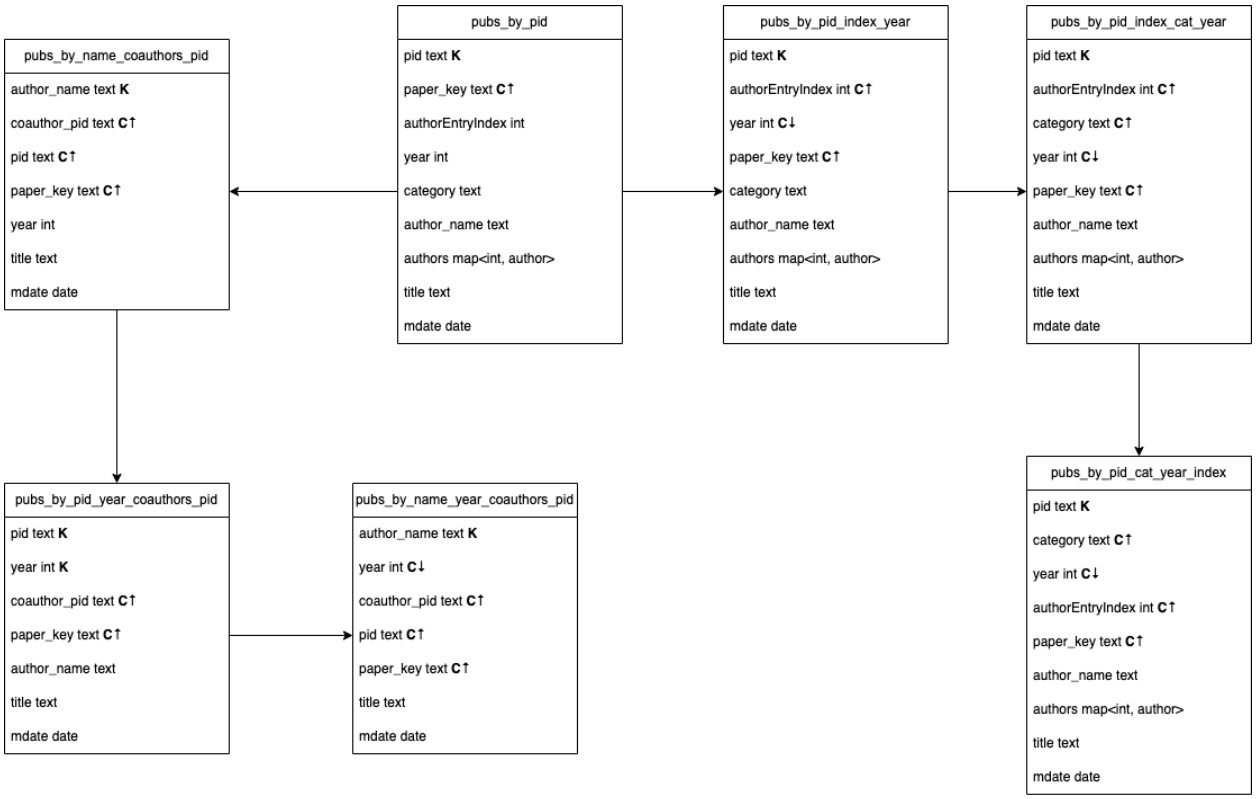
IS3107 Individual Course Project (Sem1 AY2022/2023)
Data Pipeline Implementation with DBLP Data

Name: Andre Heng
Matric No: A0233806H
Email: e0725806@u.nus.edu

1. Physical Data Model of Local Cassandra Database



- The tables of the database fit the queries for the queries below and the query outcomes stored in the cloud database.
- Minimizing the number of partitions being read

Q2.1.1

Given the PID of a researcher, find all the publications of a certain category he/she gets published and signs as the 1st/2nd/3rd… author in a certain year.

Q1. SELECT title FROM pubs_by_pid_cat_year_index WHERE pid='pid' AND category = 'category' AND year = 'year' AND authorEntryIndex < 4
Primary key((pid), category, year, authorEntryIndex, paper_key)

Q2.1.2

Given the name/PID of a researcher, find the number of times he/she has collaborated with any one of his/her coauthors in a certain year.

Q2a. SELECT count(*) FROM pubs_by_pid_year_coauthors_pid WHERE pid='pid' AND year = 'year' GROUP BY coauthors_pid
Primary key((pid, year), coauthors_pid, paper_key)

Q2b. SELECT count(*) FROM pubs_by_name_year_coauthors_pid WHERE author_name=author_name AND year = 'year' GROUP BY coauthors_pid
Primary key((author_name, year), coauthors_pid, pid, paper_key)

2. CQL DDL & DML statements

Connect to Cassandra

```python
from cassandra.cluster import Cluster

cluster = Cluster(['127.0.0.1'])
session = cluster.connect()
```

✓ 0.4s Python

Keyspace setup

```python
# Drop Keyspace localdb
session.execute('DROP KEYSPACE IF EXISTS localdb;')

# Create Keyspace localdb
session.execute(
    "CREATE KEYSPACE IF NOT EXISTS localdb WITH REPLICATION = {'class' : 'SimpleStrategy', 'replication_factor': '1'}")
session.set_keyspace('localdb')

# Create and Register UDTs
session.execute(
    "CREATE TYPE author (author_name text, orcid text, pid text)")
create_position = '''
        CREATE TYPE position (
                number text,
                volume text,
                pages text
            );
        '''
session.execute(create_position)
cluster.register_user_type('localdb', 'author', dict)
cluster.register_user_type('localdb', 'position', dict)
```

✓ 2.8s Python

Drop Tables

```python
# Drop
# 0
drop_table_pubs_by_pid = '''
            DROP TABLE IF EXISTS localdb.pubs_by_pid;
            '''
session.execute(drop_table_pubs_by_pid)
# 1
drop_table_pubs_by_pid_cat_year_index = '''
            DROP TABLE IF EXISTS localdb.pubs_by_pid_cat_year_index;
            '''
session.execute(drop_table_pubs_by_pid_cat_year_index)
# 2
drop_table_pubs_by_pid_year_coauthors_pid = '''
            DROP TABLE IF EXISTS localdb.pubs_by_pid_year_coauthors_pid;
            '''
session.execute(drop_table_pubs_by_pid_year_coauthors_pid)
# 3
drop_table_pubs_by_name_year_coauthors_pid = '''
            DROP TABLE IF EXISTS localdb.pubs_by_name_year_coauthors_pid;
            '''
session.execute(drop_table_pubs_by_name_year_coauthors_pid)
# 4
drop_table_pubs_by_pid_index_cat_year = '''
            DROP TABLE IF EXISTS localdb.pubs_by_pid_index_cat_year;
            '''
session.execute(drop_table_pubs_by_pid_index_cat_year)
# 5
drop_table_pubs_by_pid_index_year = '''
            DROP TABLE IF EXISTS localdb.pubs_by_pid_index_year;
            '''
session.execute(drop_table_pubs_by_pid_index_year)
# 6
drop_table_pubs_name_coauthors_pid = '''
            DROP TABLE IF EXISTS localdb.pubs_name_coauthors_pid;
            '''
session.execute(drop_table_pubs_name_coauthors_pid)
```

