

DATABASE MANAGEMENT SYSTEMS DESIGN

ASSIGNMENT 4

DAVID GOMEZ CAMARGO

PART I

1. PRINT THE BARS THAT SERVE A BEER JOHN SMITH LIKES.

```
SELECT  DISTINCT S.BAR
FROM    SERVES AS S, LIKES AS L
WHERE   L.BEER = S.BEER AND L.DRINKER = 'John Smith';
```

2. PRINT THE DRINKERS THAT FREQUENT AT LEAST ONE BAR THAT SERVE A BEER JOHN SMITH LIKES.

```
SELECT  DISTINCT F.DRINKER
FROM    FREQUENTS AS F
WHERE   F.BAR IN (
        SELECT  DISTINCT S.BAR
        FROM    SERVES AS S, LIKES AS L
        WHERE   L.BEER = S.BEER AND L.DRINKER = 'John Smith' );
```

3. PRINT THE DRINKERS THAT FREQUENT AT LEAST ONE BAR THAT SERVES A BEER THEY LIKE.

```
SELECT  DISTINCT F.DRINKER
FROM    FREQUENTS AS F, LIKES AS L, SERVES AS S
WHERE   F.DRINKER = L.DRINKER AND F.BAR = S.BAR AND L.BEER = S.BEER;
```

4. PRINT THE DRINKERS THAT FREQUENT ONLY BARS THAT SERVE SOME BEER THEY LIKE.

```
SELECT  DISTINCT F1.DRINKER
FROM    FREQUENTS AS F1
WHERE   NOT EXISTS (
        SELECT  *
        FROM    FREQUENTS AS F2
        WHERE   F2.DRINKER = F1.DRINKER AND NOT EXISTS (
                SELECT  *
                FROM    SERVES AS S, LIKES AS L
                WHERE   S.BEER = L.BEER AND F2.BAR = S.BAR AND F2.DRINKER = L.DRINKER ));
```

5. PRINT THE DRINKERS THAT FREQUENT ALL THE BARS THAT SERVE A BEER THEY LIKE.

```
SELECT  DISTINCT F1.DRINKER
FROM    FREQUENTS AS F1
WHERE   NOT EXISTS (
        SELECT  S.BAR
        FROM    SERVES AS S, LIKES AS L
        WHERE   S.BEER = L.BEER AND F1.DRINKER = L.DRINKER AND NOT EXISTS (
                SELECT  *
                FROM    FREQUENTS AS F2
                WHERE   F2.BAR = S.BAR AND F2.DRINKER = F1.DRINKER ));
```

6. PRINT THE DRINKERS THAT FREQUENT NO BAR THAT SERVES A BEER THAT THEY LIKE.

```
SELECT DISTINCT F.DRINKER
FROM FREQUENTS AS F
WHERE NOT EXISTS (
    SELECT *
    FROM SERVES AS S, LIKES AS L
    WHERE S.BEER = L.BEER AND F.BAR = S.BAR AND F.DRINKER = L.DRINKER );
```

7. PRINT THE DRINKERS THAT FREQUENT ALL THE BARS THAT SERVE A BEER JOHN SMITH LIKES.

```
SELECT DISTINCT F.DRINKER
FROM FREQUENTS AS F
WHERE NOT EXISTS (
    SELECT S.BAR
    FROM SERVES AS S, LIKES AS L
    WHERE L.BEER = S.BEER AND L.DRINKER = 'John Smith' AND NOT EXISTS (
        SELECT *
        FROM FREQUENTS AS F2
        WHERE F2.BAR = S.BAR AND F2.DRINKER = F1.DRINKER ));
```

8. PRINT PAIRS OF DRINKERS THAT FREQUENT AT LEAST ONE COMMON BAR.

```
SELECT F1.DRINKER, F2.DRINKER
FROM FREQUENTS AS F1, FREQUENTS AS F2
WHERE F1.DRINKER != F2.DRINKER AND F1.BAR = F2.BAR;
```

