

Bases de Datos y Sistemas de Información

Grado en Ingeniería Informática

Unidad Didáctica 2: El lenguaje SQL: manipulación de
datos

Parte 2: Ejercicios (Práctica 1)

(Doc. UD2.2)

Curso 2018/2019



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Índice

- 1 Introducción1
- 2 Presentación de la base de datos CINE 2
- 3 Ejercicios sobre la base de datos Cine.....3
 - 3.1 Consultas sobre una sola tabla 3
 - 3.2 Consultas sobre varias tablas..... 4
 - 3.3 Consultas con subconsultas 5
 - 3.4 Consultas universalmente cuantificadas7
 - 3.5 Consultas agrupadas..... 9
 - 3.6 Consultas con concatenación 10
 - 3.7 Consultas conjuntistas 12
 - 3.8 Consultas generales 12
- 4 Presentación de la base de datos MÚSICA..... 13
- 5 Ejercicios sobre la base de datos MÚSICA..... 15
 - 5.1 Consultas sobre una sola relación15
 - 5.2 Consultas sobre varias relaciones..... 16
 - 5.3 Consultas con subconsultas 17
 - 5.4 Consultas con cuantificación universal17
 - 5.5 Consultas agrupadas..... 18
 - 5.6 Consultas generales 19
- 6 Presentación de la base de datos BIBLIOTECA 20
- 7 Ejercicios sobre la base de datos BIBLIOTECA 22
 - 7.1 Consultas sobre una sola relación22
 - 7.2 Consultas sobre varias relaciones..... 23
 - 7.3 Consultas con subconsultas 24
 - 7.4 Consultas con cuantificación universal.....25
 - 7.5 Consultas agrupadas..... 25
 - 7.6 Consultas generales 27
- 8 Presentación de la base de datos DEPARTAMENTO 28
- 9 Ejercicios sobre la base de datos DEPARTAMENTO 30
 - 9.1 Consultas generales 30
- 10 Presentación de la base de datos CICLISMO 41
- 11 Ejercicios sobre la base de datos CICLISMO 43
 - 11.1 Consultas sobre una sola relación 43
 - 11.2 Consultas sobre varias relaciones..... 44
 - 11.3 Consultas con subconsultas 45

11.4 Consultas con cuantificación universal.....46

11.5 Consultas agrupadas.....48

11.6 Consultas generales.....49

1 INTRODUCCIÓN

El objetivo de esta primera práctica es aprender a realizar consultas en lenguaje SQL. Para ello vamos a utilizar la herramienta *SQL Developer* de ORACLE.

El lenguaje de manipulación de datos del SQL de ORACLE es prácticamente estándar SQL/92 (ver el documento UD2.1). En particular, en esta práctica vamos a utilizar únicamente la sentencia de consulta SELECT.

Después de la presentación de cada una de las bases de datos con las que se va a trabajar¹, se proponen una serie de consultas que se pueden realizar con el SQL de ORACLE. Estas consultas pueden ser de distintos tipos:

- **Consultas sobre una sola relación:** éstas son las consultas más sencillas ya que para resolverlas sólo es necesario utilizar una relación de la base de datos.
- **Consultas sobre varias relaciones:** consultas que se pueden resolver especificando varias relaciones en la cláusula FROM de la sentencia SELECT. La conexión entre estas relaciones se establece en la cláusula WHERE con las comparaciones oportunas.
- **Consultas con subconsultas:** consultas que se pueden resolver con una o varias subconsultas en la cláusula WHERE.
- **Consultas con cuantificación universal:** estas consultas que tendrían una solución natural con un cuantificador universal. Dado que el SQL de ORACLE no lo proporciona, la solución exige la representación de la cuantificación universal en términos de negación y de la cuantificación existencial. En esencia la transformación es la siguiente: “Todo elemento E del conjunto C cumple la propiedad P” es equivalente a “No existe un elemento E del conjunto C que no cumpla la propiedad P”. Así pues se propone buscar soluciones a estas consultas utilizando el predicado NOT EXISTS (...). Alternativamente se pueden resolver con el operador conjuntista UNION.
- **Consultas agrupadas:** consultas cuya solución utiliza la cláusula GROUP BY.
- **Consultas conjuntistas:** consultas que se pueden resolver con los operadores conjuntistas.
- **Consultas con concatenación:** consultas que se pueden resolver son el JOIN.
- **Consultas generales:** consultas de cualquier tipo.

Es importante destacar que una consulta puede admitir varias soluciones por lo que podría haberse incluido en varios apartados. Después de cada consulta se ha incluido el resultado de la misma en forma tabular, si el resultado que obtenido con la solución propuesta no coincide con el que se incluye aquí, la consulta está mal resuelta; si sí que coincide “podría” estar bien. Cuando el resultado de la consulta contiene muchas filas, no se han incluido todas, los puntos suspensivos (...) en medio de la tabla indican que faltan filas, sin embargo, sí que se ha puesto al final de la consulta cuántas filas devuelve (sólo cuando devuelve más de 6).

IMPORTANTE: MANEJO DE FECHAS

- Para no tener problemas con los atributos de tipo fecha cuando los valores posibles de un atributo abarquen distintos siglos, es importante que el formato de las fechas en el *SQL Developer* tenga el año con cuatro dígitos, para ello, en: Herramientas >> Preferencias >> Bases de Datos>>NLS, hay que poner el Formato de Fecha = DD/MM/RRRR.
- La función EXTRACT permite obtener el día, mes o año de un atributo de tipo fecha, así, si suponemos que X='02/06/1965':
 - EXTRACT (DAY FROM X) devuelve 2;
 - EXTRACT (MONTH FROM X) devuelve 6;
 - EXTRACT (YEAR FROM X) devuelve 1965;

¹ Algunos SGBD no aceptan caracteres especiales, como p. ej. letras acentuadas o el carácter “ñ”, en los nombres de atributos o relaciones. Para evitar problemas hemos decidido no acentuar ninguna palabra que figure en la definición de una relación.

2 PRESENTACIÓN DE LA BASE DE DATOS CINE

Se desea almacenar información sobre películas referente a qué actores han actuado en cada película, quién la ha dirigido, de qué géneros están clasificadas, en qué libro está basado el guion, etc.; para ello se ha diseñado una base de datos relacional cuyo esquema se muestra a continuación²:

CS_PAIS(cod_pais:char(5), nombre:char(20))

CP:{cod_pais}

VNN:{nombre}

CS_ACTOR(cod_act:char(5), nombre:char(70), fecha_nac:date, cod_pais:char(5))

CP:{cod_act}

VNN:{nombre, fecha_nac, cod_pais}

CAj:{cod_pais} → CS_Pais(cod_pais)

CS_LIBRO(cod_lib:char(5), titulo:char(70), anyo:number, autor:char(80))

CP:{cod_lib}

VNN:{titulo, autor}

CS_PELICULA(cod_peli:char(5), titulo:char(70), anyo:number, duracion:number,
cod_lib:char(5), director:char(70))

CP:{cod_peli}

VNN:{titulo, duracion}

CAj:{cod_lib} → CS_Libro(cod_lib)

CS_GENERO(cod_gen:char(5), nombre:char(30))

CP:{cod_gen}

CS_ACTUA(cod_act:char(5), cod_peli:char(5), papel:char(10))

CP:{cod_act, cod_peli}

VNN:{papel}

CAj:{cod_peli} → CS_Pelicula(cod_peli)

CAj:{cod_act} → CS_Actor(cod_act)

CS_CLASIFICACION(cod_gen:char(5), cod_peli:char(5))

CP:{cod_gen, cod_peli}

CAj:{cod_peli} → CS_Pelicula(cod_peli)

CAj:{cod_gen} → CS_Genero(cod_gen)

Para aclarar por completo el esquema, a continuación, se explica el significado de cada atributo en cada relación:

- **CS_Pais:**

- *cod_pais*: código del país.
- *nombre*: nombre del país.

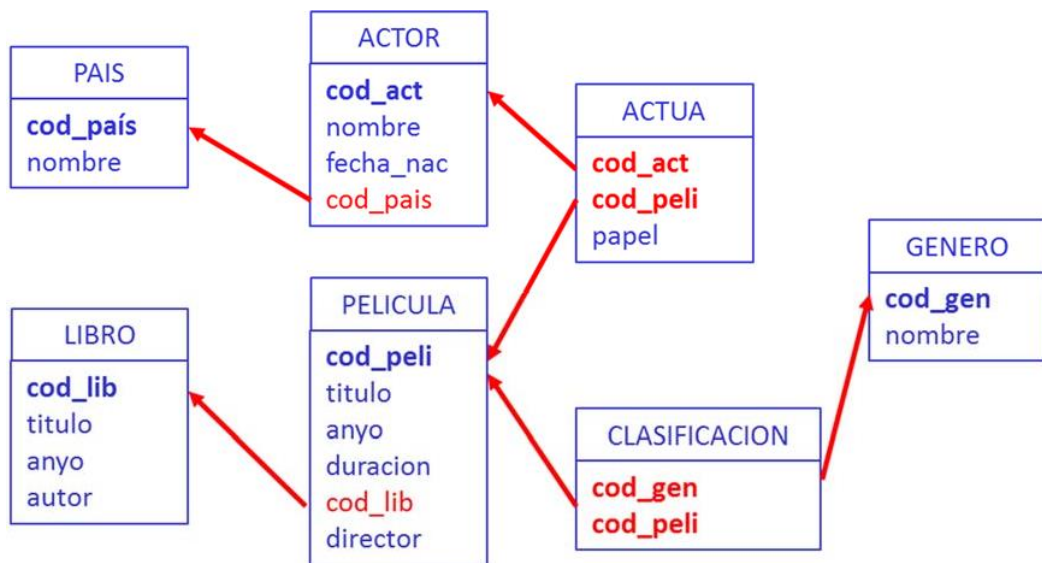
- **CS_Actor:**

- *cod_act*: código del actor.
- *nombre*: nombre del actor.
- *fecha_nac*: día de nacimiento del actor.
- *cod_pais*: código del país de origen del actor.

² Esta base de datos se diseñó para un concurso de SQL que preparamos los profesores de la asignatura, ése es el motivo del prefijo CS_ en el nombre de todas las relaciones.

- **CS_Libro:**
 - *cod_lib*: código del libro.
 - *titulo*: título del libro.
 - *anyo*: año de publicación del libro.
 - *autor*: nombre del autor del libro.
- **CS_Pelicula:**
 - *cod_peli*: código de la película.
 - *titulo*: título de la película.
 - *anyo*: año de publicación del estreno de la película.
 - *duracion*: cuántos minutos dura la película.
 - *cod_lib*: código del libro en el que está basada la película.
 - *director*: nombre del director de la película.
- **CS_Genero:**
 - *cod_gen*: código del género.
 - *nombre*: nombre del género.
- **CS_Actua:** el actor de código *cod_act* ha actuado en la película de código *cod_peli* en un papel *papel*.
- **CS_Clasificacion:** la película de código *cod_peli* está clasificada en el género de código *cod_gen*.

Gráficamente el esquema relacional es el siguiente:



3 EJERCICIOS SOBRE LA BASE DE DATOS CINE

3.1 Consultas sobre una sola tabla

1. Obtener ordenados ascendentemente los códigos de los países de donde son los actores.

```

COD_P
-----
ad63
gg74
hg45
nb12
rt89
sd53
  
```

```
sf15
ty11
we74
zf58
10 filas seleccionadas
```

```
select distinct cod_pais
from cs_actor
order by cod_pais;
```

2. Obtener el código y el título de las películas de año anterior a 1970 que no estén basadas en ningún libro ordenadas por título.

```
COD_P TITULO
```

```
-----
357L Cleopatra
365N Cortina rasgada
332D Dos hombres y un destino
```

```
Select distinct cod_peli, titulo
From cs_pelicula
Where anyo<'1970' and cod_lib is null
Order by cod_peli;
```

3. Obtener el código y el nombre de los actores cuyo nombre incluye "John".

```
COD_A NOMBRE
```

```
-----
A62 John Goodman
```

```
Select distinct cod_act, nombre
From cs_actor
Where nombre like '%John%';
```

4. Obtener el código y el título de las películas de más de 120 minutos de la década de los 80.

```
COD_P TITULO
```

```
-----
365A Indiana Jones y la última cruzada
```

```
Select cod_peli, titulo
From cs_pelicula
Where duracion>120 and anyo between 1980 and 1989;
```

5. Obtener el código y el título de las películas que estén basadas en algún libro y cuyo director se apellide 'Pakula'.

```
COD_P TITULO
```

```
-----
856A El informe pelícano
```

```
Select cod_peli, titulo
From cs_pelicula
Where cod_lib is not null and director like '% Pakula';
```

6. ¿Cuántas películas hay de más de 120 minutos de la década de los 80?

```
COUNT (*)
```

```
-----
1
```

```
Select Count(*)
From cs_pelicula
where duracion>120 and anyo between 1980 and 1989;
```

7. ¿Cuántas películas se han clasificado de los géneros de código 'BB5' o 'GG4' o 'JH6'.

CUÁNTAS_PELIS

```
-----
43
```

```
Select Count(DISTINCT cod_peli) CUÁNTAS_PELIS
From cs_clasificacion
where cod_gen = 'BB5' or cod_gen = 'GG4' or cod_gen = 'JH6';
```

8. ¿De qué año es el libro más antiguo?

AÑO

```
----
1877
```

```
Select MIN(anyo)
From cs_libro;
```

9. ¿Cuál es la duración media de las películas del año 1987?

DURACIÓN_MEDIA

```
-----
119,5
```

```
Select avg(duracion)
From cs_pelicula
where anyo = '1987';
```

10. ¿Cuántos minutos ocupan todas las películas dirigidas por 'Steven Spielberg'?

DURAN_MIN

```
-----
296
```

```
Select sum(duracion)
From cs_pelicula
where director = 'Steven Spielberg';
```

3.2 Consultas sobre varias tablas

11. Obtener el código y el título de las películas en las que actúa un actor con el mismo nombre que el director de la película (ordenadas por título).

COD_P TITULO

```
-----
654J Buenas noches, y buena suerte
778E Sin perdón

455K The monuments men
118E Un mundo perfecto
```

```
Select p.cod_peli,p.titulo
From cs_pelicula p, cs_actor a, cs_actua t
where p.director = a.nombre and p.cod_peli = t.cod_peli and
a.cod_act = t.cod_act
order by p.titulo;
```


12. Obtener el código y el título de las películas clasificadas del género de nombre 'Comedia' (ordenadas por título).

COD_P TITULO

```

258S Cuando Harry encontró a Sally
369F Desayuno con diamantes
456G El chip prodigioso
888T El golpe
548J Jamón, Jamón
147D Los búfalos de Durham
874G Los picapiedra
789B The mexican
8 filas seleccionadas

```

```

select p.cod_peli, p.titulo
from cs_pelicula p, cs_genero g, cs_clasificacion c
where g.nombre='Comedia' and
      g.cod_gen = c.cod_gen and
      p.cod_peli = c.cod_peli
order by p.titulo;

```

13. Obtener el código y el título de las películas basadas en algún libro anterior a 1950.

COD_P TITULO

```

159A Ana Karenina
123V Anna Karenina
159X Anna Karenina
123N Lo que el viento se llevó
123S My Fair Lady
258M Un tranvía llamado deseo
6 filas seleccionadas

```

```

select p.cod_peli, p.titulo
from cs_pelicula p, cs_libro l
where l.cod_lib = p.COD_LIB and l.anyo<1950
order by p.titulo;

```

14. Obtener el código y el nombre de los países de los actores que actúan en películas clasificadas del género de nombre 'Comedia' (ordenados por nombre).

COD_P NOMBRE

```

ad63 Bélgica
we74 España
sf15 USA
3 filas seleccionadas

```

```

Select distinct p.cod_pais, p.nombre
from cs_pais p, cs_actor a, cs_actua t, cs_clasificacion c, cs_genero g
where g.nombre = 'Comedia' and
      g.cod_gen = c.cod_gen and
      c.cod_peli = t.cod_peli and
      t.cod_act = a.cod_act and
      a.cod_pais = p.cod_pais
order by p.nombre;

```

3.3 Consultas con subconsultas

15. Resolver los ejercicios 11, 12, 13 y 14 con subconsultas. Las soluciones son exactamente las mismas.

Ejercicio 11:

```
select p.cod_peli, p.titulo
from cs_pelicula p
where p.director IN (select a.nombre
                     from cs_actor a
                     where a.cod_act IN (select c.cod_act
                                         from cs_actua c
                                         where c.cod_peli = p.cod_peli)
                     ) order by p.titulo;
```

Ejercicio 12:

```
select p.cod_peli, p.titulo
from cs_pelicula p
where p.cod_peli IN (select c.cod_peli
                    from cs_clasificacion c
                    where c.cod_gen IN (select g.cod_gen
                                       from cs_genero g
                                       where g.nombre = 'Comedia')
                    ) order by p.titulo;
```

Ejercicio 13:

```
select distinct p.cod_peli, p.titulo
from cs_pelicula p
where p.cod_lib IN (select cod_lib
                  from cs_libro
                  where anyo < 1950)
order by p.titulo;
```

Ejercicio 14:

```
select cod_pais, nombre
from cs_pais
where cod_pais IN (select cod_pais
                  from cs_actor
                  where cod_act IN (select cod_act
                                    from cs_actua
                                    where cod_peli IN (select cod_peli
                                                        from cs_clasificacion
                                                        where cod_gen IN (select cod_gen
                                                                    from cs_genero
                                                                    )
                                                        )
                                    )
                  )
```

```

        where nombre = 'Comedia')
    )
)
)order by nombre;

```

16. Obtener el código y el nombre de los actores nacidos antes de 1950 que actúan con un papel 'Principal' en alguna película (ordenados por nombre).

COD_A NOMBRE

```

-----
Z15  Al Pacino
D49  Audrey Hepburn
L54  Christopher Plummer
L59  Clint Eastwood
L45  Elizabeth Taylor
S56  Elke Sommer
J47  Gene Hackman
V88  George Peppard
J45  Harrison Ford
X45  Julie Andrews
J56  Marlon Brandon
D14  Martin Sheen
U88  Morgan Freeman
W34  Paul Newman
T44  Rex Harrison
F56  Richard Burton
M45  Richard Gere
E56  Robert de Niro
H45  Robert Redford
W32  Sean Connery
E45  Susan Sarandon
D01  Vivien Leigh
22 filas seleccionadas

```

```

select cod_act, nombre
from cs_actor
where fecha_nac < '01/01/1950'
      and cod_act in (select cod_act
                      from cs_actua
                      where papel = 'Principal')
order by nombre;

```

17. Obtener el código, el título y el autor de los libros en los que se ha basado alguna película de la década de los 90 (ordenados por título).

COD_L TITULO

AUTOR

```

-----
GJ7  Ana Karenina                      Leon Tolstoi
GJ6  El informe pelícano                John Grisham
UU4  El padrino                        Mario Puzo
DF6  Entrevista con el vampiro          Anne Rice
LP9  Rita Hayworth y la redención de Shawshank Stephen King
AR3  Vida de este chico                 Tobias Wolff
6 filas seleccionadas

```

```

select cod_lib, titulo, autor
from cs_libro
where cod_lib in (select cod_lib
                  from cs_pelicula
                  where anyo between 1990 and 1999)
order by titulo;

```

18. Obtener el código, el título y el autor de los libros en los que no se haya basado ninguna película.

COD_L TITULO

AUTOR

FA6 La caída de los gigantes Ken Follet

```
select cod_lib, titulo, autor
from cs_libro
where cod_lib not in (select cod_lib
                      from cs_pelicula
                      where cod_lib is not null)
order by titulo;
```

19. Obtener el nombre del género o géneros a los que pertenecen películas en las que no actúa ningún actor (ordenados por nombre).