✓ 0.1s Python

Create Tables

```
# pubs_by_pid
create_table_pubs_by_pid = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid (
                pid text,
                paper_key text,
                authorEntryIndex int,
                category text,
                year int,
                author_name text,
                authors map<int, frozen<author>>,
                title text,
                mdate date,
                PRIMARY KEY (pid, paper_key)
            ) WITH CLUSTERING ORDER BY (paper_key ASC) AND
            comment = 'pubs_by_pid';'''
session.execute(create_table_pubs_by_pid)

# pubs_by_pid_cat_year_index
create_table_pubs_by_pid_cat_year_index = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid_cat_year_index (
                pid text,
                paper_key text,
                authorEntryIndex int,
                category text,
                year int,
                author_name text,
                authors map<int, frozen<author>>,
                title text,
                mdate date,
                PRIMARY KEY (pid, category, year, authorEntryIndex, paper_key)
            ) WITH CLUSTERING ORDER BY (category ASC, year DESC, authorEntryIndex ASC, paper_key ASC) AND
            comment = 'pubs_by_pid_cat_year_index';'''
session.execute(create_table_pubs_by_pid_cat_year_index)

# pubs_by_pid_year_coauthors_pid
create_table_pubs_by_pid_year_coauthors_pid = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid_year_coauthors_pid (
                pid text,
                author_name text,
                coauthor_pid text,
                paper_key text,
                year int,
                title text,
                mdate date,
                PRIMARY KEY ((pid, year), coauthor_pid, paper_key)
            ) WITH CLUSTERING ORDER BY (coauthor_pid ASC, paper_key ASC) AND
            comment = 'pubs_by_pid_year_coauthors_pid';'''
session.execute(create_table_pubs_by_pid_year_coauthors_pid)

# pubs_by_name_year_coauthors_pid
create_table_pubs_by_name_year_coauthors_pid = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_name_year_coauthors_pid (
                pid text,
                author_name text,
                coauthor_pid text,
                paper_key text,
                year int,
                title text,
                mdate date,
                PRIMARY KEY ((author_name, year), coauthor_pid, pid, paper_key)
            ) WITH CLUSTERING ORDER BY (coauthor_pid ASC, pid ASC, paper_key ASC) AND
            comment = 'pubs_by_name_year_coauthors_pid';'''
session.execute(create_table_pubs_by_name_year_coauthors_pid)

# pubs_by_pid_index_cat_year
create_table_pubs_by_pid_index_cat_year = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid_index_cat_year (
                pid text,
                paper_key text,
                authorEntryIndex int,
                category text,
                year int,
                author_name text,
                authors map<int, frozen<author>>,
                title text,
                mdate date,
                PRIMARY KEY (pid, authorEntryIndex, category, year, paper_key)
            ) WITH CLUSTERING ORDER BY (authorEntryIndex ASC, category ASC, year DESC, paper_key ASC) AND
            comment = 'pubs_by_pid_index_cat_year';'''
session.execute(create_table_pubs_by_pid_index_cat_year)
```

```python
# pubs_by_pid_index_cat_year
create_table_pubs_by_pid_index_cat_year = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid_index_cat_year (
                pid text,
                paper_key text,
                authorEntryIndex int,
                category text,
                year int,
                author_name text,
                authors map<int, frozen<author>>,
                title text,
                mdate date,
                PRIMARY KEY (pid, authorEntryIndex, category, year, paper_key)
            ) WITH CLUSTERING ORDER BY (authorEntryIndex ASC, category ASC, year DESC, paper_key ASC) AND
            comment = 'pubs_by_pid_index_cat_year';'''
session.execute(create_table_pubs_by_pid_index_cat_year)

# pubs_by_pid_index_year
create_table_pubs_by_pid_index_year = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_pid_index_year (
                pid text,
                paper_key text,
                authorEntryIndex int,
                category text,
                year int,
                author_name text,
                authors map<int, frozen<author>>,
                title text,
                mdate date,
                PRIMARY KEY (pid, authorEntryIndex, year, paper_key)
            ) WITH CLUSTERING ORDER BY (authorEntryIndex ASC, year DESC, paper_key ASC) AND
            comment = 'pubs_by_pid_index_year';'''
session.execute(create_table_pubs_by_pid_index_year)

# pubs_by_name_coauthors_pid
create_table_pubs_by_name_coauthors_pid = '''CREATE TABLE IF NOT EXISTS localdb.pubs_by_name_coauthors_pid (
                pid text,
                author_name text,
                coauthor_pid text,
                paper_key text,
                year int,
                title text,
                mdate date,
                PRIMARY KEY (author_name, coauthor_pid, pid, paper_key)
            ) WITH CLUSTERING ORDER BY (coauthor_pid ASC, pid ASC, paper_key ASC) AND
            comment = 'pubs_by_name_coauthors_pid';'''
session.execute(create_table_pubs_by_name_coauthors_pid)
```

✓ 8.4s Python

Prepare Insert Statements

```python
#index
insert_data_pubs_by_pid = '''
    INSERT INTO localdb.pubs_by_pid (
        pid,
        paper_key,
        authorEntryIndex,
        category,
        year,
        author_name,
        authors,
        title,
        mdate
    ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
    '''
insert_stmt1 = session.prepare(insert_data_pubs_by_pid)

insert_data_pubs_by_pid_index_year = '''
    INSERT INTO localdb.pubs_by_pid_index_year (
        pid,
        paper_key,
        authorEntryIndex,
        category,
        year,
        author_name,
        authors,
        title,
        mdate
    ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
    '''
insert_stmt2 = session.prepare(insert_data_pubs_by_pid_index_year)
```

