Cool new things in MySQL 5.0

Stewart Smith

April 5th, 2005

MySQL AB

5.0 At A Glance

- MySQL 5.0.3 Beta released
- FEDERATED Storage Engine
- Updatable Views
- Stored Procedures
- Cursors
- Rudimentary Triggers
- Explicit RTREE indexes on MyISAM tables
- Dynamic length rows for MEMORY tables
- BIT data type
- INFORMATION_SCHEMA
- True VARCHAR
- Speed improvements (galore!)

MySQL 5.0.3 Beta

- Released 23rd March
- Current Binaries:
  - Linux (x86, IA64, Alpha, PowerPC, AMD64), Windows (x86), Solaris 8/9/10 (SPARC 32/64, x86), FreeBSD 4.x (x86), MacOS X (PowerPC), HPUX (11.00, 11.11, 11.23), AIX 5.2, QNX, SGI IRIX 6.5, DEC OSF 5.1
- More coming
  - apparently the build team requires sleep too.
- All tests pass
- Bugs still exist (varying severity)
- Will not release GA until there are no fatal run-time bugs.
- Please try it and report any bugs!

Storage Engines

- No one way of storing data is ideal for all tables
- Choice between different methods
  - at CREATE TABLE time
  - can ALTER later
  - default: “set storage_engine=ndbcluster;”
- Easy to code your own
- Many to choose from

Storage Engines (cont.)

- MyISAM
  - Faster than a speeding bullet!
  - No transactions
  - No foreign keys (yet)
  - Not crash safe (no logging)
  - FULLTEXT indexing
  - Can store indexes away from data (e.g. separate disk)
- MERGE
  - Merge two identical MyISAM tables into one
- MEMORY/HEAP
  - In-memory
  - **really** fast

Storage Engines (cont.)

- BDB
  - Use BDB as a store for your MySQL tables
  - an SQL interface on BDB
    - Just what the storage engine architecture is good for doing!
  - transactions
  - crash recovery
  - page-level locking
- EXAMPLE
  - does nothing.
  - good example code
- ARCHIVE
  - compressed tables (zlib)
  - no indexes, deletes, replaces or updates

Storage Engines (cont.)

- CSV
  - a CSV text file as a table
  - Good example code for a functional storage engine
  - Import your table directly into Excel!
- InnoDB
  - ACID
  - Foreign Keys
  - Terrabytes of data possible (64 current max)
  - Separate log files (could put on another disk)
  - Multiple storage files (add more disk, add more innodb storage space)
  - Row level locking
- Federated
- Cluster

FEDERATED storage engine

- Added in MySQL 5.0.3
- Accesses data in tables of remote databases rather than in local tables.
- In this first version, only to other MySQL servers

FEDERATED example

- MySQL Server 1

```
CREATE TABLE test_table (
    id     int(20) NOT NULL auto_increment,
    name   varchar(32) NOT NULL default '',
    other  int(20) NOT NULL default '0',
    PRIMARY KEY  (id),
    KEY name (name),
    KEY other_key (other)
)
ENGINE=MyISAM
DEFAULT CHARSET=latin1 ;
```

FEDERATED Example (cont)

- MySQL Server 2

```
CREATE TABLE federated_table (
    id     int(20) NOT NULL auto_increment,
    name   varchar(32) NOT NULL default '',
    other  int(20) NOT NULL default '0',
    PRIMARY KEY  (id),
    KEY name (name),
    KEY other_key (other)
)
ENGINE=FEDERATED
DEFAULT CHARSET=latin1
COMMENT='mysql://root@remote_host:9306/federated/test_table';
```

Updatable Views

- Views make (some) big queries a lot nicer
- MySQL has been previously criticized for not having views
- Views are updatable
- Real Cool Things(TM) are possible:

```
mysql> CREATE TABLE t (qty INT, price INT);
mysql> INSERT INTO t VALUES(3, 50);
mysql> CREATE VIEW v AS SELECT qty, price,
  qty*price AS value FROM t;
mysql> SELECT * FROM v;
+------+-------+-------+
| qty  | price | value |
+------+-------+-------+
|    3 |    50 |   150 |
+------+-------+-------+
```

