

FICHEROS Y BASES DE DATOS (E44)

Ejercicios de Algebra Relacional y Cálculo Relacional SOLUCIONES

Ejercicio 2

2.1.

Algebra Relacional

```
T1 := (PRESTAMO JOIN CINTA) [CODSOCIO,CODPELI,PRES_DEV]
T2 := (T1 WHERE PRES_DEV='prestada') [CODSOCIO,CODPELI]
      INTERSECT
      (T1 WHERE PRES_DEV='devuelta') [CODSOCIO,CODPELI]
RDO := (T2 JOIN SOCIO) [CODSOCIO,NOMBRE]
```

Cálculo Relacional

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE WHERE $ PRESTAMOX $ PRESTAMOY $ CINTAX $ CINTAY
(SOCIOX.CODSOCIO=PRESTAMOX.CODSOCIO AND PRESTAMOX.PRES_DEV='prestada' AND
PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND CINTAX.CODPELI=CINTAY.CODPELI AND
CINTAY.CODCINTA=PRESTAMOY.CODCINTA AND PRESTAMOY.PRES_DEV='devuelta' AND
PRESTAMOY.CODSOCIO=SOCIOX.CODSOCIO)
```

2.2.

Algebra Relacional

```
T1 := SUMMARIZE (PRESTAMO JOIN CINTA) GROUP BY (CODPELI) ADD COUNT(*) AS VECES
T2 := SUMMARIZE T1 GROUP BY () ADD MAX(VECES) AS VECES
RDO := (T1 JOIN T2 JOIN PELICULA) [CODPELI,TITULO]
```

2.3.

Algebra Relacional

```
T1 := (PRESTAMO JOIN CINTA) [CODSOCIO,CODPELI]
RDO := ((T1 DIVIDEBY SOCIO[CODSOCIO]) JOIN PELICULA) [CODPELI,TITULO]
```

Cálculo Relacional

títulos de las películas tales que para todos y cada uno de los socios existe al menos un préstamo (correspondiente a cada socio) de alguna de las copias (cintas) de esas películas

```
PELICULAX.CODPELI,PELICULAX.TITULO WHERE " SOCIOX $ PRESTAMOX $ CINTAX
(SOCIOX.CODSOCIO=PRESTAMOX.CODSOCIO AND PRESTAMOX.CODCINTA=CINTAX.CODCINTA
AND CINTAX.CODPELI=PELICULAX.CODPELI)
```

2.4.

Algebra Relacional

```
T1 := (PRESTAMO JOIN CINTA) [CODSOCIO,CODPELI]
T2 := SUMMARIZE T1 GROUP BY (CODSOCIO) ADD COUNT(*) AS PELIS_SOCIO
T3 := SUMMARIZE T2 GROUP BY () ADD MAX(PELIS_SOCIO) AS PELIS_SOCIO
RDO := (T2 JOIN T3 JOIN SOCIO) [CODSOCIO,NOMBRE,DIRECCION]
```

2.5.

Algebra Relacional

(todas las películas) menos (las películas que han sido prestadas alguna vez)

```
T1 := (PRESTAMO JOIN CINTA) [CODPELI]
RDO := ((PELICULA [CODPELI] MINUS T1) JOIN PELICULA) [CODPELI,TITULO]
```

Cálculo Relacional

títulos de las películas tales que para todas las cintas de esas películas, no existe préstamo de ellas

```
PELICULAX.CODPELI,PELICULAX.TITULO WHERE " CINTAX
(IF CINTAX.CODPELI=PELICULAX.CODPELI
THEN NOT $ PRESTAMOX (CINTAX.CODCINTA=PRESTAMOX.CODCINTA))
```

2.6.

Algebra Relacional

la lista de espera más larga corresponderá a la película que tenga el mayor número de socios

```
T1 := SUMMARIZE LISTA_ESPERA GROUP BY (CODPELI) ADD COUNT(*) AS LONG_LISTA
T2 := SUMMARIZE T1 GROUP BY ( ) ADD MAX(LONG_LISTA) AS LONG_LISTA
RDO := (T1 JOIN T2 JOIN PELICULA) [CODPELI,TITULO]
```

2.7.

Algebra Relacional

```
T1 := (SOCIO JOIN PRESTAMO JOIN CINTA JOIN
      (PELICULA WHERE TITULO='Blancanieves')) [CODSOCIO,NOMBRE]
T2 := (SOCIO JOIN LISTA_ESPERA JOIN
      (PELICULA WHERE TITULO='Blancanieves')) [CODSOCIO,NOMBRE]
RDO := (T1 UNION T2) [CODSOCIO,NOMBRE]
```

Cálculo Relacional

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE WHERE $ PELICULAX
(PELICULAX.TITULO='Blancanieves' AND
 (( $ PRESTAMOX $ CINTAX (SOCIOX.CODSOCIO=PRESTAMOX.CODSOCIO AND
 PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND CINTAX.CODPELI=PELICULAX.CODPELI) )
 OR
 ( $ LISTA_ESPERAX (SOCIOX.CODSOCIO=LISTA_ESPERAX.CODSOCIO AND
 LISTA_ESPERAX.CODPELI=PELICULAX.CODPELI) ) ) )
```

2.8.

