

## Ficheros y Bases de Datos

## Curso 99/00

### Soluciones a los ejercicios de álgebra relacional y cálculo relacional

#### Esquema 2

##### 2.1 Álgebra 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]
```

##### 2.1 Cálculo relacional

```
SOCX.codsocio,SOCX.nombre WHERE ∃ PRESX ∃ PRESY ∃ CINUX ∃ CINY
(SOCX.codsocio = PRESX.codsocio AND PRESX.pres_dev = 'prestada' AND
PRESX.codcinta = CINUX.codcinta AND CINUX.codpeli = CINY.codpeli AND
CINY.codcinta = PRESY.codcinta AND PRESY.pres_dev = 'devuelta' AND
PRESY.codsocio = SOCX.codsocio)
```

##### 2.2 Álgebra 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,título]
```

##### 2.3 Álgebra relacional

```
T1 := (PRESTAMO JOIN CINTA) [codsocio,codpeli]
RDO := ((T1 DIVIDEBY SOCIO[codsocio]) JOIN PELICULA) [codpeli,título]
```

##### 2.3 Cálculo relacional

Hay que encontrar los títulos de las películas tales que para todos y cada uno de los socios ( $\forall$  SOCX) existe al menos un préstamo ( $\exists$  PRESX) de alguna de sus copias ( $\exists$  CINUX).

```
PELIX.codpeli,PELIX.título WHERE ∀ SOCX ∃ PRESX ∃ CINUX
(SOCX.codsocio = PRESX.codsocio AND PRESX.codcinta = CINUX.codcinta AND
CINUX.codpeli = PELIX.codpeli)
```

##### 2.4 Álgebra 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,dirección]
```

##### 2.5 Álgebra 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,título]
```

#### Ejercicios de álgebra relacional y cálculo relacional

##### 2.5 Cálculo relacional

Títulos de las películas tales que para todas las cintas ( $\forall$  CINUX) de esas películas (IF), no existe préstamo (NOT  $\exists$  PRESX) de ellas.

```
PELIX.codpeli,PELIX.título WHERE ∀ CINUX
(IF CINUX.codpeli = PELIX.codpeli
THEN NOT ∃ PRESX (CINUX.codcinta = PRESX.codcinta))
```

##### 2.6 Álgebra relacional

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

```
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,título]
```

##### 2.7 Álgebra relacional

```
T1 := (SOCIO JOIN PRESTAMO JOIN CINTA JOIN
      (PELICULA WHERE título = 'Blancanieves')) [codsocio,nombre]
T2 := (SOCIO JOIN LISTA_ESPERA JOIN
      (PELICULA WHERE título = 'Blancanieves')) [codsocio,nombre]
RDO := (T1 UNION T2) [codsocio,nombre]
```

##### 2.7 Cálculo relacional

```
SOCX.codsocio,SOCX.nombre WHERE ∃ PELIX
(PELIX.título='Blancanieves' AND
((∃ PRESX ∃ CINUX
(SOCX.codsocio=PREX.codsocio AND PRESX.codcinta=CINUX.codcinta AND
CINUX.codpeli=PELIX.codpeli))
OR
(∃ LISTX (SOCX.codsocio=LISTX.codsocio AND LISTX.codpeli=PELIX.codpeli))))
```

##### 2.8 Álgebra relacional

```
T1 := (SOCIO JOIN PRESTAMO JOIN CINTA JOIN
      (PELICULA WHERE título = 'Blancanieves')) [codsocio,nombre]
T2 := (SOCIO JOIN LISTA_ESPERA JOIN
      (PELICULA WHERE título = 'Blancanieves')) [codsocio,nombre]
RDO := (T1 INTERSECT T2) [codsocio,nombre]
```

