MySQL Restrictions and Limitations
Abstract

This is the MySQL Restrictions and Limitations extract from the MySQL 5.1 Reference Manual.

For legal information, see the Legal Notices.

For help with using MySQL, please visit either the MySQL Forums or MySQL Mailing Lists, where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Document generated on: 2016-01-13 (revision: 46080)
## Table of Contents

Preface and Legal Notices ........................................................................................................... v  
1 Restrictions and Limits ............................................................................................................ 1  
2 Restrictions on Stored Programs ............................................................................................ 3  
3 Restrictions on Server-Side Cursors ....................................................................................... 9  
4 Restrictions on Subqueries .................................................................................................... 11  
5 Restrictions on Views ............................................................................................................. 15  
6 Restrictions on XA Transactions ............................................................................................ 19  
7 Restrictions on Character Sets .............................................................................................. 21  
8 Limits in MySQL ..................................................................................................................... 23  
  8.1 Limits on Joins ................................................................................................................... 23  
  8.2 Limits on Number of Databases and Tables ...................................................................... 23  
  8.3 Limits on Table Size .......................................................................................................... 23  
  8.4 Limits on Table Column Count and Row Size ................................................................. 25  
  8.5 Limits Imposed by .frm File Structure ............................................................................. 26  
  8.6 Windows Platform Limitations ......................................................................................... 27  
9 Restrictions and Limitations on Partitioning ......................................................................... 31  
  9.1 Partitioning Keys, Primary Keys, and Unique Keys ............................................................. 37  
  9.2 Partitioning Limitations Relating to Storage Engines .......................................................... 40  
  9.3 Partitioning Limitations Relating to Functions .................................................................... 42  
  9.4 Partitioning and Table-Level Locking .............................................................................. 42  
10 MySQL Differences from Standard SQL ................................................................................ 45  
  10.1 SELECT INTO TABLE ....................................................................................................... 45  
  10.2 UPDATE ......................................................................................................................... 45  
  10.3 Transactions and Atomic Operations ............................................................................. 45  
  10.4 Foreign Key Differences .................................................................................................. 48  
  10.5 ‘*’ as the Start of a Comment ......................................................................................... 48  
11 Known Issues in MySQL ......................................................................................................... 51  
A Licenses for Third-Party Components .................................................................................. 55  
  A.1 ANTLR 3 License .............................................................................................................. 57  
  A.2 dtoa.c License .................................................................................................................. 57  
  A.3 Editline Library (libedit) License ....................................................................................... 58  
  A.4 FindGTest.cmake License ............................................................................................... 60  
  A.5 Fred Fish’s Dbg Library License ....................................................................................... 61  
  A.6 getarg License .................................................................................................................. 62  
  A.7 GNU General Public License Version 2.0, June 1991 ......................................................... 62  
  A.8 GNU Lesser General Public License Version 2.1, February 1999 ...................................... 68  
  A.9 GNU Libtool License ....................................................................................................... 76  
  A.10 GNU Readline License .................................................................................................... 76  
  A.11 Google Controlling Master Thread I/O Rate Patch License ............................................... 77  
  A.12 Google PerfTools (TCMalloc utility) License .................................................................. 77  
  A.13 Google SMP Patch License ............................................................................................. 78  
  A.14 lib_sql.cc License .......................................................................................................... 79  
  A.15 libevent License .............................................................................................................. 79  
  A.16 Linux-PAM License ......................................................................................................... 81  
  A.17 md5 (Message-Digest Algorithm 5) License ................................................................. 82  
  A.18 memcached License ........................................................................................................ 82  
  A.19 nt_servc (Windows NT Service Class library) License ................................................... 83  
  A.20 OpenPAM License ......................................................................................................... 83  
  A.21 Paramiko License .......................................................................................................... 83  
  A.22 Percona Multiple I/O Threads Patch License .................................................................. 84  
  A.23 RegEx-Spencer Library License ...................................................................................... 84
A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License ........................................ 85
A.25 Richard A. O'Keefe String Library License ........................................................... 85
A.26 SHA-1 in C License ...................................................................................................... 86
A.27 zlib License .................................................................................................................. 86
Preface and Legal Notices

This is the MySQL Restrictions and Limitations extract from the MySQL 5.1 Reference Manual.

Legal Notices

Copyright © 1997, 2016, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Documentation Accessibility

For information about Oracle’s commitment to accessibility, visit the Oracle Accessibility Program website at

v
Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.
Chapter 1 Restrictions and Limits

The discussion here describes restrictions that apply to the use of MySQL features such as subqueries or views.
Chapter 2 Restrictions on Stored Programs

These restrictions apply to the features described in Stored Programs and Views.

Some of the restrictions noted here apply to all stored routines; that is, both to stored procedures and stored functions. There are also some restrictions specific to stored functions but not to stored procedures. The restrictions for stored functions also apply to triggers. There are also some restrictions specific to triggers.

The restrictions for stored procedures also apply to the DO clause of Event Scheduler event definitions. There are also some restrictions specific to events.

SQL Statements Not Permitted in Stored Routines

Stored routines cannot contain arbitrary SQL statements. The following statements are not permitted:

• The locking statements LOCK TABLES and UNLOCK TABLES.
• ALTER VIEW. (Before MySQL 5.1.21, this restriction is enforced only for stored functions.)
• LOAD DATA and LOAD TABLE.
• SQL prepared statements (PREPARE, EXECUTE, DEALLOCATE PREPARE) can be used in stored procedures, but not stored functions or triggers. Thus, stored functions and triggers cannot use dynamic SQL (where you construct statements as strings and then execute them).
• Generally, statements not permitted in SQL prepared statements are also not permitted in stored programs. For a list of statements supported as prepared statements, see SQL Syntax for Prepared Statements.
• Because local variables are in scope only during stored program execution, references to them are not permitted in prepared statements created within a stored program. Prepared statement scope is the current session, not the stored program, so the statement could be executed after the program ends, at which point the variables would no longer be in scope. For example, SELECT ... INTO local_var cannot be used as a prepared statement. This restriction also applies to stored procedure and function parameters. See PREPARE Syntax.
• Inserts cannot be delayed. INSERT DELAYED syntax is accepted, but the statement is handled as a normal INSERT.
• Within all stored programs (stored procedures and functions, triggers, and events), the parser treats BEGIN [WORK] as the beginning of a BEGIN ... END block. To begin a transaction in this context, use START TRANSACTION instead.

Restrictions for Stored Functions

The following additional statements or operations are not permitted within stored functions. They are permitted within stored procedures, except stored procedures that are invoked from within a stored function or trigger. For example, if you use FLUSH in a stored procedure, that stored procedure cannot be called from a stored function or trigger.

• Statements that perform explicit or implicit commit or rollback. Support for these statements is not required by the SQL standard, which states that each DBMS vendor may decide whether to permit them.
• Statements that return a result set. This includes SELECT statements that do not have an INTO var_list clause and other statements such as SHOW, EXPLAIN, and CHECK TABLE. A function
Restrictions for Triggers

can process a result set either with `SELECT ... INTO var_list` or by using a cursor and `FETCH` statements. See `SELECT ... INTO` Syntax, and Cursors.

- `FLUSH` statements.
- Stored functions cannot be used recursively.
- A stored function or trigger cannot modify a table that is already being used (for reading or writing) by the statement that invoked the function or trigger.
- If you refer to a temporary table multiple times in a stored function under different aliases, a `Can't reopen table: 'tbl_name'` error occurs, even if the references occur in different statements within the function.

Restrictions for Triggers

For triggers, the following additional restrictions apply:

- Triggers are not activated by foreign key actions.
- When using row-based replication, triggers on the slave are not activated by statements originating on the master. The triggers on the slave are activated when using statement-based replication. For more information, see Replication and Triggers.
- The `RETURN` statement is not permitted in triggers, which cannot return a value. To exit a trigger immediately, use the `LEAVE` statement.
- Triggers are not permitted on tables in the `mysql` database.
- The trigger cache does not detect when metadata of the underlying objects has changed. If a trigger uses a table and the table has changed since the trigger was loaded into the cache, the trigger operates using the outdated metadata.

Restrictions on Calling Stored Procedures

Before MySQL 5.1.4, `CALL` statements cannot be prepared, meaning that stored procedures cannot be called from dynamic SQL. This is true both for server-side prepared statements and for SQL prepared statements.

Name Conflicts within Stored Routines

The same identifier might be used for a routine parameter, a local variable, and a table column. Also, the same local variable name can be used in nested blocks. For example:

```
cREATE PROCEDURE p (i INT)
BEGIN
    DECLARE i INT DEFAULT 0;
    SELECT i FROM t;
    BEGIN
        DECLARE i INT DEFAULT 1;
        SELECT i FROM t;
        END;
END;
```

In such cases, the identifier is ambiguous and the following precedence rules apply:

- A local variable takes precedence over a routine parameter or table column.
• A routine parameter takes precedence over a table column.

• A local variable in an inner block takes precedence over a local variable in an outer block.

The behavior that variables take precedence over table columns is nonstandard.

Replication Considerations

Use of stored routines can cause replication problems. This issue is discussed further in Binary Logging of Stored Programs.

The `--replicate-wild-do-table=db_name.tbl_name` option applies to tables, views, and triggers. It does not apply to stored procedures and functions, or events. To filter statements operating on the latter objects, use one or more of the `--replicate-*/-db` options.

Introspection Considerations

`INFORMATION_SCHEMA` does not have a `PARAMETERS` table until MySQL 5.5; this table enables stored routines or client applications to determine the names, types, and default values of parameters for stored routines. For releases without this `INFORMATION_SCHEMA.PARAMETERS` table, to examine these types of metadata, you must use workarounds such as parsing the output of `SHOW CREATE` statements or the `param_list` column of the `mysql.proc` table. `param_list` contents can be processed from within a stored routine, unlike the output from `SHOW`.

Debugging Considerations

There are no stored routine debugging facilities.

Unsupported Syntax from the SQL:2003 Standard

The MySQL stored routine syntax is based on the SQL:2003 standard. The following items from that standard are not currently supported:

• `UNDO` handlers

• `FOR` loops

Concurrency Considerations

To prevent problems of interaction between sessions, when a client issues a statement, the server uses a snapshot of routines and triggers available for execution of the statement. That is, the server calculates a list of procedures, functions, and triggers that may be used during execution of the statement, loads them, and then proceeds to execute the statement. While the statement executes, it does not see changes to routines performed by other sessions.

For maximum concurrency, stored functions should minimize their side-effects; in particular, updating a table within a stored function can reduce concurrent operations on that table. A stored function acquires table locks before executing, to avoid inconsistency in the binary log due to mismatch of the order in which statements execute and when they appear in the log. When statement-based binary logging is used, statements that invoke a function are recorded rather than the statements executed within the function. Consequently, stored functions that update the same underlying tables do not execute in parallel. In contrast, stored procedures do not acquire table-level locks. All statements executed within stored procedures are written to the binary log, even for statement-based binary logging. See Binary Logging of Stored Programs.
Event Scheduler Restrictions

The following limitations are specific to the Event Scheduler:

• In MySQL 5.1.6 only, any table referenced in an event's action statement must be fully qualified with the
  name of the schema in which it occurs (that is, as schema_name.table_name).

• Beginning with MySQL 5.1.8, event names are handled in case-insensitive fashion. For example, this
  means that you cannot have two events in the same database (and—prior to MySQL 5.1.12—with the
  same definer) with the names anEvent and AnEvent.

  Important

  If you have events created in MySQL 5.1.7 or earlier which are assigned to
  the same database and have the same definer, and whose names differ only
  with respect to lettercase, then you must rename these events to respect case-
  sensitive handling before upgrading to MySQL 5.1.8 or later.

• An event may not be created, altered, or dropped by a stored routine, trigger, or another event. An event
  also may not create, alter, or drop stored routines or triggers. (Bug #16409, Bug #18896)

• Event timings using the intervals YEAR, QUARTER, MONTH, and YEAR_MONTH are resolved in months;
  those using any other interval are resolved in seconds. There is no way to cause events scheduled
  to occur at the same second to execute in a given order. In addition—due to rounding, the nature of
  threaded applications, and the fact that a nonzero length of time is required to create events and to
  signal their execution—events may be delayed by as much as 1 or 2 seconds. However, the time shown
  in the INFORMATION_SCHEMA.EVENTS table's LAST_EXECUTED column or the mysql.event table's
  last_executed column is always accurate to within one second of the actual event execution time.
  (See also Bug #16522.)

• Each execution of the statements contained in the body of an event takes place in a new connection;
  thus, these statements has no effect in a given user session on the server's statement counts such
  as Com_select and Com_insert that are displayed by SHOW STATUS. However, such counts are
  updated in the global scope. (Bug #16422)

• Prior to MySQL 5.1.12, you could not view another user's events in the
  INFORMATION_SCHEMA.EVENTS table. In other words, any query made against this table was treated
  as though it contained the condition DEFINER = CURRENT_USER() in the WHERE clause.

• Events do not support times later than the end of the Unix Epoch; this is approximately the beginning of
  the year 2038. Prior to MySQL 5.1.8, handling in scheduled events of dates later than this was buggy;
  starting with MySQL 5.1.8, such dates are specifically not permitted by the Event Scheduler. (Bug
  #16396)

• In MySQL 5.1.6, INFORMATION_SCHEMA.EVENTS shows NULL in the SQL_MODE column. Beginning
  with MySQL 5.1.7, the SQL_MODE displayed is that in effect when the event was created.

• In MySQL 5.1.6, the only way to drop or alter an event created by a user who was not the definer of that
  event was by manipulation of the mysql.event system table by the MySQL root user or by another
  user with privileges on this table. Beginning with MySQL 5.1.7, DROP USER drops all events for which
  that user was the definer; also beginning with MySQL 5.1.7 DROP SCHEMA drops all events associated
  with the dropped schema.

• References to stored functions, user-defined functions, and tables in the ON SCHEDULE clauses of
  CREATE EVENT and ALTER EVENT statements are not supported. Beginning with MySQL 5.1.13, these
  sorts of references are not permitted. (See Bug #22830 for more information.)
• Generally speaking, statements that are not permitted in stored routines or in SQL prepared statements are also not permitted in the body of an event. For more information, see SQL Syntax for Prepared Statements.

