Parsing out call numbers

Use the following expressions to parse out the LC Class, LC Class Number, First Cutter, Second Cutter and internal year from the call number:

SUBSTRING (call_number, '^([a-zA-z]{1,3})') **AS lc_class**

SUBSTRING (call_number, '\d{1,}\.{0,}\d{0,}') **AS lc_class_number**

SUBSTRING (call_number, '\.{0,}[A-Z][0-9]{1,}') **AS first_cutter**

SUBSTRING (call_number, '\s{1}[A-Z]{1}\d{1,}x{0,}a{0,}b{0,}') **AS second_cutter**

SUBSTRING (call_number, '\s\d{4}\s\.') **AS internal_year**

Notes:

- for **lc_class_number**, you may want to cast it as NUMERIC if you are going to use it to sort by call number (otherwise it will sort as Text). It will then display with commas, but when you export it to Excel, the commas will disappear: **SUBSTRING (call_number,'\d{1,}\.{0,}\d{0,}')::NUMERIC as lc_class_number**
- **To get rid of a trailing "." at the end of the parsed-out LC class number, use this expression:**

  trim (**TRAILING** '.' **FROM substring** (he.call_number, '\d{1,}\.{0,}\d{0,}'))

- **For first and second cutters,** the result will show a decimal point followed by the letter and number components.

  To eliminate the leading decimal point, use these expression:

  first cutter: trim (LEADING '.' FROM SUBSTRING (call_number, '\.{0,}[A-Z][0-9]{1,}'))
  second cutter: trim (LEADING '.' FROM SUBSTRING (call_number, '\s{1}[A-Z]{1}\d{1,}x{0,}a{0,}b{0,}'))

- **internal_year** is for parsing out thesis years and other locally-entered numbers in a collection (Law items, for example)