Column Types

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

FUNCTIONS TO USE IN SELECT AND WHERE CLAUSES

COMPARISON OPERATORS

```
=   <>   !=   <=   <   >=   >   <=>

COALESCE(list)
expr BETWEEN min AND
  max
expr IN (value,...)
expr NOT IN
  (value,...)
INTERVAL(N,N1,N2,N3,..
  .)
IS NOT NULL
IS NULL
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)
BIN(N)
BIT_LENGTH(str)
CHAR(N,...)
CHAR_LENGTH(str)
CHARACTER_LENGTH(str)
CONCAT(str1,str2,...)
CONCAT_WS(separator, str1,
  str2,...)
CONV(N,from_base,to_base)
ELT(N,str1,str2,str3,...)
EXPORT_SET(bits,on,off,[separator
  ,[number_of_bits]])
FIELD(str,str1,str2,str3,...)
FIND_IN_SET(str,strlist)
HEX(N_or_S)
INSERT(str,pos,len,newstr)
INSTR(str,substr)
LCASE(str)
LEFT(str,len)
LENGTH(str)
LOAD_FILE(file_name)
LOCATE(substr,str)
LOCATE(substr,str,pos)
LOWER(str)
LPAD(str,len,padstr)
LTRIM(str)
MAKE_SET(bits,str1,str2,...)
MID(str,pos,len)
OCT(N)
OCTET_LENGTH(str)
ORD(str)
POSITION(substr IN str)
REPEAT(str,count)
REPLACE(str,from_str,to_str)
REVERSE(str)
RIGHT(str,len)
RPAD(str,len,padstr)
RTRIM(str)
SOUNDEX(str)
SPACE(N)
SUBSTRING(str FROM pos FOR len)
SUBSTRING(str FROM pos)
SUBSTRING(str,pos)
SUBSTRING(str,pos,len)
SUBSTRING_INDEX(str,delim,count)
TRIM([[BOTH | LEADING | TRAILING]
  [remstr] FROM] str)
UCASE(str)
UPPER(str)
```

MATHEMETICAL FUNCTIONS

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

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

CAST FUNCTIONS

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

BIT FUNCTIONS

```
|   &   <<   >>   ~   BIT_COUNT(N)
```

MISCELLANEOUS FUNCTIONS

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

FUNCTONS FOR USE WITH GROUP BY CLAUSES

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

deepX Ltd.

Dublin, Ireland

info@deepX.com
http://www.deepX.com/