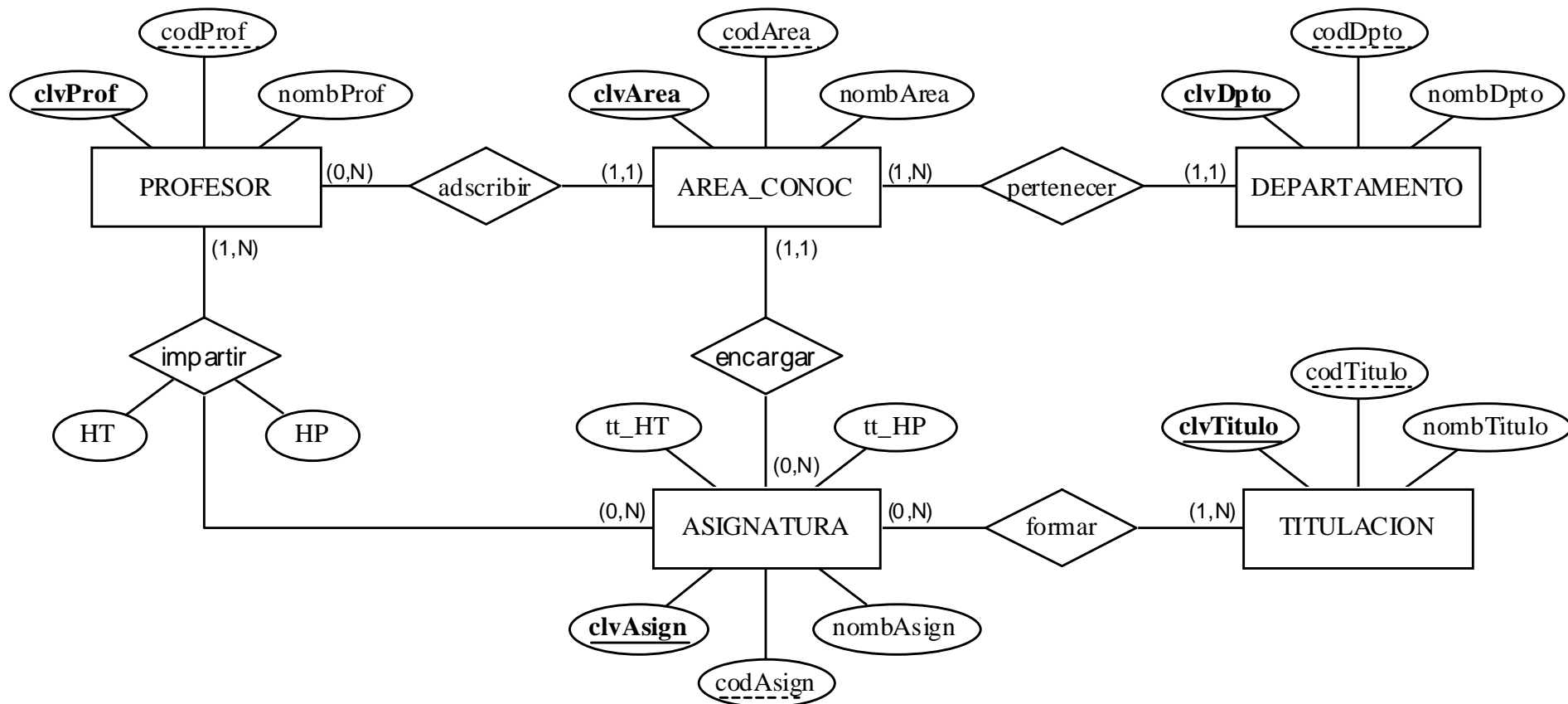


esquema E/R de la B.D. Universidad



esquema relacional de la B.D. Universidad

esquema relacional simplificado de la B.D. (se han omitido algunas restricciones de integridad y los dominios)

```
Departamento(clvDpto: entero; codDpto: tpCadena; nombDpto: tpCadena);

AreaConoc    (clvArea: entero; codArea: tpCadena; nombArea: tpCadena; clvDpto: entero;
              (clvDpto) es clave ajena de Departamento y no nulo)

Profesor     (clvProf: entero; codProf: tpCadena; nombProf: tpCadena; clvArea: entero;
              (clvArea) es clave ajena de AreaConoc y no nulo)