Insert Initial Data

```
import requests
import pandas as pd
import xml.etree.ElementTree as ET

cs_researchers = pd.read_csv("/Users/andre/Plan3 Design & Build Dropbox/Andre Heng/Mac/Documents/NUS Y2S1/IS3107/IS3107 Mini Project/cs_researchers.csv")

def check_if_pub_exists(paper_key):

    create_idx_q1 = '''CREATE INDEX IF NOT EXISTS ON pubs_by_pid (paper_key);'''

    probe_query = f'''
        SELECT title FROM pubs_by_pid WHERE paper_key = '{paper_key}';
    '''
    session.execute(create_idx_q1)
    rows = session.execute(probe_query)
    check = []
    for (title) in rows:
        check.append({'title':title})
    check = pd.DataFrame(check)
    if check.empty:
        return True
    else:
        return False
```

```
def extract_from_xml(file_to_process):
    pub_list = []
    tree = ET.parse(file_to_process)
    root = tree.getroot()
    type_of_pub_arr = ("article", "inproceedings", "proceedings", "book", "incollection", "phdthesis", "masterthesis", "www", "person", "data")
    type_of_author_arr = ("author", "editor")

    for dblpperson in root.iter('dblpperson'):
        pid = dblpperson.attrib['pid']
        author_name = dblpperson.attrib['name']

    for item in root.findall('r'):

        for child in item:

            pub_attribs = {}
            pub_attribs["pid"] = pid
            pub_attribs["author_name"] = author_name

            for i in type_of_pub_arr:
                type = item.find(i)
                if type is not None:
                    paper_key = type.attrib['key']
                    pub_attribs["paper_key"] = paper_key
                    pub_attribs["mdate"] = type.attrib['mdate']
                    break
```

```
            coauthors_pid = []
            authors = {}
            count = 1
            for i in type_of_author_arr:
                for auth in child.findall(i):
                    name = auth.text
                    if auth.get('orcid'):
                        orcid = auth.attrib['orcid']
                    else:
                        orcid = None
                    diff_pid = auth.attrib['pid']
                    if diff_pid == pub_attribs["pid"]:
                        pub_attribs["authorEntryIndex"] = count
                    else:
                        coauthors_pid.append(diff_pid)

                    authors[count] = (name, orcid, diff_pid)
                    count += 1

                pub_attribs["authors"] = authors
                pub_attribs["coauthors_pid"] = coauthors_pid
                # if pub_attribs['authorEntryIndex'] is None:
                #     pub_attribs['authorEntryIndex'] = "HAHAHAHAHAHBABABABABABABA~!!!"

            pub_attribs["category"] = pub_attribs["paper_key"].split('/')[0].rstrip('s')

            year = child.find('year')
            if year is not None:
                year = int(year.text)
                pub_attribs["year"] = year

            title = child.find("title")
            if title is not None:
                title = title.text
                pub_attribs["title"] = title

            ees = set()
            for one_ee in child.findall("ee"):
                ees.add(one_ee.text)
            pub_attribs["ee"] = ees

            #do check here before appending
            if check_if_pub_exists(paper_key) == True:
                pub_list.append(pub_attribs)
                # print(pub_list)
            # print(pub_list)
    # print(pub_list)
    return pub_list
```

```python
def load_pub_ls(pub_ls):
    for pub in pub_ls:
        session.execute(insert_stmt1,
                        [pub['pid'], pub['paper_key'], pub['authorEntryIndex'], pub['category'], pub['year'], pub['author_name'], pub['authors'], pub['title'], pub['mdate']])

        session.execute(insert_stmt2,
                        [pub['pid'], pub['paper_key'], pub['authorEntryIndex'], pub['category'], pub['year'], pub['author_name'], pub['authors'], pub['title'], pub['mdate']])

        session.execute(insert_stmt3,
                        [pub['pid'], pub['paper_key'], pub['authorEntryIndex'], pub['category'], pub['year'], pub['author_name'], pub['authors'], pub['title'], pub['mdate']])

        session.execute(insert_stmt4,
                        [pub['pid'], pub['paper_key'], pub['authorEntryIndex'], pub['category'], pub['year'], pub['author_name'], pub['authors'], pub['title'], pub['mdate']])

        for coauthor_pid in pub['coauthors_pid']:
                session.execute(insert_stmt5,
                                [pub['pid'], pub['author_name'], coauthor_pid, pub['paper_key'], pub['year'], pub['title'], pub['mdate']])

                session.execute(insert_stmt6,
                                [pub['pid'], pub['author_name'], coauthor_pid, pub['paper_key'], pub['year'], pub['title'], pub['mdate']])

                session.execute(insert_stmt7,
                                [pub['pid'], pub['author_name'], coauthor_pid, pub['paper_key'], pub['year'], pub['title'], pub['mdate']])
    return pub_ls

api = "https://dblp.org/pid/"
for index, row in cs_researchers.iterrows():
    pid = row['PID']
    name = row['Name']
    current_api = api + pid
    current_api += ".xml"

    response = requests.get(current_api)
    if response.status_code == 200:
        with open('xmlfile.xml', 'wb+') as f:
            f.write(response.content)
            f.close()

        pub_ls = extract_from_xml('xmlfile.xml')
        load_pub_ls(pub_ls)
```

✓ 15m 24.1s Python

Astra DB Set Up

Connect to AstraDB

```
from cassandra.auth import PlainTextAuthProvider

#append files accordingly
cloud_config= {
        'secure_connect_bundle': '/Users/andre/Plan3 Design & Build Dropbox/Andre Heng/Mac/Documents/NUS Y2S1/IS3107/IS3107 Mini Project/secure-connect-is3107astradb.zip'
}
auth_provider = PlainTextAuthProvider('XZGMaIyaQCJBSOZOXBxQHGrX', '7KhXw2ywLrDsJDNryIP1mTohNE5lDgxccpqjKjl17EPDIAhY,-WbqTKv8,k6IZ.hrKxr-g5xZ2eL39QBywTAJZCyO3j5euY2mWS85dpQxh9qYdLZ7QwsaUsh,hrTb
```

✓ 0.7s Python

Astra DB Setup

```
cluster = Cluster(cloud=cloud_config, auth_provider=auth_provider)
session = cluster.connect()
session.set_keyspace('astradb')

row = session.execute("select release_version from system.local").one()
if row:
    print(row[0])
else:
    print("An error occurred.")

# Create and Register UDTs
session.execute(
    "CREATE TYPE author (author_name text, orcid text, pid text)")
create_position = '''
    CREATE TYPE position (
            number text,
            volume text,
            pages text
        );
    '''
session.execute(create_position)
cluster.register_user_type('astradb', 'author', dict)
cluster.register_user_type('astradb', 'position', dict)
```

