Calpont InifiniDB^{TM} SQL Syntax Guide

Release 1.0.3
Document Version 1.0.3-1
March 2010

Table of Contents

Chapter 4   **SQL Syntax**   **17**

Chapter 5   **Operating Mode**   **33**

InfiniDB Introduction

Calpont InfiniDB is a high performance, scalable data warehouse storage engine. To enable the use of Calpont InfiniDB by end users and applications, it is necessary to integrate Calpont InfiniDB with a database front end. This document describes the supported SQL syntax available to users in the combined deployment of Calpont InfiniDB and MySQL v5.1.

- ◆ Querying the database
- ◆ Inserting, updating, and deleting rows
- ◆ Altering, creating, and deleting objects

This guide does not document all MySQL SQL statements. This guide specifically lists the datatypes, expressions, and conditions that are allowed using the Calpont InfiniDB. In addition, it lists SQL statements that run native on Calpont InfiniDB. Statements that run native on Calpont InfiniDB deliver optimal performance results that exceed the results of other MySQL commands not documented in this guide. See the InfiniDB Operating Mode section for more information.

This guide assumes that Calpont InfiniDB has been installed and configured with MySQL.

Audience

This guide is intended for database administrators and business intelligence managers that use Calpont InfiniDB to perform queries for business intelligence reporting and updates to the data warehouse.

Document standards

The following typographical conventions and user alerts are used throughout this guide:

Table 1: Typographical Conventions

Item	Description
Bold Typeface	Characters you type exactly as shown. For example: Type **getLogInfo** You would type: **getLogInfo**
Bold Italic Typeface	Used as a variable or placeholder. Type the appropriate replacement text. Variables identified with more than one word are connected by underscores. Examples: Type ***ID*** You would type the ID number: **34878** Type ***IP_address*** You would type the IP address: **110.68.52.01**

Table 2: User Alerts

Item	Description
	Note: Informs you of helpful information.
	Warning: Alerts you to possible hardware and/or software failures that can lead to loss of data or corruption.

List of documentation

Calpont InfiniDB documentation consists of several guides intended for different audiences. The documentation is described in the following table:

Table 3: Documentation

Document	Description
Calpont InfiniDB Administrator Guide	Provides detailed steps for maintaining Calpont InfiniDB.
Calpont InfiniDB Minimum Recommended Technical Specifications	Lists the minimum recommended hardware and software specifications for implementing Calpont InfiniDB.
Calpont InfiniDB JumpStart Guide	Provides quick setup tasks for installing and configuring software.
Calpont InfiniDB Installation Guide	Contains a summary of steps needed to perform an install of Calpont InfiniDB in a distributed configuration
Calpont InfiniDB Concepts Guide	Introduction to the Calpont InfiniDB analytic database.
Performance Tuning for the InfiniDB Analytics Database	Provides help for tuning the InfiniDB analytic database for parallelization and scalability.

Obtaining documentation

These guides reside on our http://www.infinidb.org/ and http://www.calpont.com websites. Contact support@calpont.com for any additional assistance.

Documentation feedback

We encourage feedback, comments, and suggestions so that we can improve our documentation. Send comments to support@calpont.com along with the document name, version, comments, and page numbers.

Additional resources

If you need help installing, tuning, or querying your data with Calpont InfiniDB, you can contact support@calpont.com.

SQL Overview

This chapter provides information necessary to perform DDL and DML statements using Calpont InfiniDB. It also lists the guidelines for naming objects.

1.1 Notes

This section provides details important to running SQL statements. See "SQL Syntax" on page 17 for information about specific SQL statements.

1.2 Naming conventions

The following are the maximum lengths for MySQL objects in Calpont InfiniDB:

- ◆ User and schema names: 64 characters

- ◆ Other object names, including table and column names: 64 characters

- ◆ To ensure full compatibility with internal Calpont InfiniDB usage, the first character of all table and column names should be a letter (a-z).

Datatypes and Functions

This chapter describes the naming conventions, datatypes, and functions that are native to Calpont InfiniDB.

2.1 Datatypes

Calpont InfiniDB supports the ANSI-92 datatypes listed in the table below:

Table 1: Datatypes

Datatypes	Column Size	Description
BIGINT	8-bytes	A large integer. Numeric value with scale 0. Min/Max Signed: -9,223,372,036,854,775,806 to +9,223,372,036,854,775,807
CHAR	1, 2, 4, or 8 bytes	Holds letters and special characters of fixed length. Max length is 255. Default and minimum size is 1 byte.
DATE	4-bytes	Date has year, month, and day. The internal representation of a date is a string of 4 bytes. The first 2 bytes represent the year, .5 bytes the month, and .75 bytes the day in the following format: YYYY-MM-DD. Supported range is 1000-01-01 to 9999-12-31.
DATETIME	8-bytes	A date and time combination. Supported range is 1000-01-01 00:00:00 to 9999-12-31 23:59:59.
DECIMAL/ NUMERIC	2, 4, or 8 bytes	A packed fixed-point number that can have a specific total number of digits and with a set number of digits after a decimal. The maximum precision (total number of digits) that can be specified is 18.

Datatypes	Column Size	Description
DOUBLE/REAL	8 bytes	Stored in 64-bit IEEE-754 floating point format. As such, the number of significant digits is about 15 and the range of values is approximately +/- 1e308. The MySQL extension to specify precision and scale is not supported. "REAL" is a synonym for "DOUBLE".
FLOAT	4 bytes	Stored in 32-bit IEEE-754 floating point format. As such, the number of significant digits is about 6 and the range of values is approximately +/- 1e38. The MySQL extension to specify precision and scale is not supported.
INTEGER/INT	4-bytes	A normal-size integer. Numeric value with scale 0. Min/Max Signed: -2,147,483,646 to 2,147,483,647.
SMALLINT	2-bytes	A small integer. Min/Max Signed: -32,766 to 32,767.
TINYINT	1-byte	A very small integer. Numeric value with scale 0. Min/Max Signed: -126 to +127..
VARCHAR	1, 2, 4, or 8 bytes or 8-byte token	Holds letters, numbers, and special characters of variable length. Max length = 8000 bytes or characters and minimum length = 1 byte or character.

