Archiving Audit Data

Last Revision Date: 3/12/14

Your LAVA app performs auditing of all user events and data modifications and this audit data is stored in the audit_*_work tables. Over time these tables will grow quite large and should be archived. We recommend archiving this data to an audit archive database that is separate from your production database. We have a script to create the audit history tables in an audit archive database, as well as a script which moves the data over from the production database into an audit archive database. Below I have included the instructions to archive your audit data and these instructions reference these two scripts.

How often you should archive the audit data varies widely depending upon how much data is entered into the database over time. So there is no specific recommendation about how often you should do this. You will have to get a sense for this yourself over time. Eventually, if your audit tables got so large you would see some performance issues and database errors.

Note: the audit_*_history tables in your database are not used in production for auditing. These would be used if you were archiving your audit_*_work tables to these audit_*_history tables in the same database. Rather than doing this we recommend a separate database, and the instructions below cover archiving to a separate database.

1) Create an audit archive database, e.g.
Assuming production database is lava_APP
create database lava_APP_audit_history

2) Grant all privileges on lava_APP_audit_history to the MySQL user for corresponding production database, lava_APP

3) Create the audit history tables in the audit archive database using this script:
lava-core/database/core/development/model/audit_archive_tables.sql

4) Copy the following move_audit_logs.sql script to your app project and replace _AUDIT_HISTORY_SCHEMA_ with your audit archive database, e.g. lava_APP_audit_history
/lava-core/database/core/development/util/move_audit_logs.sql

5) Run the modified move_audit_logs.sql script on your production database (not your audit archive database). Run the script as the database user granted privileges in Step 2) such that the user has the necessary privileges in both the production database and the audit archive database. All of the audit data will be moved from your audit_*_work tables in the production database to the audit_*_history tables in the audit archive database.