• When upgrading to MySQL 5.1.18 or 5.1.19 from a previous MySQL version where scheduled events were in use, the upgrade utilities `mysql_upgrade` and `mysql_fix_privilege_tables` do not accommodate changes in system tables relating to the Event Scheduler. This issue was fixed in MySQL 5.1.20 (see Bug #28521).

**Stored routines and triggers in MySQL Cluster.** Stored procedures, stored functions, and triggers are all supported by tables using the NDB storage engine; however, it is important to keep in mind that they do not propagate automatically between MySQL Servers acting as Cluster SQL nodes. This is because of the following:

• Stored routine definitions are kept in tables in the `mysql` system database using the MyISAM storage engine, and so do not participate in clustering.

• The `.TRN` and `.TRG` files containing trigger definitions are not read by the NDB storage engine, and are not copied between Cluster nodes.

Any stored routine or trigger that interacts with MySQL Cluster tables must be re-created by running the appropriate `CREATE PROCEDURE`, `CREATE FUNCTION`, or `CREATE TRIGGER` statements on each MySQL Server that participates in the cluster where you wish to use the stored routine or trigger. Similarly, any changes to existing stored routines or triggers must be carried out explicitly on all Cluster SQL nodes, using the appropriate `ALTER` or `DROP` statements on each MySQL Server accessing the cluster.

**Warning**

Do not attempt to work around the issue described in the first item mentioned previously by converting any `mysql` database tables to use the NDB storage engine. **Altering the system tables in the mysql database is not supported** and is very likely to produce undesirable results.
Chapter 3 Restrictions on Server-Side Cursors

Server-side cursors are implemented in the C API using the `mysql_stmt_attr_set()` function. The same implementation is used for cursors in stored routines. A server-side cursor enables a result set to be generated on the server side, but not transferred to the client except for those rows that the client requests. For example, if a client executes a query but is only interested in the first row, the remaining rows are not transferred.

In MySQL, a server-side cursor is materialized into an internal temporary table. Initially, this is a MEMORY table, but is converted to a MyISAM table when its size exceeds the minimum value of the `max_heap_table_size` and `tmp_table_size` system variables. The same restrictions apply to internal temporary tables created to hold the result set for a cursor as for other uses of internal temporary tables. See Internal Temporary Table Use in MySQL. One limitation of the implementation is that for a large result set, retrieving its rows through a cursor might be slow.

Cursors are read only; you cannot use a cursor to update rows.

`UPDATE WHERE CURRENT OF` and `DELETE WHERE CURRENT OF` are not implemented, because updatable cursors are not supported.

Cursors are nonholdable (not held open after a commit).

Cursors are a sensitive.

Cursors are nonscrollable.

Cursors are not named. The statement handler acts as the cursor ID.

You can have open only a single cursor per prepared statement. If you need several cursors, you must prepare several statements.

You cannot use a cursor for a statement that generates a result set if the statement is not supported in prepared mode. This includes statements such as `CHECK TABLE`, `HANDLER READ`, and `SHOW BINLOG EVENTS`. 
Chapter 4 Restrictions on Subqueries

- In MySQL 5.1 before 5.1.16, if you compare a NULL value to a subquery using `ALL`, `ANY`, or `SOME`, and the subquery returns an empty result, the comparison might evaluate to the nonstandard result of `NULL` rather than to `TRUE` or `FALSE`. As of 5.1.16, the comparison evaluates to `TRUE` or `FALSE` except for subqueries inside `IS NULL`, such as this:

```sql
SELECT ... WHERE NULL IN (SELECT ...) IS NULL
```

As of 5.1.32, the `IS NULL` limitation is removed and the comparison evaluates to `TRUE` or `FALSE`.

- Subquery optimization for `IN` is not as effective as for the `=` operator or for the `IN(value_list)` operator.

  A typical case for poor `IN` subquery performance is when the subquery returns a small number of rows but the outer query returns a large number of rows to be compared to the subquery result.

  The problem is that, for a statement that uses an `IN` subquery, the optimizer rewrites it as a correlated subquery. Consider the following statement that uses an uncorrelated subquery:

```sql
SELECT ... FROM t1 WHERE t1.a IN (SELECT b FROM t2);
```

  The optimizer rewrites the statement to a correlated subquery:

```sql
SELECT ... FROM t1 WHERE EXISTS (SELECT 1 FROM t2 WHERE t2.b = t1.a);
```

  If the inner and outer queries return $M$ and $N$ rows, respectively, the execution time becomes on the order of $O(M \times N)$, rather than $O(M + N)$ as it would be for an uncorrelated subquery.

  An implication is that an `IN` subquery can be much slower than a query written using an `IN(value_list)` operator that lists the same values that the subquery would return.

- In general, you cannot modify a table and select from the same table in a subquery. For example, this limitation applies to statements of the following forms:

```sql
DELETE FROM t WHERE ... (SELECT ... FROM t ...);
UPDATE t ... WHERE col = (SELECT ... FROM t ...);
{INSERT|REPLACE} INTO t (SELECT ... FROM t ...);
```

  Exception: The preceding prohibition does not apply if you are using a subquery for the modified table in the `FROM` clause. Example:

```sql
UPDATE t ... WHERE col = (SELECT * FROM (SELECT ... FROM t...) AS _t ...);
```

  Here the result from the subquery in the `FROM` clause is stored as a temporary table, so the relevant rows in $t$ have already been selected by the time the update to $t$ takes place.

- Row comparison operations are only partially supported:

  - For `expr [NOT] IN subquery, expr` can be an $n$-tuple (specified using row constructor syntax) and the subquery can return rows of $n$-tuples. The permitted syntax is therefore more specifically expressed as `row_constructor [NOT] IN table_subquery`
• For `expr op {ALL|ANY|SOME} subquery, expr` must be a scalar value and the subquery must be a column subquery; it cannot return multiple-column rows.

In other words, for a subquery that returns rows of \( n \)-tuples, this is supported:

\[
(expr_1, \ldots, expr_n) \text{ [NOT] IN } table\_subquery
\]

But this is not supported:

\[
(expr_1, \ldots, expr_n) \text{ op } {ALL|ANY|SOME} \text{ subquery}
\]

The reason for supporting row comparisons for \( \text{IN} \) but not for the others is that \( \text{IN} \) is implemented by rewriting it as a sequence of \( = \) comparisons and \( \text{AND} \) operations. This approach cannot be used for \( \text{ALL}, \text{ANY}, \text{or SOME} \).

• Prior to MySQL 5.1.12, row constructors were not well optimized; of the following two equivalent expressions, only the second could be optimized:

\[
(col1, col2, \ldots) = (val1, val2, \ldots)\\
coll = val1 \text{ AND } col2 = val2 \text{ AND } ...
\]

In MySQL 5.1.12 and later, all row equalities are converted into conjunctions of equalities between row elements, and handled by the optimizer in the same way. (Bug #16081)

• Subqueries in the `FROM` clause cannot be correlated subqueries. They are materialized in whole (evaluated to produce a result set) before evaluating the outer query, so they cannot be evaluated per row of the outer query.

• MySQL does not support \( \text{LIMIT} \) in subqueries for certain subquery operators:

```
mysql> SELECT * FROM t1
    -> WHERE s1 IN (SELECT s2 FROM t2 ORDER BY s1 LIMIT 1);
ERROR 1235 (42000): This version of MySQL doesn't yet support
'LIMIT & IN/ALL/ANY/SOME subquery'
```

• The optimizer is more mature for joins than for subqueries, so in many cases a statement that uses a subquery can be executed more efficiently if you rewrite it as a join.

An exception occurs for the case where an \( \text{IN} \) subquery can be rewritten as a `SELECT DISTINCT` join. Example:

```
SELECT col FROM t1 WHERE id_col IN (SELECT id_col2 FROM t2 WHERE condition);
```

That statement can be rewritten as follows:

```
SELECT DISTINCT col FROM t1, t2 WHERE t1.id_col = t2.id_col AND condition;
```

But in this case, the join requires an extra `DISTINCT` operation and is not more efficient than the subquery.

• MySQL permits a subquery to refer to a stored function that has data-modifying side effects such as inserting rows into a table. For example, if \( f() \) inserts rows, the following query can modify data:
SELECT ... WHERE x IN (SELECT f() ...);

This behavior is an extension to the SQL standard. In MySQL, it can produce indeterminate results because \( f() \) might be executed a different number of times for different executions of a given query depending on how the optimizer chooses to handle it.

For statement-based or mixed-format replication, one implication of this indeterminism is that such a query can produce different results on the master and its slaves.
Chapter 5 Restrictions on Views

View processing is not optimized:

- It is not possible to create an index on a view.
- Indexes can be used for views processed using the merge algorithm. However, a view that is processed with the temptable algorithm is unable to take advantage of indexes on its underlying tables (although indexes can be used during generation of the temporary tables).

Subqueries cannot be used in the \textit{FROM} clause of a view.

There is a general principle that you cannot modify a table and select from the same table in a subquery. See \textit{Chapter 4, Restrictions on Subqueries}.

The same principle also applies if you select from a view that selects from the table, if the view selects from the table in a subquery and the view is evaluated using the merge algorithm. Example:

\begin{verbatim}
CREATE VIEW v1 AS
  SELECT * FROM t2 WHERE EXISTS (SELECT 1 FROM t1 WHERE t1.a = t2.a);
UPDATE t1, v2 SET t1.a = 1 WHERE t1.b = v2.b;
\end{verbatim}

If the view is evaluated using a temporary table, you \textit{can} select from the table in the view subquery and still modify that table in the outer query. In this case the view will be stored in a temporary table and thus you are not really selecting from the table in a subquery and modifying it “at the same time.” (This is another reason you might wish to force MySQL to use the temptable algorithm by specifying \texttt{ALGORITHM = TEMPTABLE} in the view definition.)

You can use \texttt{DROP TABLE} or \texttt{ALTER TABLE} to drop or alter a table that is used in a view definition. No warning results from the \texttt{DROP} or \texttt{ALTER} operation, even though this invalidates the view. Instead, an error occurs later, when the view is used. \texttt{CHECK TABLE} can be used to check for views that have been invalidated by \texttt{DROP} or \texttt{ALTER} operations.

A view definition is “frozen” by certain statements:

- If a statement prepared by \texttt{PREPARE} refers to a view, the view definition seen each time the statement is executed later will be the definition of the view at the time it was prepared. This is true even if the view definition is changed after the statement is prepared and before it is executed. Example:

\begin{verbatim}
CREATE VIEW v AS SELECT RAND();
PREPARE s FROM 'SELECT * FROM v';
ALTER VIEW v AS SELECT NOW();
EXECUTE s;
\end{verbatim}

The result returned by the \texttt{EXECUTE} statement is a random number, not the current date and time.

- If a statement in a stored routine refers to a view, the view definition seen by the statement are its definition the first time that statement is executed. For example, this means that if the statement is executed in a loop, further iterations of the statement see the same view definition, even if the definition is changed later in the loop. Example:

\begin{verbatim}
CREATE VIEW v AS SELECT 1;
delimiter //
CREATE PROCEDURE p ()
BEGIN
  DECLARE i INT DEFAULT 0;
\end{verbatim}
WHILE i < 5 DO
    SELECT * FROM v;
    SET i = i + 1;
    ALTER VIEW v AS SELECT 2;
END WHILE;
END;
//
delimiter ;
CALL p();

When the procedure p() is called, the SELECT returns 1 each time through the loop, even though the view definition is changed within the loop.

As of MySQL 5.1.21, ALTER VIEW is prohibited within stored routines, so this restriction does not apply.

With regard to view updatability, the overall goal for views is that if any view is theoretically updatable, it should be updatable in practice. This includes views that have UNION in their definition. Not all views that are theoretically updatable can be updated. The initial view implementation was deliberately written this way to get usable, updatable views into MySQL as quickly as possible. Many theoretically updatable views can be updated now, but limitations still exist:

- Updatable views with subqueries anywhere other than in the WHERE clause. Some views that have subqueries in the SELECT list may be updatable.
- You cannot use UPDATE to update more than one underlying table of a view that is defined as a join.
- You cannot use DELETE to update a view that is defined as a join.

There exists a shortcoming with the current implementation of views. If a user is granted the basic privileges necessary to create a view (the CREATE VIEW and SELECT privileges), that user will be unable to call SHOW CREATE VIEW on that object unless the user is also granted the SHOW VIEW privilege.

That shortcoming can lead to problems backing up a database with mysqldump, which may fail due to insufficient privileges. This problem is described in Bug #22062.

The workaround to the problem is for the administrator to manually grant the SHOW VIEW privilege to users who are granted CREATE VIEW, since MySQL doesn’t grant it implicitly when views are created.

Views do not have indexes, so index hints do not apply. Use of index hints when selecting from a view is not permitted.

SHOW CREATE VIEW displays view definitions using an AS alias_name clause for each column. If a column is created from an expression, the default alias is the expression text, which can be quite long. As of MySQL 5.1.23, aliases for column names in CREATE VIEW statements are checked against the maximum column length of 64 characters (not the maximum alias length of 256 characters). As a result, views created from the output of SHOW CREATE VIEW fail if any column alias exceeds 64 characters. This can cause problems in the following circumstances for views with too-long aliases:

- View definitions fail to replicate to newer slaves that enforce the column-length restriction.
- Dump files created with mysqldump cannot be loaded into servers that enforce the column-length restriction.

A workaround for either problem is to modify each problematic view definition to use aliases that provide shorter column names. Then the view will replicate properly, and can be dumped and reloaded without causing an error. To modify the definition, drop and create the view again with DROP VIEW and CREATE VIEW, or replace the definition with CREATE OR REPLACE VIEW.
For problems that occur when reloading view definitions in dump files, another workaround is to edit the dump file to modify its `CREATE VIEW` statements. However, this does not change the original view definitions, which may cause problems for subsequent dump operations.
Chapter 6 Restrictions on XA Transactions

XA transaction support is limited to the InnoDB storage engine.