Notes to Datatypes:

◆ The Calpont InfiniDB engine, unlike the MyISAM engine, treats a zero-length string as a NULL value.

◆ The Calpont InfiniDB engine, like the MyISAM engine, employs "saturation semantics" on integer values. This means that if a value is inserted into an integer field that is too big/small for it to hold (i.e. it is more negative or more positive than the values indicated above), Calpont InfiniDB will "saturate" that value to the min/max value indicated above as appropriate. For example, for a SMALLINT column, if 32800 is attempted, the actual value inserted will be 32767.

◆ Calpont InfiniDB's largest negative number appears to be 2 less than what MySQL supports. Calpont InfiniDB reserves these 2 most-negative numbers for its internal use and cannot be used. For example, if there is a need to store -128 in a column, the TINYINT datatype cannot be used; the SMALLINT datatype

must be used instead. If the value -128 is inserted into a TINYINT column, Calpont InfiniDB will saturate it to -126 (and issue a warning).

◆ Calpont InfiniDB truncates rather than rounds decimal constants that have too many digits after the decimal point during bulk load and when running SELECT statements. For INSERT and UPDATE, however, the MySQL parser will round such constants. You should verify that ETL tools used and any INSERT/UPDATE statements only specify the correct number of decimal digits to avoid potential confusion.

◆ The Calpont InfiniDB engine, unlike the MyISAM engine, will error when a value is inserted for date or datetime that is out of range or otherwise illegal for the type.

◆ An optional display width may be added to the BIGINT, INTEGER/INT, SMALLINT & TINYINT columns. As with MyISAM tables, this value does not affect the internal storage requirements of the column nor does it affect the valid value ranges.

◆ All columns in InfiniDB are nullable and the default value for any column is NULL. You may optionally specify 'DEFAULT NULL' as an attribute on any column. You cannot specify any other default value.

2.2 Distributed Calpont InfiniDB Functions

2.2.1 Distributed Aggregate Functions

Calpont InfiniDB supports the following aggregate functions. These functions are currently supported only in the projection (SELECT) and ORDER BY portions of the SQL statement.

Table 2: Distributed Calpont InfiniDB Aggregate Functions

Function	Description
AVG([DISTINCT] column)	Return the average value of a number (INT variations, NUMERIC, DECIMAL) datatype column.
COUNT (*, [DISTINCT] *column*)	The number of rows returned by a query. All datatypes described above are supported.

Function	Description
MAX ([DISTINCT] *column*)	The maximum value of a column. All datatypes described above are supported.
MIN ([DISTINCT] *column*)	The minimum value of a column. All datatypes described above are supported.
SUM([DISTINCT] column)	Return the sum of *of a* number (INT variations, NUMERIC, DECIMAL) datatype column.

2.2.2 Distributed Functions

Calpont InfiniDB supports the following functions. These functions can be specified in the projection (SELECT), WHERE and ORDER BY portions of the SQL statement and will be processed in a distributed manner.

Table 3: Distributed Calpont InfiniDB Functions

Function	Description
ABS()[1]	Return the absolute value.
ACOS()[2]	Return the arc cosine.
ASIN()[2]	Return the arc sine.
ATAN()[2]	Return the arc tangent.
BETWEEN...AND...	Check whether a value is within a range of values.
CASE()	Case operator.
CEIL() CEILING()	Returns the smallest integer value not less than argument.
CONCAT()	Return concatenated string.
COS()[2]	Return the cosine.
COT()[2]	Return the cotangent.
DATEDIFF()	Return the difference between two dates.
DATE_ADD()	Add time values (intervals) to a date value.

Function	Description
DATE_FORMAT()	Format date as specified.
DATE_SUB()	Subtract two dates.
DAY()	Return the day of the month (0-31)
EXP()^2	Raise to the power of.
GREATEST()	Return the largest argument.
IF()	If/else construct.
IFNULL()	Null if/else construct.
IN	Check whether a value is within a set of values. Currently support only literal values.
INSTR()	Return the index of the first occurrence of substring.
LCASE()	Synonym for LOWER().
LEAST()	Return the smallest argument.
LEFT()	Return the leftmost number of characters as specified.
LENGTH()	Return the length of a string in bytes.
LIKE	Simple pattern matching.
LN()^2	Return the natural logarithm of the argument.
LOG()^2	Return the natural logarithm of the first argument.
LOG2()^2	Return the base-2 logarithm of the argument.
LOG10()^2	Return the base-10 logarithm of the argument
LOWER()	Return the argument in lowercase.
LPAD()	Return the string argument, left-padded with the specified string.
LOCATE()	Return the position of the first occurrence of substring.
LTRIM()	Remove leading spaces.
MINUTE()	Return the minute from the argument.
MONTH()	Return the month from the date passed.

Function	Description
POSITION()	A synonym for LOCATE().
POW() POWER()	Return the argument raised to the specified power.
REPLACE()	Replace occurrences of a specified string.
ROUND()	Round the argument.
RPAD()	Append string the specified number of times.
RTRIM()	Remove traling spaces.
SIGN()	Return the sign of the argument.
SIN()[2]	Return the sine of the argument.
SQRT()[2]	Return the square root of the argument.
SUBSTR() SUBSTRING()	Return the substring as specified.
TAN()[2]	Return the tangent of the argument.
TIME()	Return the time part of a time or datetime expression as a string
TIMEDIFF()	Return the difference between two times.
TRIM()	Remove leading and trailing spaces.
TRUNCATE()	Truncate the specified number of decimal places.
UCASE()	Synonym for UPPER().
UPPER()	Convert to uppercase.
WEEK()	Return the week number.
YEAR()	Return the year.

1. Consult the MySQL function reference for details on these functions. As these are integer functions, they only produce predictable results on integer columns.