Algebra Relacional

```
T1 := (SOCIO JOIN PRESTAMO JOIN CINTA JOIN
      (PELICULA WHERE TITULO='Blancanieves')) [CODSOCIO,NOMBRE]
T2 := (SOCIO JOIN LISTA_ESPERA JOIN
      (PELICULA WHERE TITULO='Blancanieves')) [CODSOCIO,NOMBRE]
RDO := (T1 INTERSECT T2) [CODSOCIO,NOMBRE]
```

Cálculo Relacional

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE WHERE $ PELICULAX
(PELICULAX.TITULO='Blancanieves' AND
 $ PRESTAMOX $ CINTAX (SOCIOX.CODSOCIO=PRESTAMOX.CODSOCIO AND
 PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND CINTAX.CODPELI=PELICULAX.CODPELI) AND
 $ LISTA_ESPERAX (SOCIOX.CODSOCIO=LISTA_ESPERAX.CODSOCIO AND
 LISTA_ESPERAX.CODPELI=PELICULAX.CODPELI) )
```

2.9.

Algebra Relacional

```
T1 := SUMMARIZE (PRESTAMO JOIN CINTA JOIN PELICULA)
      GROUP BY (GENERO) ADD COUNT(*) AS NUM_PRESTAMOS
RDO := (T1 WHERE NUM_PRESTAMOS>100) [GENERO]
```

2.10.

Algebra Relacional

```
T1 := LISTA_ESPERA JOIN SOCIO JOIN (PELICULA WHERE TITULO='E.T.')
T2 := SUMMARIZE T1 GROUP BY ( ) ADD MIN(FECHA) AS FECHA
RDO := (T1 JOIN T2) [CODSOCIO,NOMBRE,TELEFONO]
```

2.11.

Algebra Relacional

```
T1 := SOCIO JOIN PRESTAMO JOIN CINTA
T2 := SUMMARIZE T1 GROUP BY (CODSOCIO,NOMBRE,CODPELI) ADD COUNT(*) AS VECES
T3 := ((T2 WHERE VECES>=3) JOIN PELICULA) [CODSOCIO,NOMBRE,TITULO]
```

Ejercicio 3

3.1.

Algebra Relacional

```
T1 := PRESTAMO JOIN CINTA JOIN
      (PELICULA WHERE PRIMER_ACTOR='Alfredo Landa' OR SEGUNDO_ACTOR='Alfredo Landa')
T2 := SUMMARIZE T1[CODSOCIO,CODPELI] GROUP BY (CODSOCIO) ADD COUNT(*) AS NUMERO
T3 := SUMMARIZE T2 GROUP BY ( ) ADD MAX(NUMERO) AS NUMERO
RDO := (T2 JOIN T3 JOIN SOCIO) [CODSOCIO,NOMBRE,DIRECCION]
```

3.2.

Algebra Relacional

(las películas de los géneros que le gustan) menos (las películas que ya ha tomado prestadas)

```
T1 := ((GUSTA_GENERO WHERE CODSOCIO='194') JOIN PELICULA) [CODPELI,TITULO]
T2 := ((PRESTAMO WHERE CODSOCIO='194') JOIN CINTA JOIN PELICULA) [CODPELI,TITULO]
RDO := (T1 MINUS T2) [CODPELI,TITULO]
```

Cálculo Relacional

películas de algún género que le gusta al socio 194 tales que para toda cinta de esas películas, no existe ningún préstamo de ellas a ese socio

```
PELICULAX.CODPELI,PELICULAX.TITULO WHERE $ GUSTA_GENERO
(GUSTA_GENERO.CODSOCIO='194' AND PELICULAX.GENERO=GUSTA_GENERO.GENERO AND
" CINTAX (IF CINTAX.CODPELI=PELICULAX.CODPELI
THEN NOT $ PRESTAMO (PRESTAMO.CODSOCIO='194' AND
PRESTAMO.CODCINTA=CINTAX.CODCINTA)
```

3.3.

Algebra Relacional

(socios y películas que han tomado prestadas) dividido (películas de Ariadna Gil)

```
T1 := (PELICULA WHERE PRIMER_ACTOR='Ariadna Gil' OR SEGUNDO_ACTOR='Ariadna Gil') [CODPELI]
T2 := (T1 JOIN PRESTAMO JOIN CINTA) [CODSOCIO,CODPELI]
RDO := ((T2 DIVIDEBY T1) JOIN SOCIO) [CODSOCIO,NOMBRE,TELEFONO]
```

Cálculo Relacional

socios tales que para todas y cada una de las películas de Ariadna Gil, han tenido al menos un préstamo de alguna de sus copias (cintas)

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE, SOCIOX.TELEFONO WHERE " PELICULAX
(IF PELICULAX.PRIMER_ACTOR='Ariadna Gil' OR PELICULAX.SEGUNDO_ACTOR='Ariadna Gil'
THEN $ PRESTAMO $ CINTAX
(PRESTAMO.CODCINTA=CINTAX.CODCINTA AND CINTAX.CODPELI=PELICULAX.CODPELI AND
PRESTAMO.CODSOCIO=SOCIOX.CODSOCIO))
```

3.4.

