

# LDP and Metadb Known Issues

Issue	Reported Date	Description and Temporary/Workaround Solution	Reported by	Solution Status and Date
users_groups	5/13/24	LDP: the first subquery (creating the user_departments table) uses the users_departments_unpacked derived table, but doesn't specify "folio_reporting" as the schema. The department information from the custom fields (of the user_users table) is not properly extracted.  MetaDB: the code does not specify the folio_derived schema for the user_departments subquery. Also, the MetaDB version is lacking the "user_tags" and "user_custom_fields" fields that are in the LDP version.	Joanne	
po_instance table generating duplicates	4/30/24	MetaDB: po_instance derived table has duplicate rows and null values for location fields	Joanne	Unknown
finance_transaction_purchase_order table for Metadb has two errors	4/30/24	The MetaDB derived table finance_transaction_purchase_order table has two errors in it:  <pre> jsonb_extract_path_text(pol.jsonb, 'poLineNumber') AS pol_number, jsonb_extract_path_text(pol.jsonb, 'description') AS pol_description, jsonb_extract_path_text(pol.jsonb, 'acquisition_method') AS pol_acquisition_method, jsonb_extract_path_text(po.jsonb, 'order_type') AS po_order_type, jsonb_extract_path_text(po.jsonb, 'vendor')::uuid AS po_vendor_id, jsonb_extract_path_text(oo.jsonb, 'name') AS po_vendor_name </pre> The extracted elements should be "acquisitionMethod" and "orderType". Also – to get the acquisition method name, instead of just getting the id, they could add join to the "folio_orders.acquisition_method__t" table and get the "value" entry. This would be a great addition to the derived table.	Joanne	Unknown
LDP not showing invoice charge	4/18/24	When hosting fixes an invoice, some associated charges may not appear in LDP tables. For example, invoice 327443 has a shipping charge for 29.48 that is not appearing on the fund transactions table in the LDP. Therefore, our reports may show a discrepancy in invoice charges.		Poppy (one problem) and Quesnalia (another problem) may fix invoice input issues so there will be better data quality going forward
po_organization derived table	4/15/24	LDP and MetaDB - The table lacks contact names. The code for the derived table is not correct, because it matches the organization_organizations.id to the organization_contacts.id, which are not the same thing. The organization_contacts id has to be gotten out of the data array in the organization_organizations table ("contacts").	Joanne	
Fiscal Year field on Invoice	1/31/24	<ul style="list-style-type: none"> <li>New feature allows approval of invoices without fiscal year associated with it</li> <li>CUL will not use the feature</li> <li>Does not default to the current fiscal year - now it does</li> <li>not mandatory field -</li> <li>only appears if permission is assigned, if you don't have the permission, the current fiscal year is assigned by default</li> </ul>		
Problems with patron data	12/20/23	Graduate students have been reverted to Undergraduate in the Patron Groups, except for about 300 records; this may impact all queries and dashboards using patron data		Unresolved
Fund 9	12/13/23	An invoice was inadvertently deleted out of the Fund screen in FOLIO, which means balances for fund 9 will be off by \$18,000 until this is fixed.  Fixed as of 1/31/24		Resolved
Publisher	12/13/23	There have been some inconsistencies in the availability of the publication date and publisher on tables with this data. Advised to get this data from the instance, not the PO for now		Unresolved
Fully paid order still shows encumbrance	6/1/23	In orders reports, even though payment status is showing the "Fully Paid" value, there are still encumbrances showing		Unresolved
FY23-FY24 Fiscal Year Rollover Changes to Fiscal Year Dates	7/15/23	FY23-FY24 Fiscal Year Rollover Changes <ul style="list-style-type: none"> <li>End date of FY2023 is <b>7/3/2023</b> and start date for FY2024 is <b>7/4/2023</b></li> <li>Cannot use 7/1/23 or 7/2/23 as FY24 start date</li> <li>Change needed to accommodate a lost book cost invoice under Fund 310 that was not attached to a PO</li> <li>Impacts on queries – will impact everything with a transaction</li> <li>Impacts on dashboards</li> <li>Date range and Fiscal Year may not match</li> <li>Add note to show changed date range</li> <li>No impacts to collection counts</li> <li>Notes on FY2024 change have been added to finance dashboards</li> </ul>		None

