

Useful Custom Formulas for Alma Analytics

- **Product:** Alma

Background

Alma Analytics (aka Oracle Analytics Server) is a general purpose reporting and business intelligence tool, which does not automatically "understand" data structures common in the library world (such as ISBN/ISSNs, call numbers, etc.).

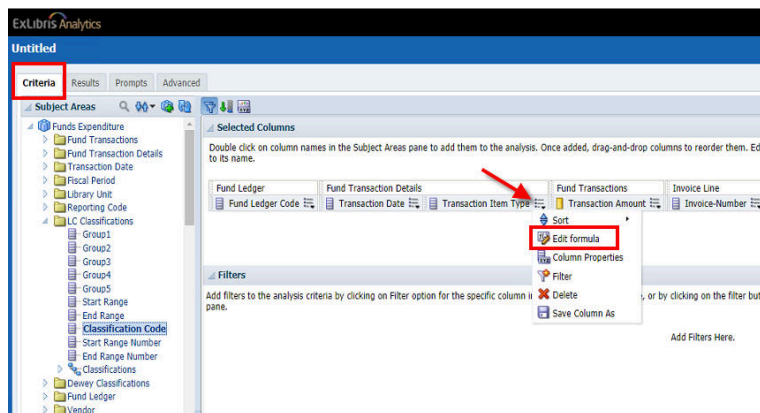
The custom formulas below have helped libraries manipulate data available in Analytics that was not originally formatted in a way that was useful for their reporting needs. These formulas can be easily changed to work for other types of fields or data.

Two of the most commonly used functions are `regexp_substr` and `regexp_replace`. For more information on these functions, please see the [Oracle documentation](#).

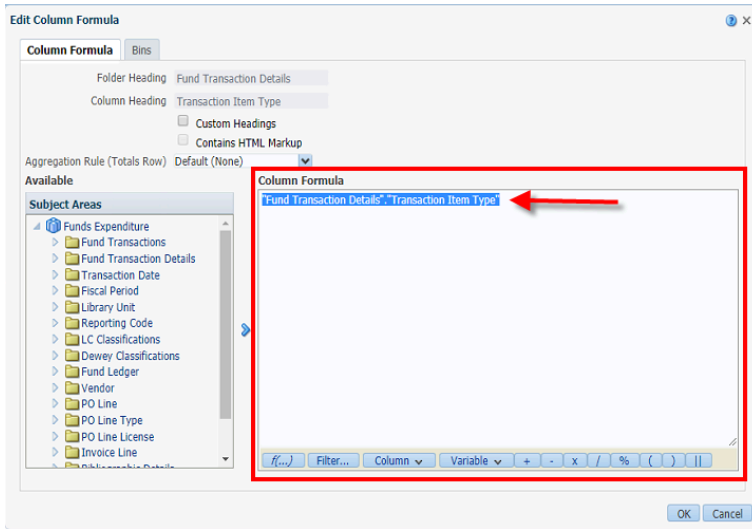
Instructions

To add or edit a formula in an Analytics report:

1. Open the 'Criteria' tab of your report. From the hamburger menu of the field you would like to change, select 'Edit Formula'.



2. In the 'Column Formula' text box, replace the current formula with your custom formula and click 'OK'.



Example formulas

Retrieve the first occurrence of an OCLC Number (change the second 1 to 2, 3, etc. for the second, third, etc. occurrence):

```
Evaluate('regexp_substr(%1,%2,%3,%4,%5,%6)', "Bibliographic Details"."Network Number",
'(\(?[oO0][cC]\w+\)?\W?\d+)', 1,1, NULL, 0)
```

Retrieve the first occurrence of subfield b in the 852 MARC field (the double-daggers character should be replaced by the delimiter used by your library, and the "b" may be changed to the desired subfield):

```
Evaluate('regexp_substr(%1,%2,%3,%4,%5,%6)', "Holding Details"."852 MARC", '#b([\#]+)',
1, 1, NULL, 0)
```

Note that some subfield delimiter characters may need to be escaped using a backslash. For example, if your library uses \$\$ as a subfield delimiter, the formula would look like this:

```
Evaluate('regexp_substr(%1,%2,%3,%4,%5,%6)', "Holding Details"."852 MARC",
'\$\$b([\^$\$]+)', 1, 1, NULL, 0)
```

Remove duplicate normalized ISSNs from the ISSN field:

```
Evaluate('regexp_replace(%1, '\; [0-9Xx]{8}', '')', "Bibliographic Details"."ISSN")
```

Extract numbers from a string field and convert them to a number data type for sorting purposes (for instance, if volume numbers were entered in the Description rather than Enum field):

```
CAST(Evaluate('regexp_substr(%1,%2,%3,%4,%5,%6)', "Physical Item
```

```
Details"."Description", 'v\.\s*(\d+)', 1, 1, NULL, 1) AS SMALLINT)
```

Create a measure for manual renewals (as Renewals combines manual and auto-renewals, and there is only a separate measure for Auto Renewals):

```
CASE WHEN SIGN("Loan"."Renewals"- "Loan"."Auto Renewals")=1 THEN "Loan"."Renewals"-  
"Loan"."Auto Renewals" ELSE 0 END
```

Combine the Loan Date and Loan Time fields into a single timestamp:

```
TO_DATETIME(CAST("Loan Date"."Loan Date" as CHAR) || ' ' || CAST("Loan Date"."Loan Time" as  
CHAR), 'dd-mon-yy hh:mi:ss')
```

Count the occurrences of a pattern (in this case the number of semicolons, which can be used to confirm how many values appear in a multi-value row):

```
Evaluate('regexp_count(%1,%2)', "Bibliographic Details"."ISSN", '\;')
```

Return a user's Preferred First Name if it exists, or else return their First Name:

```
CASE WHEN "User Details"."Preferred First Name" is not null THEN "User  
Details"."Preferred First Name" ELSE "User Details"."First Name" END
```

Pull the numbers from a subfield and convert them to an integer:

```
CAST(Evaluate('regexp_substr(%1,%2,%3,%4,%5,%6)', "Bibliographic Details"."Local Param  
26", '\$p (\d+)', 1, 1, NULL, 1) AS SMALLINT)
```

Additional Information

Additional tips, tricks, and useful formulas for Alma Analytics are available here: [Tips and Tricks for Alma Analytics](#)

More information is also available here: [Common Analytics Procedures](#)

-
- **Article last edited:** 12-Sep-2024