

# Database Ingest in DC3b

## LSST Database

Qserv software will *not* be used for data ingest, it will only be used for data analysis on read-only data.

Data ingest in DC3b-PT1 will be the same as in DC3a: no partitioning.

Data ingest in DC3b PT2 and PT3: all partitionable tables will be statically pre-partitioned, and data will be ingested directly to appropriate partitions. Partitions could be distributed across multiple nodes, but given we will only have one server (lsst10), in practice all partitions will reside on a single server.

If we determine it is necessary, in DC4 we will introduce post-processing, e.g., to fine-tune partitioning or remap objectIds or sourceIds.

In some cases pipelines expect to (a) read data from database, and (b) update some tables. For these reasons, there will be a temporary *scratch space* in between pipelines and the final database. This scratch space will consist of 2 parts:

1. a database, likely centrally located but potentially per-node, which will contain:
  - ◆ all data that data release pipelines (eg astro cal) expect to read from database
  - ◆ all data that needs to be updated from data release pipelines (such as exposures)
2. TSV files, which will contain all non-updatable data. These TSV files will be processed by the DbIngest pipeline, which will use the partitioner to re-partition the TSV files and the loader to load them into appropriate tables. This pipeline may run at the same time as other pipelines in the Data Release Production or after the others. The partitioner and loader will be configurable by their respective stage policies.

We expect the scratch database to be relatively small, e.g., we expect we won't need to keep the entire Object or Source or ForcedSource catalog - we assume pipelines will work with individual tiles, and keeping data for "current" tile or a small set of "current" tiles will be sufficient.

Note that we do not plan to partition moving objects in DC3b. This is mostly because of small size (6 million rows expected in production). We might need to partition ForcedSourceForMovingObject (which will trigger partitioning for the MovingObject table) after DC3b due to its potentially large size.