2. Consult the MySQL function reference for details on these functions. As these are floating-point trigonometric functions, they only produce predictable results on FLOAT and DOUBLE columns.

2.3 Non-Distributed Post-Processed Functions

The following functions can be used in a post-processing manner where data is returned by Calpont InfiniDB first and then MySQL executes the function on the data returned. These functions are currently supported only in the projection (SELECT) and ORDER BY portions of the SQL statement.

Table 4: Non-Distributed Post-Processed Functions

Function	Description
CONCAT_WS()	Return concatenated with separator.
EXTRACT	Extract part of a date.
FORMAT()	Return a number formatted to specified number of decimal places.
HOUR()	Extract the hour.
LOCALTIME(), LOCALTIME	Synonym for NOW().
LOCALTIMESTAMP(), LOCALTIMESTAMP	Synonym for NOW().
NOW()	Return the current date and time.
RAND()	Return a random floating-point value.
RIGHT()	Return the specified rightmost number of characters.
SEC_TO_TIME()	Converts seconds to 'HH:MM:SS' format.
SECOND()	Return the second (0-59).
SPACE()	Return a string of the specified number of spaces.
SUBTIME()	Subtract times.
SYSDATE()	Return the time at which the function executes.

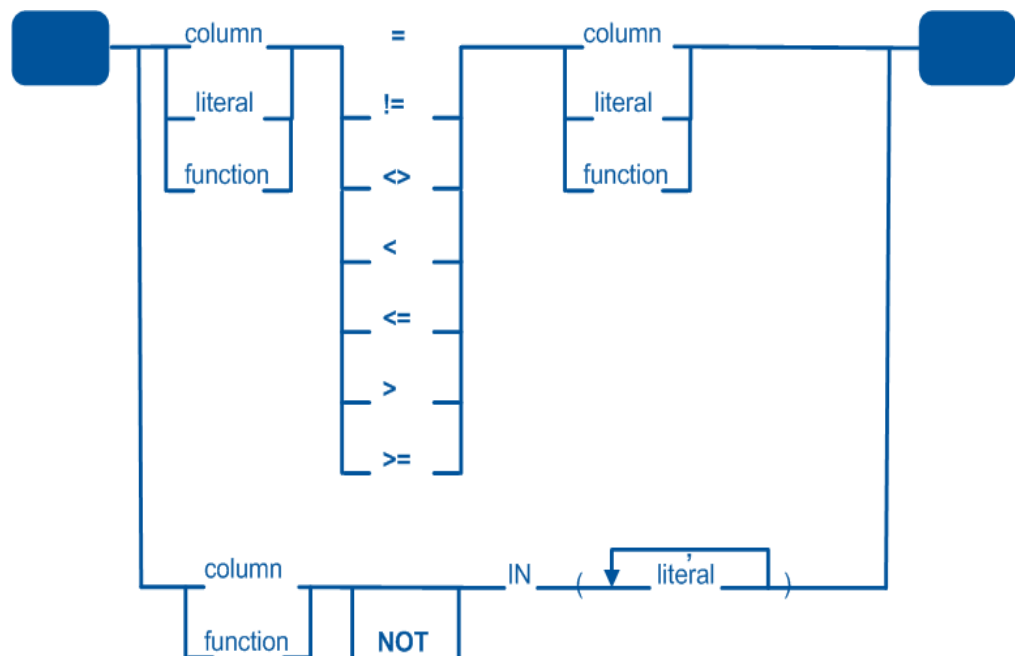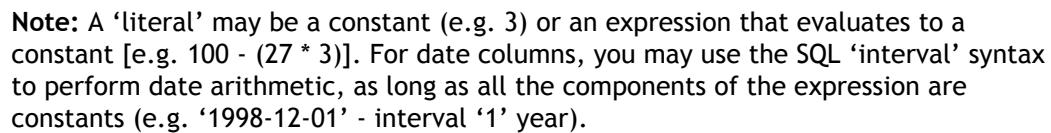| Function	Description
TIME_TO_SEC()	Return the argument converted to seconds.
TO_DAYS()	Return the date argument converted to days.
UNIX_TIMESTAMP()	Return a UNIX timestamp.
VERSION()	Returns a string that indicates the MySQL server version.
WEEKDAY()	Return the weekday number.
WEEKOFYEAR()	Return the calendar week of the date (0-53).
YEARWEEK()	Return the year and week.

Conditions

A condition is a combination of expressions and operators that return TRUE, FALSE or NULL.

The following diagram shows the conditions that can be used to return a TRUE, FALSE, or NULL condition.

3.1 filter

See the next page for additional filter conditions.

Note: A 'literal' may be a constant (e.g. 3) or an expression that evaluates to a constant [e.g. 100 - (27 * 3)]. For date columns, you may use the SQL 'interval' syntax to perform date arithmetic, as long as all the components of the expression are constants (e.g. '1998-12-01' - interval '1' year).

3.1.1 String Comparison

Calpont InfiniDB engine, unlike the MyISAM engine, is case sensitive for string comparisons used in filters.

For the most accurate results, and to avoid confusing results, make sure string filter constants are no longer than the column width itself.

3.1.2 Pattern Matching

Pattern matching as described with the LIKE condition allows you to use "_" to match any single character and "%" to match an arbitrary number of characters (including

zero characters). To test for literal instances of a wildcard character, ("%" or "_"), precede it by the "\" character.

3.1.3 OR Processing

OR Processing has the following restrictions:

◆   Only column comparisons against a literal are allowed in conjunction with an OR. The following query would be allowed since all comparisons are against literals.

> SELECT count(*) from lineitem WHERE l_partkey < 100 OR l_linestatus = 'F';

◆   Calpont InfiniDB binds AND's more tightly than OR's, just like any other SQL parser. Therefore you must enclose OR-relations in parentheses, just like in any other SQL parser.

