

Column Types

[NATIONAL] CHAR(M) [BINARY]	LONGBLOB
[NATIONAL] VARCHAR(M) [BINARY]	LONGTEXT
BIGINT[(M)] [UNSIGNED] [ZEROFILL]	MEDIUMBLOB
BIT	MEDIUMINT[(M)] [UNSIGNED]
BLOB	[ZEROFILL]
BOOL	MEDIUMTEXT
CHAR	NUMERIC(M,D) [ZEROFILL]
DATE	REAL[(M,D)] [ZEROFILL]
DATETIME	SET('value1','value2',...)
DECIMAL[(M,D)] [ZEROFILL]	SMALLINT[(M)] [UNSIGNED]
DOUBLE PRECISION[(M,D)]	[ZEROFILL]
[ZEROFILL]	TEXT
DOUBLE[(M,D)] [ZEROFILL]	TIME
ENUM('value1','value2',...)	TIMESTAMP[(M)]
FLOAT(precision) [ZEROFILL]	TINYBLOB
FLOAT[(M,D)] [ZEROFILL]	TINYINT[(M)] [UNSIGNED]
INT[(M)] [UNSIGNED] [ZEROFILL]	[ZEROFILL]
INTEGER[(M)] [UNSIGNED]	TINYTEXT
[ZEROFILL]	YEAR[(2 4)]

FUNCTIONS TO USE IN SELECT AND WHERE CLAUSES

COMPARISON OPERATORS

=	<>	!=	<=	<	>=	>	<=>
COALESCE(list)	expr NOT IN	IS NOT NULL					
expr BETWEEN min AND max	(value,...)	IS NULL					
expr IN (value,...)	INTERVAL(N,N1,N2,N3,...)	ISNULL(expr)					

LOGICAL OPERATORS

AND (&&)	NOT (!)	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(expr1,expr2,expr3)
IFNULL(expr1,expr2)
NULLIF(expr1,expr2)
```

STRING FUNCTIONS

ASCII(str)	LPAD(str,len,padstr)
BIN(N)	LTRIM(str)
BIT_LENGTH(str)	MAKE_SET(bits,str1,str2,...)
CHAR(N,...)	MID(str,pos,len)
CHAR_LENGTH(str)	OCTET(N)
CHARACTER_LENGTH(str)	OCTET_LENGTH(str)
CONCAT(str1,str2,...)	ORD(str)
CONCAT_WS(separator, str1, str2,...)	POSITION(substr IN str)
CONV(N,from_base,to_base)	REPEAT(str,count)
ELT(N,str1,str2,str3,...)	REPLACE(str,from_str,to_str)
EXPORT_SET(bits,on,off,[separator ,number_of_bits])	REVERSE(str)
FIELD(str,str1,str2,str3,...)	RIGHT(str,len)
FIND_IN_SET(str,strlist)	RPAD(str,len,padstr)
HEX(N_or_S)	RTRIM(str)
INSERT(str,pos,len,newstr)	SOUNDEX(str)
INSTR(str,substr)	SPACE(N)
LCASE(str)	SUBSTRING(str FROM pos FOR len)
LEFT(str,len)	SUBSTRING(str FROM pos)
LENGTH(str)	SUBSTRING(str,pos)
LOAD_FILE(file_name)	SUBSTRING(str,pos,len)
LOCATE(substr,str)	SUBSTRING_INDEX(str,delim,count)
LOCATE(substr,str,pos)	TRIM([BOTH LEADING TRAILING] [remstr] FROM str)
LOWER(str)	UCASE(str)
	UPPER(str)

MATHEMETICAL FUNCTIONS

-	COS(X)	LOG10(X)	ROUND(X)
ABS(X)	COT(X)	MOD(N,M)	ROUND(X,D)
ACOS(X)	DEGREES(X)	PI()	SIGN(X)
ASIN(X)	EXP(X)	POW(X,Y)	SIN(X)
ATAN(X)	FLOOR(X)	POWER(X,Y)	SQRT(X)
ATAN(Y,X)	GREATEST(X,Y,...)	RADIANS(X)	TAN(X)
ATAN2(Y,X)	LEAST(X,Y,...)	RAND()	TRUNCATE(X,D)
CEILING(X)	LOG(X)	RAND(N)	

STRING COMPARISON FUNCTIONS

```
expr LIKE pat [ESCAPE 'escape-char']
expr NOT LIKE pat [ESCAPE 'escape-char']
expr NOT REGEXP pat
expr NOT RLIKE pat
expr REGEXP pat
expr RLIKE pat
MATCH (col1,col2,...) AGAINST (expr IN BOOLEAN MODE)
STRCMP() returns 0 if the strings are the same, -1 if the MATCH (col1,col2,...) AGAINST (expr)
STRCMP(expr1,expr2)
```

ARITHMETIC OPERATIONS

+ - * /

DATE AND TIME FUNCTIONS

