos de dados do servidor.
<i>*</i>	Atribui os privilégios a todas as tabelas do banco de dados corrente.
<i>bd.*</i>	Atribui os privilégios a todas as tabelas do banco de dados especificado.
<i>bd.tabela</i>	Atribui os privilégios a todas as colunas da tabela especificada.
<i>tabela</i>	Atribui os privilégios a todas as colunas da tabela especificada no banco de dados default.
TO	Especifica o(s) nome(s) do(s) usuário(s) a ter(em) privilégios atribuídos. Formato: <i>usuário@host</i> .
<i>usuário</i>	Nome do usuário, ou um string vazio para especificar um usuário anônimo.
<i>host</i>	Nome do host. Pode ser: localhost, um nome de host, ou um endereço IP. Os caracteres '%' e '_' podem ser utilizados.
IDENTIFIED BY	Atribui uma senha ao usuário, ou altera a senha de um usuário existente.
WITH GRANT OPTION	Permite ao usuário atribuir a outros usuários qualquer um dos seus privilégios no nível de privilégio especificado.

INSERT

Inserir linhas em uma tabela.

```
INSERT [ LOW_PRIORITY | DELAYED ] [ IGNORE ]  
  [ INTO ] tabela [(coluna,...)]  
  VALUES (expr [,...]) [, (...)] ...
```

```
INSERT [ LOW_PRIORITY | DELAYED ] [ IGNORE ]  
  [ INTO ] tabela [(coluna,...)]  
  SELECT ...
```

```
INSERT [ LOW_PRIORITY | DELAYED] [ IGNORE ]  
  [ INTO ] tabela  
  SET coluna=expr [,coluna=expr] ...
```

Parâmetro	Descrição
LOW_PRIORITY	Retarda a execução do comando até que nenhum outro programa cliente esteja lendo a tabela.
DELAYED ^[3.22.15]	Coloca as linhas em uma fila para inserção posterior, quando nenhum outro cliente estiver acessando a tabela.
IGNORE ^[3.22.10]	Descarta linhas com valores duplicados para chaves UNIQUE. Se esta opção não for especificada, ocorrerá um erro se forem encontrados valores duplicados e nenhuma outra linha será inserida.
[INTO] tabela	Nome da tabela. INTO é opcional.
VALUES	Inserir linhas baseado nos valores especificados. A partir da versão 3.22.5 múltiplas listas de valores podem ser especificadas, permitindo a inserção de múltiplas linhas em um único INSERT.
SELECT	Inserir as linhas retornadas pelo comando SELECT.
SET ^[3.22.10]	Atribui os valores das expressões às colunas. Colunas não especificadas terão seus valores default.

KILL ^[3.22.9]

Elimina um thread. Cada conexão com o mysqld é executada em um thread separado. Utilize SHOW PROCESSLIST para ver quais threads estão sendo executados.

```
KILL id_thread
```

LOAD DATA

Inserir linhas de um arquivo em uma tabela.

```
LOAD DATA [ LOW_PRIORITY ] [ LOCAL ]  
  INFILE 'nomearq.txt' [ IGNORE | REPLACE ]  
  INTO TABLE tabela  
  delimitadores  
  [ IGNORE n LINES ] (coluna,...)]
```

delimitadores:

```
[ FIELDS
  [ TERMINATED BY 'string' ]
  [ OPTIONALLY ] ENCLOSED BY 'char' ]
  [ ESCAPED BY 'char' ] ]
[ LINES TERMINATED BY 'string' ]
```

Parâmetro	Descrição
LOW_PRIORITY ^[3.23]	Retarda a execução do comando até que nenhum outro programa cliente esteja lendo a tabela.
LOCAL ^[3.22.15]	Indica que o arquivo será lido do host cliente e enviado pela rede para o servidor. Se não especificado, o arquivo será lido do servidor.
INFILE	Especifica o nome do arquivo texto.
IGNORE REPLACE	Indicam qual procedimento deve ser adotado no caso dos dados da chave primária a serem importados já existirem no banco de dados. A opção REPLACE faz com que o dados antigos sejam removidos e os novos inseridos. A opção IGNORE faz com que o mysql ignore os dados. Caso nenhuma das opções tenha sido escolhida, exibe um erro e ignora o resto do arquivo.
INTO TABLE <i>delimitadores</i> FIELDS	Nome da tabela que receberá os dados. Faz com que as opções TERMINATED BY, [OPTIONALLY] ENCLOSED BY e ESCAPED BY sejam facultativas, mas pelo menos uma deve ser declarada.
FIELDS TERMINATED BY	Especifica o caractere ou caracteres utilizados para delimitar valores na linha (default=TERMINATED BY '\t').
OPTIONALLY	Somente colunas dos tipos CHAR e VARCHAR recebem o caractere especificado por ENCLOSED BY.
FIELDS ENCLOSED BY	Especifica o caractere utilizado para delimitar a string de caracteres (default=ENCLOSED BY '"').
FIELDS ESCAPED BY	Especifica o caractere utilizado para indicar que o próximo caractere não é um caractere especial (default=ESCAPED BY '\\').
LINES TERMINATED BY	Especifica o caractere ou caracteres que indicam o final de uma linha (default='\n').
IGNORE <i>n</i> LINES	Ignora as primeiras <i>n</i> linhas do arquivo.

LOCK TABLES

Bloqueia uma ou mais tabelas. Espera até conseguir bloquear todas as tabelas especificadas.

```
LOCK TABLES tabela [ AS alias ]
    { READ | [ LOW_PRIORITY ] WRITE }
    [, tabela { READ | [ LOW_PRIORITY ] WRITE } ...]
```

Parâmetro	Descrição
AS	Alias que deverá ser utilizado nas consultas subseqüentes que dependam das tabelas bloqueadas.
READ	Permite aos outros clientes somente a leitura da tabela.
LOW_PRIORITY ^[3.22.5]	Bloqueios WRITE normalmente possuem prioridade maior que bloqueios READ, para garantir que as atualizações sejam processadas o mais rapidamente possível. LOW_PRIORITY funciona no sentido de outros threads poderem receber READ locks enquanto o thread espera que a sua requisição de WRITE lock seja aceita, a qual será aceita quando não existir nenhum READ LOCK na tabela.
WRITE	Não permite aos outros clientes ler ou gravar na tabela.

OPTIMIZE TABLE ^[3.22.7]

Otimiza o espaço ocupado por uma tabela. Uma tabela poderá ficar fragmentada quando ocorrerem muitas eliminações, principalmente com registros de tamanhos diferentes. Recomenda-se, nesse caso, executar periodicamente o comando OPTIMIZE TABLE, para manter o desempenho. Durante o processo de otimização o acesso à tabela é feito normalmente, exceto que as gravações na tabela são direcionadas para uma tabela temporária.

```
OPTIMIZE TABLE tabela
```

REPAIR TABLE

Repara tabelas danificadas. Somente com tabelas MyISAM. Similar a myisamchk -r *tabela*. Similar a myisamchk -m *tabela*. Retorna uma tabela com as seguintes colunas:

Coluna	Significado
Table	Nome da tabela.
Op	Sempre "repair".
Msg_type	Um dos status: error, info ou warning.
Msg_text	Mensagem de erro.

```
REPAIR TABLE tabela [,tabela...]
```

REPLACE

Similar ao comando **INSERT**, exceto que se a linha a ser inserida tiver um valor para um índice **UNIQUE** igual ao valor em uma linha já existente na tabela, esta será eliminada antes da nova linha ser inserida.

```
REPLACE [ LOW_PRIORITY | DELAYED ]
        [ INTO ] tabela [(coluna,...)]
        VALUES (expressão,...)
```

```
REPLACE [ LOW_PRIORITY | DELAYED ]
        [ INTO ] tabela [(coluna,...)] SELECT ...
```

```
REPLACE [ LOW_PRIORITY | DELAYED ]
        [ INTO ] tabela
        SET coluna=expressão, coluna=expressão,...
```

Parâmetro	Descrição
LOW_PRIORITY	Retarda a execução do comando até que nenhum outro programa cliente esteja lendo a tabela.
DELAYED [3.22.15]	Coloca as linhas em uma fila para inserção posterior, quando nenhum outro programa cliente estiver acessando a tabela.
[INTO] <i>tabela</i>	Nome da tabela. INTO é opcional.
VALUES	Insere linhas baseado nos valores especificados. A partir da versão 3.22.5, múltiplas listas de valores podem ser especificadas, permitindo a inserção de múltiplas linhas em um único INSERT .
SELECT	Insere as linhas retornadas pelo comando SELECT .
SET [3.22.10]	Atribui os valores das expressões às colunas. Colunas não especificadas terão valores default.

REVOKE [3.22.11]

Retira os privilégios atribuídos a usuários do MySQL através do comando **GRANT**.

```
REVOKE privilégio [(colunas)]
        [, privilégio [(colunas)] ...]
        ON {tabela | * | *.* | nomebd.*}
        FROM usuário [, ...]
```

Parâmetro	Descrição
<i>privilégio</i>	Privilégio a ser retirado.
ALL [PRIVILEGES]	Todos os privilégios.
ALTER	Alterar tabelas e índices.
CREATE	Criar bancos de dados e tabelas.
DELETE	Eliminar linhas de tabelas.
DROP	Eliminar bancos de dados e tabelas.
FILE	Ler e gravar arquivos no servidor.
INDEX	Criar e eliminar índices.
INSERT	Inserir linhas em tabelas.

Comandos SQL

PROCESS	Visualizar e eliminar quaisquer threads.
RELOAD	Recarregar as tabelas internas do MySQL (ver comando FLUSH).
SELECT	Ler dados de tabelas.
SHUTDOWN	Desativar o servidor.
UPDATE	Alterar linhas em tabelas.
USAGE	Sem privilégios.
<i>colunas</i>	Nomes das colunas das quais os privilégios devem ser retirados.
ON	Especifica o nível de privilégio.
<i>**</i>	Retira os privilégios de todas as tabelas em todos os bancos de dados do servidor.
<i>*</i>	Retira os privilégios de todas as tabelas do banco de dados corrente.
<i>bd.*</i>	Retira os privilégios de todas as tabelas do banco de dados especificado.
<i>bd.tabela</i>	Retira os privilégios de todas as colunas da tabela especificada.
<i>tabela</i>	Retira os privilégios de todas as colunas da tabela especificada do banco de dados default.
FROM	Especifica o(s) nome(s) do(s) usuário(s) a ter(em) privilégios retirados. Formato: <i>usuário@host</i> .
<i>usuário</i>	Nome do usuário ou um string vazio para especificar um usuário anônimo.
<i>host</i>	Nome do host. Pode ser: localhost, um nome de host ou um endereço IP. Os caracteres curinga '%' e '_' podem ser utilizados.

SELECT

Seleciona linhas de uma ou mais tabelas.

```
SELECT [ predicado ] [ opções_select ]  
    coluna [,... ]  
    [ INTO {OUTFILE | DUMPFILE} 'nomearq'  
    delimitadores ]  
    [ FROM ref_tabelas ]  
    [ WHERE condição ]  
    [ GROUP BY coluna,... ] [ HAVING condição ]  
    [ ORDER BY coluna [ ASC | DESC ] ,... ]  
    [ LIMIT [ linha_inicial, ] num_linhas ]  
    [ PROCEDURE nome ]  
    ]  
  
opções_select:  
    [ STRAIGHT_JOIN ]  
    [ SQL_SMALL_RESULT | SQL_BIG_RESULT ]  
    [ HIGH_PRIORITY ]
```

delimitadores:

```
[ FIELDS
  [ TERMINATED BY 'string' ]
  [ OPTIONALLY ] ENCLOSED BY 'char' ]
  [ ESCAPED BY 'char' ] ]
[ LINES TERMINATED BY 'string' ]
```

Parâmetro	Descrição
<i>predicado</i>	Restringe as linhas a serem retornadas.
DISTINCT	Omite registros que contenham dados duplicados nos campos selecionados.
DISTINCTROW	Similar a DISTINCT , mas seleciona somente quando todos os campos das linhas forem idênticos.
ALL	Seleciona todas as linhas que atendam às condições especificadas.
<i>opções_select</i>	
STRAIGHT_JOIN	Força a junção das tabelas na ordem em que elas estão especificadas na cláusula FROM. Pode ser usada caso você ache que o MySQL não está fazendo a junção da melhor forma.
SQL_SMALL_RESULT ^[3.22.12]	Pode ser usado com GROUP BY ou DISTINCT para informar ao MySQL que o resultado do SELECT será pequeno. Nesse caso, o MySQL usa tabelas temporárias para armazenar os resultados em vez de ordená-los.
SQL_BIG_RESULT ^[3.23]	Pode ser usado com GROUP BY ou DISTINCT para informar ao MySQL que o resultado do SELECT será grande. Nesse caso, o MySQL ordena os resultados em vez de usar uma tabela temporária. No caso de falta de memória, tabelas temporárias em disco podem ser utilizadas pelo MySQL.
HIGH_PRIORITY ^[3.22.9]	Atribui ao comando SELECT prioridade maior que aos comandos que atualizam tabelas. Deve ser usada somente para consultas que são muito rápidas e que devam ser feitas uma única vez.
<i>coluna</i>	Nome da coluna ou colunas a serem retornadas pela consulta. Pode-se atribuir um <i>alias</i> a uma coluna, usando a sintaxe: AS alias . O <i>alias</i> pode ser utilizado no lugar do nome nas cláusulas GROUP BY, ORDER BY e HAVING.

INTO	Redireciona a saída do comando SELECT para um arquivo. É necessário ter o privilégio FILE e o arquivo de saída não deve existir.
OUTFILE	Redireciona para um arquivo que é criado no servidor.
DUMPFIL ^[3.23.6]	Escreve apenas uma linha no arquivo sem mudança de linha e sem modificar o conteúdo. Útil para armazenar o conteúdo de um campo BLOB em um arquivo.
<i>delimitadores</i>	
FIELDS	Faz com que as opções TERMINATED BY, [OPTIONALLY] ENCLOSED BY e ESCAPED BY sejam facultativas, mas pelo menos uma deve ser declarada.
FIELDS TERMINATED BY	Especifica o caractere ou caracteres utilizados para delimitar valores na linha (default= '\t').
OPTIONALLY	Somente colunas dos tipos CHAR e VARCHAR recebem o caractere especificado por ENCLOSED BY.
FIELDS ENCLOSED BY	Especifica o caractere utilizado para delimitar o string de caracteres (default=").
FIELDS ESCAPED BY	Especifica o caractere utilizado para indicar que o próximo caractere não é um caractere especial (default= '\').
LINES TERMINATED BY	Especifica o caractere ou caracteres que indicam o final de uma linha (default= '\n').
FROM <i>ref_tabelas</i>	Especifica o nome das tabelas dos campos a serem utilizados no comando SELECT. Ver cláusula JOIN na página 21.
WHERE	Especifica a condição que determina quais linhas serão retornadas.
GROUP BY	Retorna um valor para cada grupo de registros. Cria um registro de resumo para cada registro definido.
HAVING	Similar à cláusula WHERE, mas restringe somente os registros a serem retornados com a cláusula GROUP BY.
ORDER BY	Especifica as colunas a serem usadas como chave para ordenar os registros retornados. A partir da versão 3.23.2 pode ser especificada uma fórmula.
DESC	Ordem descendente.
ASC	Ordem ascendente (default).

LIMIT	Limita o número de linhas a serem retornadas pelo comando SELECT.
<i>linha_inicial</i>	Especifica o número da primeira linha a ser retornada (inicia em 0).
<i>num_linhas</i>	Especifica o número de linhas a serem retornadas.
PROCEDURE	Especifica uma procedure para a qual os dados retornados serão enviados antes de serem retornados ao cliente. A partir da versão 3.23, pode-se usar PROCEDURE ANALYSE() para obter informações sobre as características dos dados nas colunas.

Cláusula JOIN

As seguintes sintaxes JOIN são permitidas em comandos SELECT:

ref_tabela1, *ref_tabela2*

ref_tabela1 [**CROSS**] **JOIN** *ref_tabela2* ^[3.21.18]

ref_tabela1 **INNER JOIN** *ref_tabela2* ^[3.23]

ref_tabela1 **STRAIGHT_JOIN** *ref_tabela2*

ref_tabela1 **LEFT** [**OUTER**] **JOIN** *ref_tabela2* **ON**
condição

ref_tabela1 **LEFT** [**OUTER**] **JOIN** *ref_tabela2* **USING**
(colunas)

ref_tabela1 **NATURAL LEFT** [**OUTER**] **JOIN**
ref_tabela2

oj ref_tabela1 **LEFT OUTER JOIN** *ref_tabela2* **ON**
condição

ref_tabela:

tabela [[**AS**] *alias*] [**USE INDEX** (*chaves*)] [**IGNORE INDEX** (*chaves*)]

Tipo de junção	Descrição
, (vírgula)	Retorna todas as combinações possíveis de linhas das tabelas. Normalmente a condição de junção é especificada na cláusula WHERE.
JOIN	Equivale à utilização da vírgula.
CROSS JOIN	O mesmo que JOIN .
INNER JOIN	O mesmo que JOIN .
STRAIGHT_JOIN	Similar a JOIN , exceto que a tabela da esquerda é sempre lida antes da tabela da direita. Utilizado quando o otimizador efetua a junção de forma errada.

Comandos SQL

LEFT JOIN	Retorna todas as linhas da tabela da esquerda mesmo que não exista equivalência na tabela da direita (nesse caso, preenche com NULL). A condição para a junção é determinada pelas cláusulas ON ou USING .
ON	Especifica a condição para a junção das tabelas.
USING	Especifica a lista de colunas que devem existir em ambas as tabelas. Exemplo: A LEFT B JOIN USING (C1, C2, C3, ...).
LEFT OUTER JOIN	O mesmo que LEFT JOIN. Somente para compatibilidade com ODBC.
NATURAL LEFT JOIN	Equivalente a LEFT JOIN USING.

SET [OPTION]

Define opções para a sessão corrente do MySQL.

SET [OPTION] *opção= valor, ...*

Opções	Significado
CHARACTER SET { <i>character_set</i> DEFAULT }	Especifica o <i>character_set</i> utilizado pelo programa cliente. Atualmente, a única opção é cp1251_koi8 (alfabeto Russo). A opção DEFAULT retorna o <i>character_set</i> original.
INSERT_ID = <i>n</i>	Especifica o valor a ser usado pelo próximo comando INSERT quando estiver inserindo em uma coluna AUTO_INCREMENT .
LAST_INSERT_ID = <i>n</i>	Especifica o valor a ser retornado pela função LAST_INSERT_ID() .
PASSWORD = PASSWORD ('senha')	Define a senha para o usuário corrente.
PASSWORD [FOR <i>usuário@host</i>] = PASSWORD ('senha')	Define a senha para um usuário específico no servidor host corrente.
SQL_AUTO_IS_NULL = { 0 1 } ^[3.23.5]	Se definida como 1, a última linha inserida contendo um valor AUTO_INCREMENT pode ser selecionada usando a cláusula WHERE auto_inc_col IS NULL . Usada somente por alguns programas ODBC, tal como o Microsoft Access (default=1).
SQL_BIG_SELECTS = { 0 1 }	Determina se o MySQL deve ser abortado (0) ou não (1) quando SELECT retornar mais que max_join_size linhas (default=1).
SQL_BIG_TABLES = { 0 1 }	Determina se as tabelas temporárias serão armazenadas em disco (1) ou na memória RAM (0) (default=0).

SQL_LOG_OFF = { 0 | 1 }

Determina se o arquivo de log padrão deve ser atualizado (0) ou não (1). O arquivo de log padrão registra quando o servidor iniciou, terminou, ou quando aconteceu algum erro. Esta opção não afeta o arquivo de update log.