Asignatura   (clvAsign: entero; codAsign: tpCadena; nombAsign: tpCadena;
              tt_HT: entero; tt_HP: entero; clvArea: entero;
              (clvArea) es clave ajena de AreaConoc y no nulo)

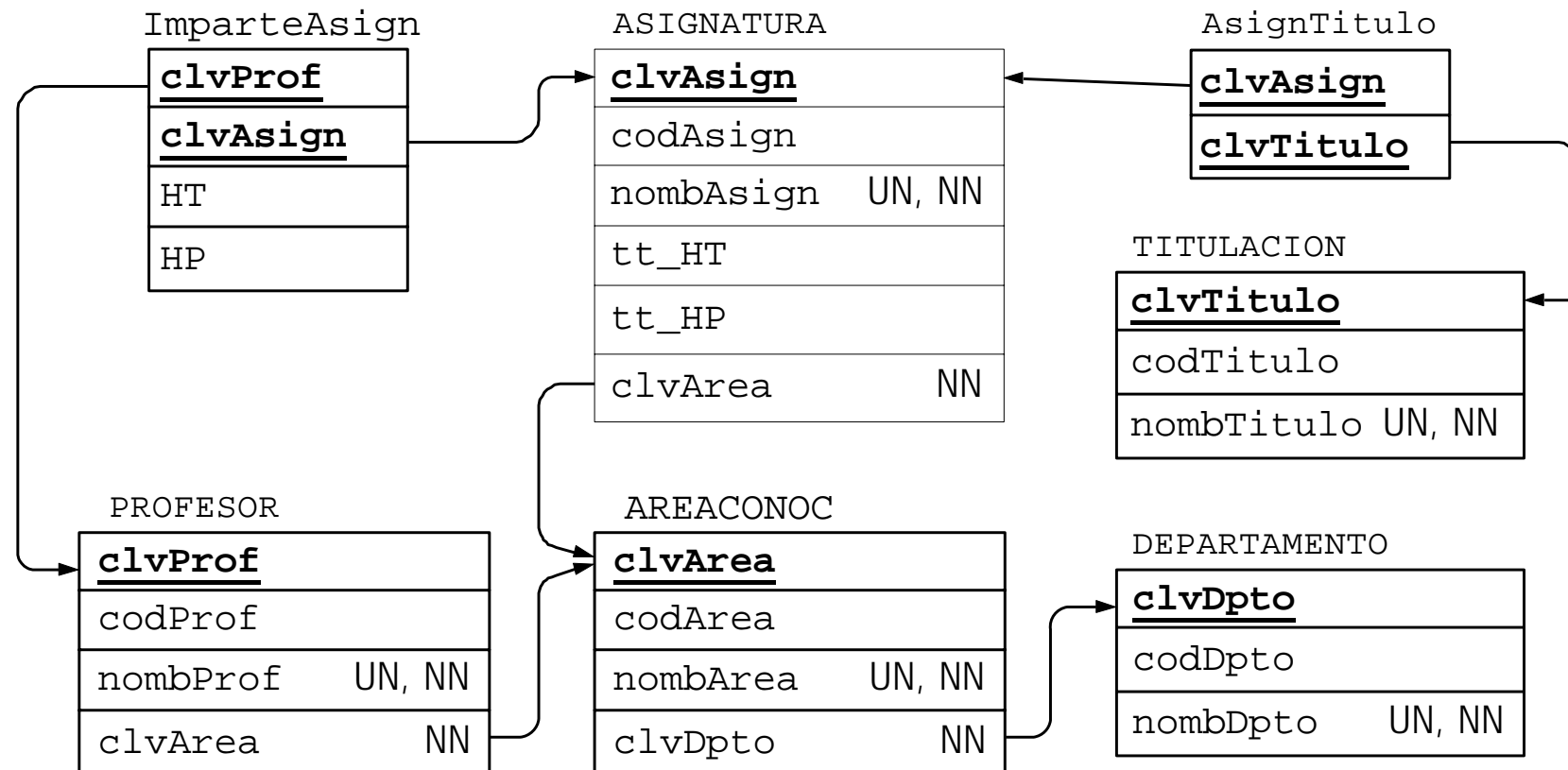
Titulacion   (clvTitulo: entero; codTitulo: tpCadena; nombTitulo: tpCadena);

ImparteAsign(clvProf: entero; clvAsign: entero; HT: entero; HP: entero);
              (clvProf) es clave ajena de Profesor;
              (clvAsign) es clave ajena de Asignatura)

AsignTitulo  (clvAsign: entero; clvTitulo: entero;
              (clvAsign) es clave ajena de Asignatura;
              (clvTitulo) es clave ajena de Titulación)
```