NOMBRE

Animación
Aventuras
Drama

```
select nombre
from cs_genero
where cod_gen in (select cod_gen
                  from cs_clasificacion
                  where cod_peli not in (select cod_peli
                                        from cs_actua
                                        where cod_act is not null))
order by nombre;
```

20. Obtener el título de los libros en los que se haya basado alguna película en la que no hayan actuado actores del país de nombre 'USA' (ordenados por título).

TITULO

Ana Karenina
Lo que el viento se llevó
Pigmalion
The sound of music

```
select titulo
from cs_libro
where cod_lib in (select cod_lib
                  from cs_pelicula
                  where cod_lib is not null
                  and cod_peli not in (select cod_peli
                                      from cs_actua
                                      where cod_act in (select cod_act
                                                        from cs_actor
                                                        where cod_pais in (select cod_pais
                                                                              from cs_pais
                                                                              where nombre = 'USA')))))
order by titulo;
```

21. ¿Cuántas películas hay clasificadas del género de nombre 'Comedia' y en las que sólo aparece un actor con el papel 'Secundario'?

COUNT(P.COD_PELI)

2

```
select count(p.cod_peli)
from cs_pelicula p
where cod_peli in (select cod_peli
                  from cs_clasificacion
                  where cod_gen in (select cod_gen
                                    from cs_genero
                                    where nombre='Comedia'))
and 1 = (select count(cod_act)
        from cs_actua
        where papel = 'Secundario'
        and cod_peli = p.cod_peli);
```

22. Obtener el año de la primera película en la que el actor de nombre 'Jude Law' tuvo un papel como 'Principal'.

ANYO

2001

```
select min(anyo)
from cs_pelicula
where cod_peli in (select cod_peli
                   from cs_actua
                   where papel = 'Principal'
                   and cod_act in (select cod_act
                                   from cs_actor
                                   where nombre = 'Jude Law'));
```

23. Obtener el código y el nombre de actor o actores más viejos.

COD_A NOMBRE

K58 Stanley Holloway

```
select cod_act, nombre
from cs_actor
where fecha_nac <= all (select fecha_nac
                       from cs_actor);
```

24. Obtener el código, el nombre y la fecha de nacimiento del actor más viejo nacido en el año 1940.

COD_A NOMBRE

FECHA_NAC

C89 James Caan 26/03/1940

```
select cod_act, nombre, fecha_nac
from cs_actor
where fecha_nac <= all (select fecha_nac
                       from cs_actor
                       where fecha_nac >= '01/01/1940')
and fecha_nac >= '01/01/1940';
```

25. Obtener el nombre del género (o de los géneros) en los que se ha clasificado la película más larga.

NOMBRE

Bélica
Drama
Romance

```
select nombre
from cs_genero
where cod_gen in (select cod_gen
                  from cs_clasificacion
                  where cod_peli in (select cod_peli
                                      from cs_pelicula
                                      where duracion >= all(select duracion
                                                            from cs_pelicula)))
order by nombre;
```

26. Obtener el código y el título de los libros en los que se han basado películas en las que actúan actores del país de nombre España (ordenados por título).

COD_L TITULO

ZF4 Come, reza, ama
PP4 Desayuno en Tiffanys
DF6 Entrevista con el vampiro

```

select cod_lib, titulo
from cs_libro
where cod_lib in (select cod_lib
                  from cs_pelicula
                  where cod_lib is not null
                     and cod_peli in (select cod_peli
                                      from cs_actua
                                      where cod_act in (select cod_act
                                                         from cs_actor
                                                         where cod_pais in (select cod_pais
                                                                                   from cs_pais
                                                                                   where nombre =
                                                                                       'España'))))
order by titulo;

```

27. Obtener el título de las películas anteriores a 1950 clasificadas en más de un género (ordenadas por título).

TITULO

 Lo que el viento se llevó

```

select titulo
from cs_pelicula p
where anyo < 1950
      and 1 < (select count(cod_gen)
              from cs_clasificacion
              where cod_peli = p.cod_peli)
order by titulo;

```

28. Obtener la cantidad de películas en las que han participado menos de 4 actores.

COUNT(*)

 68

```

select count(*)
from cs_pelicula p
where 4 > (select count(cod_act)
          from cs_actua
          where cod_peli = p.cod_peli);

```

29. Obtener los directores que han dirigido más de 250 minutos entre todas sus películas.

DIRECTOR

 Steven Soderbergh
 Clint Eastwood
 Steven Spielberg
 Francis Ford Coppola
 Guy Ritchie

```

select distinct director
from cs_pelicula p
where 250 < (select sum(duracion)
            from cs_pelicula
            where director = p.director);

```

30. Obtener el año o años en el que nacieron más de 3 actores.

AÑO

 1954
 1940

```

select distinct extract(year from fecha_nac) AÑO
from cs_actor
group by extract(year from fecha_nac)
having count(extract(year from fecha_nac)) > 3
order by extract(year from fecha_nac) desc;

```

31. Obtener el código y nombre del actor más joven que ha participado en una película clasificada del género de código 'DD8'.

```
COD_A NOMBRE
-----
S47    Kevin Costner
```

```
select cod_act, nombre
from cs_actor
where fecha_nac = (select max(fecha_nac)
                   from cs_actor
                   where cod_act in (select cod_act
                                     from cs_actua
                                     where cod_peli in (select cod_peli
                                                         from cs_clasificacion
                                                         where cod_gen = 'DD8')));
```

3.4 Consultas universalmente cuantificadas

32. Obtener el código y el nombre de los países con actores y tales que todos los actores de ese país han nacido en el siglo XX (ordenados por nombre).

```
COD_P NOMBRE
-----
hg45  Alemania
zf58  Australia
rt89  Austria
ad63  Bélgica
gg74  Canadá
nb12  Cuba
we74  España

sd53  Francia
sf15  USA
9 filas seleccionadas
```

```
select cod_pais, nombre
from cs_pais p
where not exists (select *
                  from cs_actor
                  where cod_pais = p.cod_pais
                     and extract(year from fecha_nac) not between 1900 and 2000)
and exists (select *
            from cs_actor
            where cod_pais = p.cod_pais)
order by nombre;
```

33. Obtener el código y el nombre de los actores tales que todos los papeles que han tenido son de 'Secundario'. Sólo interesan aquellos actores que hayan actuado en alguna película.

```
COD_A NOMBRE
-----
E22    Diane Keaton
C89    James Caan
F77    José L. de Villalonga
Q47    Ludwig Donath
C15    Robert Duvall
K58    Stanley Holloway
6 filas seleccionadas
```

```
select cod_pais, nombre
from cs_actor a
where not exists (select *
                  from cs_actua
                  where cod_act = a.cod_act
                     and papel <> 'Secundario'
                     and cod_peli in (select cod_peli
                                       from cs_pelicula))
and exists (select *
            from cs_actua
            where cod_act = a.cod_act
               and cod_peli in (select cod_peli
                                from cs_pelicula))
order by nombre;
```

34. Obtener el código y el nombre de los actores que han aparecido en todas las películas del director 'Guy Ritchie' (sólo si ha dirigido al menos una).

COD_A NOMBRE

A47 Robert Downey Jr.
A52 Jude Law

```
select cod_act, nombre
from cs_actor a
where not exists (select *
                  from cs_pelicula p
                  where director = 'Guy Ritchie'
                  and not exists (select *
                                  from cs_actua
                                  where cod_peli = p.cod_peli
                                  and cod_act = a.cod_act))
and exists (select *
            from cs_pelicula
            where director = 'Guy Ritchie');
```

35. Resolver la consulta anterior pero para el director de nombre 'John Steel'.

no se ha seleccionado ninguna fila

```
select cod_act, nombre
from cs_actor a
where not exists (select *
                  from cs_pelicula p
                  where director = 'John Steel'
                  and not exists (select *
                                  from cs_actua
                                  where cod_peli = p.cod_peli
                                  and cod_act =
a.cod_act))
and exists (select *
            from cs_pelicula
            where director = 'John Steel');
```

36. Obtener el código y el título de las películas de menos de 100 minutos en las que todos los actores que han actuado son de un mismo país.

COD_P TITULO

258S Cuando Harry encontró a Sally
548J Jamón, Jamón
654J Buenas noches, y buena suerte
874G Los picapiedra
951D Al caer el sol

```
select cod_peli, titulo
from cs_pelicula p
where duracion < 100
and cod_peli in (select cod_peli
                 from cs_actua a
                 where cod_act in (select cod_act
                                     from cs_actor i
                                     where not exists (select *
                                                         from cs_actor t
                                                         where t.cod_pais <> i.cod_pais
                                                         and cod_act in (select cod_act
                                                                           from cs_actua g
                                                                           where g.cod_peli =
a.cod_peli))))
order by cod_peli;
```


| COD_P | TITULO | ANYO |
|------------------------|---------------------------------|------|
| 159X | Anna Karenina | 1948 |
| 159D | Bajo sospecha | 2000 |
| 357L | Cleopatra | 1963 |
| 365N | Cortina rasgada | 1966 |
| 369F | Desayuno con diamantes | 1961 |
| 332D | Dos hombres y un destino | 1969 |
| 888T | El golpe | 1973 |
| 144H | El premio | 1963 |
| 753N | La gata sobre el tejado de zinc | 1958 |
| 123N | Lo que el viento se llevó | 1939 |
| 123S | My Fair Lady | 1964 |
| 778E | Sin perdón | 1992 |
| 589B | Sonrisas y lágrimas | 1965 |
| 258M | Un tranvía llamado deseo | 1951 |
| 14 filas seleccionadas | | |

38. Obtener el código y el nombre de cada país si se cumple que todos sus actores han actuado en al menos una película de más de 120 minutos. (Ordenados por nombre).

```
select cod_pais, nombre  
from cs_pais p  
where not exists (select *  
                  from cs_actor  
                  where cod_pais = p.cod_pais  
                      and cod_act not in (select cod_act  
                                          from cs_actua a  
                                          where cod_peli in (select cod_peli  
                                                              from cs_película  
                                                              where duracion>120)))  
  
and exists (select *  
            from cs_actor  
            where cod_pais = p.cod_pais  
                and cod_act in (select cod_act  
                                from cs_actua  
                                where cod_peli in (select cod_peli  
                                                    from cs_película))))  
order by nombre;
```

3.5 Consultas agrupadas

39. Obtener el código y el título del libro o libros en que se ha basado más de una película, indicando cuántas películas se han hecho sobre él.

| COD_L | TITULO | CUÁNTAS |
|-------|--------------|---------|
| UU4 | El padrino | 3 |
| GJ7 | Ana Karenina | 3 |

```
select l.cod_lib, l.titulo, count (p.cod_lib) CUÁNTAS
from cs_libro l, cs_pelicula p
where l.cod_lib = p.cod_lib
group by l.cod_lib, l.titulo
having count(p.cod_lib) >1;
```

40. Obtener para cada género en el que se han clasificado más de 5 películas, el código y el nombre del género indicando la cantidad de películas del mismo y duración media de sus películas. (Ordenados por nombre). (La función ROUND redondea al entero más cercano).

| COD_G | NOMBRE | CUÁNTAS | DUR_MEDI |
|-----------------------|-----------|---------|----------|
| DR5 | Acción | 8 | 138 |
| DF2 | Biografía | 6 | 146 |
| JJ9 | Comedia | 8 | 110 |
| GG4 | Crimen | 18 | 132 |
| BB5 | Drama | 38 | 134 |
| KK4 | Misterio | 6 | 127 |
| HH2 | Romance | 8 | 127 |
| 7 filas seleccionadas | | | |

```
select g.cod_gen, g.nombre, count(c.cod_gen) CUÁNTAS,
round(avg(p.duracion)) DUR_MEDIA
from cs_genero g, cs_clasificacion c, cs_pelicula p
where g.cod_gen = c.cod_gen and c.cod_peli = p.cod_peli
group by g.cod_gen, g.nombre
having count(c.cod_peli)>5
order by g.nombre;
```

41. Obtener el código y el título de las películas de año posterior al 2000 junto con el número de géneros en que están clasificadas, si es que están en alguno. (Ordenadas por título).

| COD_P | TITULO | CUÁNTOS |
|------------------------|-----------------------------------|---------|
| 159A | Ana Karenina | 1 |
| 654J | Buenas noches, y buena suerte | 2 |
| 145K | Camino a la perdición | 3 |
| 465H | El código da Vinci | 1 |
| 158S | Enemigo a las puertas | 3 |
| 369J | Golpe de efecto | 2 |
| 457P | Invictus | 3 |
| 159U | Mi novio es un ladrón | 1 |
| 326F | Mystic river | 3 |
| 189G | Ocean's Thirteen | 2 |
| 658G | Sherlock Holmes | 3 |
| 452W | Sherlock Holmes: Juego de sombras | 3 |
| 789B | The mexican | 3 |
| 455K | The monuments men | 3 |
| 14 filas seleccionadas | | |

```
select p.cod_peli, p.titulo, count(c.cod_gen) CUÁNTOS
from cs_pelicula p, cs_clasificacion c
where p.anyo > 2000 and p.cod_peli = c.cod_peli
group by p.cod_peli, p.titulo
having count(c.cod_gen)>0
order by p.titulo;
```

42. Obtener los directores que tienen la cadena 'George' en su nombre y que han dirigido exactamente dos películas.

DIRECTOR

George Roy Hill
George Clooney

```
select director
from cs_pelicula
where director like '%George%'
group by director
having count(cod_peli)=2;
```

43. Obtener para cada película clasificada exactamente en un género y en la que haya actuado algún actor, el código, el título y la cantidad de actores que actúan en ella.

| COD_P | TITULO | CUÁNTOS |
|-----------------------|---------------------------------|---------|
| 159A | Ana Karenina | 2 |
| 159X | Anna Karenina | 1 |
| 365N | Cortina rasgada | 3 |
| 465H | El código da Vinci | 1 |
| 475A | Filadelfia | 3 |
| 753N | La gata sobre el tejado de zinc | 2 |
| 159U | Mi novio es un ladrón | 2 |
| 778E | Sin perdón | 3 |
| 258M | Un tranvía llamado deseo | 2 |
| 9 filas seleccionadas | | |

```
select p.cod_peli, p.titulo, count(a.cod_act) CUÁNTOS
from cs_pelicula p, cs_actua a, cs_clasificacion c
where p.cod_peli = a.cod_peli and p.cod_peli= c.cod_peli
group by p.cod_peli, p.titulo
having count(a.cod_act)>0 and count(distinct c.cod_gen)=1
order by p.titulo;
```

44. Obtener el código y el nombre de todos los países con actores indicando cuántos actores de cada país han actuado en al menos una película de la década de los 60.

| COD_P | NOMBRE | CUÁNTOS |
|-----------------------|----------|---------|
| hg45 | Alemania | 1 |
| rt89 | Austria | 1 |
| ad63 | Bélgica | 1 |
| gg74 | Canadá | 1 |
| we74 | España | 1 |
| ty11 | UK | 4 |
| sf15 | USA | 4 |
| 7 filas seleccionadas | | |

```
select p.cod_pais, p.nombre, count(a.cod_act) CUÁNTOS
from cs_pais p, cs_actor a, cs_actua t, cs_pelicula l
where p.cod_pais = a.cod_pais
      and a.cod_act = t.cod_act
      and t.cod_peli = l.cod_peli
      and l.anyo between '1960' and '1969'
group by p.cod_pais, p.nombre
having count(a.cod_act)>0
```

45. Obtener el código, el nombre del género en el que hay clasificadas más películas (puede haber más de uno).

COD_G NOMBRE

BB5 Drama

```
select g.cod_gen, g.nombre
from cs_genero g, cs_clasificacion c
where g.cod_gen = c.cod_gen
group by g.cod_gen, g.nombre
having count(distinct c.cod_peli)>= ALL(select count(distinct cod_peli)
                                     from cs_clasificacion
                                     group by cod_gen);
```

46. Obtener el código, el título y el autor del libro en el que se han basado más películas (puede haber más de uno).

COD_L TITULO

AUTOR

UU4 El padrino

Mario Puzo

GJ7 Ana Karenina

Leon Tolstoi

```
select l.cod_lib, l.titulo, l.autor
from cs_libro l, cs_pelicula p
where l.cod_lib = p.cod_lib
group by l.cod_lib, l.titulo, l.autor
having count(distinct p.cod_peli)>= ALL(select count(distinct cod_peli)
                                     from cs_pelicula
                                     where cod_lib is not null
                                     group by cod_lib)

order by l.titulo desc;
```

47. Obtener el código y el nombre del país que más actores tiene que hayan participado exactamente en 2 películas.

COD_P NOMBRE

sf15 USA

```
select p.cod_pais, p.nombre
from cs_pais p, cs_actor a, cs_actua t
where p.cod_pais = a.cod_pais
      and a.cod_act = t.cod_act
group by p.cod_pais, p.nombre
having count(a.cod_act)>= ALL(select max(count(cod_act))
                             from cs_actor
                             group by cod_pais);

#ESTA BIEN PERO NO PONE EXACTAMENTE 2#
```

48. Obtener el año o años en el que nacieron más de 3 actores indicando cuántos nacieron exactamente.

AÑO CUÁNTOS

1954 4

1940 4

```
select extract(year from fecha_nac) AÑO, count(distinct cod_act) CUÁNTOS
from cs_actor
group by extract(year from fecha_nac)
having count(distinct cod_act)>3;
```

49. Resolver la consulta 36.

Ver resultado en consulta 36.

```
select p.cod_peli, p.titulo
from cs_pelicula p, cs_actua t, cs_actor a
where p.cod_peli = t.cod_peli and t.cod_act = a.cod_act
and p.duracion < 100
group by p.cod_peli, p.titulo
having count(distinct a.cod_pais)=1
order by p.cod_peli;
```

3.6 Consultas con concatenación

50. Obtener para todos los países que hay en la base de datos, el código, el nombre y la cantidad de actores que hay de ese país.

| COD_P | NOMBRE | CUÁNTOS |
|-------|-----------|---------|
| hg45 | Alemania | 1 |
| zf58 | Australia | 1 |
| rt89 | Austria | 1 |
| ad63 | Bélgica | 1 |
| gg74 | Canadá | 1 |
| nb12 | Cuba | 1 |
| we74 | España | 5 |
| sd53 | Francia | 1 |
| hy76 | Italia | 0 |
| ty11 | UK | 9 |
| sf15 | USA | 38 |

11 filas seleccionadas

```
select cod_pais, a.nombre, count(cod_act) CUÁNTOS
from cs_pais a LEFT JOIN cs_actor using (cod_pais)
group by cod_pais, a.nombre
order by nombre;
```

51. Obtener el código y el título de todos los libros de la base de datos de año posterior a 1980 junto con la cantidad de películas a que han dado lugar.