Algebra Relacional

```
T1 := (PELICULA JOIN LISTA_ESPERA) [CODPELI,TITULO]
() [NOMBRE,CODPELI]
T2 := (T1 JOIN CINTA JOIN (PRESTAMO WHERE SYSDATE-FECHA>3 AND PRES_DEV='prestada')
JOIN SOCIO) [CODSOCIO,NOMBRE,TITULO]
```

Cálculo Relacional

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE,PELICULAX.TITULO WHERE $ PRESTAMO $ CINTAX
(PRESTAMO.CODSOCIO=SOCIOX.CODSOCIO AND PRESTAMO.CODCINTA=CINTAX.CODCINTA
AND CINTAX.CODPELI=PELICULAX.CODPELI AND PRESTAMO.PRES_DEV='prestada'
AND PRESTAMO.FECHA<'10-MAY-97' AND $ LISTA_ESPERAX
(LISTA_ESPERAX.CODPELI=PELICULAX.CODPELI))
```

3.5.

Algebra Relacional

```
T1 := (PELICULA WHERE PRIMER_ACTOR='Jorge Sanz' OR SEGUNDO_ACTOR='Jorge Sanz') [CODPELI]
T2 := (LISTA_ESPERA JOIN T1 JOIN SOCIO) [CODSOCIO,NOMBRE,TELEFONO]
T3 := ((PRESTAMO WHERE PRES_DEV='prestada') JOIN
CINTA JOIN T1 JOIN SOCIO) [CODSOCIO,NOMBRE,TELEFONO]
RDO := T2 UNION T3
```

Cálculo Relacional

```
SOCIOX.CODSOCIO,SOCIOX.NOMBRE,SOCIOX.TELEFONO WHERE $ PELICULAX
((PELICULAX.PRIMER_ACTOR='Jorge Sanz' OR PELICULAX.SEGUNDO_ACTOR='Jorge Sanz') AND
($ LISTA_ESPERAX (LISTA_ESPERAX.CODPELI=PELICULAX.CODPELI AND
LISTA_ESPERAX.CODSOCIO=SOCIOX.CODSOCIO)
OR
$ PRESTAMO $ CINTAX (PRESTAMO.CODCINTA=CINTAX.CODCINTA
AND PRESTAMO.CODSOCIO=SOCIOX.CODSOCIO AND CINTAX.CODPELI=PELICULAX.CODPELI
AND PRESTAMO.PRES_DEV='prestada')))
```

Ejercicio 4

4.1.

Algebra Relacional

```
T1 := (PELICULA JOIN REPARTO) [ACTOR,DIRECTOR,CODPELI]
RDO := SUMMARIZE T1 GROUP BY (ACTOR,DIRECTOR) ADD COUNT(*) AS NUM_PELIS
```

Algebra Relacional

(todos los socios) menos (socios que han tomado prestada alguna película de Vicente Aranda)

```
T1 := (PRESTAMO JOIN CINTA JOIN (PELICULA WHERE DIRECTOR='Vicente Aranda')) [CODSOCIO]
RDO := ((SOCIO [CODSOCIO] MINUS T1) JOIN SOCIO ) [CODSOCIO,NOMBRE]
```

Cálculo Relacional

socios tales que todos y cada uno de sus préstamos son de películas que no han sido dirigidas por Vicente Aranda

```
SOCIOX.CODSOCIO, SOCIOX.NOMBRE WHERE " PRESTAMOX
  (IF PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO
    THEN $ CINTAX $ PELICULAX
      (PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND CINTAX.CODPELI=PELICULAX.CODPELI AND
        PELICULAX.DIRECTOR<>'Vicente Aranda'))
```

4.3.

Algebra Relacional

(actor y películas en que ha intervenido) dividido (películas dirigidas por Fernando Trueba)

```
RDO := REPARTO DIVIDE BY (PELICULA WHERE DIRECTOR='Fernando Trueba')[CODPELI]
```

Cálculo Relacional

actores tales que para todas y cada una de las películas dirigidas por Fernando Trueba, han intervenido en ellas (están en el reparto)

```
REPARTOX.ACTOR WHERE " PELICULAX
  (IF PELICULAX.DIRECTOR='Fernando Trueba' THEN $ REPARTOY
    (REPARTOY.ACTOR=REPARTOX.ACTOR AND REPARTOY.CODPELI=PELICULAX.CODPELI))
```

4.4.

Algebra Relacional

```
T1 := (GUSTA_ACTOR WHERE ACTOR='Gabino Diego') [CODSOCIO]
T2 := ((PRESTAMO WHERE FECHA>='1-ENE-94' AND FECHA<='31-DIC-94') JOIN CINTA JOIN
  (REPARTO WHERE ACTOR='Gabino Diego') JOIN T1) [CODSOCIO, CODPELI]
T3 := SUMMARIZE T2 GROUP BY (CODSOCIO) ADD COUNT(*) AS NUM_PELIS
RDO := ((T3 WHERE NUM_PELIS>=3) JOIN SOCIO) [CODSOCIO, NOMBRE, DIRECCION, TELEFONO]
```

4.5.

Algebra Relacional

(socios a los que les gusta algún actor y algún director) menos (socios a los que les gustan actores que han trabajado en alguna de las películas dirigidas por sus directores favoritos)

```
T1 := (GUSTA_ACTOR JOIN GUSTA_DIRECTOR JOIN PELICULA) [CODSOCIO, ACTOR, DIRECTOR, CODPELI]
T2 := (T1 JOIN REPARTO) [CODSOCIO]
T3 := GUSTA_ACTOR [CODSOCIO] INTERSECT GUSTA_DIRECTOR [CODSOCIO]
RDO := (T3 MINUS T2) JOIN SOCIO
```

Cálculo Relacional

socios a los que les gusta algún actor y algún director, tales que para todos y cada uno de los directores que le gustan al socio y para todas las películas dirigidas por esos directores, no hay ninguna tupla en REPARTO que diga que ese actor que le gusta al socio, ha participado en ellas