##### 2.8 Cálculo relacional

```
SOCX.codsocio,SOCX.nombre WHERE ∃ PELIX
(PELIX.título = 'Blancanieves' AND
∃ PRESX ∃ CINUX (SOCX.codsocio = PRESX.codsocio AND
PRESX.codcinta = CINUX.codcinta AND
CINUX.codpeli = PELIX.codpeli) AND
∃ LISTX (SOCX.codsocio = LISTX.codsocio AND
LISTX.codpeli = PELIX.codpeli))
```

##### 2.9 Álgebra relacional

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

## 2.10 Álgebra relacional

```
T1 := LISTA_ESPERA JOIN SOCIO JOIN (PELICULA WHERE título = 'E.T.')
T2 := SUMMARIZE T1 GROUP BY () ADD MIN(fecha) AS fecha
RDO := (T1 JOIN T2) [codsocio,nombre,teléfono]
```

## 2.10 Álgebra relacional

```
SOCX.codsocio,SOCX.nombre,SOCX.teléfono WHERE
  ∃ LISTX ∃ PELIX
  (LISTX.codsocio = SOCX.codsocio AND LISTX.codpeli = PELIX.codpeli AND
  PELIX.título = 'E.T.' AND ∀ LISTY (IF LISTY.codpeli = LISTX.codpeli
  THEN LISTY.fecha >= LISTX.fecha))
```

## 2.11 Álgebra 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,título]
```

**Esquema 3**

## 3.1 Álgebra 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 número
T3 := SUMMARIZE T2 GROUP BY () ADD MAX(NUMERO) AS número
RDO := (T2 JOIN T3 JOIN SOCIO) [codsocio,nombre,dirección]
```

## 3.2 Álgebra relacional

(las películas de los géneros que le gustan al socio 194) menos (las películas que ya ha tomado prestadas)

```
T1 := ((GUSTA_GENERO WHERE codsocio = '194') JOIN PELICULA) [codpeli,título]
T2 := ((PRESTAMO WHERE codsocio = '194')
  JOIN CINTA JOIN PELICULA) [codpeli,título]
RDO := (T1 MINUS T2) [codpeli,título]
```

## 3.2 Cálculo relacional

Películas de algún género que le gusta al socio 194 ( $\exists$  GUST\_GENX) tales que para toda cinta ( $\forall$  CINX) de esas películas (IF), no existe ningún préstamo (NOT  $\exists$  PRESX) de ellas a ese socio

```
PELIX.codpeli,PELIX.título WHERE ∃ GUST_GENX
  (GUST_GENX.codsocio = '194' AND PELIX.género = GUST_GENX.género AND
  ∀ CINX (IF CINX.codpeli = PELIX.codpeli
  THEN NOT ∃ PRESX (PRESX.codsocio = '194' AND
  PRESX.codcinta = CINX.codcinta)))
```

## 3.3 Álgebra 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,teléfono]
```

## 3.3 Cálculo relacional

Socios tales que para todas y cada una de las películas ( $\forall$  PELIX) de Ariadna Gil (IF), han tenido al menos un préstamo ( $\exists$  PRESX) de alguna de sus copias ( $\exists$  CINX).

```
SOCX.codsocio,SOCX.nombre,SOCX.teléfono WHERE ∀ PELIX
  (IF PELIX.primer_actor = 'Ariadna Gil' OR
  PELIX.segundo_actor = 'Ariadna Gil'
  THEN ∃ PRESX ∃ CINX (PRESX.codcinta = CINX.codcinta AND
  CINX.codpeli = PELIX.codpeli AND PRESX.codsocio = SOCX.codsocio))
```

## 3.4 Álgebra relacional

```
T1 := (PELICULA JOIN LISTA_ESPERA) [codpeli,título]
RDO := (T1 JOIN CINTA JOIN
  (PRESTAMO WHERE SYSDATE-fecha>3 AND pres_dev = 'prestada')
  JOIN SOCIO) [codsocio,nombre,título]
```

## 3.4 Cálculo relacional

