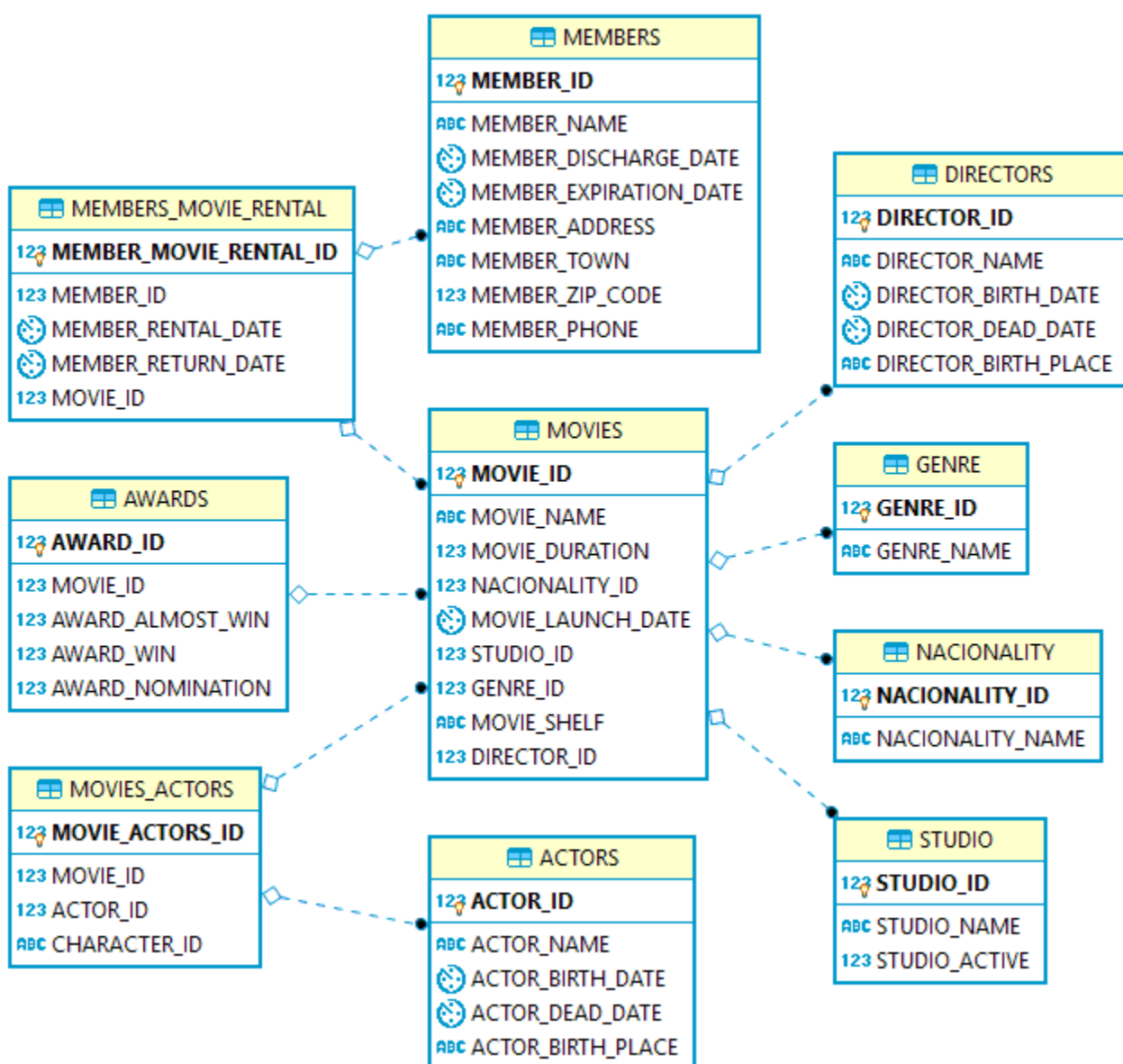




PM **Pablo Martínez**



- **Devuelve todas las películas**

```
SELECT
    MOVIE_NAME
FROM
    PUBLIC.MOVIES
```

- **Devuelve todos los géneros existentes**

```
SELECT
    GENRE_NAME
FROM PUBLIC.GENRE
```