Example real-world view

```
-- current_memberships
-- -------------------
--
-- current memberships of all members of all orgs,
--           of all types, which provide membership
CREATE or replace view current_memberships AS
select distinct
  members.*,
  orgs.id as org_id,
  orgs.name as org_name,
  member_types.id as member_type_id,
  member_types.type as member_type
from
  members,
  orgs,
  org_members,
  member_types
where
      members.id = org_members.member_id
  AND org_members.org_id = orgs.id
  AND org_members.member_type_id = member_types.id
  AND member_types.org_id = orgs.id
  and (
--
-- you are a current member if your membership:
-- - period started before now, and ends after now
-- - never ends, always existed
-- - started before now, never ends
-- - started at unknown, ends after now
(org_members.start_date < now()
  and org_members.expiry > now())
     or (org_members.start_date is null
  and org_members.expiry is null)
     or (org_members.start_date<now()
  and org_members.expiry is null)
     or (org_members.start_date is null
  and org_members.expiry>now())
    )
  AND member_types.validates_membership = true
  AND not exists (select member_id from org_members,member_types
where org_members.member_type_id = member_types.id
and org_members.member_id = members.id
and member_types.revokes_membership = true
and member_types.validates_membership = false
and (
--
-- you are a current member if your membership:
-- - period started before now, and ends after now
-- - never ends, always existed
-- - started before now, never ends
-- - started at unknown, ends after now
(org_members.start_date < now()
  and org_members.expiry > now())
     or (org_members.start_date is null
  and org_members.expiry is null)
     or (org_members.start_date<now()
  and org_members.expiry is null)
     or (org_members.start_date is null
  and org_members.expiry>now())
    )
)
;
```

→ select * from current_memberships;

Stored Procedures

- MySQL follows the SQL:2003 syntax for stored procedures, which is also used by IBM's DB2.
- One of the most requested features
- Things you can do
  - SQL queries
  - call SQL functions
  - have IN and OUT parameters
  - local variables
  - Use cursors
- Future enhancements
  - Framework to support external stored procedures (e.g. PHP)
  - Lifting of limitations with CREATE FUNCTION and some references to tables.