```
ADDDATE(date,INTERVAL expr type) MONTHNAME(date)
CURDATE() NOW()
CURRENT_DATE PERIOD_ADD(P,N)
CURRENT_TIME PERIOD_DIFF(P1,P2)
CURRENT_TIMESTAMP QUARTER(date)
CURTIME() SEC_TO_TIME(seconds)
DATE_ADD(date,INTERVAL expr type) SECOND(time)
DATE_FORMAT(date,format) SUBDATE(date,INTERVAL expr type)
DATE_SUB(date,INTERVAL expr type) SYSDATE()
DAYNAME(date) TIME_FORMAT(time,format)
DAYOFMONTH(date) TIME_TO_SEC(time)
DAYOFWEEK(date) TO_DAYS(date)
DAYOFYEAR(date) UNIX_TIMESTAMP()
EXTRACT(type FROM date) UNIX_TIMESTAMP(date)
FROM_DAYS(N) WEEK(date)
FROM_UNIXTIME(unix_timestamp) WEEK(date,first)
FROM_UNIXTIME(unix_timestamp,form at) WEEKDAY(date)
HOUR(time) YEAR(date)
MINUTE(time) YEARWEEK(date)
MONTH(date) YEARWEEK(date,first)
```

CAST FUNCTIONS

```
CAST(expression AS type) CONVERT(expression,type)
```

BIT FUNCTIONS

| & << >> ~ BIT_COUNT(N)

MISCELLANEOUS FUNCTIONS

```
BENCHMARK(count,expr) GET_LOCK(str,timeout)
CONNECTION_ID() INET_ATON(expr)
DATABASE() INET_NTOA(expr)
DECODE(encrypt_str,pass_str) LAST_INSERT_ID(expr)
des_decrypt(string_to_decrypt [, key_string]) MASTER_POS_WAIT(log_name, log_pos)
des_encrypt(string_to_encrypt, flag, [, (key_number | key_string) ]) MD5(string)
PASSWORD(str)
ENCODE(str,pass_str) RELEASE_LOCK(str)
ENCRYPT(str,[salt]) SESSION_USER()
FORMAT(X,D) SYSTEM_USER()
FOUND_ROWS() USER()
```

FUNCTIONS FOR USE WITH GROUP BY CLAUSES

```
COUNT(expr) AVG(expr) STD(expr)
COUNT(DISTINCT expr,[expr...]) MIN(expr) STDDEV(expr)
MAX(expr) BIT_OR(expr)
SUM(expr) BIT_AND(expr)
```

DATA MANIPULATION LANGUAGE

INSERT

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name [(col_name,...)] VALUES (expression,...),(...),...
INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name [(col_name,...)] SELECT ...
INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name SET col_name=expression, col_name=expression, ...
INSERT [LOW_PRIORITY] [IGNORE] [INTO] tbl_name [(column list)] SELECT ...
INSERT DELAYED ...
```

SELECT

```
SELECT [STRAIGHT_JOIN] [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT] [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS] [HIGH_PRIORITY] [DISTINCT | DISTINCTROW | ALL] select_expression,...
[INTO [OUTFILE | DUMPFILE] 'file_name' export_options]
[FROM table_references
[WHERE where_definition]
[GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
[HAVING where_definition]
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
[LIMIT [offset,] rows]
[PROCEDURE procedure_name]
[FOR UPDATE | LOCK IN SHARE MODE]]
```

JOIN

```
table_reference, table_reference
table_reference [CROSS] JOIN table_reference
table_reference INNER JOIN table_reference join_condition
table_reference STRAIGHT_JOIN table_reference
table_reference LEFT [OUTER] JOIN table_reference join_condition
table_reference LEFT [OUTER] JOIN table_reference
table_reference NATURAL [LEFT [OUTER]] JOIN table_reference
{ o_j table_reference LEFT OUTER JOIN table_reference ON conditional_expr }
table_reference RIGHT [OUTER] JOIN table_reference join_condition
table_reference RIGHT [OUTER] JOIN table_reference
table_reference NATURAL [RIGHT [OUTER]] JOIN table_reference
```

HANDLER

```
HANDLER table OPEN [ AS alias ]
HANDLER table READ index { = | >= | <= | < } (value1, value2, ... ) [ WHERE ... ] [LIMIT ... ]
HANDLER table READ index { FIRST | NEXT | PREV | LAST } [ WHERE ... ] [LIMIT ... ]
HANDLER table READ { FIRST | NEXT } [ WHERE ... ] [LIMIT ... ]
HANDLER table CLOSE
```

UPDATE

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name SET col_name1=expr1, [col_name2=expr2, ...] [WHERE where_definition] [LIMIT #]
```

DELETE

```
DELETE [LOW_PRIORITY | QUICK] FROM table_name [WHERE where_definition] [ORDER BY ...] [LIMIT rows]
DELETE [LOW_PRIORITY | QUICK] table_name[.*] [,table_name[.*] ...] FROM table_references [WHERE where_definition]
DELETE [LOW_PRIORITY | QUICK] FROM table_name[.*] [,table_name[.*] ...] USING table_references [WHERE where_definition]
```

TRUNCATE