Date/Time in LDP not matching Folio	5/17/23	<p>This issue has been found when using the folio_reporting.feesfines_account_actions. It may very well affect many other tables. The transaction date result from an SQL query using LDP is showing a timestamp difference of 4 hours.</p> <p>If the query is referring to an original table where the time was recorded as Timestamptz, then the date in the derived table needs to be casted as Timestamptz, otherwise the system will add 4 hours automatically.</p> <p>It will affect the query results when the date/time was recorded after 8PM UTC. Examples are Fees, Fines, loans etc. A review of all derived tables using dates should be done.</p> <p><u>Example:</u></p> <p>Existing code: <code>json_extract_path_text(ff.data, 'dateAction') AS transaction_date,</code></p> <p>Replace by: <code>json_extract_path_text(fa.data, 'dateCreated') ::timestamptz AS transaction_date</code></p> <p>As a temporary fix, if your query is affected by a date not casted as Timestamptz, update your query by changing the data type to Timestamptz. The results should show the correct date/time. Derived tables will need to be updated by the Folio Community.</p>		<p>Unresolved</p> <p>This issue (along with several others) is scheduled to be fixed in <a href="https://github.com/folio-org/folio-analytics/issues/501">https://github.com/folio-org/folio-analytics/issues/501</a></p>
feesfines_comments table	4/26/23	<p>feesfines-comments table not populating with comments</p> <p>-not requested in reports yet</p> <p>-not high priority, but good to know</p>		Unresolved
The derived table "feesfines_accounts_actions" is no longer correct because the source table (public.feesfines_accounts) has changed. Several fields in feesfines_accounts_actions are now blank.	4/19/23	<p>The derived table "feesfines_accounts_actions" is no longer correct because the source table (public.feesfines_accounts) has changed. Several fields in feesfines_accounts_actions are now blank.</p> <p>We discussed this issue in the reporting development meeting. Angela will update the feesfines_accounts_actions derived table so that these fields populate from the public.feesfines_accounts table. She will also bring up the problem of not finding out about field changes to Product Council on 4/20/23.</p> <ul style="list-style-type: none"> <li>• payment_status (object)</li> <li>• payment_method</li> <li>• fine_updated_date</li> <li>• fine_date</li> </ul>		Unresolved, but plan in place
srs_marctab instance count is low in LDP Production	3/14/23	<p>reported to EBSCO</p> <p>-The number of srs_marctab instances in the LDP Production instance is down by a factor of 10 compared to LDP Test (600 million down to 60 million)</p> <p>-The number of srs_marctab instances in LDP Test is now down by 318624 records compared to 3/14/23</p> <p>Resolution: Setting statement_timeout = 7200000 parameter for postgres addressed issue with srs_marctab table not getting updated fully in LDP Test and LDP Prod</p>		<p>Unresolved</p> <p>-still seeing intermittent problems</p> <p>LDP system configuration setting addressed issue</p>
Preserving Circulation Demographic data	1/17/23	<p>Preserving Circulation Demographic data</p> <ul style="list-style-type: none"> <li>• patron id is wiped from circulation tables when items are checked in, which results in losing connection to demographic data in custom fields for reporting</li> <li>• cul derived local table to grab and save demographic data daily; does not get items checked out and returned in the same day</li> <li>• need something like the circulation_transaction_stats table we had in Voyager</li> <li>• take snapshot of loans_items, then save demographic (custom fields) data before the patron id is wiped</li> <li>• may need to ask for a feature working with RA SIG</li> <li>• needs to be discussed at FOLIO Governance</li> <li>• Joanne has submitted a FOLIO Application enhancement request to create a way to retain patron demographics (department, college) in circulation loan history</li> <li>• Tom Trutt submitting an enhancement request for this issue</li> </ul> <p>Note: Metadb will include history for tables, so patron demographics could be gathered this way</p>		<p>Workaround in place</p> <p>local_core.circ_snapshot 3 updated automatically daily at 7am</p>
Call Number Order on Holdings-Level Call Numbers		<p>-see <a href="https://issues.folio.org/browse/UXPROD-3496">https://issues.folio.org/browse/UXPROD-3496</a></p> <p>-this is a feature that allows you to order your lists of items in correct call number order with the holdings level call number</p> <p>-Currently, we are unable to sort holdings record results by correct LC call number order. For item records, we can do this through the "effective_shelving_order" field in the inventory_items table (and with applying the "collate "C" function in the ORDER BY part of the query). We can't do this with holdings records. It never comes out right. We had a JIRA ticket for this work, but we can't find it.</p>		Unresolved