| COD_L | TITULO | CUÁNTAS |
|-------|-------------------------------------------|---------|
| GJ6 | El informe pelícano | 1 |
| GH4 | El código da Vinci | 1 |
| AR3 | Vida de este chico | 1 |
| AE8 | El color del dinero | 1 |
| FA6 | La caída de los gigantes | 0 |
| LP9 | Rita Hayworth y la redención de Shawshank | 1 |
| KS5 | El factor humano | 1 |
| ZF4 | Come, reza, ama | 1 |

8 filas seleccionadas

```
select cod_lib, l.titulo, count(cod_peli) CUANTAS
from cs_libro l left join cs_pelicula using(cod_lib)
where l.anyo > 1980
group by cod_lib, l.titulo;
```

52. Obtener para todos los países que hay en la base de datos, el código, el nombre y la cantidad de actores que hay de ese país que hayan tenido un papel como "Secundario" en alguna película.

| COD_P | NOMBRE | CUÁNTOS |
|-------|-----------|---------|
| hg45 | Alemania | 0 |
| zf58 | Australia | 0 |
| rt89 | Austria | 1 |
| ad63 | Bélgica | 0 |
| gg74 | Canadá | 0 |
| nb12 | Cuba | 1 |
| we74 | España | 3 |
| sd53 | Francia | 0 |
| hy76 | Italia | 0 |
| ty11 | UK | 4 |
| sf15 | USA | 16 |

11 filas seleccionadas

```
select cod_pais, p.nombre, count(distinct a.cod_act) CUÁNTOS
from cs_pais p LEFT JOIN (cs_actor a
                        JOIN cs_actua t
                        ON papel = 'Secundario'
                        and a.cod_act = t.cod_act) using (cod_pais)
group by cod_pais, p.nombre
order by p.nombre;
```

53. Obtener para cada película que hay en la base de datos que dure más de 140 minutos, el código, el título, la cantidad de géneros en los que está clasificado y la cantidad de actores que han actuado en ella.

| COD_P | TITULO | GEN | ACT |
|-------|----------------------------------|-----|-----|
| 123V | Anna Karenina | 1 | 0 |
| 963L | Apocalypse now | 0 | 4 |
| 666F | Atrápame si puedes | 0 | 2 |
| 438S | Cadena perpetua | 2 | 2 |
| 357L | Cleopatra | 3 | 3 |
| 465H | El código da Vinci | 1 | 1 |
| 856A | El informe pelícano | 0 | 2 |
| 123X | El padrino | 2 | 5 |
| 741G | El padrino II | 2 | 4 |
| 741S | El padrino III | 2 | 3 |
| 123N | Lo que el viento se llevó | 3 | 1 |
| 123S | My Fair Lady | 3 | 3 |
| 314G | Robin Hood, príncipe de ladrones | 3 | 2 |
| 951L | Salvar al soldado Ryan | 3 | 2 |
| 589B | Sonrisas y lágrimas | 3 | 2 |
| 996H | Titanic | 0 | 2 |
| 874F | Un domingo cualquiera | 0 | 3 |
| 321N | Wyatt Earp | 3 | 3 |

18 filas seleccionadas

```
select cod_peli, titulo, count(cod_gen) GEN, count(distinct cod_act) ACT
from (cs_pelicula LEFT JOIN cs_clasificacion using(cod_peli))
     LEFT JOIN cs_actua using(cod_peli)
where duracion > 140
group by cod_peli, titulo
order by titulo;
```

3.7 Consultas conjuntistas

54. Obtener los años, ordenados ascendentemente, que aparecen en la base de datos como año en el que se editó un libro o se filmó una película. Sólo interesan años en los que no aparezca el dígito 9.

```

      ANYO
-----
      1877
      2000
      2001
      2002
      2003
      2004
      2005
      2006
      2007
      2008
      2010
      2011
      2012
      2013
      2014
15 filas seleccionadas

```

```

select anyo
from cs_libro
where anyo is not null
      and anyo not like '%9%'
UNION
select anyo
from cs_pelicula
where anyo is not null
      and anyo not like '%9%';

```

3.8 Consultas generales

55. Obtener el nombre del género (o de los géneros) a los que pertenece la película de duración máxima.

```

COD_G  NOMBRE
-----
BB5    Drama
HH2    Romance
OI9    Bélica

```

```

select cod_gen, nombre
from cs_genero LEFT JOIN cs_clasificacion using (cod_gen)
where cod_peli = (select cod_peli
                  from cs_pelicula
                  where duracion = (select max(duracion)
                                    from cs_pelicula));

```

56. Obtener, para cada actor nacido antes de 1948 y que haya actuado en al menos 2 películas en cualquier papel, el código, el nombre y la fecha de nacimiento indicando en cuántas películas ha actuado con el papel de 'Principal'.

| COD_A | NOMBRE | FECHA_NAC | CUANTOS_PRINCIP |
|-------|------------------|------------|-----------------|
| Z15 | Al Pacino | 25/04/1940 | 4 |
| D49 | Audrey Hepburn | 04/05/1929 | 2 |
| L59 | Clint Eastwood | 31/05/1930 | 3 |
| E22 | Diane Keaton | 05/01/1946 | 0 |
| L45 | Elizabeth Taylor | 27/02/1932 | 2 |
| J47 | Gene Hackman | 30/01/1930 | 4 |
| J45 | Harrison Ford | 13/07/1943 | 1 |
| X45 | Julie Andrews | 01/10/1935 | 2 |

| | | | |
|------------------------|----------------|------------|---|
| J56 | Marlon Brandon | 03/04/1924 | 3 |
| U88 | Morgan Freeman | 01/06/1937 | 3 |
| W34 | Paul Newman | 26/01/1925 | 8 |
| T44 | Rex Harrison | 05/03/1908 | 1 |
| E56 | Robert de Niro | 17/08/1943 | 2 |
| C15 | Robert Duvall | 05/01/1931 | 0 |
| H45 | Robert Redford | 18/08/1936 | 2 |
| W32 | Sean Connery | 25/08/1930 | 2 |
| E45 | Susan Sarandon | 04/10/1946 | 3 |
| D01 | Vivien Leigh | 05/11/1913 | 3 |
| 18 filas seleccionadas | | | |

```

select a.cod_act, nombre, fecha_nac, count(cod_peli) CUANTOS_PRINCIPAL
from cs_actor a LEFT JOIN cs_actua t ON papel = 'Principal'
                                and a.cod_act = t.cod_act
where a.cod_act in (select cod_act
                    from cs_actor
                    where fecha_nac < '01/01/1948'
                    and cod_act in (select cod_act
                                    from cs_actua
                                    group by cod_act
                                    having count(distinct cod_peli) >= 2))
group by a.cod_act, nombre, fecha_nac
order by nombre;

```

4 PRESENTACIÓN DE LA BASE DE DATOS MÚSICA

Se desea almacenar información sobre una discoteca referente a qué discos contiene, qué compañías los han editado, qué canciones hay grabadas y por quién, etcétera; para ello se ha diseñado una base de datos relacional cuyo esquema se muestra a continuación:

CANCION(cod:entero,título:char(30),duración:real)

CP:{cod}
VNN:{título}

COMPANYIA(cod:char(3),nombre:char(30),dir:char(30),fax:char(10),
tfno:char(10))

CP:{cod}
VNN:{nombre}

DISCO(cod:char(3),nombre:char(30),fecha:date,cod_comp:char(3),
cod_gru:char(3))

CP:{cod}
CAj:{cod_comp}→COMPANYIA
VNN:{cod_comp}
CAj:{cod_gru}→GRUPO
VNN:{cod_gru}

ESTA(can:entero,cod:char(3))

CP:{can,cod}
CAj:{can}→CANCIÓN
CAj:{cod}→DISCO

GRUPO(cod:char(3),nombre:char(30),fecha:date,pais:char(10))

CP:{cod}
VNN:{nombre}

ARTISTA(dni:char(10),nombre:char(30))

CP:{dni}

VNN: {nombre}

CLUB (cod:char(3), nombre:char(30), sede:char(30), num:entero,
cod_gru:char(3))
CP: {cod}
CAj: {cod_gru} → GRUPO
VNN: {cod_gru}
VNN: {nombre}

PERTENECE (dni:char(10), cod:char(3), funcion:char(10))
CP: {dni, cod}
CAj: {dni} → ARTISTA
CAj: {cod} → GRUPO

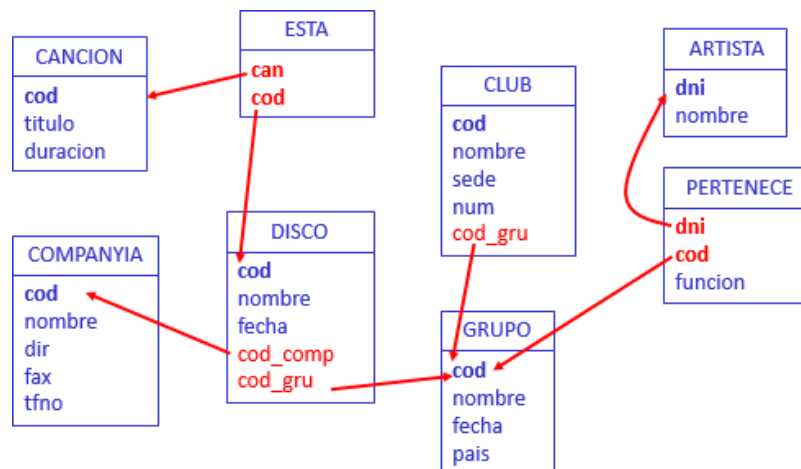
Para aclarar por completo el esquema, a continuación se explica el significado de cada atributo en cada relación:

- **Canción:**
 - *cod*: código de la canción.³
 - *título*: cómo se llama la canción.
 - *duración*: cuánto dura aproximadamente la canción.
- **Companyia:**
 - *cod*: código de la compañía editora de discos.
 - *nombre*: cómo se llama la compañía.
 - *dir*: dirección postal de la compañía.
 - *fax*: número de fax de contacto.
 - *tfno.*: número de teléfono de las oficinas.
- **Disco:**
 - *cod*: código que identifica cada disco.
 - *nombre*: título del disco.
 - *fecha*: fecha en que se editó el disco.
 - *cod_comp*: código de la compañía que ha editado el disco.
 - *cod_gru*: código del grupo que ha grabado el disco.
- **Está:**
 - Mantiene información sobre qué canciones hay grabadas en cada disco. Es decir, *can* es el código de la canción que está grabada en el disco de código *cod*.
- **Grupo:**
 - *cod*: código del grupo.
 - *nombre*: cómo se llama el grupo.
 - *fecha*: cuándo se constituyó el grupo.
 - *país*: dónde surgió el grupo.
- **Artista:**
 - *dni*: nº de dni del artista.
 - *nombre*: cómo se llama el artista.
- **Club:**
 - *cod*: código del club de fans.
 - *nombre*: cómo se llama el club.
 - *sede*: dónde tiene la sede.
 - *num*: cuántas personas pertenecen al club.
 - *cod_gru*: código del grupo que se admira.

- **Pertenece:**

- Mantiene información sobre quién forma parte de cada grupo. Es decir, el artista *dni* forma parte del grupo de código *cod* y realiza en ésta la tarea especificada en *función*.

Gráficamente el esquema relacional es el siguiente:



5 EJERCICIOS SOBRE LA BASE DE DATOS MÚSICA

5.1 Consultas sobre una sola relación

1. ¿Cuántos discos hay?

DISCOS

```
-----
18
```

```
select distinct count(cod)
from Disco;
```

2. Selecciona el nombre de los grupos que no sean de España.

NOMBRE

```
-----
```

U2

Simple Minds

Mike + The Mechanics

Genesis

```
select nombre
from Grupo
where pais <> 'España';
```

3. Obtener el título de las canciones con más de 5 minutos de duración.

TITULO

```
-----
```

7 Deadly Sins

Lemon

So Cruel

Zooropa

```
select titulo
from cancion
where duracion > 5;
```

4. Obtener la lista de las distintas funciones que se pueden realizar en un grupo.

FUNCION

bajo
batería
guitarra
teclado
voz

```
select distinct funcion
from pertenece
order by funcion;
```

5. Obtener la lista de clubs de fans junto con su tamaño (número de personas). La lista debe estar ordenada de menor a mayor según el tamaño del club.

| CLUB | TAMAÑO |
|-------------------------|--------|
| ----- | |
| FanMike | 11 |
| Implicado | 25 |
| Bonoculture | 129 |
| Waterfront | 234 |
| Presuntos | 237 |
| Che U2 | 239 |
| Los Culpables | 355 |
| Jardin Botanico | 357 |
| Troglominds | 999 |
| The best mind | 1413 |
| u2foryou | 1700 |
| Mentes Fuertes | 1984 |
| Zoomania | 2508 |
| Machines | 7789 |
| Futuristas | 9850 |
| Fanaticgens | 12002 |
| Genefans | 23412 |
| 17 filas seleccionadas. | |

```
select nombre, num
from club
order by num;
```

6. Selecciona el nombre y la sede de los clubes de fans con más de 500 socios.

NOMBRE

SEDE

| | |
|------------------------|----------------------|
| Zoomania | 33, Abbey Road |
| Machines | Calle 3, Lab 3 |
| u2foryou | 23, 11th Street |
| Troglominds | C/Lepe 22 |
| Mentes Fuertes | Ramon y Cajal 14 |
| The best mind | 24, Homeround |
| Genefans | C/Visitacion 34 |
| Fanaticgens | Av. H. Dominicos 155 |
| Futuristas | C/Alboraya 10 |
| 9 filas seleccionadas. | |

```
select nombre, sede
from club
where num>500;
```

5.2 Consultas sobre varias relaciones

7. Obtener el nombre y la sede de cada club de fans de grupos de España así como el nombre del grupo al que admiran.

| NOMBRE | SEDE | NOMBRE |
|-----------------|----------------------|----------------------|
| Jardin Botanico | 203,Valencia 46004 | Radio Futura |
| Presuntos | C/Albacete 12, bajo | Presuntos Implicados |
| Implicado | Torrejon de Ardoz 12 | Presuntos Implicados |
| Los Culpables | C/Maria Cristina 67 | Presuntos Implicados |
| Futuristas | C/Alboraya 10 | Radio Futura |

```
select distinct c.nombre, c.sede, g.nombre
from club c, grupo g
where c.cod_gru = g.cod and g.pais = 'España';
```

8. Obtener el nombre de los artistas que pertenezcan a un grupo de España.

```
NOMBRE
-----
Carlos Torero
Enrique Sierra
J.L. Giménez
Luis Auseron
Nacho Maño
Santiago Auseron
Soledad Giménez
7 filas seleccionadas.
```

```
select distinct a.nombre
from artista a, pertenece p, grupo g
where a.dni=p.dni and p.cod = g.cod and g.pais = 'España'
order by nombre;
```

9. Obtener el nombre de los discos que contienen alguna canción que dure más de 5 minutos.

```
NOMBRE
-----
Achtung baby
Good news F.N. world
Zooropa
```

```
select distinct d.nombre
from disco d, esta e, cancion c
where c.cod=e.can and e.cod=d.cod and c.duracion>5
order by nombre;
```

10. Obtener los nombres de las canciones que dan nombre al disco en el que aparecen.

```
TITULO
-----
Alma de blues
De sol a sol

Invisible touch
Living years
October
Ser de agua
The unforgettable fi
Word of mouth
Zooropa
Once upon a time
10 filas seleccionadas.
```

```
select distinct c.titulo
from disco d, esta e, cancion c
where c.cod=e.can and e.cod=d.cod and c.titulo = d.nombre
order by c.titulo;
```

11. Obtener los nombres de compañías y direcciones postales de aquellas compañías que han grabado algún disco que empiece por 'A'.

| NOMBRE | DIR |
|--------|------------|
| ----- | ----- |
| WEA | L Hoyos 42 |
| Island | 67, JB St. |

```
select distinct c.nombre, c.dir
from disco d, compania c
where c.cod = d.cod_comp and d.nombre like 'A%';
```

12. DNI de los artistas que pertenecen a más de un grupo.

| DNI |
|------------|
| ----- |
| 8884566666 |

```
select distinct a.dni
from pertenece a, pertenece c
where a.dni = c.dni and a.cod <> c.cod;
```

5.3 Consultas con subconsultas

13. Obtener el nombre de los discos del grupo más viejo.

| NOMBRE |
|-----------------|
| ----- |
| We can't dance |
| Invisible touch |
| Seconds out |

```
select nombre
from disco
where cod_gru IN (select cod
                  from grupo
                  where fecha = (select min(fecha)
                                from grupo)
                  );
```

14. Obtener el nombre de los discos grabados por grupos con club de fans con más de 5000 personas.

| NOMBRE |
|------------------------|
| ----- |
| Word of mouth |
| Living years |
| We can't dance |
| Invisible touch |
| Seconds out |
| La ley del desierto |
| La canción de Jperro |
| 7 filas seleccionadas. |

```
select nombre
from disco
where cod_gru IN (select cod_gru
                  from club
                  where num>5000);
```

15. Obtener el nombre de los clubes con mayor número de fans indicando ese número.

| NOMBRE | NUM |
|----------|-------|
| ----- | ----- |
| Genefans | 23412 |

```
select nombre, num
from club
where num = (select MAX(num)
             from club);
```

16. Obtener el título de las canciones de mayor duración indicando la duración.

| TITULO | DURACION |
|---------------|----------|
| 7 Deadly Sins | 6 |
| Lemon | 6 |
| So Cruel | 6 |
| Zooropa | 6 |

```
select titulo, duracion
from cancion
where duracion = (select MAX(duracion)
                  from cancion);
```

5.4 Consultas con cuantificación universal

17. Obtener el nombre de las compañías discográficas que no han trabajado con grupos españoles.

| NOMBRE |
|-----------------|
| Island |
| Virgin ATLANTIC |
| PoliDiscos |
| PoliDiscos |

```
select nombre
from companya c
where cod not in (select cod_comp
                  from disco
                  where cod_gru in (select cod
                                    from grupo
                                    where pais = 'España'));
```

18. Obtener el nombre de las compañías discográficas que sólo han trabajado con grupos españoles.

| NOMBRE |
|--------|
| ARIOLA |
| WEA |

```
select nombre
from companya c
where cod in (select cod_comp
              from disco
              where cod_gru in (select cod
                                from grupo
                                where pais = 'España'));
```

19. Obtener el nombre y la dirección de aquellas compañías discográficas que han grabado todos los discos de algún grupo.

| NOMBRE | DIR |
|----------|------------|
| ARIOLA | Aragon 204 |
| ATLANTIC | 12, E St. |
| Island | 67, JB St. |
| Virgin | 2,23th St. |
| WEA | L Hoyos 42 |

```
select nombre, dir
from companya c
where cod in (select cod_comp
              from disco e
              where cod_gru in (select cod
                                from grupo
                                where cod in (select cod_gru
                                              from disco t
                                              where t.cod_comp = e.cod_comp)))
order by nombre;
```

5.5 Consultas agrupadas

20. Obtener el nombre de los grupos que sean de España y la suma de sus fans.

| NOMBRE | FANS |
|----------------------|-------|
| Presuntos Implicados | 617 |
| Radio Futura | 10207 |

```
select g.nombre, sum(c.num) FANS
from grupo g, club c
where g.pais = 'España' and g.cod = c.cod_gru
group by g.nombre;
```

21. Obtener para cada grupo con más de dos componentes el nombre y el número de componentes del grupo.

| NOMBRE | NUMERO |
|----------------------|--------|
| Genesis | 3 |
| Mike + The Mechanics | 4 |
| Presuntos Implicados | 3 |
| Radio Futura | 4 |
| U2 | 4 |

```
select g.nombre, count(p.dni) NUMERO
from grupo g, pertenece p
where g.cod = p.cod
group by g.nombre
having count(p.dni)>2
order by g.nombre;
```

22. Obtener el número de discos de cada grupo.

| NOMBRE | DISCOS |
|----------------------|--------|
| U2 | 4 |
| Simple Minds | 4 |
| Mike + The Mechanics | 2 |
| Genesis | 3 |
| Presuntos Implicados | 3 |
| Radio Futura | 2 |

6 filas seleccionadas.

```
select g.nombre, count(d.cod) DISCOS
from grupo g, disco d
where g.cod = d.cod_gru
group by g.nombre
order by g.nombre desc;
```

23. Obtener el número de canciones que ha grabado cada compañía discográfica y su dirección.

| NOMBRE | CANCIONES | DIR |
|------------|-----------|---------------|
| ARIOLA | 22 | Aragon 204 |
| ATLANTIC | 54 | 12, E St. |
| Island | 43 | 67, JB St. |
| PoliDiscos | 0 | Cami de Vera |
| PoliDiscos | 0 | Polynesia St. |
| Virgin | 34 | 2,23th St. |
| WEA | 31 | L Hoyos 42 |

7 filas seleccionadas.

```
select c.nombre, count(can) CANCIONES, c.dir
from compania c LEFT JOIN (disco d LEFT JOIN esta t
                        on d.cod = t.cod)
                        on c.cod = d.cod_comp
group by c.nombre, c.dir
order by c.nombre;
```

5.6 Consultas generales

24. Obtener los nombre de los artistas de grupos con clubes de fans de más de 500 personas y que el grupo sea de Inglaterra.