- **Devuelve el nombre de todas las películas y el nombre del estudio que las ha realizado**

```
SELECT
    M.MOVIE_NAME,
    S.STUDIO_NAME
FROM PUBLIC.MOVIES M
INNER JOIN PUBLIC.STUDIO S ON
    M.STUDIO_ID = S.STUDIO_ID
```

- **Devuelve el nombre y la edad de todos los directores menores o iguales de 50 años que estén vivos**

```
SELECT
    DIRECTOR_NAME,
    DATEDIFF(YEAR, DIRECTOR_BIRTH_DATE, TODAY()) AS "AGE"
FROM PUBLIC.DIRECTORS
WHERE
    DATEDIFF(YEAR, DIRECTOR_BIRTH_DATE, TODAY()) <= 50
    AND DIRECTOR_DEAD_DATE IS NULL
```

- **Devuelve el nombre y la edad de todos los actores menores de 50 años que hayan fallecido**

La siguiente query no es correcta, ya que no tiene en cuenta la edad que tenían cuando murieron si no la edad que tendrían el día de hoy:

```
SELECT
    ACTOR_NAME,
    DATEDIFF(YEAR, ACTOR_BIRTH_DATE, TODAY()) AS "AGE"
FROM PUBLIC.ACTORS
WHERE
    DATEDIFF(YEAR, ACTOR_BIRTH_DATE, TODAY()) < 50
    AND ACTOR_DEAD_DATE IS NOT NULL
```

Para tener en cuenta la edad que tenían en el momento de su muerte, la query sería la siguiente:

```
SELECT
    ACTOR_NAME,
    DATEDIFF(YEAR, ACTOR_BIRTH_DATE, ACTOR_DEAD_DATE) AS
"AGE"
FROM PUBLIC.ACTORS
WHERE
    DATEDIFF(YEAR, ACTOR_BIRTH_DATE, ACTOR_DEAD_DATE) < 50
    AND ACTOR_DEAD_DATE IS NOT NULL;
```

- **Devuelve el nombre de todos los directores menores o iguales de 40 años que estén vivos**

```
SELECT
    DIRECTOR_NAME,
    DATEDIFF(YEAR, DIRECTOR_BIRTH_DATE, TODAY()) AS "AGE"
FROM PUBLIC.DIRECTORS
WHERE
    DATEDIFF(YEAR, DIRECTOR_BIRTH_DATE, TODAY()) <= 40
    AND DIRECTOR_DEAD_DATE IS NULL
```

- **Devuelve la lista de todos los estudios de grabación que estén activos**

```
SELECT
    STUDIO_NAME
FROM PUBLIC.STUDIO
WHERE
    STUDIO_ACTIVE = TRUE
```

- **Devuelve el nombre y el año de todas las películas que han sido producidas por un estudio que actualmente no esté activo**

```
SELECT
    M.MOVIE_NAME,
    YEAR(M.MOVIE_LAUNCH_DATE) AS YEAR
FROM PUBLIC.STUDIO S
INNER JOIN PUBLIC.MOVIES M ON
    S.STUDIO_ID = M.STUDIO_ID
WHERE
    STUDIO_ACTIVE = FALSE
```

- **Devuelve una lista de los 20 últimos miembros en anotarse al videoclub**

```
SELECT
    MEMBER_NAME
FROM PUBLIC.MEMBERS
ORDER BY
    MEMBER_DISCHARGE_DATE DESC LIMIT 20
```

- **Devuelve una lista de las últimas 10 películas que se han alquilado**

```
SELECT
    M.MOVIE_NAME
FROM
    PUBLIC.MEMBERS_MOVIE_RENTAL MMR
INNER JOIN PUBLIC.MOVIES M ON
    MMR.MOVIE_ID = M.MOVIE_ID
ORDER BY
    MEMBER_RENTAL_DATE DESC
LIMIT 10
```

- **Indica cual es el nombre del director del que más películas se han alquilado**

La siguiente query no está bien, ya que no tiene en cuenta que un mismo director puede aparecer varias veces en la subquery y por tanto habría que sumar sus películas:

```
SELECT
    GROUP_CONCAT(DIRECTOR_NAME) AS DIRECTOR_NAME
FROM
    (
```

```
SELECT COUNT(MOVIE_ID) AS NUM_MOV,  
       D.DIRECTOR_NAME  
FROM PUBLIC.MEMBERS_MOVIE_RENTAL MR  
INNER JOIN PUBLIC.MOVIES M ON  
       M.MOVIE_ID = MR.MOVIE_ID  
INNER JOIN PUBLIC.DIRECTORS D ON  
       D.DIRECTOR_ID = M.DIRECTOR_ID  
GROUP BY  
       M.MOVIE_ID,  
       M.DIRECTOR_ID,  
       D.DIRECTOR_NAME  
ORDER BY  
       NUM_MOV DESC)  
GROUP BY  
       NUM_MOV  
LIMIT 1
```

