1. Find the movies where both Tom Cruise and Pen… C… have starred together.

```
SELECT movie
FROM StarredIn
WHERE celeb='Tom Cruise'
  INTERSECT
SELECT movie
FROM StarredIn
WHERE celeb LIKE 'Pen% C%';
```

2. Find all the co-stars of Nicolas Cage.

```
SELECT Y.celeb
FROM StarredIn X, StarredIn Y
WHERE X.celeb='Nicolas Cage' AND Y.celeb<>'Nicolas Cage' AND X.movie=Y.movie;
```

3. Find the movies where Tom Cruise co-starred with a celeb he is (or has been) in relationship with. The result should be (costar, movie) pairs.

```
SELECT Y.celeb AS costar, X.movie
FROM StarredIn X, StarredIn Y, Relationships R
WHERE X.celeb='Tom Cruise' AND X.movie=Y.movie AND
     R.celeb1=X.celeb AND R.celeb2=Y.celeb;
```

4. Find the movies where a celeb co-starred with another celeb he/she is (or has been) in relationship with. The result should be (celeb1 celeb2 movie) triples.

```
SELECT X.celeb AS celeb1, Y.celeb AS celeb2, X.movie
FROM StarredIn X, StarredIn Y, Relationships R
WHERE X.movie=Y.movie AND X.celeb=R.celeb1 AND Y.celeb=R.celeb2;
```

5. Find how many movies each celeb has starred in. Order the results by the number of movies (in descending order). Show only the celebs who have starred in at least 10 movies.

```
SELECT celeb, COUNT(movie) AS moviecount
FROM StarredIn
GROUP BY celeb
HAVING COUNT(movie)>=10
ORDER BY moviecount DESC;
```

6. Find the celebs that have been in relationship with the same celeb. The result should be (celeb1, celeb2, celeb3) triples, meaning that celeb1 and celeb2 have been in relationship with celeb3.

```
SELECT X.celeb1, Y.celeb1, X.celeb2
FROM Relationships X, Relationships Y
WHERE X.celeb2=Y.celeb2 AND X.celeb1<Y.celeb1;
```

7. For each pair of enemies give the number of movies each has starred in. The result should be a set of (celeb1 celeb2 n1 n2) quadruples, where n1 and n2 are the number of movies that celeb1 and celeb2 have starred in, respectively. Observe that there might be celebs with zero movies they have starred in.

```
CREATE VIEW celebMovieCounts AS
```

```
SELECT name AS celeb, COUNT(movie) AS moviecount
FROM Celebs LEFT OUTER JOIN StarredIn ON name=celeb
GROUP BY name;

SELECT E.celeb1, E.celeb2, X.moviecount, Y.moviecount
FROM Enemies E, celebMovieCounts X, celebMovieCounts Y
WHERE E.celeb1=X.celeb AND E.celeb2=Y.celeb;

DROP VIEW celebMovieCounts;
```

8. Find how many albums each celeb has released. Order the results by the number of albums (in descending order). Show only the celebs who have released at least 2 albums.

```
SELECT celeb, COUNT(album) AS albumcount
FROM Releases
GROUP BY celeb
HAVING COUNT(album)>=2
ORDER BY albumcount DESC;
```

9. Find those celebs that have starred in some movie and have released some album.

```
SELECT celeb
FROM StarredIn
  INTERSECT
SELECT celeb
FROM Releases;
```

10. For each celeb that has both starred in some movie and released some album give the numbers of movies and albums he/she has starred in and released, respectively. The result should be a set of (celeb, number_of_movies, number_of_albums) triples.

```
SELECT *
FROM
  (SELECT celeb, COUNT(movie)
   FROM StarredIn
   GROUP BY celeb)
      NATURAL JOIN
  (SELECT celeb, COUNT(album)
   FROM Releases
   GROUP BY celeb);
```

11. Find the earliest and the latest relationship (w.r.t the start date) recorded in this database. Hint. This needs two (similar) queries.

```
SELECT *
FROM Relationships
WHERE started=( SELECT MIN(started) FROM Relationships );

SELECT *
FROM Relationships
WHERE started=( SELECT MAX(started) FROM Relationships );
```