NOMBRE

```
-----
Adam Clayton
Adrian Lee
Bono
C. Burchill
Edge
Jim Kerr
Larry Jr.Mullen
M. Rutherford
P. van Hooke
Paul Young
Phil Collins
Tony Banks
12 filas seleccionadas.
```

```
select nombre
from artista
where dni in (select dni
              from pertenece
              where cod in (select cod
                           from grupo
                           where pais='Inglaterra'
                           and cod in (select cod_gru
                                       from club
                                       where num > 500)))

order by nombre;
```

25. Obtener el título de las canciones de todos los discos del grupo U2.

TITULO

```
-----
4th of July
A sort of homecoming
Artitoestoy
Babyface
Bad
Daddys Goma pay for
Dirty day
Elvis Presley & USA
Even Better Than...
Fire
Fly
Gloria
I Fall Down
I Threw a Brick
...
43 filas seleccionadas.
```

```
select titulo
from cancion
where cod in (select can
              from esta
              where cod in (select cod
                           from disco
                           where cod_gru in (select cod
                                             from grupo
                                             where nombre = 'U2'))))

order by titulo;
```


26. El dúo dinámico por fin se jubila; para sustituirles se pretende hacer una selección sobre todos los pares de artistas de grupos españoles distintos tales que el primero sea voz y el segundo guitarra. Obtener dicha selección.

| VOZ | GUITARRA |
|------------------|----------------|
| ----- | ----- |
| Soledad Giménez | Enrique Sierra |
| Santiago Auseron | J.L. Giménez |

```
select a.nombre VOZ, v.nombre GRUITARRA
from artista a, artista v, pertenece p, grupo
g, pertenece s, grupo t
where p.cod <> s.cod
      and a.dni = p.dni
      and p.funcion = 'voz'
      and p.cod = g.cod
      and g.pais = 'España'
      and v.dni = s.dni
      and s.funcion = 'guitarra'
      and s.cod = t.cod
      and t.pais = 'España'
      and g.pais = t.pais;
```

27. Obtener el nombre de los artistas que pertenecen a más de un grupo.

| NOMBRE |
|---------------|
| ----- |
| M. Rutherford |

```
select nombre
from artista a, pertenece p
where a.dni = p.dni
group by nombre
having count(cod)>1;
```

28. Obtener el título de la canción de mayor duración si es única.

no se ha seleccionado ninguna fila

```
select titulo
from cancion
where 1 = (select count(distinct cod)
           from cancion
           where duracion > ALL(select duracion
                                from cancion));
```

29. Obtener el décimo (debe haber sólo 9 por encima de él) club con mayor número de fans indicando ese número.

| NOMBRE | NUM |
|-----------------|-------|
| ----- | ----- |
| Jardin Botanico | 357 |

```
select nombre, num
from club c
where 10 = (select count(distinct cod)
            from club
            where num >= c.num);
```

30. Obtener el nombre de los artistas que tengan la función de bajo en un único grupo y que además éste tenga más de dos miembros.

| NOMBRE |
|--------------|
| ----- |
| Adam Clayton |
| Luis Auseron |
| Nacho Maño |

```

select nombre
from artista LEFT JOIN pertenece p using (dni)
where funcion = 'bajo'
      and dni not in(select dni
                      from pertenece
                      where cod <> p.cod
                      and funcion = 'bajo')
      and 2 < (select count(dni)
               from pertenece
               where cod = p.cod)
order by nombre;

```

31. ¿Cuál es la compañía discográfica que más canciones ha grabado?

| NOMBRE | CANCIONES |
|----------|-----------|
| ATLANTIC | 54 |

```

select c.nombre, count(can) CANCIONES
from companya c LEFT JOIN (disco d LEFT JOIN esta using (cod))
on c.cod = d.cod_comp
group by c.nombre, c.cod
having count(can) = ALL (select max(count(can))
                        from companya t LEFT JOIN (disco s LEFT JOIN esta using (cod))
                        on t.cod = s.cod_comp
                        group by t.cod);

```

6 PRESENTACIÓN DE LA BASE DE DATOS BIBLIOTECA

Se desea mantener información de una biblioteca doméstica; para ello se ha definido una base de datos relacional cuyo esquema se muestra a continuación:

AUTOR(autor_id: char(4), nombre: char(35), nacionalidad: char(20))
 CP: {autor_id}
 VNN: {nombre}

LIBRO(id_lib: char(10), titulo: char(80), año: entero, num_obras: entero)
 CP: {id_lib}

TEMA(tematica: char(20), descripcion: char(50))
 CP: {tematica}

OBRA(cod_ob: entero, titulo: char(80), tematica: char(20))
 CP: {cod_ob}
 CAj: {tematica} → TEMA
 VNN: {titulo}

AMIGO(num: entero, nombre: char(60), telefono: char(10))
 CP: {num}
 VNN: {nombre}

LEER(num: entero, cod_ob: entero)
 CP: {num, cod_ob}
 CAj: {num} → AMIGO
 CAj: {cod_ob} → OBRA

ESTA_EN(cod_ob: entero, id_lib: char(10))
 CP: {cod_ob, id_lib}
 CAj: {cod_ob} → OBRA
 CAj: {id_lib} → LIBRO

ESCRIBIR(cod_ob: entero, autor_id: char(4))
 CP: {cod_ob, autor_id}

CAj: {cod_ob} → OBRA

CAj: {autor_id} → AUTOR

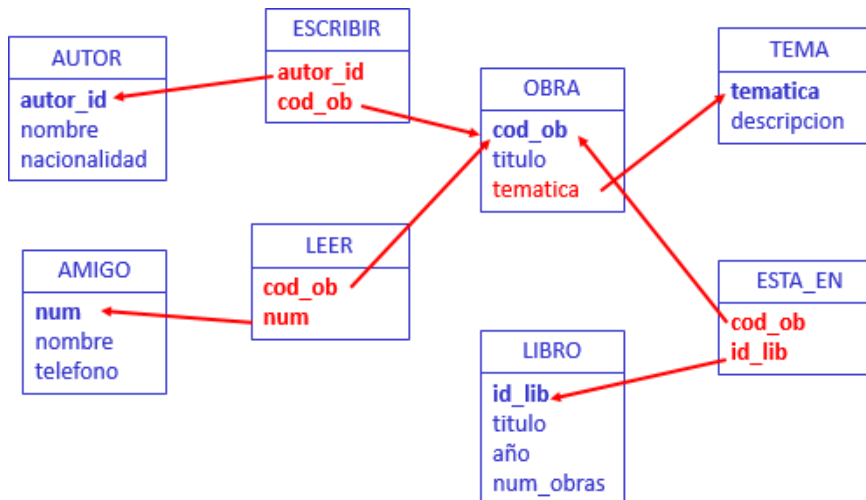
Para aclarar por completo el esquema, a continuación se explica el significado de cada atributo en cada relación:

- **Autor:**
 - *autor_id*: identificador del autor.
 - *nombre*: nombre del autor.
 - *nacionalidad*: nacionalidad del autor.
- **Libro:**
 - *id_lib*: identificador del libro.
 - *título*: título del libro.
 - *año*: año de publicación del libro.
 - *num_obras*: cuántas obras hay en el libro.
- **Tema:**
 - *temática*: identificador de cada tema.
 - *descripción*: breve descripción del tema.
- **Obra:**
 - *cod_ob*: identificador de la obra.
 - *título*: título de la obra.
 - *temática*: tema en el que se clasifica la obra.
- **Amigo:**
 - *num*: identificador de cada amigo.
 - *nombre*: nombre del amigo.
 - *teléfono*: teléfono del amigo.
- **Leer**: el amigo identificado por *num* ha leído la obra de código *cod_ob*.
- **Esta_en**: la obra de código *cod_ob* está incluida en el libro identificado por *id_lib*.
- **Escribir**: el autor con identificador *autor_id* ha escrito la obra de código *cod_ob*.

Además, los datos deben cumplir las propiedades siguientes:

- El valor del atributo *num_obras* de un libro siempre debe ser igual al número de tuplas de *Esta_en* donde aparece dicho libro.
- Todo libro contiene al menos una obra.
- Si un libro tiene título y sólo consta de una obra, entonces se asume que el título del libro es el título de la obra.

Gráficamente el esquema relacional es el siguiente:



7 EJERCICIOS SOBRE LA BASE DE DATOS BIBLIOTECA

7.1 Consultas sobre una sola relación

1. Obtener el nombre de los autores de nacionalidad 'Argentina'.

NOMBRE

 Bioy Casares, Adolfo
 Borges, Jorge Luis
 Cortázar, Julio

```
select nombre
from autor
where nacionalidad = 'Argentina';
```

2. Obtener los títulos de las obras que contengan la palabra 'mundo'.

TITULO

 Un mundo feliz
 El ahogado más hermoso del mundo

```
select titulo
from obra
where titulo like '%mundo%';
```

3. Obtener el identificador de los libros anteriores a 1990 y que contengan más de una obra indicando el número de obras que contiene.

ID_LIB NUM_OBRAS

 LIB-000187 2

```
select id_lib, num_obras
from libro
where año < 1990 and num_obras > 1;
```

4. ¿Cuántos libros hay de los que se conozca el año de adquisición?

LIB_AÑO

 92

```
select count(*) LIB_AÑO
from libro
where año is not null;
```

5. ¿Cuántos libros tienen más de una obra? Resolver este ejercicio utilizando el atributo *num_obras*.

MÁS_1_OB

30

```
select count(*) MÁS_1_OB
from libro
where num_obras > 1;
```

6. Obtener el identificador de los libros del año 1997 que no tienen título.

ID_LIB

LIB-000045

LIB-000046

LIB-000048

LIB-000310

LIB-000311

LIB-000424

6 filas seleccionadas.

```
select id_lib
from libro
where año = 1997 and titulo is null;
```

7. Mostrar todos los títulos de los libros que tienen título en orden alfabético descendente.

TITULO

Vuelva usted mañana y otros artículos

Vox

Tres pastiches victorianos

Todos los cuentos. EL balneario y las ataduras

Sherlock Holmes. Obras completas III

Sherlock Holmes. Obras completas II

Sherlock Holmes. Obras completas I

Relatos que me asustaron

Raymon Chandler. Obras selectas II

Raymon Chandler. Obras selectas I

Pequeños cuentos misóginos

Narraciones extraordinarias

Lo infinitamente pequeño

La mano parda y otros relatos

La increíble y triste historia de la cándida Eréndida y su abuela desalmada

Inglés-Español, VOX

Francés-Español, Sopena

Doce cuentos peregrinos Cuentos juveniles

Cuentos de la taberna del ciervo blanco

Cuentos

Clásicos de Grecia y Roma

Blanco en azul

Algunos cuentos chilenos

24 filas seleccionadas.

```
select titulo
from libro
where titulo is not null
order by titulo desc;
```

8. Obtener cuántas obras hay en los libros publicados entre 1990 y 1999.

OBRAS

127

```
select sum(num_obras) OBRAS
from libro
where año between 1990 and 1999;
```

7.2 Consultas sobre varias relaciones

9. Obtener cuántos autores han escrito alguna obra con la palabra “ciudad” en su título.

AUTORES

4

```
select count(distinct autor_id) AUTORES
from escribir e, obra t
where titulo like '%ciudad%' and e.cod_ob = t.cod_ob;
```

10. Obtener el título de todas las obras escritas por el autor de nombre ‘Camús, Albert’.

TITULO

El extranjero

```
select titulo
from escribir e, obra t, autor a
where nombre = 'Camús, Albert' and a.autor_id = e.autor_id and e.cod_ob = t.cod_ob;
```

11. ¿Quién es el autor de la obra de título ‘La tata’?

NOMBRE

Martín Gaité, Carmen

```
select nombre
from escribir e, obra t, autor a
where titulo = 'La tata' and a.autor_id = e.autor_id and e.cod_ob = t.cod_ob;
```

12. Obtener el nombre de los amigos que han leído alguna obra del autor de identificador ‘RUKI’.

NOMBRE

Isabel Peiró García

Eloy Prim Gros

```
select distinct nombre
from escribir e, obra t, leer l, amigo a
where autor_id = 'RUKI'
      and e.cod_ob = t.cod_ob
      and t.cod_ob = l.cod_ob
      and l.num = a.num;
```

13. Obtener el título y el identificador de los libros que tengan título y más de una obra. Resolver este ejercicio sin utilizar el atributo *num_obras*.

ID_LIB TÍTULO

```
LIB-000084 Raymon Chandler. Obras selectas II
LIB-000066 Sherlock Holmes. Obras completas I
LIB-000301 Cuentos juveniles
LIB-000177 Doce cuentos peregrinos
LIB-000030 Clásicos de Grecia y Roma
LIB-000076 La mano parda y otros relatos
LIB-000100 Pequeños cuentos misóginos
LIB-000197 Cuentos
LIB-000266 Blanco en azul
LIB-000288 Vuelva usted mañana y otros artículos
LIB-000017 Tres pastiches victorianos
LIB-000070 Sherlock Holmes. Obras completas II
LIB-000081 Raymon Chandler. Obras selectas I
LIB-000180 La increíble y triste historia de la cándida Eréndida y su abuela
LIB-000044 Algunos cuentos chilenos
```

```
LIB-000470 Narraciones extraordinarias
LIB-000226 Relatos que me asustaron
LIB-000349 Todos los cuentos. EL balneario y las ataduras
LIB-000073 Sherlock Holmes. Obras completas III
LIB-000131 Lo infinitamente pequeño
LIB-000034 Cuentos de la taberna del ciervo blanco
21 filas seleccionadas
```

```
select l.id_lib, l.titulo
from libro l
where l.titulo is not null
      and 1 < (select count(cod_ob)
               from esta_en e
               where e.id_lib = l.id_lib);
```

7.3 Consultas con subconsultas

14. Obtener el título de las obras escritas sólo por un autor si éste es de nacionalidad "Francesa" indicando también el nombre del autor.

| TITULO | NOMBRE |
|-----------------------------------------|-----------------------|
| ----- | ----- |
| Bella del señor | Cohen, Albert |
| El método Montignac | Montignac, Michel |
| Madame Bovary | Flaubert, Gustave |
| La hierba roja | Vian, Boris |
| Con las mujeres no hay quien pueda | Vian, Boris |
| Que se mueran los feos | Vian, Boris |
| Escupiré sobre vuestras tumbas | Vian, Boris |
| El lobo hombre | Vian, Boris |
| El extranjero | Camús, Albert |
| Bosquejo de una teoría de las emociones | Sartre, Jean-Paul |
| El amante | Duras, Marguerite |
| Ana, soror... | Yourcenar, Marguerite |
| Opus nigrum | Yourcenar, Marguerite |
| Los amotinados de la "Bounty" | Verne, Jules |
| 14 filas seleccionadas. | |

```
select titulo, nombre
from obra, autor a
where nacionalidad = 'Francesa'
      and cod_ob in (select cod_ob
                    from escribir
                    where autor_id = a.autor_id
                    and cod_ob not in (select cod_ob
                                       from escribir
                                       where autor_id <> a.autor_id))

order by cod_ob;
```

15. ¿Cuántos autores hay en la base de datos de los que no se tiene ninguna obra?

SIN_OBRA

3

```
select count(autor_id) SIN_OBRA
from autor
where autor_id not in (select autor_id
                      from escribir);
```

16. Obtener el nombre de esos autores.

NOMBRE

Peris Rossi, Cristina
Apollinaire, Guillaume
García Hortelano, Juan

```
select nombre
from autor
where autor_id not in (select autor_id
                      from escribir)
order by autor_id;
```

17. Obtener el nombre de los autores de nacionalidad “Española” que han escrito dos o más obras.

NOMBRE

Buero Vallejo, Antonio
Pérez Reverte, Arturo
Martínez Ruíz, Jose; Azorín
Cela, Camilo José
Martín Gaité, Carmen
García Sánchez, Javier
Marías, Javier
Borrell, Joaquín
Pla, Josep
Sampedro, José Luis
Millás, Juan José
De Vega, Lope
De Larra, Mariano José
Vázquez Montalbán, Manuel
Delibes, Miguel
Hernández, Miguel
Pérez de Ayala, Ramón
Sender, Ramón J.
18 filas seleccionadas.

```
select a.nombre
from autor a
where nacionalidad = 'Española'
      and 2 <= (select count(e.cod_ob)
               from escribir e
               where e.autor_id = a.autor_id)
order by a.autor_id;
```

18. Obtener el nombre de los autores de nacionalidad “Española” que han escrito alguna obra que está en dos o más libros.