Python

Drop AstraDB Tables

```
# Drop
# 0
drop_table_volume_update = '''
            DROP TABLE IF EXISTS volume_update;
            '''
session.execute(drop_table_volume_update)
# 1
drop_table_author_pub_update = '''
            DROP TABLE IF EXISTS author_pub_update;
            '''
session.execute(drop_table_author_pub_update)
# 2
drop_table_query_outcomes = '''
            DROP TABLE IF EXISTS query_outcomes;
            '''
session.execute(drop_table_query_outcomes)
```

✓ 0.6s Python

```
<cassandra.cluster.ResultSet at 0x119028688>
```

```
#create tables
create_tab_1 = '''
CREATE TABLE IF NOT EXISTS volume_update (
    timestamp text,
    num_new_pubs int,
    num_unique_pubs int,
    PRIMARY KEY (timestamp)
) WITH comment='astradb volume update log';
'''
session.execute(create_tab_1)
```

```
create_tab_2 = '''
CREATE TABLE IF NOT EXISTS author_pub_update (
    timestamp text,
    paper_key text,
    title text,
    authors map<int, frozen<author>>,
    ee set<text>,
    PRIMARY KEY (timestamp, paper_key)
) WITH comment='astradb author pub update log';
'''
session.execute(create_tab_2)

create_tab_3 = '''
CREATE TABLE IF NOT EXISTS query_outcomes (
    query_num text,
    query_outcome text,
    PRIMARY KEY (query_num)
) WITH comment='query_outcomes';
'''
session.execute(create_tab_3)
```