SQL_LOG_UPDATE = { 0 | 1 } ^[3.22.5]

Determina se o arquivo de update-log deve ser atualizado (1) ou não (0). O arquivo de update log registra todos os comandos que alteram as tabelas, sendo muito utilizado para backup. Esta opção não afeta o arquivo de log padrão.

SQL_LOW_PRIORITY_UPDATES = { 0 | 1 } ^[3.22.5]

Determina se os comandos que alteram tabelas (DELETE, INSERT, REPLACE, UPDATE) devem aguardar (1) ou não (0) até que não exista nenhum comando SELECT ativo ou pendente na tabela afetada.

SQL_SELECT_LIMIT = { *n* | **DEFAULT** }

Especifica o número máximo de registros a serem retornados por um comando SELECT. O valor DEFAULT retorna todos os registros. A cláusula LIMIT do comando SELECT tem precedência sobre esta opção.

SQL_WARNINGS = { 0 | 1 } ^[3.22.11]

Se definida como 1, o MySQL deve informar a contagem de linhas, mesmo que seja para a inserção de uma única linha. Normalmente a contagem é informada somente para comandos INSERT, que inserem múltiplas linhas (default=0).

TIMESTAMP = { *valor* | **DEFAULT** }

Especifica o valor TIMESTAMP (data e hora) da sessão. O valor DEFAULT retorna a data e a hora do sistema.

SHOW**SHOW COLUMNS**

Exibe informações sobre as colunas de uma tabela. Se os tipos de colunas forem diferentes do especificado no comando CREATE TABLE, lembre-se que o MySQL às vezes altera automaticamente os tipos das colunas. SHOW FIELDS é um sinônimo de SHOW COLUMNS.

SHOW COLUMNS FROM *tabela*[**FROM** *nomebd*] [**LIKE** *curinga*]

Parâmetro	Descrição
LIKE	Limita as informações a serem exibidas. Ver operador LIKE na página 27.

SHOW DATABASES

Lista os bancos de dados no servidor MySQL.

SHOW DATABASES [LIKE *curinga*]

Parâmetro	Descrição
LIKE	Limita as informações a serem exibidas. Ver operador LIKE na página 27.

SHOW GRANTS ^[3.23.4]

Exibe informações sobre as permissões de um usuário.

SHOW GRANTS FOR *usuário@host*

SHOW INDEX

Exibe informações sobre os índices de uma tabela. SHOW KEYS é um sinônimo de SHOW INDEX.

SHOW INDEX FROM *tabela* [FROM *nomebd*]

SHOW PROCESSLIST

Exibe informações sobre os threads que estão sendo executados no servidor.

SHOW [FULL] PROCESSLIST

Parâmetro	Descrição
FULL ^[3.23.7]	Exibe todos os caracteres de cada consulta (default=100 primeiros).

SHOW STATUS ^[3.21.21]

Exibe as variáveis de status do servidor e seus valores.

SHOW STATUS

SHOW TABLE STATUS ^[3.23]

Exibe informações sobre as tabelas em um banco de dados.

**SHOW TABLE STATUS
[FROM *nomebd*] [LIKE *curinga*]**

Parâmetro	Descrição
LIKE	Limita as informações a serem exibidas. Ver operador LIKE na página 27.

SHOW TABLES

Lista as tabelas em um banco de dados.

SHOW TABLES [FROM *nomebd*] [LIKE *curinga*]

Parâmetro	Descrição
LIKE	Limita as informações a serem exibidas. Ver operador LIKE na página 27.

SHOW VARIABLES [3.21.22]

Exibe uma lista das variáveis do servidor e seus valores. Veja na página 54 a descrição dessas variáveis.

SHOW VARIABLES [LIKE *curinga*]

Parâmetro	Descrição
LIKE	Limita as informações a serem exibidas. Ver operador LIKE na página 27.

UNLOCK TABLES

Desbloqueia as tabelas que foram bloqueadas pelo usuário corrente.

UNLOCK TABLES**UPDATE**

Altera o conteúdo de linhas em uma tabela.

UPDATE [LOW_PRIORITY] *tabela*
SET *coluna=expr,coluna=expr,...*
[WHERE condição] [LIMIT *n*]

Parâmetro	Descrição
LOW_PRIORITY [3.22.5]	Retarda a execução do comando até que nenhum outro programa cliente esteja lendo a tabela.
<i>tabela</i>	Nome da tabela a ser atualizada.
SET	Especifica as colunas a serem atualizadas e os respectivos valores a serem atribuídos.
WHERE	Especifica a condição que determina quais linhas serão atualizadas. Se não especificada, todas as linhas serão atualizadas.
LIMIT [3.23.3]	Limita o número de linhas a serem atualizadas.

USE

Identifica o banco de dados default, o qual será utilizado quando as referências a uma tabela não especificarem explicitamente o nome do banco de dados.

USE *nomebd*

Operadores

Operadores Aritméticos

Operador	Sintaxe	Significado
+	$x + y$	Adiciona dois valores numéricos.
-	$x - y$	Subtrai um valor numérico de outro.
*	$x * y$	Multiplica dois valores numéricos.
/	x / y	Divide um valor numérico por outro. A divisão por zero produz um resultado NULL.
%	$x \% y$	Retorna o módulo aritmético de dois valores numéricos.

Operadores Bitwise

Operador	Sintaxe	Significado
	$x y$	Executa um "bitwise OR" em dois valores inteiros.
&	$x \& y$	Executa um "bitwise AND" em dois valores inteiros.
<<	$x \ll y$	Desloca à esquerda, em x, o número de bits definido por y.
>>	$x \gg y$	Desloca à direita, em x, o número de bits definido por y.
~	$\sim x$	Inverte todos os bits. Converte bit 1 em bit 0 e bit 0 em bit 1.

Operadores Lógicos

Operador	Sintaxe	Significado
NOT ou !	NOT x	Retornará 1 (True) se o argumento for 0 (False); caso contrário, retornará 0 (False). Exceção: NOT NULL retorna NULL.
OR ou	x OR y	Retornará 1 (True) se um dos argumentos não for 0 (False) e nem NULL.
AND ou &&	x AND y	Retornará 0 (False) se um dos argumentos for 0 (False) ou NULL; caso contrário, retornará 1 (True).

Operadores de Comparação

Operador	Sintaxe	Significado
=	$x = y$	Igualdade.
<> ou !=	$x \lt \> y$	Diferente.
<=	$x \leq y$	Menor que ou igual.
<	$x < y$	Menor que.
>=	$x \geq y$	Maior que ou igual.
>	$x > y$	Maior que.
<=> [3.23]	$x \lt \> y$	True se os dois operandos forem iguais, mesmo para comparações com NULL.
IS [NOT] NULL		Retornará 1 (True) se um valor for NULL; caso contrário, retornará 0 (False). Se NOT for especificado, retornará 1 (True) se o valor não for NULL.

expr **BETWEEN** *min* **AND** *max*

Determina se o valor de uma expressão está dentro de um intervalo de valores. A cláusula **BETWEEN** é inclusiva, isso significa que tanto *min* como *max* serão incluídos na comparação. Se *expr* for maior ou igual a *min* e *expr* for menor ou igual a *max*, **BETWEEN** retornará 1; caso contrário, retornará 0.

expr **[NOT] IN** (*valor*,...)

Retornará 1 se *expr* for um dos valores na lista **IN**; caso contrário, retornará 0. Se **NOT** for especificado, retornará 1 se não estiver na lista.

Operador Cast (BINARY) [3.23]

O operador **BINARY** converte (casts) o string que o segue para string binário. Essa é uma maneira de forçar uma comparação de coluna ser case sensitive, mesmo que a coluna não esteja definida como **BINARY** ou **BLOB**. Somente com tipos **CHAR** e **VARCHAR**.

Exemplos:

"novatec" = "NOVATEC" -> 1

"novatec" = **BINARY** "NOVATEC" -> 0

BINARY "novatec" = "NOVATEC" -> 0

Operador LIKE

Retornará 1 (True) se uma expressão corresponder a um padrão. Permite a utilização de expressões numéricas (isto é, uma extensão ao padrão ANSI SQL). Se **NOT** for especificado, retornará 1 se não possuir tiver correspondência.

expr **[NOT] LIKE** *padrão* [**ESCAPE** 'escape-char']

Parâmetro	Descrição
<i>padrão</i>	Seqüência de caracteres em relação a qual <i>expr</i> é comparada. Não é feita distinção entre minúsculas e maiúsculas. Os caracteres '%' e '_' dentro do padrão têm a seguinte interpretação:
%	Especifica qualquer seqüência de zero ou mais caracteres.
_	Especifica qualquer caractere.
ESCAPE	Especifica o caractere escape (default='\'). Um caractere escape remove o significado especial do caractere % ou \ que o segue.

Operador REGEXP

Executa a procura de um padrão dentro da string *expr*. Retornará 1 no caso de sucesso e 0 em caso de falha. Todo caractere '\' deve ser colocado como '\\' e, a partir da versão 3.23.4 é “case insensitive” para strings normais (não binárias). REGEXP usa o character set corrente (ISO-8859-1 Latin1 por default) quando tem que decidir o tipo de um caractere.

expr **REGEXP** *padrão*

expr **NOT REGEXP** *padrão*

O mesmo que **NOT** (*expr* **REGEXP** *padrão*).

expr **NOT RLIKE** *padrão*

O mesmo que **NOT REGEXP**.

expr **RLIKE** *padrão*

O mesmo que **REGEXP**.

Caracteres Especiais usados em REGEXP

Permite que diversos formatos de palavras possam ser encontrados utilizando expressões regulares.

Seqüência	Significado
^	Somente combina o início de um string.
\$	Somente combina o final de um string.
.	Combina qualquer caractere (incluindo newline).
[...]	Combina qualquer caractere entre os colchetes. Dentro dos colchetes pode ser especificado um intervalo, como por exemplo: [0-9] para selecionar os dígitos de 0 a 9.
[^...]	Combina qualquer caractere que não esteja entre os colchetes.
e*	Combina zero ou mais ocorrências do elemento e.
e+	Combina uma ou mais ocorrências do elemento e.
e?	Combina zero ou uma ocorrência do elemento e.
e1 e2	Combina com o elemento e1 ou e2.
e {m}	Combina m ocorrências do elemento e.
e {m,}	Combina m ou mais ocorrências do elemento e.
e {,n}	Combina zero ou n ocorrências do elemento e.
e {m,n}	Combina de m a n ocorrências do elemento e.
(...)	Agrupa elementos em um simples elemento.
outro	Caracteres normais combinam com eles mesmos.

Para mais informações sobre expressões regulares, consulte o manual on-line do MySQL.

Funções de Agregação

AVG(*expr*)

Retorna a média de *expr* para valores não-NULL nas linhas selecionadas.

BIT_AND(*expr*) [3.21.11]

Retorna o resultado da operação lógica AND em todos os bits de *expr*.

BIT_OR(*expr*) [3.21.11]

Retorna o resultado da operação lógica OR em todos os bits de *expr*.

COUNT(*expr*)

Retorna o número de valores não-NULL nas linhas retornadas pelo comando SELECT.

COUNT(*)

Retorna o número de valores nas linhas retornadas pelo comando SELECT (inclusive valores NULL).

COUNT(DISTINCT *expr* [,...]) [3.23.2]

Retorna o número de valores não-NULL nas linhas retornadas pelo comando SELECT excluindo os valores duplicados.

MAX(*expr*)

Retorna o valor máximo de *expr* para valores não-NULL nas linhas selecionadas.

MIN(*expr*)

Retorna o valor mínimo de *expr* para valores não-NULL nas linhas selecionadas.

STD(*expr*)

Retorna o desvio-padrão de *expr* para valores não-NULL nas linhas selecionadas.

STDDEV(*expr*)

O mesmo que **STD()**.

SUM(*expr*)

Retorna a soma de *expr* para valores não-NULL nas linhas selecionadas.

Funções de Comparação

GREATEST(*expr1, expr2,...*) ^[3.22.5]

Retorna o maior valor dentre os argumentos fornecidos. A função **GREATEST** substituiu a função **MAX** a partir da versão 3.22.5.

IF(*expr1,expr2,expr3*)

Retornará *expr2* se *expr1* for True (não 0 ou NULL); caso contrário, retornará *expr3*. **IF()** retornará um valor numérico ou string, dependendo do contexto em que for utilizado.

IFNULL(*expr1, expr2*)

Retornará *expr2* se *expr1* = NULL; caso contrário, retornará *expr1*.

INTERVAL(*n,n1,n2,...*)

Retornará 0 se $n < n1$, 1 se $n < n2$ e assim por diante, ou -1 se *n* for NULL. Os valores *n1*, *n2*, ... devem estar em ordem crescente ($n1 < n2 < \dots$) porque é feita uma pesquisa binária.

ISNULL(*expr*)

Retornará 1 (True) se a *expr* for NULL; caso contrário, retornará 0 (False). Observe que uma comparação de valores NULL usando o caractere "=" será sempre False.

LEAST(*expr1, expr2,...*) ^[3.22.5]

Retorna o menor valor dentre os argumentos fornecidos. A função **LEAST** substituiu a função **MIN** a partir da versão 3.22.5.

STRCMP(*str1, str2*)

Retornará 0 se os strings forem iguais, -1 se o primeiro argumento for menor que o segundo de acordo com a ordem de classificação corrente; e 1 caso contrário.

Função CASE ^[3.23.3]

CASE

Formato 1:

Compara uma expressão a um conjunto de expressões simples para determinar o resultado. Retorna o resultado onde *valor = valor_compara*.

CASE valor

WHEN [*valor_compara*] **THEN** *resultado*
 [**WHEN** [*valor_compara*] **THEN** *resultado...*]
 [**ELSE** *resultado*]

END

Argumento	Significado
WHEN	Especifica o valor a ser comparado.
THEN	Especifica o resultado se houver correspondência com o valor especificado em WHEN.
ELSE	Se não for encontrada nenhuma correspondência de valores, então será retornado o resultado após ELSE. Se ELSE não for especificado, então será retornado NULL.

Formato 2:

Avalia um conjunto de expressões booleanas para determinar o resultado. Retorna o resultado da primeira condição verdadeira (TRUE).

CASE

WHEN [*condição*] **THEN** *resultado*
 [**WHEN** [*condição*] **THEN** *resultado...*]
 [**ELSE** *resultado*]

END

Argumento	Significado
WHEN	Especifica a condição a ser verificada.
THEN	Especifica o resultado caso a condição WHEN for satisfeita.
ELSE	Se nenhuma condição for satisfeita, então será retornado o resultado após ELSE. Se ELSE não for especificado, então será retornado NULL.

Exemplo:

```
mysql> SELECT CASE WHEN 1>0 THEN "true" ELSE "false" END;  
-> "true"
```


Funções Numéricas

ABS(x)

Retorna o valor absoluto de x .

ACOS(x) ^[3.21.8]

Retorna o arco co-seno de x , ou NULL se x não estiver entre -1 e 1.

ASIN(x) ^[3.21.8]

Retorna o arco seno de x , ou NULL se x não estiver entre -1 e 1.

ATAN(x) ^[3.21.8]

Retorna o arco tangente de x .

ATAN2(x, y)

Retorna o arco tangente de duas variáveis x e y . Similar a ATAN2(x, y), exceto que usa os sinais dos argumentos para determinar o quadrante do valor retornado.

CEILING(x)

Retorna o menor número inteiro maior ou igual a x .

COS(x) ^[3.21.8]

Retorna o co-seno *de* x , onde x é fornecido em radianos.

COT(x) ^[3.21.16]

Retorna a cotangente de x , onde x é fornecido em radianos.

DEGREES(x) ^[3.21.16]

Retorna o valor de x convertido de radianos para graus.

EXP(x)

Retorna e^x ($e=2.711828$).

FLOOR(x)

Retorna o maior número inteiro menor ou igual a x .

LOG(x)

Retorna o logaritmo natural (base e) de x .

LOG10(x)

Retorna o logaritmo (base 10) de x .

MOD(m,n)

Retorna o resto da divisão de m por n . O mesmo que $m \% n$.

PI() ^[3.21.8]

Retorna o valor de PI.

POW(x,y)

Retorna o valor de x^y .

POWER(x,y) ^[3.21.16]

O mesmo que POW(x,y).

RADIANS(x) ^[3.21.16]

Retorna o valor de x convertido de graus para radianos.

RAND([n])

Retorna um valor aleatório de ponto-flutuante no intervalo de 0.0 a 1.0.

Argumento	Descrição
-----------	-----------

n	Valor da semente para a geração de números aleatórios. Chamadas com o mesmo valor de n sempre retornam o mesmo resultado.
-----	---

ROUND(x)

Retorna o valor de x arredondado para um inteiro.

ROUND(x,d)

Retorna o valor de x arredondado para um número com d casas decimais. Se $d = 0$, o resultado não terá ponto decimal ou parte fracionária.

SIGN(x)

Retorna -1, 0 ou 1, dependendo de x ser negativo, zero ou positivo.

SIN(x) ^[3.21.8]

Retorna o seno de x , onde x é fornecido em radianos.

SQRT(x)

Retorna a raiz quadrada de x .

TAN(x) ^[3.21.8]

Retorna a tangente de x , onde x é fornecido em radianos.

TRUNCATE(x,d) ^[3.21.16]

Retorna o valor de x truncado para d casas decimais.

Funções de String

ASCII(*str*) ^[3.21.2]

Retorna o código ASCII do caractere mais à esquerda de *str*. Retornará 0 se *str* for um string vazio ou NULL se *str* for NULL.

BIN(*n*) ^[3.22.4]

Retorna um string com a representação binária do valor de *n*, onde *n* é um número Longlong (BIGINT). Equivale à função **CONV**(*n*,10,2).

CHAR(*n1,n2,...*) ^[3.21.0]

Retorna os caracteres correspondentes ao códigos ASCII especificado.

CHAR_LENGTH(*str*)

O mesmo que **LENGTH**().

CHARACTER_LENGTH(*str*)

O mesmo que **LENGTH**().

COALESCE(*expr1, expr2, ...*) ^[3.23.3]

Retorna o primeiro elemento não-NULL da lista.

CONCAT(*str1,str2,...*)

Retorna o string resultante da concatenação dos strings especificados. Retornará NULL se qualquer um dos argumentos for NULL.

CONV(*n,base1,base2*) ^[3.22.4]

Retorna a representação string do número *n* convertido da base *base1* para a base *base2*.

ELT(*n,str1,str2,...*)

Retorna o *n*-ésimo string da lista *str1,str2,...*. Retornará NULL se não existir o *n*-ésimo elemento ou se o string em *n* for NULL. **ELT**() é o complemento de **FIELD**().

EXPORT_SET(*n, on, off, [separador, [numbits]]*) ^[3.23.2]

Retorna um string consistindo dos strings *on* e *off*, separados pelo string *separador*.

Parâmetro	Significado
<i>on</i>	É usado para cada bit ligado a <i>n</i> .
<i>off</i>	É usado para bit desligado de <i>n</i> .
<i>numbits</i>	Número máximo de bits a serem verificados em <i>n</i> (default=64).
<i>separador</i>	String utilizado para separar os valores <i>on</i> e <i>off</i> (default=",").

FIELD(*str, str1, str2, ...*)

Retorna o índice de *str* na lista de strings *str1, str2, ...*. Retornará 0 se *str* não for encontrado ou se *str* for NULL. **FIELD()** é o complemento de **ELT()**.

FIND_IN_SET(*str, lista_str*) ^[3.21.22]

Retorna o índice de *str* no string *lista-str*. *lista_str* é um string que consiste de substrings separados por vírgulas.

FORMAT(*x, d*)

Retorna o número *x* formatado ('#',###,###.##'), arredondado para *d* casas decimais.

HEX(*n*) ^[3.22.4]

Retorna um string com a representação hexadecimal do valor *n*. Equivale a **CONV(*n*,10,16)**.

