

Section E

Attempted questions: 9

Attached question: 9

a. The `pid` column in the `ActsIn` table is a reference to the `pid` column in the `People` table. This means that there cannot exist a value of `ActsIn.pid` which cannot also be found in `People.pid`. This associates each row in `ActsIn` with exactly one row in `People`.

b. There is redundancy between the `ActsIn` table and the `HasRole` table. That is to say that if a person has a role in a movie, they also acted in it. If the relevant row of `ActsIn` were deleted or corrupted, the data therein would be entirely recoverable from the fact that there exists a row relating the person and the movie in the `HasRole` table. This means that if the database needed to be updated e.g. it was discovered that the person acting in the movie was actually a different person to whom it was thought to be, the `pid` of the relevant rows of both the `ActsIn` table *and* the `HasRole` table would need to be updated. The fact that multiple records need to be updated to reflect the change of one single piece of data leaves more room for error.

c.

```
SELECT DISTINCT
g1.genre AS genre1,
g2.genre AS genre2,
COUNT(m.mid) AS total
FROM Genres AS g1
INNER JOIN HasGenre AS h1 ON g1.gid = h1.gid
INNER JOIN Movies AS m ON h1.mid = m.mid
INNER JOIN HasGenre AS h2 ON m.mid = h2.mid
INNER JOIN Genres AS g2 ON h2.gid = g2.gid
WHERE g1.genre < g2.genre
GROUP BY g1.genre, g2.genre;
```

d.

```
SELECT p.pid FROM People
INNER JOIN ActsIn AS a1 ON a1.pid = p.pid
INNER JOIN Movies AS m1 ON a1.mid = m1.mid
INNER JOIN ActsIn AS a2 ON a2.mid = m1.mid
INNER JOIN People AS p1 ON a2.pid = p1.pid
INNER JOIN ActsIn AS a3 ON a3.pid = p1.pid
INNER JOIN Movies AS m2 ON a3.mid = m2.mid
INNER JOIN ActsIN AS a4 ON a4.mid = m2.mid
WHERE m1.mid != m2.mid
AND p.pid != p1.pid
AND p.pid != a4.pid
AND p1.pid != a4.pid
AND a4.pid = "kid";
```

e

```
CREATE VIEW BACON_NUMBER_ONE AS
SELECT p.pid AS pid FROM People AS p
INNER JOIN ActsIn AS a1 ON a1.pid = p.pid
INNER JOIN Movies AS m ON a1.mid = m.mid
INNER JOIN ActsIn AS a2 ON a2.mid = m.mid
WHERE p.pid != a2.pid
AND a2.pid = "kid";

CREATE VIEW BACON_NUMBER_TWO AS
SELECT p.pid AS pid FROM People AS p
INNER JOIN ActsIn AS a1 ON a1.pid = p.pid
INNER JOIN Movies AS m ON a1.mid = m.mid
INNER JOIN ActsIn AS a2 ON a2.mid = m.mid
INNER JOIN BACON_NUMBER_ONE AS b ON b.pid = a2.pid
WHERE p.pid != b.pid;

CREATE VIEW BACON_NUMBER AS
SELECT pid, MIN(bacon) FROM (
    SELECT "kid" AS pid, 0 AS bacon
    UNION
    SELECT b1.pid AS pid, 1 AS bacon FROM BACON_NUMBER_ONE AS b1
    UNION
    SELECT b2.pid AS pid, 2 AS bacon FROM BACON_NUMBER_TWO AS b2;
) GROUP BY pid;

SELECT pid FROM BACON_NUMBER WHERE bacon = 2;
```