✓ 1.4s Python

3. Tables with data for local and cloud database

```
cqlsh:localdb> select * from pubs_by_pid;

 pid            | paper_key                     | author_name        | authorentryindex | authors

                                                                                                                                                                        | category | m
date      | title                                                                                                    | year
----------------+-------------------------------+--------------------+------------------+------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------+----------+--
----------+------------------------------------------------------------------------------------------------------------+------
       145/9981 |      conf/3dim/AlhaijaMTJN0R20 |        Justus Thies |                3 |
                             {1: {author_name: 'Hassan Abu Alhaija', orcid: null, pid: '169/1205'}, 2: {author_name: 'Siva Karthik Mustikovela', orcid: null, pid: '188/6172'},
 3: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 4: {author_name: 'Varun Jampani', orcid: null, pid: '124/2785'}, 5: {author_name: 'Matthias Nießner', orcid: null
, pid: '84/8221'}, 6: {author_name: 'Andreas Geiger 0001', orcid: null, pid: '40/5825-1'}, 7: {author_name: 'Carsten Rother', orcid: null, pid: 'r/CarstenRother'}} |     conf | 2
021-02-01 |              Intrinsic Autoencoders for Joint Deferred Neural Rendering and Intrinsic Image Decomposition. | 2020
       145/9981 |       conf/cvpr/AzinovicMGNT22 |        Justus Thies |                5 |

      {1: {author_name: 'Dejan Azinovic', orcid: null, pid: '197/9522'}, 2: {author_name: 'Ricardo Martin-Brualla', orcid: null, pid: '16/7968'}, 3: {author_name: 'Dan B. Goldman
', orcid: null, pid: '28/5892'}, 4: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 5: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}} |     conf | 2
022-10-05 |                                                                  Neural RGB-D Surface Reconstruction. | 2022
       145/9981 |         conf/cvpr/BozicPZTDN21 |        Justus Thies |                4 |
                                                                                                                               {1: {author_name: 'Aljaz Bozic', orcid: null, pid:
'203/9936'}, 2: {author_name: 'Pablo R. Palafox', orcid: null, pid: '248/0218'}, 3: {author_name: 'Michael Zollhöfer', orcid: null, pid: '52/8573'}, 4: {author_name: 'Justus Thie
s', orcid: null, pid: '145/9981'}, 5: {author_name: 'Angela Dai', orcid: null, pid: '149/1202'}, 6: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |     conf | 2
022-07-18 |                                      Neural Deformation Graphs for Globally-Consistent Non-Rigid Reconstruction. | 2021
       145/9981 |      conf/cvpr/CozzolinoTRNV21 |        Justus Thies |                2 |

           {1: {author_name: 'Davide Cozzolino', orcid: null, pid: '120/2275'}, 2: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 3: {author_name: 'Andreas Rössler'
, orcid: null, pid: '217/2041'}, 4: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 5: {author_name: 'Luisa Verdoliva', orcid: null, pid: '62/92'}} |     conf | 2
022-07-18 |                                                                   SpoC: Spoofing Camera Fingerprints. | 2021
       145/9981 |           conf/cvpr/DaiSTVN21 |        Justus Thies |                3 |

               {1: {author_name: 'Angela Dai', orcid: null, pid: '149/1202'}, 2: {author_name: 'Yawar Siddiqui', orcid: null, pid: '213/7787'}, 3: {author_name: 'Justus Thies', o
rcid: null, pid: '145/9981'}, 4: {author_name: 'Julien Valentin', orcid: null, pid: '21/11088'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |     conf | 2
022-07-18 |                                 SPSG: Self-Supervised Photometric Scene Generation From RGB-D Scans. | 2021
       145/9981 |          conf/cvpr/GafniTZN21 |        Justus Thies |                2 |
```

```
cqlsh:localdb> select * from pubs_by_name_coauthors_pid;;

 author_name   | coauthor_pid | pid      | paper_key                         | mdate      | title
                              | year
---------------+--------------+----------+-----------------------------------+------------+------------------------------------------------------------------------------
-------------------------------+------
 Sherjil Ozair |       04/8342 | 139/0736 |    journals/corr/abs-2206-15378 | 2022-09-28 |                                      Mastering the Game of Stratego with Model-Free Multi
agent Reinforcement Learning. | 2022
 Sherjil Ozair |        05/726 | 139/0736 |           conf/icml/OzairLRAOV21 | 2021-08-25 |                                                                                 Vector Q
uantized Models for Planning. | 2021
 Sherjil Ozair |        05/726 | 139/0736 |    journals/corr/abs-2106-04615 | 2021-06-15 |                                                                                 Vector Q
uantized Models for Planning. | 2021
 Sherjil Ozair |       05/7814 | 139/0736 |      conf/icml/AmodeiABCCCCCD16 | 2019-07-22 |                                                Deep Speech 2 : End-to-End Speech Recogni
tion in English and Mandarin. | 2016
 Sherjil Ozair |       05/7814 | 139/0736 | journals/corr/AmodeiABCCCCCD15 | 2019-07-22 |                                                 Deep Speech 2: End-to-End Speech Recogni
tion in English and Mandarin. | 2015
 Sherjil Ozair |       06/1767 | 139/0736 |      conf/icml/AmodeiABCCCCCD16 | 2019-07-22 |                                                Deep Speech 2 : End-to-End Speech Recogni
tion in English and Mandarin. | 2016
 Sherjil Ozair |       06/1767 | 139/0736 | journals/corr/AmodeiABCCCCCD15 | 2019-07-22 |                                                 Deep Speech 2: End-to-End Speech Recogni
tion in English and Mandarin. | 2015
 Sherjil Ozair |       08/8178 | 139/0736 |      conf/icml/AmodeiABCCCCCD16 | 2019-07-22 |                                                Deep Speech 2 : End-to-End Speech Recogni
tion in English and Mandarin. | 2016
 Sherjil Ozair |       08/8178 | 139/0736 | journals/corr/AmodeiABCCCCCD15 | 2019-07-22 |                                                 Deep Speech 2: End-to-End Speech Recogni
tion in English and Mandarin. | 2015
 Sherjil Ozair |       09/3058 | 139/0736 |    journals/corr/abs-2206-15378 | 2022-09-28 |                                      Mastering the Game of Stratego with Model-Free Multi
agent Reinforcement Learning. | 2022
 Sherjil Ozair |      116/4760 | 139/0736 |    journals/corr/abs-1905-09334 | 2019-05-29 |                                          The Journey is the Reward: Unsupervised Learning
 of Influential Trajectories. | 2019
 Sherjil Ozair |      117/0712 | 139/0736 |      conf/icml/AmodeiABCCCCCD16 | 2019-07-22 |                                                Deep Speech 2 : End-to-End Speech Recogni
tion in English and Mandarin. | 2016
 Sherjil Ozair |      117/0712 | 139/0736 | journals/corr/AmodeiABCCCCCD15 | 2019-07-22 |                                                 Deep Speech 2: End-to-End Speech Recogni
tion in English and Mandarin. | 2015
 Sherjil Ozair |      118/3205 | 139/0736 |           conf/icml/OzairLRAOV21 | 2021-08-25 |                                                                                 Vector Q
uantized Models for Planning. | 2021
 Sherjil Ozair |      118/3205 | 139/0736 |            conf/icml/PooleOOAT19 | 2022-04-19 |                                                                               On Variational
Bounds of Mutual Information. | 2019
```

```
cqlsh:localdb> select * from pubs_by_pid_year_coauthors_pid;

 pid                | year | coauthor_pid | paper_key                     | author_name        | mdate      | title
--------------------+------+--------------+-------------------------------+--------------------+------------+----------------------------------------------------------------------------
     s/ReidGSimmons | 2021 |     130/8276 |              conf/chi/MashRS21 |    Reid G. Simmons | 2021-07-25 | DSWorkFlow: A Framework for Capturing Data Scientists' Workflows.
     s/ReidGSimmons | 2021 |     159/0410 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |     192/7165 |             conf/hri/KaushikS21 |    Reid G. Simmons | 2021-07-26 | Perception of Emotion in Torso and Arm Movements on Humanoid Robot Quori.
     s/ReidGSimmons | 2021 |     192/7165 |          conf/socrob/KaushikS21 |    Reid G. Simmons | 2021-11-05 | Early Prediction of Student Engagement-Related Events from Facial and Contextual Features.
     s/ReidGSimmons | 2021 |     201/5416 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |      22/4189 |              conf/chi/MashRS21 |    Reid G. Simmons | 2021-07-25 | DSWorkFlow: A Framework for Capturing Data Scientists' Workflows.
     s/ReidGSimmons | 2021 |     299/5231 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |     299/5231 |           conf/ijcai/KoppolAS21 |    Reid G. Simmons | 2021-08-25 | Interaction Considerations in Learning from Humans.
     s/ReidGSimmons | 2021 |      44/7075 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |      44/7075 |           conf/ijcai/KoppolAS21 |    Reid G. Simmons | 2021-08-25 | Interaction Considerations in Learning from Humans.
     s/ReidGSimmons | 2021 |      44/7075 |          journals/firai/LeeAS21 |    Reid G. Simmons | 2021-07-26 | Machine Teaching for Human Inverse Reinforcement Learning.
     s/ReidGSimmons | 2021 |      53/3346 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |      62/8399 |          conf/ijcai/CuiKANSSF21 |    Reid G. Simmons | 2021-11-05 | Understanding the Relationship between Interactions and Outcomes in Human-in-the-Loop Machine Learning.
     s/ReidGSimmons | 2021 |      88/1572 |          journals/firai/LeeAS21 |    Reid G. Simmons | 2021-07-26 | Machine Teaching for Human Inverse Reinforcement Learning.
            46/2181 | 2006 |      07/1026 |             conf/vee/YuGNLC06 |            Yang Yu | 2018-11-06 | A feather-weight virtual machine for windows applications.
```

```
cqlsh:localdb> select * from pubs_by_name_year_coauthors_pid;

 author_name        | year | coauthor_pid       | pid          | paper_key                         | mdate      | title
--------------------+------+--------------------+--------------+-----------------------------------+------------+------------------------------------------------------------------------
     Yongqiang Wang | 2017 |             02/739 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           149/0431 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           153/4444 |      69/4019 |          conf/ksem/ZhangWQLW17 | 2017-08-14 | Collaborative Filtering Based on Pairwise User-Item Blocking Structure (PBCF): A General Framework and Its Implementation.
     Yongqiang Wang | 2017 |           159/2069 |      69/4019 |              conf/ccs/RuanAW17 | 2018-11-06 | Secure and Privacy-Preserving Average Consensus.
     Yongqiang Wang | 2017 |           159/2069 |      69/4019 |          journals/corr/RuanW17 | 2018-08-13 | Secure and Privacy-Preserving Average Consensus.
     Yongqiang Wang | 2017 |           170/6430 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           186/1764 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           193/8320 |      69/4019 |        journals/corr/AngleaW17 | 2018-08-13 | Synchronization with Guaranteed Clock Continuity using Pulse-Coupled Oscillators.