INSERT(*str, pos, tam, novostr*)

Retorna o string *str*, com a parte iniciando na posição *pos* e com o tamanho *tam* sendo substituída pelo string *novostr*.

INSTR(*str, substr*)

Retorna a posição da primeira ocorrência do substring *substr* no string *str*. Similar à função **LOCATE(*substr, str*)**, exceto que os argumentos são invertidos.

LCASE(*str*)

Retorna o string *str* com os caracteres convertidos para minúsculas.

LEFT(*str, tam*)

Retorna *tam* caracteres iniciais do string *str*.

LENGTH(*str*)

Retorna o número de caracteres do string *str*.

LOCATE(*substr*,*str*,*pos*)

Retorna a posição da primeira ocorrência do substring *substr* no string *str*, iniciando a pesquisa a partir da posição *pos*. Retornará 0 se *substr* não estiver contido em *str*.

LOCATE(*substr*,*str*)

Retorna a posição da primeira ocorrência do substring *substr* no string *str*. Retornará 0 se *substr* não estiver contido em *str*.

LOWER(*str*)

O mesmo que LCASE().

LPAD(*str*, *tam*, *padstr*) ^[3.22.2]

Retorna um string consistindo do valor do string *str*, preenchido à esquerda com o string *padstr* até o tamanho *tam*.

LTRIM(*str*)

Retorna o string *str* sem os espaços em branco iniciais.

MAKE_SET(*n*,*str_bit0*,*str_bit1*,...) ^[3.22.2]

Retorna o valor SET (um string contendo substrings separados por vírgulas). Para cada bit ligado a *n*, o string correspondente ao bit (*str_bit0*, *str_bit1*,...) será incluído no resultado.

MID(*str*,*pos*,*tam*)

Retorna um substring do string *str*, a partir da posição *pos* e tamanho *tam*.

OCT(*n*) ^[3.22.4]

Retorna um string com a representação octal do valor *n*. Equivale a **CONV**(*n*,10,8).

OCTET_LENGTH(*str*)

O mesmo que LENGTH().

POSITION(*substr* **IN** *str*)

O mesmo que LOCATE(*substr*, *str*).

REPEAT(*str*,*n*) ^[3.21.10]

Retorna um string consistindo do string *str* repetido *n* vezes.

REPLACE(*str*,*de_str*,*para_str*)

Retorna um string consistindo do string *str* com todas as ocorrências do string *de_str* substituídas pelo string *para_str*.

REVERSE(*str*) ^[3.21.19]

Retorna o string *str* com a ordem dos caracteres invertida.

RIGHT(*str*,*tam*)

Retorna *tam* caracteres finais do string *str*.

RPAD(*str*,*tam*,*padstr*) ^[3.22.2]

Retorna um string consistindo do valor do string *str*, preenchido à direita com o string *padstr* até o tamanho *tam*.

RTRIM(*str*)

Retorna o string *str* sem os espaços em branco finais.

SOUNDEX(*str*)

Retorna o código fonético de um string. Utilizada em aplicações para pesquisar nomes com sons parecidos.

SPACE(*n*) ^[3.21.16]

Retorna um string com *n* espaços.

SUBSTRING(*str*,*pos*,*tam*)

O mesmo que MID(*str*, *pos*, *tam*).

SUBSTRING(*str* FROM *pos* FOR *tam*)

O mesmo que MID(*str*, *pos*, *tam*).

SUBSTRING(*str*,*pos*) ou SUBSTRING(*str* FROM *pos*)

Retorna um substring do string *str* iniciando na posição *pos*.

SUBSTRING_INDEX(*str*,*delim*,*n*) ^[3.21.15]

Retorna um substring do string *str* após *n* ocorrências de *delim*. Se *n* for positivo, retornará os caracteres à esquerda do delimitador *delim* encontrado; se negativo, retornará os caracteres à direita do delimitador *delim*. Se *delim* não for encontrado retornará o string inteiro.

TRIM([[opção] [remstr] FROM] str) ^[3.21.12]

Retorna o string *str* com os caracteres *remstr* iniciais e/ou finais removidos.

Parâmetro	Descrição
<i>opção</i>	Indica a localização dos caracteres a serem removidos.
BOTH	Remove os caracteres <i>remstr</i> iniciais e finais de <i>str</i> (default).
LEADING	Remove os caracteres <i>remstr</i> iniciais de <i>str</i> .
TRAILING	Remove os caracteres <i>remstr</i> finais de <i>str</i> .
<i>remstr</i>	String com os caracteres a serem removidos de <i>str</i> . Se não especificado removerá os caracteres em banco.
<i>str</i>	String fonte.

UCASE(str)

Retorna o string *str* com os caracteres convertidos para maiúsculas.

UPPER(str)

O mesmo que UCASE().

Funções de Data e Hora

ADDDATE(*data inicial*, **INTERVAL** *expr tipo*) ^[3.22]

O mesmo que DATE_ADD().

CURDATE()

Retorna a data corrente como um string no formato "AAAA-MM-DD", ou como um número no formato AAAAMMDD, dependendo do contexto utilizado.

CURRENT_DATE

O mesmo que CURDATE(), exceto que não utiliza parênteses ().

CURRENT_TIME

O mesmo que CURTIME(), exceto que não utiliza parênteses ().

CURRENT_TIMESTAMP

O mesmo que NOW(), exceto que não utiliza parênteses ().

CURTIME() ^[3.21.12]

Retorna o horário corrente como um string no formato "HH:MM:SS", ou como um número no formato HHMMSS, dependendo do contexto utilizado.

DATE_ADD(*data inicial*, **INTERVAL** *expr intervalo*) ^[3.22.4]

Retorna o resultado de uma data (ou de uma data e hora) adicionada de um intervalo especificado.

Argumento	Descrição	
<i>data inicial</i>	Data inicial no formato DATETIME ou DATE.	
<i>expr</i>	Valor do intervalo a ser adicionado à data inicial. Pode ser especificado como um número ou um string. Por exemplo: INTERVAL "2:3" YEAR_MONTH.	
<i>intervalo</i>	Especifica como interpretar o intervalo.	
Intervalo	Significado	Formato do valor
SECOND	Segundos	ss
MINUTE	Minutos	mm
HOURL	Horas	hh
DAY	Dias	DD
MONTH	Meses	MM
YEAR	Anos	YY
MINUTE_SECOND	Minutos e segundos	"mm:ss"
HOURL_MINUTE	Horas e minutos	"hh:mm"
HOURL_SECOND	Horas, minutos e segundos	"hh:mm:ss"
DAY_HOURL	Dias e horas	"DD hh"
DAY_MINUTE	Dias, horas e minutos	"DD hh:mm"
DAY_SECOND	Dias, horas, minutos e segundos	"DD hh:mm:ss"
YEAR_MONTH	Anos e meses	"YY-MM"

DATE_FORMAT(*data*,*formato*) ^[3.21.14]

Retorna uma data formatada. Os seguintes especificadores podem ser utilizados no string de formatação.

Especificador Significado

%S, %s	Segundos (00..59)
%i	Minutos (00..59)
%H	Hora, no formato numérico - 24 horas (00..23)
%h, %l	Hora, no formato numérico - 12 horas (01..12)
%k	Hora, no formato numérico - 24 horas (0..23)
%l	Hora, no formato numérico - 12 horas (1..12)
%T	Horário no formato 24 horas (hh:mm:ss)
%r	Horário no formato 12 horas (hh:mm:ss AM/PM)
%p	AM ou PM
%W	Dia da semana por extenso (Sunday..Saturday)
%a	Dia da semana abreviado (Sun..Sat)
%d	Dia do mês (00..31)
%e	Dia do mês (0..31)
%D	Dia do mês com sufixo inglês (1st, 2nd, 3rd, ...)
%w	Dia da semana (0=Domingo..6=Sábado)
%j	Dia do ano (001..366)
%U	Semana (0..52), onde Domingo é o primeiro dia da semana
%u	Semana (0..52), onde Segunda é o primeiro dia da semana
%M	Nome do mês por extenso (January..December)
%b	Nome do mês abreviado (Jan..Dec)
%m	Mês (01..12)
%c	Mês (1..12)
%Y	Ano (4 dígitos)
%y	Ano (2 dígitos)
%%	Um literal '%'

Todos os outros caracteres são apenas copiados para o resultado sem nenhuma interpretação.

DATE_SUB(*data*inicial, INTERVAL *expr* *tipo*) ^[3.22.4]

Retorna o resultado de uma data (ou de uma data e hora) subtraído um intervalo de tempo especificado.

Argumento Descrição

<i>data</i> inicial	Data inicial no formato DATETIME ou DATE.
<i>expr</i>	Valor do intervalo a ser subtraído da data inicial. Pode ser especificado como um número ou um string. Por exemplo: INTERVAL "2:3" YEAR_MONTH.
<i>intervalo</i>	Especifica como interpretar o intervalo. Veja a tabela na função DATE_ADD().

DAYNAME(*data*) ^[3.21.23]

Retorna o dia da semana por extenso de uma data.

DAYOFMONTH(*data*) ^[3.21.22]

Retorna o dia do mês (1-31) de uma data.

DAYOFWEEK(*data*) ^[3.21.15]

Retorna o valor numérico do dia da semana de uma data (1=Domingo,...,7=Sábado).

DAYOFYEAR(*data*) ^[3.21.22]

Retorna o dia do ano (1-366) de uma data.

EXTRACT(*intervalo* FROM *data*) ^[3.23.0]

Retorna a parte da data indicada por intervalo. Veja na função DATE_ADD() os especificadores para intervalo.

FROM_DAYS(*numdias*)

Retorna a data correspondente a um número de dias desde o ano 0. Somente para datas do calendário Gregoriano (1582 em diante).

FROM_UNIXTIME(*unix_timestamp*) ^[3.21.5]

Retorna a representação do argumento *unix_timestamp*, como uma data e hora no formato "AAAA-MM-DD HH:MM:SS", ou como um número no formato AAAAMMDDHHMMSS, dependendo do contexto utilizado.

FROM_UNIXTIME(*unix_timestamp*,*formato*) ^[3.21.8]

Retorna a representação string do Unix timestamp de acordo com o formato especificado.

Argumento	Descrição
-----------	-----------

formato	Utiliza o mesmos especificadores da função DATE_FORMAT().
---------	---

HOURL(*horário*) ^[3.21.22]

Retorna as horas (0-23) de um horário.

MINUTE(*horário*) ^[3.21.22]

Retorna os minutos (0-59) de um horário.

MONTH(*data*) ^[3.21.22]

Retorna o mês (1-12) de uma data.

MONTHNAME(*data*) ^[3.21.23]

Retorna o mês por extenso de uma data.

NOW()

Retorna a data e hora corrente como um string no formato "AAAA-MM-DD HH:MM:SS", ou como um número no formato AAAAMMDDHHMMSS, dependendo do contexto utilizado.

PERIOD_ADD(*período*, *n*)

Retorna o resultado da adição de *n* meses ao período especificado. O formato do resultado é AAAAMM. O formato do período pode ser AAAAMM ou AAMM.

PERIOD_DIFF(*per1*,*per2*)

Retorna o número de meses entre os períodos *per1* e *per2*. O formato dos argumentos *per1* e *per2* pode ser AAAAMM ou AAMM.

QUARTER(*data*) ^[3.21.22]

Retorna o trimestre (1-4) do ano de uma data.

SEC_TO_TIME(*segundos*) ^[3.21.5]

Retorna o argumento *segundos*, convertido para horas, minutos e segundos, como um string no formato "HH:MM:SS", ou como um número no formato HHMMSS, dependendo do contexto utilizado.

SECOND(*horário*) ^[3.21.22]

Retorna os segundos (0-59) de um horário.

SUBDATE(*data_inicial*, INTERVAL *expr* *tipo*) ^[3.22]

O mesmo que DATE_SUB().

SYSDATE()

O mesmo que NOW().

TIME_FORMAT(*horário*,*formato*) ^[3.21.3]

Similar à função **DATE_FORMAT()**, mas o string formato pode conter somente especificadores de formato que manipulem horas, minutos e segundos. Outros especificadores retornam um valor NULL ou 0.

TIME_TO_SEC(*horário*) ^[3.21.16]

Retorna um horário ("hh:mm:ss") convertido para o número de segundos.

TO_DAYS(*data*)

Retorna o número de dias desde a data especificada até o ano 0.

UNIX_TIMESTAMP([*data*])

Retorna o número de segundos desde a data especificada, ou desde "1970-01-01 00:00:00 GMT" se a data não for especificada.

Argumento	Descrição
-----------	-----------

<i>data</i>	Pode ser um string DATE, um string DATETIME, um TIMESTAMP, ou um número no formato AAMMDD ou AAAAMMDD na hora local.
-------------	--

WEEK(*data*) ^[3.21.22]

Retorna o número da semana (0-52) de uma data, sendo Domingo o primeiro dia da semana.

WEEK(*data*, *primeiro*) ^[3.21.1]

Retorna o número da semana (0-52) de uma data. O argumento *primeiro* especifica se a semana inicia no Domingo (0) ou na Segunda (1).

WEEKDAY(*data*)

Retorna o valor numérico do dia da semana de uma data (0=Segunda,...,6=Domingo).

YEAR(*data*) ^[3.21.22]

Retorna o ano de uma data, no intervalo 1000 a 9999.

Miscelânea

BENCHMARK(*n*,*expr*) ^[3.22.5]

Executa *n* vezes uma expressão. Utilizada para medir o tempo necessário para o utilitário cliente *mysql* processar uma expressão para análise de desempenho.

BIT_COUNT(*n*)

Retorna o número de bits ligados ao argumento *n* (BIGINT de 64 bits).

DATABASE()

Retorna o nome do banco de dados corrente, ou um string vazio se não existir.

DECODE(*str_cript*,*senha*)

Descriptografa *str_cript* utilizando a senha especificada. Contrasta com **ENCODE**()

ENCODE(*str*,*senha*)

Retorna um string binário com o string *str* criptografado utilizando a senha especificada. Para descriptografar o resultado, utilize a função **DECODE**()

ENCRYPT(*str*) ^[3.21.12]

Retorna o string *str* criptografado através da system call *crypt*() do Unix. Se *crypt*() não estiver disponível, **ENCRYPT**() retornará NULL.

ENCRYPT(*str*, *salt*) ^[3.22.16]

Retorna o string *str* criptografado através da system call *crypt*() do Unix. Se *crypt*() não estiver disponível, **ENCRYPT**() retornará NULL.

Parâmetro	Significado
-----------	-------------

<i>salt</i>	String de dois caracteres nas versões anteriores a versão 3.22.16. A especificação de salt faz com que o resultado da criptografia do string <i>str</i> seja sempre o mesmo.
-------------	--

GET_LOCK(*str*,*timeout*) ^[3.21.7]

Tenta efetuar um lock em *str*, no tempo especificado por *timeout* (em segundos). Retornará 1 se tiver sucesso, 0 se tiver ocorrido timeout, NULL se tiver ocorrido um erro. O lock é desbloqueado quando um **RELEASE_LOCK**() ou um novo **GET_LOCK**() é executado, ou quando o thread termina.

LAST_INSERT_ID()

Retorna o último valor `AUTO_INCREMENT` gerado na sessão corrente, ou 0 se tal valor não tiver sido gerado.

LAST_INSERT_ID(*expr*) ^[3.22.9]

Retorna o valor de *expr* como se tivesse sido gerado automaticamente para `AUTO_INCREMENT`.

LOAD_FILE(*nomearq*)

Lê um arquivo no servidor e retorna o seu conteúdo em um string. Você deve ter o privilégio **FILE**. Retornará `NULL` se o arquivo não puder ser lido por alguma razão. O tamanho do arquivo deve ser menor que **max_allowed_packet**.

MD5(*string*) ^[3.23.2]

Retorna um checksum (hexa 32 long) baseado no algoritmo MD5 (Message-Digest Algorithm da RSA) do string.

PASSWORD(*str*)

Retorna uma senha criptografada a partir do string *str*. Não utiliza o mesmo algoritmo para criptografia do Unix.

RELEASE_LOCK(*str*) ^[3.21.7]

Desbloqueia o lock do nome *str* criado com a função **GET_LOCK()**. Retornará 1 se o lock tiver sido desbloqueado, 0 se o lock não tiver sido bloqueado por este thread (continua bloqueado) e `NULL` se o o lock *str* não existir.

SYSTEM_USER()

O mesmo que `USER()`.

SESSION_USER()

O mesmo que `USER()`.

USER()

Retorna o nome do usuário corrente do MySQL, no formato "*usuário@host*".

VERSION() ^[3.21.13]

Retorna um string indicando a versão do servidor MySQL e um dos seguintes sufixos:

Sufixo	Significado
-log	O logging está ativado.
-debug	O servidor está sendo executado no modo debug.
-demo	O servidor está sendo executado no modo demo.
-shareware	A versão do servidor é shareware.

Comentários no código SQL

Os caracteres #, -- e /* */ são utilizados para indicar comentários no código SQL:

```
# comentário para o fim da linha
-- comentário para o fim da linha. Requer pelo menos um espaço após --
/* comentário de uma única linha */
/*
este comentário ocupa
mais de uma linha
*/
```

Existem algumas limitações no uso de /*...*/:

- Caracteres como apóstrofo e aspas indicam o início ou o fim de um string, mesmo dentro do comentário.
- Ponto e vírgula indicam o final do comando. Tudo que estiver após o ponto e vírgula é interpretado como um novo comando.

Tipo especial de comentário

A partir do MySQL 3.23 é possível desabilitar recursos específicos do MySQL, utilizando a forma especial de comentário /*! ... */ dentro de um comando. O sinal de ! indica que o MySQL deve interpretar e executar o código dentro dos comentários, enquanto outros servidores ignoram o que estiver entre /* e */. Por exemplo:

```
SELECT /*! STRAIGHT_JOIN */ coluna
FROM tabela1, tabela2 WHERE ...
```

No exemplo acima somente o MySQL considerará a opção STRAIGHT_JOIN.

Também é possível desabilitar recursos de versões antigas do MySQL. Se você adicionar o número da versão após o sinal '!', a sintaxe será executada somente se a versão do MySQL for igual ou mais recente que o número da versão utilizada:

```
CREATE /*!32302 TEMPORARY */ TABLE (a int);
```

O comando acima significa que o MySQL utilizará a palavra-chave TEMPORARY se a versão for 3.23.02 ou mais recente.

mysqld

O servidor MySQL. O script `safe_mysqld` pode ser utilizado para iniciar e monitorar o servidor `mysqld`.

isamchk e myisamchk

Utilizados para mostrar informações, otimizar e remover erros de tabelas. Mais detalhes na página 56.

myisampack e pack_isam

Utilitários para a criação de tabelas compactadas, somente para leitura. Mais detalhes na página 58.

mysql

Utilizado para interagir com o servidor de banco de dados enviando comandos e recebendo respostas. Mais detalhes na página 60.

mysqlaccess

Script para exibir as permissões de acesso a bancos de dados. Mais detalhes na página 62.

mysqladmin

Verifica o status do servidor MySQL e executa tarefas administrativas. Mais detalhes na página 63.

mysqlbug

Script utilizado para preencher um relatório de erros e problemas encontrados em programas do MySQL. Mais detalhes na página 65.

mysqldump

Efetua dump da estrutura e dos dados do bancos de dados e tabelas MySQL. Mais detalhes na página 66.

mysqlimport

Importa dados de arquivos de texto e insere nas respectivas tabelas usando `LOAD DATA INFILE`. Mais detalhes na página 69.

mysqlshow

Exibe informações sobre bancos de dados, tabelas, colunas e índices. Mais detalhes na página 71.

Opções de Programas

As opções de programas podem ser especificadas de três maneiras: na linha de comando, em arquivos de opções, ou através de variáveis de ambientes. As opções especificadas na linha de comando têm precedência sobre as opções especificadas de outra forma e as opções em arquivos de opções têm precedência sobre os valores de variáveis de ambiente.

