Building SELECT lists with SQL

If you are using SQL for ingestion and you want to rename columns

PascalCase to snake_case

```
-- PascalCase to snake_case
-- , OtherNFLStats AS other_nfl_stats

SELECT
    CONCAT(', ', column_name, ' AS ', LOWER(
      REGEXP_REPLACE(
        REGEXP_REPLACE(column_name, '([a-z0-9])([A-Z])', '$1_$2'), -- Pass 1: Lower to Upper
        '([A-Z])([A-Z][a-z])', '$1_$2'                             -- Pass 2: Acronym to Word
      )
    )) AS formatted_column
FROM information_schema.columns
WHERE table_name = 'my_table'
    AND table_schema = 'my_schema'
ORDER BY ordinal_position;
```

PascalCase to snake_case with substitution

```
-- PascalCase to snake_case
-- , OverageAmount AS overage_amt
WITH RawColumns AS (
    SELECT  column_name
        , LOWER(
```

```
        REGEXP_REPLACE(
            REGEXP_REPLACE(column_name, '([a-z0-9])([A-Z])', '$1_$2')
            , '([A-Z])([A-Z][a-z])', '$1_$2'
        )
    ) AS snake_name
  FROM information_schema.columns
  WHERE table_name = 'my_table'
  AND table_schema = 'my_schema'
)
SELECT CONCAT(', ', column_name, ' AS ',
    AGGREGATE( -- AGGREGATE is Spark-specific (thus applicable to databricks)
        -- Define the list of [pattern, replacement]
        ARRAY(
            STRUCT('_amount$'      AS pat, '_amt' AS rep)
          , STRUCT('_percent$'     AS pat, '_pct' AS rep)
          , STRUCT('_flag$'        AS pat, '_flg' AS rep)
          , STRUCT('_code$'        AS pat, '_cd'  AS rep)
          , STRUCT('_number$'      AS pat, '_nbr' AS rep)
          , STRUCT('_name$'        AS pat, '_nm'  AS rep)
          , STRUCT('_description$' AS pat, '_dsc' AS rep)
          , STRUCT('_desc$'        AS pat, '_dsc' AS rep)
          , STRUCT('_date$'        AS pat, '_dts' AS rep)
        )
        , snake_name -- Initial value
        , (acc, x) -> REGEXP_REPLACE(acc, x.pat, x.rep) -- Iterative function
    )
  ) AS formatted_column
FROM RawColumns
ORDER BY column_name;
```

snake_case to PascalCase + custom rules + suffix

```
-- snake_case to PascalCase with suffix substitution
WITH base AS (
  SELECT
```

```
      column_name
    , CASE
        WHEN column_name LIKE '%\_%' THEN  ARRAY_JOIN( TRANSFORM(SPLIT(LOWER(column_name), '_'), x ->
INITCAP(x) ), '' )  -- underscore to PascalCase
        ELSE INITCAP(column_name)
      END AS pascal_name
    , ordinal_position
  FROM my_catalog.information_schema.columns
  WHERE table_schema = 'my_schema'
    AND table_name = 'my_table'
)
SELECT
  CASE
    -- Parsed Excel files
    WHEN column_name = '_FILENAME' THEN CONCAT(', ', 'regexp_extract(', column_name, ', \'[^/]+$\', 0)', '
AS FileNM')
    WHEN column_name = '_ROW_NUM' THEN CONCAT(', ', column_name, ' AS RowNBR')
    WHEN column_name = '_SHEETNAME' THEN CONCAT(', ', column_name, ' AS SheetNM')
    -- metadata cols
    WHEN column_name = 'hcsys_file_name' THEN CONCAT(', ', 'regexp_extract(', column_name, ', \'[^/]+$\',
0)', ' AS FileNM') -- SourceHCSystemFileNM?
    WHEN column_name = 'hcsys_row_num' THEN CONCAT(', ', column_name, ' AS SourceHCSystemRowNBR')
    WHEN column_name = 'meta_updated' THEN CONCAT(', ', column_name, ' AS  EDWLastModifiedDTS')
    WHEN column_name = 'meta_deleted' THEN CONCAT(', ', column_name, ' AS MetaDeletedFLG')
    WHEN column_name = 'meta_surrogate_key' THEN CONCAT(', ', column_name, ' AS MetaSurrogateKeyID')
    WHEN column_name = 'meta_checksum' THEN CONCAT(', ', column_name, ' AS MetaChecksumID')
    WHEN column_name = 'meta_location' THEN CONCAT(', ', column_name, ' AS MetaLocationDSC')
    -- Replace end word w/ suffix
    WHEN pascal_name RLIKE '(?i)Amount$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Amount$', 'AMT'))
    WHEN pascal_name RLIKE '(?i)Percent$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Percent$', 'PCT'))
    WHEN pascal_name RLIKE '(?i)Flag$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Flag$', 'FLG'))
    WHEN pascal_name RLIKE '(?i)Code$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Code$', 'CD'))
    WHEN pascal_name RLIKE '(?i)Number$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Number$', 'NBR'))
    WHEN pascal_name RLIKE '(?i)Name$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Name$', 'NM'))
```

```
        WHEN pascal_name RLIKE '(?i)Description$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Description$', 'DSC'))
        WHEN pascal_name RLIKE '(?i)Desc$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Desc$', 'DSC'))
        WHEN pascal_name RLIKE '(?i)Date$' THEN CONCAT(', try_to_timestamp(', column_name, ', \'yyyy-MM-dd
HH:mm:ss\') AS ', REGEXP_REPLACE(pascal_name, '(?i)Date$', 'DTS'))  -- chg to try_to_date if no time data
        WHEN pascal_name RLIKE '(?i)Count$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Count$', 'CNT'))
        WHEN pascal_name RLIKE '(?i)Id$' THEN CONCAT(', ', column_name, ' AS ', REGEXP_REPLACE(pascal_name,
'(?i)Id$', 'ID'))
        WHEN pascal_name RLIKE '(?i)Address1$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Address1$', 'Address01TXT'))
        WHEN pascal_name RLIKE '(?i)Address2$' THEN CONCAT(', ', column_name, ' AS ',
REGEXP_REPLACE(pascal_name, '(?i)Address2$', 'Address02TXT'))
        -- Appended suffixes
        WHEN pascal_name RLIKE '(?i)Gender$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name,
'CD'))
        WHEN pascal_name RLIKE '(?i)Age$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name, 'NBR'))
        WHEN pascal_name RLIKE '(?i)Address$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name,
'TXT'))
        WHEN pascal_name RLIKE '(?i)City$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name, 'NN'))
        WHEN pascal_name RLIKE '(?i)State$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name, 'NN'))

        WHEN pascal_name RLIKE '(?i)Zip$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name, 'CD'))
        WHEN pascal_name RLIKE '(?i)Phone$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name,
'NBR'))
        WHEN pascal_name RLIKE '(?i)Fax$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name, 'NBR'))
        WHEN pascal_name RLIKE '(?i)Active$' THEN CONCAT(', ', column_name, ' AS ', CONCAT(pascal_name,
'FLG'))
        ELSE CONCAT(', ', column_name, ' AS ', pascal_name, 'TXT')
    END AS c
FROM base
ORDER BY ordinal_position;
```

Revision #9
Created 17 February 2026 22:05:50 by Brian Dill
Updated 1 April 2026 16:13:34 by Brian Dill