     Yongqiang Wang | 2017 |           193/8320 |      69/4019 |         journals/tsp/AngleaW17 | 2020-03-10 | Phase Desynchronization: A New Approach and Theory Using Pulse-Based Interaction.
     Yongqiang Wang | 2017 |           206/5502 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           206/5691 |      69/4019 |  journals/sensors/LiuYQWLLLA17 | 2018-11-14 | Curvature and Temperature Measurement Based on a Few-Mode PCF Formed M-Z-I and an Embedded FBG.
     Yongqiang Wang | 2017 |           215/1913 |      69/4019 |              conf/ccs/RuanAW17 | 2018-11-06 | Secure and Privacy-Preserving Average Consensus.
     Yongqiang Wang | 2017 |            45/5247 |      69/4019 |          conf/ksem/ZhangWQLW17 | 2017-08-14 | Collaborative Filtering Based on Pairwise User-Item Blocking Structure (PBCF): A General Framework and Its Implementation.
     Yongqiang Wang | 2017 |            65/6731 |      69/4019 |       journals/corr/ZhangW17aa | 2018-08-13 | Privacy-preserving Decentralized Optimization Based on ADMM.
     Yongqiang Wang | 2017 |             74/918 |      69/4019 |          conf/ksem/ZhangWQLW17 | 2017-08-14 |
```

```
cqlsh:localdb> select * from pubs_by_pid_index_year;;

 pid              | authorentryindex | year | paper_key                      | author_name        | authors | category | mdate      | title
------------------+------------------+------+--------------------------------+--------------------+---------+----------+------------+---------------------------------------------
        145/9981 |                1 | 2020 |          conf/iclr/ThiesZTSN20 |        Justus Thies | {1: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 2: {author_name: 'Michael Zollhöfer', orcid: null, pid: '52/8573'}, 3: {author_name: 'Christian Theobalt', orcid: null, pid: '55/3346'}, 4: {author_name: 'Marc Stamminger', orcid: null, pid: 's/MarcStamminger'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |     conf | 2020-05-07 | Image-guided Neural Object Rendering.
        145/9981 |                1 | 2020 |   journals/corr/abs-2007-14808 |        Justus Thies | {1: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 2: {author_name: 'Michael Zollhöfer', orcid: null, pid: '52/8573'}, 3: {author_name: 'Marc Stamminger', orcid: null, pid: 's/MarcStamminger'}, 4: {author_name: 'Christian Theobalt', orcid: null, pid: '55/3346'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |  journal | 2020-08-03 | Face2Face: Real-time Face Capture and Reenactment of RGB Videos.
        145/9981 |                1 | 2019 |      journals/cacm/ThiesZSTN19 |        Justus Thies | {1: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 2: {author_name: 'Michael Zollhöfer', orcid: null, pid: '52/8573'}, 3: {author_name: 'Marc Stamminger', orcid: null, pid: 's/MarcStamminger'}, 4: {author_name: 'Christian Theobalt', orcid: null, pid: '55/3346'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |  journal | 2019-05-09 | Face2Face: real-time face capture and reenactment of RGB videos.
        145/9981 |                1 | 2019 |   journals/corr/abs-1904-12356 |        Justus Thies | {1: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 2: {author_name: 'Michael Zollhöfer', orcid: null, pid: '52/8573'}, 3: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}} |  journal | 2019-05-02 | Deferred Neural Rendering: Image Synthesis using Neural Textures.
        145/9981 |                1 | 2019 |            journals/it/Thies19 |        Justus Thies | {1: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}} |  jour
```

```
---MORE---select * from pubs_by_pid_index_cat_year;
 pid               | authorentryindex | year | paper_key                          | author_name         | authors



                                                                                                                                      | category | mdate      | title
------------------+------------------+------+------------------------------------+---------------------+--------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------+----------+------------+-------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
  c/SatishChandra |                1 | 2006 |   journals/ibmsj/ChandraVFHKKNRX06 | Satish Chandra 0001 |
                                    {1: {author_name: 'Satish Chandra 0001', orcid: null, pid: 'c/SatishChandra'}, 2: {author_name: 'Jackie De Vries', orcid: null, pid: '56/172'}
, 3: {author_name: 'John Field', orcid: null, pid: '72/4043'}, 4: {author_name: 'Howard M. Hess', orcid: null, pid: '72/2970'}, 5: {author_name: 'Manivannan Kalidasan', orcid: nu
ll, pid: '40/1832'}, 6: {author_name: 'Raghavan Komondoor', orcid: null, pid: '97/3731'}, 7: {author_name: 'Frans Nieuwerth', orcid: null, pid: '13/6486'}, 8: {author_name: 'Gane
san Ramalingam', orcid: null, pid: 'r/GRamalingam'}, 9: {author_name: 'Justin Xue', orcid: null, pid: '72/1578'}} |  journal | 2021-10-18 |                                   Techn
ical forum: Using logical data models for understanding and transforming legacy business applications.
  c/SatishChandra |                1 | 2004 |              conf/icdcit/Chandra04 | Satish Chandra 0001 |



                                   {1: {author_name: 'Satish Chandra 0001', orcid: null, pid: 'c/SatishChandra'}} |     conf | 2021-10-18 |
                                              Tools and Techniques for Multi-site Sotware Development.
  c/SatishChandra |                1 | 2002 |              conf/icse/ChandraGP02 | Satish Chandra 0001 |


                                                                              {1: {author_name: 'Satish Chandra 0001', orcid: null, pid: 'c/SatishChandra'}, 2: {author_name: 'Pat
rice Godefroid', orcid: null, pid: '20/1639'}, 3: {author_name: 'Christopher Palm', orcid: null, pid: '73/2041'}} |     conf | 2018-11-06 |
                                        Software model checking in practice: an industrial case study.
  c/SatishChandra |                1 | 1999 |              conf/paste/ChandraR99 | Satish Chandra 0001 |


                                                                                                                                                     {1: {author_name: 'Satish Chand
ra 0001', orcid: null, pid: 'c/SatishChandra'}, 2: {author_name: 'Thomas W. Reps', orcid: null, pid: 'r/TWReps'}} |     conf | 2018-11-06 |
                                                                     Physical Type Checking for C.
  c/SatishChandra |                1 | 1999 |        journals/sigsoft/ChandraR99 | Satish Chandra 0001 |


                                                                                                                                                     {1: {author_name: 'Satish Chand
ra 0001', orcid: null, pid: 'c/SatishChandra'}, 2: {author_name: 'Thomas W. Reps', orcid: null, pid: 'r/TWReps'}} |  journal | 2020-09-17 |
```

```
cqlsh:localdb> select * from pubs_by_pid_cat_year_index;

 pid             | category | year | authorentryindex | paper_key                      | author_name         | authors



    | mdate      | title
-----------------+----------+------+------------------+--------------------------------+---------------------+--------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----+------------+------------------------------------------------------------------------------------------------------------
        145/9981 |     conf | 2022 |                2 |      conf/eccv/SiddiquiTMSND22 |        Justus Thies |
                                                                                                                                                                          {1: {author_name: 'Yaw
ar Siddiqui', orcid: null, pid: '213/7787'}, 2: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 3: {author_name: 'Fangchang Ma', orcid: null, pid: '143/0493'}, 4: {a
uthor_name: 'Qi Shan', orcid: null, pid: '14/1682'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 6: {author_name: 'Angela Dai', orcid: null, pid: '149/1202
'}} | 2022-11-15 |                                                   Texturify: Generating Textures on 3D Shape Surfaces.
        145/9981 |     conf | 2022 |                3 |         conf/eccv/ZielonkaBT22 |        Justus Thies |

                                                                                                                                                                                {1: {author_nam
e: 'Wojciech Zielonka', orcid: null, pid: '84/6152'}, 2: {author_name: 'Timo Bolkart', orcid: null, pid: '119/4795'}, 3: {author_name: 'Justus Thies', orcid: null, pid: '145/9981
'}} | 2022-11-10 |                                                       Towards Metrical Reconstruction of Human Faces.
        145/9981 |     conf | 2022 |                5 |       conf/cvpr/AzinovicMGNT22 |        Justus Thies |

                          {1: {author_name: 'Dejan Azinovic', orcid: null, pid: '197/9522'}, 2: {author_name: 'Ricardo Martin-Brualla', orcid: null, pid: '16/7968'}, 3: {author_nam
e: 'Dan B. Goldman', orcid: null, pid: '28/5892'}, 4: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 5: {author_name: 'Justus Thies', orcid: null, pid: '145/9981
'}} | 2022-10-05 |                                                                  Neural RGB-D Surface Reconstruction.
        145/9981 |     conf | 2022 |                6 |       conf/cvpr/GrassalPLRNT22 |        Justus Thies |
                                                                                                                                  {1: {author_name: 'Philip-William Grassal', orcid:
null, pid: '223/4312'}, 2: {author_name: 'Malte Prinzler', orcid: null, pid: '308/0688'}, 3: {author_name: 'Titus Leistner', orcid: null, pid: '249/2899'}, 4: {author_name: 'Cars
ten Rother', orcid: null, pid: 'r/CarstenRother'}, 5: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 6: {author_name: 'Justus Thies', orcid: null, pid: '145/9981
'}} | 2022-10-05 |                                                    Neural Head Avatars from Monocular RGB Videos.
        145/9981 |     conf | 2022 |                7 |          conf/cvpr/YiHTKH0TB22 |        Justus Thies |                                                                              {
1: {author_name: 'Hongwei Yi', orcid: null, pid: '250/4400'}, 2: {author_name: 'Chun-Hao P. Huang', orcid: null, pid: '289/7223'}, 3: {author_name: 'Dimitrios Tzionas', orcid: nu
ll, pid: '134/3198'}, 4: {author_name: 'Muhammed Kocabas', orcid: null, pid: '223/4068'}, 5: {author_name: 'Mohamed Hassan', orcid: null, pid: '65/4298'}, 6: {author_name: 'Siyu
Tang 0001', orcid: null, pid: '22/845-1'}, 7: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 8: {author_name: 'Michael J. Black', orcid: null, pid: 'b/MichaelJBlack
'}} | 2022-10-04 |                                      Human-Aware Object Placement for Visual Environment Reconstruction.
        145/9981 |     conf | 2021 |                2 |      conf/cvpr/CozzolinoTRNV21 |        Justus Thies |

                               {1: {author_name: 'Davide Cozzolino', orcid: null, pid: '120/2275'}, 2: {author_name: 'Justus Thies', orcid: null, pid: '145/9981'}, 3: {author_name:
 'Andreas Rössler', orcid: null, pid: '217/2041'}, 4: {author_name: 'Matthias Nießner', orcid: null, pid: '84/8221'}, 5: {author_name: 'Luisa Verdoliva', orcid: null, pid: '62/92
```