```
SOCIOX WHERE $ GUSTA_DIRECTORX $ GUSTA_ACTORX
  (GUSTA_DIRECTORX.CODSOCIO=SOCIOX.CODSOCIO AND GUSTA_ACTORX.CODSOCIO=SOCIOX.CODSOCIO)
  AND " GUSTA_DIRECTORX " PELICULAX " REPARTOX
    (IF (SOCIOX.CODSOCIO=GUSTA_DIRECTORX.CODSOCIO
      AND GUSTA_DIRECTORX.DIRECTOR=PELICULAX.DIRECTOR
      AND PELICULAX.CODPELI=REPARTOX.CODPELI)
      THEN NOT $ GUSTA_ACTORX
        (GUSTA_ACTORX.ACTOR=REPARTOX.ACTOR AND
          GUSTA_ACTORX.CODSOCIO=SOCIOX.CODSOCIO))
```

4.6.

(socios a los que les gusta algún actor)
menos

(socios a los que les gusta alguno de los actores que han participado en alguna de las películas que han tomado prestadas más veces)

```
T1 := GUSTA_ACTOR [CODSOCIO]
T2 := (PELICULA JOIN CINTA JOIN PRESTAMO)
T3 := SUMMARIZE T2 GROUP BY (CODSOCIO, CODPELI) ADD COUNT(*) AS NUMVECES
T4 := SUMMARIZE T3 GROUP BY (CODSOCIO) ADD MAX (NUMVECES) AS NUMVECES
T5 := (T3 JOIN T4)[CODSOCIO, CODPELI]
T5 : socios y películas que han tomado prestadas más veces
T6 := (T5 JOIN REPARTO JOIN GUSTA_ACTOR) [CODSOCIO]
T6 : socios a los que les gusta alguno de los actores que han participado en alguna de las películas que han tomado prestadas más veces
RDO := ( (T1 MINUS T6) JOIN SOCIO ) [CODSOCIO, NOMBRE]
```

4.7.

Obtiene los socios que han visto todas las películas del año 95 que hay en el vídeo club de un género.

4.8.

```
SOCIOX.NOMBRE WHERE
  " GUSTA_DIRECTORX " PELICULAX
  (IF (GUSTA_DIRECTORX.CODSOCIO = SOCIOX.CODSOCIO AND
      GUSTA_DIRECTORX.DIRECTOR = PELICULAX.DIRECTOR)
      THEN $ PRESTAMOX $ CINTAX
      (SOCIOX.CODSOCIO = PRESTAMOX.CODSOCIO AND
      PRESTAMOX.CODCINTA = CINTAX.CODCINTA AND
      CINTAX.CODPELI = PELICULAX.CODPELI)
  AND " GUSTA_ACTORX " REPARTOX
  (IF (GUSTA_ACTORX.CODSOCIO = SOCIOX.CODSOCIO AND
      GUSTA_ACTORX.ACTOR = REPARTOX.ACTOR)
      THEN $ PRESTAMOX $ CINTAX
      (SOCIOX.CODSOCIO = PRESTAMOX.CODSOCIO AND
      PRESTAMOX.CODCINTA = CINTAX.CODCINTA AND
      CINTAX.CODPELI = REPARTOX.CODPELI)
```

4.9.

Obtiene los títulos de las películas de las que hay copias en todos los idiomas de los que hay películas en el vídeo club.

Ejercicio 5

5.1.

```
T1 := (OBRA_MAESTRA JOIN PELICULA) WHERE GENERO='ciencia ficción'
T2 := SUMMARIZE T1 GROUP BY (DIRECTOR) ADD COUNT(*) AS NUMPELIS
T3 := SUMMARIZE T2 GROUP BY ( ) ADD MAX(NUMPELIS) AS NUMPELIS
RDO := (T2 JOIN T3) [DIRECTOR]
```

5.2.

La expresión obtiene el nombre de los socios que no han tomado prestada ninguna película donde Fernando Fernán Gómez participaba en el reparto como actor, pero que sí han tomado prestada alguna película dirigida por él.

5.3.

```
SOCIOX.NOMBRE WHERE $ OBRA_MAESTRAX $ CINTAX $ PRESTAMOX
  (OBRA_MAESTRAX.DIRECTOR='Luis Buñuel' AND OBRA_MAESTRAX.CODPELI=CINTAX.CODPELI AND
  CINTAX.CODCINTA=PRESTAMOX.CODCINTA AND PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO)
```

5.4.

La expresión del cálculo obtiene los directores cuyas películas están todas consideradas como obras maestras.

5.5.

```
T1 := PRESTAMO JOIN CINTA JOIN REPARTO
T2 := SUMMARIZE T1 GROUP BY (ACTOR,CODPELI) ADD COUNT(*) AS NUM_PRESTAMOS
T3 := SUMMARIZE T2 GROUP BY (ACTOR) ADD MAX(NUM_PRESTAMOS) AS NUM_PRESTAMOS
RDO := (T2 JOIN T3 JOIN PELICULA) [ACTOR,TITULO]
```

5.6.

La expresión obtiene los actores que han participado en el reparto de todas las obras maestras de Luis Buñuel.

5.7.

```
REPARTOX.ACTOR WHERE
  " OBRA_MAESTRAX (IF OBRA_MAESTRAX.DIRECTOR='Luis Buñuel' THEN $ REPARTOY
  (REPARTOY.ACTOR=REPARTOX.ACTOR AND REPARTOY.CODPELI=OBRA_MAESTRAX.CODPELI))
  AND
  " PELICULAX (IF (PELICULAX.DIRECTOR='Luis Buñuel' AND
  NOT $ OBRA_MAESTRAX (PELICULAX.CODPELI=OBRA_MAESTRAX.CODPELI))
  THEN NOT $ REPARTOY (REPARTOY.ACTOR=REPARTOX.ACTOR AND
  REPARTOY.CODPELI=PELICULAX.CODPELI))
```

5.8.

La expresión obtiene los actores que no han participado en el reparto de ninguna película del director Luis Buñuel.

Ejercicio 6

6.1.

