CSCI 620

Assignment 7

Abhishek Shah, as5553

Question 1

```
create table Popular_Movie_Actor as
(select ta.actor, ta.title
from title_actor as ta
join title as t on
t.id = ta.title
where t.type = 'movie' and t.avgrating > 5)

-- 129 msec
```

Question 2

```
create table L1 as(
select actor, count(*)
from Popular_Movie_Actor
group by actor
having count(*) >= 5
);

-- 208 msec
```

Question 3

```
create table L2 as
select actor1, actor2, count
from
(select actor1, actor2, count(*) as count
from
(select combi2.title, actor1, actor2
from
(select combi1.title as title , actor1, actor2
from
(select pma1.title, pma1.actor as actor1, pma2.actor as actor2
from Popular_Movie_Actor as pma1
join Popular_Movie_Actor as pma2
```

```
on pma1.title = pma2.title) as combi1
where combi1.actor1 <> combi1.actor2) as combi2
where exists(select actor1 from L1 where L1.actor1 = combi2.actor1)
and exists(select actor1 from L1 where L1.actor1 = combi2.actor2)) as combi3
group by  actor1, actor2) as combi4
where count >= 5;

-- 381 msec
```

Question 4

```
create table L3 as
select actor1, actor2, actor3, count
from
(select actor1, actor2, actor3, count(*) as count
from
(select combi2.title, actor1, actor2, actor3
from
(select combi1.title as title , actor1, actor2, pma3.actor as actor3
from
(select pma1.title, pma1.actor as actor1, pma2.actor as actor2
from Popular_Movie_Actor as pma1
join Popular_Movie_Actor as pma2
on pma1.title = pma2.title) as combi1
join Popular_Movie_Actor as pma3
on pma3.title = combi1.title
where actor1 <> actor2 and actor2 < pma3.actor) as combi2
where exists(select actor1, actor2 from L2 where L2.actor1 = combi2.actor1 and L2.actor2 =
combi2.actor2)
and exists(select actor1, actor2 from L2 where L2.actor1 = combi2.actor1 and L2.actor2 =
combi2.actor3)
and exists(select actor1, actor2 from L2 where L2.actor1 = combi2.actor2 and L2.actor2 =
combi2.actor3)) as combi3
group by  actor1, actor2, actor3) as combi4
where count >= 5;

-- 266 msec
```

Question 5

```
import psycopg2


def connection(host, database, user, password, port):
    try:
```

```
        connection = psycopg2.connect(
            host=host,
            database=database,
            user=user,
            password=password,
            port=port)
        if connection:
            print("Connection Successful!")
            return connection
    except Exception as err:
        print(err)


def aprioriAlgorithm(connection):
    cursor = connection.cursor()
    minSupport = 5
    level = 2
    previousLevel = 1
    flag = True

    while flag:
        currentLevelActors = []
        for i in range(2, level+1):
            if i == level:
                while i > 0:
                    currentLevelActors.append("actor" + str(i))
                    i = i - 1
            else:
                currentLevelActors.clear()
                currentLevelActors.append("actor" + str(i))

        actorsLevel = ', '.join(str(e) for e in currentLevelActors)
        # print(actorsLevel)
        currentLevelActors.pop(0)
        actorsPreviousLevel = ', '.join(str(e) for e in currentLevelActors)

        if level == 2:
            level2Query = f"""select pma1.title, pma1.actor as actor1, pma2.actor as actor2
                        from Popular_Movie_Actor as pma1
                        join Popular_Movie_Actor as pma2
                        on pma1.title = pma2.title) as combi1"""

            level2Duplicates = """where combi1.actor1 <> combi1.actor2) as combi2
                        where exists(select actor1 from L1 where L1.actor1 = combi2.actor1)
```

```
                and exists(select actor1 from L1 where L1.actor1 = combi2.actor2)) as
combi3 """
    else:
        level2Query = ""
        level2Duplicates = ""

    if level > 2:
        level2OrMoreJoinQuery = """select pma1.title, pma1.actor as actor1, pma2.actor as
actor2
                from Popular_Movie_Actor as pma1
                join Popular_Movie_Actor as pma2
                on pma1.title = pma2.title) as combi1
                join Popular_Movie_Actor as pma3
                on pma3.title = combi1.title
                where actor1 <> actor2 and actor2 < pma3.actor) as combi2"""

        level2OrMoreDuplicates = """where combi1.actor1 <> combi1.actor2) as combi2
                        where exists(select actor1 from L1 where L1.actor1 = combi2.actor1)
                        and exists(select actor1 from L1 where L1.actor1 = combi2.actor2)) as
combi3"""
    else:
        level2OrMoreJoinQuery = ""
        level2OrMoreDuplicates = ""

    createTableQuery = f""" create table L{level} as
    select {actorsLevel}, count
    from
    (select {actorsLevel}, count(*) as count
    from
    (select combi2.title, {actorsLevel}
    from
    (select combi1.title as title, {actorsLevel}
    from
    ({level2Query}{level2OrMoreJoinQuery}
    {level2Duplicates}{level2OrMoreDuplicates}
    group by {actorsLevel}) as combi4
    where count >= {minSupport}
    """

    cursor.execute(createTableQuery)
    # print the current level
    currentLevel = f"L{level}"
    # if rows is empty, that is end of the lattice
    if cursor.rowcount == 0:
        flag = False
```

```
        else:
            previousLevel = level
            level = level + 1


if __name__ == "__main__":
    db_connection = connection('localhost', 'imdb6202',
                               'postgres', password='Abhishek@123', port=5432)

    aprioriAlgorithm(db_connection)
```