Advanced MySQL topics

Presented by:

John A Mahady

AndrewInfoServices.com
**Topics**

- Introduction
- Stored Procedures
- Views
- Triggers
- Cursors
- ODBC & OO Base
Press Release - April 15, 2009

"Next Week's MySQL Conference & Expo Spotlights Obama Campaign's Web Team; Open Source's Contribution to Presidential Campaign Explored"

http://press.oreilly.com/pub/pr/2271
Move processing and logic to the DBMS
LAMP Installation in Ubuntu
Linux, Apache, MySQL and PHP

1. Optionally, install SSH Client and Server (for remote access to this server)
   `sudo apt-get install ssh`

2. Install Database Server
   `sudo apt-get install mysql-server`

3. Install Apache2 web server
   `sudo apt-get install apache2`

4. Install PHP5
   `sudo apt-get install php5 libapache2-mod-php5`

5. Install PHP5-MySQL support
   `sudo apt-get install php5-mysql`

6. Restart Apache
   `sudo /etc/init.d/apache2 restart`

7. Optionally, install phpMyAdmin
   `sudo apt-get install phpmyadmin`
Where do you put your web pages?

In your browser type

'LOCALHOST' or 127.0.0.1

In Ubuntu/Linux this is equal to directory

/var/www
Download sample database 'world'

This database is used in MySQL certifications and training!

Download and unzip from [http://dev.mysql.com/doc/](http://dev.mysql.com/doc/)

```
shell> mysql -u root -p
mysql> CREATE DATABASE world;
mysql> USE world;
mysql> SOURCE world.sql;

mysql> SHOW TABLES;
mysql> DESCRIBE Country;
mysql> DESCRIBE City;
mysql> DESCRIBE CountryLanguage;
```
Topics

- Introduction
- Stored Procedures
- Views
- Triggers
- Cursors
- ODBC & OO Base
SELECT * FROM INFORMATION_SCHEMA.ROUTINES;

The INFORMATION_SCHEMA database is available to all MySQL users so they can know the environment and objects. The tables in this database are read-only views based on the tables in the mysql database which is only accessible to those with the privileges since the tables are modifiable.

SELECT * FROM mysql.proc;

This is equivalent to the above statement but this is the live table the view is selecting data from.
Information on Stored Procedures

SHOW PROCEDURE STATUS;

database, name, type, creator, creation and modification dates, and character set information.

SHOW CREATE PROCEDURE procname\G

returns the exact string that can be used to re-create the named stored procedure.

SELECT * FROM INFORMATION_SCHEMA.ROUTINES\G
Passing back values

Stored procedures pass values back to the calling program by:

- OUT or INOUT parameters
- Global session variables that persistent after the procedure ends
- The result set from one or more SELECT statements
- Stored Functions use the RETURNS statement.
Stored Procedures - variables

- Data types **exception**: can use all the scalar types a.k.a. single values but NO arrays, records, or structures.

- Scope of **User variables** is global to the session. But its recommended you reduce scope through:
  - DECLARE private variables inside the Procedure or
  - pass session/user variables as parameters with IN, OUT, INOUT settings so you can obtain its value when the procedure returns.
Stored Proc – passing a literal value

Example:

- CREATE PROCEDURE sp_mealtip( IN tip float )
  - SELECT .15 * tip;

- mysql> CALL `demo`.`sp_mealtip`(300);
- Query OK, 0 rows affected (0.00 sec)

+---------+
| .15 * tip |
+---------+
| 45      |
+---------+
- 1 row in set (0.00 sec)
Example:

- `CREATE PROCEDURE sp_mealtip(INOUT tip float)`
  
  `SET tip = .15 * tip;`

- `mysql> set @tip = 200;`

- `mysql> call demo.sp_mealtip(@tip);`

- `mysql> select @tip;`

```
+--------+
|    @tip |
+--------+
|      30 |
+--------+
1 row in set (0.00 sec)
```
Backuping up databases

`mysqldump` will backup by default all the triggers but NOT the stored procedures/functions. There are 2 `mysqldump` parameters that control this behavior:

```
--routines - FALSE by default
--triggers - TRUE by default
```

This means that if you want to include triggers and stored procedures in an existing backup script you only need to add the `routines` command line parameter:

```
mysqldump -pdemodba1 -udemodba srcdb > demo2.sql

mysqldump --routines -p -udemodba srcdb > demo2.sql
```
Stored Proc Security Feature

Use MySQL Administrator or commands

```
REVOKE ALL PRIVILEGES .....;
```

```
SHOW GRANTS FOR CURRENT_USER;
```

Restrict user to EXECUTE privilege.