Opções padrão de linha de comando

Opção	Significado
-# opções, --debug=opções	Opções de debugging, separadas por ":". Somente tem efeito se o MySQL foi gerado com suporte de debugging habilitado. O conjunto de opções mais comum é "d:t:o, <i>nomearq</i> ", onde <i>nomearq</i> é o nome do arquivo de log de debug. Veja o arquivo <code>dbug/dbug.c</code> da distribuição fonte do MySQL para a descrição de todas as opções disponíveis.
-C, --compress ^[3.22.3]	Habilita a compressão para o protocolo usado para comunicação entre o programa cliente e o servidor, se ambos permitirem. Opção usada somente por programas clientes.
--defaults-file=path ^[3.22.23]	Especifica o path para um arquivo de opções. Se esta opção não for especificada, o programa pesquisará a existência de arquivos de opções em vários diretórios. Deve ser a primeira opção na linha de comandos.
-?, --help	Exibe o texto de ajuda do utilitário.
-h nome, --host=nome	Especifica o host onde o servidor MySQL está sendo executado. Opção usada somente por programas clientes.
--no-defaults	Suprime o uso de arquivos de opções. Deve ser a primeira opção na linha de comandos.
-p[senha], --password=[senha]	Senha utilizada para conexão no servidor MySQL. Se a senha não for especificada após o nome da opção, o programa solicitará a senha. Opção usada somente por programas clientes.
-W, --pipe	Usa um "named pipe" para se conectar ao servidor MySQL. Opção usada somente por programas clientes no Windows.

-P <i>n</i>, --port=<i>n</i>	Para programas clientes especifica o número da porta usada em conexões TCP/IP com o servidor MySQL. Para o servidor mysqld, especifica o número da porta onde ele receberá solicitações de aplicações clientes.
--print-defaults	Exibe as opções default do programa se as mesmas não tiverem sido especificadas na linha de comando. Deve ser a primeira opção na linha de comandos.
-O <i>variável=opção</i>, --set-variable <i>variável=opção</i>	Atribui um valor a uma variável de ambiente. Ver a lista de variáveis na página 51.
-s, --silent	Exibe somente mensagens de erro. Pode ser especificada várias vezes para incrementar o efeito (-ss , -sss ,...).
-S <i>path</i>, --socket=<i>path</i>	Path completo do arquivo de socket para conexão com o servidor MySQL. Utilizado em conexões localhost . Opção usada somente por programas clientes.
-u <i>nome</i>, --user=<i>nome</i>	Nome do usuário usado para conexão com o servidor MySQL. Se esta opção não for especificada, assumirá o nome de login no Unix e ODBC no Windows. Para o mysqld, indica o nome da conta no Unix usada para executar o mysqld (deve ser root para esta opção ser efetiva).
-v, --verbose	Exibe informações detalhadas sobre a execução do programa. Pode ser especificada várias vezes para incrementar o efeito (-vv , -vvv ,...).
-V, --version	Exibe informações sobre a versão do programa.

Arquivos de Opções ^[3.22]

As opções de programas do MySQL podem ser especificadas em arquivos de opções. O formato destes é similar ao formato INI do Windows. O arquivo é dividido em grupos, sendo cada grupo identificado pelo nome do grupo entre []. As opções relacionadas ao grupo são especificadas em seguida ao nome. Cada grupo consiste do nome de um programa cliente MySQL ou um programa servidor no qual se deseja que as opções tenham efeito. O grupo [client] afeta todos os programas clientes (tudo exceto mysqld). Comentários são precedidos de # ou ;.

Localização default no Unix:

Arquivo	Descrição
/etc/my.cnf	Opções globais.
DATADIR/my.cnf	Opções específicas do servidor.
~/my.cnf	Opções específicas do usuário.

Localização default no Windows:

Arquivo	Descrição
windows\my.ini	Opções globais.
C:\my.cnf	Opções globais.
DATADIR\my.cnf	Opções específicas do servidor.

O MySQL tenta ler os arquivos de opções na ordem especificada acima. A abrangência das opções tem diferentes significados, dependendo da localização do arquivo de opções. Por exemplo, no caso do Unix, se o arquivo estiver armazenado em /etc/my.cnf, as opções aplicar-se-ão a todos os servidores MySQL e clientes no computador. Se estiver localizado no diretório data do servidor MySQL (isto é, /usr/local/mysql/data/my.cnf) as opções aplicar-se-ão somente ao servidor MySQL. Se estiver localizado no diretório home do usuário, afetará os clientes executados pelo cliente.

Veja abaixo um exemplo de arquivo de opções:

[client]

port=9999

socket=/dev/mysql

[mysqld]

port=9999

socket=/dev/mysql

set-variable = join_buffer=1M

[mysql]

host=dbhost

unbuffered

Variáveis de Ambiente

Os programas MySQL utilizam variáveis de ambiente ("environment") para obter definições de opções. As variáveis de ambiente têm precedência menor, podem ser sobrepostas pelas opções definidas no arquivo de opções ou na linha de comando.

MYSQL_DEBUG

Opções usadas em depuração. Similar à opção **--debug**.

MYSQL_PWD

Especifica a senha utilizada para conexão com o servidor MySQL. Similar à opção **--password**.

MYSQL_TCP_PORT

Para programas clientes, é o número da porta usada para estabelecer uma conexão TCP/IP com o servidor MySQL. Para mysqld, é o número de porta usada para atender conexões TCP/IP.

MYSQL_UNIX_PORT

Para programas clientes, é o pathname do arquivo socket Unix usado para estabelecer uma conexão com o servidor MySQL. Para o servidor mysqld, é o socket utilizado para atender conexões locais. Similar à opção **--socket**.

TMPDIR

Path do diretório de arquivos temporários (default='/tmp'). Similar à opção **--tmpdir**.

USER

Nome do usuário utilizado para se conectar ao servidor MySQL. Somente para programas clientes executados sob o Windows. Similar à opção **--user**.

mysqld

Servidor de banco de dados MySQL. Deve estar sempre em execução para que as tabelas possam ser acessadas.

O script `safe_mysqld` inicia o servidor `mysqld` com algumas características de segurança como iniciar o servidor novamente quando um erro ocorrer e escrever informações sobre a execução do servidor em um arquivo de log.

mysqld [opções]

Opções padrão usadas pelo mysqld

Opção	Significado
<code>--debug</code>	Opções de debugging.
<code>--help</code>	Exibe o texto de ajuda.
<code>--port</code>	Número da porta para receber solicitações de aplicações cliente.
<code>--socket</code>	Path completo para o arquivo socket.
<code>--user</code>	Nome do usuário da conta usada para executar o servidor.
<code>---version</code>	Exibe a versão do programa.

Veja a descrição detalhada dessas opções na página 48.

Opções específicas do mysqld

Opção	Significado
<code>-b path, --basedir=path</code>	Path do diretório de instalação do MySQL.
<code>--big-tables</code>	Armazena os resultados temporários em disco permitindo que o resultados das queries possam ser maiores, evitando a maioria dos erros de <i>"table full"</i> .
<code>--bind-address=endIP</code>	Endereço IP no qual o servidor deve receber as requisições.
<code>--bootstrap</code>	Opção utilizada pelos scripts de instalação do MySQL.
<code>--chroot=path</code> ^[3.22.2]	Usa <code>chroot</code> no servidor de mysql durante a inicialização (Unix).
<code>-h path, --datadir=path</code>	Localização dos arquivos de banco de dados.
<code>--default-table-type=tipo</code> ^[3.23]	Especifica o tipo de tabela: ISAM, MYISAM ou HEAP (default=MYISAM).
<code>--enable-locking</code> ^[3.22.4]	Habilita o sistema de LOCK.
<code>-T [n], --exit-info[=n]</code> ^[3.22]	Exibe informações de debug quando o programa é encerrado.
<code>--init-file=arquivo</code>	Especifica um arquivo de comandos SQL a ser executado na inicialização (startup time).
<code>-L nome, --language=nome</code>	Linguagem utilizada para exibir as mensagens de erro. Por exemplo: <code>german</code> .

-l path, --log[=path]	Especifica o arquivo de log geral onde serão armazenadas as conexões e consultas de programas cliente.
--log-isam[=arquivo]	Especifica o arquivo onde devem ser armazenadas as informações sobre mudanças no isam.
--log-long-format ^[3.22.7]	Armazena informações adicionais no arquivo de log de atualização (update).
--log-update[=path]	Especifica o arquivo onde serão registrados os textos das consultas que alteram o banco de dados.
--low-priority-updates ^[3.23]	Atribui prioridade menor a operações de atualização do banco de dados.
-n, --new	Utiliza recursos do MySQL que ainda não foram declarados estáveis.
-o, --old-protocol	Utiliza o protocolo para comunicação cliente/servidor das versões anteriores à MySQL 3.21.
--one-thread ^[3.22.2]	Utiliza somente um thread (para depuração no Linux).
--pid-file=path	Especifica o path do arquivo PID (process ID).
--safe-mode	Desabilita alguns sistemas de otimização do mysql (usado em testes).
-Sg, --skip-grant-tables	Não utiliza as tabelas de permissão para verificar conexões de clientes. Permite acesso total às tabelas a qualquer usuário.
--skip-host-cache	Não armazena os nomes dos computadores no cache interno.
--skip-locking	Desabilita o sistema de bloqueio (file system locking). Para utilizar o isamchk/myisamchk o servidor de banco de dados deve estar parado.
--skip-name-resolve	Não efetua resolução de nomes de hosts. Todos os nomes de hosts devem ser endereços IP ou "localhost".
--skip-networking	Não permite conexão remota TCP/IP. Somente conexões locais usando localhost são aceitas.
--skip-new	Não utiliza rotinas que foram implementadas a pouco tempo.
--skip-show-database ^[3.23]	Não permite que usuários sem permissão executem SHOW DATABASES ou SHOW TABLES em bancos de dados aos quais eles não tenham acesso.
--skip-thread-priority	Todos os threads do MySQL têm a mesma prioridade. Utilizado em sistemas nos quais o gerenciador de threads do MySQL não funciona perfeitamente.
-t path, --tmpdir=path ^[3.22.4]	Path para arquivos temporários.

Opções específicas para o ambiente Windows

Opção	Significado
--console ^[3.22.4]	Exibe uma janela para mensagens de erro.
--flush ^[3.22.9]	Grava as tabelas em disco após cada alteração. Embora reduza os riscos em caso de acidente, degrada o desempenho.
--install	Instala o mysqld -nt como um serviço no Windows NT.
--remove	Remove o mysqld -nt como um serviço do Windows NT.
--standalone	Inicia o mysqld -nt como um programa standalone em vez de serviço.

Váriaveis específicas do mysqld

Estas variáveis pode ser definidas com a opção **--set-variables**.

Variável	Significado
back_log	Número de conexões TCP que podem ser colocadas em fila (default=5).
connect_timeout	Número de segundos que o servidor mysqld espera por um pacote de conexão antes de responder com Bad handshake.
delayed_insert_limit	Número de linhas de comandos INSERT DELAYED que serão inseridos em uma tabela antes de verificar a existência de qualquer comando SELECT pendente. Se existir um comando SELECT, a operação de inserção será suspensa para permitir a execução do comando SELECT.
delayed_insert_timeout	Especifica quanto tempo um thread INSERT DELAYED deve esperar por comandos INSERT DELAYED antes terminar.
delayed_queue_size	Número de linhas que podem ser colocadas em "queue" para comandos INSERT DELAYED. Se a "queue" ficar cheia, os comandos INSERT DELAYED seguintes serão bloqueados até que exista espaço na "queue".
flush_time	Se esta opção estiver definida, todas as tabelas serão fechadas a cada flush_time segundos, para liberar recursos e sincronizar alterações para disco (default=30 minutos).

key_buffer_size	Tamanho do buffer usado para blocos de índices.
long_query_time	Número de segundos que definem uma consulta lenta; utilizado para definir quando incrementar o contador Slow-queries.
max_allowed_packet	Tamanho máximo do buffer usado para comunicação entre o servidor e o cliente.
max_connections	Número máximo permitido de conexões simultâneas de clientes.
max_connect_errors	Número máximo permitido de conexões falhas de um host antes do host ser bloqueado.
max_delayed_insert_threads	Número máximo de handlers que serão iniciados para manipular comandos INSERT DELAYED.
max_join_size	O otimizador MySQL estima quantas linhas um join retornará. Se o número estimado exceder max_join_size linhas, um erro será retornado.
max_sort_length	Valores BLOB ou TEXT são ordenados usando os primeiros max_sort_length bytes de cada valor.
net_buffer_length	Tamanho inicial do buffer usado para comunicação entre o servidor e o cliente.
record_buffer	Tamanho do buffer usado por threads que executam varreduras sequenciais de tabelas.
sort_buffer	Tamanho do buffer usado por threads que executam operações de ordenação (GROUP BY ou ORDER BY).
table_cache	Número máximo de tabelas que podem ser abertas.
tmp_table_size	Número máximo de bytes permitidos para tabelas temporárias. Se uma tabela exceder esse número, mysqld retornará um erro ao cliente.
thread_stack	Tamanho do stack para cada thread.
wait_timeout	Número de segundos que uma conexão pode permanecer inativa antes do servidor fechá-la.

isamchk e myisamchk

Utilizados para mostrar informações, otimizar e remover erros de tabelas.

isamchk [*opções*] *tabela*[.ISM] ...

Utilizado para tabelas no formato ISAM (.ISD e .ISM).

myisamchk [*opções*] *tabela*[.MYI] ...

Utilizado para tabelas no formato MyISAM (.MYD e .MYI).

Opções padrão usadas pelo isamchk e myisamchk

Opção	Significado
--debug	Opções de debugging.
--help	Exibe o texto de ajuda.
--set-variable	Atribui um valor a uma variável.
--silent	Exibe somente mensagens de erro.
--verbose	Exibe informações detalhadas sobre a execução do programa.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do isamchk e myisamchk

Opção	Significado
-a, --analyze	Analisa a distribuição das chaves e faz ajustes (se aplicáveis) para melhorar o desempenho das tabelas.
-b n, --block-search=n	Imprime o início da linha que contém o bloco <i>n</i> (somente para debugging).
-d, --description	Exibe informações sobre a tabela.
-e, --extend-check	Executa uma verificação exaustiva da integridade da tabela.
-f, --force	Força a verificação ou reparação da tabela mesmo que encontre um arquivo temporário para a tabela. Caso encontre algum erro, reiniciará o myisamchk com a opção --recover .
-i, --information	Mostra informações estatísticas sobre o conteúdo da tabela.
-k n, --keys-used=n	Utilizada com a opção --recover . Indica que o MySQL deve atualizar somente os primeiros <i>n</i> índices, fazendo com que as operações INSERT, DELETE e UPDATE sejam mais rápidas. Para desativar todos os índices utilize -k 0 .
-l, --no-symlinks	Não segue links simbólicos quando estiver reparando tabelas. O myisamchk normalmente segue links simbólicos (Unix).

-q, --quick	Usado com -recover para recuperação rápida. Recupera baseado no conteúdo do arquivo de índice sem utilizar o arquivo de dados.
-r, --recover	Recupera a maior parte dos problemas, exceto valores duplicados em índices que deveriam ser únicos.
-o, --safe-recover	Similar à opção --recover . Embora seja mais lento que --recover , pode recuperar mais problemas.
-S, --sort-index	Ordena os blocos de índices da tabela para otimizar a leitura sequencial de blocos.
-R <i>n</i>, --sort-records=<i>n</i>	Ordena os registros da tabela de acordo com o índice <i>n</i> . Utilize SHOW INDEX para ver a numeração dos índices.
-u, --unpack	Descompacta uma tabela previamente compactada com <code>myisampack</code> .
-w, --wait	Caso uma tabela esteja bloqueada, espera até que seja desbloqueada. Se esta opção não for especificada, o programa esperará 10 segundos antes de exibir uma mensagem de erro.

Opções específicas do `myisamchk`

Opção	Significado
-c, --check	Checa erros em tabelas (opção default).
-D <i>n</i>, --data-file-length=<i>n</i>	Tamanho máximo (em bytes) que um arquivo de dados recuperado poderá ter.
-F, --fast	Não checa tabelas que não foram modificadas desde a última checagem.
-m, --medium-check	Método mais rápido que <code>extended-check</code> , mas menos minucioso.
-T, --read-only	Não identifica a tabela como sido checada.
-A [<i>n</i>], --set-auto-increment[=<i>n</i>]	Força os valores de <code>AUTO_INCREMENT</code> iniciarem em <i>n</i> , ou em um valor maior que <i>n</i> se a tabela possuir registros com valores <code>AUTO_INCREMENT</code> maiores que <i>n</i> .
-t <i>path</i>, --tmpdir=<i>path</i>	Path do diretório usado para arquivos temporários.

Variáveis para `isamchk` e `myisamchk`

Variável	Significado
<code>key_buffer_size</code>	Tamanho do buffer usado para blocos de índices.
<code>read_buffer_size</code>	Tamanho do buffer de leitura.
<code>write_buffer_size</code>	Tamanho do buffer de gravação.
<code>sort_buffer_size</code>	Tamanho do buffer para ordenação de valores de chaves.
<code>sort_key_blocks</code>	Variável relacionada com a profundidade da estrutura B-tree. Não deve ser alterada.
<code>decode_bits</code>	Número de bits usados na descompactação de tabelas.

myisampack ^[3.23] e pack_isam

Utilitários para a criação de tabelas compactadas, somente para leitura. São fornecidos apenas para clientes que possuam mais de 10 licenças ou um contrato de suporte estendido.

myisampack [opções] *nomearq* ...

pack_isam [opções] *nomearq* ...

Parâmetro	Significado
<i>nomearq</i>	Cada <i>nomearq</i> deve ser o nome de um arquivo de índices ('.MYI'). Caso o diretório corrente não seja o mesmo da localização do arquivo .myi, deve-se digitar a localização completa do mesmo. A extensão .MYI pode ser omitida.

Opções padrão usadas pelo myisampack e pack_isam

Opção	Significado
--debug	Opções de debugging.
--help	Exibe o texto de ajuda.
--silent	Exibe somente mensagens de erro.
--verbose	Exibe informações detalhadas sobre a execução do programa.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do myisampack e pack_isam

Opção	Significado
-b, --backup	Efetua um backup de <i>nomearq</i> no arquivo <i>nomearq.OLD</i> , antes de compactar.
-f, --force	Com <i>—force</i> , myisampack compacta a tabela de qualquer jeito, mesmo que o resultado seja uma tabela de tamanho maior ou que exista o arquivo temporário (myisampack cria o arquivo temporário ' <i>nome_tabela.TMD</i> ' enquanto efetua a compressão da tabela). Normalmente o myisampack retorna uma mensagem de erro se encontra o arquivo temporário <i>nome_tabela.TMD</i> . O arquivo temporário pode não ser removido se o myisampack for interrompido no meio do processo de compactação.
-?, --help	Exibe o texto de ajuda do utilitário.

-j <i>big_tabela</i> , --join = <i>big_tabela</i>	Junta todas as tabelas especificadas na linha de comando em uma única tabela (<i>big_tabela</i>). Todas as tabelas a serem combinadas devem ter a mesma estrutura (mesmos nomes e tipos de colunas, mesmos índices, etc.).
-t , --test	Efetua apenas um teste de compactação, exibindo os resultados como se tivesse realmente compactado.
-T <i>path</i> , --tmp_dir = <i>path</i>	Especifica o diretório para arquivos temporários.
-w , --wait	Espera e tenta novamente se a tabela estiver em uso.