> SELECT count(*) FROM orders, lineitem WHERE (lineitem.l_orderkey < 100 OR lineitem.l_linenumber > 10) AND lineitem.l_orderkey = orders.o_orderkey;
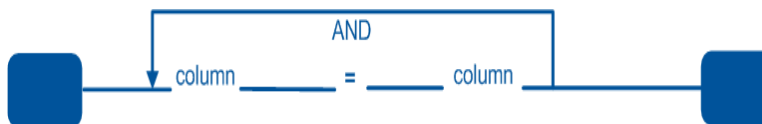
3.2 table_filter

The following diagram show the conditions you can use when executing a condition against two columns. Note that the columns must be from the same table.
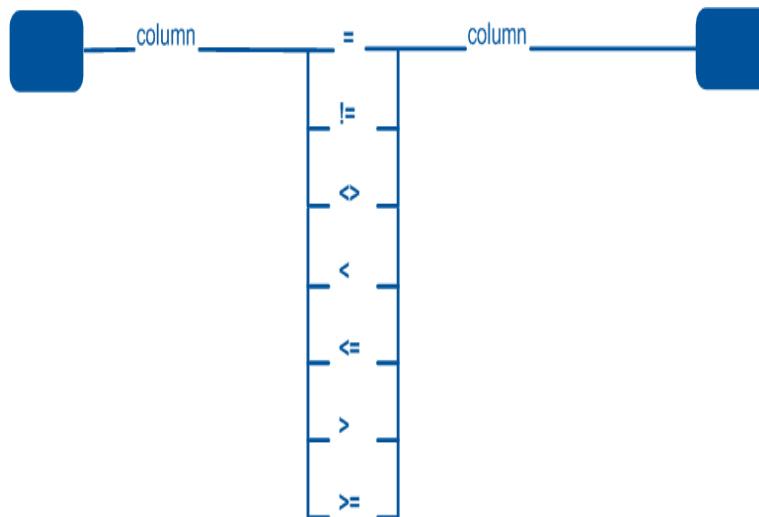
3.3 join

The following diagram show the conditions you can use when executing a join on two tables.

Notes for Joins:

◆ Calpont InfiniDB tables can only be joined with non-Calpont InfiniDB tables in table mode only. See Chapter 5 - Operating Mode for information.

◆ Calpont InfiniDB will require a join in the WHERE clause for each set of tables in the FROM clause. No cartesian product queries will be allowed.

◆ Calpont InfiniDB requires that joins must be on the same datatype. In addition, number datatypes (INT variations, NUMERIC, DECIMAL) may be mixed in the join if they have the same scale.

◆ The MySQL optimizer has some behaviors that are incompatible with Calpont InfiniDB. If syntax errors are received from Calpont InfiniDB, some of these incompatibilities can be mitigated by specifying 'straight_join' between each join table.

◆ Circular joins are not supported in Calpont InfiniDB. Please see the Troubleshooting section in the Calpont InfiniDB Administrator's Guide for further reference.

Chapter 4

SQL Syntax

Calpont InfiniDB is a high performance SQL engine that supports SQL Syntax Statements. This chapter provides the syntax that must be adhered to when performing INSERT, UPDATE, or DELETE operations.

This guide lists the SELECT syntax that is native to Calpont InfiniDB and provides exceptionally fast query executions.

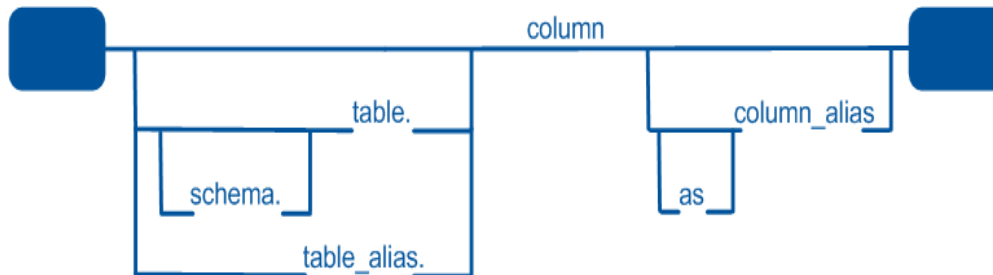## 4.1 Table/Column Reference

4.1.1 table

The following chart describes the guidelines when referencing a table in SQL statements.

NOTE: in order to query into a non-Calpont InfiniDB schema, the user must change to that schema.

4.1.2 column

The following chart describes the guidelines when referencing a column in SQL statements.

4.2 DDL Statements

DDL statements define database structures. These structures include columns, and tables. Supported DDL statements are listed below in alphabetical order with descriptions and sample syntax statements.

4.2.1 ALTER TABLE

The ALTER TABLE statement modifies existing tables. This includes adding, deleting and renaming columns as well as renaming tables.
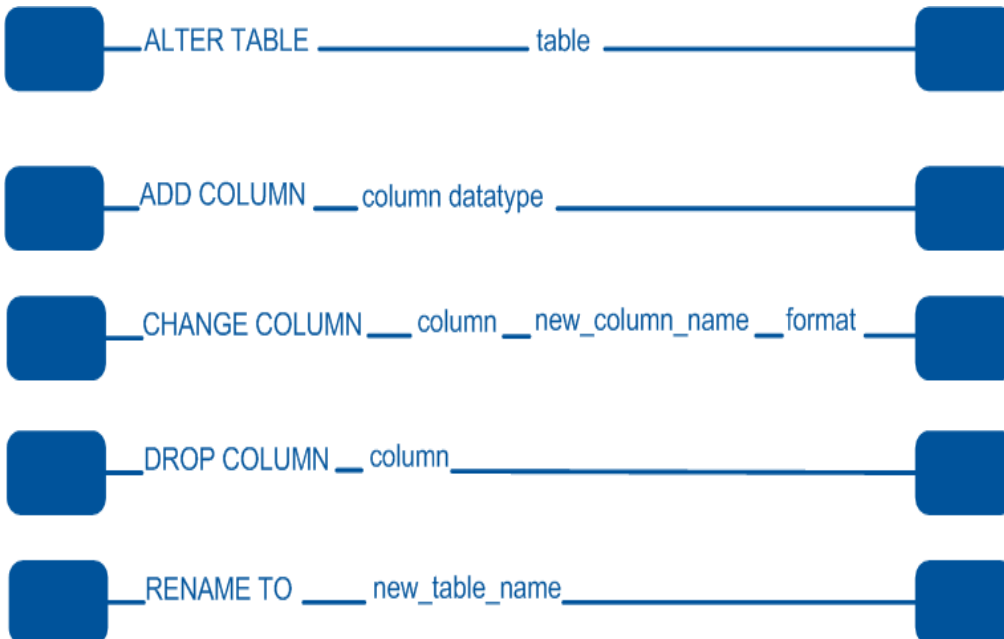
4.2.1.1 ADD