La siguiente query si sería correcta:

```
SELECT DIRECTOR_NAME, COUNT(MR.MEMBER_MOVIE_RENTAL_ID)  
TOTAL_ALQUILADAS  
FROM DIRECTORS D  
JOIN MOVIES M ON D.DIRECTOR_ID = M.DIRECTOR_ID  
JOIN MEMBERS_MOVIE_RENTAL MR ON MR.MOVIE_ID = M.MOVIE_ID  
GROUP BY D.DIRECTOR_ID  
ORDER BY TOTAL_ALQUILADAS DESC LIMIT 1;
```

- **Indica cuantos premios han ganado cada uno de los estudios con las películas que han creado**

```
SELECT SUM (A.AWARD_WIN) AS AWARDS_WIN,  
       S.STUDIO_NAME  
FROM PUBLIC.MOVIES M  
INNER JOIN PUBLIC.STUDIO S ON  
       M.STUDIO_ID = S.STUDIO_ID  
INNER JOIN PUBLIC.AWARDS A ON  
       A.MOVIE_ID = M.MOVIE_ID  
GROUP BY  
       S.STUDIO_NAME
```

- **Indica cuántas películas ha realizado cada director antes de cumplir 41 años**

```
SELECT COUNT(M.MOVIE_ID) AS NUM_MOVIES,  
       D.DIRECTOR_NAME  
FROM PUBLIC.DIRECTORS D  
INNER JOIN PUBLIC.MOVIES M ON  
       M.DIRECTOR_ID = D.DIRECTOR_ID  
WHERE  
       DATEDIFF(YEAR, D.DIRECTOR_BIRTH_DATE,  
M.MOVIE_LAUNCH_DATE) < 41  
GROUP BY  
       D.DIRECTOR_NAME
```

- **Indica la edad media de los directores vivos**

```
SELECT AVG(DATEDIFF(YEAR, DIRECTOR_BIRTH_DATE, TODAY()))  
AS AVERAGE_AGE  
FROM PUBLIC.DIRECTORS  
WHERE  
       DIRECTOR_DEAD_DATE IS NULL
```

- **Indica la edad media de los actores que han fallecido**

```
SELECT AVG (DATEDIFF(YEAR, ACTOR_BIRTH_DATE,  
ACTOR_DEAD_DATE)) AS AVERAGE_AGE  
FROM PUBLIC.ACTORS  
WHERE  
       ACTOR_DEAD_DATE IS NOT NULL
```

- **Indica cuál es el género favorito de cada uno de los directores cuando dirigen una película**

--Esta respuesta en StackOverflow explica cómo conseguirlo: <<https://stackoverflow.com/a/7745635>>

```
SELECT  
       GROUPID.DIRECTOR_NAME,  
       GROUP_CONCAT(GROUPID.GENRE_NAME) AS GENRE_NAME  
FROM
```

```
(
SELECT COUNT(G.GENRE_NAME) AS NUM_MOVIES,
      D.DIRECTOR_ID,
      D.DIRECTOR_NAME,
      G.GENRE_ID,
      G.GENRE_NAME
FROM PUBLIC.MOVIES M
INNER JOIN PUBLIC.GENRE AS G ON
      M.GENRE_ID = G.GENRE_ID
INNER JOIN PUBLIC.DIRECTORS D ON
      M.DIRECTOR_ID = D.DIRECTOR_ID
GROUP BY G.GENRE_ID,
      D.DIRECTOR_ID
ORDER BY
      D.DIRECTOR_ID) GROUPID
INNER JOIN (
SELECT
      DIRECTOR_ID,
      MAX(NUM_MOVIES) NUM_MOVIES
FROM
      (
      SELECT COUNT(G.GENRE_NAME) AS NUM_MOVIES,
            D.DIRECTOR_ID,
            D.DIRECTOR_NAME,
            G.GENRE_ID,
            G.GENRE_NAME
      FROM PUBLIC.MOVIES M
      INNER JOIN PUBLIC.GENRE AS G ON
            M.GENRE_ID = G.GENRE_ID
      INNER JOIN PUBLIC.DIRECTORS D ON
            M.DIRECTOR_ID = D.DIRECTOR_ID
      GROUP BY G.GENRE_ID,
            D.DIRECTOR_ID
      ORDER BY
            D.DIRECTOR_ID)
GROUP BY
      DIRECTOR_ID ) MAXVAL ON
GROUPID.DIRECTOR_ID = MAXVAL.DIRECTOR_ID
```

```
AND GROUPID.NUM_MOVIES = MAXVAL.NUM_MOVIES
GROUP BY
    GROUPID.DIRECTOR_NAME
```