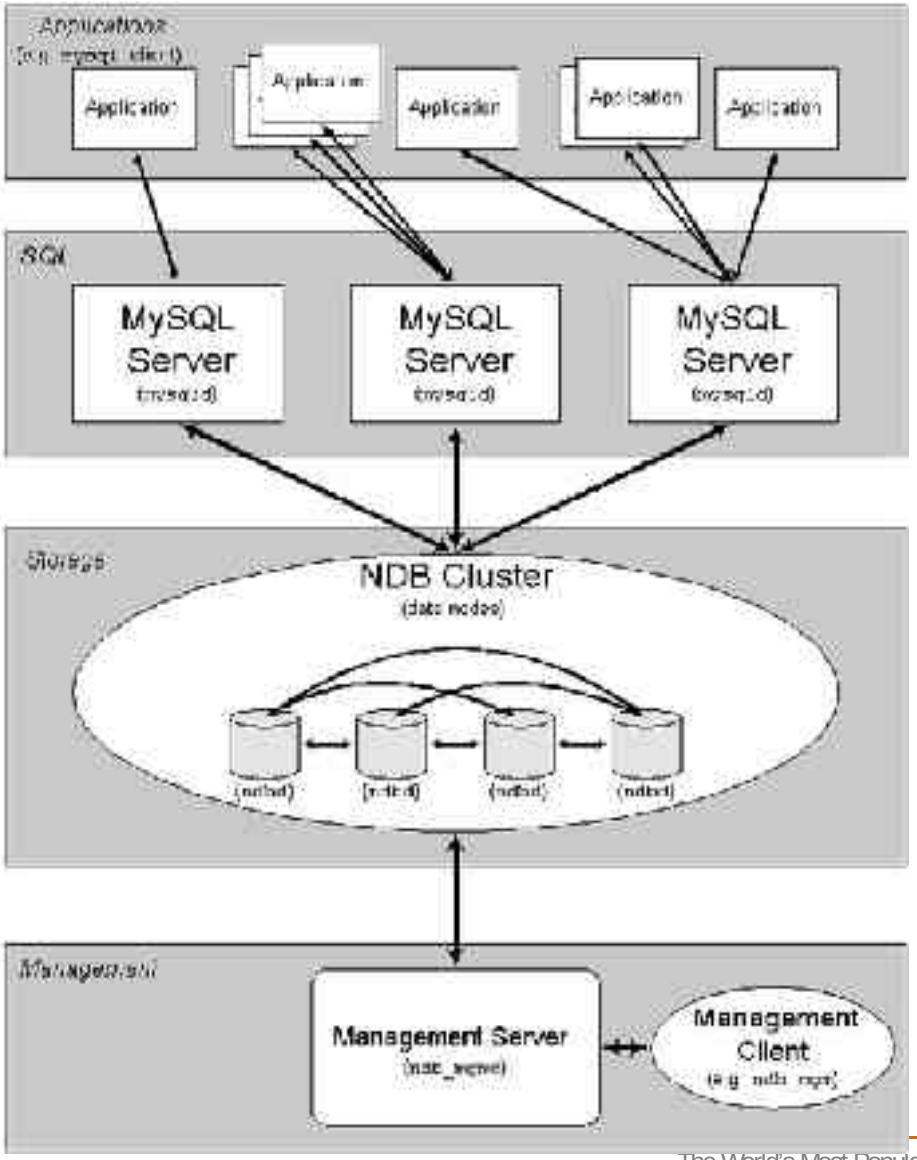
Stored Procedure Example

```
CREATE PROCEDURE curdemo()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE a CHAR(16);
  DECLARE b,c INT;
  DECLARE cur1 CURSOR FOR SELECT id,data FROM test.t1;
  DECLARE cur2 CURSOR FOR SELECT i FROM test.t2;
  DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;

  OPEN cur1;
  OPEN cur2;

  REPEAT
    FETCH cur1 INTO a, b;
    FETCH cur2 INTO c;
    IF NOT done THEN
       IF b < c THEN
          INSERT INTO test.t3 VALUES (a,b);
       ELSE
          INSERT INTO test.t3 VALUES (a,c);
       END IF;
    END IF;
  UNTIL done END REPEAT;

  CLOSE cur1;
  CLOSE cur2;
END
```

