Database Schema Versioning and Management Tools

Michael Coburn, Percona
Percona Live 2016
About Me

- 4 years (yesterday!) with Percona
- First as a Consultant, then Manager of Consulting, now...
- Principal Technical Account Manager at Percona
- Tasked with building & maintaining long-term relationships with Percona's highest revenue customers
Why this talk?

- Developers can sometimes "forget" to migrate changes past Dev or QA database
- Or changes can be applied haphazard by non-DBAs
- You're left comparing schemas across environments
- Generally there is little process around schema changes
- Keeping It Simple is generally better
Agenda

- ORMs and Everything Else
- Defining Schema Changes
- Persisting your Schema beyond the DB
- Managing Schemas across Environments
- Applying Schema changes
- The Team Approach
The Process of Schema Changes

1. Developer and DBA decide on change
   ○ new table/column/index
   ○ ALTER TABLE syntax is output
2. Change applied to Development, then QA environment
   ○ This is where you, the DBA, will take the lead
3. New schema is dumped and committed to repo
4. Change applied to Stage
5. Change applied to Production
What are the Components of a Database Schema?

1. Schema - container to hold your Tables
2. Table definitions - columns, data types, indexes
   - mysqldump --no-data --skip-dump-date
3. User permissions
   - pt-show-grants --no-header --no-timestamp
4. Views, triggers, stored procedures
   - Not included by default in mysqldump
   ● It should NOT be the environment's data!
Where Does Your Schema Exist?

1. In your database
2. In Application logic
3. In backups
4. In a code repository (git, svn)
The Goals of Managing Schemas

• To accurately track and deploy schema changes in sync with the Application

• Generally split into 2 Camps:
  a. Those that use an ORM Framework
  b. Those that don't

• We'll be focusing mostly on the second, but just quickly...
Why Object Relational Mapping?

• Developers work with Objects
• ORMs connects these Objects to tables in MySQL
• Popular examples:
  • Ruby on Rails + ActiveRecord
  • Yii PHP Framework
Why Object Relational Mapping?

• ORMs already include systems to keep the versions of the schema in sync with the Application release
  • ActiveRecord has the schema_migrations table
    ▪ Migrations are expressed in plain English (well, Ruby code)
• Those deserve their own talk, not this one :)
So you don't have an ORM...

- You still need to track schema changes
  - PUT IT IN A REPO!
- Commercial
  - Redgate's MySQL Compare
- Open Source
  - Liquidbase
    - Probably the best option...
  - mysqldump --no-data --skip-dump-date
So what do you do with your Schema?

1. Store the schema alongside the code
2. Split the schema apart from the data
3. Timestamp your sql dump files
   - `mysqldump mySchema > mySchema_`date +%F`.sql`
Managing Multiple Environments

• Your Application should run in more than a single environment called Production
  • You do this already, right?

• Consider:
  • Development (generally Dev's local machine)
  • QA - often Developer pushes to QA themselves
  • Stage - deploys managed by someone other than the Developer
  • Production - where the magic happens
Managing Multiple Environments (2)

• Don't lose sight of where your schema changes are!
• It is common to see a QA and Stage environment be refreshed with Production data on a regular basis
• Do you need to compare versions?
  • Again, mysqldump!
  • diff schema1.sql schema2.sql
Deploying Schema Changes

1. ALTER TABLE is the default action
   - Useful on tables < 10k rows

2. pt-online-schema-change
   - --chunk-time=0.5
   - --chunk-size=1000
Team Approach

1. Building consensus with your Development team
   ○ You are all in it together
   ○ Don't just throw it over the wall

2. Spend the time to think through the process
   ○ It's ok to come up with the method yourself
   ○ But involve others too!

3. Make using the system a priority
   ○ Don't work around it, fix it!
Thank You! Questions?

• Michael Coburn michael.coburn@percona.com
  • Principal Technical Account Manager, USA