

HOMEWORK 7:

- Anirudh Narayanan(an9425)

README:

Run Queries for Questions 1-4.

Run Python Program for question 5 to create all levels of lattice.

Then run Sql queries in the end of the pdf to replace member id with member name and show the result.

Queries from the next page.

Q1.

```
-- Create Popular_Movie_Actors
Create table Popular_Movie_Actors
  as select actor, title from title_actor
JOIN title t on title_actor.title = t.tid
WHERE t.averagerating > 5 and t.titletype = 'movie';
```

O/P:

```
[2023-04-05 17:04:07] 530,106 rows affected in 4 s 61 ms
```

Q2.

```
-- Create table L1
Create table L1 as
Select * from (select pma1.actor as actor1, count(pma1.title) as movies from
popular_movie_actors pma1
              group by actor1) as t
              where movies > 4;
```

Table:

	actor ↕	movies ↕
1	790	6
2	788415	7
3	991810	8
4	23851	14
5	655799	24
6	2525754	6
7	1552	14
8	352914	13
9	2463412	14
10	417314	68
11	353717	25
12	1595	27
13	4584689	10
14	2655472	9
15	685345	15
16	1066382	11
17	898192	11
18	145284	9
19	482611	6
20	757104	19

Q3.

```
-- Create table L2
Create table L2 as
Select * from (select pma1.actor as actor1, pma2.actor as actor2,
count(pma2.title) as movies from popular_movie_actors pma1
  join popular_movie_actors pma2 on pma1.title = pma2.title
 where pma1.actor < pma2.actor
    and exists (select actor1 from L1)
 group by ( actor1, actor2)) as t
  where movies > 4;
```

Table:

	actor1 ↕	actor2 ↕	movies ↕
1	154118	154146	13
2	561	416499	7
3	2157	1365905	6
4	151540	373571	6
5	7123	1388202	10
6	35067	463953	6
7	329730	419688	12
8	7123	39596	7
9	660233	705880	6
10	461677	626314	6
11	4417	893147	17
12	92482	865396	6
13	357491	612876	9
14	415549	893449	11
15	707901	1173060	19
16	405859	827409	6
17	48040	957372	6
18	252843	406545	8
19	314775	2846621	12

Q4.

```
-- Create table L3
Create table L3 as
Select * from
(select pma.actor as actor1, pma2.actor as actor2 ,pma3.actor as actor3,
count(pma3.title) as movies from popular_movie_actors pma
      join popular_movie_actors pma2 on pma.title = pma2.title
      join popular_movie_actors pma3 on pma.title = pma3.title
      where pma.actor < pma2.actor and
            pma.actor < pma3.actor and
            pma2.actor < pma3.actor
      and exists (select l2.actor1, l2.actor2 from l2 )
```

```
group by (actor1, actor2, actor3)) as t
where movies > 4;
```

Table:

	actor1	actor2	actor3	movies
1	50	555597	555617	14
2	78	181003	855579	6
3	491	1459	5380	5
4	559	638	1150	6
5	559	638	1420	6
6	559	1150	1420	6
7	561	1000	909145	6
8	638	1150	1420	6
9	810	1359	1457663	5
10	1620	20942	1405361	11
11	1651	115558	388615	5
12	1678	103703	1022326	6
13	1678	222596	1022326	9
14	1678	371025	1022326	14
15	1678	428160	1022326	6
16	1678	634226	684793	6
17	1678	936728	1022326	5
18	1889	197715	931054	5
19	1889	416228	931054	12
20	2935	4310	220388	7
21	3706	11389	179131	6
22	3706	369058	427659	8

Q5.

```

/Users/anirudhramesh/Desktop/INTRO_TO_BIGDATA/ASSN4/venv/bin/python /Users/anirudhramesh/Desktop/INTRO_TO_BIGDATA/
Connection Successful
Created L1
Found 19854 itemsets of actors for Level 1.
Created L2
Found 2876 itemsets of actors for Level 2.
Created L3
Found 279 itemsets of actors for Level 3.
Created L4
Found 50 itemsets of actors for Level 4.
Created L5
Found 10 itemsets of actors for Level 5.
Created L6
Found 1 itemsets of actors for Level 6.
Created L7
Found 0 itemsets of actors for Level 7.
Completed check for all Lattice Levels. Level 7 returns an empty table.

```

Found 1 item set in level 6 and none in level7.

	actor1	actor2	actor3	actor4	actor5	actor6	movies
1	7158704	7158706	7158707	7158708	7158709	7158710	5

Run the Following sql queries to get the name of the actors in L6 from the members table.

```

--Q5
-- Change data types
alter table l6
    alter column actor1
type varchar using(actor1::varchar);
alter table l6
    alter column actor2
type varchar using(actor2::varchar);
alter table l6
    alter column actor3
type varchar using(actor3::varchar);
alter table l6
    alter column actor4
type varchar using(actor4::varchar);
alter table l6
    alter column actor5
type varchar using(actor5::varchar);
alter table l6
    alter column actor6
type varchar using(actor6::varchar);

update l6 set actor1 = nm.primaryname
from l6 l
join name_basics nm on (cast(l.actor1 as int)=nm.nid);
update l6 set actor2 = nm.primaryname

```

```

from l6 l
join name_basics nm on (cast(l.actor2 as int)=nm.nid);
update l6 set actor3 = nm.primaryname
from l6 l
join name_basics nm on (cast(l.actor3 as int)=nm.nid);
update l6 set actor4= nm.primaryname
from l6 l
join name_basics nm on (cast(l.actor4 as int)=nm.nid);
update l6 set actor5 = nm.primaryname
from l6 l
join name_basics nm on (cast(l.actor5 as int)=nm.nid);
update l6 set actor6 = nm.primaryname
from l6 l
join name_basics nm on (cast(l.actor6 as int)=nm.nid);

select * from l6;

```

O/P:

	actor1	actor2	actor3	actor4	actor5	actor6	movies
1	Vikram Barn	Simon Minter	Josh Bradley	Ethan Payne	Tobi Brown	Harry Lewis	5