----- significa clave alternativa \equiv UNICO y NO NULO

diagrama relacional de la B.D. Universidad



ejercicios con la B.D. Universidad

- 1 Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área. **(álgebra y SQL)**
- 2 listado de los pares de asignaturas (clvAsign1, clvAsign2) que están asignadas a exactamente las mismas titulaciones. No deberán aparecer parejas repetidas.
- 3 listado de los profesores del departamento de Informática (clave, nombre y total de horas impartido) que imparten menos de 200 horas
- 4 Contratación del profesor 'Luis Pérez' para impartir la docencia de la asignatura 23. El profesor se identificará por el número 357 (codProf='inf054'), y estará asignado al área de la asignatura. Se eliminará esta docencia a los profesores que la impartían.

consultas con álgebra relacional a la B.D. Universidad (ej1)

1a *Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área.*

$R1 = \prod_{clvArea, clvAsign} (Asignatura) \equiv$ *asignaturas encargadas a cada área*

$R2 = \prod_{clvProf, clvAsign} (ImparteAsign) \equiv$ *profesores que imparten las asignaturas*

$R3 = R1 \times \prod_{clvProf} (Profesor) \equiv$ *todas las posibles imparticiones de las asignaturas, junto con el área a que están asignadas*

$R4 = \prod_{clvArea, clvAsign, clvProf} (R1 \bowtie R2) \equiv$ *imparticiones reales de las asignaturas, junto con el área a que están asignadas*

$R5 = R3 - R4 \equiv$ *profesores con asignaturas no impartidas y área a que están asignadas*

$R6 = \prod_{clvProf, clvArea} (R5) \equiv$ *profesores y áreas en las que hay alguna asignatura encargada que no es impartida por dicho profesor*

$R7 = \prod_{clvProf, clvArea} (R4) \equiv$ *profesores y áreas en las que imparten docencia*

$R8 = R7 - R6 \equiv$ *profesores y áreas en que imparte docencia en todas las asignaturas encargadas*

$$R = \prod_{clvProf} (R8)$$

consultas con álgebra relacional a la B.D. Universidad (ej1)

1b

Puesto que asignatura determina el área de conocimiento a que está asignada, no es necesario construir las ternas (clvProf, clvAsign, clArea)

$$R9 = (\prod_{clvProf} (Profesor) \times \prod_{clvAsign} (Asignatura)) - \prod_{clvProf, clvAsign} (ImparteAsign)$$

\equiv profesores y asignaturas que no imparten

$$R6 = \prod_{clvProf, clvArea} (R9 \bowtie Asignatura) \equiv \text{profesores y áreas en las que el profesor no imparte alguna asignatura encargada al área}$$

$$R7 = \prod_{clvProf, clvArea} (ImparteAsign \bowtie Asignatura) \equiv \text{profesores y áreas en que imparten docencia}$$

$$R = \prod_{clvProf} (R7 - R6)$$

consultas con álgebra relacional a la B.D. Universidad (ej1)

1c

También se pueden contar las asignaturas encargadas a cada área, y las asignaturas de cada área que imparte un profesor . . .