```
TRUNCATE TABLE table_name
```

REPLACE

```
REPLACE [LOW_PRIORITY | DELAYED] [INTO] tbl_name [(col_name,...)] VALUES (expression,...),(...),...
REPLACE [LOW_PRIORITY | DELAYED] [INTO] tbl_name [(col_name,...)] SELECT ...
REPLACE [LOW_PRIORITY | DELAYED] [INTO] tbl_name SET col_name=expression, col_name=expression,...
```

UNION

```
SELECT ...
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 '\t']
[[OPTIONALLY] ENCLOSED BY ''
[ESCAPED BY '\\']]
[LINES TERMINATED BY '\n']
[IGNORE number LINES]
[(col_name,...)]
```

DATA DEFINITION LANGUAGE

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_name[, 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

```
{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 TRANSACTIONS

```
SET [GLOBAL | SESSION] TRANSACTION ISOLATION LEVEL [READ UNCOMMITTED
| READ COMMITTED | REPEATABLE READ | SERIALIZABLE]
```

GENERAL REPLACEMENTS

type:

```
see COLUMN TYPES
```

index_col_name:

```
col_name [(length)]
```

join_condition:

```
{ ON conditional_expr | USING (column_list) }
```

create_definition:

```
{ col_name type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [PRIMARY KEY] [reference_definition]
| PRIMARY KEY (index_col_name,...)
| KEY [index_name] (index_col_name,...)
| INDEX [index_name] (index_col_name,...)
| UNIQUE [INDEX] [index_name] (index_col_name,...)
| FULLTEXT [INDEX] [index_name] (index_col_name,...)
| [CONSTRAINT symbol] FOREIGN KEY [index_name]
(index_col_name,...) [reference_definition]
| CHECK (expr) }
```

reference_definition:

```
REFERENCES tbl_name [(index_col_name,...)]
[MATCH FULL | MATCH PARTIAL]
[ON DELETE reference_option]
[ON UPDATE reference_option]
```

reference_option:

```
{ RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT }
```

table_options:

```
{ TYPE = {BDB | HEAP | ISAM | InnoDB | MERGE | MRG_MYISAM | MYISAM }
| AUTO_INCREMENT = #
| AVG_ROW_LENGTH = #
| CHECKSUM = {0 | 1}
| COMMENT = "string"
| MAX_ROWS = #
| MIN_ROWS = #
| PACK_KEYS = {0 | 1 | DEFAULT}
| PASSWORD = "string"
| DELAY_KEY_WRITE = {0 | 1}
| ROW_FORMAT= { default | dynamic | fixed | compressed }
| RAID_TYPE= {1 | STRIPED | RAID0 } RAID_CHUNKS=#RAID_CHUNKSIZE=#
| UNION = (table_name,[table_name...])
| INSERT_METHOD= {NO | FIRST | LAST}
| DATA DIRECTORY="absolute path to directory"
| INDEX DIRECTORY="absolute path to directory" }
```

select_statement:

```
[IGNORE | REPLACE] SELECT ... (Some legal select statement)
```

alter_spec:

```
{ ADD [COLUMN] create_definition [FIRST | AFTER column_name ]
| ADD [COLUMN] (create_definition, create_definition,...)
| ADD INDEX [index_name] (index_col_name,...)
| ADD PRIMARY KEY (index_col_name,...)
| ADD UNIQUE [index_name] (index_col_name,...)
| ADD FULLTEXT [index_name] (index_col_name,...)
| ADD [CONSTRAINT symbol] FOREIGN KEY index_name
(index_col_name,...) [reference_definition]
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name create_definition
[FIRST | AFTER column_name]
| MODIFY [COLUMN] create_definition
[FIRST | AFTER column_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP INDEX index_name
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO] new_tbl_name
| ORDER BY col
| table_options }
```

table-reference:

```
table_name [[AS] alias] [USE INDEX (key_list)]
[IGNORE INDEX (key_list)]
```

where_definition:

```
(NOT) { where_expr
| where_expr [ AND | OR ] where_expr }
```

where_expr:

```
{ column_name > | >= | = | <> | <= | < | column_name_or_constant
| column_name LIKE column_name_or_constant
| column_name IS NULL
| column_name IS NOT NULL
| ( where_definition ) }
```



Quick Reference

MySQL

Version 4.0.2-alpha

<http://www.mysql.com/documentation/>

Table of Contents:

Column Types

Functions to use in SELECT and WHERE clauses

Data Manipulation Language

Data Definition Language

Basic MySQL User Utility Commands

MySQL Transactional and Locking Commands

Benjamin Jung
Trinity College Dublin,
Ireland

Benjamin.Jung@cs.tcd.ie
<http://www.cs.tcd.ie/Benjamin.Jung/>