Cloud databases

1. The initial run of the DAG was to update the initial data onto both the databases

```
 timestamp  | num_new_pubs | num_unique_pubs
------------+--------------+-----------------
 21/11/2022 |        33536 |       (33536,)

(1 rows)
token@cqlsh:astradb>
```

2. Second triggered run of the DAG updates the num_new_pubs and the entry as timestamp is the primary key, only one entry will be added per date @weekly

```
token@cqlsh:astradb> select * from volume_update;

 timestamp  | num_new_pubs | num_unique_pubs
------------+--------------+-----------------
 21/11/2022 |            0 |       (33536,)

(1 rows)
token@cqlsh:astradb>
```

```
token@cqlsh:astradb> select * from author_pub_update;

 timestamp  | paper_key                             | authors
                                                                   | ee
                          | title
------------+---------------------------------------+------------------------------------------
-------------------------------------------------------------------+------------------------------
--------------------------+---------------------------------------------------------------------
 21/11/2022 |                  books/aw/BernsteinHG87 |                                                              {1: {auth
or_name: 'Philip A. Bernstein', orcid: null, pid: 'b/PhilipABernstein'}} |                {'http://research.microsoft.com/en-us/pe
ople/philbe/ccontrol.aspx'} |                        Concurrency Control and Recovery in Database Systems.
 21/11/2022 |                books/crc/14/McGregorH14 |
 {1: {author_name: 'Alon Y. Halevy', orcid: null, pid: 'h/AlonYHalevy'}} |
                      null |                         Opportunities and Challenges in Data Journalism.
                                                                                                                           {1:
 21/11/2022 |                books/crc/chb/Franklin14 |
 {author_name: 'Michael J. Franklin', orcid: null, pid: 'f/MJFranklin'}} |
                      null |                                                Concurrency Control and Recovery.
 21/11/2022 |                books/crc/chb/SongASOC14 |
            {1: {author_name: 'Yang Song', orcid: null, pid: '24/4470'}} |
                      null |                                           Virtualization of Storage and Systems.
 21/11/2022 |                books/crc/dong13/FengD13 |
        {1: {author_name: 'Mengling Feng', orcid: null, pid: '31/7025'}} |
                      null |                                     Incremental Maintenance of Emerging Patterns.
 21/11/2022 |             books/crc/linked14/BonczEP14 |
 {1: {author_name: 'Peter A. Boncz', orcid: null, pid: 'b/PeterABoncz'}} |                             {'http://www.crcnetbase.com/
doi/abs/10.1201/b16859-13'} |                         Experiences with Virtuoso Cluster RDF Column Store.
 21/11/2022 |            books/crc/tucker97/Franklin97 |                                                                   {1:
 {author_name: 'Michael J. Franklin', orcid: null, pid: 'f/MJFranklin'}} |
                      null |                                                Concurrency Control and Recovery.
 21/11/2022 |                  books/daglib/0019294 |                                                          {1: {author_n
```