$R10 (clvArea, totAsign) = \mathbf{AGRUPAR}_{contar}(Asignatura; clvArea)$
 $\equiv \text{áreas y total de asignaturas encargadas}$

$R11 = \prod_{clvProf, clvAsign, clvArea} (\text{ImparteAsign} \bowtie \text{Asignatura}) \equiv \text{imparticiones junto con el área de la asignatura}$

$R12 (clvProf, clvArea, totAsign) = \mathbf{AGRUPAR}_{contar}(R11; clvProf, clvArea)$
 $\equiv \text{profesores y áreas en que imparten docencia, junto con el total de asignaturas impartidas en dicha área}$

$$R = \prod_{clvProf} (R12 \bowtie R10)$$

consultas en SQL a la B.D. Universidad (ej1)

- 1a) *Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área.*

```
SELECT DISTINCT clvProf
FROM Profesor P
WHERE EXISTS (SELECT * FROM AreaConoc AC
              WHERE clvArea IN (SELECT clvArea FROM Asignatura)
              AND NOT EXISTS
              (SELECT * FROM Asignatura A
              WHERE A.clvArea = AC.clvArea
              AND NOT EXISTS (SELECT * FROM ImparteAsign IA
                              WHERE P.clvProf = IA.clvProf AND
                              A.clvAsign = IA.clvAsign))));
```

*areas con alguna
asignatura encargada*

*asignaturas encargadas al área,
no impartidas por el profesor*

consultas en SQL a la B.D. Universidad (ej1)

- 1b** *Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área.*

```
SELECT DISTINCT clvProf
FROM Profesor P
WHERE EXISTS (SELECT * FROM Asignatura AC
              WHERE NOT EXISTS
              (SELECT * FROM Asignatura A
               WHERE A.clvArea = AC.clvArea
                AND NOT EXISTS (SELECT * FROM ImparteAsign IA
                               WHERE P.clvProf = IA.clvProf AND
                                       A.clvAsign = IA.clvAsign))));
```

areas con alguna asignatura encargada

asignaturas encargadas al área, no impartidas por el profesor

consultas en SQL a la B.D. Universidad (ej1)

- 1c) *Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área.*

```
SELECT DISTINCT clvProf, AC.clvArea
FROM Profesor P, Asignatura AC
WHERE NOT EXISTS (SELECT * FROM Asignatura A
                  WHERE A.clvArea = AC.clvArea
                  AND NOT EXISTS (SELECT * FROM ImparteAsign IA
                                  WHERE P.clvProf = IA.clvProf AND
                                         A.clvAsign = IA.clvAsign));
```

*areas con alguna
asignatura encargada*

*asignaturas encargadas al área,
no impartidas por el profesor*

consultas en SQL a la B.D. Universidad (ej1)

1d *Listado de los profesores que imparten docencia en al menos todas las asignaturas que tiene encargadas algún área de conocimiento. Sugerencia: obtener los pares (clvProf, clvArea) correspondientes a los profesores que no imparten docencia en alguna asignatura de esa área.*

```
SELECT DISTINCT clvProf, AC.clvArea
FROM Profesor P, Asignatura AC
WHERE NOT EXISTS
```

*areas con alguna
asignatura encargada*

```
(SELECT clvAsign FROM Asignatura A WHERE A.clvArea = AC.clvArea
MINUS
SELECT clvAsign FROM ImparteAsign IA WHERE P.clvProf = IA.clvProf);
```

*asignaturas encargadas al área,
no impartidas por el profesor*

consultas en SQL a la B.D. Universidad (ej1)

1e) *Creando dos vistas auxiliares en las que se cuentan las asignaturas: por área y por prof.-área*

```
CREATE VIEW totAsignArea (clvArea, totAsign) AS
  SELECT clvArea, count(*) FROM Asignatura GROUP BY clvArea;
```

```
CREATE VIEW profTotAsignArea (clvProf, clvArea, totAsign) AS
  SELECT clvProf, clvArea, count(*) FROM ImparteAsign IA, Asignatura A
  WHERE IA.clvAsign = A.clvAsign GROUP BY clvProf, clvArea;
```

```
SELECT DISTINCT clvProf, AC.clvArea
FROM profTotAsignArea P, totAsignArea AC
WHERE P.clvArea = AC.clvArea AND P.totAsign = AC.totAsign;
```

```
DROP VIEW totAsignArea;
DROP VIEW profTotAsignArea;
```

