functions to use in select and where clauses

comparision operators

- = <> <= > >= <>

case when (compare-value) then result when [compare-value] then result ... else result end when [condition] then result when [condition] then result ... [else result] end if(exp1,exp2,exp3) ifnull(exp1,exp2)

logical operators

and (and) or () or ()

control flow functions

case value when [compare-value] then result when [compare-value] then result ... [else result] end case when [condition] then result when [condition] then result ... [else result] end if(exp1,exp2,exp3)

string functions

ascii(str) bin(n) bit_length(str) chr(n)

cast function

case (expression as type) convert (expression, type)

data manipulation language

insert into table_name (col_name,...)

set col_name1=expression, col_name2=expression,...

on duplicate key update col_name1=expression, col_name2=expression,...

update table_name (col_name,...)

set col_name1=expression, col_name2=expression,...

on duplicate key update col_name1=expression, col_name2=expression,...

delete from table_name (col_name,...)

on delete cascade

on delete restrict

on delete null

on delete set col_name=expression, col_name2=expression,...

truncate table table_name

replace into table_name (col_name,...)

on duplicate key update col_name1=expression, col_name2=expression,...

on delete cascade

on delete restrict

on delete null

on delete set col_name=expression, col_name2=expression,...

select (column list)

from table_name

where column=expression, column2=expression,...

[union select ...

[into outfile 'file_name' export_options]

[set column_name=expression, column2=expression,...]

[where where_definition]

[order by order_definition]

[limit #]

[offset,] rows

function

coalesce(list) expr not in

is not null

expr between min and

max

expr in (value,...)

LOGICAL OPERATORS

and (and) or () or ()

control flow functions

case when (compare-value) then result when [compare-value] then result ... else result end when [condition] then result when [condition] then result ... [else result] end if(exp1,exp2,exp3) ifnull(exp1,exp2)

logical operators

and (and) or () or ()

function

coalesce(list) expr not in

is not null

expr between min and

max

expr in (value,...)

string functions

ascii(str) bin(n) bit_length(str) chr(n)

cast function

case (expression as type) convert (expression, type)

data manipulation language

insert into table_name (col_name,...)

set col_name1=expression, col_name2=expression,...

on duplicate key update col_name1=expression, col_name2=expression,...

update table_name (col_name,...)

set col_name1=expression, col_name2=expression,...

on duplicate key update col_name1=expression, col_name2=expression,...

delete from table_name (col_name,...)

on delete cascade

on delete restrict

on delete null

on delete set col_name=expression, col_name2=expression,...

truncate table table_name

replace into table_name (col_name,...)

on duplicate key update col_name1=expression, col_name2=expression,...

on delete cascade

on delete restrict

on delete null

on delete set col_name=expression, col_name2=expression,...

union

select ...

union select ...

union select ...

union (all)
LOAD DATA INFILE
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name.txt'
[REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
[TERMINATED BY '"'
[OPTIONALLY] ENCLOSED BY '"'
[ESCAPED BY '\']
[LINES TERMINATED BY '
']
[IGNORE number LINES]
[AS column_name,...]
[REPLACE | IGNORE]
INTO TABLE tbl_name
[FIELDS
[TERMINATED BY '	'
[OPTIONALLY] ENCLOSED BY '"'
[ESCAPED BY '\']
[LINES TERMINATED BY '
']
[IGNORE number LINES]
[AS column_name,...]

CREATE DATABASE
CREATE DATABASE [IF NOT EXISTS] db_name
DROP DATABASE
DROP DATABASE [IF EXISTS] db_name

CREATE TABLE
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
[(create_definition,...)
   [table_options]
   [select_statement]]
ALTER TABLE
ALTER [IGNORE] TABLE tbl_name alter_spec [alter_spec ...]
RENAME TABLE
RENAME TABLE tbl_name TO new_table_name1, tbl_name2 TO new_table_name2,...
DROP TABLE
DROP TABLE [IF EXISTS] tbl_name [, tbl_name,...] [RESTRICT | CASCADE]
CREATE INDEX
CREATE [UNIQUE|FULLTEXT] INDEX index_name ON tbl_name
(col_name[(length)],...)
DROP INDEX
DROP INDEX index_name ON tbl_name

Basic MySQL User Utility Commands
USE
USE db_name
DESCRIBE
DESC tbl_name (col_name | wild)

MySQL Transactional and Locking Commands
BEGIN/COMMIT/ROLLBACK
BEGIN;
...
COMMIT;
LOCK/UNLOCK TABLES
LOCK TABLES tbl_name [AS alias]
[READ | READ LOCAL | [LOW_PRIORITY] WRITE]
[, tbl_name [READ | [LOW_PRIORITY] WRITE] ...]...
UNLOCK TABLES
SET TRANSACTIONAL
SET [GLOBAL | SESSION] TRANSACTION ISOLATION LEVEL [READ UNCOMMITTED | READ COMMITTED | REPEATABLE READ | SERIALIZABLE]

GENERAL REPLACEMENTS

<table>
<thead>
<tr>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NOT] where_expr</td>
</tr>
</tbody>
</table>

| index_col_name: |
| col_name [(length)] |

| join_condition: |
| ON conditional_expr | USING (column_list) |