NOMBRE

Valera, Juan

```
select nombre
from autor
where nacionalidad = 'Española'
      and autor_id in (select autor_id
                      from escribir s
                      where 2 <= (select count(e.id_lib)
                                from esta_en e
                                where s.cod_ob = e.cod_ob));
```


19. Obtener el título y el código de las obras que tengan más de un autor.

| COD_OB | TITULO |
|--------|------------------|
| 151 | El quinto jinete |
| 170 | A escullar |

```
select t.cod_ob, titulo
from obra t
where 1 < (select count(autor_id)
           from escribir
           where cod_ob = t.cod_ob);
```

7.4 Consultas con cuantificación universal

20. Obtener el nombre de los amigos que han leído todas las obras del autor de identificador 'RUKI'.

NOMBRE

Isabel Peiró García

```
select nombre
from amigo a
where not exists (select *
                  from escribir
                  where autor_id = 'RUKI'
                     and cod_ob not in (select cod_ob
                                         from leer
                                         where num = a.num))

and exists (select *
            from escribir
            where autor_id = 'RUKI');
```

21. Resolver de nuevo la consulta anterior pero para el autor de identificador 'GUAP'.

No se ha seleccionado ninguna fila.

```
select nombre
from amigo a
where not exists (select *
                  from escribir
                  where autor_id = 'GUAP'
                     and cod_ob not in (select cod_ob
                                         from leer
                                         where num = a.num))

and exists (select *
            from escribir
            where autor_id = 'GUAP');
```

22. Obtener el nombre de los amigos que han leído todas las obras de algún autor de los que hay en la tabla autor.

NOMBRE

Isabel Peiró García
Yolanda Milanés Cuba

```
select distinct a.nombre
from amigo a, leer l, escribir e
where a.num = l.num
   and l.cod_ob = e.cod_ob
   and not exists (select *
                   from escribir
                   where autor_id = e.autor_id
                      and cod_ob not in (select cod_ob
                                          from leer
                                          where num = a.num))

and exists (select *
            from escribir
            where autor_id = e.autor_id);
```

23. Resolver la consulta anterior indicando también el nombre de ese autor.

NOMBRE_AMIGO

NOMBRE_AUTOR

| | |
|----------------------|------------------|
| Isabel Peiró García | Maalouf, Amin |
| Yolanda Milanés Cuba | Vian, Boris |
| Isabel Peiró García | Kipling, Rudyard |

```
select distinct a.nombre, t.nombre
from amigo a JOIN leer l
            on (a.num = l.num)
            NATURAL JOIN escribir e
            JOIN autor t
            on e.autor_id = t.autor_id
where not exists (select *
                  from escribir
                  where autor_id = e.autor_id
                  and cod_ob not in (select cod_ob
                                     from leer
                                     where num = a.num))

and exists (select *
            from escribir
            where autor_id = e.autor_id);
```

24. Obtener el nombre de los amigos que sólo han leído obras del autor de identificador 'CAMA'.

NOMBRE

Pepe Pérez Pérez

```
select nombre
from amigo a
where num not in (select num
                  from leer
                  where cod_ob not in (select cod_ob
                                       from escribir
                                       where autor_id = 'CAMA'))

and num in (select num
            from leer);
```

25. Resolver de nuevo la consulta anterior pero para el autor de identificador 'GUAP'.

No se ha seleccionado ninguna fila.

```
select nombre
from amigo a
where num not in (select num
                  from leer
                  where cod_ob not in (select cod_ob
                                       from escribir
                                       where autor_id = 'GUAP'))

and num in (select num
            from leer);
```

26. Obtener el nombre de los amigos tales que todas las obras que han leído son del mismo autor.

NOMBRE

Eloy Prim Gros
Pepe Pérez Pérez
Yolanda Milanés Cuba

```
select nombre
from amigo a
where num in (select num
              from leer
              where cod_ob in (select cod_ob
                               from escribir e
                               where not exists(select *
                                                from escribir
                                                where autor_id <> e.autor_id
                                                and cod_ob in (select cod_ob
                                                                from leer
                                                                where num = a.num))))

order by nombre;
```

27. Resolver la consulta anterior indicando también el nombre del autor.

AMIGO

AUTOR

 Eloy Prim Gros
 Pepe Pérez Pérez
 Yolanda Milanés Cuba

 Kipling, Rudyard
 Martín Gaité, Carmen
 Vian, Boris

```
select a.nombre AMIGO, t.nombre AUTOR
from amigo a, autor t
where num not in (select num
                  from leer
                  where cod_ob not in (select cod_ob
                                      from escribir
                                      where autor_id = t.autor_id))
      and num in (select num
                  from leer)
order by t.nombre;
```

28. Obtener el nombre de los amigos que han leído todas las obras de algún autor y no han leído nada de ningún otro indicando también el nombre del autor.

NOMBRE_AMIGO

NOMBRE

 Yolanda Milanés Cuba Vian, Boris

```
select distinct a.nombre, t.nombre
from amigo a, leer l, escribir e, autor t
where a.num = l.num
      and l.cod_ob = e.cod_ob
      and e.autor_id = t.autor_id
      and not exists (select *
                     from escribir
                     where autor_id = t.autor_id
                        and cod_ob not in (select cod_ob
                                         from leer
                                         where num = a.num))
      and not exists (select *
                     from escribir
                     where autor_id <> t.autor_id
                        and cod_ob in (select cod_ob
                                     from leer
                                     where num = a.num));
```

29. Resolver el ejercicio 13 usando la cláusula GROUP BY.

Ver resultado en consulta 13.

```
select l.id_lib, titulo
from libro l, esta_en e
where l.id_lib = e.id_lib
      and titulo is not null
group by l.id_lib, titulo
having count(cod_ob)>1;
```

30. Obtener el nombre de los amigos que han leído más de 3 obras indicando también la cantidad de obras leídas.

NOMBRE

CUÁNTAS

 Isabel Peiró García
 Yolanda Milanés Cuba

 7
 5

```
select a.nombre, count(cod_ob) CUÁNTAS
from amigo a, leer l
where a.num = l.num
group by a.nombre
having count(cod_ob)>3;
```

31. Obtener, de los temas con alguna obra, la temática y la cantidad de obras con ese tema.

| TEMATICA | NUM_OBRAS |
|------------------------|-----------|
| ----- | ----- |
| Antropología | 4 |
| Artículo | 57 |
| Aventuras | 2 |
| Biografía | 6 |
| Ciencia Ficción | 6 |
| Clásico | 14 |
| Cocina | 10 |
| Cuento | 164 |
| Experiencias | 1 |
| Filosofía | 3 |
| Histórica | 16 |
| Intriga | 1 |
| Inventada | 1 |
| Juvenil | 18 |
| Lógica | 3 |
| Misterio | 60 |
| Mitología | 1 |
| Negra | 23 |
| Novela | 139 |
| Poesía | 9 |
| Teatro | 7 |
| Viajes | 10 |
| 22 filas seleccionadas | |

```
select tematica, count(cod_ob) NUM_OBRAS
from obra o
where tematica is not null
group by tematica
order by tematica;
```

32. Obtener, de todos los temas de la base de datos, la temática y la cantidad de obras con ese tema.

| TEMATICA | NUM_OBRAS |
|------------------------|-----------|
| ----- | ----- |
| Antropología | 4 |
| Artículo | 57 |
| Aventuras | 2 |
| Biografía | 6 |
| Ciencia Ficción | 6 |
| Clásico | 14 |
| Cocina | 10 |
| Cuento | 164 |
| Diccionario | 0 |
| Ensayo | 0 |
| Experiencias | 1 |
| Filosofía | 3 |
| Histórica | 16 |
| Intriga | 1 |
| Inventada | 1 |
| Juvenil | 18 |
| Lógica | 3 |
| Misterio | 60 |
| Mitología | 1 |
| Negra | 23 |
| Novela | 139 |
| Poesía | 9 |
| Teatro | 7 |
| Viajes | 10 |
| 24 filas seleccionadas | |

```
select t.tematica, count(cod_ob) NUM_OBRAS
from tema t, obra o
where t.tematica = o.tematica
group by t.tematica
having count(cod_ob) >= 0
order by tematica;
```

33. Obtener el nombre del autor (o autores) que más obras han escrito.

NOMBRE

Pla, Josep

```

select nombre
from autor a, escribir e
where a.autor_id = e.autor_id
group by nombre
having count(cod_ob) >= ALL(select max(count(cod_ob))
                           from escribir
                           where autor_id in (select autor_id
                                              from autor)
                           group by autor_id);

```

34. Obtener la nacionalidad (o nacionalidades) menos frecuentes.

NACIONALIDAD

Alemana
Checa
Colombiana
Danesa
Griega
Mejicana

```

select nacionalidad
from autor
group by nacionalidad
having count(autor_id) <= ALL(select min(count(autor_id))
                             from autor
                             group by nacionalidad)
order by nacionalidad;

```

35. Obtener el nombre del amigo (o amigos) que han leído más obras.

NOMBRE

Isabel Peiró García

```

select nombre
from amigo a, leer l
where a.num = l.num
group by nombre
having count(cod_ob) >= ALL(select max(count(cod_ob))
                           from leer
                           group by num);

```

7.6 Consultas generales**36. Obtener el título y el identificador de los libros que tengan título y que contengan sólo una obra.
no se ha seleccionado ninguna fila.**

```

select l.titulo, l.id_lib
from libro l, esta_en e
where l.titulo is not null
      and l.id_lib = e.id_lib
      and 1 = (select count(cod_ob)
               from esta_en
               where id_lib = l.id_lib);

```

37. Como se concluye del resultado de la consulta anterior, los libros con una sola obra no tienen título propio. Asumiendo en este caso que su título es el de la obra que contienen, obtener la lista de todos los títulos de libros que hay en la base de datos tengan las obras que tengan.

TITULO

...
301 filas seleccionadas.

```
select l.titulo
from libro l, esta_en e, obra t
where t.cod_ob = e.cod_ob
      and l.id_lib = e.id_lib
      and num_obras = 1;

#FALTAN 22#
```

38. Obtener el nombre de los amigos que han leído alguna obra del autor de identificador 'CAMA'.

NOMBRE

Pepe Pérez Pérez
Isabel Peiró García
Isidro Catalá Ferrer

```
select distinct a.nombre
from amigo a, leer l, escribir e
where a.num = l.num
      and l.cod_ob = e.cod_ob
      and autor_id = 'CAMA';
```

39. Obtener el nombre de los amigos que no han leído ninguna obra del autor de identificador 'CAMA'.

NOMBRE

Marina Sánchez Vidal
Eloy Prim Gros
Yolanda Milanés Cuba
Félix Díaz Drac

```
select distinct nombre
from amigo a
where num not in(select num
                  from leer
                  where cod_ob in (select cod_ob
                                   from escribir
                                   where autor_id = 'CAMA'))

order by nombre;
```

40. Obtener el nombre de los amigos que no han leído ninguna obra del autor de identificador 'CAMA' pero que han leído algo.

NOMBRE

Eloy Prim Gros
Yolanda Milanés Cuba

```
select distinct nombre
from amigo a
where num not in(select num
                  from leer
                  where cod_ob in (select cod_ob
                                   from escribir
                                   where autor_id = 'CAMA'))

      and num in (select num
                  from leer)

order by nombre;
```

41. Obtener el nombre del amigo (o amigos) que han leído más obras. No utilizar GROUP BY.

NOMBRE

Isabel Peiró García

```
select nombre
from amigo a
where num in (select num
              from leer l
              where (select count(cod_ob)
                    from leer
                    where num = l.num) >= ALL (select (select count(cod_ob)
                                                    from leer
                                                    where num = p.num)
                                              from leer p));
```

8 PRESENTACIÓN DE LA BASE DE DATOS DEPARTAMENTO

Este esquema relacional es el más complicado de todos los propuestos, además, deliberadamente, las consultas ya no se presentan agrupadas por el tipo de solución. Se aconseja dejar este esquema para cuando se hayan trabajado completamente al menos dos de los tres anteriormente presentados.

El DSIC desea crear una base de datos para almacenar información sobre la gestión de la docencia en el departamento; para ello se ha diseñado una base de datos relacional cuyo esquema se muestra a continuación:

CENTRO(ccen:char(6),nmcen:char(100))

CP:{ccen}

PROFESOR(nip:entero,ap1:char(20),ap2:char(20),nom:char(20),ctg:char(5),
ccen:char(6),doctor:char(1),hded:real,area:char(3),
grupo_inv:char(10))

CP:{nip}

CAj:{ccen}→Centro

CAj:{grupo_inv}→Grupo_inv(cgi)

TITULACION(ctit:char(6),nomtit:char(45))

CP:{ctit}

ASIGNATURA(ccen:char(6),ctit:char(6),casg:char(3),cu:char(2),
nmasg:char(45),resp:entero,udo:char(3),nprofvig:entero,
caracter:char(2),gt:entero,gp:entero,ct:real,cp:real,
nalm:entero,area:char(3),ciclo:entero)

CP:{ccen,ctit,casg}

CAj:{ccen}→Centro Borrado en cascada

CAj:{resp}→Profesor f(resp) = nip Borrado en cascada

CAj:{ctit}→Titulacion Borrado en cascada

CAj:{udo}→Udocente Borrado en cascada

GRUPO_INV(cgi:char(10),nombre:char(100),responsable:entero,
grupo_princ:char(10))

CP:{cgi}

CAj:{responsable}→Profesor(nip) Borrado en cascada

CAj:{grupo_princ}→Grupo_inv(cgi)

VNN:{nombre,responsable}

DOCENCIA(ccen:char(6),ctit:char(6),casg:char(3),nip:entero,gtp:real,
gpp:real)

CP:{ccen,ctit,casg,nip}

CAj:{ccen,ctit,casg}→Asignatura Borrado en cascada
 CAj:{nip}→Profesor Borrado en cascada

UDOCENTE (udo:char(3), nudoc:char(40), resp:entero)

CP:{udo}

CAj:{resp}→Profesor(nip)

VIGILA_EXAMEN (conv:entero, ccen:char(6), ctit:char(6), casg:char(3),
 nip:entero)

CP:{conv,ccen,ctit,casg,nip}

CAj:{nip}→Profesor Borrado en cascada

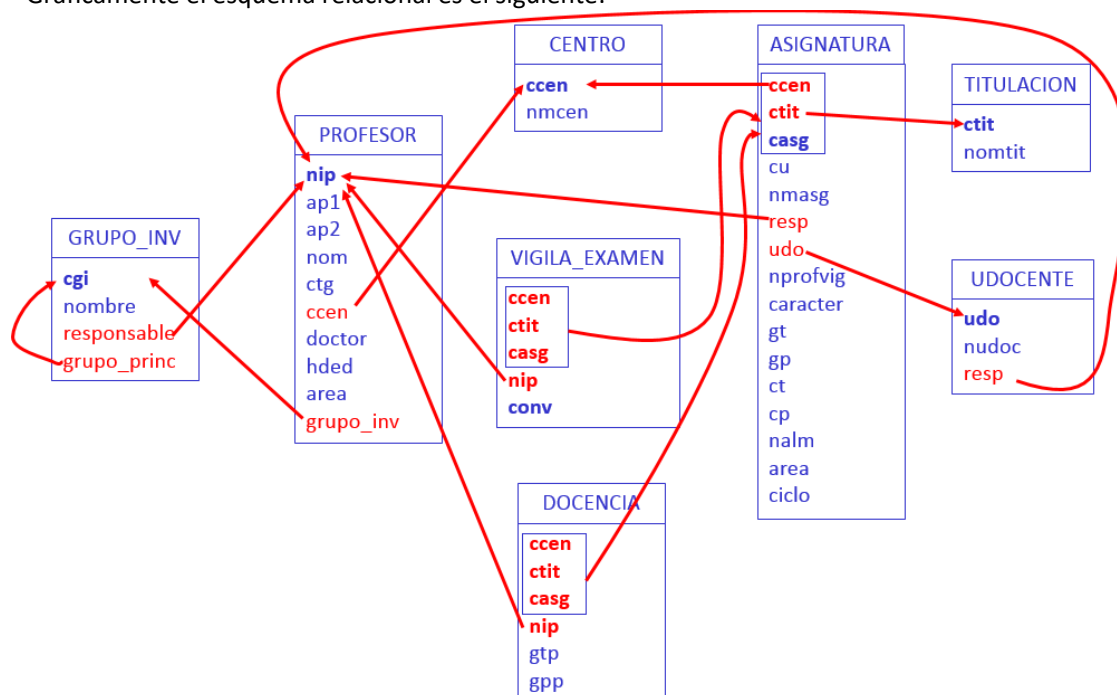
CAj:{ccen,ctit,casg}→Asignatura Borrado en cascada

Donde las distintas relaciones y atributos tienen el significado siguiente:

- **Centro:** centros con docencia de asignaturas del departamento.
 - *ccen*: código del centro.
 - *nmcen*: nombre del centro.
- **Titulación:** titulaciones con asignaturas impartidas por el departamento.
 - *ctit*: código de la titulación.
 - *nomtit*: nombre de la titulación.
- **Profesor:** profesores del departamento.
 - *nip*: el número identificador del profesor.
 - *ap1*: primer apellido.
 - *ap2*: segundo apellido.
 - *nom*: nombre.
 - *ctg*: categoría.
 - *ccen*: centro de adscripción.
 - *doctor*: si es o no doctor (1 es doctor, 0 no es doctor).
 - *hded*: créditos de dedicación del profesor.
 - *area*: área de conocimiento a la que pertenece el profesor.
 - *grupo_inv*: grupo de investigación al que pertenece.
- **Asignatura:** asignaturas del departamento.
 - *ccen*: código del centro.
 - *ctit*: código de la titulación.
 - *casg*: código de la asignatura en una titulación de un centro.
 - *cu*: curso y semestre de la asignatura.
 - *nmasg*: nombre de la asignatura.
 - *resp*: identificador del profesor responsable.
 - *udo*: código de la Unidad Docente a la que está adscrita la asignatura.
 - *nprofvig*: número de profesores solicitados inicialmente para la organización de la vigilancia de exámenes. (No es un atributo derivado).
 - *caracter*: carácter de la asignatura (OB es obligatoria, TR es troncal, OP es optativa)
 - *gt*: número de grupos de teoría en el plan de estudios.
 - *gp*: número de grupos de prácticas en el plan de estudios.
 - *ct*: créditos por cada grupo de teoría.
 - *cp*: créditos por cada grupo de prácticas.
 - *nalm*: número de alumnos matriculados.
 - *area*: área de conocimiento a la que pertenece la asignatura.
 - *ciclo*: ciclo en el que se imparte.
- **Grupo_inv:** grupos de investigación del departamento.
 - *cgi*: código grupo de investigación.

- *nombre*: nombre del grupo de investigación.
 - *responsable*: identificador del profesor responsable.
 - *grupo_princ*: grupo de investigación principal al que está subordinado.
 - **Docencia**: docencia de asignaturas por profesores.
 - (*ccen*, *ctit*, *casg*): código de la asignatura.
 - *nip*: identificador del profesor.
 - *gtp*: número de grupos de teoría que imparte el profesor en la asignatura.
 - *gpp*: número de grupos de práctica que imparte el profesor en la asignatura.
-
- **Udocente**: unidades docentes en las que se organiza el departamento.
 - *udo*: código de la unidad docente.
 - *nudoc*: nombre de la unidad docente.
 - *resp*: identificador del profesor responsable de la unidad docente.
 - **Vigila_examen**: vigilancia de exámenes de asignaturas del departamento realizada por profesores del departamento. El profesor identificado por *nip* ha vigilado el examen de la asignatura identificada por (*ccen*, *ctit*, *casg*) en la convocatoria *conv*.

Gráficamente el esquema relacional es el siguiente:



9 EJERCICIOS SOBRE LA BASE DE DATOS DEPARTAMENTO

9.1 Consultas generales

- Obtener el nombre (*ap1*, *ap2*, *nom*) de los profesores con menor número de créditos de dedicación.

| AP1 | AP2 | NOM |
|----------|-------|---------|
| Castillo | Serra | Alfredo |

```
Select ap1, ap2, nom
from Profesor
where hded IN (select MIN(hded)
               from Profesor);
```

2. ¿Cuántas asignaturas hay cuyo responsable no imparta docencia en ella?

