

# How to convert LDP queries to Metadb format

1. Familiarize yourself with the Metadb documentation
  - a. The Metadb documentation from Index Data is revised continuously with the latest information on using Metadb
  - b. See <https://metadb.dev/doc/>
  - c. For the conversion process, the important section of the documentation to focus on is [4.1.4 Migration from LDP](#)
2. Convert LDP table references to their Metadb equivalents
  - a. use the Migration from LDP table conversion chart supplied in the Metadb documentation
  - b. [https://metadb.dev/doc/#\\_migrating\\_from\\_ldp](https://metadb.dev/doc/#_migrating_from_ldp)
  - c. table names in metadb are case sensitive
  - d. many tables names that use singular instead of plural form now
  - e. for dashboards, if table names have changed it doesn't make a difference, but field names do need to be changed back to LDP names
3. Update Derived Table references
  - a. update your derived table references from LDP's folio\_reporting schema to Metadb's folio\_derived schema
4. Update JSON to JSONB
  - a. for example change `json_extract_path_text(pol.data, 'description')` AS `pol_description` TO `jsonb_extract_path_text(pol.jsonb, 'description')` AS `pol_description`,
5. Update Data Fields that have Changed in Metadb
  - a. See [LDP to Metadb field name changes](#)
  - b. Please add notes to this page if you find additional field name changes we should know about - thanks!
6. Verify that the data type is set to text or VARCHAR for any UUID fields used in JOINS. In LDP, the UUIDs were mostly stored as strings. In Metadb, all UUIDs are of UUID datatype, so if you are joining on a UUID you need to cast it as text or varchar, e.g., `uuid::VARCHAR = uuid.VARCHAR`.
7. Test Your Query
  - a. Once you appear to have a working Metadb version of your query, it is important to test it
  - b. Do the LDP and Metadb versions of your query produce the same results? (assuming you have production data going to both LDP and Metadb)
  - c. Ask someone else on the reporting team to test your query
8. Document and Publish Your Query
  - a. With the move to Metadb, we are taking the opportunity to make our naming conventions and query documentation better and more consistent
  - b. LDP to Metadb Query Naming Convention
    - i. Leave CR numbers in place, even if query is archived, to avoid confusion
    - ii. To revise a CR or AHR query for LDP to Metadb format, use same CR or AHR number, but precede with M to maintain a connection between the 2 queries for testing purposes. For example, CR100 becomes MCR100, AHR120 becomes MAHR120
    - iii. For new Metadb queries, we start in the 400 range, so MCR400 is the first metadb-only query
9. Post the Metadb version of your LDP Query on the cul-folio-analytics GitHub repository
  - a. log into GitHub and navigate to new metadb section at <https://github.com/cul-it/cul-folio-analytics/tree/main/metadb>
  - b. make sure you know the name of this query
  - c. create a new directory for the query and publish it at the same time as you create the new file in the metadb section of the repository, e.g.,
  - d. click to create a new file and type in "MCR120/daily\_appr\_inv\_control.sql" to create the directory and name the file in one step
  - e. paste the text for your revised query into the new file and commit the file to publish it