The ADD clause allows you to add columns to a table. You must specify the data type after the column name.

The following statement adds a **priority** column with an **integer** datatype to the **orders** table:

```
ALTER TABLE orders ADD COLUMN priority INTEGER;
```

4.2.1.1.1 Online alter table add column

The InfiniDB engine fully supports online DDL (one session can be adding columns to a table while another session is querying that table). MySQL, unfortunately, does not support this. Until this is fixed in the MySQL server, we have provided the following workaround. This workaround is intended for adding columns to a table, one at a time only. Do not attempt to use it for any other purpose. Follow the example below as closely as possible.

Scenario: add an INT column named col7 to the existing table foo:

select calonlinealter('alter table foo add column col7 int;');
alter table foo add column col7 int comment 'schema sync only';

The select statement may take several tens of seconds to run, depending on how many rows are currently in the table. Regardless, other sessions can select against the table during this time (but they won't be able to see the new column yet). The alter table statement will take less than 1 second (depending on how busy MySQL is) and during this brief time interval, other table reads will be held off.

4.2.1.2 CHANGE

The CHANGE clause allows you to rename a column in a table.

Notes to CHANGE COLUMN:
- ◆ You cannot currently use CHANGE COLUMN to change the definition of that column.
- ◆ You can only change a single column at a time.

The following example renames the **order_qty** field to **quantity** in the **orders** table:

```
ALTER TABLE orders CHANGE COLUMN order_qty quantity
INTEGER;
```

4.2.1.3 DROP

The DROP clause allows you to drop columns. All associated data is removed when the column is dropped. You can DROP COLUMN (column_name).

The following example alters the **orders** table to drop the **priority** column:

```
ALTER TABLE orders DROP COLUMN priority;
```

4.2.1.4 RENAME

The RENAME clause allows you to rename a table.

The following example renames the **orders** table:

```
ALTER TABLE orders RENAME TO customer_orders;
```

4.2.2 COMMIT

The COMMIT statement makes changes to a table permanent. You should only commit changes after you have verified the integrity of the changed data.

Once data is committed, it cannot be undone with the ROLLBACK statement. To return the database to its former state, you must restore the data from backups. See "ROLL-BACK" on page 23.

4.2.3 CREATE PROCEDURE

Creates a stored routine in the Calpont InfiniDB database.

NOTE: Calpont InfiniDB currently accepts definition of stored procedures with only input arguments and a single query while in Operating Mode = 1 (VTABLE mode). How-

ever, while in the Operating Mode = 0 (TABLE mode), Calpont InfiniDB will allow additional complex definition of stored procedures (i.e., OUT parameter, declare, cursors, etc.) See Chapter 5 - Operating Mode for information on Operating Modes.

The following statements create and call the sp_complex_variable stored procedure:

```
CREATE PROCEDURE sp_complex_variable(in arg_key int, in arg_date date)
    begin
        Select *
        from lineitem, orders
        where o_custkey < arg_key
        and    l_partkey < 10000
        and    l_shipdate>arg_date
        and    l_orderkey = o_orderkey
        order by l_orderkey, l_linenumber;
    end

    call sp_complex_variable(1000, '1998-10-10');
```

4.2.4 CREATE TABLE

A database consists of tables that store user data. You can create multiple columns in the create table statement. The datatype follows the column name when adding columns.

Notes to CREATE TABLE:

◆ Calpont InfiniDB tables should not be created in the mysql, information_schema or test databases.

◆ For maximum compatibility with external tools InfiniDB will accept the following table attributes:

MIN_ROWS
MAX_ROWS
AUTO_INCREMENT
DEFAULT CHARSET
COMMENT

All of these are ignored by InfiniDB.

The following statement creates a table called **orders** with two columns: **orderkey** with datatype **integer** and **customer** with datatype **varchar**:

```
CREATE TABLE orders (orderkey INTEGER, customer VAR-
CHAR(45)) ENGINE=INFINIDB;
```

4.2.5 DROP PROCEDURE

The DROP PROCEDURE statement deletes a stored procedure from the Calpont InfiniDB database.
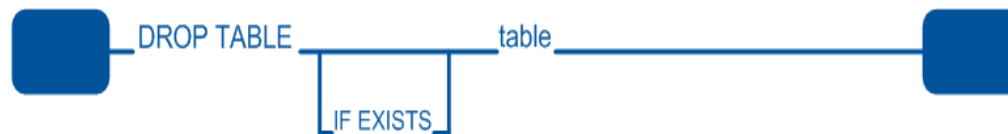

The following statement drops the **sp_complex_variable** procedure:

```
DROP PROCEDURE sp_complex_variable;
```

4.2.6 DROP TABLE

The DROP TABLE statement deletes a table from the Calpont InfiniDB database.

The following statement drops the **orders** table:

```
DROP TABLE orders;
```

4.2.7 RENAME TABLE

The RENAME TABLE statement renames one or more tables in the Calpont InfiniDB database.

Notes to RENAME TABLE:

◆ You cannot currently use RENAME TABLE to move a table from one database to another.

◆ See the ALTER TABLE syntax for alternate way to RENAME table.