NUM

72

```

Select count(a.nmasg) NUM
from Asignatura a
where a.resp NOT IN (select d.nip
                     from Docencia d
                     where d.nip = a.resp and a.ccen = d.ccen
                     and a.ctit = d.ctit and a.casg = d.casg );

```

3. Obtener el nombre de las asignaturas del centro de código 'ETSINF' y de la titulación 'II' que solicitan más profesores para la vigilancia de exámenes.

NMSG

Ingeniería de la Programación
Lenguajes y Paradigmas de la Programación

```

select nmsg
from asignatura c
where nprofvig in (select max(a.nprofvig)
                  from asignatura a
                  where a.ccen = 'ETSINF' and a.ctit = 'II'
                  and a.ccen = c.ccen and a.ctit = c.ctit);

```

a las

| | | |
|-----------------------------------------------|-----------|-----------|
| Holgura UD Programación | Wun | Máximo |
| Bases de Datos | Caballero | Alejandro |
| Introducción a la Inform. y a la Programación | Calvo | Sava |
| Informática | Granel | José |
| Holgura Responsabilidad Asignaturas Grandes | Aroca | Héctor |
| Tesis de master | Pardo | Alexandro |
| Tesis de master | Herrero | Viviana |
| Tesis de master | Gutiérrez | Sara |
| Tesis | Luis | Isaac |
| Holgura UD Desarrollo del Software | Gorrís | Celia |
| Metodología y Tecnología de la Programación | Ramírez | Monica |
| Holgura UD Programación Básica | Montés | Tadeusz |
| Programación | Ruiz | Enrique |
| Proyecto Fin de Carrera | Ochando | Alfonso |
| Estructuras de Datos y Algoritmos | Izquierdo | Angel |

15 filas seleccionadas

```

Select a.nmasg, p.ap1, p.nom
from Asignatura a, Profesor p
where a.resp = p.nip AND a.gt > 6;

```

5. Obtener el nombre de los grupos de investigación que no tengan profesores.

NOMBRE

Grupo de Planificación y Scheduling
Tecnología Informática
Programación Lógica e Ingeniería del Software

```

Select g.nombre
from Grupo_inv g
where not exists (select *
                  from Profesor p
                  where p.grupo_inv = g.cgi);

```

6. Obtener el nombre (ap1, ap2, nom) de los profesores doctores que no pertenecen a ningún grupo de investigación.

| AP1 | AP2 | NOM |
|----------|------------|---------|
| ----- | ----- | ----- |
| Aguilar | Almiñana | Enric |
| Alabarta | Salvador | Amancio |
| Alonso | Monteagudo | David |
| Ariguel | Fay | Gustavo |
| . . . | | |
| Vila | Quilis | Carmen |
| Vilar | Palop | Mario |
| Zafra | Galisteo | Yaiza |

55 filas seleccionadas

```
select p.ap1, p.ap2, p.nom
from Profesor p
where p.doctor = 1 and p.grupo_inv is null
order by p.ap1;
```

7. Obtener el código y el nombre de los centros que no tienen profesores ni asignaturas.

CCEN NMCEN

```
select ce.ccen, ce.nmcen
from Centro ce
where ce.ccen not in (select p.ccen
                      from Profesor p
                      where p.ccen = ce.ccen)
AND ce.ccen not in (select a.ccen
                    from Asignatura a
                    where a.ccen = ce.ccen);
```

8. Obtener el nombre de los grupos de investigación cuyo profesor responsable también lo es de una unidad docente indicando también el nombre (ap1, nom) de ese profesor.

| NOMBRE | AP1 | NOM |
|----------------------------------------------------|-------|-------|
| ----- | ----- | ----- |
| Teoría de Lenguajes, Computabilidad y Criptografía | Rocha | Julio |

```
select g.nombre, p.ap1, p.nom
from Grupo_inv g, Profesor p, Udocente t
where g.responsable = p.nip and t.resp = g.responsable;
```

9. Obtener el código del centro, el código de la titulación y el nombre de las asignaturas de la unidad docente cuyo responsable es el profesor de nombre 'Pedro Parcet Lorente'.

| CCEN | CTIT | NMASG |
|-------|--------|-----------------------------------------------|
| ----- | ----- | ----- |
| DSIC | HO_UD | Holgura UD Máster IARFID |
| MAS | MIARFI | Programación Gráfica |
| MAS | MIARFI | Planificación de Sistemas Inteligentes de TR |
| . . . | | |
| MAS | MIARFI | Métodos Estadísticos en Tecn. del Lenguaje |
| MAS | MIARFI | Negociación y Confianza en Sistemas Multiagen |

38 filas seleccionadas.

```
select c.ccen, t.ctit, a.nmasg
from Udocente e, Profesor p, Centro c, Titulacion t,
Asignatura a
where e.resp = p.nip
and p.nom = 'Pedro'
and p.ap1 = 'Parcet'
and p.ap2 = 'Lorente'
and p.ccen = c.ccen
and a.udo = e.udo
and t.ctit = a.ctit;
```

10. Obtener el nombre (ap1, ap2, nom) de los profesores que imparten docencia en la titulación de nombre

'Grado en Ing. Inform.' del centro de código 'ETSINF'.

| AP1 | AP2 | NOM |
|----------|-----------|---------|
| ----- | ----- | ----- |
| Cubillo | Prefasi | Carmelo |
| Calvo | Mollá | Sava |
| Galisteo | Risueño | Ana |
| Limorti | Díez | Blas |
| Cárdeno | Gaspar | Carmen |
| Montagud | Salvador | Marc |
| Wun | Sancho | Máximo |
| Ruiz | Verdet | Enrique |
| Valentín | Soriano | Paula |
| Sorli | Hernandez | Aitor |
| Nieves | Jarl | Guillem |
| Nácher | Andújar | Celia |
| Esteban | Tarrega | Xavier |
| Artola | Molino | Ricardo |
| Delgado | Cervantes | Camilla |
| Dolz | Eyob | Marlon |

```
select distinct p.ap1, p.ap2, p.nom
from Profesor p, Docencia d, Titulacion t, Centro c
where t.nomtit = 'Grado en Ing. Inform.'
      and c.ccen = 'ETSINF'
      and d.ctit = t.ctit
      and d.ccen = c.ccen
      and p.nip = d.nip;
```

11. Obtener el código (ccen, ctit, casg) de las asignaturas tales que el profesor de primer apellido 'Escobedo' ha vigilado algún examen de la asignatura.

| CCEN | CTIT | CASG |
|--------|--------|-------|
| ----- | ----- | ----- |
| ETSINF | II | RN |
| MAS | MIARFI | TA |
| MAS | MIARFI | ARN |

```
select distinct a.ccen, a.ctit, a.casg
from Asignatura a, Profesor p, Vigila_examen v
where ap1 = 'Escobedo' and p.nip = v.nip and a.ccen = v.ccen
      and a.ctit = v.ctit and a.casg = v.casg;
```

12. Obtener el código, el nombre y el nombre del responsable (ap1, nom) de los grupos de investigación que son grupos principales de algún otro grupo.

| CGI | NOMBRE | AP1 | NOM |
|-------|-----------------------------------------------|--------|--------|
| ----- | ----- | ----- | ----- |
| GTI | Tecnología Informática | Parcet | Pino |
| PLIS | Programación Lógica e Ingeniería del Software | Lozano | Silvia |

```
select g.cgi, g.nombre, p.ap1, p.nom
from grupo_inv g, profesor p
where p.nip = g.responsable
      and exists (select *
                  from grupo_inv t
                  where t.grupo_princ = g.cgi);
```

13. Obtener, para cada grupo de investigación con profesores, el nombre del grupo y el número de profesores que pertenecen al grupo.

| NOMBRE | PROFS |
|-------------------------------------------------------------|-------|
| ----- | ----- |
| Bases de Datos, Razonamiento Automático y Lenguaje Natural | 8 |
| Inteligencia Artificial | 16 |
| Sistemas Distribuidos | 6 |
| Grupo de Reconocimiento de Formas y Tecnología del Lenguaje | 11 |

| | |
|----------------------------------------------------------|----|
| Grupo de Métodos de Producción de Software | 8 |
| Grupo de Computación Paralela | 16 |
| Modelado Conceptual Orientado a Objetos y Bases de Datos | 5 |
| Informática Gráfica | 10 |
| Teoría de Lenguajes, Computabilidad y Criptografía | 8 |
| Reconocimiento de Formas e Inteligencia Artificial | 10 |
| Extensiones de la Programación Lógica | 12 |
| 11 filas seleccionadas | |

```
select g.nombre, count(p.nip) PROFS
from Grupo_Inv g, Profesor p
where exists (select*
              from Profesor t
              where t.grupo_inv = g.cgi)
and p.grupo_inv = g.cgi
group by g.nombre;
```

14. Obtener, para cada centro que tenga adscritos más de 100 profesores, el código del centro y la cantidad de titulaciones que se imparten en él.

```
CCEN    TITULACIONES
-----
ETSINF  7
```

```
select distinct a.ccen, COUNT(t.ctit) TITULACIONES
from Asignatura a, Titulacion t
where 100 < (select count(nip)
            from profesor
            where ccen = a.ccen)
and a.ctit = t.ctit
group by a.ccen;

NO ES CORRECTOO
```

15. Obtener, para cada categoría de profesor con docencia, la categoría y cuántos son los créditos totales impartidos entre todos sus profesores y cuántos son los profesores que tiene esa categoría.

| CTG | CREDITOS | PROFS |
|------------------------|----------|-------|
| ----- | ----- | ----- |
| ASO3 | 19,2 | 7 |
| ASO4 | 13,41 | 3 |
| ASO6 | 90,65 | 21 |
| AYDOC | 38,3 | 6 |
| BFPI | 3 | 7 |
| BFPU | 0 | 12 |
| COLAB | 4,5 | 9 |
| CONTD | 176,96 | 17 |
| CU | 139,35 | 14 |
| PCOL | 61,9 | 7 |
| TEU | 334,13 | 28 |
| TU | 577,45 | 58 |
| 12 filas seleccionadas | | |

16. Obtener el nombre (ap1, ap2, nom) de los profesores del área 'CC' que no vigilan exámenes de asignaturas en las que no tienen docencia.

| AP1 | AP2 | NOM |
|--------|---------|---------|
| ----- | ----- | ----- |
| Cortes | Maria | Rubén |
| Nieves | Jarl | Guillem |
| Martí | Chorda | Alfredo |
| Ribera | Pascual | Marcos |

| | | |
|--------------------|-----------|----------|
| Aznar | Gutiérrez | Francesc |
| Sanmartin | Hurtado | Yaiza |
| Carbonell | Villarin | Rubén |
| Sorli | Hernandez | Aitor |
| Beristain | Solano | Rafael |
| Miret | Hernández | Jorge |
| Fernandez-Calvillo | Piles | Rafael |
| Gutiérrez | Lairón | Sara |

12 filas seleccionadas

17. Obtener el código y el nombre de los grupos de investigación con profesores tales que todos sus profesores son doctores.

| CGI | NOMBRE |
|-------|-------------------------------------------------------------|
| PRHLT | Grupo de Reconocimiento de Formas y Tecnología del Lenguaje |
| TLCC | Teoría de Lenguajes, Computabilidad y Criptografía |
| OCMDB | Modelado Conceptual Orientado a Objetos y Bases de Datos |
| SIG | Informática Gráfica |

18. Obtener el código y el nombre de las unidades docentes que tienen asignaturas en todos los centros cuyo nombre contiene la cadena 'Inform'.

| UDO | NUDOC |
|-----|----------------------------------|
| CNU | Computación Numerica |
| COM | Computación |
| DSW | Desarrollo Software |
| IAR | Inteligencia Artificial |
| IGR | Informática Gráfica y Multimedia |
| PBA | Programación Básica |
| PRG | Programación |
| SIN | Sistemas Informacion |
| SIS | Sistemas |

9 filas seleccionadas.

19. Obtener el código y el nombre de las unidades docentes que sólo tienen asignaturas en un centro.

| UDO | NUDOC |
|-----|-------------------|
| DOC | Doctorado |
| MAS | Masteres del DSIC |
| SIC | DSIC |

20. Obtener el código (ccen, ctit, casg) de las asignaturas en las que algún profesor imparte más de 2 grupos de prácticas o más de 2 de teoría indicando también el nip de ese profesor y el número total de créditos impartidos por ese profesor en la asignatura.

| CCEN | CTIT | CASG | NIP | CREDITOS |
|--------|--------|------|--------|----------|
| ETSINF | II | PFC | 1234 | 2,3 |
| ETSINF | II | PFC | 2782 | 2,3 |
| ETSINF | II | PFC | 237338 | 2,3 |
| ... | | | | |
| MAS | MISMFS | TES | 425226 | 4 |
| MAS | MCPD | TES | 2424 | 2,7 |
| MAS | MCPD | TES | 3534 | 2,5 |

39 filas seleccionadas.

21. Obtener el nombre (ap1, ap2, nom) de los profesores que vigilan exámenes de más de una unidad docente.

| AP1 | AP2 | NOM |
|----------|----------|---------|
| Albiñana | Lucán | Carla |
| Alvarez | Pozo | Saul |
| Ariguel | Fay | Gustavo |
| ... | | |
| Vilar | Palop | Mario |
| Yusá | Vidaurre | Albert |

Zafra Galisteo Yaiza
86 filas seleccionadas.

22. Obtener el código (ccen, ctit, casg) y el nombre de las asignaturas cuyo nombre contiene la cadena 'Datos' indicando también el nombre (ap1, nom) de cada profesor que la imparte.

| CCEN | CTIT | CASG | NMASG | AP1 | NOM |
|------|--------|------|----------------------------------------------|-----------|---------|
| EPSA | ITIG | APB | Desarrollo de Aplicaciones de Bases de Datos | Chamorro | Felipe |
| EPSA | ITIG | BDA | Bases de Datos | Chamorro | Felipe |
| ... | | | | | |
| MAS | MCALSI | GI | Gestión de Información: Migración de Datos | Leiva | Nicolás |
| MAS | MCPD | GVD | Gestión y Visual. de Datos en Comp. Cient. | Carbonell | Rubén |
| MAS | MISMFS | ISD | Integración Semántica de Datos | Ponz | Marta |

81 filas seleccionadas

23. Obtener el código de las unidades docentes indicando cuántas asignaturas tiene que son impartidas por más de cuatro profesores (sólo si tienen más de una).

UDO NUM
--- ---

PBA 10
COM 4
SIS 3
MIA 2
SIN 5
CNU 4
PRG 9
IAR 2
GAN 3
DSW 6
IGR 2

11 filas seleccionadas.

24. Obtener el código y el nombre de todos los grupos de investigación que hay en la base de datos indicando cuántos profesores tiene cada uno de ellos.

| CGI | NOMBRE | PROFS |
|--------|-------------------------------------------------------------|-------|
| DB | Bases de Datos, Razonamiento Automático y Lenguaje Natural | 8 |
| ELP | Extensiones de la Programación Lógica | 12 |
| GCP | Grupo de Computación Paralela | 16 |
| GPS | Grupo de Planificación y Scheduling | 0 |
| GTI | Tecnología Informática | 0 |
| GTI-IA | Inteligencia Artificial | 16 |
| OOCMDB | Modelado Conceptual Orientado a Objetos y Bases de Datos | 5 |
| OOM | Grupo de Métodos de Producción de Software | 8 |
| PLIS | Programación Lógica e Ingeniería del Software | 0 |
| PRHLT | Grupo de Reconocimiento de Formas y Tecnología del Lenguaje | 11 |
| RFIA | Reconocimiento de Formas e Inteligencia Artificial | 10 |
| SiDi | Sistemas Distribuidos | 6 |
| SIG | Informática Gráfica | 10 |
| TLCC | Teoría de Lenguajes, Computabilidad y Criptografía | 8 |

14 filas seleccionadas

25. Obtener el código y el nombre de todos los centros que hay en la base de datos indicando cuántos profesores doctores tienen adscritos.

| CCEN | NMCEN | DOCS |
|--------|----------------------------------------------------|------|
| — | Sin centro asignado | 7 |
| ADE | Facultad de Administración y Direc. de Empresas | 0 |
| DSIC | Dpto. Sistemas Informáticos y Computación | 6 |
| EPSA | E. Politécnica Superior de Alcoy | 3 |
| EPSG | E. Politécnica Superior de Gandia | 5 |
| ETSCCP | E.T.S. de Ingenieros de Caminos, Canales y Puertos | 0 |
| ETSIA | E.T.S. de Ingenieros Agrónomos | 0 |
| ETSID | E.T.S. de Ingeniería del Diseño | 1 |
| ETSII | E.T.S. de Ingenieros Industriales | 5 |

| | | |
|------------------------|------------------------------------------------------------|-----|
| ETSINF | E.T.S. de Ingeniería Informática | 110 |
| ETSIT | E.T.S. de Ingenieros de Telecomunicación | 2 |
| ETSITG | E.T.S. de Ingeniería Geodésica, Cartográfica y Topográfica | 0 |
| EUITAG | E.U. de Ingeniería Técnica Agrícola | 0 |
| MAS | Master Oficiales | 0 |
| RECT | Rectorado | 0 |
| 15 filas seleccionadas | | |

26. Obtener el nombre (ap1, ap2, nom) de los profesores que imparten todas las asignaturas de una unidad docente.

| AP1 | AP2 | NOM |
|-------------------------|----------|-----------|
| ----- | ----- | ----- |
| Caballero | Mondejar | Alejandro |
| Pedreira | Arraez | Antony |
| Calvo | Mollá | Sava |
| Alegre | Rocha | Moises |
| Gorrís | Arastey | Celia |
| Beristain | Solano | Rafael |
| Quintero | Benito | Yaiza |
| Ramírez | Torres | Monica |
| Ruiz | Verdet | Enrique |
| Izquierdo | Alarcon | Angel |
| 10 filas seleccionadas. | | |

27. Obtener el código (ccen, ctit, casg) y el nombre de todas las asignaturas de la base de datos que pertenezca a una unidad docente con más de 50 asignaturas indicando para cada asignatura cuántos profesores imparten docencia en ella.

| CCEN | CTIT | CASG | NMASG | TOTAL |
|------------------------|-------|-------|------------------------------------|-------|
| ----- | ----- | ----- | ----- | ----- |
| ADE | LADE | OFI | Ofimática para ADE | 0 |
| ETSID | IAREO | Inr | Informática Aplicada (aeronáutico) | 0 |
| ... | | | | |
| ETSII | GITI | INF | Informática | 10 |
| ETSII | SGITI | INF | SD Informática | 10 |
| 52 filas seleccionadas | | | | |

28. Obtener el nombre (ap1, ap2, nom) de todos los profesores doctores responsables de alguna unidad docente indicando también la cantidad de asignaturas que imparte que tengan más de 2 créditos de teoría.