```
T1 := SUMMARIZE (PREMIO WHERE AÑO='1994') GROUP BY (CODPELI) ADD COUNT(*) AS PREMIOS
T2 := SUMMARIZE T1 GROUP BY ( ) ADD MAX(PREMIOS) AS PREMIOS
T3 := (T1 JOIN T2) [CODPELI]
```

```
T4 := (PRESTAMO JOIN CINTA JOIN T3) [CODSOCIO,CODPELI]
T5 := SUMMARIZE T4 GROUP BY (CODPELI) ADD COUNT(*) AS NUM_SOCIOS
```

6.2.

La expresión obtiene los nombres de los socios sólo han tomado prestadas películas de nacionalidad española.

6.3.

```
PELICULAX.TITULO WHERE
  $ PREMIOX (PREMIOX.PREMIO LIKE 'Goya%' AND PREMIOX.AÑO='1993'
    AND PREMIOX.CODPELI=PELICULAX.CODPELI)
  AND NOT $ PRESTAMOX $ CINTAX (PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND
    CINTAX.CODPELI=PELICULAX.CODPELI AND PRESTAMOX.FECHA>='1-MAR-96')
```

6.4.

La expresión obtiene los nombres de los socios que sólo han tomado prestadas películas que tienen algún premio.

6.5.

```
T1 := (PELICULA JOIN (PREMIO WHERE PREMIO='Oscar') JOIN DISTRIBUIDORA) [CODDISTR,NOMBRE]
RDO := (DISTRIBUIDORA [CODDISTR,NOMBRE] MINUS T1) [NOMBRE]
```

6.6.

La expresión obtiene los datos de la/s compañía/s que distribuye/n el mayor número de películas españolas que han sido premiadas en el mismo año en que se rodaron.

6.7.

Suponemos que lo que se quiere obtener son los socios que han tomado prestadas todas las versiones en búlgaro de las películas del vídeo club.
si como mucho hay una copia en búlgaro por película:

```
SOCIOX.NOMBRE WHERE " PELICULAX $ CINTAX
  (IF (PELICULAX.CODPELI=CINTAX.CODPELI AND CINTAX.IDIOMA='Búlgaro')
    THEN $ PRESTAMOX (PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND
      PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO))
  AND NOT $ PRESTAMOX $ CINTAX
    (PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO AND PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND
      CINTAX.IDIOMA<>'Búlgaro')
```

si puede haber varias copias en búlgaro por película:

```
SOCIOX.NOMBRE WHERE " PELICULAX $ CINTAX
  (IF (PELICULAX.CODPELI=CINTAX.CODPELI AND CINTAX.IDIOMA='Búlgaro')
    THEN $ PRESTAMOX $ CINTAY (CINTAY.IDIOMA='Búlgaro' AND PRESTAMOX.CODCINTA=CINTAY.CODCINTA
      AND CINTAY.CODPELI=CINTAX.CODPELI AND PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO))
  AND NOT $ PRESTAMOX $ CINTAX
    (PRESTAMOX.CODSOCIO=SOCIOX.CODSOCIO AND PRESTAMOX.CODCINTA=CINTAX.CODCINTA AND
      CINTAX.IDIOMA<>'Búlgaro')
```

6.8.

La expresión obtiene el nombre de los socios que sólo han tomado prestadas películas que son de una misma compañía distribuidora.

6.9.

```
(películas de terror prestadas más de 100 veces)
  intersección
(películas de terror más premiadas)

T0 := (PELICULA WHERE GENERO='terror')[CODPELI]
T1 := SUMMARIZE (T0 JOIN CINTA JOIN PRESTAMO) GROUP BY (CODPELI)
      ADD COUNT(*) AS PRESTAMOS
T2 := (T1 WHERE PRESTAMOS>100) [CODPELI]
T3 := SUMMARIZE (T0 JOIN PREMIO) GROUP BY (CODPELI) ADD COUNT(*) AS PREMIOS
T4 := SUMMARIZE T3 GROUP BY ( ) ADD MAX(PREMIOS) AS PREMIOS
T5 := (T3 JOIN T4) [CODPELI]
RDO := ((T2 INTERSECT T5) JOIN PELICULA)[CODPELI,TITULO]
```

6.10.

Directores cuyas películas (todas ellas) son distribuidas por la misma compañía. El nombre de la distribuidora se muestra junto con el del director.

(es una división generalizada : funciona como si hiciésemos una división para cada director distinto, obteniendo las distribuidoras que distribuyen todas las películas del director)

6.11.

Para recorrer los directores se define una nueva variable tupla:
RANGE OF DIRECTORX IS PELICULAX.DIRECTOR

```
DIRECTORX WHERE " PELICULAX
  (IF PELICULAX.DIRECTOR=DIRECTORX.DIRECTOR THEN $CINTAX $CINTAY
```

```
( CINTAX.CODPELI=PELICULAX.CODPELI AND CINTAX.IDIOMA='francés' AND  
  CINTAY.CODPELI=PELICULAX.CODPELI AND CINTAY.IDIOMA='español' ))
```

6.12.

Distribuidoras de las que no tenemos copia en inglés de ninguna sus películas premiadas.

6.13.

```
T0 := (PRESTAMO JOIN CINTA JOIN PELICULA) [CODSOCIO,DIRECTOR,CODPELI,TITULO]  
T1 := SUMMARIZE T0 GROUP BY (DIRECTOR,TITULO) ADD COUNT(*) AS NUMSOCIOS  
T2 := SUMMARIZE T1 GROUP BY (DIRECTOR) ADD MAX(NUMSOCIOS) AS NUMSOCIOS  
RDO := (T1 JOIN T2) [DIRECTOR,TITULO]
```


6.14.

Películas de nacionalidad inglesa que no están prestadas actualmente y de las que hay copia en inglés.

6.15.

