Synthetic Data

LSST Database

No need to support any of this in DC3b

Synthetic data = inject at pixel level some fictitious "sources" in order to verify how well pipelines are able detect them.

When synthetic sources are injected, information about them will be recorded in a place inaccessible to users and pipeline developers. Only these with appropriate privileges will be able to access this information. This information will likely be stored in a special Source table, and the schema will likely be a superset of LSST Source table schema + imageSim schema.

We need to use association pipeline to to match information about the synthetic sources with the Source catalog produced by LSST pipelines.

Important issues:

- we need to quarantine synthetic sources from the lsst science db, can't leak them there. Current thinking: any image that is touched by synthetic sources will have a non-polluted copy. We will do separate runs to process images containing synthetic sources
- we need to hide info about synthetic sources from users.

Hiding information in database

We are currently granting "SELECT on *.*" to all users. Given we are not using any common prefix for naming databases, it is not immediately obvious how to narrow down this restriction.

One solution would be to keep it on a different database server, which makes it hard to correlate synthetic sources with the catalogs produced by dc3b pipelines.

It might be a good idea to introduce a common prefix, we will need it for later, eg for restricting access to level 3 and level 4 data.

Potentially useful (from KT):

```
GRANT USAGE ON *.* TO 'user'@'%'
GRANT SELECT ON `%`.* TO 'user'@'%'
GRANT SHOW VIEW ON `test`.* TO 'user'@'%'
```

Note the % instead of *. This puts this grant in the mysql.db table instead of the mysql.user table.

This actually removes all privileges except SHOW VIEW from the user for the test database, since this grant is more specific than the previous one. In particular, it removes the SELECT privilege. Unfortunately, it doesn't look like it's possible to turn off *all* privileges in this manner without direct updating of the mysql.db table. SHOW VIEW seems to be the least intrusive one. Still more unfortunately, as long as the user has *any* privilege for a database, it appears that the user can see that database in SHOW DATABASES, can USE that database, and can even SHOW TABLES inside it.