Opção específica para **pack_isam**

Opção	Significado
-p <i>n</i> , --packlength = <i>n</i>	Especifica o número de bytes (1 a 3) utilizados para armazenar o tamanho de cada registro na tabela compactada.

mysql

Permite interagir com o servidor de banco de dados enviando comandos e recebendo respostas. Pode ser usado no modo interativo, no qual o usuário digita os comandos, ou no modo não-interativo, quando recebe os comandos de um arquivo texto : `mysql banco_dados < script.sql > saida.txt`

`mysql [opções] [nome_bd]`

Opções padrão usadas pelo mysql

Opção	Significado
<code>--compress</code>	Utiliza compressão na comunicação com o servidor.
<code>--debug</code>	Opções de debugging.
<code>--help</code>	Exibe o texto de ajuda do programa.
<code>--host</code>	Endereço do host do servidor.
<code>--password</code>	Senha para conexão com o servidor.
<code>--pipe</code>	Usa "named pipe" para se conectar ao servidor.
<code>--port</code>	Número da porta para conexão com o servidor.
<code>--set-variable</code>	Atribui um valor a uma variável.
<code>--silent</code>	Exibe somente mensagens de erro.
<code>--socket</code>	Path completo para o arquivo socket.
<code>--user</code>	Nome do usuário.
<code>--verbose</code>	Exibe informações detalhadas sobre a execução do programa.
<code>--version</code>	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysql

Opção	Significado
<code>-B, --batch</code>	Imprime os resultados da consulta utilizando 'tab' como separador de colunas. Inclui uma linha inicial de cabeçalho, que pode ser suprimida com a opção <code>--skip-column-names</code> .
<code>-T, --debug-info</code>	Exibe informações de depuração quando o programa termina.
<code>-e "query", --execute="query"</code>	Executa a consulta e sai. Múltiplas consultas podem ser especificadas no string <i>query</i> , separando-as com ";").
<code>-f, --force</code>	Continua o processamento das consultas de um arquivo, mesmo que ocorra erro.
<code>-i, --ignore-space</code>	Ignora espaços entre os nomes de funções e o caractere "(" que introduz a lista de argumentos.
<code>-A, --no-auto-rehash</code>	Não efetua a complementação automática de nomes, tornando a inicialização do mysql mais rápida.
<code>-o, --one-database</code>	Atualiza apenas a tabela padrão na atualização de bancos de dados baseados no conteúdo do update log.

-q, --quick	Exibe cada linha do resultado retornado do servidor. Normalmente o resultado é exibido somente após ter sido totalmente retornado.
-r, --raw	Imprime os campos sem nenhuma conversão. Utilizado com --batch .
-N, --skip-column-names	Não exibe o nome das colunas na saída.
-L, --skip-line-numbers	Não exibe o número das linhas na saída.
-t, --table	Produz saída em formato tabular.
-n, --unbuffered	Limpa o buffer após cada comando.
-E, --vertical	Imprime os resultados na vertical.
-w, --wait	Espera e tenta novamente se não conseguir conectar-se com o servidor.

Variáveis de ambiente para mysql

Variável	Significado
max_allowed_packet	Tamanho máximo do buffer para comunicação entre o servidor e o cliente.
net_buffer_length	Tamanho inicial do buffer para comunicação entre o servidor e o cliente.

Comandos mysql

Comando	Descrição
clear, \c	Cancela a consulta corrente.
connect [<i>nomebd</i> [<i>host</i>]], \r [<i>nomebd</i> [<i>host</i>]]	Efetua a conexão com um banco de dados em um dado host.
edit, \e	Edita a consulta corrente.
ego, \G	Envia a consulta corrente para o servidor e exibe o resultado verticalmente.
exit	Sai do mysql.
go, \g, ;	Envia a consulta corrente para o servidor e exibe o resultado.
help, \h, ?	Exibe uma mensagem de ajuda descrevendo os comandos mysql.
print, \p	Imprime a consulta corrente.
quit, \q	Sai do mysql.
rehash, \#	Recalcula a informação necessária para a complementação automática de nomes de bancos de dados, tabelas e colunas.
status, \s	Exibe a informação de status do servidor.
use <i>nomebd</i> , \u <i>nomebd</i>	Identifica o banco de dados default.

Variáveis de ambiente do programa mysql

Variável	Significado
MYSQL_HISTFILE	Nome do arquivo usado para armazenar os comandos digitados na linha de comando (default=\$HOME/.mysql_history).
MYSQL_HOST	Nome do host utilizado para conectar-se ao servidor MySQL.

mysqlaccess

Mostra as permissões para a combinação de um usuário, endereço e banco de dados. O usuário e o banco de dados devem ser fornecidos.

mysqlaccess [*host* [*usuário* [*db*]]] *opções*

Opções padrão usadas pelo mysqlaccess

Opção	Significado
--host	Endereço do host do servidor.
--password	Senha para conexão com o servidor.
--user	Nome do usuário.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysqlaccess

Opção	Significado
-b, --brief	Exibe os resultados em linhas.
--commit	Copia as tabelas temporárias para as tabelas de permissões do MySQL.
--copy	Copia as tabelas originais de permissões para tabelas de permissões temporárias.
-d nomebd, --db=nomebd	Nome do banco de dados que será usado na verificação.
--debug=nível	Seleciona o nível de debug (0..3).
--howto	Exibe alguns exemplos de como usar o mysqlaccess.
--old_server	Conecta-se a um servidor antigo (3.21 ou anterior).
--plan	Imprime idéias sobre as futuras versões do mysqlaccess.
--preview	Exibe as diferenças entre as tabelas de permissões temporárias e as originais.
--relnotes	Exibe o histórico das versões do programa mysqlaccess.
-H nome, --rhost=nome	Endereço do servidor MySQL ao qual deve conectar-se.
--rollback	Desfaz as alterações feitas nas tabelas de permissões temporárias.
-P senha, --spassword=senha	Senha para se conectar ao servidor como superusuário.
-U nome, --superuser=nome	Especifica o nome de usuário do superusuário do MySQL.
-t, --table	Exibe os resultados em formato tabular.

mysqladmin

Executa operações administrativas, tais como criar e eliminar bancos de dados, recarregar as tabelas de permissões, gravar os dados das tabelas no disco e reabrir arquivos de log. Também pode ser utilizado para obter informações sobre o estado do servidor, versão e os processos correntes.

mysqladmin [*opções*] *comando* [*opções*] *comando*...

Opções padrão usadas pelo mysqladmin

Opção	Significado
--compress	Utiliza compressão no protocolo de comunicação com o servidor.
--debug	Opções de debugging.
--help	Exibe o texto de ajuda do programa.
--host	Endereço do host do servidor.
--password	Senha para conexão com o servidor.
--pipe	Usa "named pipe" para conectar-se ao servidor.
--port	Número da porta para conexão com o servidor.
--silent	Exibe somente mensagens de erro.
--socket	Path completo para o arquivo socket.
--user	Nome do usuário.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysqladmin

Opção	Significado
-f, --force	Não pede confirmação para remover uma tabela do banco de dados. Continua o processamento mesmo que ocorra um erro.
--relative	Exibe a diferença entre os valores atuais e anteriores quando usada com a opção -sleep . Disponível somente com o comando extended-status .
-i n, --sleep=n	Faz com que os comandos sejam executados com um tempo de espera entre os mesmos.
-t n, --timeout=n	Tempo máximo para conseguir a conexão com o servidor.
-w [n], --wait[=n]	Espera e tenta novamente caso não consiga conectar-se com o servidor.

Comandos mysqladmin

Comando	Descrição
create <i>nomebd</i>	Cria um novo banco de dados. Similar ao comando SQL CREATE DATABASE <i>nomebd</i> .
drop <i>nomebd</i>	Elimina um banco de dados e todas as suas tabelas. Similar ao comando SQL DROP DATABASE <i>nomebd</i> .
extended-status	Mostra o status do servidor, variáveis, uso dos índices, número de consultas ao banco de dados, etc. Similar ao comando SQL SHOW STATUS.
flush-hosts	Limpa a tabela de cache que contém os endereços dos servidores. Usado quando o endereço IP de um dos servidores é alterado ou quando aparece a mensagem de erro "Host... is blocked". Similar ao comando SQL FLUSH HOSTS.
flush-logs	Fecha e reabre todos os arquivos padrão de log. Similar ao comando SQL FLUSH LOGS.
flush-privileges	Recarrega os privilégios das tabelas grant no banco de dados mysql. Deve ser usado quando tiver sido feita alguma alteração nas tabelas de permissões dos usuários (insert, update). Similar ao comando SQL FLUSH PRIVILEGES.
flush-status	Limpa as variáveis de status. Similar ao comando SQL FLUSH STATUS.
flush-tables	Grava todos os dados das tabelas abertas em memória para o disco rígido. Similar ao comando SQL FLUSH TABLES.
kill <i>id,id,...</i>	Elimina threads do mysql. Similar ao comando SQL KILL <i>id</i> .
password <i>senha</i> ^[3.22]	Altera a senha. Similar ao comando SQL SET PASSWORD <i>senha</i> .
ping	Verifica se o servidor mysqld está ativo.
processlist	Exibe a lista dos threads ativos no servidor. Similar ao comando SQL SHOW PROCESSLIST.
refresh	Grava todos os dados das tabelas abertas em memória para o disco rígido e abre os arquivos de log.
reload	Recarrega as tabelas grant. Similar ao comando SQL FLUSH PRIVILEGES.
shutdown	Termina a execução do servidor.
status	Mostra uma pequena mensagem com o status do servidor, tempo que está ativo, número de consultas ao banco de dados, etc.
variables	Exibe as variáveis disponíveis. Similar ao comando SQL SHOW VARIABLES.
version	Exibe a versão do servidor.

mysqlbug

Script utilizado para preencher um relatório de erros e problemas encontrados em programas ou utilitários do MySQL. Coleta informações detalhadas sobre o seu MySQL e sistema operacional e envia um relatório dos problemas à lista de bugs do MySQL, que será analisada pela equipe do MySQL. Os erros serão corrigidos ou documentados na próxima atualização do MySQL.

Antes de usar o mysqlbug para reportar um erro, tente usar a versão mais nova do MySQL (pois o problema pode ter sido corrigido) e tenha em mente que o problema encontrado deve ser facilmente reproduzido com o script gerado pelo mysqlbug, caso contrário será difícil reproduzir o erro em outros ambientes.

mysqlbug [*endereço*]

Parâmetro	Significado
------------------	--------------------

<i>endereço</i>	Endereço de e-mail para onde deve ser enviado o relatório de bug. Se não especificado, o relatório será enviado à lista do MySQL.
-----------------	---

mysqldump

Efetua dump da estrutura e dos dados do bancos de dados e tabelas MySQL. O resultado do mysqldump pode ser escrito em um arquivo de texto contendo os comandos SQL que permitem criar a tabela e/ou inserir os dados, ou em um arquivo contendo dados separados por tab.

mysqldump [*opções*] *nomebd* [*tabelas*]

Parâmetros	Significado
<i>nomebd</i>	Nome do banco de dados.
<i>tabelas</i>	Nome das tabelas a serem descarregadas. Se não for especificada nenhuma tabela, o banco de dados inteiro será descarregado (dumped).

Opções padrão usadas pelo mysqldump

Opção	Significado
--compress	Utiliza compressão no protocolo de comunicação com o servidor.
--debug	Opções de debugging.
--help	Exibe o texto de ajuda do programa.
--host	Endereço do host do servidor.
--password	Senha para conexão com o servidor.
--pipe	Usa "named pipe" para conectar-se ao servidor.
--port	Número da porta para conexão com o servidor.
--set-variable	Atribui um valor a uma variável.
--socket	Path completo para o arquivo socket.
--user	Nome do usuário.
--verbose	Exibe informações detalhadas sobre a execução do programa.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysqldump

Opção	Significado
--add-drop-table	Inclui um comando DROP TABLE antes de cada comando CREATE TABLE.
--add-locks	Adiciona o comando LOCK TABLES antes e o comando UNLOCK TABLE após cada dump de tabela. Assim, nenhum outro cliente poderá acessar as tabelas e as operações de insert serão mais rápidas.
--allow-keywords	Permite a criação de nomes de colunas que são palavras-chave. Cada nome de coluna terá o nome da tabela como prefixo.
-c, --complete-insert	Utiliza comandos INSERT completos (com nomes de colunas).
--delayed-insert	Insere linhas com o comando INSERT DELAYED.
-e, --extended-insert	Usa o formato de INSERT que permite que múltiplas linhas sejam especificadas em um único comando INSERT. Sintaxe: INSERT INTO tabela VALUES(...), (...).
-F, --flush-logs	Escreve todos os logs do servidor de MySQL em disco antes de iniciar a operação.
-f, --force	Continua o processamento mesmo que ocorra um erro durante o dump da tabela.
--full	Inclui informações adicionais, tais como o tipo da tabela e os comandos CREATE TABLE gerados pelo mysqldump.
-l, --lock-tables	Bloqueia todas as tabelas antes de iniciar o dump.
-t, --no-create-info	Não utiliza o comando CREATE TABLE antes de iniciar as operações de INSERT.
-d, --no-data	Não utiliza o comando INSERT, apenas mostra a estrutura da tabela (CREATE TABLE por exemplo).
--opt	O mesmo que <code>--quick --add-drop-table --add-locks --extended-insert --lock-tables</code> . Deve criar um arquivo de dump, que é o mais rápido, para ser inserido no MySQL.
-q, --quick	Exibe os dados imediatamente, sem colocar no buffer. Se não for especificada, o MySQL carregará todos os dados da tabela na memória antes de mostrar os resultados, o que poderá acarretar problemas de falta de memória se a tabela for muito grande.

Programas Cliente

- T** *path*, **--tab=***path* Cria um arquivo `table_name.sql` file, que contém os comandos CREATE, e um arquivo `table_name.txt` file, que contém os dados, para cada tabela.
- w** "*condição*", **--where=**"*condição*" ^[3.22.7] Retorna somente os registros selecionados pela condição especificada. Deve estar entre aspas.

Opções de formato de dados

Estas opções são usadas com a opção **-T** e possuem o mesmo significado das cláusulas correspondentes do comando LOAD DATA INFILE.

Opção	Significado
--fields-enclosed-by= <i>char</i>	Especifica o caractere utilizado para delimitar string de caracteres.
--fields-escaped-by= <i>char</i>	Especifica o caractere utilizado para indicar que o próximo caractere não é um caractere especial.
--fields-optionally-enclosed-by= <i>char</i>	Especifica o caractere utilizado para delimitar string de caracteres.
--fields-terminated-by= <i>char</i>	Especifica o caractere ou caracteres utilizados para delimitar valores na linha.
--lines-terminated-by= <i>string</i>	Especifica o caractere ou caracteres que indicam o final de uma linha.

Variáveis de ambiente para mysqldump

Variável	Significado
max_allowed_packet	Tamanho máximo do buffer para comunicação entre o servidor e o cliente.
net_buffer_length	Tamanho inicial do buffer para comunicação entre o servidor e o cliente.

mysqlimport

Importa dados de arquivos de texto, e insere-os nas respectivas tabelas usando `LOAD DATA INFILE`. Fornece um interface tipo linha de comando para o comando `LOAD DATA INFILE SQL`. Muitas opções do `mysqlimport` correspondem diretamente a opções do comando `LOAD DATA INFILE`.

Para cada arquivo de texto na linha de comando, `mysqlimport` corta qualquer extensão do nome do arquivo e usa o resultado para determinar para qual tabela deve ser copiado o conteúdo dos arquivos. Por exemplo, arquivos com os nomes 'patient.txt', 'patient.text' e 'patient' deverão ser importados para a tabela `patient`.

mysqlimport [*opções*] *nomebd nomearq ...*

Opções padrão usadas pelo mysqlimport

Opção	Significado
--compress	Utiliza compressão na comunicação com o servidor.
--debug	Opções de debugging.
--help	Exibe o texto de ajuda do programa.
--host	Endereço do host do servidor.
--password	Senha para conexão com o servidor.
--pipe	Usa "named pipe" para conectar-se ao servidor.
--port	Número da porta para conexão com o servidor.
--silent	Exibe somente mensagens de erro.
--socket	Path completo para o arquivo socket.
--user	Nome do usuário.
--verbose	Exibe informações detalhadas sobre a execução do programa.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysqlimport

Opção	Significado
-d, --delete	Elimina todos os dados atuais da tabela antes de inserir os dados importados.
-f, --force	Continua o processamento mesmo que ocorra algum erro.
-i, --ignore	Veja a descrição da opção --replace .
-L, --local	Faz a leitura dos arquivos da máquina cliente. Normalmente os arquivos são lidos da máquina na qual o servidor está instalado, mas existem algumas situações nas quais o cliente <code>mysqlimport</code> está sendo executado em outra máquina e os arquivos estão na máquina do cliente.

Programas Cliente

-l, --lock-tables	Bloqueia todas as tabelas para leitura ou gravação antes de importar qualquer arquivo, fazendo com que todos os valores sejam importados antes que qualquer comando, como inserts e updates, altere algum registro.
-r, --replace	As opções --replace e --ignore indicam para o mysql qual procedimento deve ser adotado no caso dos dados da chave primária, a serem importados, já existirem no banco de dados. A opção --replace faz com que o dados antigos sejam removidos e os novos inseridos. A opção --ignore faz com que o mysql ignore os dados. Caso nenhuma das opções tenha sido escolhida, o mysqlimport mostrará um erro e ignorará o resto do arquivo.

Opções de formato de dados

Estas opções são usadas com a opção **-T** e possuem o mesmo significado das cláusulas correspondentes do comando **LOAD DATA INFILE**.

Opção	Significado
--fields-enclosed-by=char	Especifica o caractere utilizado para delimitar string de caracteres.
--fields-escaped-by=char	Especifica o caractere utilizado para indicar que o próximo caractere não é um caractere especial.
--fields-optionally-enclosed-by=char	Especifica o caractere utilizado para delimitar string de caracteres.
--fields-terminated-by=char	Especifica o caractere ou caracteres utilizados para delimitar valores na linha.
--lines-terminated-by=string	Especifica o caractere ou caracteres que indicam o final de uma linha.

mysqlshow

Exibe informações sobre bancos de dados, tabelas, colunas e índices. Sem nenhum parâmetro, lista todos os bancos de dados.

mysqlshow [opções] [*nomebd* [*tabela* [*campo*]]]

Opção	Significado
<i>nomebd</i>	Exibe informações sobre o banco de dados especificado.
<i>tabela</i>	Exibe informações sobre a tabela especificada.
<i>campo</i>	Exibe informações sobre o campo especificado.

Opções padrão usadas pelo mysqlshow

Opção	Significado
--compress	Utiliza compressão no protocolo de comunicação com o servidor.
--debug	Opções de debugging.
--help	Exibe o texto de ajuda do programa.
--host	Endereço do host do servidor.
--password	Senha para conexão com o servidor.
--pipe	Usa "named pipe" para conectar-se ao servidor.
--port	Número da porta para conexão com o servidor.
--socket	Path completo para o arquivo socket.
--user	Nome do usuário.
--version	Exibe a versão do programa.

Veja a descrição completa dessas opções na página 48.

Opções específicas do mysqlshow

Opção	Significado
-i, --status ^[3.23]	Exibe informações sobre as tabelas em um banco de dados. Similar ao comando SQL SHOW TABLES STATUS.
-k, --keys	Exibe as chaves de uma tabela.

Tipos de Colunas

As seguintes convenções são utilizadas na descrição dos tipos de colunas:

Convenção	Significado
<i>M</i>	Representa o número de caracteres usados para exibir os valores da coluna (1 a 255 caracteres).
<i>D</i>	Número de casas decimais em campos no formato ponto flutuante (0 a 30). Não pode ser maior que <i>M</i> -2; caso contrário, o valor de <i>M</i> será ajustado para <i>D</i> +2.
<i>L</i>	Indica o tamanho da informação armazenada no campo STRING.
NATIONAL <small>[3.23.5]</small>	Especificação ANSI SQL que define o grupo de caracteres padrão a ser utilizado.
UNSIGNED	Modificador que faz com que uma coluna aceite apenas os números positivos (sem sinal). Somente para tipos Integer.
ZEROFILL	Adiciona zeros de preenchimento à esquerda de cada número. Por exemplo: um campo INT(5) ZEROFILL exibirá o número 235 como 00235. Observe que se for especificado ZEROFILL para uma coluna, MySQL automaticamente adicionará o atributo UNSIGNED à coluna.
AUTO_INCREMENT	Tipo especial de coluna numérica que pode ser atualizada automaticamente. Quando um valor NULL (ou 0) for inserido na coluna, o MySQL o substituirá automaticamente pelo próximo número na sequência da coluna.