For “external XA,” a MySQL server acts as a Resource Manager and client programs act as Transaction Managers. For “Internal XA”, storage engines within a MySQL server act as RMs, and the server itself acts as a TM. Internal XA support is limited by the capabilities of individual storage engines. Internal XA is required for handling XA transactions that involve more than one storage engine. The implementation of internal XA requires that a storage engine support two-phase commit at the table handler level, and currently this is true only for InnoDB.

For XA START, the JOIN and RESUME clauses are not supported.

For XA END, the SUSPEND [FOR MIGRATE] clause is not supported.

The requirement that the bqual part of the xid value be different for each XA transaction within a global transaction is a limitation of the current MySQL XA implementation. It is not part of the XA specification.

If an XA transaction has reached the PREPARED state and the MySQL server is killed (for example, with kill -9 on Unix) or shuts down abnormally, the transaction can be continued after the server restarts. However, if the client reconnects and commits the transaction, the transaction will be absent from the binary log even though it has been committed. This means the data and the binary log have gone out of synchrony. An implication is that XA cannot be used safely together with replication.

It is possible that the server will roll back a pending XA transaction, even one that has reached the PREPARED state. This happens if a client connection terminates and the server continues to run, or if clients are connected and the server shuts down gracefully. (In the latter case, the server marks each connection to be terminated, and then rolls back the PREPARED XA transaction associated with it.) It should be possible to commit or roll back a PREPARED XA transaction, but this cannot be done without changes to the binary logging mechanism.
Chapter 7 Restrictions on Character Sets

- Identifiers are stored in `mysql` database tables (`user`, `db`, and so forth) using `utf8`, but identifiers can contain only characters in the Basic Multilingual Plane (BMP). Supplementary characters are not permitted in identifiers.

- The `ucs2` character sets has the following restrictions:
  
  - It cannot be used as a client character set, which means that it does not work for `SET NAMES` or `SET CHARACTER SET`. (See Connection Character Sets and Collations.)
  
  - It is currently not possible to use `LOAD DATA INFILE` to load data files that use this character set.
  
  - `FULLTEXT` indexes cannot be created on a column that this character set. However, you can perform `IN BOOLEAN MODE` searches on the column without an index.
  
  - The use of `ENCRIPT()` with this character set is not recommended because the underlying system call expects a string terminated by a zero byte.
  
  - The `REEXP` and `RLIKE` operators work in byte-wise fashion, so they are not multibyte safe and may produce unexpected results with multibyte character sets. In addition, these operators compare characters by their byte values and accented characters may not compare as equal even if a given collation treats them as equal.
Chapter 8 Limits in MySQL

Table of Contents

8.1 Limits on Joins ................................................................. 23
8.2 Limits on Number of Databases and Tables ....................................................... 23
8.3 Limits on Table Size ........................................................... 23
8.4 Limits on Table Column Count and Row Size ..................................................... 25
8.5 Limits Imposed by .frm File Structure ................................................................. 26
8.6 Windows Platform Limitations ................................................................. 27

This section lists current limits in MySQL 5.1.

8.1 Limits on Joins

The maximum number of tables that can be referenced in a single join is 61. This also applies to the number of tables that can be referenced in the definition of a view.

8.2 Limits on Number of Databases and Tables

MySQL has no limit on the number of databases. The underlying file system may have a limit on the number of directories.

MySQL has no limit on the number of tables. The underlying file system may have a limit on the number of files that represent tables. Individual storage engines may impose engine-specific constraints. InnoDB permits up to 4 billion tables.

8.3 Limits on Table Size

The effective maximum table size for MySQL databases is usually determined by operating system constraints on file sizes, not by MySQL internal limits. The following table lists some examples of operating system file-size limits. This is only a rough guide and is not intended to be definitive. For the most up-to-date information, be sure to check the documentation specific to your operating system.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>File-size Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win32 w/ FAT/FAT32</td>
<td>2GB/4GB</td>
</tr>
<tr>
<td>Win32 w/ NTFS</td>
<td>2TB (possibly larger)</td>
</tr>
<tr>
<td>Linux 2.2-Intel 32-bit</td>
<td>2GB (LFS: 4GB)</td>
</tr>
<tr>
<td>Linux 2.4+</td>
<td>(using ext3 file system) 4TB</td>
</tr>
<tr>
<td>Solaris 9/10</td>
<td>16TB</td>
</tr>
<tr>
<td>OS X w/ HFS+</td>
<td>2TB</td>
</tr>
<tr>
<td>NetWare w/NSS file system</td>
<td>8TB</td>
</tr>
</tbody>
</table>

Windows users, please note that FAT and VFAT (FAT32) are not considered suitable for production use with MySQL. Use NTFS instead.
On Linux 2.2, you can get MyISAM tables larger than 2GB in size by using the Large File Support (LFS) patch for the ext2 file system. Most current Linux distributions are based on kernel 2.4 or higher and include all the required LFS patches. On Linux 2.4, patches also exist for ReiserFS to get support for big files (up to 2TB). With JFS and XFS, petabyte and larger files are possible on Linux.

For a detailed overview about LFS in Linux, have a look at Andreas Jaeger’s Large File Support in Linux page at http://www.suse.de/~aj/linux_lfs.html.

If you do encounter a full-table error, there are several reasons why it might have occurred:

- The disk might be full.

- The InnoDB storage engine maintains InnoDB tables within a tablespace that can be created from several files. This enables a table to exceed the maximum individual file size. The tablespace can include raw disk partitions, which permits extremely large tables. The maximum tablespace size is 64TB.

  If you are using InnoDB tables and run out of room in the InnoDB tablespace. In this case, the solution is to extend the InnoDB tablespace. See Changing the Number or Size of InnoDB Redo Log Files.

- You are using MyISAM tables on an operating system that supports files only up to 2GB in size and you have hit this limit for the data file or index file.

- You are using a MyISAM table and the space required for the table exceeds what is permitted by the internal pointer size. MyISAM permits data and index files to grow up to 256TB by default, but this limit can be changed up to the maximum permissible size of 65,536TB (256^7 – 1 bytes).

  If you need a MyISAM table that is larger than the default limit and your operating system supports large files, the CREATE TABLE statement supports AVG_ROW_LENGTH and MAX_ROWS options. See CREATE TABLE Syntax. The server uses these options to determine how large a table to permit.

  If the pointer size is too small for an existing table, you can change the options with ALTER TABLE to increase a table’s maximum permissible size. See ALTER TABLE Syntax.

    ```sql
    ALTER TABLE tbl_name MAX_ROWS=1000000000 AVG_ROW_LENGTH=nnn;
    ```

    You have to specify AVG_ROW_LENGTH only for tables with BLOB or TEXT columns; in this case, MySQL can’t optimize the space required based only on the number of rows.

    To change the default size limit for MyISAM tables, set the myisam_data_pointer_size, which sets the number of bytes used for internal row pointers. The value is used to set the pointer size for new tables if you do not specify the MAX_ROWS option. The value of myisam_data_pointer_size can be from 2 to 7. A value of 4 permits tables up to 4GB; a value of 6 permits tables up to 256TB.

    You can check the maximum data and index sizes by using this statement:

    ```sql
    SHOW TABLE STATUS FROM db_name LIKE 'tbl_name';
    ```

    You also can use myisamchk -dv /path/to/table-index-file. See SHOW Syntax, or myisamchk — MyISAM Table-Maintenance Utility.

Other ways to work around file-size limits for MyISAM tables are as follows:

- If your large table is read only, you can use myisampack to compress it. myisampack usually compresses a table by at least 50%, so you can have, in effect, much bigger tables. myisampack also can merge multiple tables into a single table. See myisampack — Generate Compressed, Read-Only MyISAM Tables.
• MySQL includes a **MERGE** library that enables you to handle a collection of **MyISAM** tables that have identical structure as a single **MERGE** table. See The **MERGE** Storage Engine.

• You are using the **NDB** storage engine, in which case you need to increase the values for the **DataMemory** and **IndexMemory** configuration parameters in your **config.ini** file. See MySQL Cluster Data Node Configuration Parameters.

• You are using the **MEMORY** (HEAP) storage engine; in this case you need to increase the value of the **max_heap_table_size** system variable. See Server System Variables.

### 8.4 Limits on Table Column Count and Row Size

There is a hard limit of 4096 columns per table, but the effective maximum may be less for a given table. The exact limit depends on several interacting factors.

• Every table (regardless of storage engine) has a maximum row size of 65,535 bytes. Storage engines may place additional constraints on this limit, reducing the effective maximum row size.

The maximum row size constrains the number (and possibly size) of columns because the total length of all columns cannot exceed this size. For example, utf8 characters require up to three bytes per character, so for a **CHAR(255) CHARACTER SET utf8** column, the server must allocate 255 × 3 = 765 bytes per value. Consequently, a table cannot contain more than 65,535 / 765 = 85 such columns.

Storage for variable-length columns includes length bytes, which are assessed against the row size. For example, a **VARCHAR(255) CHARACTER SET utf8** column takes two bytes to store the length of the value, so each value can take up to 767 bytes.

**BLOB** and **TEXT** columns count from one to four plus eight bytes each toward the row-size limit because their contents are stored separately from the rest of the row.

Declaring columns **NULL** can reduce the maximum number of columns permitted. For **MyISAM** tables, **NULL** columns require additional space in the row to record whether their values are **NULL**. Each **NULL** column takes one bit extra, rounded up to the nearest byte. The maximum row length in bytes can be calculated as follows:

\[
\text{row length} = 1 + (\text{sum of column lengths}) + (\text{number of NULL columns} + \text{delete_flag} + 7)/8 + (\text{number of variable-length columns})
\]

**delete_flag** is 1 for tables with static row format. Static tables use a bit in the row record for a flag that indicates whether the row has been deleted. **delete_flag** is 0 for dynamic tables because the flag is stored in the dynamic row header. For information about **MyISAM** table formats, see MyISAM Table Storage Formats.

For InnoDB tables, storage size is the same for **NULL** and **NOT NULL** columns, so the preceding calculations do not apply.

The following statement to create table **t1** succeeds because the columns require 32,765 + 2 bytes and 32,766 + 2 bytes, which falls within the maximum row size of 65,535 bytes:

```sql
mysql> CREATE TABLE t1
-> (c1 VARCHAR(32765) NOT NULL, c2 VARCHAR(32766) NOT NULL)
-> ENGINE = MyISAM CHARACTER SET latin1;
Query OK, 0 rows affected (0.02 sec)
```
The following statement to create table \texttt{t2} fails because the columns are \texttt{NULL} and \texttt{MyISAM} requires additional space that causes the row size to exceed 65,535 bytes:

```sql
mysql> CREATE TABLE t2
    -> (c1 VARCHAR(32765) NULL, c2 VARCHAR(32766) NULL)
    -> ENGINE = MyISAM CHARACTER SET latin1;
ERROR 1118 (42000): Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. You have to change some columns to TEXT or BLOBs
```

The following statement to create table \texttt{t3} fails because, although the column length is within the maximum length of 65,535 bytes, two additional bytes are required to record the length, which causes the row size to exceed 65,535 bytes:

```sql
mysql> CREATE TABLE t3
    -> (c1 VARCHAR(65535) NOT NULL)
    -> ENGINE = MyISAM CHARACTER SET latin1;
ERROR 1118 (42000): Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. You have to change some columns to TEXT or BLOBs
```

Reducing the column length to 65,533 or less permits the statement to succeed.

- Individual storage engines might impose additional restrictions that limit table column count. Examples:
  - \texttt{InnoDB} permits up to 1000 columns.
  - \texttt{InnoDB} restricts row size to something less than half a database page (approximately 8000 bytes), not including \texttt{VARBINARY}, \texttt{VARCHAR}, \texttt{BLOB}, or \texttt{TEXT} columns. For more information, see Limits on InnoDB Tables.
  - Different \texttt{InnoDB} storage formats (\texttt{COMPRESSED}, \texttt{RENDUNDANT}) use different amounts of page header and trailer data, which affects the amount of storage available for rows.
  - Each table has an \texttt{.frm} file that contains the table definition. The definition affects the content of this file in ways that may affect the number of columns permitted in the table. For more information, see Section 8.5, “Limits Imposed by .frm File Structure”.

### 8.5 Limits Imposed by .frm File Structure

Each table has an \texttt{.frm} file that contains the table definition. The server uses the following expression to check some of the table information stored in the file against an upper limit of 64KB:

```c
if (info_length+(ulong) create_fields.elements*FCOMP+288+
    n_length+int_length+com_length > 65535L || int_count > 255)
```

The portion of the information stored in the \texttt{.frm} file that is checked against the expression cannot grow beyond the 64KB limit, so if the table definition reaches this size, no more columns can be added.

The relevant factors in the expression are:

- \texttt{info_length} is space needed for “screens.” This is related to MySQL’s Unireg heritage.
- \texttt{create_fields.elements} is the number of columns.
- \texttt{FCOMP} is 17.
Windows Platform Limitations

• **n_length** is the total length of all column names, including one byte per name as a separator.

• **int_length** is related to the list of values for **ENUM** and **SET** columns. In this context, “int” does not mean “integer.” It means “interval,” a term that refers collectively to **ENUM** and **SET** columns.

• **int_count** is the number of unique **ENUM** and **SET** definitions.

• **com_length** is the total length of column comments.

The expression just described has several implications for permitted table definitions:

• Using long column names can reduce the maximum number of columns, as can the inclusion of **ENUM** or **SET** columns, or use of column comments.

• A table can have no more than 255 unique **ENUM** and **SET** definitions. Columns with identical element lists are considered the same against this limit. For example, if a table contains these two columns, they count as one (not two) toward this limit because the definitions are identical:

  ```
e1 ENUM('a','b','c')
e2 ENUM('a','b','c')
  ```

• The sum of the length of element names in the unique **ENUM** and **SET** definitions counts toward the 64KB limit, so although the theoretical limit on number of elements in a given **ENUM** column is 65,535, the practical limit is less than 3000.

### 8.6 Windows Platform Limitations

The following limitations apply to use of MySQL on the Windows platform:

• **Number of file descriptors**

  The number of open file descriptors on Windows is limited to a maximum of 2048, which may limit the ability to open a large number of tables simultaneously. This limit is due not to Windows but to C runtime library compatibility functions used to open files on Windows that use the POSIX compatibility layer.

  This limitation will also cause problems if you try to set **open_files_limit** to a value greater than the 2048 file limit.