*eliminar las
vistas creadas*

consultas en SQL a la B.D. Universidad (ej2)

listado de los pares de asignaturas (clvAsign1, clvAsign2) que están asignadas a exactamente las mismas titulaciones. No deberán aparecer parejas repetidas.

```
SELECT DISTINCT A1.clvAsign, A2.clvAsign
FROM Asignatura A1, Asignatura A2
WHERE A1.clvAsign < A2.clvAsign AND
      NOT EXISTS( (SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A1.clvAsign
                  MINUS
                  SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A2.clvAsign)
                UNION
                (SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A2.clvAsign
                  MINUS
                  SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A1.clvAsign));
```

consultas en SQL a la B.D. Universidad (ej2)

listado de los pares de asignaturas (clvAsign1, clvAsign2) que están asignadas a exactamente las mismas titulaciones. No deberán aparecer parejas repetidas.

```
SELECT DISTINCT A1.clvAsign, A2.clvAsign
FROM Asignatura A1, Asignatura A2
WHERE A1.clvAsign < A2.clvAsign AND
      NOT EXISTS(SELECT clvTitulo FROM asignTitulo
                 WHERE clvAsign = A1.clvAsign OR clvAsign = A2.clvAsign
                 MINUS
                 SELECT AT1.clvTitulo FROM asignTitulo AT1, asignTitulo AT2
                 WHERE AT1.clvAsign = A1.clvAsign AND AT2.clvAsign = A2.clvAsign AND
                       AT1.clvTitulo = AT2.clvTitulo);
```

consultas en SQL a la B.D. Universidad (ej2)

listado de los pares de asignaturas (clvAsign1, clvAsign2) que están asignadas a exactamente las mismas titulaciones. No deberán aparecer parejas repetidas.

```
SELECT DISTINCT A1.clvAsign, A2.clvAsign
FROM Asignatura A1, Asignatura A2
WHERE A1.clvAsign < A2.clvAsign AND
      NOT EXISTS(SELECT clvTitulo FROM asignTitulo
                 WHERE clvAsign = A1.clvAsign OR clvAsign = A2.clvAsign
                 MINUS (
                 SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A1.clvAsign
                 INTERSECT
                 SELECT clvTitulo FROM asignTitulo WHERE clvAsign = A2.clvAsign));
```

(ej3 a)

listado de los profesores del departamento de Informática (clave, nombre y total de horas impartido) que imparten menos de 200 horas

```
SELECT P.clvProf, nombProf, sum(ht)+sum(hp) "tot.horas"
FROM Profesor P, ImparteAsign IA, AreaConoc AC, Departamento D
WHERE P.clvArea = AC.clvArea AND AC.clvDpto = D.clvDpto
AND D.nombDpto = 'Informática' AND P.clvProf = IA.clvProf (+)
GROUP BY P.clvProf, nombProf
HAVING sum(ht)+sum(hp) < 200 OR sum(ht)+sum(hp) IS NULL;
```

```
CREATE VIEW horasProf (clvProf, totHoras) AS
SELECT clvProf, sum(ht+hp)
FROM ImparteAsign
GROUP BY clvProf
HAVING sum(ht)+sum(hp) < 200
UNION
SELECT clvProf, 0
FROM profesor
WHERE clvProf NOT IN (SELECT clvProf FROM ImparteAsign);

SELECT P.clvProf, nombProf, totHoras
FROM Profesor P, horasProf H, AreaConoc AC, Departamento D
WHERE P.clvArea = AC.clvArea AND AC.clvDpto = D.clvDpto
AND D.nombDpto = 'Informática' AND P.clvProf = H.clvProf AND H.totHoras < 200;

DROP VIEW horasProf;
```


(ej3 b)

listado de los profesores del departamento de Informática (clave, nombre y total de horas impartido) que imparten menos de 200 horas

```
SELECT P.clvProf, nombProf, sum(ht)+sum(hp) "tot.horas"  
FROM Profesor P, ImparteAsign IA, AreaConoc AC, Departamento D  
WHERE P.clvArea = AC.clvArea AND AC.clvDpto = D.clvDpto AND  
D.nombDpto = 'Informática' AND P.clvProf = IA.clvProf  
GROUP BY P.clvProf, nombProf  
HAVING sum(ht)+sum(hp) < 200  
UNION  
SELECT P.clvProf, nombProf, 0  
FROM Profesor P, ImparteAsign IA, AreaConoc AC, Departamento D  
WHERE P.clvArea = AC.clvArea AND AC.clvDpto = D.clvDpto AND D.nombDpto = 'Informática'  
AND P.clvProf NOT IN (SELECT clvProf FROM ImparteAsign);
```