Tipos de Dados do MySQL

Numéricos

TINYINT[(M)]

Um inteiro de tamanho muito pequeno.

Atributos: AUTO_INCREMENT, UNSIGNED, ZEROFILL.

Intervalo: Entre -128 e 127 com sinal,
ou entre 0 e 255 sem sinal.

Espaço req: 1 byte.

SMALLINT[(M)]

Um inteiro pequeno.

Atributos: AUTO_INCREMENT, UNSIGNED, ZEROFILL.

Intervalo: Entre -32768 e 32767 com sinal,
ou entre 0 e 65535 sem sinal.

Espaço req: 2 bytes.

MEDIUMINT[(M)]

Um inteiro de tamanho médio.

Atributos: AUTO_INCREMENT, UNSIGNED, ZEROFILL.

Intervalo: Entre -8388608 e 8388607 com sinal,
ou entre 0 e 16777215 sem sinal.

Espaço req: 3 bytes

INT[(M)]

Um inteiro de tamanho normal.

Atributos: AUTO_INCREMENT, UNSIGNED, ZEROFILL.

Intervalo: Entre -2147483648 e 2147483647 com sinal, ou
entre 0 e 4294967295 sem sinal.

Espaço req: 4 bytes.

INTEGER[(M)]

Sinônimo de INT.

BIGINT[(M)]

Um inteiro grande.

Atributos: AUTO_INCREMENT, UNSIGNED, ZEROFILL.

Intervalo: Entre -9223372036854775808 e
9223372036854775807 com sinal, ou entre 0 e
18446744073709551615 sem sinal.

Espaço req: 8 bytes.

FLOAT[(M,D)]

Um número de ponto-flutuante com precisão simples.

Atributos: ZEROFILL.

Intervalo: Entre -3.402823466E+38 e -1.175494351E-38, 0
(zero) e entre 1.175494351E-38 e
3.402823466E+38.

Espaço req: 4 bytes

DOUBLE[(M,D)]

Um número de ponto-flutuante com precisão dupla.

Atributos: ZEROFILL.

Intervalo: Entre -1.7976931348623157E+308 e -
2.2250738585072014E-308, 0 (zero) e entre
2.2250738585072014E-308 e
1.7976931348623157E+308.

Espaço req: 8 bytes.

DOUBLE PRECISION[(M,D)]

Sinônimo de DOUBLE.

REAL[(M,D)]

Sinônimo de DOUBLE.

DECIMAL[(M[,D])]

Um número de ponto-flutuante armazenado com um string (1 byte por dígito, ponto decimal, ou sinal "-").

Atributos: ZEROFILL

Intervalo: O intervalo máximo é o mesmo de DOUBLE.

Espaço req: M bytes para versões anteriores à 3.23 e M+2 bytes para 3.23 e mais recentes.

NUMERIC(M,D)

Sinônimo de DECIMAL.

Data e Hora

DATE

Uma data no formato "YYYY-MM-DD".

Intervalo: Entre '1000-01-01' e '9999-12-31'.

Espaço req: 3 bytes.

DATETIME

Uma combinação de data e hora no formato "YYYY-MM-DD HH:MM:SS".

Espaço req: 8 bytes.

Intervalo: Entre "1000-01-01 00:00:00" e "9999-12-31 23:59:59".

TIMESTAMP[(M)]

Um timestamp (combinação da data e hora corrente) no formato YYYYMMDDhhmmss.

Intervalo: Entre 19700101000000 e algum dia do ano 2037.

Espaço req: 4 bytes.

Valor M	Formato
TIMESTAMP(14)	YYYYMMDDHHMMSS
TIMESTAMP(12)	YYMMDDHHMMSS
TIMESTAMP(10)	YYMMDDHHMM
TIMESTAMP(8)	YYYYMMDD
TIMESTAMP(6)	YYMMDD
TIMESTAMP(4)	YYMM
TIMESTAMP(2)	YY

TIME

Um horário no formato "HH:MM:SS".

Intervalo: Entre '-838:59:59' e '838:59:59'.

Espaço req: 3 bytes.

YEAR ^[3.22]

Um ano no formato YYYY.

Intervalo: 1900 a 2155.

Espaço req: 1 byte.

String

Os tipos CHAR, VARCHAR e TEXT não distinguem minúsculas de maiúsculas em comparações, exceto se o atributo BINARY for especificado. Os tipos BLOB distinguem minúsculas de maiúsculas.

CHAR(M)

Um string de caracteres de tamanho fixo (0 a 255 caracteres).

Atributos: BINARY.

Espaço req: M bytes.

VARCHAR(M)

Um string de caracteres de tamanho variável (0 a 255 caracteres).

Atributos: BINARY.

Espaço req: Tamanho do valor + 1 byte para registrar o tamanho.

TINYBLOB

Um valor BLOB pequeno (0 a 255 bytes).

Espaço req: Tamanho do valor + 1 byte para registrar o tamanho.

BLOB

Um valor BLOB de tamanho normal (0 a 65535 bytes).

Espaço req: Tamanho do valor + 2 bytes para registrar o tamanho.

MEDIUMBLOB

Um valor BLOB de tamanho médio (0 a 16777215 bytes).

Espaço req: Tamanho do valor + 3 bytes para registrar o tamanho.

Tipos de Dados

LONGBLOB

Um valor BLOB grande (0 a 4294967295 bytes).

Espaço req: Tamanho do valor + 4 bytes para registrar o tamanho.

TINYTEXT

Um valor TEXT pequeno (0 a 255 bytes).

Espaço req: Tamanho do valor + 1 byte para registrar o tamanho.

TEXT

Um valor TEXT de tamanho normal (0 a 65535 bytes).

Espaço req: Tamanho do valor + 21 bytes para registrar o tamanho.

MEDIUMTEXT

Um valor TEXT de tamanho médio (0 a 16777215 bytes).

Espaço req: Tamanho do valor + 3 bytes para registrar o tamanho.

LONGTEXT

Um valor TEXT grande (0 a 4294967295 bytes).

Espaço req: Tamanho do valor + 4 bytes para registrar o tamanho.

ENUM('valor1','valor2',...)

Uma enumeração. Permite especificar uma lista de valores, dos quais um pode ser atribuído à coluna.

Espaço req: 1 byte para enum de 1 a 255 valores;
2 bytes para enum de 256 a 65535 valores.

SET('valor1','valor2',...)

Um grupo. Permite especificar uma lista de valores dos quais um ou mais podem ser atribuídos a uma coluna.

Espaço req: 1 byte para setS de 1 a 8 membros;
2 bytes para setS de 9 a 16 membros;
3 bytes para setS de 17 a 24 membros;
4 bytes para setS de 25 a 32 membros;
8 bytes para setS de 33 a 64 membros.

Literais

Strings

Um string é uma sequência de caracteres delimitados por apóstrofe ou aspas. Exemplos:

'um string'

“outro string”

Dentro de um string, certas seqüências têm um significado especial. Cada uma dessas seqüências começa com uma barra invertida (\) conhecida como caractere escape.

Caractere	Significado
\0	Um caractere ASCII 0 (NUL).
\n	Um caractere Newline.
\t	Um caractere Tab.
\r	Um caractere Carriage Return.
\b	Um caractere Backspace.
\'	Um caractere apóstrofe.
\"	Um caractere aspas.
\\	Um caractere barra invertida ('\').
\%	Um caractere '%'. O caractere '%' é utilizado na cláusula WHERE.
_	Um caractere '_'. O caractere '_' é utilizado na cláusula WHERE.

Existem várias maneiras de incluir aspas dentro de um string:

- Um caractere apóstrofe dentro de um string entre apóstrofes pode ser escrito como ". Exemplo (1).
- Um caractere aspas dentro de um string entre aspas pode ser escrito como “. Exemplo (2).
- Preceder o caractere aspas com um caractere escape ('\'). Exemplo (3).
- Um caractere apóstrofe dentro de um string entre aspas ou um caractere aspas dentro de um string entre apóstrofes não precisam de tratamento diferenciado.

Na inserção de dados binários em uma coluna BLOB, os seguintes caracteres devem ser representados por seqüências escape:

Caractere	Significado
NUL	ASCII 0. Inserido através de '\0'.
\	ASCII 92. Barra invertida, inserido através de '\\'.
'	ASCII 39. Apóstrofe, inserido através de '\'
“	ASCII 34. Aspas, inserido através de '\".

Números

Inteiros são representados como uma sequência de dígitos. Floats usam '.' como separador decimal.

Exemplos de números inteiros válidos:

1560

0

-11

Exemplos de números de ponto-flutuante válidos:

15.34

-1203.7381e+10

142.00

Um inteiro pode ser utilizado em um contexto de ponto-flutuante, mas será interpretado com um número de ponto-flutuante.

Valores Hexadecimais

O MySQL suporta valores hexadecimais. Em contexto de números funciona como um integer (precisão de 64 bits). Em contexto de strings funciona como um string binário, onde cada par de dígito hexa é convertido para um caractere.

```
mysql> SELECT 0xb+0
```

```
> 11
```

```
mysql> select 0x64656667;
```

```
> defg
```

Valores NULL

O valor NULL significa “sem dados” e é diferente de valores tais como: 0 para tipos numéricos ou string vazio para tipos string.

NULL pode ser representado por \N em formatos de arquivo texto (LOAD DATA INFILE, SELECT ... INTO OUTFILE).

Controle de Acesso a Banco de dados

Durante a instalação do MySQL é criado um banco de dados, chamado “mysql”, contendo cinco tabelas: db, host, user, tables_priv e columns_priv, que implementam o controle de acesso aos dados no MySQL.

Tabelas

Tabela	Função
user	Determina se o servidor deve aceitar ou rejeitar a conexão baseada no par endereço/usuário. Se aceitar, determina os privilégios que o par endereço/usuário possui em todos os bancos de dados.
db	Controla o acesso aos bancos de dados baseada no par endereço/usuário. Determina os privilégios que o par endereço/usuário possui sobre os bancos de dados.
host	Controla o acesso aos bancos de dados. Determina os privilégios que cada endereço possui sobre o banco de dados. A tabela será utilizada como extensão da tabela db no momento que se desejar que um registro da tabela db seja válido para diversos registros.
tables_priv	Funciona de modo parecido ao da tabela db, porém controla o acesso às tabelas.
columns_priv	Funciona de modo parecido ao da tabela db, porém controla o acesso às colunas das tabelas.

Privilégios

alter

Permite executar o comando ALTER TABLE. Caso seja dado o privilégio ALTER para a base de dados mysql, o usuário poderá renomear qualquer tabela de privilégios.

Coluna: Alter_priv.

Contexto: tabelas

create

Permite criar novas tabelas e bancos de dados.

Coluna: Create_priv.

Contexto: bases de dados, tabelas ou índices.

delete

Permite remover registros das tabelas.

Coluna: Delete_priv.

Contexto: tabelas.

drop

Permite remover tabelas e base de dados. Caso seja dado o privilégio DROP para a base de dados mysql, o usuário poderá apagar qualquer tabela de privilégios.

Coluna: Drop_priv.

Contexto: base de dados e tabelas.

file

Permite ler ou gravar arquivos no servidor usando instruções LOAD DATA INFILE e SELECTINTO OUTFILE. Tome cuidado para não dar este privilégio a qualquer usuário, pois ele pode ler arquivos importantes do sistema. O usuário que recebe este privilégio pode ler ou escrever qualquer arquivo que o servidor MySQL possar ler ou escrever. Por exemplo, o usuário pode ler o arquivo de senhas do Unix (passwd) com o comando LOAD DATA INFILE, inserir o conteúdo do arquivo em uma tabela e depois recuperar os dados com um comando SELECT.

Coluna: File_priv.

Contexto: Arquivos que estão no sistema do MySQL.

grant

Permite que o usuário habilite os seus privilégios para outros usuários. Dois usuários que tenham o privilégio grant podem combinar seus privilégios.

Coluna: Grant_priv.

Contexto: base de dados ou tabelas.

index

Permite criar e remover índices.

Coluna: Index_priv.

Contexto: tabelas.

insert

Permite inserir registros nas tabelas.

Coluna: Insert_priv.

Contexto: tabelas.

process

Permite executar os seguintes comandos do utilitário mysqladmin:

Comando	Significado
processlist	Mostra os threads que estão sendo executados no servidor.
kill	Elimina threads do servidor. Um usuário sempre pode eliminar os seus próprios threads, mas necessita do privilégio process para ver e eliminar processos de outros usuários. Thread pode ser vagarosamente associado a um comando que o usuário executa. Por exemplo, um comando SELECT que demora muito para retornar pode ser eliminado vendo o seu ID com o comando processlist e depois eliminado com kill.

Coluna: Process_priv.

Contexto: administração do servidor.

reload

Permite executar os seguintes comandos mysqladmin: reload, refresh, flush-privileges, flush-hosts, flush-logs e flush-tables. Para mais informações veja o comando FLUSH na página 12.

Coluna: Reload_priv.

Contexto: administração do servidor.

select

Permite executar comandos SELECT.

Coluna: Select_priv.

Contexto: tabelas.

shutdown

Permite executar o comando shutdown do utilitário mysqladmin, isto é, parar a execução do servidor.

Coluna: Shutdown_priv.

Contexto: administração do servidor.

update

Permite alterar linhas em tabelas.

Coluna: Update_priv.

Contexto: tabelas.

Nomes de Identificadores

Nomes de bancos de dados, tabelas, índices, colunas e alias seguem as mesmas regras no **MySQL**.

Identificador	Tam. máx	Caracteres permitidos
Banco de dados	64	Qualquer caractere permitido em um nome de diretório, exceto /.
Tabela	64	Qualquer caractere permitido em um nome de arquivo, exceto / ou .
Coluna	64	Todos os caracteres
Alias	255	Todos os caracteres

As regras para nomes são as seguintes:

- Um nome pode conter caracteres alfanuméricos do character set corrente e também os caracteres “_” e “\$”. O character set default é ISO-8859-1 (Latin1), que pode ser alterado recompilando o **MySQL**.
- Um nome pode iniciar com qualquer caractere válido, inclusive um número, mas não pode conter somente números.
- O caractere “.” (ponto) não pode ser utilizado dentro do nome pois é utilizado nas referências de colunas, que podem ser feitas das seguintes formas:

Referência	Significado
<i>coluna</i>	Nome da coluna que é utilizada dentro de um comando select, update, etc.
' <i>coluna</i> '	Nome de uma coluna delimitado com apóstrofe indica que o nome da coluna é uma palavra reservada ou contém caracteres especiais.
<i>tabela.coluna</i>	Nome da coluna da tabela especificada do banco de dados corrente.
<i>bd.tabela.coluna</i> ^[3.22]	Nome da coluna da tabela e banco de dados especificados.

Uso de maiúsculas e minúsculas

No **MySQL**, bancos de dados e tabelas correspondem a diretórios e arquivos dentro desses diretórios. Em razão disso, no Unix é feita distinção entre minúsculas e maiúsculas nos nomes de tabelas e bancos de dados e no Win32 não é feita distinção.

Nos nomes de colunas não é feita distinção em nenhum dos casos.

Palavras Reservadas do MySQL

action	full	outfile
add	function	pack_keys
aggregate	global	partial
all	grant	password
alter	grants	precision
after	group	primary
and	having	procedure
as	heap	process
asc	high_priority	processlist
avg	hour	privileges
avg_row_length	hour_minute	read
auto_increment	hour_second	real
between	hosts	references
bigint	identified	reload
bit	ignore	regexp
binary	in	rename
blob	index	replace
bool	infile	restrict
both	inner	returns
by	insert	revoke
cascade	insert_id	rlike
case	int	row
char	integer	rows
character	interval	second
change	int1	select
check	int2	set
checksum	int3	show
column	int4	shutdown
columns	int8	smallint
comment	into	soname
constraint	if	sql_big_tables
create	is	sql_big_selects
cross	isam	sql_low_priority_updates
current_date	join	sql_log_off
current_time	key	sql_log_update
current_timestamp	keys	sql_select_limit
data	kill	sql_small_result
database	last_insert_id	sql_big_result
databases	leading	sql_warnings
date	left	straight_join
datetime	length	starting
day	like	status
day_hour	lines	string
day_minute	limit	table
day_second	load	tables
dayofmonth	local	temporary
dayofweek	lock	terminated
dayofyear	logs	text
dec	long	then
decimal	longblob	time
default	longtext	timestamp
delayed	low_priority	tinyblob
delay_key_write	max	tinytext
delete	max_rows	tinyint
desc	match	trailing
describe	mediumblob	to
distinct	mediumtext	type
distinctrow	mediumint	use
double	middleint	using
drop	min_rows	unique
end	minute	unlock
else	minute_second	unsigned
escape	modify	update
escaped	month	usage
enclosed	monthname	values
enum	myisam	varchar
explain	natural	variables
exists	numeric	varying
fields	no	varbinary
file	not	with
first	null	write
float	on	when
float4	optimize	where
float8	option	year
flush	optionally	year_month
foreign	or	zerofill
from	order	
for	outer	

Funções PHP relacionadas ao MySQL

O PHP deve ser compilado no Unix com a diretiva `--with-mysql=DIR` para que o MySQL possa ser utilizado. Uma maneira de verificar se o PHP foi instalado com suporte ao MySQL é através da função `phpinfo()`. A lista de funções abaixo foi extraída do guia de consulta rápida PHP de Herbert G. Fischer, desta mesma Editora.

mysql_affected_rows

Retorna o número de registros afetados em uma operação MySQL anterior.

```
int mysql_affected_rows(int [id_link] );
```

mysql_change_user

Altera o usuário ativo no momento. Requer PHP 3.0.13 e MySQL 3.23.3.

```
int mysql_change_user(string usuário, string senha,  
string [database] , int [link_identifier] );
```

mysql_close

Fecha uma conexão MySQL. Retornará True se tiver sucesso; caso contrário, retornará False.

```
int mysql_close(int [id_link] );
```

mysql_connect

Abre uma conexão com um servidor MySQL. Retornará *id_link* se tiver sucesso; caso contrário, retornará False.

```
int mysql_connect(string [hostname[:port]][:/path/to/  
socket]], string [usuário], string [senha]);
```

mysql_create_db

Cria um banco de dados MySQL.

```
int mysql_create_db(string nomebd, int [id_link] );
```

mysql_data_seek

Move o ponteiro interno de resultados. Retornará True se tiver sucesso; caso contrário, retornará False.

```
int mysql_data_seek(int id_resultado, int num_linha);
```

mysql_db_query

Envia uma consulta MySQL. Retornará *id_resultado* se tiver sucesso; caso contrário, retornará False.

```
int mysql_db_query(string nomebd, string consulta,  
int [id_link] );
```

mysql_drop_db

Elimina um banco de dados MySQL. Retornará True se tiver sucesso; caso contrário, retornará False.

int mysql_drop_db(string *nomebd*, int [*id_link*]);

mysql_errno

Retorna o erro de uma operação MySQL anterior.

int mysql_errno(int [*id_link*]);

mysql_error

Retorna a mensagem de erro de uma operação MySQL anterior.

string mysql_error(int [*id_link*]);

mysql_fetch_array

Retorna um array que corresponde ao próximo registro da consulta ou False se não houver mais registros.