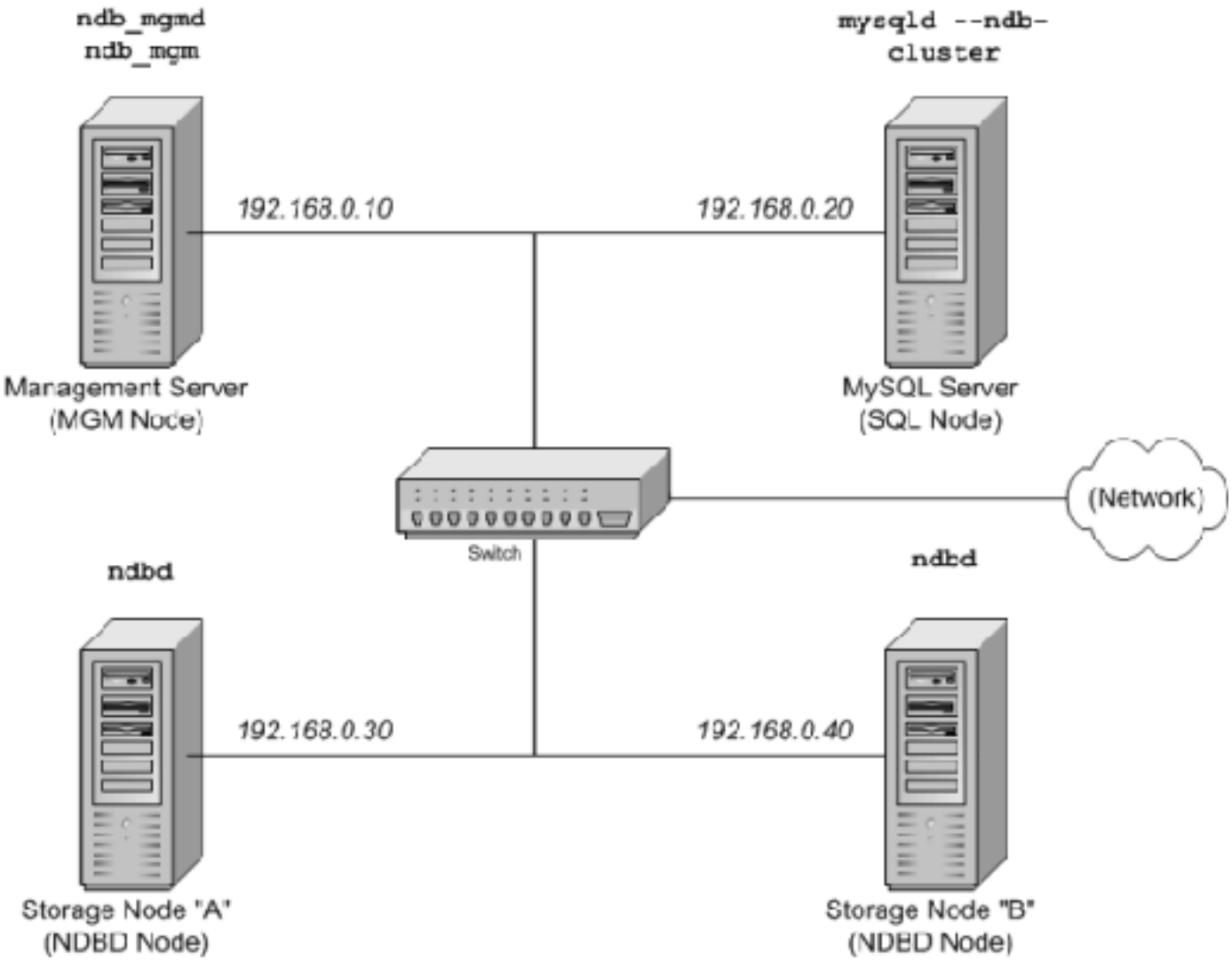
A Quick Introduction to MySQL Cluster

- NDB is a high availability, clustered, in-memory, share-nothing storage engine for MySQL
- Supports transactions
- Designed (and delivers) five 9's uptime
- Multiple MySQL Servers can connect to the one cluster
- Data is checkpointed to disk
- Online backup
- Support of multiple interconnects (TCP/IP, SCI and others)
- Part of the MAX builds (mysql-max, not MaxDB)
  - note that Cluster binaries are in separate RPMs