```
SOCX.codsocio,SOCX.nombre,PELIX.título WHERE ∃ PRESX ∃ CINX
  (PRESX.codsocio = SOCX.codsocio AND PRESX.codcinta = CINX.codcinta
  AND CINX.codpeli = PELIX.codpeli AND PRESX.pres_dev = 'prestada'
  AND SYSDATE-PRESX.fecha > 3 AND ∃ LISTX (LISTX.codpeli = PELIX.codpeli))
```

## 3.5 Álgebra relacional

```
T1 := (PELICULA WHERE primer_actor = 'Jorge Sanz' OR
  segundo_actor = 'Jorge Sanz') [codpeli]
T2 := (LISTA_ESPERA JOIN T1 JOIN SOCIO) [codsocio,nombre,teléfono]
T3 := ((PRESTAMO WHERE pres_dev = 'prestada') JOIN
  CINTA JOIN T1 JOIN SOCIO) [codsocio,nombre,teléfono]
RDO := T2 UNION T3
```

## 3.5 Cálculo relacional

```
SOCX.codsocio,SOCX.nombre,SOCX.teléfono WHERE ∃ PELIX
  ((PELIX.primer_actor = 'Jorge Sanz' OR
  PELIX.segundo_actor = 'Jorge Sanz') AND
  (∃ LISTX (LISTX.codpeli = PELIX.codpeli AND
  LISTX.codsocio = SOCX.codsocio)
  OR
  ∃ PRESX ∃ CINX (PRESX.codcinta = CINX.codcinta AND
  PRESX.codsocio = SOCX.codsocio AND
  CINX.codpeli = PELIX.codpeli AND
  PRESX.pres_dev = 'prestada'))))
```

**Esquema 4**

## 4.1 Álgebra relacional

```
T1 := (PELICULA JOIN REPARTO) [actor,director,codpeli]
RDO := SUMMARIZE T1 GROUP BY (actor,director) ADD COUNT(*) AS NUM_PELIS
```

## 4.2 Álgebra relacional

(todos los socios) menos (los 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]
```

## 4.2 Cálculo relacional

Socios tales que todos y cada uno ( $\forall$ PRESX) de sus (IF) préstamos son de películas ( $\exists$ CINX  $\exists$ PELIX) que no han sido dirigidas por Vicente Aranda.

```
SOCX.codsocio,SOCX.nombre WHERE  $\forall$  PRESX
(IF PRESX.codsocio = SOCX.codsocio
 THEN  $\exists$  CINX  $\exists$  PELIX (PRESX.codcinta = CINX.codcinta AND
 CINX.codpeli = PELIX.codpeli AND PELIX.director <> 'Vicente Aranda'))
```

## 4.3 Álgebra relacional

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

```
REPARTO DIVIDEBY (PELICULA WHERE director = 'Fernando Trueba') [codpeli]
```

## 4.3 Cálculo relacional

Actores tales que para todas y cada una de las películas ( $\forall$  PELIX) dirigidas por Fernando Trueba (IF), han intervenido en ellas (están en el reparto) ( $\exists$  REPY).

```
REPX.actor WHERE  $\forall$  PELIX
(IF PELIX.director = 'Fernando Trueba'
 THEN  $\exists$  REPY (REPY.actor = REPX.actor AND REPY.codpeli = PELIX.codpeli))
```

## 4.4 Álgebra 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,dirección,teléfono]
```

## 4.5 Álgebra 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
```

## 4.5

Un socio sale en el resultado con un género si ha visto todas las películas de ese género del año 95.

## 4.6

Obtiene los títulos de las películas de las que hay copias en todos los idiomas (teniendo en cuenta sólo los idiomas de los que hay películas en el vídeo club).

## Esquema 5

## 5.1.

```
T1 := (OBRA_MAESTRA JOIN PELICULA) WHERE género = '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 Fernández Gómez participaba en el reparto como actor, pero que sí han tomado prestada alguna película dirigida por él.

## 5.3.