• **Process memory**

  On Windows 32-bit platforms it is not possible by default to use more than 2GB of RAM within a single process, including MySQL. This is because the physical address limit on Windows 32-bit is 4GB and the default setting within Windows is to split the virtual address space between kernel (2GB) and user/applications (2GB).

  Some versions of Windows have a boot time setting to enable larger applications by reducing the kernel application. Alternatively, to use more than 2GB, use a 64-bit version of Windows.

• **File system aliases**

  When using **MyISAM** tables, you cannot use aliases within Windows link to the data files on another volume and then link back to the main MySQL **datadir** location.

  This facility is often used to move the data and index files to a RAID or other fast solution, while retaining the main **.frm** files in the default data directory configured with the **datadir** option.

• **Limited number of ports**
Windows Platform Limitations

Windows systems have about 4,000 ports available for client connections, and after a connection on a port closes, it takes two to four minutes before the port can be reused. In situations where clients connect to and disconnect from the server at a high rate, it is possible for all available ports to be used up before closed ports become available again. If this happens, the MySQL server appears to be unresponsive even though it is running. Ports may be used by other applications running on the machine as well, in which case the number of ports available to MySQL is lower.

For more information about this problem, see http://support.microsoft.com/default.aspx?scid=kb;en-us;196271.

• Concurrent reads

MySQL depends on the `pread()` and `pwrite()` system calls to be able to mix `INSERT` and `SELECT`. We use mutexes to emulate `pread()` and `pwrite()`. We intend to replace the file level interface with a virtual interface in the future so that we can use the `readfile()`/`writefile()` interface to get more speed. The current implementation limits the number of open files that MySQL 5.1 can use to 2,048, which means that you cannot run as many concurrent threads on Windows as on Unix.

This problem is fixed in MySQL 5.5.

• Blocking read

Before MySQL 5.1.41, MySQL uses a blocking read for each connection. That has the following implications if named-pipe connections are enabled:

• A connection is not disconnected automatically after eight hours, as happens with the Unix version of MySQL.
• If a connection hangs, it is not possible to break it without killing MySQL.
• `mysqladmin kill` does not work on a sleeping connection.
• `mysqladmin shutdown` cannot abort as long as there are sleeping connections.

• DROP DATABASE

You cannot drop a database that is in use by another session.

• Case-insensitive names

File names are not case sensitive on Windows, so MySQL database and table names are also not case sensitive on Windows. The only restriction is that database and table names must be specified using the same case throughout a given statement. See Identifier Case Sensitivity.

• Directory and file names

On Windows, MySQL Server supports only directory and file names that are compatible with the current ANSI code pages. For example, the following Japanese directory name will not work in the Western locale (code page 1252):

```sql
datadir="C:\私たちのプロジェクトのデータ"
```

The same limitation applies to directory and file names referred to in SQL statements, such as the data file path name in `LOAD DATA INFILE`.

• The “\” path name separator character
Path name components in Windows are separated by the "\" character, which is also the escape character in MySQL. If you are using `LOAD DATA INFILE` or `SELECT ... INTO OUTFILE`, use Unix-style file names with "/" characters:

| mysql> LOAD DATA INFILE 'C:/tmp/skr.txt' INTO TABLE skr; |
| mysql> SELECT * INTO OUTFILE 'C:/tmp/skr.txt' FROM skr; |

Alternatively, you must double the "\" character:

| mysql> LOAD DATA INFILE 'C:\tmp\skr.txt' INTO TABLE skr; |
| mysql> SELECT * INTO OUTFILE 'C:\tmp\skr.txt' FROM skr; |

- **Problems with pipes**

  Pipes do not work reliably from the Windows command-line prompt. If the pipe includes the character `^Z / CHAR(24)`, Windows thinks that it has encountered end-of-file and aborts the program.

  This is mainly a problem when you try to apply a binary log as follows:

  | C:\> mysqlbinlog binary_log_file | mysql --user=root |

  If you have a problem applying the log and suspect that it is because of a `^Z / CHAR(24)` character, you can use the following workaround:

  | C:\> mysqlbinlog binary_log_file --result-file=/tmp/bin.sql |
  | C:\> mysql --user=root --execute "source /tmp/bin.sql" |

  The latter command also can be used to reliably read in any SQL file that may contain binary data.
Chapter 9 Restrictions and Limitations on Partitioning

Table of Contents

9.1 Partitioning Keys, Primary Keys, and Unique Keys .................................................. 37
9.2 Partitioning Limitations Relating to Storage Engines .................................................. 40
9.3 Partitioning Limitations Relating to Functions ............................................................. 42
9.4 Partitioning and Table-Level Locking ............................................................................ 42

This section discusses current restrictions and limitations on MySQL partitioning support.

Prohibited constructs. Beginning with MySQL 5.1.12, the following constructs are not permitted in
partitioning expressions:

• Stored procedures, stored functions, UDFs, or plugins.
• Declared variables or user variables.

For a list of SQL functions which are permitted in partitioning expressions, see Section 9.3, “Partitioning
Limitations Relating to Functions”.

Arithmetic and logical operators. Use of the arithmetic operators +, −, and * is permitted in
partitioning expressions. However, the result must be an integer value or NULL (except in the case of
[LINEAR] KEY partitioning, as discussed elsewhere in this chapter; see Partitioning Types, for more
information).

Beginning with MySQL 5.1.23, the DIV operator is also supported, and the / operator is not permitted.
(Bug #30188, Bug #33182)

Beginning with MySQL 5.1.12, the bit operators |, &, ^, <<, >>, and ~ are not permitted in partitioning
expressions.

HANDLER statements. In MySQL 5.1, the HANDLER statement is not supported with partitioned tables.

Server SQL mode. Tables employing user-defined partitioning do not preserve the SQL mode in
effect at the time that they were created. As discussed in Server SQL Modes, the results of many MySQL
functions and operators may change according to the server SQL mode. Therefore, a change in the SQL
mode at any time after the creation of partitioned tables may lead to major changes in the behavior of such
tables, and could easily lead to corruption or loss of data. For these reasons, it is strongly recommended
that you never change the server SQL mode after creating partitioned tables.

Examples. The following examples illustrate some changes in behavior of partitioned tables due to a
change in the server SQL mode:

1. Error handling. Suppose that you create a partitioned table whose partitioning expression is one
such as column DIV 0 or column MOD 0, as shown here:

```sql
mysql> CREATE TABLE tn (c1 INT)
    -> PARTITION BY LIST(1 DIV c1) {
    -> PARTITION p0 VALUES IN (NULL),
    -> PARTITION p1 VALUES IN (1)
    ->);
Query OK, 0 rows affected (0.05 sec)
```

The default behavior for MySQL is to return NULL for the result of a division by zero, without producing
any errors:
However, changing the server SQL mode to treat division by zero as an error and to enforce strict error handling causes the same INSERT statement to fail, as shown here:

```
mysql> SET sql_mode='STRICT_ALL_TABLES,ERROR_FOR_DIVISION_BY_ZERO';
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO tn VALUES (NULL), (0), (1);
ERROR 1365 (22012): Division by 0
```

2. **Table accessibility.** Sometimes a change in the server SQL mode can make partitioned tables unusable. The following CREATE TABLE statement can be executed successfully only if the NO_UNSIGNED_SUBTRACTION mode is in effect:

```
mysql> SELECT @@sql_mode;
+------------+
| @@sql_mode |
+------------+
1 row in set (0.00 sec)
mysql> CREATE TABLE tu (c1 BIGINT UNSIGNED)
    -> PARTITION BY RANGE(c1 - 10) (
    -> PARTITION p0 VALUES LESS THAN (-5),
    -> PARTITION p1 VALUES LESS THAN (0),
    -> PARTITION p2 VALUES LESS THAN (5),
    -> PARTITION p3 VALUES LESS THAN (10),
    -> PARTITION p4 VALUES LESS THAN (MAXVALUE)
    -> );
ERROR 1563 (HY000): Partition constant is out of partition function domain
```

If you remove the NO_UNSIGNED_SUBTRACTION server SQL mode after creating `tu`, you may no longer be able to access this table:
Server SQL modes also impact replication of partitioned tables. Differing SQL modes on master and slave can lead to partitioning expressions being evaluated differently; this can cause the distribution of data among partitions to be different in the master's and slave's copies of a given table, and may even cause inserts into partitioned tables that succeed on the master to fail on the slave. For best results, you should always use the same server SQL mode on the master and on the slave.

**Performance considerations.** Some affects of partitioning operations on performance are given in the following list:

- **File system operations.** Partitioning and repartitioning operations (such as `ALTER TABLE with PARTITION BY ...`, `REORGANIZE PARTITIONS`, or `REMOVE PARTITIONING`) depend on file system operations for their implementation. This means that the speed of these operations is affected by such factors as file system type and characteristics, disk speed, swap space, file handling efficiency of the operating system, and MySQL server options and variables that relate to file handling. In particular, you should make sure that `large_files_support` is enabled and that `open_files_limit` is set properly. For partitioned tables using the MyISAM storage engine, increasing `myisam_max_sort_file_size` may improve performance; partitioning and repartitioning operations involving InnoDB tables may be made more efficient by enabling `innodb_file_per_table`. See also Maximum number of partitions.

- **MyISAM and partition file descriptor usage.** For a partitioned MyISAM table, MySQL uses 2 file descriptors for each partition, for each such table that is open. This means that you need many more file descriptors to perform operations on a partitioned MyISAM table than on a table which is identical to it except that the latter table is not partitioned, particularly when performing `ALTER TABLE` operations that change the table's partitioning scheme.

Assume a MyISAM table `t` with 100 partitions, such as the table created by this SQL statement:

```sql
CREATE TABLE t (c1 VARCHAR(50))
PARTITION BY KEY (c1) PARTITIONS 100
ENGINE=MYISAM;
```

**Note**

For brevity, we use `KEY` partitioning for the table shown in this example, but file descriptor usage as described here applies to all partitioned MyISAM tables, regardless of the type of partitioning that is employed. Partitioned tables using other storage engines such as InnoDB are not affected by this issue.

Now assume that you wish to repartition `t` so that it has 101 partitions, using the statement shown here:

```sql
ALTER TABLE t PARTITION BY KEY (c1) PARTITIONS 101;
```

To process this `ALTER TABLE` statement, MySQL uses 402 file descriptors—that is, two for each of the 100 original partitions, plus two for each of the 101 new partitions. This is because all partitions (old and new) must be opened concurrently during the reorganization of the table data. It is recommended that, if
you expect to perform such operations, you should make sure that `--open-files-limit` is not set too low to accommodate them.

- **Table locks.** The process executing a partitioning operation on a table takes a write lock on the table. Reads from such tables are relatively unaffected; pending `INSERT` and `UPDATE` operations are performed as soon as the partitioning operation has completed.

- **Storage engine.** Partitioning operations, queries, and update operations generally tend to be faster with `MyISAM` tables than with `InnoDB` or `NDB` tables.

- **Indexes; partition pruning.** As with nonpartitioned tables, proper use of indexes can speed up queries on partitioned tables significantly. In addition, designing partitioned tables and queries on these tables to take advantage of `partition pruning` can improve performance dramatically. See [Partition Pruning](#), for more information.

- **Performance with LOAD DATA.** Prior to MySQL 5.1.23, `LOAD DATA` performed very poorly when importing into partitioned tables. The statement now uses buffering to improve performance; however, the buffer uses 130 KB memory per partition to achieve this. (Bug #26527)

**Maximum number of partitions.**
The maximum possible number of partitions for a given table (that does not use the `NDB` storage engine) is 1024. This number includes subpartitions.

The maximum possible number of user-defined partitions for a table using the `NDBCLUSTER` storage engine is determined according to the version of the MySQL Cluster software being used, the number of data nodes, and other factors. See [NDB and user-defined partitioning](#), for more information.

If, when creating tables with a large number of partitions (but less than the maximum), you encounter an error message such as `Got error ... from storage engine: Out of resources when opening file`, you may be able to address the issue by increasing the value of the `open_files_limit` system variable. However, this is dependent on the operating system, and may not be possible or advisable on all platforms; see [‘File’ Not Found and Similar Errors](#), for more information. In some cases, using large numbers (hundreds) of partitions may also not be advisable due to other concerns, so using more partitions does not automatically lead to better results.

See also [File system operations](#).

**Foreign keys not supported for partitioned InnoDB tables.**
Partitioned tables using the `InnoDB` storage engine do not support foreign keys. More specifically, this means that the following two statements are true:

1. No definition of an `InnoDB` table employing user-defined partitioning may contain foreign key references; no `InnoDB` table whose definition contains foreign key references may be partitioned.

2. No `InnoDB` table definition may contain a foreign key reference to a user-partitioned table; no `InnoDB` table with user-defined partitioning may contain columns referenced by foreign keys.

The scope of the restrictions just listed includes all tables that use the `InnoDB` storage engine. `CREATE TABLE` and `ALTER TABLE` statements that would result in tables violating these restrictions are not allowed.

**ALTER TABLE ... ORDER BY.** An `ALTER TABLE ... ORDER BY column` statement run against a partitioned table causes ordering of rows only within each partition.

**Effects on REPLACE statements by modification of primary keys.** It can be desirable in some cases (see Section 9.1, “Partitioning Keys, Primary Keys, and Unique Keys”) to modify a table’s primary key. Be aware that, if your application uses `REPLACE` statements and you do this, the results of these statements can be drastically altered. See [REPLACE Syntax](#), for more information and an example.
**FULLTEXT indexes.**
Partitioned tables do not support **FULLTEXT** indexes or searches. This includes partitioned tables employing the **MyISAM** storage engine.

**Spatial columns.** Columns with spatial data types such as **POINT** or **GEOMETRY** cannot be used in partitioned tables.

**Temporary tables.**
As of MySQL 5.1.8, temporary tables cannot be partitioned. (Bug #17497)

**Log tables.** Beginning with MySQL 5.1.20, it is no longer possible to partition the log tables; beginning with that version, an **ALTER TABLE ... PARTITION BY ...** statement on such a table fails with an error. (Bug #27816)

**Data type of partitioning key.**
A partitioning key must be either an integer column or an expression that resolves to an integer. Expressions employing **ENUM** columns cannot be used. The column or expression value may also be **NULL**. (See How MySQL Partitioning Handles NULL.)