Cluster Architecture



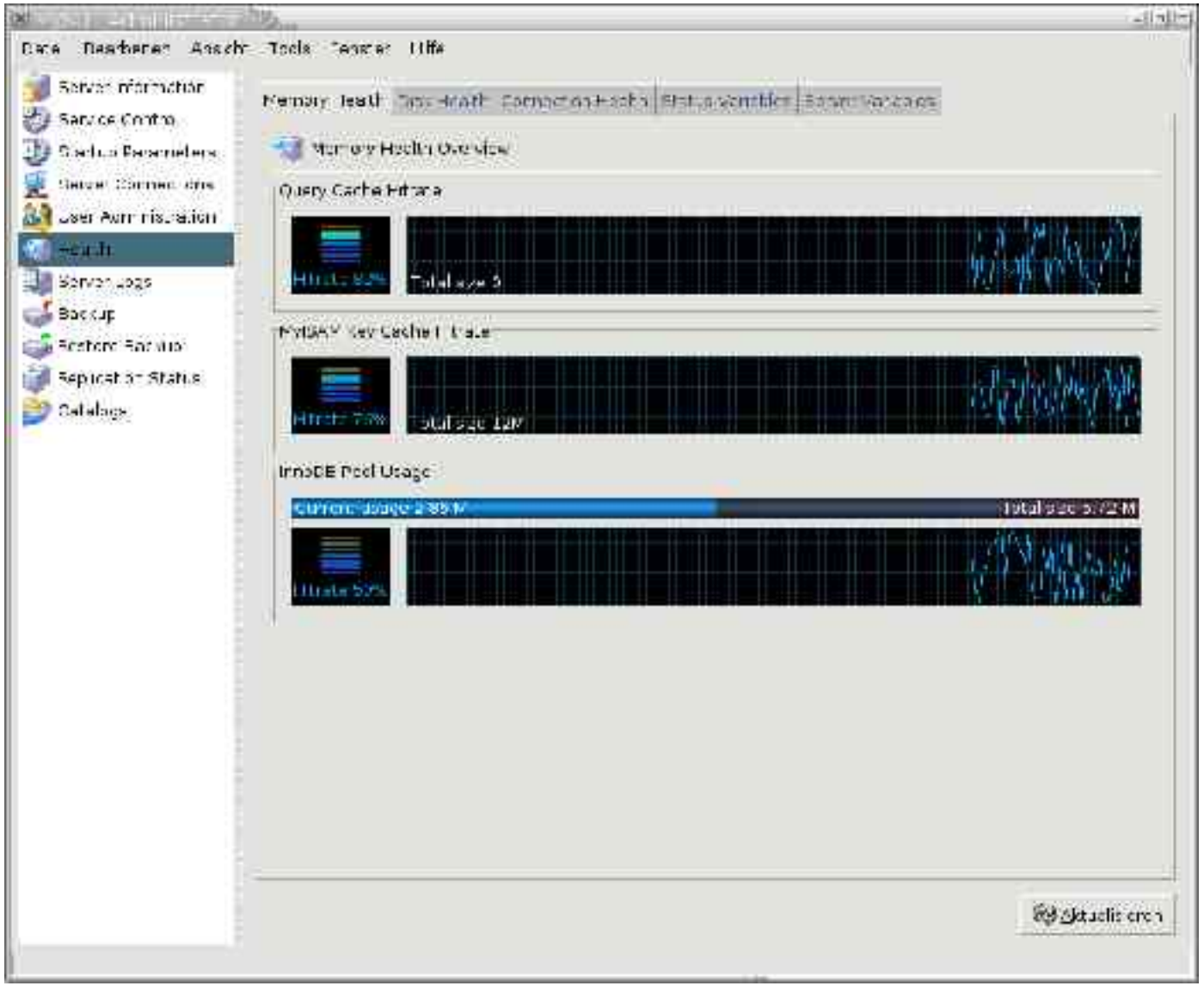
Example Cluster



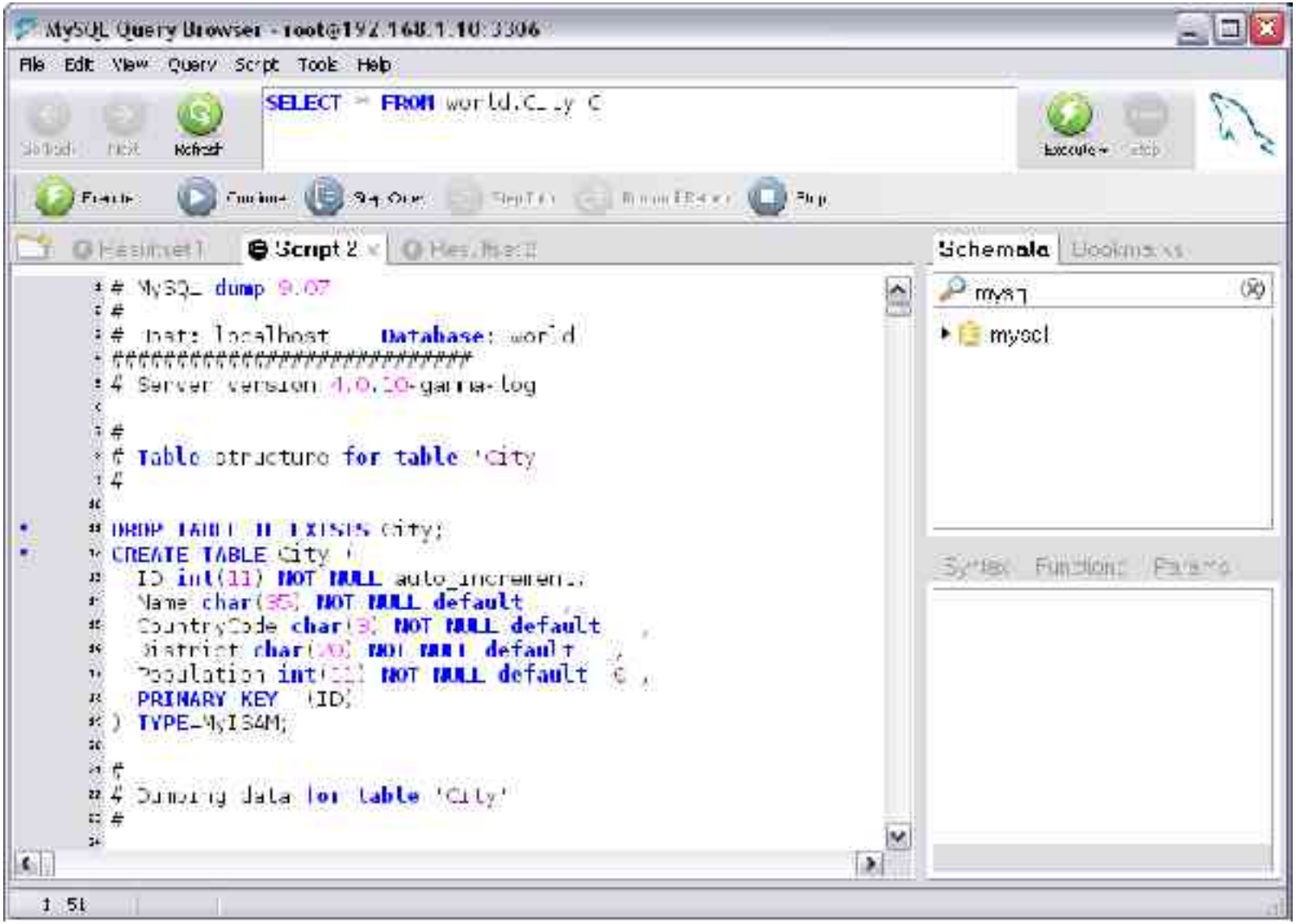
MySQL GUI Tools

- MySQL Administrator
  - Linux, Windows, MacOS X binaries available
- MySQL Query Browser
  - Linux and Windows binaries available
- MySQL Migration Tool
  - Java based migration tool
  - Windows Binaries available (1.0.1-alpha)

