Sample example using Rapt

Date	@August 30, 2021
Tags	

Source code of Rapt:

https://pypi.org/project/rapt/

install by:

```
pip install rapt
```

or alternatively added in Pycharm Library

Rapt requires pyparsing library as well

Import

```
from rapt.rapt import Rapt
```

Change from relation algebra to Sql

```
def change_to_sql():

    #Q1
    str = "\project_{BranName}(\s_{State='Indiana' } (LibraryBranch) );"


    with open('config.json') as config_file:
        config = json.load(config_file)
    r = Rapt(**config)
    f = open('schema.json')
    schema = json.load(f)

    return r.to_sql(str, schema)
```

str - refer to the the relational algebra

config.json includes the set of operators that supported by rapt

```
{
  "grammar": "Extended Grammar",
  "syntax": {
    "not_op": "not",
    "and_op": "and",
    "or_op": "or",

    "params_start": "_{",
    "params_stop": "}",
    "paren_left": "(",
    "paren_right": ")",

    "equal_op": "=",
    "not_equal_op": "!=",
    "not_equal_alt_op": "<>",
    "less_than_op": "<",
    "less_than_equal_op": "<=",
    "greater_than_op": ">",
    "greater_than_equal_op": ">=",

    "select_op": "\\s",
    "project_op": "\\project",
    "rename_op": "\\r",
    "assign_op": "\\leftarrow",

    "join_op": "\\j",
    "theta_join_op": "\\theta_join",
    "natural_join_op": "\\natural_join",
    "left_outer_join_op": "\\leftouterjoin",

    "union_op": "+",
    "difference_op": "-",
    "intersect_op": "\\i"
  }
}
```

schema.json includes the data structure

```
{
  "Book": [
    "BookId",
    "Title",
    "PublId"
  ],
  "Author": [
    "AuthId",
    "AuthName"
  ],
  "AuthorBook": [
    "AuthId",
    "BookId"
  ],
  "Publisher": [
    "PublID",
    "PublName",
    "Address",
    "Phone"
  ],
  "BookCopies": [
    "BookId",
    "BranId",
    "Copies"
  ],
  "BookLoans": [
    "BookId",
    "BranId",
    "MembId",
    "IssueDate",
    "DueDate"
  ],
  "Member": [
    "MembId",
    "MembName",
    "Address",
    "Phone"
  ],
  "LibraryBranch": [
    "BranId",
    "BranName",
    "State"
  ]
}
```

change_to_sql function will return list of corresponding sql lines

Then, print out their corresponding sql line at the end.

```
if __name__ == '__main__':
    sql_list = change_to_sql()

    print('SQL: Queries')

    for q in sql_list:
        print(q)

    print()
```