The following statement renames the **orders** table:

```
RENAME TABLE orders TO customer_order;
```

The following statement renames both the **orders** table and **customer** table:

```
RENAME TABLE orders TO customer_orders,
             customer TO customers;
```

You may also use RENAME TABLE to swap tables. This example swaps the **customer** and **vendor** tables (assuming the **temp_table** does not already exist):

```
RENAME TABLE customer TO temp_table,
             vendor TO customer,
             temp_table to vendor;
```

4.2.8 ROLLBACK

The ROLLBACK statement undoes transactions that have not been permanently saved to the database with the COMMIT statement.

You cannot rollback changes to table properties including ALTER, CREATE, or DROP TABLE statements.

ROLLBACK

4.3 DML Statements

DML statements are used for manipulating data in tables. This includes deleting, inserting, and updating data. DML statements that are native to Calpont InfiniDB are listed below in alphabetical order with descriptions and sample syntax statements.

4.3.1 DELETE

The DELETE statement is used to remove rows from tables.

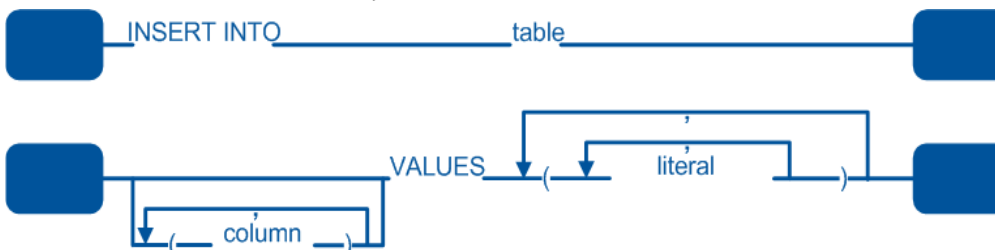DELETE _____ table _____
FROM

WHERE simple_filter

The following statement deletes **customer** records with a customer key identification between **1001** and **1999**:

```
DELETE FROM customer WHERE custkey > 1000 and custkey <
2000;
```

4.3.2 INSERT

The INSERT statement allows you to add data to tables.

INSERT INTO _____ table _____

VALUES (literal)

(column)

The following statement inserts a row with all column values into the **customer** table:

```
INSERT INTO customer (custno, custname, custaddress,
phoneno, cardnumber, comments) VALUES (12, 'John
Smith', '100 First Street, Dallas', '(214) 555-1212',
100, 'On Time');
```

The following statement inserts two rows with all column values into the **customer** table:

```
INSERT INTO customer (custno, custname, custaddress,
phoneno, cardnumber, comments) VALUES (12, 'John
Smith', '100 First Street, Dallas', '(214) 555-1212',
100, 'On Time'), (13, 'John Q Public', '200 Second
Street, Dallas', '(972) 555-1234', 200, 'Late
Payment');
```

4.3.3 LOAD DATA INFILE

The LOAD DATA INFILE statement reads rows from a text file into a table at a very high speed. The file name must be given as a literal string.

```
LOAD DATA INFILE 'file_name'
   INTO TABLE tbl_name
   [CHARACTER SET charset_name]
   [{FIELDS | COLUMNS}
      [TERMINATED BY 'string']
      [[OPTIONALLY] ENCLOSED BY 'char']
      [ESCAPED BY 'char']
   ]
   [LINES
      [STARTING BY 'string']
      [TERMINATED BY 'string']
   ]
```

The following example loads data into the a simple 5 column table:

A file named simpletable.tbl has the following data in it.
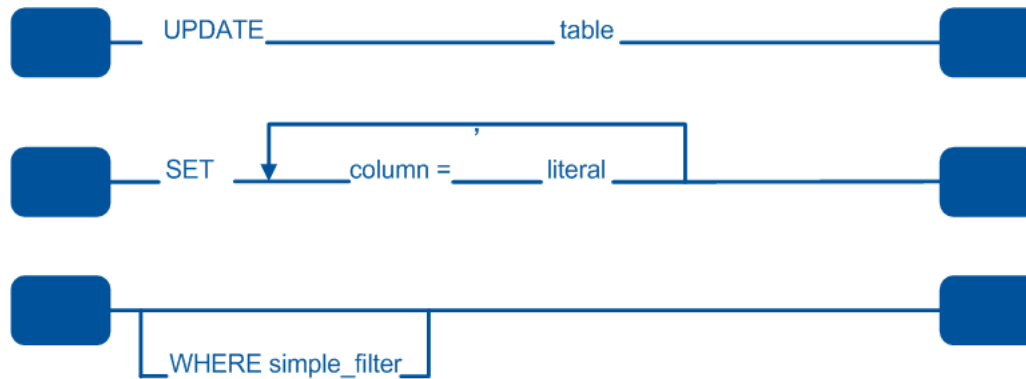
    1|100|1000|10000|Test Number 1|

```
2|200|2000|20000|Test Number 2|
3|300|3000|30000|Test Number 3|
```

The data can then be loaded into the simpletable table with the following syntax:

```
LOAD DATA INFILE 'simpletable.tbl' INTO TABLE simpletable
  FIELDS TERMINATED BY '|';
```

4.3.4 UPDATE

The UPDATE statement changes data stored in rows.