```
SOCX.nombre WHERE  $\exists$  OMX  $\exists$  PELIX  $\exists$  CINX  $\exists$  PRESX
(PELIX.director = 'Luis Buñuel' AND PELIX.codpeli = OMX.codpeli AND
 OMX.codpeli = CINX.codpeli AND CINX.codcinta = PRESX.codcinta AND
 PRESX.codsocio = SOCX.codsocio)
```

## 5.4.

La expresión 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_préstamos
T3 := SUMMARIZE T2 GROUP BY (actor) ADD MAX(num_préstamos) AS num_préstamos
RDO := (T2 JOIN T3 JOIN PELICULA) [actor,título]
```

## 5.6.

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

## 5.7.

```
REPX.actor WHERE  $\forall$  OMX  $\exists$  PELIX
(IF OMX.codpeli = PELIX.codpeli AND
 PELIX.director = 'Luis Buñuel'
 THEN  $\exists$  REPY (REPY.actor = REPX.actor AND
 REPY.codpeli = PELIX.codpeli))
```

## 5.8.

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

## Esquema 6

## 6.1

```
T1 := SUMMARIZE (PREMIO WHERE año = '1994')
      GROUP BY (codpeli) ADD COUNT(*) AS premios
T2 := SUMMARIZE T1 GROUP BY () ADD MAX(premlios) 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 que sólo han tomado prestadas películas de nacionalidad española.

6.3

```

PELIX.título WHERE
  ∃ PREM (PREM.PREMIO LIKE 'Goya%' AND PREM.año = '1993' AND
    PREM.codpeli = PELIX.codpeli)
  AND NOT ∃ PRESX ∃ CINCX (PRESX.codcinta = CINCX.codcinta AND
    CINCX.codpeli = PELIX.codpeli AND PRESX.fecha >= '1-MAR-00')

```

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

Si como mucho hay una copia en búlgaro por película:

```

SOCX.nombre WHERE ∀ PELIX ∃ CINCX
  (IF PELIX.codpeli = CINCX.codpeli AND CINCX.IDIOMA = 'Búlgaro'
    THEN ∃ PRESX (PRESX.codcinta = CINCX.codcinta AND
      PRESX.codsocio = SOCX.codsocio))
  AND NOT ∃ PRESX ∃ CINCX (PRESX.codsocio = SOCX.codsocio AND
    PRESX.codcinta = CINCX.codcinta AND CINCX.IDIOMA <> 'Búlgaro')

```

Si puede haber varias copias en búlgaro por película:

```

SOCX.nombre WHERE ∀ PELIX ∃ CINCX
  (IF (PELIX.codpeli = CINCX.codpeli AND CINCX.IDIOMA = 'Búlgaro')
    THEN ∃ PRESX ∃ CINY (CINY.IDIOMA = 'Búlgaro' AND
      PRESX.codcinta = CINY.codcinta AND CINY.codpeli = CINCX.codpeli AND
      PRESX.codsocio = SOCX.codsocio))
  AND NOT ∃ PRESX ∃ CINCX (PRESX.codsocio = SOCX.codsocio AND
    PRESX.codcinta = CINCX.codcinta AND CINCX.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 género='terror') [codpeli]
T1 := SUMMARIZE (T0 JOIN CINTA JOIN PRESTAMO)
  GROUP BY (codpeli) ADD COUNT(*) AS préstamos
T2 := (T1 WHERE préstamos > 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,título]

```

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 DIRX IS PELIX.director
```

```

DIRX WHERE ∀PELIX
  (IF PELIX.director=DIRX.director THEN ∃CINCX ∃CINY
    ( CINCX.codpeli=PELIX.codpeli AND CINCX.IDIOMA='francés' AND
      CINY.codpeli=PELIX.codpeli AND CINY.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,título]
T1 := SUMMARIZE T0 GROUP BY (director,título) ADD COUNT(*) AS numsocios
T2 := SUMMARIZE T1 GROUP BY (director) ADD MAX(numsocios) AS numsocios
RDO := (T1 JOIN T2) [director,título]

```

6.14

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

6.15

```

PELIX.director,PELIX.título WHERE ∃ PELIY
  (PELIY.título = 'Mars Attacks'
  AND ∀ PREMY (IF PREMY.codpeli = PELIY.codpeli
    THEN ∃ PREM (PELIX.codpeli = PREM.codpeli AND
      PREMY.premio = PREM.premio) )

```

6.16

Socios que no tienen prestada ninguna película premiada.