- **Indica cuál es la nacionalidad favorita de cada uno de los estudios en la producción de las películas**

--Esta respuesta en StackOverflow explica cómo conseguirlo: <<https://stackoverflow.com/a/7745635>>

```
SELECT
    STUDIO_NAME,
    GROUP_CONCAT(NACIONALITY_NAME)
FROM
    (
        SELECT
            GROUP_ID.STUDIO_ID,
            GROUP_ID.NUM_MOVIES,
            GROUP_ID.NACIONALITY_ID,
            GROUP_ID.NACIONALITY_NAME,
            GROUP_ID.STUDIO_NAME
        FROM
            (
                SELECT COUNT(M.MOVIE_ID) AS NUM_MOVIES ,
                    M.NACIONALITY_ID ,
                    N.NACIONALITY_NAME ,
                    M.STUDIO_ID ,
                    S.STUDIO_NAME
                FROM PUBLIC.MOVIES M
                INNER JOIN PUBLIC.NACIONALITY N ON
                    M.NACIONALITY_ID = N.NACIONALITY_ID
                INNER JOIN PUBLIC.STUDIO S ON
                    S.STUDIO_ID = M.STUDIO_ID
            )
        GROUP BY
            M.NACIONALITY_ID,
            N.NACIONALITY_NAME,
            M.STUDIO_ID,
            S.STUDIO_NAME
```



```

ORDER BY
    STUDIO_ID,
    NUM_MOVIES DESC ) GROUP_ID
INNER JOIN (
    SELECT
        STUDIO_ID,
        MAX(NUM_MOVIES) NUM_MOVIES
    FROM
        (
            SELECT COUNT(M.MOVIE_ID) AS
NUM_MOVIES ,
                M.NACIONALITY_ID ,
                N.NACIONALITY_NAME ,
                M.STUDIO_ID ,
                S.STUDIO_NAME
            FROM PUBLIC.MOVIES M
            INNER JOIN PUBLIC.NACIONALITY N ON
                M.NACIONALITY_ID =
N.NACIONALITY_ID
            INNER JOIN PUBLIC.STUDIO S ON
                S.STUDIO_ID = M.STUDIO_ID
        )
    GROUP BY
        M.NACIONALITY_ID,
        N.NACIONALITY_NAME,
        M.STUDIO_ID,
        S.STUDIO_NAME
    ORDER BY
        STUDIO_ID,
        NUM_MOVIES DESC )
GROUP BY
    STUDIO_ID ) MAX_VALUE ON
GROUP_ID.STUDIO_ID = MAX_VALUE.STUDIO_ID
AND GROUP_ID.NUM_MOVIES =
MAX_VALUE.NUM_MOVIES)
GROUP BY
    STUDIO_NAME

```

- Indica cuál es la media de duración de las películas de cada director

```
SELECT
    DIRECTORS.DIRECTOR_NAME,
    AVG(MOVIE_DURATION) AS AVERAGE_MOVIE_DURATION
FROM PUBLIC.MOVIES
INNER JOIN PUBLIC.DIRECTORS ON
    MOVIES.DIRECTOR_ID = DIRECTORS.DIRECTOR_ID
GROUP BY
    DIRECTOR_NAME
ORDER BY
    DIRECTOR_NAME
```