The lone exception to this restriction occurs when partitioning by **[LINEAR] KEY**, it is possible to use columns of any valid MySQL data type other than **TEXT** or **BLOB** as partitioning keys, because MySQL’s internal key-hashing functions produce the correct data type from these types. For example, the following two **CREATE TABLE** statements are valid:

```sql
CREATE TABLE tkc (cl CHAR)
PARTITION BY KEY(cl)
PARTITIONS 4;
CREATE TABLE tke
  ( cl ENUM('red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet') )
PARTITION BY LINEAR KEY(cl)
PARTITIONS 6;
```

**Subqueries.**
A partitioning key may not be a subquery, even if that subquery resolves to an integer value or **NULL**.

**Issues with subpartitions.**
Subpartitions must use **HASH** or **KEY** partitioning. Only **RANGE** and **LIST** partitions may be subpartitioned; **HASH** and **KEY** partitions cannot be subpartitioned.

**SUBPARTITION BY KEY** requires that the subpartitioning column or columns be specified explicitly, unlike the case with **PARTITION BY KEY**, where it can be omitted (in which case the table’s primary key column is used by default). Consider the table created by this statement:

```sql
CREATE TABLE ts (  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(30)
);
```

You can create a table having the same columns, partitioned by **KEY**, using a statement such as this one:

```sql
CREATE TABLE ts (  
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(30)
)  
PARTITION BY KEY()  
PARTITIONS 4;
```
The previous statement is treated as though it had been written like this, with the table’s primary key column used as the partitioning column:

```
CREATE TABLE ts {
    id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(30)
}
PARTITION BY KEY(id)
PARTITIONS 4;
```

However, the following statement that attempts to create a subpartitioned table using the default column as the subpartitioning column fails, and the column must be specified for the statement to succeed, as shown here:

```
mysql> CREATE TABLE ts {
    ->     id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(30)
    -> }
-> PARTITION BY RANGE(id)
-> SUBPARTITION BY KEY()
-> SUBPARTITIONS 4
-> {
    -> PARTITION p0 VALUES LESS THAN (100),
    -> PARTITION p1 VALUES LESS THAN (MAXVALUE)
    ->
};
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ')
```

This is a known issue (see Bug #51470).

**Query cache not supported.**
The query cache is not supported for partitioned tables. Beginning with MySQL 5.1.63, the query cache is automatically disabled for queries involving partitioned tables, and cannot be enabled for such queries. (Bug #53775)

**Key cache not supported.**
Caching is not supported for partitioned tables. The `CACHE INDEX` and `LOAD INDEX INTO CACHE` statements, when you attempt to use them on tables having user-defined partitioning, fail with the errors `The storage engine for the table doesn't support assign_to_keycache` and `The storage engine for the table doesn't support preload_keys`, respectively. This issue is fixed in MySQL 5.5.

**DELAYED option not supported.** Use of `INSERT DELAYED` to insert rows into a partitioned table is not supported. Beginning with MySQL 5.1.23, attempting to do so fails with an error (see Bug #31210).

**DATA DIRECTORY and INDEX DIRECTORY options.** `DATA DIRECTORY` and `INDEX DIRECTORY` are subject to the following restrictions when used with partitioned tables:
• Beginning with MySQL 5.1.23, table-level `DATA DIRECTORY` and `INDEX DIRECTORY` options are ignored (see Bug #32091).

• On Windows, the `DATA DIRECTORY` and `INDEX DIRECTORY` options are not supported for individual partitions or subpartitions (Bug #30459).

**Repairing and rebuilding partitioned tables.** The statements `CHECK TABLE`, `OPTIMIZE TABLE`, `ANALYZE TABLE`, and `REPAIR TABLE` are supported for partitioned tables beginning with MySQL 5.1.27. (See Bug #20129.)

In addition, you can use `ALTER TABLE ... REBUILD PARTITION` to rebuild one or more partitions of a partitioned table; `ALTER TABLE ... REORGANIZE PARTITION` also causes partitions to be rebuilt. Both of these statements were added in MySQL 5.1.5. See `ALTER TABLE Syntax`, for more information about these two statements.

`mysqlcheck`, `myisamchk`, and `myisampack` are not supported with partitioned tables.

### 9.1 Partitioning Keys, Primary Keys, and Unique Keys

This section discusses the relationship of partitioning keys with primary keys and unique keys. The rule governing this relationship can be expressed as follows: All columns used in the partitioning expression for a partitioned table must be part of every unique key that the table may have.

In other words, *every unique key on the table must use every column in the table's partitioning expression.* (This also includes the table's primary key, since it is by definition a unique key. This particular case is discussed later in this section.) For example, each of the following table creation statements is invalid:

```sql
CREATE TABLE t1 (  
  col1 INT NOT NULL,  
  col2 DATE NOT NULL,  
  col3 INT NOT NULL,  
  col4 INT NOT NULL,  
  UNIQUE KEY (col1, col2)  
)  
PARTITION BY HASH(col3)  
PARTITIONS 4;  
CREATE TABLE t2 (  
  col1 INT NOT NULL,  
  col2 DATE NOT NULL,  
  col3 INT NOT NULL,  
  col4 INT NOT NULL,  
  UNIQUE KEY (col1),  
  UNIQUE KEY (col3)  
)  
PARTITION BY HASH(col1 + col3)  
PARTITIONS 4;  
```

In each case, the proposed table would have at least one unique key that does not include all columns used in the partitioning expression.

Each of the following statements is valid, and represents one way in which the corresponding invalid table creation statement could be made to work:

```sql
CREATE TABLE t1 (  
  col1 INT NOT NULL,  
  col2 DATE NOT NULL,  
  col3 INT NOT NULL,  
  col4 INT NOT NULL,  
  UNIQUE KEY (col1, col2, col3)  
)
```
PARTITIONING KEYS, PRIMARY KEYS, AND UNIQUE KEYS

PARTITION BY HASH(col3)
PARTITIONS 4;
CREATE TABLE t2 {
    col1 INT NOT NULL,
    col2 DATE NOT NULL,
    col3 INT NOT NULL,
    col4 INT NOT NULL,
    UNIQUE KEY (col1, col3)
} PARTITION BY HASH(col1 + col3)
PARTITIONS 4;

This example shows the error produced in such cases:

mysql> CREATE TABLE t3 {
    col1 INT NOT NULL,
    col2 DATE NOT NULL,
    col3 INT NOT NULL,
    col4 INT NOT NULL,
    UNIQUE KEY (col1, col2),
    UNIQUE KEY (col3)
} PARTITION BY HASH(col1 + col3)
PARTITIONS 4;
ERROR 1491 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function

The CREATE TABLE statement fails because both col1 and col3 are included in the proposed partitioning key, but neither of these columns is part of both of unique keys on the table. This shows one possible fix for the invalid table definition:

mysql> CREATE TABLE t3 {
    col1 INT NOT NULL,
    col2 DATE NOT NULL,
    col3 INT NOT NULL,
    col4 INT NOT NULL,
    UNIQUE KEY (col1, col2, col3),
    UNIQUE KEY (col3)
} PARTITION BY HASH(col3)
PARTITIONS 4;
Query OK, 0 rows affected (0.05 sec)

In this case, the proposed partitioning key col3 is part of both unique keys, and the table creation statement succeeds.

The following table cannot be partitioned at all, because there is no way to include in a partitioning key any columns that belong to both unique keys:

CREATE TABLE t4 {
    col1 INT NOT NULL,
    col2 INT NOT NULL,
    col3 INT NOT NULL,
    col4 INT NOT NULL,
    UNIQUE KEY (col1, col3),
    UNIQUE KEY (col2, col4)
};

Since every primary key is by definition a unique key, this restriction also includes the table's primary key, if it has one. For example, the next two statements are invalid:
In both cases, the primary key does not include all columns referenced in the partitioning expression. However, both of the next two statements are valid:

If a table has no unique keys—this includes having no primary key—then this restriction does not apply, and you may use any column or columns in the partitioning expression as long as the column type is compatible with the partitioning type.

For the same reason, you cannot later add a unique key to a partitioned table unless the key includes all columns used by the table’s partitioning expression. Consider the partitioned table created as shown here:

It is possible to add a primary key to `t_no_pk` using either of these `ALTER TABLE` statements:
**Partitioning Limitations Relating to Storage Engines**

```sql
mysql> ALTER TABLE t_no_pk ADD PRIMARY KEY(c1);
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0
# drop this PK
mysql> ALTER TABLE t_no_pk DROP PRIMARY KEY;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0
# use another possible PK
mysql> ALTER TABLE t_no_pk ADD PRIMARY KEY(c1, c2);
Query OK, 0 rows affected (0.12 sec)
Records: 0  Duplicates: 0  Warnings: 0
# drop this PK
mysql> ALTER TABLE t_no_pk DROP PRIMARY KEY;
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

However, the next statement fails, because `c1` is part of the partitioning key, but is not part of the proposed primary key:

```sql
# fails with error 1503
mysql> ALTER TABLE t_no_pk ADD PRIMARY KEY(c2);
ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function
```

Since `t_no_pk` has only `c1` in its partitioning expression, attempting to add a unique key on `c2` alone fails. However, you can add a unique key that uses both `c1` and `c2`.

These rules also apply to existing nonpartitioned tables that you wish to partition using `ALTER TABLE` ...

```sql
PARTITION BY. Consider a table `np_pk` created as shown here:

mysql> CREATE TABLE np_pk (  
    ->  id INT NOT NULL AUTO_INCREMENT,  
    ->  name VARCHAR(50),  
    ->  added DATE,  
    ->  PRIMARY KEY (id)  
    -> );
Query OK, 0 rows affected (0.08 sec)
```

The following `ALTER TABLE` statement fails with an error, because the `added` column is not part of any unique key in the table:

```sql
mysql> ALTER TABLE np_pk  
    ->  PARTITION BY HASH( TO_DAYS(added) )  
    ->  PARTITIONS 4;
ERROR 1503 (HY000): A PRIMARY KEY must include all columns in the table's partitioning function
```

However, this statement using the `id` column for the partitioning column is valid, as shown here:

```sql
mysql> ALTER TABLE np_pk  
    ->  PARTITION BY HASH(id)  
    ->  PARTITIONS 4;
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

In the case of `np_pk`, the only column that may be used as part of a partitioning expression is `id`; if you wish to partition this table using any other column or columns in the partitioning expression, you must first modify the table, either by adding the desired column or columns to the primary key, or by dropping the primary key altogether.

### 9.2 Partitioning Limitations Relating to Storage Engines
The following limitations apply to the use of storage engines with user-defined partitioning of tables.

**MERGE storage engine.** User-defined partitioning and the MERGE storage engine are not compatible. Tables using the MERGE storage engine cannot be partitioned. Partitioned tables cannot be merged.

**FEDERATED storage engine.** Partitioning of FEDERATED tables is not supported. Beginning with MySQL 5.1.15, it is not possible to create partitioned FEDERATED tables at all.

**CSV storage engine.** Partitioned tables using the CSV storage engine are not supported. Starting with MySQL 5.1.12, it is not possible to create partitioned CSV tables at all.

**BLACKHOLE storage engine.** Prior to MySQL 5.1.6, tables using the BLACKHOLE storage engine also could not be partitioned.

**InnoDB storage engine.** InnoDB foreign keys and MySQL partitioning are not compatible. Partitioned InnoDB tables cannot have foreign key references, nor can they have columns referenced by foreign keys. InnoDB tables which have or which are referenced by foreign keys cannot be partitioned.

In addition, `ALTER TABLE ... OPTIMIZE PARTITION` does not work correctly with partitioned tables that use the InnoDB storage engine. Use `ALTER TABLE ... REBUILD PARTITION` and `ALTER TABLE ... ANALYZE PARTITION`, instead, for such tables. For more information, see [ALTER TABLE Partition Operations](#).

**User-defined partitioning and the NDB storage engine (MySQL Cluster).** Partitioning by `KEY` (including `LINEAR KEY`) is the only type of partitioning supported for the NDBCLUSTER storage engine. Beginning with MySQL 5.1.12, it is not possible under normal circumstances to create a MySQL Cluster table using any partitioning type other than `[LINEAR] KEY`, and attempting to do so fails with an error.

*Exception (not for production):* It is possible to override this restriction by setting the new system variable on MySQL Cluster SQL nodes to `ON`. If you choose to do this, you should be aware that tables using partitioning types other than `[LINEAR] KEY` are not supported in production. *In such cases, you can create and use tables with partitioning types other than KEY or LINEAR KEY, but you do this entirely at your own risk.*

The maximum number of partitions that can be defined for an NDBCLUSTER table depends on the number of data nodes and node groups in the cluster, the version of the MySQL Cluster software in use, and other factors. See [NDB and user-defined partitioning](#), for more information.

Beginning with MySQL Cluster NDB 6.2.18, MySQL Cluster NDB 6.3.25, and MySQL Cluster NDB 7.0.6, `CREATE TABLE` and `ALTER TABLE` statements that would cause a user-partitioned NDBCLUSTER table not to meet either or both of the following two requirements are not permitted, and fail with an error (Bug #40709):

1. The table must have an explicit primary key.
2. All columns listed in the table's partitioning expression must be part of the primary key.

*Exception.* If a user-partitioned NDBCLUSTER table is created using an empty column-list (that is, using `PARTITION BY KEY()` or `PARTITION BY LINEAR KEY()`), then no explicit primary key is required.

**Upgrading partitioned tables.** When performing an upgrade, tables which are partitioned by `KEY` and which use any storage engine other than NDBCLUSTER must be dumped and reloaded.

**Same storage engine for all partitions.** All partitions of a partitioned table must use the same storage engine and it must be the same storage engine used by the table as a whole. In addition, if one does not specify an engine on the table level, then one must do either of the following when creating or altering a partitioned table:
9.3 Partitioning Limitations Relating to Functions

This section discusses limitations in MySQL Partitioning relating specifically to functions used in partitioning expressions.

Beginning with MySQL 5.1.12, only the MySQL functions allowed in the following table are allowed in partitioning expressions.