```

PELICULAX.DIRECTOR,PELICULAX.TITULO WHERE $ PELICULAY
  (PELICULAY.TITULO = 'Mars Attacks'
   AND " PREMIOY (IF PREMIOY.CODPELI = PELICULAY.CODPELI
               THEN $ PREMIOX
                   (PELICULAX.CODPELI = PREMIOX.CODPELI AND
                    PREMIOY.PREMIO = PREMIOX.PREMIO) )

```

6.16.

Socios que no tienen prestada ninguna película premiada.

Ejercicio 7

7.1.

```

T1 := (REPARTO WHERE ACTOR='Morgan Fernández')[CODPELI]
T2 := (REPARTO WHERE ACTOR='Flanagan García')[CODPELI]
T3 := T1 INTERSECT T2
T4 := (PRESTAMO JOIN CINTA)[CODSOCIO,CODPELI]
T5 := T4 DIVIDE BY T3
RDO := SUMMARIZE T5 GROUP BY ( ) ADD COUNT(*) AS NUMSOCIOS

```

7.2.

```

T1 := (PELICULA JOIN REPARTO JOIN PREMIO [CODPELI]) [DIRECTOR,ACTOR]
T2 := (PELICULA JOIN PREMIO [CODPELI,ACTOR]) [DIRECTOR,ACTOR]
RDO := T1 MINUS T2

```

7.3.

```

SOCIOX.CODSOCIO,SOCIOX.NOMBRE,SOCIOX.TELEFONO WHERE
  $ PELICULAX $ CINTAX $ PRESTAMOX
    (PELICULAX.DIRECTOR = 'Paco Smith' AND PELICULA.AÑO<1992
     AND PELICULAX.CODPELI = CINTAX.CODPELI
  AND CINTAX.CODCINTA = PRESTAMOX.CODCINTA
     AND PRESTAMOX.CODSOCIO = SOCIOX.CODSOCIO)
  AND " PELICULAX (IF (PELICULAX.DIRECTOR = 'Paco Smith' AND PELICULAX.AÑO>1992)
               THEN $ CINTAX NOT $ PRESTAMOX
                   (PELICULAX.CODPELI = CINTAX.CODPELI AND
                    CINTAX.CODCINTA = PRESTAMOX.CODCINTA AND
                     PRESTAMOX.CODSOCIO = SOCIOX.CODSOCIO))

```

7.4.

```

PELICULAX.DIRECTOR,REPARTOX.ACTOR WHERE
  (PELICULAX.CODPELI = REPARTOX.CODPELI AND
   " PELICULAY (IF PELICULAY.DIRECTOR = PELICULAX.DIRECTOR AND
               $ REPARTOY (REPARTOY.ACTOR = REPARTOX.ACTOR AND
                           REPARTOY.CODPELI = PELICULAY.CODPELI)
               THEN $ PREMIOX (PREMIOX.CODPELI = PELICULAY.CODPELI AND
                               PREMIOX.ACTOR = REPARTOX.ACTOR)))

```

7.5.

Algebra Relacional

```

T1 := ( PELICULA JOIN PREMIO[CODPELI,PREMIO] ) [DIRECTOR,PREMIO]
      directores y tipos de premio que se han llevado
T2 := PREMIO [PREMIO]
      tipos de premio
RDO := T1 DIVIDEBY T2

```

Cálculo Relacional

```

RANGE OF DIRECTORX IS PELICULAX.DIRECTOR
RANGE OF TIPOX  IS PREMIOX.PREMIO

```

```

DIRECTORX WHERE " TIPOX $ PELICULAX $ PREMIOX
  ( DIRECTORX.DIRECTOR = PELICULAX.DIRECTOR AND
    PELICULAX.CODPELI = PREMIOX.CODPELI AND
    PREMIOX.PREMIO = TIPOX.PREMIO )

```

o bien

```

PELICULAX.DIRECTOR WHERE " PREMIOX $ PELICULAY $ PREMIOY
  ( PELICULAX.DIRECTOR = PELICULAY.DIRECTOR AND
    PELICULAY.CODPELI = PREMIOY.CODPELI AND
    PREMIOY.PREMIO = PREMIOX.PREMIO )

```

7.6.

Algebra Relacional

```
T1 := PELICULA [DIRECTOR, GENERO]
      directores y géneros de los que tienen películas
T2 := SUMMARIZE T1 GROUP BY (DIRECTOR) ADD COUNT(*) AS NUM_GENEROS
      directores y número de géneros distintos de los que tienen películas
T3 := (T2 WHERE NUM_GENEROS = 1) [DIRECTOR]
      directores que siempre hacen películas de un mismo género
T4 := (PELICULA JOIN PREMIO[CODPELI]) [DIRECTOR]
      directores cuyas películas han ganado algún premio
RDO := T3 MINUS T4
      directores que siempre han hecho películas de un mismo género y
      que no han ganado ningún premio
```

Cálculo Relacional

```
PELICULAX.DIRECTOR WHERE " PELICULAY
      ( IF PELICULAY.DIRECTOR = PELICULAX.DIRECTOR
        THEN (PELICULAY.GENERO = PELICULAX.GENERO
              AND NOT $ PERMIOX
              (PREMIOX.CODPELI=PELICULAY.CODPELI)))
```

7.7.

Algebra Relacional