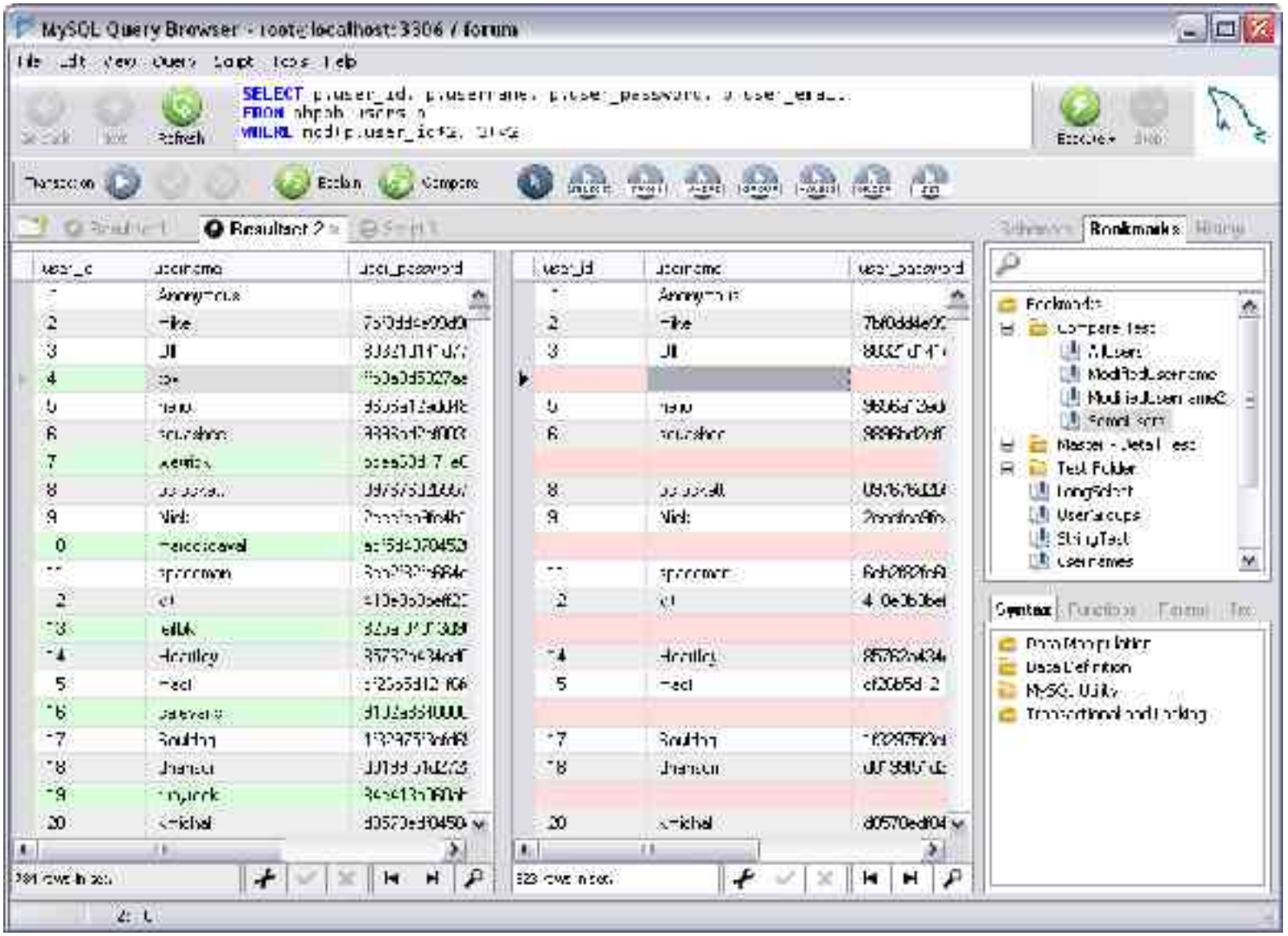
MySQL Administrator



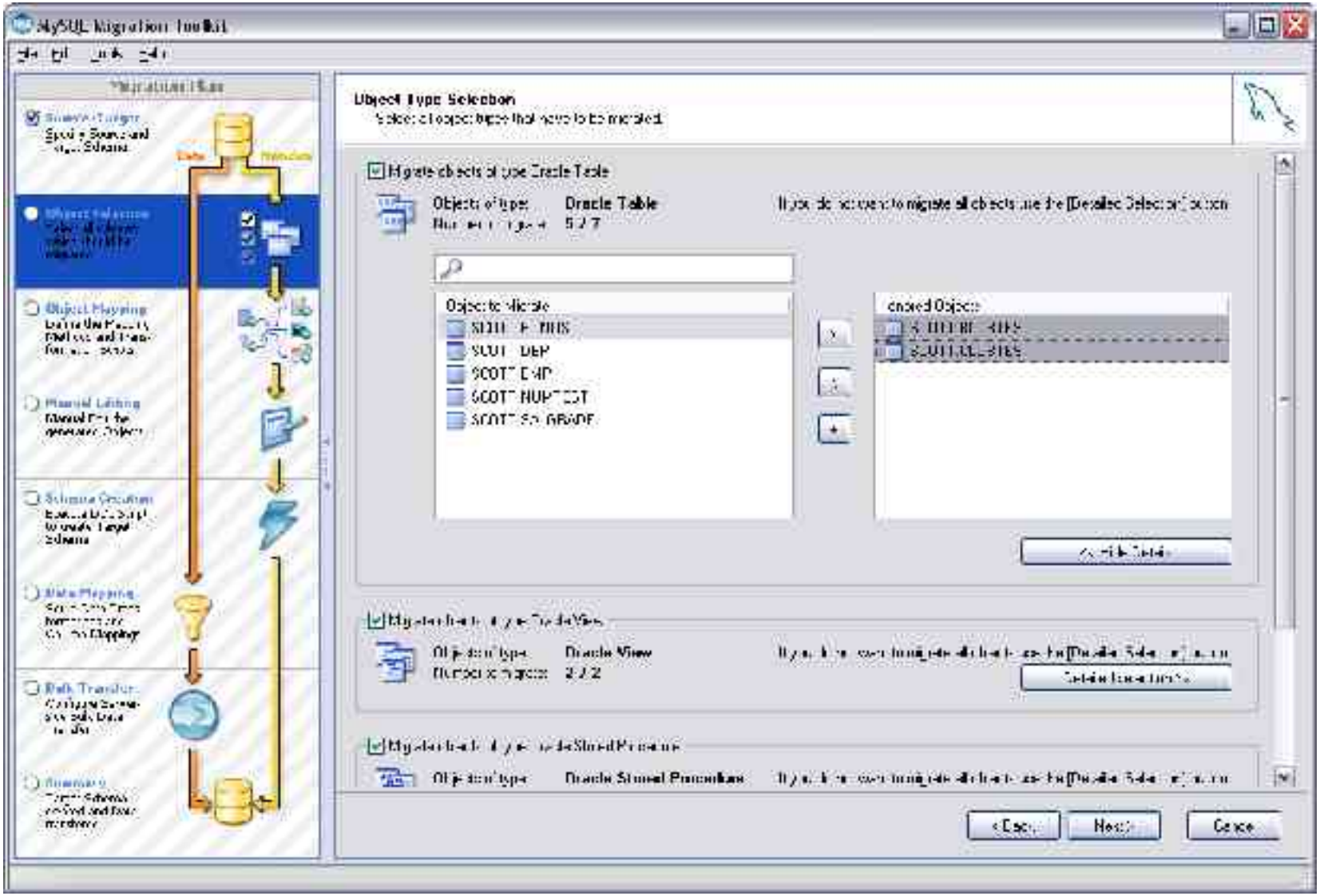
MySQL Query Browser



MySQL Query Browser



MySQL Migration Suite



The future

- We aim toward full compliance with ANSI/ISO SQL. There are no features we plan not to implement.

- Note: If you are an enterprise-level user with an urgent need for a particular feature, please contact <sales@mysql.com> to discuss sponsoring options. Targeted financing by sponsor companies allows us to allocate additional resources for specific purposes. One example of a feature sponsored in the past is replication.

Contact

Stewart Smith

Software Engineer, MySQL Cluster

MySQL AB

stewart@mysql.com

http://www.mysql.com/