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS()</td>
<td></td>
</tr>
<tr>
<td>CEILING() (see CEILING() and FLOOR())</td>
<td></td>
</tr>
<tr>
<td>DAY()</td>
<td></td>
</tr>
<tr>
<td>DAYOFMONTH()</td>
<td></td>
</tr>
<tr>
<td>DAYOFWEEK()</td>
<td></td>
</tr>
<tr>
<td>DAYOFYEAR()</td>
<td></td>
</tr>
<tr>
<td>DATEDIFF()</td>
<td></td>
</tr>
<tr>
<td>EXTRACT() (see EXTRACT() function with WEEK specifier)</td>
<td></td>
</tr>
<tr>
<td>FLOOR() (see CEILING() and FLOOR())</td>
<td></td>
</tr>
<tr>
<td>HOUR()</td>
<td></td>
</tr>
<tr>
<td>MICROSECOND()</td>
<td></td>
</tr>
<tr>
<td>MINUTE()</td>
<td></td>
</tr>
<tr>
<td>MOD()</td>
<td></td>
</tr>
<tr>
<td>MONTH()</td>
<td></td>
</tr>
<tr>
<td>QUARTER()</td>
<td></td>
</tr>
<tr>
<td>SECOND()</td>
<td></td>
</tr>
<tr>
<td>TIME_TO_SEC()</td>
<td></td>
</tr>
<tr>
<td>TO_DAYS()</td>
<td></td>
</tr>
<tr>
<td>UNIX_TIMESTAMP()</td>
<td>(permitted in MySQL 5.1.43 and later, with TIMESTAMP columns)</td>
</tr>
<tr>
<td>WEEKDAY()</td>
<td></td>
</tr>
<tr>
<td>YEAR()</td>
<td></td>
</tr>
<tr>
<td>YEARWEEK()</td>
<td></td>
</tr>
</tbody>
</table>

In MySQL 5.1, range optimization can be used only for the TO_DAYS() and YEAR() functions. See Partition Pruning, for more information.

CEILING() and FLOOR(). Each of these functions returns an integer only if it is passed an argument of an exact numeric type, such as one of the INT types or DECIMAL. This means, for example, that the following CREATE TABLE statement fails with an error, as shown here:

```
mysql> CREATE TABLE t (c FLOAT) PARTITION BY LIST( FLOOR(c) ) (  
    -> PARTITION p0 VALUES IN (1,3,5),  
    -> PARTITION p1 VALUES IN (2,4,6)  
  );
ERROR 1490 (HY000): The PARTITION function returns the wrong type
```

EXTRACT() function with WEEK specifier. The value returned by the EXTRACT() function, when used as EXTRACT(WEEK FROM col), depends on the value of the default_week_format system variable. For this reason, beginning with MySQL 5.1.55, EXTRACT() is no longer permitted as a partitioning function when it specifies the unit as WEEK. (Bug #54483)

See Mathematical Functions, for more information about the return types of these functions, as well as Numeric Types.

9.4 Partitioning and Table-Level Locking

For storage engines such as MyISAM that actually execute table-level locks when executing DML or DDL statements, such a statement affecting a partitioned table imposes a lock on the table as a whole; that is, all partitions are locked until the statement was finished. For example, a SELECT from a partitioned MyISAM table causes a lock on the entire table.
In practical terms, what this means is that the statements discussed later in this section tend to execute more slowly as the number of partitions increases. This limitation is greatly reduced in MySQL 5.6, with the introduction of *partition lock pruning* in MySQL 5.6.6.

This is not true for statements effecting partitioned tables using storage engines such as *InnoDB*, that employ row-level locking and do not actually perform (or need to perform) the locks prior to partition pruning.

The next few paragraphs discuss the effects of MySQL statements on partitioned tables using storage engines that employ table-level locks.

**DML statements**

- **SELECT** statements lock the entire table. **SELECT** statements containing unions or joins lock all tables named in the union or join.
- **UPDATE** also locks the entire table.
- **REPLACE** and **INSERT** (including **INSERT ... ON DUPLICATE KEY UPDATE**) lock the entire table.
- **INSERT ... SELECT** locks both the source table and the target table.

**Note**

**INSERT DELAYED** is not supported for partitioned tables.

A **LOAD DATA** statement on a partitioned table locks the entire table.

A trigger on a partitioned table, once activated, locks the entire table.

**DDL statements**

- **CREATE VIEW** causes a lock on any partitioned table from which it reads.
- **ALTER TABLE** locks the affected partitioned table.

**Other statements**

- **LOCK TABLES** locks all partitions of a partitioned table.

Evaluating the **expr** in a **CALL stored_procedure(expr)** statement locks all partitions of any partitioned table referenced by **expr**.

**ALTER TABLE** also takes a metadata lock on the table level.
Chapter 10 MySQL Differences from Standard SQL

Table of Contents

10.1 SELECT INTO TABLE ................................................................. 45
10.2 UPDATE .............................................................................. 45
10.3 Transactions and Atomic Operations ........................................... 45
10.4 Foreign Key Differences ............................................................. 48
10.5 '--' as the Start of a Comment ................................................... 48

We try to make MySQL Server follow the ANSI SQL standard and the ODBC SQL standard, but MySQL Server performs operations differently in some cases:

• There are several differences between the MySQL and standard SQL privilege systems. For example, in MySQL, privileges for a table are not automatically revoked when you delete a table. You must explicitly issue a `REVOKE` statement to revoke privileges for a table. For more information, see REVOKE Syntax.

• The `CAST()` function does not support cast to `REAL` or `BIGINT`. See Cast Functions and Operators.

10.1 SELECT INTO TABLE

MySQL Server doesn't support the `SELECT ... INTO TABLE` Sybase SQL extension. Instead, MySQL Server supports the `INSERT INTO ... SELECT` standard SQL syntax, which is basically the same thing. See INSERT ... SELECT Syntax. For example:

```
INSERT INTO tbl_temp2 (fld_id)
SELECT tbl_temp1.fld_order_id
FROM tbl_temp1 WHERE tbl_temp1.fld_order_id > 100;
```

Alternatively, you can use `SELECT ... INTO OUTFILE` or `CREATE TABLE ... SELECT`.

You can use `SELECT ... INTO` with user-defined variables. The same syntax can also be used inside stored routines using cursors and local variables. See SELECT ... INTO Syntax.

10.2 UPDATE

If you access a column from the table to be updated in an expression, `UPDATE` uses the current value of the column. The second assignment in the following statement sets `col2` to the current (updated) `col1` value, not the original `col1` value. The result is that `col1` and `col2` have the same value. This behavior differs from standard SQL.

```
UPDATE t1 SET col1 = col1 + 1, col2 = col1;
```

10.3 Transactions and Atomic Operations

MySQL Server (version 3.23-max and all versions 4.0 and above) supports transactions with the InnoDB transactional storage engine. InnoDB provides full ACID compliance. See Storage Engines. For information about InnoDB differences from standard SQL with regard to treatment of transaction errors, see InnoDB Error Handling.
The other nontransactional storage engines in MySQL Server (such as MyISAM) follow a different paradigm for data integrity called “atomic operations.” In transactional terms, MyISAM tables effectively always operate in `autocommit = 1` mode. Atomic operations often offer comparable integrity with higher performance.

Because MySQL Server supports both paradigms, you can decide whether your applications are best served by the speed of atomic operations or the use of transactional features. This choice can be made on a per-table basis.

As noted, the tradeoff for transactional versus nontransactional storage engines lies mostly in performance. Transactional tables have significantly higher memory and disk space requirements, and more CPU overhead. On the other hand, transactional storage engines such as InnoDB also offer many significant features. MySQL Server's modular design enables the concurrent use of different storage engines to suit different requirements and deliver optimum performance in all situations.

But how do you use the features of MySQL Server to maintain rigorous integrity even with the nontransactional MyISAM tables, and how do these features compare with the transactional storage engines?

• If your applications are written in a way that is dependent on being able to call `ROLLBACK` rather than `COMMIT` in critical situations, transactions are more convenient. Transactions also ensure that unfinished updates or corrupting activities are not committed to the database; the server is given the opportunity to do an automatic rollback and your database is saved.

If you use nontransactional tables, you must resolve potential problems at the application level by including simple checks before updates and by running simple scripts that check the databases for inconsistencies and automatically repair or warn if such an inconsistency occurs. You can normally fix tables perfectly with no data integrity loss just by using the MySQL log or even adding one extra log.

• More often than not, critical transactional updates can be rewritten to be atomic. Generally speaking, all integrity problems that transactions solve can be done with `LOCK TABLES` or atomic updates, ensuring that there are no automatic aborts from the server, which is a common problem with transactional database systems.

• To be safe with MySQL Server, regardless of whether you use transactional tables, you only need to have backups and have binary logging turned on. When that is true, you can recover from any situation that you could with any other transactional database system. It is always good to have backups, regardless of which database system you use.

The transactional paradigm has its advantages and disadvantages. Many users and application developers depend on the ease with which they can code around problems where an abort appears to be necessary, or is necessary. However, even if you are new to the atomic operations paradigm, or more familiar with transactions, do consider the speed benefit that nontransactional tables can offer on the order of three to five times the speed of the fastest and most optimally tuned transactional tables.

In situations where integrity is of highest importance, MySQL Server offers transaction-level reliability and integrity even for nontransactional tables. If you lock tables with `LOCK TABLES`, all updates stall until integrity checks are made. If you obtain a `READ LOCAL` lock (as opposed to a write lock) for a table that enables concurrent inserts at the end of the table, reads are permitted, as are inserts by other clients. The newly inserted records are not be seen by the client that has the read lock until it releases the lock. With `INSERT DELAYED`, you can write inserts that go into a local queue until the locks are released, without having the client wait for the insert to complete. See Concurrent Inserts, and INSERT DELAYED Syntax.

“Atomic,” in the sense that we mean it, is nothing magical. It only means that you can be sure that while each specific update is running, no other user can interfere with it, and there can never be an automatic
rollback (which can happen with transactional tables if you are not very careful). MySQL Server also guarantees that there are no dirty reads.

Following are some techniques for working with nontransactional tables:

- Loops that need transactions normally can be coded with the help of LOCK TABLES, and you don't need cursors to update records on the fly.

- To avoid using ROLLBACK, you can employ the following strategy:
  1. Use LOCK TABLES to lock all the tables you want to access.
  2. Test the conditions that must be true before performing the update.
  3. Update if the conditions are satisfied.
  4. Use UNLOCK TABLES to release your locks.

This is usually a much faster method than using transactions with possible rollbacks, although not always. The only situation this solution doesn't handle is when someone kills the threads in the middle of an update. In that case, all locks are released but some of the updates may not have been executed.

- You can also use functions to update records in a single operation. You can get a very efficient application by using the following techniques:
  - Modify columns relative to their current value.
  - Update only those columns that actually have changed.

  For example, when we are updating customer information, we update only the customer data that has changed and test only that none of the changed data, or data that depends on the changed data, has changed compared to the original row. The test for changed data is done with the WHERE clause in the UPDATE statement. If the record wasn't updated, we give the client a message: “Some of the data you have changed has been changed by another user.” Then we show the old row versus the new row in a window so that the user can decide which version of the customer record to use.

  This gives us something that is similar to column locking but is actually even better because we only update some of the columns, using values that are relative to their current values. This means that typical UPDATE statements look something like these:

  ```
  UPDATE tablename SET pay_back=pay_back+125;
  UPDATE customer
    SET
      customer_date='current_date',
      address='new address',
      phone='new phone',
      money_owed_to_us=money_owed_to_us-125
    WHERE
      customer_id=id AND address='old address' AND phone='old phone';
  ```

  This is very efficient and works even if another client has changed the values in the pay_back or money_owed_to_us columns.

- When managing unique identifiers, you can avoid statements such as LOCK TABLES or ROLLBACK by using an AUTO_INCREMENT column and either the LAST_INSERT_ID() SQL function or the mysql_insert_id() C API function. See Information Functions, and mysql_insert_id().
For situations that require row-level locking, use InnoDB tables. Otherwise, with MyISAM tables, you can use a flag column in the table and do something like the following:

```
UPDATE tbl_name SET row_flag=1 WHERE id=ID;
```

MySQL returns 1 for the number of affected rows if the row was found and row_flag wasn't 1 in the original row. You can think of this as though MySQL Server changed the preceding statement to:

```
UPDATE tbl_name SET row_flag=1 WHERE id=ID AND row_flag <> 1;
```

### 10.4 Foreign Key Differences

MySQL’s implementation of foreign keys differs from the SQL standard in the following key respects:

- If there are several rows in the parent table that have the same referenced key value, InnoDB acts in foreign key checks as if the other parent rows with the same key value do not exist. For example, if you have defined a RESTRICT type constraint, and there is a child row with several parent rows, InnoDB does not permit the deletion of any of those parent rows.

  InnoDB performs cascading operations through a depth-first algorithm, based on records in the indexes corresponding to the foreign key constraints.

- A FOREIGN KEY constraint that references a non-UNIQUE key is not standard SQL but rather an InnoDB extension.

- If ON UPDATE CASCADE or ON UPDATE SET NULL recurses to update the same table it has previously updated during the same cascade, it acts like RESTRICT. This means that you cannot use self-referential ON UPDATE CASCADE or ON UPDATE SET NULL operations. This is to prevent infinite loops resulting from cascaded updates. A self-referential ON DELETE SET NULL, on the other hand, is possible, as is a self-referential ON DELETE CASCADE. Cascading operations may not be nested more than 15 levels deep.

- In an SQL statement that inserts, deletes, or updates many rows, foreign key constraints (like unique constraints) are checked row-by-row. When performing foreign key checks, InnoDB sets shared row-level locks on child or parent records that it must examine. MySQL checks foreign key constraints immediately; the check is not deferred to transaction commit. According to the SQL standard, the default behavior should be deferred checking. That is, constraints are only checked after the entire SQL statement has been processed. This means that it is not possible to delete a row that refers to itself using a foreign key.

For information about how the InnoDB storage engine handles foreign keys, see InnoDB and FOREIGN KEY Constraints.

### 10.5 '--' as the Start of a Comment

Standard SQL uses the C syntax /* this is a comment */ for comments, and MySQL Server supports this syntax as well. MySQL also support extensions to this syntax that enable MySQL-specific SQL to be embedded in the comment, as described in Comment Syntax.