- **Indica cuál es el nombre y la duración mínima de las películas que han sido alquiladas en los últimos 2 años por los miembros del videoclub**

```
SELECT
    GROUP_CONCAT(M.MOVIE_NAME) AS MOVIE_NAME,
    M.MOVIE_DURATION
FROM PUBLIC.MEMBERS_MOVIE_RENTAL MMR
INNER JOIN PUBLIC.MOVIES M ON
    MMR.MOVIE_ID = M.MOVIE_ID
WHERE
    DATEADD(YEAR,
        -2,
        TODAY()) < MMR.MEMBER_RENTAL_DATE
GROUP BY M.MOVIE_DURATION
ORDER BY
    M.MOVIE_DURATION ASC LIMIT 1
```

- **Indica cuál fue la primera película que alquilaron los miembros del videoclub cuyos teléfonos tengan como último dígito el ID de alguna nacionalidad**

```
SELECT
    MEMBER_NAME,
    MOVIE_NAME
FROM
    (
        SELECT
```

```
NACIONALITY_ID,
MEMBER_NAME,
MEMBER_ID,
MOVIE_ID,
MEMBER_RENTAL_DATE
FROM
(
SELECT
    NACIONALITY_ID,
    MEMBER_NAME,
    MEMBER_ID,
    MOVIE_ID,
    MEMBER_RENTAL_DATE
FROM PUBLIC.NACIONALITY N
INNER JOIN (
    SELECT
        MEMBER_NAME,
        MEMBER_ID,
        SUBSTRING(MEMBER_PHONE,
LENGTH(MEMBER_PHONE), 1) AS LAST_NUMBER
    FROM PUBLIC.MEMBERS ) M ON
    N.NACIONALITY_ID = M.LAST_NUMBER
INNER JOIN PUBLIC.MEMBERS_MOVIE_RENTAL MMR
ON
    MMR.MEMBER_ID = M.MEMBER_ID
ORDER BY
    MEMBER_ID,
    MMR.MEMBER_RENTAL_DATE ASC) a
INNER JOIN (
    SELECT
        MEMBER_ID,
        MIN(MEMBER_RENTAL_DATE) AS
MEMBER_RENTAL_DATE
FROM
    (
    SELECT
        NACIONALITY_ID,
        MEMBER_NAME,
```

```
        MEMBER_ID,  
        MOVIE_ID,  
        MEMBER_RENTAL_DATE  
FROM PUBLIC.NACIONALITY N  
INNER JOIN (  
    SELECT  
        MEMBER_NAME,  
        MEMBER_ID,  
  
SUBSTRING(MEMBER_PHONE, LENGTH(MEMBER_PHONE), 1) AS  
LAST_NUMBER  
  
        FROM PUBLIC.MEMBERS ) M ON  
N.NACIONALITY_ID =  
M.LAST_NUMBER  
  
    INNER JOIN  
PUBLIC.MEMBERS_MOVIE_RENTAL MMR ON  
        MMR.MEMBER_ID = M.MEMBER_ID  
ORDER BY  
        MEMBER_ID,  
        MMR.MEMBER_RENTAL_DATE ASC)  
GROUP BY  
        MEMBER_ID ) b ON  
a.MEMBER_ID = b.MEMBER_ID  
AND a.MEMBER_RENTAL_DATE =  
b.MEMBER_RENTAL_DATE) MEM  
INNER JOIN MOVIES ON  
MEM.MOVIE_ID = MOVIES.MOVIE_ID
```

- **Indica el número de premios a los que estuvo nominado un actor, pero que no ha conseguido (Si una película está nominada a un premio, su actor también lo está)**

```
SELECT  
    AC.ACTOR_NAME,  
    SUM(A.AWARD_NOMINATION) AS AWARD_NOMINATION  
FROM PUBLIC.MOVIES M  
INNER JOIN PUBLIC.AWARDS A ON  
    A.MOVIE_ID = M.MOVIE_ID
```

```
INNER JOIN PUBLIC.MOVIES_ACTORS MA ON
    MA.MOVIE_ID = M.MOVIE_ID
INNER JOIN PUBLIC.ACTORS AC ON
    AC.ACTOR_ID = MA.ACTOR_ID
GROUP BY
    AC.ACTOR_NAME
```

- **Indica cuantos actores y directores hicieron películas para los estudios no activos**

```
SELECT COUNT(DISTINCT M.DIRECTOR_ID) AS DIRECTOR_NUMBER,
       COUNT(DISTINCT MA.ACTOR_ID) AS ACTOR_NUMBER
FROM PUBLIC.STUDIO S
INNER JOIN PUBLIC.MOVIES M ON
    M.STUDIO_ID = S.STUDIO_ID
INNER JOIN PUBLIC.MOVIES_ACTORS MA ON
    MA.MOVIE_ID = M.MOVIE_ID
WHERE
    S.STUDIO_ACTIVE = FALSE
```