Login into MySQL Query Browser with user Spuser

Only objects available are stored procs.
Brief example of a function

CREATE FUNCTION `f_mealtip`(in_meal decimal)
    RETURNS decimal(10,0) DETERMINISTIC
RETURN in_meal * .15;

SELECT *, f_mealtip(cost) AS tip FROM demo.meals;
Examples in MySQL Query Browser

CALL world.1_sp_sel_na_ppsf()

CALL 2_sp_in_sel( IN v_continent varchar(20) )

CALL 3_sp_out_continentavglifeexpectancy(
  IN v_continent varchar(20),
  OUT avglife float)

CALL world.spRtn2selects( IN cntrycod CHAR(3) )

CALL demo.sp_tfer_funds(
  from_account int,
  to_account int,
  tfer_amount numeric(10,2),
  OUT status int, OUT message VARCHAR(30))

CALL john1.sp_get_topics;
Topics

- Introduction
- Stored Procedures
- Views – two issues
- Triggers
- Cursors
- ODBC & OO Base
Views – significant issue

- Two ways a View is processed: **MERGE vs TEMPTABLE**
- MySQL tries to use MERGE algorithm first.
- Temp table is used if sql command includes GROUP BY, DISTINCT, aggregate functions, UNION, or other inputs that break the one-to-one relationship of view to base table.
- TEMPTABLE views are not updatable because of the above point.
- Temporary tables have no indexes so table scan run slower.
- Views can be updatable even if they have a JOIN but updates must be in one table not both.
- Views can be used instead of column privileges which impact performance and prevent usage of the query cache.
- A View can not have a trigger associated with it.
Views – example

- **Original View**
  CREATE VIEW vw_oceania AS
  SELECT * FROM country WHERE Continent = 'Oceania'
  WITH CHECK OPTION;

- **Command that uses view**
  SELECT Code, Name FROM vw_oceania WHERE Name='Australia';

- **If the processor chooses TEMPTABLE solution (not updatable):**
  >CREATE TEMPORARY TABLE tmp_oceania AS
  >SELECT * FROM country WHERE Continent = 'Oceania';
  >SELECT Code, Name FROM tmp_oceania WHERE Name='Australia';

- **If the processor chooses MERGE table solution:**
  >SELECT Code, Name FROM country
  WHERE Continent = 'Oceania' AND Name='Australia';
Views – diagnostic code

- Example in Query Browser:
  Run SELECT in world.sp_mergeview then menu 'Explain Query' = SIMPLE
  Run SELECT in world.sp_temptbl then menu 'Explain Query' = DERIVED

- See result in mysql program within Terminal.
  Mysql> See how MySQL rewrites the query with EXPLAIN EXTENDED followed by SHOW WARNINGS.
  Mysql> EXPLAIN SELECT * FROM <view_name>
  If response says DERIVED = temp table.
Views – frozen schema

The view definition is “frozen” at creation time, so changes to the underlying tables afterward do not affect the view definition.

CREATE VIEW v_test AS SELECT * FROM table;

Becomes

CREATE VIEW v_test AS SELECT fld1,fld2,fld3 FROM table;
CREATE

ALGORITHM = MERGE

VIEW `john1`.`v_merge` AS

SELECT * FROM ex_tbl_trgr;

CREATE

ALGORITHM = TEMPTABLE

VIEW `john1`.`v_temptbl` AS

SELECT * FROM ex_tbl_trgr;

INSERT INTO v_merge VALUES(null,'lois','lane');

INSERT INTO v_temptbl VALUES(null,'clark','kent');  ERROR!
Topics

- Introduction
- Stored Procedures
- Views
- Triggers
- Cursors
- ODBC & OO Base
Triggers -1

- Can be helpful for automatically updating denormalized and summary tables.
- Also can be used to enforce constraints or business logic.
- One trigger per table per each event.
- MySQL supports only row-level triggers -FOR EACH ROW. No triggers on datasets currently.
- Data is consistent at all times versus periodic bulk update routines.
- Server as foreign key functionality in non-transactional tables, e.g. MYISAM.
Triggers -2

- In transactional tables (Innodb) triggers will be atomic with the statement that fired them. ROLLBACK and COMMIT.