**Esquema 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

```
SOCX.cod socio,SOCX.nombre,SOCX.teléfono WHERE
  ∃ PELIX ∃ CINX ∃ PRESX
  (PELIX.director = 'Paco Smith' AND PELICULA.año<1992
   AND PELIX.codpeli = CINX.codpeli AND CINX.codcinta = PRESX.codcinta
   AND PRESX.cod socio = SOCX.cod socio)
AND ∀ PELIX (IF (PELIX.director = 'Paco Smith' AND PELIX.año>=1992)
  THEN ∃ CINX NOT ∃ PRESX (PELIX.codpeli = CINX.codpeli AND
  CINX.codcinta = PRESX.codcinta AND
  PRESX.cod socio = SOCX.cod socio))
```

7.4

```
PELIX.director,REPX.actor WHERE
  (PELIX.codpeli = REPX.codpeli AND
  ∀ PELIY (IF PELIY.director = PELIX.director AND
  ∃ REPY (REPY.actor = REPX.actor AND
  REPY.codpeli = PELIY.codpeli)
  THEN ∃ PREMX (PREMX.codpeli = PELIY.codpeli AND
  PREMX.actor = REPX.actor)))
```

7.5 Álgebra relacional

```
T1 := ( PELICULA JOIN PREMIO[codpeli,premio] ) [director,premio]
T2 := PREMIO [premio]
RDO := T1 DIVIDEBY T2
```

7.5 Cálculo relacional

```
RANGE OF DIRX IS PELIX.director
RANGE OF TIPOX IS PREMX.premio
```

```
DIRX WHERE ∀ TIPOX ∃ PELIX ∃ PREMX
  ( DIRX.director = PELIX.director AND PELIX.codpeli = PREMX.codpeli AND
  PREMX.premio = TIPOX.premio )
```

O bien,

```
PELIX.director WHERE ∀ PREMX ∃ PELIY ∃ PREMY
  ( PELIX.director = PELIY.director AND PELIY.codpeli = PREMY.codpeli AND
  PREMY.premio = PREMX.premio )
```

7.6 Álgebra relacional

```
T1 := PELICULA [director,género]
T2 := SUMMARIZE T1 GROUP BY (director) ADD COUNT(*) AS num_géneros
T3 := (T2 WHERE num_géneros = 1) [director]
T4 := (PELICULA JOIN PREMIO[codpeli]) [director]
RDO := T3 MINUS T4
```

T1 := directores y géneros de los que tienen películas  
 T2 := directores y número de géneros distintos de los que tienen películas  
 T3 := directores que siempre hacen películas de un mismo género  
 T4 := directores cuyas películas han ganado algún premio  
 RDO := directores que siempre han hecho películas de un mismo género y que no han ganado ningún premio

7.6 Cálculo relacional

```
PELIX.director WHERE ∀ PELIY
  ( IF PELIY.director = PELIX.director
  THEN (PELIY.género = PELIX.género AND
  NOT ∃ PERMIOX (PREMX.codpeli = PELIY.codpeli)))
```

7.7 Álgebra relacional

```
T1 := (PELICULA JOIN (CINTA WHERE idioma='español')) [codpeli,título,director]
T2 := (PELICULA [codpeli,título,director]) MINUS T1
T3 := (PELICULA JOIN CINTA JOIN PRESTAMO) [codpeli,título,director]
RDO := T2 MINUS T3
```

T1 := películas de las que hay copias en español  
 T2 := películas de las que sólo hay copias en idiomas extranjeros  
 T3 := películas que han sido prestadas alguna vez  
 RDO := películas de las que sólo hay copias en idiomas extranjeros y que nunca han sido prestadas