(ej4 a) *Contratación del profesor 'Luis Pérez' para impartir la docencia de la asignatura 23. El profesor se identificará por el número 357 (codProf='inf054'), y estará asignado al área de la asignatura. Se eliminará esta docencia a los profesores que la impartían.*

```
INSERT INTO Profesor
  SELECT 357, 'inf054', 'Luis Pérez', clvArea FROM Asignatura WHERE clvAsign = 23;
DELETE FROM ImparteAsign WHERE clvAsign = 23;
INSERT INTO ImparteAsign
  SELECT 357, 23, tt_HT, tt_HP FROM Asignatura WHERE clvAsign = 23;
```

/ Se utilizan instrucciones de inserción de tuplas con los datos conocidos y, después, se añaden instrucciones de actualización. Para ello, hay que desposibilitar temporalmente la restricción NOT NULL de clvArea de Profesor. Además de eliminar y, posteriormente, añadir esta restricción, también se puede hacer:*

```
ALTER TABLE Profesor DISABLE CONSTRAINT areaProf_NN;
. . .
ALTER TABLE Profesor ENABLE CONSTRAINT areaProf_NN;
```

En vez de eliminar explícitamente la restricción, se podría haber hecho directamente:

```
ALTER TABLE Profesor MODIFY clvArea NULL;
*/
SELECT OWNER, CONSTRAINT_NAME, CONSTRAINT_TYPE, TABLE_NAME, SEARCH_CONDITION, STATUS
FROM USER_CONSTRAINTS WHERE TABLE_NAME = 'PROFESOR';
```

(ej4 b) *Contratación del profesor 'Luis Pérez' para impartir la docencia de la asignatura 23. El profesor se identificará por el número 357 (codProf='inf054'), y estará asignado al área de la asignatura. Se eliminará esta docencia a los profesores que la impartían.*

```
ALTER TABLE Profesor DROP CONSTRAINT areaProf_NN;

INSERT INTO Profesor VALUES (357, 'inf054', 'LUIS PEREZ', null);

UPDATE Profesor
  SET clvArea = (SELECT clvArea FROM Asignatura WHERE clvAsign = 23)
  WHERE clvProf = 357;

ALTER TABLE Profesor MODIFY clvArea CONSTRAINT areaProf_NN NOT NULL;

DELETE FROM ImparteAsign WHERE clvAsign=23;

INSERT INTO ImparteAsign VALUES (357, 23, null, null);

UPDATE ImparteAsign
  SET HT = (SELECT tt_HT FROM Asignatura WHERE clvAsign = 23),
      HP = (SELECT tt_HP FROM Asignatura WHERE clvAsign = 23)
  WHERE clvProf = 357 AND clvAsign = 23;
```

(ej4 c) *Contratación del profesor 'Luis Pérez' para impartir la docencia de la asignatura 23. El profesor se identificará por el número 357 (codProf='inf054'), y estará asignado al área de la asignatura. Se eliminará esta docencia a los profesores que la impartían.*

/ La diferencia con versiones anteriores es que la docencia que se asigna al nuevo profesor es la que impartían los profesores que lo sustituían*

**/*

```
ALTER TABLE Profesor DISABLE CONSTRAINT areaProf_NN;
```

```
INSERT INTO Profesor VALUES (357, 'inf054', 'LUIS PEREZ', null);
```

```
UPDATE Profesor
```

```
  SET clvArea = (SELECT clvArea FROM Asignatura WHERE clvAsign = 23)
```

```
  WHERE clvProf = 357;
```

```
ALTER TABLE Profesor ENABLE CONSTRAINT areaProf_NN;
```

```
INSERT INTO ImparteAsign
```

```
  SELECT 357, 23, sum(HT), sum(HP)
```

```
  FROM ImparteAsign
```

```
  WHERE clvAsign = 23;
```

```
DELETE FROM ImparteAsign WHERE clvAsign=23 AND clvProf <> 357;
```