9. PRINT PAIRS OF DRINKERS THAT FREQUENT EXACTLY THE SAME BARS.

```
SELECT F1.DRINKER, F2.DRINKER
FROM FREQUENTS AS F1, FREQUENTS AS F2
WHERE F1.BAR = F2.BAR AND F1.DRINKER <> F2.DRINKER
GROUP BY F1.DRINKER, F2.DRINKER
HAVING COUNT (DISTINCT F1.BAR) = (
    SELECT COUNT (DISTINCT F3.BAR)
    FROM FREQUENTS AS F3
    WHERE F3.DRINKER = F1.DRINKER)
    AND COUNT (DISTINCT F1.BAR) = (
    SELECT COUNT (DISTINCT F4.BAR)
    FROM FREQUENTS AS F4
    WHERE F4.DRINKER = F2.DRINKER );
```

10. PRINT THE NUMBER OF BEER SERVED PER BAR.

```
SELECT BAR, COUNT (DISTINCT BEER)
FROM SERVES
GROUP BY BAR;
```

11. WHICH BAR IS THE MOST POPULAR?

```
SELECT    BAR, COUNT (DISTINCT DRINKER)
FROM      FREQUENTS
GROUP BY  BAR
HAVING    COUNT (DISTINCT DRINKER) = (
    SELECT    MAX (DRINKER_COUNT)
    FROM      (
        SELECT    COUNT (DISTINCT DRINKER) AS DRINKER_COUNT
        FROM      FREQUENTS
        GROUP BY  BAR ));
```

PART II

1. LIST TWO STARS THAT SHARE AN ADDRESS.

```
SELECT  MS1.Name, MS2.Name
FROM    MovieStar AS MS1, MovieStar AS MS2
WHERE   MS1.Name != MS2.Name AND MS1.Address = MS2.Address;
```

2. QUERY THE NAMES AND ADDRESSES OF ALL FEMALE MOVIE STARS WHO ARE ALSO MOVIE EXECUTIVES WITH A NET WORTH OVER \$10,000,000

```
SELECT  MS.Name, MS.Adress
FROM    MovieStar AS MS, MovieExec AS ME
WHERE   MS.Name = ME.Name AND ME.NetWorth > 10000000 AND MS.Gender = 'Female';
```

3. QUERY ALL THE TITLES AND YEARS OF MOVIES THAT APPEARED IN EITHER THE MOVIES OR STARSIN RELATIONS.

```
(SELECT  Title, Year
FROM      Movie)
UNION
(SELECT  MovieTitle, MovieYear
FROM      StarsIn);
```

4. QUERY THE PRODUCER OF STAR WARS.

```
SELECT  ProducerC
FROM      Movie
WHERE   Title = 'Star Wars';
```

5. QUERY ALL THE PRODUCERS OF MOVIES IN WHICH LEONARDO DICAPRIO STARS.

```
SELECT  DISTINCT ProducerC
FROM      Movie AS M, StarsIn AS SI
```

WHERE M.Title = Sl.MovieTitle **AND** Sl.StarName = 'Leonardo DiCaprio';

6. QUERY THE TITLES THAT HAVE BEEN USED FOR TWO OR MORE MOVIES.

```
SELECT      Title
FROM        Movie
GROUP BY    Title
HAVING      COUNT (Title) >= 2;
```

7. FIND SUM OF THE MOVIES LENGTH EACH STUDIO IS PRODUCED.

```
SELECT      StudioName, SUM (Length)
FROM        Movie
GROUP BY    StudioName;
```

8. CREATE A LIST OF EACH PRODUCER NAME AND THE TOTAL LENGTH OF FILM PRODUCED.

```
SELECT      ProducerC, SUM (Length)
FROM        Movie
GROUP BY    ProducerC;
```

9. QUERY THE TOTAL FILM LENGTH FOR ONLY THOSE PRODUCERS WHO MADE AT LEAST ONE FILM PRIOR TO 1930.

```
SELECT      ProducerC, SUM (Length)
FROM        Movie
GROUP BY    ProducerC IN (
HAVING      COUNT (Year) >= 1 AND Year < 1930;
```