7.7 Cálculo relacional

```
PELIX.codpeli,PELIX.título,PELIX.director WHERE ∀ CINX
  (IF CINX.codpeli = PELIX.codpeli
  THEN CINX.IDIOMA <> 'español' AND NOT ∃ PRESX (PRESX.codcinta=CINX.codcinta))
```

7.8 Álgebra relacional

```
SUMMARIZE PREMIO GROUP BY (PREMIO) ADD MIN(fecha) AS fecha
```

7.8 Cálculo relacional

```
PREMX.PREMIO,PREMX.fecha WHERE
  ∀ PREMY (IF PREMY.PREMIO = PREMX.PREMIO THEN PREMY.fecha >= PREMX.fecha)
```

**Esquema 8**

8.1 Álgebra relacional

```
T1 := ACTOR JOIN REPARTO JOIN PELICULA
T2 := (ACTOR TIMES PELICULA) WHERE ACTOR.nacionalidad <> PELICULA.nacionalidad
T3 := T2 JOIN REPARTO
RDO := T1[codactor,nombre] MINUS T3[codactor,nombre]
```

T1 := actores que han hecho alguna película de su misma nacionalidad, junto con la película  
 T2 := parejas de actores y películas que son de distinta nacionalidad  
 T3 := actores que han hecho alguna película de una nacionalidad distinta de la suya, junto con la película

## 8.1 Cálculo relacional

```

ACTORX.nombre WHERE ∀ REPX ∃ PELIX
(IF REPX.codactor=ACTORX.codactor
 THEN REPX.codpeli=PELIX.codpeli AND PELIX.nacionalidad=ACTORX.nacionalidad)

```

## 8.2 Álgebra relacional

(actores premiados ) menos (actores premiados en películas de su misma nacionalidad)

```

T1 := PREMIO[codactor]
T2 := (ACTOR JOIN REPARTO JOIN PELICULA JOIN
PREMIO[codpeli,codactor])[codactor]
RDO := (( T1 MINUS T2 ) JOIN ACTOR)[nombre]

```

T1 := actores premiados  
T2 := actores que han hecho alguna película de su misma nacionalidad por la que han recibido algún premio

## 8.2 Cálculo relacional

```

RANGE OF ACT_PREMIADOX IS
  ACTORX WHERE ∃PREMX (PREMX.codactor = ACTORX.codactor)

```

```

ACT_PREMIADOX.nombre WHERE ∀REPX ∃PELIX
( IF REPX.codactor = ACT_PREMIADOX.codactor AND
  PELIX.codpeli = REPX.codpeli AND
  PELIX.nacionalidad = ACT_PREMIADOX.nacionalidad
 THEN NOT ∃PREMX ( PREMX.codpeli = PELIX.codpeli AND
  PREMX.codactor = ACT_PREMIADOX.codactor ) )

```

## 8.3 Álgebra relacional

```

T1 := ACTOR[codactor,año_nacimiento]
      TIMES
      DIRECTOR[coddirector,año_nacimiento]
T2 := T1 WHERE ACTOR.año_nacimiento < DIRECTOR.año_nacimiento
T3 := PELICULA JOIN T2 JOIN REPARTO
RDO := ((REPARTO[codactor] MINUS T3[codactor]) JOIN ACTOR)[nombre]

```

T1 := parejas de actores y directores  
T2 := parejas de actores y directores donde el director es más joven que el actor  
T3 := parejas de actores y directores que han trabajado juntos, siendo el director más joven que el actor

## 8.3 Cálculo relacional

```

ACTORX.nombre WHERE ∃ REPX
(IF REPX.codactor = ACTORX.codactor
 THEN ∃ DIRX ∃ PELIX
  (PELIX.codpeli = REPX.codpeli AND PELIX.coddirector = DIRX.coddirector
  AND DIRX.año_nacimiento <= ACTORX.año_nacimiento))