| AP1 | AP2 | NOM | ASIGS |
|------------------------|-----------|-----------|-------|
| ----- | ----- | ----- | ----- |
| Yudici | Cosme | Alexandro | 0 |
| Miret | Hernández | Jorge | 1 |
| Solano | Bonora | Sava | 4 |
| Herrero | Sáez | Viviana | 2 |
| Gorrís | Arastey | Celia | 1 |
| Castelló | Pedros | Sergio | 2 |
| Perez | Alejo | Gracia | 2 |
| Parcet | Lorente | Pedro | 2 |
| Calvo | Margaix | Mario | 3 |
| Rocha | Piles | Julio | 4 |
| Leiva | Luna | Nicolás | 5 |
| Aroca | Ginestar | Héctor | 1 |
| Urgessa | Saez | Pablo | 7 |
| Yusá | Vidaurre | Albert | 2 |
| Quintero | Benito | Yaiza | 2 |
| 15 filas seleccionadas | | | |

29. Obtener el código y el nombre de todos los grupos de investigación que hay en la base de datos indicando cuántos profesores doctores tiene y cuántos no doctores.

| CGI | NOMBRE | DOCS | NODOCS |
|-------|------------------------------------------------------------|-------|--------|
| ----- | ----- | ----- | ----- |
| DB | Bases de Datos, Razonamiento Automático y Lenguaje Natural | 6 | 2 |
| ELP | Extensiones de la Programación Lógica | 9 | 3 |
| GCP | Grupo de Computación Paralela | 10 | 6 |

| | | | |
|------------------------|-------------------------------------------------------------|----|---|
| GPS | Grupo de Planificación y Scheduling | 0 | 0 |
| GTI | Tecnología Informática | 0 | 0 |
| GTI-IA | Inteligencia Artificial | 15 | 1 |
| OOCMDB | Modelado Conceptual Orientado a Objetos y Bases de Datos | 5 | 0 |
| OOM | Grupo de Métodos de Producción de Software | 6 | 2 |
| PLIS | Programación Lógica e Ingeniería del Software | 0 | 0 |
| PRHLT | Grupo de Reconocimiento de Formas y Tecnología del Lenguaje | 11 | 0 |
| RFIA | Reconocimiento de Formas e Inteligencia Artificial | 9 | 1 |
| SiDi | Sistemas Distribuidos | 4 | 2 |
| SIG | Informática Gráfica | 10 | 0 |
| TLCC | Teoría de Lenguajes, Computabilidad y Criptografía | 8 | 0 |
| 14 filas seleccionadas | | | |

30. Obtener el código (ccen, ctit, casg) de todas las asignaturas de la base de datos indicando el número total de créditos de docencia que tienen asignados a profesores. Si una asignatura no tiene docencia el número de créditos debe ser 0.

| CCEN | CTIT | CASG | TOTAL |
|--------------------------|--------|------|-------|
| ----- | ----- | ---- | ----- |
| ADE | GAP | IDA | 0 |
| ADE | GGAP | IAP | 9 |
| ADE | LADE | INF | 7,8 |
| . . . | | | |
| MAS | MISMFS | TSD | 0 |
| MAS | MISMFS | TSW | 3 |
| MAS | MPD | ACO | 2 |
| 310 filas seleccionadas. | | | |

31. ¿Cuántos centros hay con menor número de asignaturas que de profesores adscritos?

| CENTROS |
|---------|
| ----- |
| 4 |

32. Obtener el código (ccen, ctit, casg) y el nombre de la asignatura en la que más profesores imparten docencia.

| CCEN | CTIT | CASG | NMASG |
|--------|-------|------|-------------------------|
| ----- | ----- | ---- | ----- |
| ETSINF | II | PFC | Proyecto Fin de Carrera |

33. Obtener el nombre (ap1, ap2, nom) de cada profesor que pertenezca a un grupo de investigación con más de siete miembros indicando también en cuántos centros imparten docencia al impartir una asignatura de ese centro.

| AP1 | AP2 | NOM | CENTROS |
|-------------------------|----------|-----------|---------|
| ----- | ----- | ----- | ----- |
| Alegre | Rocha | Moises | 3 |
| Alvarez | Pozo | Saul | 3 |
| Anton | Alvaro | Roberto | 2 |
| . . . | | | |
| Wun | Sancho | Máximo | 2 |
| Yudici | Cosme | Alexandro | 1 |
| Yusá | Vidaurre | Albert | 3 |
| 99 filas seleccionadas. | | | |

34. Obtener el código (ccen, ctit, casg) y el nombre de las asignaturas de carácter no obligatorio que solicitan más profesores para la vigilancia de exámenes.

| CCEN | CTIT | CASG | NMASG |
|--------|-------|------|--------------|
| ----- | ----- | ---- | ----- |
| ETSINF | GIINF | PRG | Programación |

35. Obtener el nombre (ap1, ap2, nom) de los profesores que imparten más de 20 créditos de docencia.

| AP1 | AP2 | NOM |
|-----------|---------|---------|
| ----- | ----- | ----- |
| Ferrero | Puertes | Pino |
| Beristain | Solano | Rafael |
| Barón | López | Enrique |
| Montés | Robles | Tadeusz |

| | | |
|---------|---------|---------|
| Artola | Molino | Ricardo |
| Carmona | Navalón | David |

6 filas seleccionadas.

36. Obtener el código del centro, código de la titulación y código de la asignatura de aquellas asignaturas con mayor número de créditos por grupo de teoría.

| | | |
|--------|--------|------|
| CCEN | CTIT | CASG |
| ----- | ----- | ---- |
| ETSINF | SGIINF | IIP |
| ETSINF | SGIINF | PRG |

37. Obtener la cantidad de profesores que hay sin grupo de investigación asignado.

| |
|-------|
| PROFS |
| ----- |
| 329 |

38. Obtener el nombre (ap1, ap2, nom) de los profesores que imparten docencia en alguna de las asignaturas que pertenezcan a la titulación de código 'II' del centro de código 'ETSINF' y tal que la asignatura tenga más de 70 alumnos matriculados.

| | | |
|-----------|-----------|----------|
| AP1 | AP2 | NOM |
| ----- | ----- | ----- |
| Casamayor | Millet | Ricard |
| Mateu | Cosme | Eva |
| Aznar | Gutiérrez | Francesc |
| ... | | |
| Aroca | Ginestar | Héctor |
| Fuambuena | Carbonell | Héctor |
| Albiñana | Lucán | Carla |

21 filas seleccionadas

39. Obtener el nombre (ap1, ap2, nom) de los profesores doctores que imparten docencia en alguna asignatura del centro de código 'ETSINF'.

| | | |
|----------|----------|--------|
| AP1 | AP2 | NOM |
| ----- | ----- | ----- |
| Albiñana | Lucán | Carla |
| Alegre | Rocha | Moises |
| Alvarez | Pozo | Saul |
| ... | | |
| Verdet | Gomez | Jorge |
| Vilar | Palop | Mario |
| Zafra | Galisteo | Yaiza |

106 filas seleccionadas.

40. Obtener el código del centro, el código de la titulación y el código de las asignaturas de la unidad docente cuyo responsable es el profesor de primer apellido 'Parcet' sólo si éste vigila algún examen de dicha asignatura.

| | | |
|-------|--------|------|
| CCEN | CTIT | CASG |
| ----- | ----- | ---- |
| MAS | MIARFI | PIA |

41. Obtener el nombre de las asignaturas impartidas en la titulación de nombre 'Grado en Ing. Inform.' del centro de código 'ETSINF'.

| |
|-----------------------------------------------|
| NMASG |
| ----- |
| Introducción a la Inform. y a la Programación |
| Programación |

42. Obtener, para todos y cada uno de los centros, el código de centro, la cantidad de profesores adscritos al mismo y la cantidad de titulaciones en las que el centro está impartiendo docencia.

| | | |
|-------|-------|--------------|
| CCEN | PROFS | TITULACIONES |
| ----- | ----- | ----- |
| --- | 59 | 0 |
| ADE | 0 | 4 |
| DSIC | 84 | 2 |
| EPSA | 12 | 3 |

| | | |
|--------|-----|---|
| EPSC | 8 | 6 |
| ETSCCP | 3 | 7 |
| ETSIA | 0 | 0 |
| ETSID | 1 | 9 |
| ETSII | 6 | 9 |
| ETSINF | 168 | 7 |
| ETSIT | 2 | 4 |
| ETSITG | 1 | 3 |
| EUITAG | 0 | 0 |
| MAS | 0 | 8 |
| RECT | 2 | 0 |

15 filas seleccionadas

43. Obtener, para cada unidad docente con asignaturas adscritas, el código de la unidad docente, el nombre (ap1, ap2, nom) del profesor responsable y la cantidad de asignaturas adscritas a esa unidad docente.

| UDO | AP1 | AP2 | NOM | ASIGS |
|-----|----------|----------|-----------|-------|
| --- | ----- | ----- | ----- | ----- |
| ALC | Yudici | Cosme | Alexandro | 18 |
| CNU | Yusá | Vidaurre | Albert | 14 |
| COM | Rocha | Piles | Julio | 10 |
| | ... | | | |
| SIC | Aroca | Ginestar | Héctor | 1 |
| SIN | Leiva | Luna | Nicolás | 18 |
| SIS | Quintero | Benito | Yaiza | 12 |

18 filas seleccionadas.

44. Obtener, para cada unidad docente con dos o más asignaturas adscritas, el código de la unidad docente, el nombre (ap1, ap2, nom) del profesor responsable y la cantidad de profesores distintos que imparten algún grupo de teoría de las asignaturas adscritas a esa unidad.

| UDO | AP1 | AP2 | NOM | PROFS |
|-----|----------|----------|-----------|-------|
| --- | ----- | ----- | ----- | ----- |
| ALC | Yudici | Cosme | Alexandro | 8 |
| CNU | Yusá | Vidaurre | Albert | 12 |
| COM | Rocha | Piles | Julio | 11 |
| ... | | | | |
| PRG | Wun | Sancho | Máximo | 22 |
| SIN | Leiva | Luna | Nicolás | 21 |
| SIS | Quintero | Benito | Yaiza | 16 |

15 filas seleccionadas

45. Obtener la cantidad de asignaturas con más créditos teóricos que prácticos que son impartidas por al menos un profesor no doctor.

ASIGS

99

46. Obtener el código (ccen, ctit, casg) y nombre de las asignaturas de carácter obligatorio ('OB'), junto con la cantidad de profesores distintos que vigilan exámenes de dichas asignaturas (sólo si esta cantidad es mayor o igual a 1).

| CCEN | CTIT | CASG | NMASG | ASIGS |
|--------|--------|-------|-----------------------------------------------|-------|
| ----- | ----- | ----- | ----- | ----- |
| ADE | LADE | INF | Introducción a la informática | 3 |
| EPSG | GTU | IG | Informática de Gestión | 3 |
| ETSCCP | GIC | PMN | Conocim. básicos de prog. y métodos numéricos | 5 |
| ... | | | | |
| ETSINF | LD | EDA | Estructuras de datos y Algoritmos | 1 |
| ETSINF | LD | FDP | Fundamentos de Programación | 1 |
| MAS | MISMFS | MDI | Modelado, Diseño e Imp. de Servicios Web | 1 |

17 filas seleccionadas.

47. Obtener los apellidos y el nombre (ap1, ap2, nom) de los profesores con mayor número de asignaturas impartidas.

| AP1 | AP2 | NOM |
|-----------|--------|--------|
| ----- | ----- | ----- |
| Casamayor | Millet | Ricard |

48. Obtener el código y el nombre de los centros con menos profesores adscritos de la categoría 'CU'.

| CCEN | NMCEN |
|--------|------------------------------------------------------------|
| ----- | ----- |
| _ | Sin centro asignado |
| ADE | Facultad de Administración y Direc. de Empresas |
| DSIC | Dpto. Sistemas Informáticos y Computación |
| EPSA | E. Politécnica Superior de Alcoy |
| EPSG | E. Politécnica Superior de Gandia |
| ETSCCP | E.T.S. de Ingenieros de Caminos, Canales y Puertos |
| ETSIA | E.T.S. de Ingenieros Agrónomos |
| ETSID | E.T.S. de Ingeniería del Diseño |
| ETSII | E.T.S. de Ingenieros Industriales |
| ETSIT | E.T.S. de Ingenieros de Telecomunicación |
| ETSITG | E.T.S. de Ingeniería Geodésica, Cartográfica y Topográfica |
| EUITAG | E.U. de Ingeniería Técnica Agrícola |
| MAS | Master Oficiales |
| RECT | Rectorado |

14 filas seleccionadas

49. Obtener los apellidos y el nombre (ap1, ap2, nom) de los profesores que vigilan algún de examen de la asignatura cuyo código de centro es 'ETSINF', su código de titulación es 'ITIG', y su código de asignatura es 'DBD'.

| AP1 | AP2 | NOM |
|-----------|---------|---------|
| ----- | ----- | ----- |
| Vazquez | Angulo | Alba |
| Vara | Munuera | Marina |
| Casamayor | Millet | Ricard |
| Ayora | Forteza | Modesto |
| Mollá | Gurrea | Héctor |

50. Obtener el código y el nombre de los grupos de investigación que no tengan grupo principal.

| CGI | NOMBRE |
|------------------------|-------------------------------------------------------------|
| ----- | ----- |
| GCP | Grupo de Computación Paralela |
| PLIS | Programación Lógica e Ingeniería del Software |
| RFIA | Reconocimiento de Formas e Inteligencia Artificial |
| SiDi | Sistemas Distribuidos |
| GTI | Tecnología Informática |
| TLCC | Teoría de Lenguajes, Computabilidad y Criptografía |
| PRHLT | Grupo de Reconocimiento de Formas y Tecnología del Lenguaje |
| 7 filas seleccionadas. | |

51. Obtener el nombre (ap1, ap2, nom) de los profesores que no vigilan ningún examen.

| AP1 | AP2 | NOM |
|--------------------------|------------|-------------|
| ----- | ----- | ----- |
| Abadías | Gutiérrez | Constantino |
| Abellán | Roman | Andres |
| Adalid | Montserrat | Marc |
| ... | | |
| Vizcarro | Barrachina | Cristina |
| Wun | Canto | Francesc |
| Yudici | Cosme | Alexandro |
| 269 filas seleccionadas. | | |

52. Obtener el nip, la categoría y el número total de grupos (teóricos o prácticos) de los profesores que imparten un mayor número de grupos (teóricos o prácticos).

| NIP | CTG | GRUPOS |
|-------|-------|--------|
| ----- | ----- | ----- |
| 2782 | TU | 19,3 |

53. Obtener el nombre (ap1, ap2, nom) de los profesores que imparten todas las asignaturas de una titulación con al menos una asignatura.

| AP1 | AP2 | NOM |
|------------------------|----------|---------|
| ----- | ----- | ----- |
| Agut | Forteza | Joan |
| Alvarez | Pozo | Saul |
| Anton | Alvaro | Roberto |
| ... | | |
| Urgessa | Saez | Pablo |
| Valiño | Limorti | Joan |
| Yusá | Vidaurre | Albert |
| 54 filas seleccionadas | | |

54. Obtener el código y el nombre de los centros que no tienen asignaturas adscritas.

| CCEN | NMCEN |
|--------|-------------------------------------|
| ----- | ----- |
| ETSIA | E.T.S. de Ingenieros Agrónomos |
| EUITAG | E.U. de Ingeniería Técnica Agrícola |
| RECT | Rectorado |
| _ | Sin centro asignado |

55. Obtener el nombre de todos los centros indicando cuántos profesores tienen adscritos.

| NMCEN | PROFS |
|-------------------------------------------|-------|
| ----- | ----- |
| Dpto. Sistemas Informáticos y Computación | |

| | |
|------------------------------------------------------------|-----|
| E. Politécnica Superior de Alcoy | 12 |
| E. Politécnica Superior de Gandia | 8 |
| E.T.S. de Ingeniería del Diseño | 1 |
| E.T.S. de Ingeniería Geodésica, Cartográfica y Topográfica | 1 |
| E.T.S. de Ingeniería Informática | 168 |
| E.T.S. de Ingenieros Agrónomos | 0 |
| E.T.S. de Ingenieros de Caminos, Canales y Puertos | 3 |
| E.T.S. de Ingenieros de Telecomunicación | 2 |
| E.T.S. de Ingenieros Industriales | 6 |
| E.U. de Ingeniería Técnica Agrícola | 0 |
| Facultad de Adiministración y Direc. de Empresas | 0 |
| Master Oficiales | 0 |
| Rectorado | 2 |
| Sin centro asignado | 59 |
| 15 filas seleccionadas | |

56. Obtener el código y el nombre de las unidades docentes que tienen asignaturas en todos los centros cuyo nombre contiene la cadena 'Inform'.

UDO NUDOC

CNU Computación Numerica
 COM Computación
 DSW Desarrollo Software
 IAR Inteligencia Artificial
 IGR Informática Gráfica y Multimedia
 PBA Programación Básica
 PRG Programación
 SIN Sistemas Informacion
 SIS Sistemas
 9 filas seleccionadas.

57. Obtener el nombre de los grupos de investigación tales que todos los profesores adscritos al grupo sean de categoría distinta a 'TEU'.

NOMBRE

Grupo de Reconocimiento de Formas y Tecnología del Lenguaje

10 PRESENTACIÓN DE LA BASE DE DATOS CICLISMO

Se desea mantener información de una vuelta ciclista; para ello se ha definido una base de datos relacional cuyo esquema se muestra a continuación. Este es el mismo esquema que el presentado en la UD2.1.

EQUIPO(nomeq:char(25),director:char(30))
 CP:{nomeq}

CICLISTA(dorsal:entero,nombre:char(30),edad:entero,nomeq:char(25))
 CP:{dorsal}
 CAj:{nomeq}→EQUIPO
 VNN:{nomeq}
 VNN:{nombre}

ETAPA(netapa:entero,km:entero,salida:char(35),llegada:char(35),
 dorsal:entero)
 CP:{netapa}
 CAj:{dorsal}→CICLISTA

MAILLOT(codigo:char(3),tipo:char(30),premio:entero,color:char(25))
 CP:{codigo}

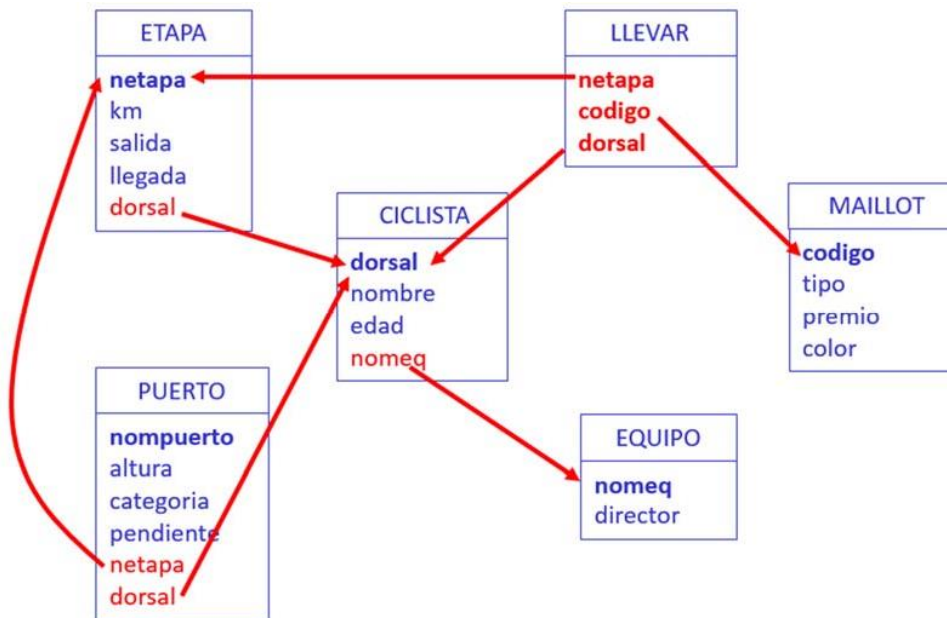
PUERTO (nompuerto:char(30), altura:entero, categoria:char(1), pendiente:real, netapa:entero, dorsal:entero)
 CP:{nompuerto}
 CAj:{netapa}→ETAPA
 CAj:{dorsal}→CICLISTA
 VNN:{netapa}

LLEVAR (dorsal:entero, netapa:entero, codigo:char(3))
 CP:{netapa, codigo}
 CAj:{netapa}→ETAPA
 CAj:{dorsal}→CICLISTA
 CAj:{codigo}→MAILLOT
 VNN:{dorsal}

Para aclarar por completo el esquema, a continuación se explica el significado de cada atributo.