array mysql_fetch_array(int *resultado*, int [*tipo_resultado*]);

mysql_fetch_field

Retorna um objeto com informações dos campos.

object mysql_fetch_field(int *resultado*, int [*offset_campo*]);

mysql_fetch_lengths

Retorna, em um array, o comprimento de cada campo dos resultados.

array mysql_fetch_lengths(int *resultado*);

mysql_fetch_object

Lê um registro de resultado e coloca em um objeto.

object mysql_fetch_object(int *resultado*, int [*tipo_resultado*]);

mysql_fetch_row

Lê um registro de resultado e coloca em um array enumerado.

array mysql_fetch_row(int *resultado*);

mysql_field_flags

Retorna as flags associadas ao campo especificado em um resultado.

```
string mysql_field_flags(int resultado, int offset_campo);
```

mysql_field_len

Retorna o tamanho do campo especificado.

```
int mysql_field_len(int resultado, int offset_campo)
```

mysql_field_name

Retorna o nome do campo especificado em um resultado.

```
string mysql_field_name(int resultado, int indice_campo);
```

mysql_field_seek

Muda o ponteiro de resultados para um campo especificado.

```
int mysql_field_seek(int resultado, int offset_campo);
```

mysql_field_table

Retorna o nome da tabela de um campo especificado.

```
string mysql_field_table(int resultado, int offset_campo);
```

mysql_field_type

Retorna o tipo do campo especificado em um resultado.

```
string mysql_field_type(int resultado, int offset_campo);
```

mysql_free_result

Libera a memória de resultados.

```
int mysql_free_result(int resultado);
```

mysql_insert_id

Retorna o id gerado por uma operação INSERT anterior.

```
int mysql_insert_id(int [id_link] );
```

mysql_list_fields

Lista os campos de resultado MySQL.

```
int mysql_list_fields(string nomebd, string nome_tabela, int [id_link] );
```

mysql_list_dbs

Lista os bancos de dados disponíveis em um servidor MySQL.

```
int mysql_list_dbs(int [id_link] );
```

mysql_list_tables

Lista as tabelas em um banco de dados MySQL.

```
int mysql_list_tables(string nomebd, int [id_link] );
```

mysql_num_fields

Retorna o número de campos em um resultado.

```
int mysql_num_fields(int resultado);
```

mysql_num_rows

Retorna o número de registros em um resultado.

```
int mysql_num_rows(int resultado);
```

mysql_pconnect

Cria uma conexão persistente com um servidor MySQL.

```
int mysql_pconnect(string [hostname[:port]][:/path/tol  
socket]], string [usuário], string [senha]);
```

mysql_query

Envia uma consulta ao servidor MySQL. Retornará True se tiver sucesso; caso contrário, retornará False.

```
int mysql_query(string consulta, int [id_link] );
```

mysql_result

Lê os dados dos resultados.

```
int mysql_result(int resultado, int linha, mixed  
[campo] );
```

mysql_select_db

Seleciona um banco de dados.

```
int mysql_select_db(string nomebd, int [id_link] );
```

mysql_tablename

Retorna o nome da tabela do campo.

```
string mysql_tablename(int resultado, int i);
```


Política de Licenças do MySQL

O MySQL não é um software totalmente "free", uma licença será necessária sob certas condições:

- Quando o MySQL for utilizado na plataforma Windows.
- Quando você desejar ganhar dinheiro vendendo-o ou vendendo serviços que o requeiram.

Número de versão do MySQL

O esquema de nomes de releases no MySQL utiliza três números e um sufixo. Por exemplo, o release `mysql-3.21.17-beta` é interpretado como:

- O primeiro número (3) descreve o formato de armazenamento de tabelas do MySQL.
- O segundo número (21) é o "release level". Existem duas opções: uma é a versão estável e a outra é a versão de desenvolvimento.
- O terceiro número (17) é o número da versão dentro do nível release. É incrementado a cada nova distribuição.

O sufixo indica o nível de estabilidade do release. Se o sufixo não for especificado, significa que trata-se de uma versão estável. Os possíveis sufixos são:

Sufixo	Significado
alpha	Indica que o release contém seções de código que não foram totalmente testadas. Bugs conhecidos deverão ser documentados na seção News do manual. Normalmente são incluídos novos comandos e extensões.
beta	Indica que o novo código foi testado. Não foram adicionadas características significativas. Não deve conter bugs conhecidos.
gamma	Depois da versão beta ser utilizada por algum tempo sem problemas, ela é identificada como gamma.

Política de atualização de versões do MySQL

- O primeiro número é incrementado quando o formato de armazenamento de tabelas é alterado.
- O segundo número, release level, é alterado quando existem novas características ou incompatibilidades com versões anteriores.
- O terceiro número, versão dentro do release, é incrementado para pequenas atualizações.

Para saber sobre novos releases/versões, consulte as listas de discussão disponíveis no site do MySQL.

Diretórios default do MySQL

Esta seção descreve o layout default dos diretórios criados na instalação do MySQL.

Distribuição Binária

É a distribuição na qual os arquivos estão compilados e prontos para serem utilizados. Por exemplo: os arquivos .rpm da distribuição Conectiva. O diretório padrão para instalação é '/usr/local/mysql' com os seguintes subdiretórios:

Diretório	Conteúdo do diretório
bin	Programas cliente (mysql, mysqladmin, mysqldump, etc..) e o servidor mysqld.
data	Arquivos de log, bancos de dados.
include	Arquivos Include (header).
lib	Bibliotecas.
scripts	mysql_install_db.
share/mysql	Arquivos de mensagens de erro.
sql-bench	Benchmarks.

Distribuição Fonte

Uma distribuição que é instalada após a configuração e a compilação. Por default, a instalação instala arquivos sob '/usr/local', nos seguintes subdiretórios:

Diretório	Conteúdo do diretório
bin	Programas cliente (mysql, mysqladmin, mysqldump, etc..) e scripts.
include/mysql	Arquivos Include (header).
info	Documentação no formato Info.
lib/mysql	Bibliotecas.
libexec	O servidor mysqld.
share/mysql	Arquivos de mensagens de erro.
sql-bench	Benchmarks e o teste crash-me.
var	Arquivos de log, bancos de dados.

Diferenças nas estruturas dos diretórios

As diferenças nas estruturas dos diretórios entre as distribuições fonte e a binária são :

- O servidor mysqld é instalado no diretório 'libexec' em vez de 'bin'.
- O diretório de dados é 'var' em vez de 'data'.
- mysql_install_db é instalado em '/usr/local/bin' em vez de '/usr/local/mysql/scripts'.
- O diretórios de bibliotecas e headers são 'include/mysql' e 'lib/mysql' em vez de 'include' e 'lib'.

Versão do MySQL utilizada no Guia

Este guia está baseado nas versões 3.22 e 3.23 do MySQL. Considerando que o MySQL é um software em contínuo desenvolvimento, é possível que no momento em que você estiver lendo este guia, novas versões do MySQL tenham surgido. Verifique periodicamente o site do MySQL (www.mysql.com), em especial o manual on-line, para obter informações atualizadas sobre o MySQL. No site você também pode se inscrever em uma das listas de discussão do MySQL. Normalmente os participantes dessas listas são prestativos em responder dúvidas sobre o MySQL.

Bibliografia sobre MySQL

- MySQL de Paul DuBois, New Riders, 2000, ISBN 0-7357-0921-1

Excelente, cobre todos os recursos do MySQL. O autor faz parte da equipe de desenvolvimento do MySQL.

- MySQL & mSQL de Randy Jay Yarger, George Reese e Tim King, O'Reilly, 1999, ISBN 1-56592-434-7.

Notação utilizada neste Guia

Notação	Significado
<i>opção1</i> <i>opção2</i> ...	Barras verticais separam itens alternativos em uma lista.
{ }	Identifica itens requeridos na sintaxe. Digite somente a informação dentro dos colchetes; não digite os colchetes.
[]	Identifica valores opcionais que podem ou não ser fornecidos pelo usuário. Digite somente a informação dentro dos parênteses; não digite os parênteses.
<i>itálico</i>	Identifica um valor que deve ser fornecido pelo usuário.
negrito	Identifica palavras-reservadas do MySQL.
... (elipses)	Indica que o item prévio pode ser repetido.
[3.23]	Em sobrescrito, indica a versão onde um recurso foi implementado no MySQL.

Comentários e Sugestões

Comentários e sugestões sobre este guia serão bastante apreciados. Podem ser enviados para o e-mail:

leitor@novateceditora.com.br

Conheça o site da Novatec Editora em:

www.novateceditora.com.br

Símbolos

- ! 26
- != 26
- % 26
- & 26
- && 26
- * 26
- + 26
- 26
- # 48
- add-drop-table 67
- add-locks 67
- allow-keywords 67
- analyze 56
- backup 58
- basedir 52
- batch 60
- big-tables 52
- bind-address 52
- block-search 56
- bootstrap 52
- brief 62
- check 57
- chroot 52
- commit 62
- complete-insert 67
- compress 48
- console 54
- copy 62
- data-file-length 57
- datadir 52
- db 62
- debug 48, 62
- debug-info 60
- default-table-type 52
- defaults-file 48
- delayed-insert 67
- delete 69
- description 56
- enable-locking 52
- execute 60
- exit-info 52
- extend-check 56
- extended-insert 67
- fast 57
- fields-enclosed-by 68, 70
- fields-escaped-by 68, 70
- fields-optionally-enclosed-by 68, 70
- fields-terminated-by 68, 70
- flush 54
- flush-logs 67
- force 56, 58, 60, 63, 67, 69
- full 67
- help 48, 58
- host 48
- howto 62
- ignore 69
- ignore-space 60
- information 56
- init-file 52
- install 54
- join 59
- keys 71
- keys-used 56
- language 52
- lines-terminated-by 68, 70
- local 69
- lock-tables 67, 70
- log 53
- log-isam 53
- log-long-format 53
- log-update 53
- low-priority-updates 53
- medium-check 57
- new 53
- no-auto-rehash 60
- no-create-info 67
- no-data 67
- no-defaults 48
- no-symlinks 56
- old-protocol 53
- old_server 62
- one-database 60
- one-thread 53
- opt 67
- packlength 59
- password 48
- pid-file 53
- pipe 48
- plan 62
- port 49
- preview 62
- print-defaults 49
- quick 57, 61, 67
- raw 61
- read-only 57
- recover 57
- relative 63
- relnotes 62
- remove 54
- replace 70
- rhost 62
- rollback 62
- safe-mode 53
- safe-recover 57
- set-auto-increment 57
- set-variable 49
- silent 49
- skip-column-names 61
- skip-grant-tables 53
- skip-host-cache 53
- skip-line-numbers 61
- skip-locking 53
- skip-name-resolve 53
- skip-networking 53
- skip-new 53
- skip-show-database 53
- skip-thread-priority 53

Índice

--sleep 63
--socket 49
--sort-index 57
--sort-records 57
--spassword 62
--standalone 54
--status 71
--superuser 62
--tab 68
--table 61, 62
--test 59
--timeout 63
--tmp_dir 59
--tmpdir 53, 57
--unbuffered 61
--unpack 57
--user 49
--verbose 49
--version 49
--vertical 61
--wait 57, 59, 61, 63
--where 68
-? 48, 58
-A 57, 60
-a 56
-B 60
-b 52, 56, 58, 62
-C 48
-c 57, 67
-D 57
-d 56, 62, 67, 69
-E 61
-e 56, 60, 67
-F 57, 67
-f 56, 58, 60, 63, 67, 69
-H 62
-h 48, 52
-i 56, 60, 63, 69, 71
-j 59
-k 56, 71
-L 52, 61, 69
-l 53, 56, 67, 70
-m 57
-N 61
-n 53, 61
-O 49
-o 53, 57, 60
-P 49, 62
-p 48, 59
-q 57, 61, 67
-R 57
-r 57, 61, 70
-S 49, 57
-s 49
-Sg 53
-T 52, 57, 59, 60, 68
-t 53, 57, 59, 61, 62, 63, 67
-U 62
-u 49, 57

-V 49
-v 49
-W 48
-w 57, 59, 61, 63, 68
/ 26
< 26
<< 26
<= 26
<=> 26
<> 26
= 26
> 26
>= 26
>> 26
\\# 61
\\c 61
\\e 61
\\G 61
\\g 61
\\h 61
\\p 61
\\q 61
\\r 61
\\s 61
\\u 61
| 26
|| 26
~ 26

A

ABS 32
ACOS 32
ADD COLUMN 5
ADD INDEX 5
ADD PRIMARY KEY 5
ADD UNIQUE 5
ADDDATE 39
AFTER 5
AGGREGATE 7
ALL 13, 17, 19
ALTER 13, 17
alter 79
ALTER COLUMN 6
ALTER TABLE 5
AND 26
Arquivos de Opções 50
AS 16
ASC 20
ASCII 34
ASIN 32
ATAN 32
ATAN2 32
AUTO_INCREMENT 9, 10, 72
AVG 29
AVG_ROW_LENGTH 10

B

back_log 54
BENCHMARK 44
BETWEEN 27
BIGINT 73

BIN 34
 BINARY 27
 BIT_AND 29
 BIT_COUNT 44
 BIT_OR 29
 BLOB 75
 BOTH 38

C

CASE 31
 CEILING 32
 CHANGE COLUMN 6
 CHAR 34, 75
 CHAR_LENGTH 34
 CHARACTER SET 22
 CHARACTER_LENGTH 34
 CHECK TABLE 7
 CHECKSUM 10
 clear 61
 COALESCE 34
 columns_priv 79
 Comentários 46
 COMMENT 10
 CONCAT 34
 connect 61
 connect_timeout 54
 CONV 34
 COS 32
 COT 32
 COUNT 29
 CREATE 13, 17
 create 64, 79
 CREATE DATABASE 7
 CREATE FUNCTION 7
 CREATE INDEX 8
 CREATE TABLE 8
 CURDATE() 39
 CURRENT_DATE 39
 CURRENT_TIME 39
 CURRENT_TIMESTAMP 39
 CURTIME() 39

D

DATABASE 44
 DATE 74
 DATE_ADD 39
 DATE_FORMAT 40
 DATE_SUB 40
 DATETIME 74
 DAYNAME 40
 DAYOFMONTH 41
 DAYOFWEEK 41
 DAYOFYEAR 41
 db 79
 DECIMAL 74
 DECODE 44
 decode_bits 57
 DEFAULT 9
 DEGREES 32
 DELAY_KEY_WRITE 10
 DELAYED 14, 17

delayed_insert_limit 54
 delayed_insert_timeout 54
 delayed_queue_size 54
 DELETE 11, 13, 17
 delete 79
 DESC 20
 DESCRIBE 11
 DISTINCT 19, 29
 DISTINCTROW 19
 DOUBLE 73
 DOUBLE PRECISION 73
 DROP 13, 17
 drop 64, 80
 DROP COLUMN 6
 DROP DATABASE 11
 DROP FUNCTION 11
 DROP INDEX 6, 11
 DROP PRIMARY KEY 6
 DROP TABLE 12
 DUMPFILE 20

E

edit 61
 ego 61
 ELSE 31
 ELT 34
 ENCODE 44
 ENCRYPT 44
 END 31
 ENUM 76
 ESCAPE 27
 exit 61
 EXP 32
 EXPLAIN 12
 EXPORT_SET 35
 extended-status 64
 EXTRACT 41

F

FIELD 35
 FIELDS 15, 20
 FIELDS ENCLOSED BY 15, 20
 FIELDS ESCAPED BY 15, 20
 FIELDS TERMINATED BY 15, 20
 FILE 13, 17
 file 80
 FIND_IN_SET 35
 FIRST 5
 FLOAT 73
 FLOOR 32
 FLUSH 12
 flush-hosts 64
 flush-logs 64
 flush-privileges 64
 flush-status 64
 flush-tables 64
 flush_time 54
 FORMAT 35
 FROM 18, 20
 FROM_DAYS 41
 FROM_UNIXTIME 41

Índice

FULL 24
FUNCTION 7

G

GET_LOCK 44
go 61
GRANT 13
grant 80
GREATEST 30
GROUP BY 20

H

HAVING 20
HEAP 10
help 61
HEX 35
HIGH_PRIORITY 19
host 79
HOSTS 12
HOUR 41

I

Identificadores 82
IDENTIFIED BY 13
IF 30
IF EXISTS 11, 12
IF NOT EXISTS 8
IFNULL 30
IGNORE 5, 9, 14, 15
IGNORE n LINES 15
IN 27
INDEX 9, 13, 17
index 80
INFILE 15
INSERT 13, 14, 17, 35
insert 80
INSERT_ID 22
INSTR 35
INT 73
INTEGER 73
INTERVAL 30, 39
INTO 14, 17, 20
INTO TABLE 15
IS [NOT] NULL 26
ISAM 10
isamchk 56
ISNULL 30

K

KEY 9
key_buffer_size 55, 57
KILL 14
kill 64

L

LAST_INSERT() 45
LAST_INSERT_ID 22, 45
LCASE 35
LEADING 38
LEAST 30
LEFT 35
LENGTH 35
LIKE 23, 24, 25, 27
LIMIT 11, 21, 25

LINES TERMINATED BY 15, 20
LOAD DATA 14
LOAD_FILE 45
LOCAL 15
LOCATE 36
LOCK TABLES 16
LOG 32
LOG10 32
LOGS 12
long_query_time 55
LONGBLOB 76
LONGTEXT 76
LOW_PRIORITY 11, 14, 15, 16, 17, 25
LOWER 36
LPAD 36
LTRIM 36

M

MAKE_SET 36
MAX 29
max_allowed_packet 55, 61, 68
max_connect_errors 55
max_connections 55
max_delayed_insert_threads 55
max_join_size 55
MAX_ROWS 10
max_sort_length 55
MD5 45
MEDIUMBLOB 75
MEDIUMINT 73
MEDIUMTEXT 76
MID 36
MIN 29
MIN_ROWS 10
MINUTE 41
MOD 33
MODIFY COLUMN 6
MONTH 41
MONTHNAME 41
MYISAM 10
myisamchk 56
myisampack 58
mysql 60
mysql_affected_rows 84
mysql_close 84
mysql_connect 84
mysql_create_db 84
mysql_data_seek 84
mysql_db_query 84
MYSQL_DEBUG 51
mysql_drop_db 85
mysql_errno 85
mysql_error 85
mysql_fetch_array 85
mysql_fetch_field 85
mysql_fetch_lengths 85
mysql_fetch_object 85
mysql_fetch_row 85
mysql_field_flags 86
mysql_field_len 86

mysql_field_name 86
 mysql_field_seek 86
 mysql_field_table 86
 mysql_field_type 86
 mysql_free_result 86
 MYSQL_HISTFILE 61
 MYSQL_HOST 61
 mysql_insert_id 86
 mysql_list_dbs 87
 mysql_list_fields 86
 mysql_list_tables 87
 mysql_num_fields 87
 mysql_num_rows 87
 mysql_pconnect 87
 MYSQL_PWD 51
 mysql_query 87
 mysql_result 87
 mysql_select_db 87
 mysql_tablename 87
 MYSQL_TCP_PORT 51
 MYSQL_UNIX_PORT 51
 mysqlaccess 62
 mysqladmin 63
 mysqlbug 65
 mysqld 52
 mysqldump 66
 mysqlimport 69
 mysqlshow 71

N

NATIONAL 72
 net_buffer_length 55, 61, 68
 NOT 26
 NOT NULL 9
 NOW() 42
 NULL 9, 78
 NUMERIC 74

O

OCT 36
 OCTET_LENGTH 36
 ON 8, 13, 18
 Opções padrão 48
 OPTIMIZE TABLE 16
 OPTION 22
 OPTIONALLY 15, 20
 OR 26
 ORDER BY 20
 OUTFILE 20

P

pack_isam 58
 PACK_KEYS 10
 Palavras Reservadas 83
 PASSWORD 10, 22, 45
 password 64
 PERIOD_ADD 42
 PERIOD_DIFF 42
 PI 33
 ping 64
 POSITION 36
 POW 33