```

## 8.4

```

T1 := REPARTO JOIN PREMIO[codpeli,premio]
T2 := SUMMARIZE T1 GROUP BY (codpeli,codactor) ADD COUNT(*) AS premios
T3 := SUMMARIZE T2 GROUP BY (codactor) ADD MAX(premios) AS premios
RDO := (T2 JOIN T3 JOIN PELICULA[codpeli,título]
  JOIN ACTOR[codactor,nombre])[nombre,título]

```

## 8.5

```

DIRX.nombre WHERE ∃ PELIX
(PELIX.coddirector=DIRX.coddirector AND PELIX.año-DIRX.año_nacimiento <= 25)

```

## Examen de Junio de 1999

## (a.1)

```

T1 := PREMIO JOIN ACTOR
T2 := SUMMARIZE T1 GROUP BY () ADD MIN(año-año_nacimiento) AS edad
T3 := (T1 TIMES T2) WHERE año-año_nacimiento = edad
RDO := T3 JOIN ACTOR

```

## (a.2)

```

T1 := PRESTAMO JOIN CINTA JOIN PELICULA
T2 := (T1 WHERE nacionalidad = 'española' AND año < 1980) [codsocio]
T3 := (T1 WHERE nacionalidad <> 'española' OR año >= 1980) [codsocio]
RDO := (T2 MINUS T3) JOIN SOCIO

```

## (b.1)

```

SOCX WHERE ∀ PELIX ∃ DIRX
(IF PELIX.año = TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-1 AND
 TO_NUMBER(TO_CHAR(SYSDATE,'YYYY'))-DIRX.año_nac < 30
 THEN ∃ PRESX ∃ CINUX
  (PRESX.codsocio = SOCX.codsocio AND PRESX.codcinta = CINUX.codcinta AND
  CINUX.codpeli = PELIX.codpeli))

```

## (b.2)

```

SOCX WHERE ∀ PRESX
(IF PRESX.codsocio = SOCX.codsocio
 THEN ∃ CINUX ∃ PELIX
  (CINUX.codcinta = PRESX.codcinta AND PELIX.codpeli = CINUX.codpeli AND
  PELIX.nacionalidad = 'española' AND PELIX.año < 1980))

```

## Examen de Septiembre de 1999

## (a.1)

```

T1 := SOCIO JOIN PRESTAMO JOIN CINTA JOIN PELICULA
T2 := (T1 WHERE año >= TO_CHAR(fecha_alta,'YYYY')) [codsocio]
T3 := (T1 WHERE año < TO_CHAR(fecha_alta,'YYYY')) [codsocio]
RDO := (T2 MINUS T3) JOIN SOCIO

```

## (a.2)

```

T1 := (REPARTO JOIN ACTOR) [codpeli,nacionalidad]
T2 := SUMMARIZE T1 GROUP BY (codpeli) ADD COUNT(*) AS número
T3 := T2 WHERE número >=2
RDO:= T3 JOIN PELICULA

```

(b.1)

```
SOCX WHERE  $\forall$  PRESX
(IF PRESX.codsocio=SOCX.codsocio
 THEN  $\exists$  PELIX  $\exists$  CINX (PELIX.codpeli=CINX.codpeli AND
      CINX.codcinta=PRESX.codcinta AND PRESX.codsocio=SOCX.codsocio AND
      TO_CHAR(SOCX.fecha_alta,'YYYY')<=PELIX.año)
```

(b.2)

```
PELIX WHERE  $\exists$  ACTOR1X  $\exists$  ACTOR2X  $\exists$  REPARTO1X  $\exists$  REPARTO2X
(PELIX.codpeli=REPARTO1X.codpeli AND REPARTO1X.codactor=ACTOR1X.codactor AND
 PELIX.codpeli=REPARTO2X.codpeli AND REPARTO2X.codactor=ACTOR2X.codactor AND
 ACTOR1X.nacionalidad<>ACTOR2X.nacionalidad)
```