- **Indica el nombre, ciudad, y teléfono de todos los miembros del videoclub que hayan alquilado películas que hayan sido nominadas a más de 150 premios y ganaran menos de 50**

```
SELECT
    M.MEMBER_NAME,
    M.MEMBER_TOWN,
    M.MEMBER_PHONE
FROM PUBLIC.MEMBERS_MOVIE_RENTAL MMR
INNER JOIN PUBLIC.MEMBERS M ON
    MMR.MEMBER_ID = M.MEMBER_ID
WHERE
    MMR.MOVIE_ID IN (
        SELECT
            MOVIE_ID
        FROM PUBLIC.AWARDS
        WHERE
            AWARD_NOMINATION > 150
            AND AWARD_WIN < 50)
```

- **Indica el número de películas que hayan hecho los directores durante las décadas de los 60, 70 y 80 que contengan la palabra "The" en cualquier parte del título**

```
SELECT
    D.DIRECTOR_NAME,
    COUNT(M.MOVIE_ID) AS MOVIE_NUMBER
FROM PUBLIC.MOVIES M
INNER JOIN PUBLIC.DIRECTORS D ON
    D.DIRECTOR_ID = M.DIRECTOR_ID
WHERE
    MOVIE_LAUNCH_DATE >= DATE '1960-01-01'
    AND MOVIE_LAUNCH_DATE <= DATE '1989-12-31'
    AND UPPER(M.MOVIE_NAME) LIKE '%THE%'
GROUP BY
    D.DIRECTOR_NAME
```

- **Indica si hay alguna coincidencia de nacimiento de ciudad (y si las hay, indicarlás) entre los miembros del videoclub y los directores.**

```
SELECT
    M.MEMBER_NAME,
    D.DIRECTOR_NAME,
    D.DIRECTOR_BIRTH_PLACE
FROM PUBLIC.MEMBERS M
INNER JOIN PUBLIC.DIRECTORS D ON
    M.MEMBER_TOWN = D.DIRECTOR_BIRTH_PLACE
```

- **Comprueba si hay errores en la BD entre las películas y directores (un director muerto en el 76 no puede dirigir una película en el 88)**

```
SELECT
    M.MOVIE_NAME,
    D.DIRECTOR_NAME,
    M.MOVIE_LAUNCH_DATE,
    D.DIRECTOR_DEAD_DATE
FROM PUBLIC.MOVIES M
INNER JOIN PUBLIC.DIRECTORS D ON
    M.DIRECTOR_ID = D.DIRECTOR_ID
```

WHERE

D.DIRECTOR_DEAD_DATE IS NOT NULL AND
D.DIRECTOR_DEAD_DATE < M.MOVIE_LAUNCH_DATE

- **Usando como condición la sentencia anterior, modifica la fecha de defunción a un año más tarde del estreno de la película (mediante sentencia SQL)**

--Este documento acerca de los MERGE en la documentación oficial de HSQL explica cómo conseguirlo: <http://hsqldb.org/doc/guide/dataaccess-chapt.html#dac_merge_statement>

```
MERGE INTO
    PUBLIC.DIRECTORS D
    USING (
    SELECT
        D.DIRECTOR_ID,
        MAX(DATEADD(YEAR, 1, M.MOVIE_LAUNCH_DATE)) AS
DIRECTOR_DEAD_DATE
    FROM
        PUBLIC.MOVIES M
    INNER JOIN PUBLIC.DIRECTORS D ON
        M.DIRECTOR_ID = D.DIRECTOR_ID
    WHERE
        D.DIRECTOR_DEAD_DATE IS NOT NULL
        AND D.DIRECTOR_DEAD_DATE < M.MOVIE_LAUNCH_DATE
    GROUP BY
        D.DIRECTOR_ID) AS L(DIRECTOR_ID,
DIRECTOR_DEAD_DATE) ON
    D.DIRECTOR_ID = L.DIRECTOR_ID
    WHEN MATCHED THEN UPDATE
    SET
        D.DIRECTOR_DEAD_DATE = L.DIRECTOR_DEAD_DATE
```