The following statement updates the WidgetFactory supplier name in the **supplier** table to WidgetsInc:

```
UPDATE supplier SET name = 'WidgetsInc.' WHERE name =
'WidgetFactory';
```

4.4 SELECT

The SELECT statement is used to query the database and display table data. You can add many clauses to filter the data.
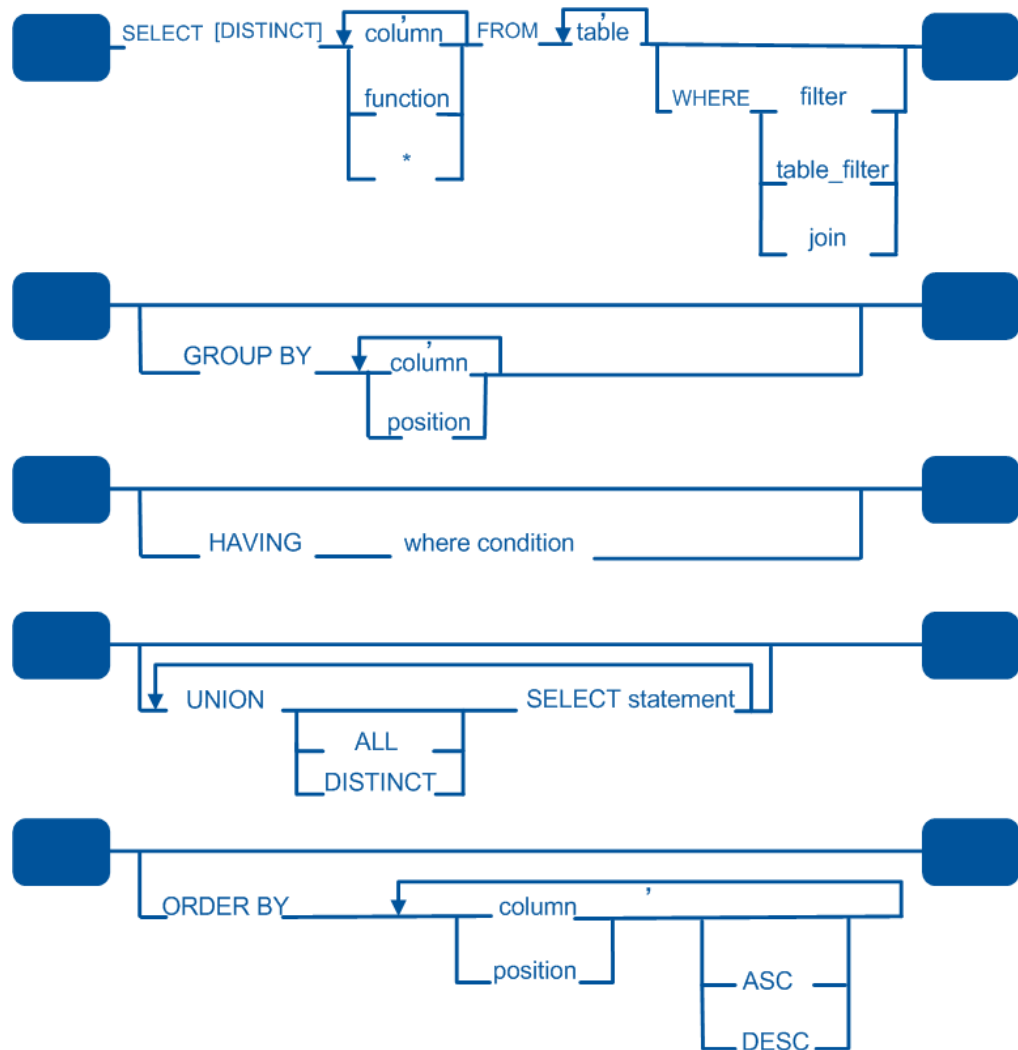
4.4.1 Projection List (SELECT) Notes

If the same column needs to be referenced more than once in the projection list, a unique name is required for each column using a column alias.

The total length of the name of a column, inclusive of length of functions, in

the projection list must be 64 characters or less.

4.4.2 WHERE

The WHERE clause filters data retrieval based on criteria. **NOTE:** column_alias cannot be used in the WHERE clause.

The following statement returns rows in the region table where the **region** = 'ASIA':

```
SELECT * FROM region WHERE name = 'ASIA';
```

4.4.3 GROUP BY

GROUP BY groups data based on values in one or more specific columns. A maximum of 10 columns will be supported in the GROUP BY clause.

The following statement returns rows from the **lineitem** table where **orderkey** is less than 1000000 and groups them by the quantity.

```
SELECT quantity, count(*) FROM lineitem WHERE orderkey
< 1000000 GROUP BY quantity;
```

4.4.4 HAVING

HAVING is used in combination with the GROUP BY clause. It can be used in a SELECT statement to filter the records that a GROUP BY returns.

The following statement returns shipping dates, and the respective quantity where the quantity is 2500 or more.

```
SELECT shipdate, count(*) FROM lineitem GROUP BY
shipdate HAVING count(*) >= 2500;
```

4.4.5 ORDER BY

The ORDER BY clause presents results in a specific order.

Note: The ORDER BY clause represents a statement that is post processed by MySQL.

The following statement returns an ordered **quantity** column from the **lineitem** table.

```
SELECT quantity FROM lineitem WHERE orderkey < 1000000
order by quantity;
```

The following statement returns an ordered **shipmode** column from the **lineitem** table.

```
Select shipmode from lineitem where orderkey < 1000000
order by 1;
```

4.4.6 UNION

Used to combine the result from multiple SELECT statements into a single result set.

The UNION or UNION DISTINCT clause returns query results from multiple queries into one display and discards duplicate results. The UNION ALL clause displays query results from multiple queries and does not discard the duplicates.

The following statement returns the p_name rows in the part table and the partno table and discards the duplicate results:

```
SELECT p_name FROM part UNION select p_name FROM
partno;
```

The following statement returns **all** the p_name rows in the part table and the partno table:

```
SELECT p_name FROM part UNION ALL select p_name FROM
partno;
```