Standard SQL uses “--” as a start-comment sequence. MySQL Server uses “#” as the start comment character. MySQL Server 3.23.3 and up also supports a variant of the “--” comment style. That is, the “--” start-comment sequence must be followed by a space (or by a control character such as a newline). The space is required to prevent problems with automatically generated SQL queries that use constructs such as the following, where we automatically insert the value of the payment for payment:
Consider about what happens if `payment` has a negative value such as `-1`:

```sql
UPDATE account SET credit=credit--1
```

`credit--1` is a valid expression in SQL, but `--` is interpreted as the start of a comment, part of the expression is discarded. The result is a statement that has a completely different meaning than intended:

```sql
UPDATE account SET credit=credit
```

The statement produces no change in value at all. This illustrates that permitting comments to start with `--` can have serious consequences.

Using our implementation requires a space following the `--` for it to be recognized as a start-comment sequence in MySQL Server 3.23.3 and newer. Therefore, `credit--1` is safe to use.

Another safe feature is that the `mysql` command-line client ignores lines that start with `--`.

The following information is relevant only if you are running a MySQL version earlier than 3.23.3:

If you have an SQL script in a text file that contains `--` comments, you should use the `replace` utility as follows to convert the comments to use `#` characters before executing the script:

```
shell> replace " --" " #" < text-file-with-funny-comments.sql \ | mysql db_name
```

That is safer than executing the script in the usual way:

```
shell> mysql db_name < text-file-with-funny-comments.sql
```

You can also edit the script file “in place” to change the `--` comments to `#` comments:

```
shell> replace " --" " #" -- text-file-with-funny-comments.sql
```

Change them back with this command:

```
shell> replace " #" " --" -- text-file-with-funny-comments.sql
```

See `replace — A String-Replacement Utility`.
Chapter 11 Known Issues in MySQL

This section lists known issues in recent versions of MySQL.

For information about platform-specific issues, see the installation and porting instructions in General Installation Guidance, and Debugging and Porting MySQL.

The following problems are known:

- Subquery optimization for `IN` is not as effective as for `=`.
- Even if you use `lower_case_table_names=2` (which enables MySQL to remember the case used for databases and table names), MySQL does not remember the case used for database names for the function `DATABASE()` or within the various logs (on case-insensitive systems).
- Dropping a `FOREIGN KEY` constraint does not work in replication because the constraint may have another name on the slave.
- `REPLACE` (and `LOAD DATA` with the `REPLACE` option) does not trigger `ON DELETE CASCADE`.
- `DISTINCT` with `ORDER BY` does not work inside `GROUP_CONCAT()` if you do not use all and only those columns that are in the `DISTINCT` list.
- If one user has a long-running transaction and another user drops a table that is updated in the transaction, there is small chance that the binary log may contain the `DROP TABLE` statement before the table is used in the transaction itself.
- When inserting a big integer value (between $2^{63}$ and $2^{64}-1$) into a decimal or string column, it is inserted as a negative value because the number is evaluated in a signed integer context.
- `FLUSH TABLES WITH READ LOCK` does not block `COMMIT` if the server is running without binary logging, which may cause a problem (of consistency between tables) when doing a full backup.
- `ANALYZE TABLE`, `OPTIMIZE TABLE`, and `REPAIR TABLE` may cause problems on tables for which you are using `INSERT DELAYED`.
- Performing `LOCK TABLE ...` and `FLUSH TABLES ...` does not guarantee that there isn't a half-finished transaction in progress on the table.
- With statement-based binary logging, the master writes the executed queries to the binary log. This is a very fast, compact, and efficient logging method that works perfectly in most cases. However, it is possible for the data on the master and slave to become different if a query is designed in such a way that the data modification is nondeterministic (generally not a recommended practice, even outside of replication).

For example:

- `CREATE TABLE ... SELECT` or `INSERT ... SELECT` statements that insert zero or `NULL` values into an `AUTO_INCREMENT` column.
- `DELETE` if you are deleting rows from a table that has foreign keys with `ON DELETE CASCADE` properties.
- `REPLACE ... SELECT, INSERT IGNORE ... SELECT` if you have duplicate key values in the inserted data.

**If and only if** the preceding queries have no `ORDER BY` clause guaranteeing a deterministic order.
For example, for `INSERT ... SELECT` with no `ORDER BY`, the `SELECT` may return rows in a different order (which results in a row having different ranks, hence getting a different number in the `AUTO_INCREMENT` column), depending on the choices made by the optimizers on the master and slave.

A query is optimized differently on the master and slave only if:

- The table is stored using a different storage engine on the master than on the slave. (It is possible to use different storage engines on the master and slave. For example, you can use InnoDB on the master, but MyISAM on the slave if the slave has less available disk space.)
- MySQL buffer sizes (`key_buffer_size`, and so on) are different on the master and slave.
- The master and slave run different MySQL versions, and the optimizer code differs between these versions.

This problem may also affect database restoration using `mysqlbinlog|mysql`.

The easiest way to avoid this problem is to add an `ORDER BY` clause to the aforementioned nondeterministic queries to ensure that the rows are always stored or modified in the same order. Using row-based or mixed logging format also avoids the problem.

- Log file names are based on the server host name if you do not specify a file name with the startup option. To retain the same log file names if you change your host name to something else, you must explicitly use options such as `--log-bin=old_host_name-bin`. See Server Command Options. Alternatively, rename the old files to reflect your host name change. If these are binary logs, you must edit the binary log index file and fix the binary log file names there as well. (The same is true for the relay logs on a slave server.)

- `mysqlbinlog` does not delete temporary files left after a `LOAD DATA INFILE` statement. See `mysqlbinlog — Utility for Processing Binary Log Files`.

- `RENAME` does not work with `TEMPORARY` tables or tables used in a `MERGE` table.

- Due to the way table format (.frm) files are stored, you cannot use character 255 (`CHAR(255)`) in table names, column names, or enumerations.

- When using `SET CHARACTER SET`, you cannot use translated characters in database, table, and column names.

- You cannot use “_” or “%” with `ESCAPE` in `LIKE ... ESCAPE`.

- The server uses only the first `max_sort_length` bytes when comparing data values. This means that values cannot reliably be used in `GROUP BY`, `ORDER BY`, or `DISTINCT` if they differ only after the first `max_sort_length` bytes. To work around this, increase the variable value. The default value of `max_sort_length` is 1024 and can be changed at server startup time or at runtime.

- Numeric calculations are done with `BIGINT` or `DOUBLE` (both are normally 64 bits long). Which precision you get depends on the function. The general rule is that bit functions are performed with `BIGINT` precision, `IF()` and `ELT()` with `BIGINT` or `DOUBLE` precision, and the rest with `DOUBLE` precision. You should try to avoid using unsigned long long values if they resolve to be larger than 63 bits (9223372036854775807) for anything other than bit fields.

- You can have up to 255 `ENUM` and `SET` columns in one table.

- In `MIN()`, `MAX()`, and other aggregate functions, MySQL currently compares `ENUM` and `SET` columns by their string value rather than by the string's relative position in the set.
• In an `UPDATE` statement, columns are updated from left to right. If you refer to an updated column, you get the updated value instead of the original value. For example, the following statement increments `KEY` by 2, not 1:

```sql
mysql> UPDATE tbl_name SET KEY=KEY+1,KEY=KEY+1;
```

• You can refer to multiple temporary tables in the same query, but you cannot refer to any given temporary table more than once. For example, the following does not work:

```sql
mysql> SELECT * FROM temp_table, temp_table AS t2;
ERROR 1137: Can't reopen table: 'temp_table'
```

• The optimizer may handle `DISTINCT` differently when you are using “hidden” columns in a join than when you are not. In a join, hidden columns are counted as part of the result (even if they are not shown), whereas in normal queries, hidden columns do not participate in the `DISTINCT` comparison.

An example of this is:

```sql
SELECT DISTINCT mp3id FROM band_downloads
WHERE userid = 9 ORDER BY id DESC;
```

and

```sql
SELECT DISTINCT band_downloads.mp3id
FROM band_downloads,band_mp3
WHERE band_downloads.userid = 9
AND band_mp3.id = band_downloads.mp3id
ORDER BY band_downloads.id DESC;
```

In the second case, using MySQL Server 3.23.x, you may get two identical rows in the result set (because the values in the hidden `id` column may differ).

Note that this happens only for queries that do not have the `ORDER BY` columns in the result.

• If you execute a `PROCEDURE` on a query that returns an empty set, in some cases the `PROCEDURE` does not transform the columns.

• Creation of a table of type `MERGE` does not check whether the underlying tables are compatible types.

• If you use `ALTER TABLE` to add a `UNIQUE` index to a table used in a `MERGE` table and then add a normal index on the `MERGE` table, the key order is different for the tables if there was an old, non-`UNIQUE` key in the table. This is because `ALTER TABLE` puts `UNIQUE` indexes before normal indexes to be able to detect duplicate keys as early as possible.
Appendix A Licenses for Third-Party Components

Table of Contents

A.1 ANTLR 3 License ................................................................. 57
A.2 dtoa.c License .................................................................. 57
A.3 Editline Library (libedit) License ........................................... 58
A.4 FindGTest.cmake License ...................................................... 60
A.5 Fred Fish’s Dbug Library License ............................................ 61
A.6 getarg License ................................................................. 62
A.7 GNU General Public License Version 2.0, June 1991 ................ 62
A.8 GNU Lesser General Public License Version 2.1, February 1999 .. 68
A.9 GNU Libtool License .......................................................... 76
A.10 GNU Readline License ....................................................... 76
A.11 Google Controlling Master Thread I/O Rate Patch License ....... 77
A.12 Google Perftools (TCMalloc utility) License ......................... 77
A.13 Google SMP Patch License ................................................. 78
A.14 lib_sql.cc License ........................................................... 79
A.15 libevent License ............................................................. 79
A.16 Linux-PAM License .......................................................... 81
A.17 md5 (Message-Digest Algorithm 5) License ......................... 82
A.18 memcached License .......................................................... 82
A.19 nt_servc (Windows NT Service class library) License .......... 83
A.20 OpenPAM License ........................................................... 83
A.21 Paramiko License ............................................................ 83
A.22 Percona Multiple I/O Threads Patch License ......................... 84
A.23 RegEX-Spencer Library License ........................................... 84
A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License ........ 85
A.25 Richard A. O’Keefe String Library License ......................... 85
A.26 SHA-1 in C License .......................................................... 86
A.27 zlib License ................................................................. 86

The following is a list of the libraries we have included with the MySQL Server source and components used to test MySQL. We are thankful to all individuals that have created these. Some of the components require that their licensing terms be included in the documentation of products that include them. Cross references to these licensing terms are given with the applicable items in the list.

- Bjorn Benson

  For his safe_malloc (memory checker) package which is used in when you build MySQL using one of the
  BUILD/compile-*debug scripts or by manually setting the -DSAFEMALLOC flag.

- GroupLens Research Project

  The MySQL Quality Assurance team would like to acknowledge the use of the MovieLens Data Sets (10
  million ratings and 100,000 tags for 10681 movies by 71567 users) to help test MySQL products and to
  thank the GroupLens Research Project at the University of Minnesota for making the data sets available.

MySQL 5.1

- Section A.2, “dtoa.c License”
- Section A.3, “Editline Library (libedit) License”
MySQL Cluster 7.1

- Section A.1, “ANTLR 3 License”
- Section A.2, “dtoa.c License”
- Section A.3, “Editline Library (libedit) License”
- Section A.4, “FindGTest.cmake License”
- Section A.5, “Fred Fish’s Dbug Library License”
- Section A.6, “getarg License”
- Section A.7, “GNU General Public License Version 2.0, June 1991”
- Section A.8, “GNU Lesser General Public License Version 2.1, February 1999”
- Section A.9, “GNU Libtool License”
- Section A.10, “GNU Readline License”
- Section A.11, “Google Controlling Master Thread I/O Rate Patch License”
- Section A.12, “Google Perftools (TCMalloc utility) License”
- Section A.13, “Google SMP Patch License”
- Section A.14, “lib_sql.cc License”
- Section A.15, “libevent License”
- Section A.16, “md5 (Message-Digest Algorithm 5) License”
- Section A.17, “nt_servc (Windows NT Service class library) License”
- Section A.18, “Percona Multiple I/O Threads Patch License”
- Section A.19, “RegEX-Spencer Library License”
- Section A.20, “RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License”
- Section A.21, “Richard A. O’Keefe String Library License”
- Section A.22, “zlib License”
• Section A.16, “Linux-PAM License”
• Section A.17, “md5 (Message-Digest Algorithm 5) License”
• Section A.18, “memcached License”
• Section A.19, “nt_servc (Windows NT Service class library) License”
• Section A.20, “OpenPAM License”
• Section A.21, “Paramiko License”
• Section A.22, “Percona Multiple I/O Threads Patch License”
• Section A.23, “RegEX-Spencer Library License”
• Section A.25, “Richard A. O’Keefe String Library License”
• Section A.26, “SHA-1 in C License”
• Section A.27, “zlib License”

A.1 ANTLR 3 License

The following software may be included in this product:

ANTLR 3

ANTLR 3 License
[The BSD License]
Copyright (c) 2003-2007, Terence Parr
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.2 dtoa.c License

The following software may be included in this product:
The author of this software is David M. Gay.

Copyright (c) 1991, 2000, 2001 by Lucent Technologies.

Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

A.3 Editline Library (libedit) License

The following software may be included in this product:

Editline Library (libedit)

Some files are:

Copyright (c) 1992, 1993
The Regents of the University of California. All rights reserved.

This code is derived from software contributed to Berkeley by Christos Zoulas of Cornell University.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Some files are:
Copyright (c) 2001 The NetBSD Foundation, Inc.
All rights reserved.

This code is derived from software contributed to The NetBSD Foundation by Anthony Mallet.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Some files are:

Copyright (c) 1997 The NetBSD Foundation, Inc.
All rights reserved.