Item Level Call Number no longer on some derived tables		<p>-derived tables <b>loans_items</b> and <b>items_holdings_instances</b> no longer capture the item level call number, because the inventory_items table (which they reference) has changed</p> <p>-The derived tables "items_holdings_instances" and "loans_items" no longer contain the item call number, because the source table (inventory_items) no longer has that field. Instead, inventory_items has the call number components parsed out of the data blob (prefix, suffix and call number core). These components need to be included in a revision of the scripts for the two derived tables. Also, the item_ext derived table has been updated to include the parsed-out call number components, but still retains the non-functional item_level_call_number field.</p>		<p>Resolved</p> <p>-Natalya has updated items_holdings_instances for Orchid 1.6</p> <p>-Natalya has updated loans_items for Orchid 1.6</p>

Orchid Breaking Changes for Subjects and Series	1/3/23	<p>Breaking changes for Orchid release: updates to instance records (subjects and series) to capture changes associated with authority control</p> <p>Cornell goes to Nolana - end of Feb, beginning of March</p> <p>Cornell goes to Orchid - hopefully before Fiscal Year Rollover</p> <ul style="list-style-type: none"> <li>◦ <a href="https://folio-project.slack.com/archives/C210RP0T1/p1672757167354339">https://folio-project.slack.com/archives/C210RP0T1/p1672757167354339</a></li> <li>◦ <a href="https://issues.folio.org/browse/MODQM-276">https://issues.folio.org/browse/MODQM-276</a></li> </ul> <ul style="list-style-type: none"> <li>• subjects field will be converted from an array of strings to an array of objects.</li> <li>• series field will be converted from an array of strings to an array of objects.</li> <li>• alternativeTitles field object will be extended with authorityId field</li> </ul> <p>The changes are mainly related to two fields: subjects field is now an array of objects:</p> <pre> "properties": {   "value": {     "type": "string",     "description": "Subject heading value"   },   "authorityId": {     "type": "string",     "description": "UUID of authority record that controls a subject heading",     "\$ref": "uuid.json"   } } </pre> <p>series field is now an array of objects:</p> <pre> "properties": {   "value": {     "type": "string",     "description": "Series title value"   },   "authorityId": {     "type": "string",     "description": "UUID of authority record that controls an series title",     "\$ref": "uuid.json"   } } </pre>	Resolved	<p>-Natalya has updated these derived tables for Orchid 1.6 Folio Analytics release</p> <p>-see more information in <a href="#">Subjects and Series documentation</a></p>
<p>Finance_Budgets table issue</p> <ul style="list-style-type: none"> <li>• on the public finance_budgets table for LDP1, several fields are not getting calculated properly due to the way the calculations are done in the FOLIO application fields</li> </ul>	1/17/23	<p>Finance_Budgets table issue</p> <ul style="list-style-type: none"> <li>• <a href="#">UXPROD-4368</a> may help address this issue, and is assigned to the Ramsons release <ul style="list-style-type: none"> <li>◦ Add interface to allow users to fix specific rollover errors</li> </ul> </li> <li>• on the public.finance_budgets table for LDP1, the following fields are not getting calculated properly due to the way the calculations are done in the FOLIO application fields: <ul style="list-style-type: none"> <li>◦ allocated</li> <li>◦ cash_balance</li> <li>◦ total_funding</li> <li>◦ available</li> <li>◦ unavailable</li> <li>◦ over_expended</li> <li>◦ over_encumbrance</li> </ul> </li> <li>• see submitted <a href="#">LDP issue 226</a> on this for more details</li> <li>• see details from FOLIO developer Serhii Nosko <ul style="list-style-type: none"> <li>◦ <a href="https://github.com/folio-org/mod-finance-storage/blob/master/src/main/java/org/folio/utils/CalculationUtils.java#L61">https://github.com/folio-org/mod-finance-storage/blob/master/src/main/java/org/folio/utils/CalculationUtils.java#L61</a>.</li> <li>◦ They are calculating each time a budget is retrieved (by budget id or by some query).</li> <li>◦ I created a story to calculate them on updating or saving budget, rather than doing it on get which is how it is working now.