6.830/6.814 PS1 Solutions

1. SELECT name FROM people WHERE name LIKE "John%";

2. SELECT title, year FROM movies ORDER BY year LIMIT 5;

3. SELECT AVG(runtime) FROM movies WHERE year = 1963;

4. SELECT title FROM movies
   JOIN (SELECT movie_id, max(rating) FROM ratings) as max_rating
   ON movies.id = max_rating.movie_id;

5. SELECT title FROM movies
   JOIN (SELECT movie_id
         FROM cast_members
         JOIN (SELECT id FROM people WHERE name='Daniel Craig') as craig_id
         ON cast_members.person_id = craig_id.id) as craig_movies
   ON movies.id = craig_movies.movie_id;

6. SELECT AVG(runtime) FROM movies
   JOIN (SELECT movie_id FROM ratings WHERE rating > 9.0) as high_ratings
   ON movies.id = high_ratings.movie_id;

7. SELECT COUNT(person_id) FROM cast_members
   JOIN (SELECT id, MAX(runtime) FROM movies) as max_runtime
   ON cast_members.movie_id = max_runtime.id;

8. SELECT id, name, COUNT() AS num_movies FROM directors INNER JOIN people ON person_id = id GROUP BY id,name ORDER BY -num_movies LIMIT 1;
9.
SELECT name
FROM (SELECT MAX(total), director_id, cast_id
      FROM (SELECT COUNT(directors.movie_id) as total,
              directors.person_id as director_id,
              cast_members.person_id as cast_id
      FROM directors, cast_members
      WHERE directors.movie_id = cast_members.movie_id AND
              directors.person_id != cast_members.person_id
      GROUP BY director_id, cast_id))
JOIN people
ON director_id=people.id or cast_id=people.id;

10.
# Select movies with at least 10 members
CREATE TEMPORARY TABLE universe AS SELECT movie_id AS id, COUNT() AS cast_count FROM cast_members GROUP BY movie_id HAVING cast_count >= 10;

# Filter cast_members to only have movies with 10 members
CREATE TEMPORARY TABLE part AS SELECT movie_id, person_id FROM cast_members, universe WHERE cast_members.movie_id = universe.id;

# Find number of cast members shared between two movies (overlap)
CREATE TEMPORARY TABLE overlap_count AS SELECT a.movie_id as id1, b.movie_id as id2, COUNT() as overlap FROM part as a, part as b WHERE a.person_id = b.person_id, a.movie_id != b.movie_id GROUP BY id1, id2 HAVING overlap >= 10;

# Select movies whose overlap count == number of cast members
CREATE TEMPORARY TABLE result AS SELECT id, cast_count, id2 FROM overlap_count,universe WHERE overlap_count.id1 = universe.id AND cast_count = overlap
ORDER BY -cast_count;

SELECT id, title from result INNER JOIN movies ON id = movie_id;

11.
SELECT name
FROM people
JOIN (SELECT MAX(career_span), person_id
      FROM (SELECT
              MAX(movies.year) - MIN(movies.year) as career_span,
              person_id
      FROM cast_members
      JOIN movies
      ON movies.id = cast_members.movie_id
      GROUP BY person_id))
      ON people.id = cast_count.person_id;
GROUP BY person_id)) as max_career
ON people.id = max_career.person_id;

12. WITH bacon_id AS (SELECT id FROM people WHERE name='Kevin Bacon')
SELECT b.person_id
FROM cast_members a, cast_members b
WHERE a.movie_id=b.movie_id and a.person_id in bacon_id;

13. CREATE TEMP TABLE coact AS
SELECT DISTINCT a.person_id AS person_a, b.person_id AS person_b
FROM cast_members a INNER JOIN cast_members b
ON a.movie_id = b.movie_id;

CREATE TEMP TABLE bacon_0 AS
SELECT id FROM people WHERE name = 'Kevin Bacon';

CREATE TEMP TABLE bacon_1 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_0;

CREATE TEMP TABLE bacon_2 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_1;

CREATE TEMP TABLE bacon_3 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_2;

CREATE TEMP TABLE bacon_4 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_3;

CREATE TEMP TABLE bacon_5 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_4;

CREATE TEMP TABLE bacon_6 AS
SELECT DISTINCT person_a AS id FROM coact WHERE person_b IN bacon_5;

CREATE TEMP TABLE bacon_degree AS
SELECT id, MIN(d) AS degree FROM (  
  SELECT id, 0 AS d FROM bacon_0
UNION ALL
  SELECT id, 1 AS d FROM bacon_1
UNION ALL
  SELECT id, 2 AS d FROM bacon_2
UNION ALL
  SELECT id, 3 AS d FROM bacon_3
UNION ALL
  SELECT id, 4 AS d FROM bacon_4
UNION ALL

SELECT id, 5 AS d FROM bacon_5
UNION ALL
SELECT id, 6 AS d FROM bacon_6)
GROUP BY id;