```
T1 := (PELICULA JOIN (CINTA WHERE IDIOMA='español'))[CODPELI,TITULO,DIRECTOR]
      películas de las que hay copias en español
T2 := (PELICULA [CODPELI,TITULO,DIRECTOR]) MINUS T1
      películas de las que solo hay copias en idiomas extranjeros
T3 := (PELICULA JOIN CINTA JOIN PRESTAMO)[CODPELI,TITULO,DIRECTOR]
      películas que han sido prestadas alguna vez
RDO := T2 MINUS T3
      películas de las que solo hay copias en idiomas extranjeros y
      que nunca han sido prestadas
```

Cálculo Relacional

```
PELICULAX.CODPELI, PELICULAX.TITULO, PELICULAX.DIRECTOR WHERE
      " CINTAX ( IF CINTAX.CODPELI=PELICULAX.CODPELI
                THEN CINTAX.IDIOMA <> 'español' AND
                NOT $ PRESTAMOX (PRESTAMOX.CODCINTA=CINTAX.CODCINTA ) )
      películas cuyas copias están todas en idiomas extranjeros y ninguna de ellas
      ha sido prestada.
```

7.8.

Algebra Relacional

```
RDO := SUMMARIZE PREMIO GROUP BY (PREMIO) ADD MIN(FECHA) AS FECHA
```

Cálculo Relacional

```
PREMIOX.PREMIO, PREMIOX.FECHA WHERE
      " PREMIOY ( IF PREMIOY.PREMIO = PREMIOX.PREMIO
                THEN PREMIOY.FECHA >= PREMIOX.FECHA )
```

Ejercicio 8

8.1.

Algebra Relacional

```
T1 := ACTOR JOIN REPARTO JOIN PELICULA
      actores que han hecho alguna película de su misma nacionalidad
T2 := PELICULA RENAME nacionalidad AS nacion_peli
T3 := (ACTOR JOIN REPARTO JOIN T2) WHERE nacionalidad <> nacion_peli
      actores que han hecho alguna película de
      una nacionalidad distinta de la suya
RDO := T1[codactor, nombre] MINUS T2[codactor, nombre]
```

Cálculo Relacional

```
ACTORX.nombre WHERE " REPARTOX $PELICULAX
      ( IF REPARTOX.codactor=ACTORX.codactor
        THEN REPARTOX.codpeli = PELICULAX.codpeli AND
              PELICULAX.nacionalidad = ACTORX.nacionalidad )
```

8.2.

Algebra Relacional

```
( ACTORES PREMIADOS ) MINUS ( ACTORES PREMIADOS EN PELIS
```

DE SU MISMA NACIONALIDAD)

```
T1 := PREMIO [codactor]
      actores premiados
T2 := ACTOR JOIN REPARTO JOIN PELICULA JOIN PREMIO[codpeli ,codactor]
      actores que han hecho alguna película de su misma nacionalidad
      por la que han recibido algún premio
RDO := ( ( T1 MINUS T2 ) JOIN ACTOR ) [nombre]
```

Cálculo Relacional

```
RANGE OF ACT_PREMIADOX IS
      ACTORX WHERE $PREMIOX (PREMIOX.codactor = ACTORX.codactor)

ACT_PREMIADOX.nombre WHERE "REPARTOX $PELICULAX
      ( IF REPARTOX.codactor = ACT_PREMIADOX.codactor AND
        PELICULAX.codpeli = REPARTOX.codpeli AND
        PELICULAX.nacionalidad = ACT_PREMIADOX.nacionalidad
      THEN NOT $PREMIOX ( PREMIOX.codpeli = PELICULAX.codpeli AND
        PREMIOX.codactor = ACT_PREMIADOX.codactor ) )
```

8.3.

Algebra Relacional

```
T1 := (ACTOR RENAME AÑO_NACIMIENTO AS AÑO_ACTOR)[CODACTOR,AÑO_ACTOR]
T2 := (DIRECTOR RENAME AÑO_NACIMIENTO AS AÑO_DIRECTOR)[CODDIRECTOR,AÑO_DIRECTOR]
T3 := (PELICULA JOIN T2 JOIN REPARTO JOIN T1) WHERE AÑO_ACTOR < AÑO_DIRECTOR
      actores que han participado en películas de directores más jóvenes que ellos
RDO := ((REPARTO[CODACTOR] MINUS T3[CODACTOR]) JOIN ACTOR)[NOMBRE]
```

Cálculo Relacional

```
ACTORX.NOMBRE WHERE " REPARTOX
      (IF REPARTOX.CODACTOR=ACTORX.CODACTOR
      THEN $ DIRECTORX $ PELICULAX
      (PELICULAX.CODEPELI=REPARTOX.CODEPELI AND
      PELICULAX.CODDIRECTOR=DIRECTORX.CODDIRECTOR AND
      DIRECTORX.AÑO_NACIMIENTO<=ACTORX.AÑO_NACIMIENTO))
```

8.4.

```
T1 := REPARTO JOIN (PREMIO[CODEPELI,PREMIO])
T2 := SUMMARIZE T1 GROUP BY (CODEPELI,CODACTOR) ADD COUNT(*) AS PREMIOS
T3 := SUMMARIZE T2 GROUP BY (CODACTOR) ADD MAX(PREMIOS) AS PREMIOS
RDO := (T2 JOIN T3 JOIN PELICULA[CODEPELI,TÍTULO]
      JOIN ACTOR[CODACTOR,NOMBRE])[NOMBRE,TÍTULO]
```

8.5.