This code is derived from software contributed to The NetBSD Foundation by Jaromir Dolecek.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

Some files are:

Copyright (c) 1998 Todd C. Miller <Todd.Miller@courtesan.com>

Permission to use, copy, modify, and distribute this
software for any purpose with or without fee is hereby
granted, provided that the above copyright notice and
this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND TODD C. MILLER
DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS. IN NO EVENT SHALL TODD C. MILLER BE LIABLE
FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

A.4 FindGTest.cmake License

The following software may be included in this product:

FindGTest.cmake helper script (part of CMake)

Copyright 2009 Kitware, Inc.
Copyright 2009 Philip Lowman
Copyright 2009 Daniel Blezek

Distributed under the OSI-approved BSD License (the "License");
see accompanying file Copyright.txt for details.

This software is distributed WITHOUT ANY WARRANTY; without even the
implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the License for more information.

(To distributed this file outside of CMake, substitute the full
License text for the above reference.)

Thanks to Daniel Blezek for the GTEST_ADD_TESTS code

Text of Copyright.txt mentioned above:

CMake - Cross Platform Makefile Generator
Copyright 2000-2009 Kitware, Inc., Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the names of Kitware, Inc., the Insight Software Consortium,
  nor the names of their contributors may be used to endorse or promote
  products derived from this software without specific prior written
  permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.5 Fred Fish's Dbug Library License

The following software may be included in this product:

Fred Fish's Dbug Library

N O T I C E

Copyright Abandoned, 1987, Fred Fish

This previously copyrighted work has been placed into the public
domain by the author and may be freely used for any purpose,
private or commercial.

Because of the number of inquiries I was receiving about the use
of this product in commercially developed works I have decided to
simply make it public domain to further its unrestricted use. I
specifically would be most happy to see this material become a
part of the standard Unix distributions by AT&T and the Berkeley
Computer Science Research Group, and a standard part of the GNU
system from the Free Software Foundation.

I would appreciate it, as a courtesy, if this notice is left in
all copies and derivative works. Thank you.

The author makes no warranty of any kind with respect to this
A.6 getarg License

The following software may be included in this product:

getarg Function (getarg.h, getarg.c files)

Copyright (c) 1997 – 2000 Kungliga Tekniska Högskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Redistribution and use in source and binary forms, with
or without modification, are permitted provided that the
following conditions are met:

1. Redistributions of source code must retain the above
copyright notice, this list of conditions and the
following disclaimer.
2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following
disclaimer in the documentation and/or other materials
provided with the distribution.
3. Neither the name of the Institute nor the names of its
contributors may be used to endorse or promote products
derived from this software without specific prior written
permission.

THIS SOFTWARE IS PROVIDED BY THE INSTITUTE AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE FOR ANY
DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

A.7 GNU General Public License Version 2.0, June 1991

The following applies to all products licensed under the GNU General
Public License, Version 2.0: You may not use the identified files
except in compliance with the GNU General Public License, Version
2.0 (the "License.") You may obtain a copy of the License at
http://www.gnu.org/licenses/gpl-2.0.txt. A copy of the license is
also reproduced below. Unless required by applicable law or agreed
to in writing, software distributed under the License is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
either express or implied. See the License for the specific language
governing permissions and limitations under the License.

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim
copies of this license document, but changing it is not
allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

We protect your rights with two steps: (1) copyright the software,
and (2) offer you this license which gives you legal permission to
copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on,
we want its recipients to know that what they have is not the original,
so that any problems introduced by others will not reflect on the
original authors' reputations.

Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and
modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License.  The "Program", below,
refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification"). Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

   b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

   c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under
the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

   a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

   c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<pre>
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

A.8 GNU Lesser General Public License Version 2.1, February 1999

The following applies to all products licensed under the GNU Lesser General Public License, Version 2.1: You may not use the identified files except in compliance with the GNU Lesser General Public License, Version 2.1 (the "License"). You may obtain a copy of the License at http://www.gnu.org/licenses/lgpl-2.1.html. A copy of the license is also reproduced below. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

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

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights. These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you. You must make sure that they, too, receive or can get the source
code. If you link other code with the library, you must provide
complete object files to the recipients, so that they can re-link them
with the library after making changes to the library and recompiling
it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that
there is no warranty for the free library. Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author’s reputation will not be affected by problems that might be
introduced by others.

Finally, software patents pose a constant threat to the existence of
any free program. We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder. Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License. This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License. We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library. The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom. The Lesser General
Public License permits more lax criteria for linking other code with
the library.

We call this license the "Lesser" General Public License because it
does Less to protect the user’s freedom than the ordinary General
Public License. It also provides other free software developers Less
of an advantage over competing non-free programs. These disadvantages
are the reason we use the ordinary General Public License for many
libraries. However, the Lesser license provides advantages in certain
special circumstances.

For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it
becomes a de-facto standard. To achieve this, non-free programs
must be allowed to use the library. A more frequent case is that
a free library does the same job as widely used non-free libraries.
In this case, there is little to gain by limiting the free library
to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software. For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.
Although the Lesser General Public License is less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

   a) The modified work must itself be a software library.

   b) You must cause the files modified to carry prominent notices
c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to
72

distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public 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.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.
15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change. You can do so by permitting
redistribution under these terms (or, alternatively, under the terms
of the ordinary General Public License).

To apply these terms, attach the following notices to the library.
It is safest to attach them to the start of each source file to most
effectively convey the exclusion of warranty; and each file should
have at least the "copyright" line and a pointer to where the full
notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the
library 'Frob' (a library for tweaking knobs) written by James
Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice
That's all there is to it!

A.9 GNU Libtool License

The following software may be included in this product:

GNU Libtool (The GNU Portable Library Tool)

If you are receiving a copy of the Oracle software in source code, you are also receiving a copy of two files (ltmain.sh and ltdl.h) generated by the GNU Libtool in source code. If you received the Oracle software under a license other than a commercial (non-GPL) license, then the terms of the Oracle license do NOT apply to these files from GNU Libtool; they are licensed under the following licenses, separately from the Oracle programs you receive.

Oracle elects to use GNU General Public License version 2 (GPL) for any software where a choice of GPL or GNU Lesser/Library General Public License (LGPL) license versions are made available with the language indicating that GPL/LGPL or any later version may be used, or where a choice of which version of the GPL/LGPL is applied is unspecified.

From GNU Libtool:

ltmain.sh - Provide generalized library-building support services.

NOTE: Changing this file will not affect anything until you rerun configure.


Originally by Gordon Matzigkeit, 1996

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

As a special exception to the GNU General Public License, if you distribute this file as part of a program that contains a configuration script generated by Autoconf, you may include it under the same distribution terms that you use for the rest of that program.

This component is licensed under Section A.7, “GNU General Public License Version 2.0, June 1991”

A.10 GNU Readline License

The following software may be included in this product:
GNU Readline Library

GNU Readline Library
With respect to MySQL Server/Cluster software licensed under GNU General Public License, you are receiving a copy of the GNU Readline Library in source code. The terms of any Oracle license that might accompany the Oracle programs do NOT apply to the GNU Readline Library; it is licensed under the following license, separately from the Oracle programs you receive. Oracle elects to use GNU General Public License version 2 (GPL) for any software where a choice of GPL license versions are made available with the language indicating that GPLv2 or any later version may be used, or where a choice of which version of the GPL is applied is unspecified.

This component is licensed under Section A.7, "GNU General Public License Version 2.0, June 1991"

A.11 Google Controlling Master Thread I/O Rate Patch License

The following software may be included in this product:

Google Controlling master thread I/O rate patch

Copyright (c) 2009, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.12 Google Perftools (TCMalloc utility) License

The following software may be included in this product:

Google Perftools (TCMalloc utility)

Copyright (c) 1998-2006, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.13 Google SMP Patch License

The following software may be included in this product:

Google SMP Patch

Copyright (c) 2008, Google Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of the Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
A.14 lib_sql.cc License

The following software may be included in this product:

lib_sql.cc

Copyright (c) 2000
SWsoft company

This material is provided "as is", with absolutely no warranty expressed or implied. Any use is at your own risk.

Permission to use or copy this software for any purpose is hereby granted without fee, provided the above notices are retained on all copies. Permission to modify the code and to distribute modified code is granted, provided the above notices are retained, and a notice that the code was modified is included with the above copyright notice.

This code was modified by the MySQL team.

A.15 libevent License

The following software may be included in this product:

libevent

Copyright (c) 2000–2007 Niels Provos <provos@citi.umich.edu>
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR `\`AS IS\'\' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

==
Parts developed by Adam Langley
==
==
log.c
Based on err.c, which was adapted from OpenBSD libc *err*warncode.

Copyright (c) 2005 Nick Mathewson
Copyright (c) 2000 Dug Song
Copyright (c) 1993 The Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

min_heap.h

Copyright (c) 2006 Maxim Yegorushkin
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

win32.c

Copyright 2000-2002 Niels Provos
Copyright 2003 Michael A. Davis
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.16 Linux-PAM License

The following software may be included in this product:

Linux-PAM (pam-devel, Pluggable authentication modules for Linux)

Copyright Theodore Ts'o, 1996. All rights reserved.

(For the avoidance of doubt, Oracle uses and distributes this component under the terms below and elects not to do so under the GPL even though the GPL is referenced as an option below.)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, and the entire permission notice in its entirety, including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

ALTERNATIVELY, this product may be distributed under the terms of the GNU Public License, in which case the provisions of the GPL are required INSTEAD OF the above restrictions. (This clause is necessary due to a potential bad interaction between the GPL and the restrictions contained in a BSD-style copyright.)

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
A.17 md5 (Message-Digest Algorithm 5) License

The following software may be included in this product:

md5 (Message-Digest Algorithm 5)

This code implements the MD5 message-digest algorithm. The algorithm is due to Ron Rivest. This code was written by Colin Plumb in 1993, no copyright is claimed. This code is in the public domain; do with it what you wish.

Equivalent code is available from RSA Data Security, Inc. This code has been tested against that, and is equivalent, except that you don't need to include two pages of legalese with every copy.

The code has been modified by Mikael Ronstroem to handle calculating a hash value of a key that is always a multiple of 4 bytes long. Word 0 of the calculated 4-word hash value is returned as the hash value.

A.18 memcached License

The following software may be included in this product:

memcached

Copyright (c) 2003, Danga Interactive, Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the Danga Interactive nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
A.19 nt_servc (Windows NT Service class library) License

The following software may be included in this product:

nt_servc (Windows NT Service class library)

Windows NT Service class library
Copyright Abandoned 1998 Irena Pancirov - Irnet Snc
This file is public domain and comes with NO WARRANTY of any kind

A.20 OpenPAM License

The following software may be included in this product:

OpenPAM

Copyright (c) 2002-2003 Networks Associates Technology, Inc.
Copyright (c) 2004-2007 Dag-Erling Smørgrav
All rights reserved.

This software was developed for the FreeBSD Project by
ThinkSec AS and Network Associates Laboratories, the
Security Research Division of Network Associates, Inc.
derunder DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"),
as part of the DARPA CHATS research program.

Redistribution and use in source and binary forms,
with or without modification, are permitted provided
that the following conditions are met:

1. Redistributions of source code must retain the above
   copyright notice, this list of conditions and the
   following disclaimer.
2. Redistributions in binary form must reproduce the
   above copyright notice, this list of conditions and
   the following disclaimer in the documentation and/or
   other materials provided with the distribution.
3. The name of the author may not be used to endorse or
   promote products derived from this software without
   specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN
NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR
ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.21 Paramiko License

The following software may be included in this product:
Paramiko

You are receiving a copy of Paramiko in both source and object code. The terms of the Oracle license do NOT apply to the Paramiko program; it is licensed under the following license, separately from the Oracle programs you receive. If you do not wish to install this program, you may delete the Paramiko folder and all its contents.

This component is licensed under Section A.8, “GNU Lesser General Public License Version 2.1, February 1999”.

A.22 Percona Multiple I/O Threads Patch License

The following software may be included in this product:

Percona Multiple I/O threads patch

Copyright (c) 2008, 2009 Percona Inc
All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of Percona Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of Percona Inc.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

A.23 RegEX-Spencer Library License

The following software may be included in this product: Henry Spencer's Regular-Expression Library (RegEX-Spencer)

Copyright 1992, 1993, 1994 Henry Spencer. All rights reserved.
This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

Permission is granted to anyone to use this software for any purpose on any computer system, and to alter it and redistribute it, subject to the following restrictions:

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.

3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.

4. This notice may not be removed or altered.

A.24 RFC 3174 - US Secure Hash Algorithm 1 (SHA1) License

The following software may be included in this product:

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

RFC 3174 - US Secure Hash Algorithm 1 (SHA1)

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement
Funding for the RFC Editor function is currently provided by the Internet Society.

A.25 Richard A. O'Keefe String Library License

The following software may be included in this product:

Richard A. O'Keefe String Library

The Richard O'Keefe String Library is subject to the following notice:

These files are in the public domain. This includes getopt.c, which is the work of Henry Spencer, University of Toronto Zoology, who says of it "None of this software is derived from Bell software. I had no access to the source for Bell's versions at the time I wrote it. This software is hereby explicitly placed in the public domain."
Sha-1 in C License

It may be used for any purpose on any machine by anyone. I would greatly prefer it if *my* material received no military use.

The tctype.h file is subject to the following notice:

Copyright (C) 1998, 1999 by Pruet Boonma, all rights reserved.
Copyright (C) 1998 by Theppitak Karoonboonyanan, all rights reserved.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies.

Smaphan Raruenrom and Pruet Boonma makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

A.26 SHA-1 in C License

The following software may be included in this product:

SHA-1 in C

SHA-1 in C
By Steve Reid <steve@edmweb.com>
100% Public Domain

A.27 zlib License

The following software may be included in this product:

zlib

Oracle gratefully acknowledges the contributions of Jean-loup Gailly and Mark Adler in creating the zlib general purpose compression library which is used in this product.

zlib.h -- interface of the 'zlib' general purpose compression library
Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.3, July 18th, 2005
Copyright (C) 1995-2005 Jean-loup Gailly and Mark Adler

zlib.h -- interface of the 'zlib' general purpose compression library
version 1.2.5, April 19th, 2010
Copyright (C) 1995-2010 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty.
In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly jloup@gzip.org
Mark Adler madler@alumni.caltech.edu