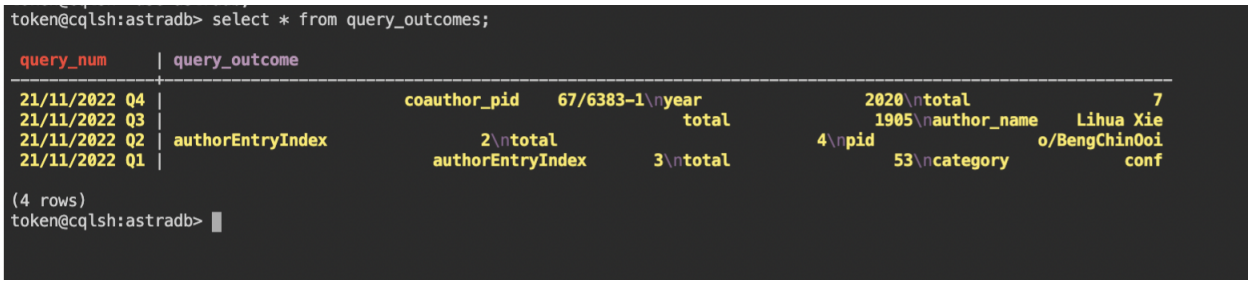
```
token@cqlsh:astradb> select * from query_outcomes;

 query_num | query_outcome
-----------+-----------------------------------------------------------------------------------------
        Q2 | authorEntryIndex          2\ntotal                  4\npid            o/BengChinOoi
        Q4 |                    coauthor_pid     67/6383-1\nyear          2020\ntotal              7
        Q1 |                      authorEntryIndex        3\ntotal          53\ncategory         conf
        Q3 |                                              total           1905\nauthor_name   Lihua Xie

(4 rows)
token@cqlsh:astradb>
```

4. Proxy query outcomes stored in cloud tables

Query outcomes stored in Astra DB database

1. Query_outcomes when manually run without the ETL pipeline

2. Query_outcomes from the data_pipeline will have timestamp appended to query_num



Queries will be updated on a weekly basis via the pipeline.

5. Rationales considered for my design of the pipeline

Extraction, Transformation and Loading were performed on one task after successful local and cloud connections due to the reasons below:

For the pipeline as **incremental extraction** was utilised:

- **Due to DBLP export person page API** returned data per author.
- No time wasted on compilation of the data into a total publication XML **increased speed of pipeline**

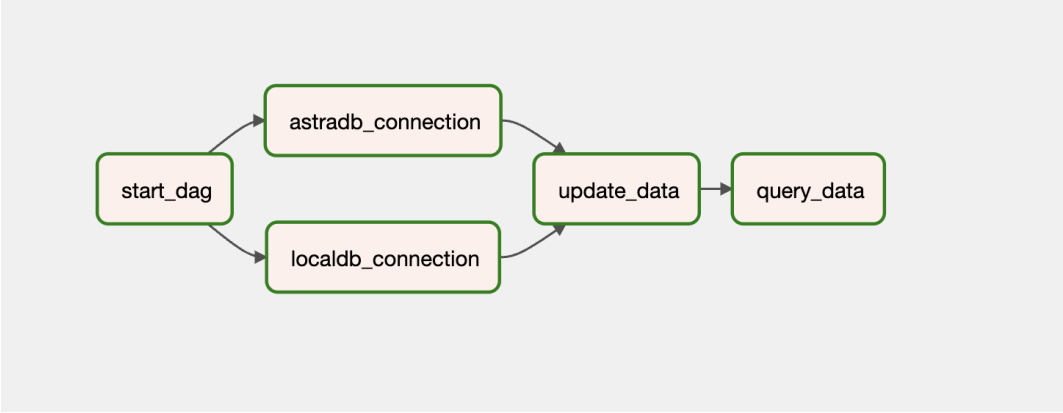
Thus, Transformation and Loading of data occurred per author after extraction from the DBLP API. Data is transformed from XML to suitable formats to support loading into local and astra cassandra databases.

As writes to AstraDB is time consuming, the data transformation process will check if the data is required to be transformed and loaded rather than loading and filtering to **increase pipeline speeds.**

The Cluster for Local DB should be specified to **ensure reliability and consistency of logging**.

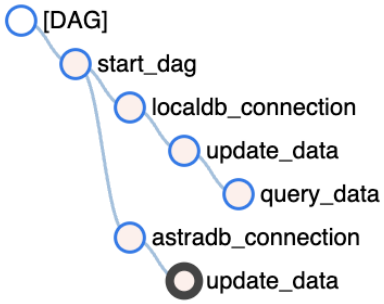
Throughput is limited by the incremental extraction having to occur sequentially.

6. Graph Visualisation

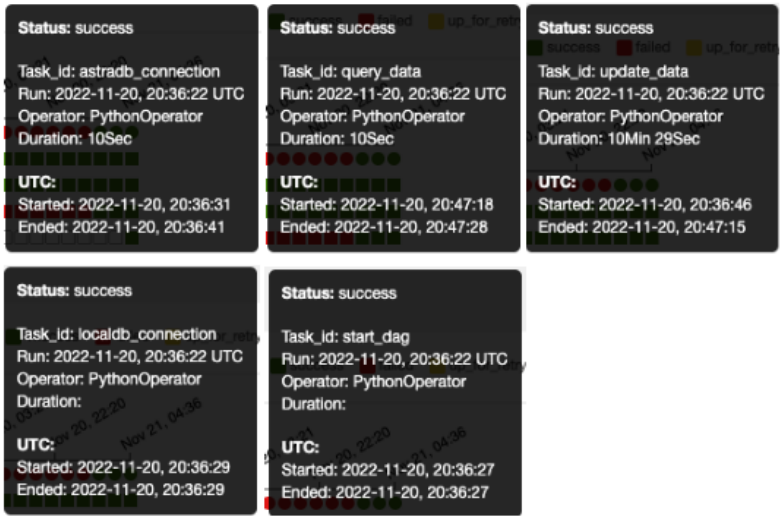


- Connection for both cloud and local databases will occur in parallel
- Data extraction, transformation and loading will occur in the update_data
- After the initial population of the database, data will only be pulled @weekly thus an Incremental ETL is preferred.
- Having the ETL process in smaller files increases the reliability of the pipeline, as one mistake in a big file could ruin the pipeline.

7. Tree view of pipeline after triggering DAG



8. Runtime of each step in pipeline



Start_db: 0s, localdb_connection: 0s, astradb_connection: 0s, update_data: 0s, query_data: 0s