POWER 33
 PRIMARY KEY 9
 print 61
 PRIVILEGES 12
 PROCEDURE 21
 PROCESS 13, 18
 process 81
 processlist 64

Q

QUARTER 42
 quit 61

R

RADIANS 33
 RAND 33
 READ 16
 read_buffer_size 57
 REAL 74
 record_buffer 55
 refresh 64
 REGEXP 28
 rehash 61
 RELEASE_LOCK 45
 RELOAD 13, 18
 reload 64, 81
 RENAME 6
 REPAIR TABLE 16
 REPEAT 36
 REPLACE 9, 15, 17, 37
 RETURNS 7
 REVERSE 37
 REVOKE 17
 RIGHT 37
 ROUND 33
 ROW_FORMAT 10
 RPAD 37
 RTRIM 37

S

safe_mysqld 47, 52
 SEC_TO_TIME 42
 SECOND 42
 SELECT 13, 14, 17, 18
 select 81
 SESSION_USER() 45
 SET 14, 17, 22, 25, 76
 SHOW 23
 SHOW COLUMNS 23
 SHOW DATABASES 24
 SHOW GRANTS 24
 SHOW INDEX 24
 SHOW PROCESSLIST 24
 SHOW STATUS 24
 SHOW TABLE STATUS 24
 SHOW TABLES 24
 SHOW VARIABLES 25
 SHUTDOWN 13, 18
 shutdown 64, 81
 SIGN 33
 SIN 33
 SMALLINT 72

Índice

SONAME 7
sort_buffer 55
sort_buffer_size 57
sort_key_blocks 57
SOUNDEX 37
SPACE 37
SQL_AUTO_IS_NULL 22
SQL_BIG_RESULT 19
SQL_BIG_SELECTS 22
SQL_BIG_TABLES 22
SQL_LOG_OFF 23
SQL_LOG_UPDATE 23
SQL_LOW_PRIORITY_UPDATES 23
SQL_SELECT_LIMIT 23
SQL_SMALL_RESULT 19
SQL_WARNINGS 23
SQRT 33
STATUS 12
status 61, 64
STD 29
STDDEV 29
STRAIGHT_JOIN 19
STRCMP 30
SUBDATE 42
SUBSTRING 37
SUBSTRING_INDEX 37
SUM 29
SYSDATE() 42
SYSTEM_USER() 45

T

table_cache 55
TABLES 12
tables_priv 79
TAN 33
TEMPORARY 8, 46
TEXT 76
THEN 31
thread_stack 55
TIME 75
TIME_FORMAT 42
TIME_TO_SEC 42
TIMESTAMP 23, 74
TINYBLOB 75
TINYINT 72
TINYTEXT 76
Tipos de Colunas 72
tmp_table_size 55
TMPDIR 51
TO_DAYS 43
TRAILING 38
TRIM 38
TRUNCATE 33
TYPE 10

U

UCASE 38
UNIQUE 8, 9
UNIX_TIMESTAMP 43
UNLOCK TABLES 25
UNSIGNED 72

UPDATE 13, 18, 25
update 81
UPPER 38
USAGE 13, 18
USE 25
use 61
USER 51
user 79
USER() 45

V

VALUES 14, 17
VARCHAR 75
variables 64
Variáveis de Ambiente 51
version 64
VERSION() 45

W

wait_timeout 55
WEEK 43
WEEKDAY 43
WHEN 31
WHERE 11, 20, 25
WITH GRANT OPTION 13
WRITE 16
write_buffer_size 57

Y

YEAR 43, 75

Z

ZEROFILL 72

Limitações do MySQL pag 115 OR

O MySQL é um produto em contínuo desenvolvimento e é possível que essa lista abaixo esteja desatualizada no momento em que você estiver lendo este guia. No manual do MySQL, para cada um dos itens ausentes existe uma justificativa. Veja em www.mysql.com/Manual_chapter/manual_Todo.html a lista dos próximos recursos a serem implementados no MySQL. Os seguintes recursos estão ausentes da versão corrente (3.23.9-alpha) do MySQL.

Sub-selects

Independent sub-selects will be probably be available in 3.24.0.

SELECT INTO TABLE

MySQL suporta INSERT INTO...SELECT, que basicamente é a mesma coisa. Alternativamente, você pode usar SELECT INTO OUTFILE... ou CREATE TABLE ... SELECT.

Transações

we are currently working on a safe replication schema that we believe to be better than any commercial replication system we know of. This system will work most reliably under the atomic operations, non-transactional, paradigm. Stay tuned.

Stored procedures e triggers

The planned update language will be able to handle stored procedures, but without triggers. Triggers usually slow down everything, even queries for which they are not needed.

Foreign Keys

The FOREIGN KEY syntax in MySQL exists only for compatibility with other SQL vendors' CREATE TABLE commands; it doesn't do anything.

Views

MySQL por enquanto não suporta views, mas existe planos para implementá-las.

'--' como início de um comentário

Alguns bancos de dados SQL usam '--' para iniciar comentários. MySQL tem '#' como caractere de início de comentário, **even if the mysql command line tool removes all lines that start with '#'**. Você também pode usar o estilo de comentário */* comentário */* da linguagem C. A partir da versão 3.23.3 do MySQL é possível utilizar o estilo de comentário '--' somente se o comentário for

Script configure

O script configure permite que o mysql seja compilado de diversas maneiras, por exemplo, em diretórios diferentes, com suporte para depurar possíveis erros. As opções mais comuns são listadas abaixo.

shell> ./configure opção

Opção	Significado
--help	Lista as opções do script de configuração.
--without-server	Compila apenas as bibliotecas e os clientes sem compilar o servidor.
--with-debug	Configura o MySQL com suporte para debug mostrando possíveis erros e o que está acontecendo.
--prefix= <i>diretório</i>	Altera o diretório de instalação default.
--localstatedir=	Altera o diretório default para os arquivos do banco de dados.
--with-unix-socket-path= <i>diretório</i>	Faz com o MySQL coloque o arquivo de socket (Unix) em outro diretório em vez de /tmp ou /var/run. O nome deve incluir a localização absoluta do arquivo.
--with-client-ldflags=-all-static \	
--with-mysqld-ldflags=-all-static	Compila os programas com as bibliotecas ligadas dentro dos programas (statically-linked). Utilizado para criar uma distribuição binária, para acelerar a execução do MySQL ou para resolver alguns problemas com a distribuição do RedHat.
--with-charset= <i>char_set</i>	Altera o character set default. Por default, MySQL usa o character set ISO-8859-1 (Latin1).
--with-low-memory	Utiliza menos memória para compilar os programas. Usado para resolver problemas relacionados a falta de memória (Virtual memory exhausted, Internal compiler error: program cc1plus got fatal signal 11, Out of virtual memory).

comp_err

Atualiza as mensagens de erro do MySQL após o arquivo errmsg.txt ter sido modificado. O arquivo errmsg.txt contém as mensagens do MySQL em diversos idiomas.

```
comp_err errmsg.txt errmsg.sys
```

make_binary_release

Cria uma distribuição binária do MySQL que pode ser utilizada em sistemas semelhantes. Pode ser enviada via FTP para o diretório '/pub/mysql/Incoming' em ftp.tcx.se para que outros usuários possam utilizá-la.

sintaxe ????

mysql2mysql

Script shell para a conversão de programas feitos em mSQL para MySQL.

sintaxe ???

Colunas de outros bancos de dados

O MySQL mapeia as colunas de outros bancos de dados para facilitar a migração de sistemas feitos nestes sistemas para o MySQL. O mapeamento é transparente para o usuário e é realizado no processo de criação de uma tabela. Pode se usar um tipo de coluna que não existe no MySQL que o mesmo é mapeada para um equivalente do MySQL.

Outro banco de dados	Tipo MySQL
BINARY(NUM)	CHAR(NUM) BINARY
CHAR VARYING(NUM)	VARCHAR(NUM)
FLOAT4	FLOAT
FLOAT8	DOUBLE
INT1	TINYINT
INT2	SMALLINT
INT3	MEDIUMINT
INT4	INT
INT8	BIGINT
LONG VARBINARY	MEDIUMBLOB
LONG VARCHAR	MEDIUMTEXT
MIDDLEINT	MEDIUMINT
VARBINARY(NUM)	VARCHAR(NUM) BINARY

Sites sobre o MySQL

- www.mysql.com/
Documentação.
- www.phpbuilder.com/
Tutoriais, instalação.
- www.devshed.com/Server_Side/MySQL/
Tutoriais, instalação.
- linuxfocus.npi.msu.su/Portugues/July1998/article10.html
Tutorial em portugues.
- www.analysisandsolutions.com/code/mybasic.htm
Tutorial, instalação em plataformas Windows.
- www2.uol.com.br/info/forum/nov2.html
Lista de discussão da Info Exame em português.
- www.conectiva.com.br/
Dicas e soluções para instalar o MySQL no Linux Conectiva.
- linux-br.conectiva.com.br/
Lista de discussão sobre Linux em geral, mas com algumas mensagens sobre MySQL.
- Lista de discussão sobre instalação, configuração e otimização do MySQL, em inglês: mysql-subscribe@lists.mysql.com
- Lista de discussão sobre MySQL em português: mysql-br-request@listas.linkway.com.br. Coloque “subscribe mysql-br seuemail@seuprovedor” no corpo da mensagem.

mysql_setpermission

Utilitário que permite modificar a senha de um usuário, incluir um novo usuário ou incluir um novo banco de dados.

O `mysql_setpermission` não verifica as permissões existentes no banco de dados. O usuário que estiver executando o `mysql_setpermission` deve ter permissão para adicionar usuários, criar bancos de dados e mudar senhas.

mysampack [opções] nomearq ...

Parâmetro	Significado
-----------	-------------

--help	Exibe o texto de ajuda do utilitário.
--user	Especifica o usuário.
--password	Senha para se conectar ao servidor.
--host	Seleciona o endereço do servidor.
--port	Número da porta usada para conexão.

mysql_install_db

Cria as tabelas de permissões com os privilégios padrão. Deve ser executado quando o MySQL for instalado pela primeira vez.

sintaxe ???

mysql_zap

Finaliza processos no Unix baseado no padrão fornecido.

mysql_zap [-sinal] [-ft] padrão

Parâmetro	Significado
-----------	-------------

-sinal	Sinal a ser enviado para os processos. Valores padrão são 15 e depois 9.
-f	Não pergunta para o usuário antes de finalizar o processo.
-t	Mostra os processos apenas no stdout
padrão	Padrão a ser procurado.

print_defaults [3.23]

Imprime todos os argumentos que são usados em um programa usando os arquivos padrão.

print_defaults [opções] groups

print_defaults —config-file=my client mysql

Opção	Significado
-------	-------------

-c, --config-file=#	Arquivo de configuração a ser utilizado. Default é 'my'.
-, --help	Exibe o texto de ajuda do utilitário.
-, --version	Exibe a versão do utilitário.

mysql_convert_table_format[3.23]

Converte o formato da tabela para outro formato. Caso nenhuma tabela seja especificada, todas tabelas do banco de dados são convertidas.

mysql_convert_table_format *opções tabela*

Opção	Significado
<i>tabela</i>	Tabela a ser convertida.
<i>opções</i>	
--force	Continua mesmo se encontrar erros.
--help, --Information	Exibe o texto de ajuda do utilitário.
--host='nome'	Seleciona o endereço do servidor.
--password='senha'	Senha utilizada para conexão no servidor MySQL.
--type='tipo'	Tipo da nova tabela. O MySQL 3.23 suporta ISAM e MyISAM. O valor padrão é MySQL.
--user='login'	Login utilizado para conexão no servidor MySQL.
--verbose	Exibe informações detalhadas sobre a execução do utilitário.
--version	Exibe a versão do utilitário.

mysql_fix_privilege_tables

Utilitário utilizado quando é feita uma atualização da versão 3.21 para a versão 3.22. Ele deve ser executado logo após a instalação da versão 3.22 e acrescenta as permissões necessárias para o uso do comando GRANT.

Access Denied é um erro encontrado se o mysql_fix_privilege_tables não for executado após a atualização.

sintaxe ???

mysql_find_rows [3.23]

Mostra todos os comandos SQL que estão no formato regexp ou contém 'use database' ou 'set ...'. O comando SQL pode conter quebras de linhas.

sintaxe ???

Opção	Significado
--help, --Information	Exibe o texto de ajuda do utilitário.
--regexp=#	Padrão a ser procurado.
--start_row=N	Começa a mostrar os resultados a partir da coluna N. O valor padrão é 1.
--rows=N	Finaliza após N colunas.

mysqlhotcopy [3.23]

Faz um cópia do banco de dados usando lock e flush evitando assim inconsistências nos dados.

mysqlhotcopy *nome_bd novo_nome_bd*

Opção	Significado
-?, --help	Exibe o texto de ajuda do utilitário.
-u, --user=login	Login utilizado para conexão no servidor MySQL.
-p, --password	Senha utilizada para conexão no servidor MySQL.
-P, --port=porta	Número da porta utilizada para conexão no servidor MySQL.
--allowold	Caso novo_nome_bd exista renomeia para novo_nome_bd_old.
--keepold	Não remove os bancos de dados antigos.
--noindices	Não copia os arquivos de índices.
--method=#	Seleciona o método para a cópia. Somente "cp" é suportado atualmente.
--quiet	Mostra somente os erros.
--debug=N	Mostra informações para debug (0..3)

perror

Mostra a mensagem de erro baseado no seu número de erro.

perror *opções código_de_erro*

Opção	Significado
<i>código_de_erro</i>	Código de erro que se deseja saber o significado.
<i>opções</i>	
--?, --help, --l, --info	Exibe o texto de ajuda do utilitário.
--verbose	Exibe informações detalhadas sobre a execução do utilitário.
--version	Exibe a versão do utilitário.
Exemplo :	
shell> perror 56	
Código de requisição inválido	

replace

Utilitário utilizado para substituir um string por outro string.

replace *opção string novostring [string novostring....]*

Opção	Significado
<i>opção</i>	
-?, -l	Exibe o texto de ajuda do utilitário.
-s	Executa em modo silencioso.
-v	Exibe informações detalhadas da execução do utilitário.
<i>string</i>	
^	O string deve estar no início da linha.
\$	O string deve estar no fim da linha.
b	Procura por um espaço e início ou fim de linha.

Quem desenvolve o MySQL

O MySQL nasceu da necessidade que a empresa sueca TcX teve em trabalhar com uma base de dados muito extensa em uma plataforma barata. Naquela época, 1996, gerenciadores como o Oracle necessitavam de grandes investimentos tanto para compra do servidor de banco de dados como para compra de equipamentos. Então, começaram utilizando um servidor de banco de dados popular na época chamado mSQL (www.hughes.com.au), porém constataram que não era o suficiente para as suas necessidades. Decidiram então desenvolver o MySQL, utilizando a linguagem C/C++, visando rapidez, robustez e economia.

Atualmente a equipe que trabalha com o MySQL é pequena, mas muitas pessoas contribuem com o desenvolvimento através de verificação de erros, respondendo as perguntas em listas de discussão (são mais de 100 mensagens todos os dias), ou enviando modificações no código fonte. O líder da equipe de desenvolvimento é Michael Widenius, mas conhecido na lista de discussão do MySQL pelo seu apelido "Monty".

Comparando o MySQL com o mSQL.

O MySQL a primeira vista é parecido com o mSQL. O mSQL é mais eficiente em alguns pontos, como nos comandos selects/create tables básicos e no tempo para estabelecer conexões entre os clientes e o servidor. Porém, o MySQL é um produto superior em diversos pontos pois:

- Está em constante desenvolvimento com atualizações sendo lançadas a todo momento. Basta comparar a seção “news” de ambos os sites : <http://www.mysql.com> e <http://www.hughes.com.au/products/msql/>
- Possui uma larga base de usuários que participam ativamente das listas de discussões.
- Maior velocidade no uso de tabelas grandes e selects complexos.
- Melhor capacidade para atender diversos cliente simultaneamente.
- Possui uma licença de uso mais flexível.
- Suporte total a HAVING e GROUP BY.
- Diversas funções podem ser utilizadas dentro do SELECT.
- Possui mais opções de interfaces para JAVA e PERL.

Estágios do controle de acesso

O controle de acesso é feito em duas partes:

Parte 1

O servidor verifica se aceita ou não a conexão com o cliente através da tabela user.

Parte 2

O servidor verifica se usuário possui permissão para executar o comando procurando nas tabelas na seguinte ordem :

- Verifica a tabela user procurando pela permissão.
- Caso não encontre, procura nas tabelas db e host pela permissão.
- Caso não encontre, procura na tabela tables_priv.
- Caso não encontre, procura na tabela columns_priv.

Exemplo

O usuário FOO do endereço 192.168.0.1 quer executar um SELECT host FROM mysql.tabela

- O servidor procura na tabela host por Host=192.168.0.1, User=FOO. Se encontrar aceita a conexão.
- O servidor procura na tabela host por Host=192.168.0.1, User=FOO, Select_priv=Y. Se encontrar executa o select.
- O servidor procura na tabela host por Host=192.168.0.1, User=FOO, Select_priv=Y, Db=mysql. Se encontrar executa o select.
- O servidor procura na tabela host por Host=192.168.0.1, Select_priv=Y, Db=mysql. Se encontrar executa o select.
- O servidor procura na tabela tables_priv por Host=192.168.0.1, Select_priv=Y, Db=mysql, User=FOO, Table_name=tabela. Se encontrar executa o select.
- O servidor procura na tabela tables_priv por Host=192.168.0.1, Select_priv=Y, Db=mysql, User=FOO, Table_name=tabela, Column_name=host. Se encontrar executa o select.

Os caracteres '%' e '_' podem ser usados no campo Host em todas as tabelas. Os caracteres '%' ou em branco significam qualquer endereço. Os campos Db, Table_name e Column_name não podem conter '%' ou estarem em branco.

Caso as tabelas sejam alteradas usando INSERT, DELETE ou UPDATE, utilize o comando FLUSH PRIVILEGES para que as atualizações tenham efeito.

A opção --skip-grant-tables faz com que o servidor (ver

Outros Programas

mysql_setpermission

Altera a senha de um usuário, inclui um novo usuário ou inclui um novo banco de dados.

mysql_install_db

Cria tabelas de permissões com os privilégios padrão.

mysql_zap

Finaliza processos no Unix baseado no padrão fornecido.

print_defaults ^[3.23]

Exibe os argumentos que são usados em um programa usando os arquivos padrão.

mysql_convert_table_format ^[3.23]

Converte o formato de uma tabela para outro formato.

mysql_fix_privilege_tables

Utilitário utilizado quando é feita uma atualização da versão 3.21 para a versão 3.22. Ele deve ser executado logo após a instalação da versão 3.22 e acrescenta as permissões necessárias para o uso do comando GRANT.

mysql_find_rows ^[3.23]

Mostra todos os comandos SQL que estão no formato regexp ou contém 'use database' ou 'set ...'.

mysqlhotcopy ^[3.23]

Copia o banco de dados usando lock e flush, evitando inconsistências nos dados.

perror

Exibe a mensagem de erro baseado no seu número de erro.

replace

Substitui um string por outro string.

MySQL

- O MySQL é um gerenciador de bancos de dados poderoso, estável, extremamente eficiente, com milhares de instalações no mundo inteiro. É a solução ideal para empresas e sites de pequeno e médio portes.
- Este Guia de Consulta Rápida contém uma referência completa dos recursos do MySQL. Descreve comandos SQL, funções, utilitários, operadores, tipos de dados e muito mais. É indicado tanto para programadores web, quanto para administradores de bancos de dados.
- Indispensável para quem quer obter o máximo proveito do MySQL, sem perder tempo consultando volumosos manuais.
- Prático para carregar e consultar.