# Question 4

## Expanded Query 3.1

### Materialized View

SELECT A.id, A.name, A.birthyear, A.deathyear --, COUNT(M.id), MIN(M.year), MAX(M.year)  
FROM (SELECT \*  
 FROM comedyactor\_mat  
 UNION  
 SELECT \*  
 FROM NonComedyActor\_mat) AS A  
JOIN (SELECT \*  
 FROM ActedIn\_mat) AS MA  
ON A.id = MA.actor  
JOIN (SELECT id, title, year, 'Comedy' AS genre  
 FROM comedymovie\_mat  
 UNION  
 SELECT id, title, year, 'NonComedy' AS genre  
 FROM noncomedymovie\_mat) AS M  
ON M.id=MA.movie  
WHERE (M.year BETWEEN 2000 AND 2005) AND  
 A.deathyear IS NULL  
GROUP BY A.id,A.name, A.deathyear, A.birthyear  
HAVING *COUNT*(M.id) > 10  
ORDER BY *COUNT*(M.id);

* Time to execute: 50 s 861 ms

### Non-Materialized View

SELECT A.id, A.name, A.birthyear, A.deathyear

FROM ComedyActor  
 UNION  
 SELECT \*  
 FROM NonComedyActor) AS A  
JOIN (SELECT \*  
 FROM ActedIn) AS MA  
ON A.id = MA.actor  
JOIN (SELECT id, title, year, 'Comedy' AS genre  
 FROM comedymovie  
 UNION  
 SELECT id, title, year, 'NonComedy' AS genre  
 FROM noncomedymovie) AS M  
ON M.id=MA.movie  
WHERE (M.year BETWEEN 2000 AND 2005) AND  
 A.deathyear IS NULL  
GROUP BY A.id,A.name, A.deathyear, A.birthyear  
HAVING *COUNT*(M.id) > 10  
ORDER BY *COUNT*(M.id);

* Time to execute: 42 m 10 s 612 ms

## Expanded Query 3.2

### Materialized View

SELECT A.id, A.name  
FROM (SELECT \*  
 FROM comedyactor\_mat  
 UNION  
 SELECT \*  
 FROM noncomedyactor\_mat) AS A  
JOIN (SELECT \*  
 FROM actedin\_mat) AS MA  
ON A.id = MA.actor  
JOIN (SELECT id, title, year, 'Comedy' AS genre  
 FROM comedymovie\_mat  
 UNION  
 SELECT id, title, year, 'NonComedy' AS genre  
 FROM noncomedymovie\_mat) AS M  
ON M.id=MA.movie  
WHERE (M.genre != 'Comedy') AND  
 (A.name LIKE ('Ja%'))  
GROUP BY A.id, A.name;

* Time to execute: 32 s 652 ms

### Non-Materialized View

SELECT A.id, A.name  
FROM (SELECT \*  
 FROM ComedyActor  
 UNION  
 SELECT \*  
 FROM NonComedyActor) AS A  
JOIN (SELECT \*  
 FROM ActedIn) AS MA  
ON A.id = MA.actor  
JOIN (SELECT id, title, year, 'Comedy' AS genre  
 FROM comedymovie  
 UNION  
 SELECT id, title, year, 'NonComedy' AS genre  
 FROM noncomedymovie) AS M  
ON M.id=MA.movie  
WHERE (M.genre != 'Comedy') AND  
 (A.name LIKE ('Ja%'))  
GROUP BY A.id, A.name;

* Time to execute: 5 m 41 s 505 ms

# Question 5

* No unnecessary joins/sources found. Query performance might be made faster if queries source data directly from tables instead of the sources defined in Q1.   
    
  Time for execution for materialized vs non-material sources approximately the same as noticed in previous part.