</li> <li>◦ <a href="https://issues.folio.org/browse/MODORDERS-837">https://issues.folio.org/browse/MODORDERS-837</a></li> <li>◦ For LDP to resolve their issue without waiting for this story to be released – they could use the API to retrieve finance budget data. Or as an workaround – make a request to retrieve necessary budgets before collecting data as after this all fields will be updated in DB.</li> <li>◦ Yes, these fields are calculated on the fly, here is a list of these fields in code that fully matches list provided by LDP (line 61-67):</li> <li>◦ these fields can be calculated in individual queries, but this is cumbersome</li> <li>◦ the data might be able to be extracted from the finance transactions table</li> <li>◦ Axel suggests we create a daily script to run against our LDP that will find every budget updated by the FOLIO API in order to trigger a daily update to the fields on the finance_budgets table</li> <li>◦ reports impacted: CR157, CR165, CR132</li> <li>◦ dashboards are not impacted</li> </ul> </li> </ul>	Workaround in place	<p>Bug submitted, to be fixed in Poppy release</p>
Fund Balance Discrepancy	1/12/23	<ul style="list-style-type: none"> <li>• discrepancy is between data in FOLIO and data shown the Fund Detail Tableau dashboard</li> <li>• The issue is when the FYRO was run at the end of FY22 – the balances were moved forward under the transaction type Rollover transfer. When capturing the Group view, the programmers didn't take into account the Rollover transfer within the Net transfer on the Group view. This has been submitted as a bug. Unfortunately, this will not be addressed until the Orchid release.</li> <li>• see <a href="https://issues.folio.org/browse/MODFIN-299">https://issues.folio.org/browse/MODFIN-299</a></li> <li>• Until FOLIO is updated – When looking at your Team information, please rely on the Tableau dashboard created by Nancy Bolduc.</li> <li>• Cornell goes to Nolana - end of Feb, beginning of March</li> </ul> <p>Cornell goes to Orchid - hopefully before Fiscal Year Rollover</p>	Unresolved	<p>Bug submitted, to be fixed in Orchid release</p>

In the public.srs_record s table, the column external_id is actually the instance_id		None needed		Fixed
In the public.srs_marctab table, all ids are in uuid format, and this does not match the format of ids in other tables		Cast all ids from tables that connect to srs_marctab as "::uuid"  Example code when joining srs_marctab AS sm to srs_records AS sr  <code>LEFT JOIN srs_marctab AS sm ON sr.id::uuid = sm.srs_id</code>		None
In the public.srs_marctab table, the instance_id column is incorrect and should not be used.		The srs_marctab table can be connected to the public.inventory_instances table via hrid, and the instance_id can be brought in from the inventory_instances table.  <b>Example code:</b>  <code>SELECT   ii.id AS instance_id,   ii.title AS title,   substring(sm."content", 7, 2) AS "format_type" FROM   public.inventory_instances AS ii LEFT JOIN srs_marctab AS sm   ON sm.instance_hrid =ii.hrid  WHERE sm.field = '000'</code>  Another way to get data from the srs_marctab table and insert a correct instance_id is via the srs_records table.  <b>Example code:</b>  <code>SELECT   sr.external_id AS instance_id,   substring (sm.content,7,2) AS format_code FROM   srs_records AS sr LEFT JOIN srs_marctab AS sm ON sr.id::uuid = sm.srs_id WHERE sm.field = '000'</code>  --NOTE that all ids in the srs_marctab table have been cast as uuids, so cast the OTHER table's id as ::uuid		<b>Fixed</b> , the instance_id column from srs.marctab has been corrected as of 9/12/22.
public.audit_circulation_logs did not start recording loan renewals until December 17, 2021.	17 Dec 2021	The circulation_loans table has a record of renewals (renewal count) starting 7/1/2021, but note that these are only for items checked out since 7/1/2021.		None
public.notes table does not contain any data, and will impact any queries using this table	updated 10-17-22	None		Resolved, but still has the contents field data that includes HTML tags  Upgrade 1.8 will solve this issue (update date to be determined). As of 10-17-22, LDP Test (1.8.2) shows the return of the Notes table.  -Sharon to ask Nassib
Invoice and fund tables migrated from Voyager show amounts 100 times greater than the actual amount	10/17/22	The data type for monetary amounts in Voyager finance tables was "Text," but when migrated to Folio was turned into "numeric." The decimals were stripped out – this means that when using Voyager historical data for funds and transaction amounts, divide your results by 100. Be careful to check your results for all queries that involve monetary values from Voyager.		No fix foreseen

To create a new FOLIO Enhancement request:  
<https://tdx.cornell.edu/TDClient/138/Portal/Home/>