```
DIRECTORX.NOMBRE WHERE $ PELICULAX
      (PELICULAX.CODDIRECTOR=DIRECTORX.CODDIRECTOR AND
      PELICULAX.AÑO-DIRECTORX.AÑO_NACIMIENTO)<=25)
```

Ejercicio 9

9.1.

```
T1 := (PREMIO[CODEPELI] JOIN PELICULA)[CODEPELI,DIRECTOR]
      directores y sus películas premiadas
T2 := (T1 JOIN CINTA JOIN PRESTAMO)[CODEPELI,DIRECTOR,CODSOCIO]
      socios y películas premiadas que han visto
T3 := SUMMARIZE T2 GROUP BY(CODSOCIO,DIRECTOR) ADD COUNT(*) AS NUMPELI
      socios y número de películas premiadas que han visto de cada director
T4 := (T3 WHERE NUMPELI >= 3)[CODSOCIO,DIRECTOR]
      socios y número de películas premiadas que han visto de cada director, si son 3 o más
T5 := (T2[CODSOCIO,CODEPELI] DIVIDE BY T1)
      socios que han visto todas las películas premiadas de un mismo director
RDO := ((T4 UNION T5) JOIN SOCIO)[NOMBRE,DIRECTOR]
```

9.2.

Nombre de los socios a los que sólo les gustan directores que tienen alguna película premiada.

9.3.

```
SOCIOX.NOMBRE WHERE ∇ PREMIOX
      IF (PREMIOX.AÑO = '1996') THEN
        (∃ PELICULAX ∃ CINTAX ∃ PRESTAMOX
          (PREMIOX.CODEPELI = PELICULAX.CODEPELI AND
          PELICULAX.CODEPELI = CINTAX.CODEPELI AND
          CINTAX.CODCINTA= PRESTAMOX.CODCINTA AND
          PRESTAMOX.CODSOCIO= SOCIOX.CODSOCIO AND
          CINTAX.IDIOMA = PELICULAX.IDIOMA )
        AND ∃ CINTAX ∃ PRESTAMOX
```

```
(PREMIOX.CODPELI = CINTAX.CODPELI AND
CINTAX.CODCINTA= PRESTAMOX.CODCINTA AND
PRESTAMOX.CODSOCIO= SOCIOX.CODSOCIO AND
CINTAX.IDIOMA = 'castellano' ))
```

9.4.

Género e idioma, tales que la versión original de todas las películas del género están en el idioma mostrado.

Ejercicio 10

10.1.

```
T1 := ((PRESTAMO JOIN CINTA)[codpeli,fecha]) WHERE TO_CHAR(fecha,'YY')='97'
T2 := EXTEND T1 ADD TO_NUMBER(TO_CHAR(fecha,'MM')) AS mes
T3 := SUMMARIZE T2 GROUP BY (codpeli) ADD COUNT(DISTINCT mes) AS num_mes
T4 := T3 WHERE num_mes = 12
      películas vistas todos los meses del año 97
T5 := (T4 JOIN PELICULA JOIN LISTA_ESPERA)[codpeli,genero]
      películas vistas todos los meses del año 97 y que están en lista de espera
T6 := SUMMARIZE T5 GROUP BY (genero) ADD COUNT(*) AS num_peli
T7 := SUMMARIZE T6 GROUP BY () ADD MAX(num_peli) AS num_peli
RDO := (T6 JOIN T7)[genero]
```

10.2.

Nombre de los socios que sólo han visto películas de un único país, que también se muestra.

10.3.

```
SOCIOX.nombre WHERE (∀ LISTA_ESPERAX
  IF (SOCIOX.codsocio = LISTA_ESPERAX.codsocio) THEN
    (∀ REPARTOX
      IF (LISTA_ESPERAX.codpeli = REPARTOX.codpeli ) THEN
        (∃ GUSTA_ACTOR
          (REPARTOX.actor = GUSTA_ACTORX.actor AND
           GUSTA_ACTORX.codsocio = SOCIOX.codsocio))))
  AND ∃ LISTA_ESPERAX
    (SOCIOX.codsocio = LISTA_ESPERAX.codsocio)
```

10.4.

Actores tales que todas sus películas están en la lista de espera.

Ejercicio 11

11.1.

```
T1 := (REPARTO JOIN LISTA_ESPERA)
T2 := SUMMARIZE T2 GROUP BY (actor) ADD COUNT(*) AS num1
      número de películas en lista de espera de cada actor
T3 := (REPARTO JOIN CINTA JOIN PRESTAMO)
      número de películas prestadas de cada actor
T4 := SUMMARIZE T3 GROUP BY (actor) ADD COUNT(*) AS num2
T5 := EXTEND (T2 JOIN T4) ADD (num1+num2) AS num
T6 := SUMMARIZE T5 GROUP BY () ADD MAX(num) AS num
RDO := (T5 JOIN T6)[actor]
```

11.2.

Socios que han visto películas en versión original, pero ninguna era británica.

11.3.

```
DISTRIBUIDORAX.nombre WHERE ∀ PELICULAX
  IF (DISTRIBUIDORAX.coddistr = PELICULAX.coddistr) THEN
    (∃ LISTA_ESPERAX
      (PELICULAX.codpeli = LISTA_ESPERAX.codpeli)
    AND
      ∀ CINTAX
        IF (CINTAX.codpeli = PELICULAX.codpeli ) THEN
          ∃ PRESTAMOX
            (PELICULAX.codpeli = CINTAX.codpeli AND
             CINTAX.codcinta = PRESTAMOX.codcinta AND
             PRESTAMOX.pres_dev = 'prestada'))
```

11.4.

Socios tales que todas las películas que han tomado prestadas alguna vez aparecen en la lista de espera.