4.5 Prepared Statements

Prepared statements are used to send SQL statements to the server to be executed. SQL syntax for prepared statements consists of 4 SQL statements: PREPARE, SET, EXECUTE and DEALLOCATE/DROP PREPARE.
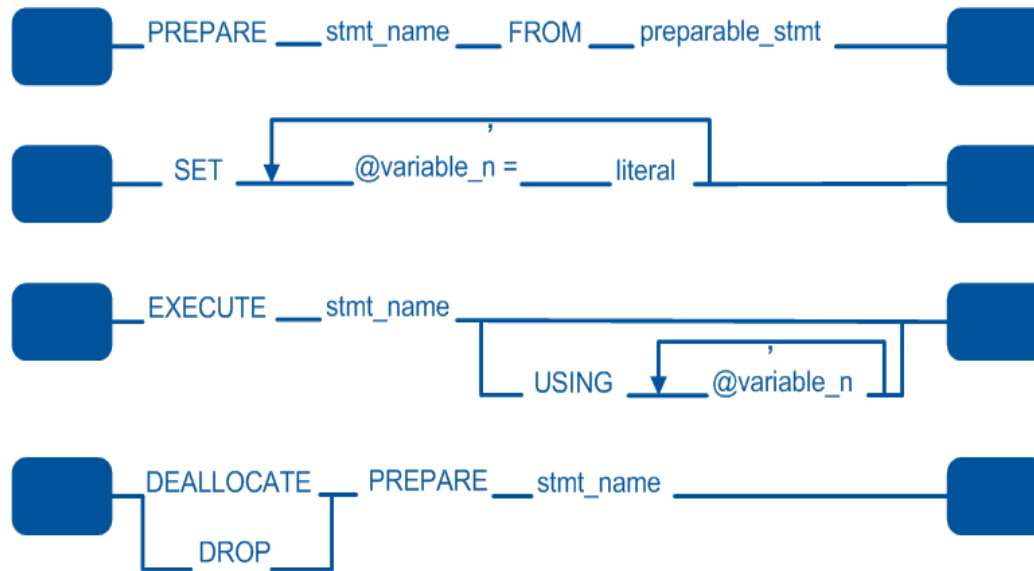
4.5.1 PREPARE

The PREPARE statement prepares a statement and assigns it a name, stmt_name, by which to refer to the statement later. The text must represent a single SQL statement. Within the statement, "?" characters can be used as parameters to indicate where data values are to be bound to the query later when you execute it. The "?" characters should not be enclosed within quotes, even if you intend to bind them to string values.

```
PREPARE test1 FROM
    "SELECT
     SUM(L_EXTENDEDPRICE*L_DISCOUNT) AS REVENUE
    FROM
     LINEITEM
    WHERE
     L_SHIPDATE >= ? AND
     L_SHIPDATE < ? + interval '1' year AND
     L_DISCOUNT BETWEEN ? - 0.01 AND ? + 0.01 AND
     L_QUANTITY < ?;";
```

4.5.2 SET

The SET statement initializes any variables used in the PREPARE statement.

```
set @v1=date '1994-01-01';
set @v2=0.06;
set @v3=24;
```

4.5.3 EXECUTE

After preparing a statement with PREPARE, you execute it with an EXECUTE statement that refers to the prepared statement name. If the prepared statement contains any parameters, you must supply a USING clause that lists user variables containing the values to be bound to the parameters. Parameter values can be supplied only by user variables, and the USING clause must name exactly as many variables as the number of parameter markers in the statement.

You can execute a given prepared statement multiple times. Use the SET statement to change any parameter values before each execution, otherwise the previously set parameter values will be used.

```
execute test1 using @v1, @v1, @v2, @v2, @v3;
```

4.5.4 DEALLOCATE/DROP PREPARE

Removes the prepared statement.

```
DROP PREPARE test1;
```

Operating Mode

Calpont InfiniDB has the ability to support full MySQL query syntax through an operating mode. This operating mode may be set as a default for the instance or set at the session level.

Please refer to the Calpont InfiniDB Administrator's Guide for setting the default operating mode.

To set the operating mode at the session level, the following command is used. Once the session has ended, any subsequent session will return to the default for the instance.

set infinidb_vtable_mode = n

where n is:

◆   0) a generic, highly compatible row-by-row processing mode. Some WHERE clause components can be processed by InfiniDB, but joins are processed entirely by mysqld using a nested-loop join mechanism

◆   1) (the default) query syntax is evaluated by InfiniDB for compatibility with distributed execution and incompatible queries are rejected. Queries executed in this mode take advantage of distributed execution and typically result in higher performance.

◆   2) auto-switch mode: InfiniDB will attempt to process the query internally, if it cannot, it will automatically switch the query to run in row-by-row mode.

NOTE: For more information on supported query syntax for modes 0 and 2, please refer to the MySQL 5.1 SQL Syntax Guide.

Decimal Scale

Calpont InfiniDB has the ability to support varied internal precision on decimal calculations.

infinidb_decimal_scale is used internally by the InfiniDB engine to control how many significant digits to the right of the decimal point are carried through in suboperations on calculated columns. If, while running a query, you receive the message 'aggregate overflow', try reducing infinidb_decimal_scale and running the query again. Note that, as you decrease infinidb_decimal_scale, you may see reduced accuracy in the least significant digit(s) of a returned calculated column.

infinidb_use_decimal_scale is used internally by the InfiniDB engine to turn on and off the use if this internal precision.

These to system variables may be set as a default for the instance or set at the session level.

Please refer to the Calpont InfiniDB Administrator's Guide for setting the default decimal scale.

6.1 Enable/Disable Decimal Scale

To enable/disable the use of the decimal scale at the session level, the following command is used. Once the session has ended, any subsequent session will return to the default for the instance.

set infinidb_use_decimal_scale = n

where *value* is 0 (disabled) or 1 (enabled).

6.2 Set Decimal Scale Level

To set the decimal scale at the session level, the following command is used. Once the session has ended, any subsequent session will return to the default for the instance.

set infinidb_decimal_scale = n

where *value* is the amount of precision desired for calculations.

GNU Free Documentation License

GNU Free Documentation License
Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is

addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by

proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all

these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS
You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- ◆ A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- ◆ B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at

least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

◆ C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

◆ D. Preserve all the copyright notices of the Document.

◆ E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

◆ F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

◆ G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

◆ H. Include an unaltered copy of this License.

◆ I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

◆ J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

◆ K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

◆ L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

◆ M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

◆ N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

◆ O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such

a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

W

WHERE 28