- They can obscure what the server is doing.

- Can be hard to debug.

- Triggers can cause nonobvious deadlocks and lock waits.

- MySQL trigger implementation is not mature yet.

- Triggers for a table are currently stored in .TRG files.

- MySQL triggers are activated by SQL statements only.
**GOAL:** From within the Query Browser tool use a Before Delete trigger to copy a record to a backup table when it is deleted.

```
CREATE TRIGGER trg_del BEFORE DELETE ON ex_tbl_trgrr FOR EACH ROW
  INSERT INTO ex_tbl_trgr_bkp (id,name,company,whenadded,action)
  VALUES(OLD.id,OLD.name,OLD.company,null,'delete');
```
Triggers – ODBC delete example

- **GOAL:** From within OpenOffice Base (similar to MS Access) connect to the MySQL world database using ODBC and delete a record in the Country table and see if the "backup" trigger works.

- CREATE TRIGGER world.trg_country_bkp
  BEFORE DELETE ON world.Country
  FOR EACH ROW
  INSERT INTO Country_bkp (Code, Name, Continent, Region)
  VALUES(OLD.Code, OLD.Name, OLD.Continent, OLD.Region)
Triggers – OO Base insert using a View example

This example will insert a new record into a MySQL View from within OO Base. Will the trigger fire?

CREATE TRIGGER trg_ins BEFORE INSERT ON ex_tbl_trgr FOR EACH ROW
INSERT INTO ex_tbl_trgr_bkp (id,name,company,whenadded) VALUES(NEW.id,NEW.name,NEW.company,null);

INSERT INTO v_merge VALUES(null,'bozo','the clown');

SELECT * FROM ex_tbl_trgr;
SELECT * FROM ex_tbl_trgr_bkp
Topics

- Introduction
- Stored Procedures
- Triggers
- Views
- Cursors
- ODBC & OO Base
Cursors

- MySQL provides read-only, forward-only, server-side cursors.

- Currently, cursors are not updatable because they create and iterate through a temporary table not the base table.

- Can only be used within a stored procedure. A stored proc can have more than one open at once.
A cursor executes the entire query when it is opened. So closing a cursor early doesn't save your server any effort.

So even if you just need to touch the first 100 records on a million record table a query of all the records is done first with the cursor OPEN command.

If a cursor involves a table with BLOB or TEXT field an on-disk temporary table will be created since in-memory temporary tables do not support these field types.
Topics

- Introduction
- Stored Procedures
- Views
- Triggers
- Cursors
- ODBC & OO Base
Connect OpenOffice Base to MySQL

- Base database engine internally is HSQLDB (OSS)
- ODBC can be loaded on a Linux as well as Windows platform
- Base can select, update, insert, or delete MySQL tables
- Local queries and reports can access MySQL data
- Base can see tables and views but not stored procedures
- Triggers will still fire from OO Base
- Demo example
Install ODBC for connections – pg 1

- Install **iodbc** - GTK+ config front end for the iODBC Driver Manager
  - to install: `sudo apt-get install iodbc`

- Install **mysql connector for odbc**
  - `sudo apt-get install libmyodbc`

- Start **iodbc**
  - `/usr/bin/Iodbcadm-gtk`

- Add the **mysql driver to the list of ODBC drivers**
  - Switch to the "ODBC Drivers" tab, click "Add a driver". Type in a description of the driver (i.e., "MySQL").
  - For "Driver file name" choose `/usr/lib/odbc/libmyodbc.so`.
  - For "Setup file name" choose `/usr/lib/odbc/libodbcmyS.so`. 
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only available to you, and can only be used on the current machine.
Install ODBC for connections - pg 2

>Create a new User DSN (the procedure is the same for System DSN)

- Click "Add", select the already created MySQL ODBC driver.
- Give a name to your datasource (i.e., "bobs_mysqldb").
- Add keywords based on your server configuration. Minimally have:
  - server
  - database
  - user
  - password

FAQ: That's it.

John A Mahady

john@AndrewInfoServices.com

312.560.8072