- **Equipo:**
 - *nomeq*: cómo se llama el equipo ciclista.
 - *director*: nombre del preparador técnico del equipo.
- **Ciclista:**
 - *dorsal*: nº de dorsal asignado al ciclista durante la carrera.
 - *nombre*: cómo se llama el corredor.
 - *edad*: cuántos años tiene.
 - *nomeq*: nombre del equipo al que pertenece.
- **Etapa:**
 - *netapa*: número de la etapa en la vuelta.
 - *km*: cuántos kilómetros tiene la etapa.
 - *salida*: nombre de la ciudad de donde parte la etapa.
 - *llegada*: nombre de la ciudad donde está la meta de la etapa.
 - *dorsal*: dorsal del ciclista que ha ganado la etapa.
- **Puerto:**
 - *nompuerto*: cómo se llama el puerto de montaña.
 - *altura*: altura máxima del puerto.
 - *categoria*: cuál es la categoría del puerto (1ª, especial, ...).
 - *pendiente*: % de pendiente media del puerto.
 - *netapa*: número de la etapa donde se sube el puerto.
 - *dorsal*: dorsal ciclista que ha ganado el puerto al pasar en primera posición.
- **Maillot:**
 - *codigo*: código del maillot.
 - *tipo*: indica qué clasificación premia ese maillot.
 - *color*: cómo es la camiseta asociada a ese premio.
 - *premio*: cuánto dinero gana el ciclista que acabe la vuelta con ese maillot.
- **Llevar**: el ciclista con dorsal *dorsal* ha llevado en la etapa de número *netapa* el maillot identificado por *codigo*.

Gráficamente el esquema relacional es el siguiente:



11 EJERCICIOS SOBRE LA BASE DE DATOS CICLISMO

11.1 Consultas sobre una sola relación

1. Obtener el código, el tipo, el color y el premio de todos los maillots que hay.

| COD TIPO | COLOR | PREMIO |
|------------------------|---------------------|---------|
| MGE General | Amarillo | 8000000 |
| MMO Montaña | Blanco y Rojo | 2000000 |
| MMS Mas Sufrido | Estrellitas moradas | 2000000 |
| MMV Metas volantes | Rojo | 2000000 |
| MRE Regularidad | Verde | 2000000 |
| MSE Sprints especiales | Rosa | 2000000 |
| 6 filas seleccionadas. | | |

```
Select codigo,tipo, color, premio
From maillot;
```

2. Obtener el dorsal y el nombre de los ciclistas cuya edad sea menor o igual que 25 años.

| DORSAL | NOMBRE |
|------------------------|-------------------|
| 38 | Javier Palacin |
| 41 | Rolf Aldag |
| 46 | Agustin Sagasti |
| 49 | Eugeni Berzin |
| 66 | Enrico Zaina |
| 98 | Eleuterio Anguita |
| 6 filas seleccionadas. | |

```
Select dorsal, nombre
From ciclista
Where edad <= 25;
```


3. Obtener el nombre y la altura de todos los puertos de categoría 'E' (Especial).

| NOMPUERTO | ALTURA |
|------------------------|--------|
| Arcalis | 2230 |
| Cerler-Circo de Ampriu | 2500 |
| Coll de Ordino | 1980 |
| Cruz de la Demanda | 1850 |
| Lagos de Covadonga | 1134 |
| Sierra Nevada | 2500 |

6 filas seleccionadas.

```
Select nompuerto, altura
From puerto
Where categoria = 'E';
```

4. Obtener el valor del atributo netapa de aquellas etapas con salida y llegada en la misma ciudad.

| NETAPA |
|--------|
| 1 |
| 8 |
| 18 |

```
Select netapa
From etapa
Where salida = llegada;
```

5. ¿Cuántos ciclistas hay?

| CICLISTAS |
|-----------|
| 100 |

```
select count(*)
from ciclista;
```

6. ¿Cuántos ciclistas hay con edad superior a 25 años?

| CICLISTAS |
|-----------|
| 94 |

```
Select count(*)
From ciclista
Where edad > 25;
```

7. ¿Cuántos equipos hay?

| COUNT (*) |
|-----------|
| 22 |

```
select count(*)
from equipo;
```

8. Obtener la media de edad de los ciclistas.

| AVG (EDAD) |
|------------|
| 29,89 |

```
select avg(edad)
from ciclista;
```

9. Obtener la altura mínima y máxima de los puertos de montaña.

| MIN (ALTURA | MAX (ALTURA |
|-------------|-------------|
| 565 | 2500 |

```
select min(altura), max(altura)
from puerto;
```

11.2 Consultas sobre varias relaciones**10. Obtener el nombre y la categoría de los puertos ganados por ciclistas del equipo 'Banesto'.**

| NOMPUERTO | C |
|-----------------------|---|
| Alto del Naranco | 1 |
| Coll de la Comella | 1 |
| Navacerrada | 1 |
| Puerto de Alisas | 1 |
| Puerto de la Morcuera | 2 |
| Puerto de Navalmoral | 2 |
| Sierra Nevada | E |

7 filas seleccionadas.

```
select nompuerto, categoria C
from puerto p, ciclista c
where p.dorsal = c.dorsal and nomeq = 'Banesto';
```

11. Obtener el nombre del cada puerto indicando el número (netapa) y los kilómetros de la etapa en la que se encuentra el puerto.

| NOMPUERTO | NETAPA | KM |
|--------------------------|--------|-----|
| Alto del Naranco | 10 | 200 |
| Arcalis | 10 | 200 |
| Cerler-Circo de Ampriu | 11 | 195 |
| Coll de la Comella | 10 | 200 |
| Coll de Ordino | 10 | 200 |
| Cruz de la Demanda | 11 | 195 |
| Lagos de Covadonga | 16 | 160 |
| Navacerrada | 19 | 190 |
| Puerto de Alisas | 15 | 207 |
| Puerto de la Morcuera | 19 | 190 |
| Puerto de Mijares | 18 | 195 |
| Puerto de Navalmoral | 18 | 195 |
| Puerto de Pedro Bernardo | 18 | 195 |
| Sierra Nevada | 2 | 180 |

14 filas seleccionadas

```
select p.nompuerto, p.netapa, km
from puerto p, etapa e
where p.netapa = e.netapa;
```

12. Obtener el nombre y el director de los equipos a los que pertenezca algún ciclista mayor de 33 años.

| NOMEQ | DIRECTOR |
|-----------------|-------------------|
| Amore Vita | Ricardo Padacci |
| Banesto | Miguel Echevarria |
| Bresciali-Refin | Pietro Armani |
| Carrera | Luigi Petroni |
| Gatorade | Gian Luca Pacceli |
| Kelme | Álvaro Pino |
| Mapei-Clas | Juan Fernandez |
| Navigare | Lonrenzo Sciacchi |

TVM Steveens Henk
 Telecom Morgan Reikcard
 10 filas seleccionadas.

```
select distinct e.nomeq, director
from equipo e, ciclista c
where e.nomeq = c.nomeq and edad>33
order by e.nomeq;
```

13. Obtener el nombre de los ciclistas con el color de cada maillot que hayan llevado.

| NOMBRE | COLOR |
|---------------------|----------|
| ----- | ----- |
| Alessio Di Basco | Rosa |
| Alex Zulle | Amarillo |
| Alfonso Gutiérrez | Rojo |
| ... | |
| Per Pedersen | Rosa |
| Stefano della Santa | Rojo |
| Tony Rominger | Amarillo |

31 filas seleccionadas.

```
select distinct nombre, color
from ciclista c, llevar l, maillot m
where c.dorsal = l.dorsal and l.codigo = m.codigo
order by nombre;
```

14. Obtener pares de nombre de ciclista y número de etapa tal que ese ciclista haya ganado esa etapa y haya llevado el maillot de color 'Amarillo' en alguna etapa.

| NOMBRE | NETAPA |
|-----------------|--------|
| ----- | ----- |
| Miguel Induráin | 1 |
| Miguel Induráin | 8 |
| Pedro Delgado | 10 |
| Pedro Delgado | 19 |
| Pedro Delgado | 20 |
| Tony Rominger | 17 |

6 filas seleccionadas.

```
select distinct c.nombre, e.netapa
from etapa e, ciclista c, llevar l, maillot m
where e.dorsal = c.dorsal
      and c.dorsal = l.dorsal
      and l.codigo = m.codigo
      and color = 'Amarillo'
order by c.nombre;
```

15. Obtener el valor del atributo netapa de las etapas que no comienzan en la misma ciudad en que acabó la anterior etapa.

NETAPA

| |
|-------|
| ----- |
| 4 |
| 7 |
| 8 |
| 10 |
| 12 |
| 15 |
| 17 |
| 18 |
| 20 |

9 filas seleccionadas.

```
select e.netapa
from etapa e, etapa d
where d.netapa = e.netapa - 1 and d.llegada <> e.salida;
```

11.3 Consultas con subconsultas

16. Obtener el valor del atributo netapa y la ciudad de salida de aquellas etapas que no tengan puertos de montaña.

| NETAPA | SALIDA |
|--------|--------------------|
| ----- | ----- |
| | 1 Valladolid |
| | 2 Salamanca |
| | 3 Almendralejo |
| | 4 Córdoba |
| | 5 Granada |
| | 6 Baza |
| | 7 Benidorm |
| | 8 Benidorm |
| | 12 Benasque |
| | 13 Zaragoza |
| | 14 Pamplona |
| | 17 Cangas de Onis |
| | 20 Segovia |
| | 21 Destilerias Dyc |

14 filas seleccionadas

```
select netapa, salida
from etapa
where netapa not in (select netapa
                    from puerto);
```

17. Obtener la edad media de los ciclistas que han ganado alguna etapa.

| MEDIA |
|---------|
| ----- |
| 30,5625 |

```
select avg(edad)
from ciclista
where dorsal in (select dorsal
                from etapa);
```

18. Selecciona el nombre de los puertos con una altura superior a la altura media de todos los puertos.

| NOMPUERTO |
|------------------------|
| ----- |
| Arcalis |
| Cerler-Circo de Ampriu |
| Coll de Ordino |
| Cruz de la Demanda |
| Navacerrada |
| Puerto de la Morcuera |
| Sierra Nevada |

7 filas seleccionadas.

```
select nompuerto
from puerto
where altura > (select avg(altura)
                from puerto);
```

19. Obtener el nombre de la ciudad de salida y de llegada de las etapas donde estén los puertos con mayor pendiente.

| SALIDA | LLEGADA |
|----------|---------|
| ----- | ----- |
| Igualada | Andorra |

```
select salida, llegada
from etapa
where netapa in (select netapa
                 from puerto
                 where pendiente = (select max(pendiente)
                                   from puerto));
```

20. Obtener el dorsal y el nombre de los ciclistas que han ganado los puertos de mayor altura.

DORSAL NOMBRE

 9 Massimo Podenzana
 26 Mikel Zarrabeitia

```
select dorsal, nombre
from ciclista
where dorsal in (select dorsal
                  from puerto
                  where altura = (select max(altura)
                                  from puerto))
order by dorsal;
```

21. Obtener el nombre del ciclista más joven.

NOMBRE

 Eugeni Berzin

```
select nombre
from ciclista
where edad = (select min(edad)
               from ciclista);
```

22. Obtener el nombre del ciclista más joven que ha ganado al menos una etapa.

NOMBRE

 Vladislav Bobrik

```
select nombre
from ciclista
where dorsal in (select dorsal
                  from etapa)
and edad = (select min(edad)
             from ciclista
             where dorsal in (select dorsal
                               from etapa));
```

23. Obtener el nombre de los ciclistas que han ganado más de un puerto.

NOMBRE

 Pedro Delgado

```
select nombre
from ciclista c
where 1 < (select count(nompuerto)
           from puerto
           where dorsal = c.dorsal);
```

11.4 Consultas con cuantificación universal

24. Obtener el valor del atributo netapa de aquellas etapas tales que todos los puertos que están en ellas tienen más de 700 metros de altura.

NETAPA

 2
 11
 16
 18
 19

```
select netapa
from etapa e
where not exists (select *
                  from puerto
                  where netapa = e.netapa
                  and altura < 700)
and exists (select *
            from puerto
            where netapa = e.netapa)
order by netapa;
```

25. Obtener el nombre y el director de los equipos tales que todos sus ciclistas son mayores de 25 años.

| NOMEQ | DIRECTOR |
|-----------------|-------------------|
| Amore Vita | Ricardo Padacci |
| Banesto | Miguel Echevarria |
| Bresciali-Refin | Pietro Armani |
| Carrera | Luigi Petroni |
| Castorama | Jean Philip |
| Gatorade | Gian Luca Pacceli |
| Jolly Club | Johan Richard |
| Kelme | Álvaro Pino |
| Lotus Festina | Suarez Cuevas |
| Mapei-Clas | Juan Fernandez |
| Mercatone Uno | Ettore Romano |
| Motorola | John Fidwell |
| Navigare | Lonrenzo Sciacchi |
| ONCE | Manuel Sainz |
| Seguros Amaya | Minguez |
| TVM | Steveens Henk |
| Wordperfect | Bill Gates |

17 filas seleccionadas.

```
select nomeq, director
from equipo e
where not exists (select *
                  from ciclista
                  where nomeq = e.nomeq
                  and edad <= 25)
and exists (select *
            from ciclista
            where nomeq = e.nomeq)
order by nomeq;
```

26. Obtener el dorsal y el nombre de los ciclistas tales que todas las etapas que han ganado tienen más de 170 km (es decir que sólo han ganado etapas de más de 170 km).

| DORSAL | NOMBRE |
|--------|----------------------|
| 8 | Jean Van Poppel |
| 10 | Mario Cipollini |
| 12 | Alessio Di Basco |
| 22 | Giorgio Furlan |
| 36 | Gian Matteo Fagnini |
| 65 | Pascal Lino |
| 83 | Hernan Buenahora |
| 86 | Juan Martinez Oliver |
| 93 | Bo Hamburger |

9 filas seleccionadas.

```
select dorsal, nombre
from ciclista c
where not exists (select *
                  from etapa
                  where dorsal = c.dorsal
                  and km <= '170')
and exists (select *
            from etapa
            where dorsal = c.dorsal)
order by dorsal;
```

27. Obtener el nombre de los ciclistas que han ganado todos los puertos de una etapa y además han ganado esa misma etapa.

NOMBRE

Pedro Delgado

```
select c.nombre
from Ciclista c, etapa e
where e.dorsal = c.dorsal
and not exists (select*
                from Puerto p
                where p.netapa = e.netapa
                and p.dorsal <> c.dorsal)
and exists (select*
            from Puerto p
            where p.netapa = e.netapa);
```

28. Obtener el nombre de los equipos tales que todos sus corredores han llevado algún maillot o han ganado algún puerto.

NOMEQ

Castorama

```
select e.nomeq
from Equipo e
where not exists (select*
                  from Ciclista c
                  where c.nomeq = e.nomeq
                  and not exists (select*
                                  from Llevar l, Maillot m
                                  where l.dorsal = c.dorsal and l.codigo =
m.codigo)
                  and not exists(select*
                                  from Puerto p
                                  where p.dorsal = c.dorsal))
and exists(select*
            from Ciclista c
            where c.nomeq = e.nomeq);
```

29. Obtener el código y el color de aquellos maillots que sólo han sido llevados por ciclistas de un mismo equipo.

COD COLOR

--- -----
MMS Estrellitas moradas

```
select distinct m.codigo, m.color
from Maillot m, Ciclista c, llevar l
where l.codigo = m.codigo
and l.dorsal = c.dorsal
and not exists (select*
                 from Ciclista d, llevar j
                 where d.nomeq <> c.nomeq
                 and d.dorsal = j.dorsal
                 and j.codigo = m.codigo
                 )
and exists(select*
            from llevar l, Maillot m, Ciclista c
            where l.codigo = m.codigo
            and l.dorsal = c.dorsal);
```

30. Obtener el nombre de aquellos equipos tales que sus ciclistas sólo hayan ganado puertos de 1ª categoría.

NOMEQ

Carrera

Gatorade

```
select e.nomeq
from Equipo e
where not exists (select*
                  from Ciclista c, Puerto p
                  where c.nomeq = e.nomeq
                  and p.dorsal = c.dorsal
                  and p.categoria <> '1')
and exists(select*
           from Ciclista c, Puerto p
           where c.nomeq = e.nomeq
           and p.dorsal = c.dorsal);
```

11.5 Consultas agrupadas

31. Obtener el valor del atributo netapa de aquellas etapas que tienen puertos de montaña indicando cuántos tiene.

NETAPA NUM_PUERTOS

| | |
|----|---|
| 2 | 1 |
| 10 | 4 |
| 11 | 2 |
| 15 | 1 |
| 16 | 1 |
| 18 | 3 |
| 19 | 2 |

7 filas seleccionadas.

```
select e.netapa, count(p.nompuerto) NUM_PUERTOS
from Etapa e, Puerto p
where p.netapa = e.netapa
group by e.netapa
order by e.netapa;
```

32. Obtener el nombre de los equipos que tengan ciclistas indicando cuántos tiene cada uno.

NOMEQ CICLISTAS

| | |
|-----------------|----|
| Amore Vita | 3 |
| Artiach | 7 |
| Banesto | 11 |
| Bresciali-Refin | 4 |
| Carrera | 3 |
| Castorama | 2 |
| Euskadi | 2 |
| Gatorade | 4 |
| Gewiss | 8 |
| Jolly Club | 2 |
| Kelme | 7 |
| Lotus Festina | 3 |
| Mapei-Clas | 7 |
| Mercatone Uno | 8 |
| Motorola | 3 |
| Navigare | 5 |
| ONCE | 5 |
| Seguros Amaya | 3 |
| TVM | 6 |
| Telecom | 4 |
| Wordperfect | 3 |

21 filas seleccionadas.

```
select c.nomeq, count(c.dorsal) CICLISTAS
From Equipo e, Ciclista c
where c.nomeq = e.nomeq
group by c.nomeq
order by c.nomeq;
```


33. Obtener el nombre de todos los equipos indicando cuántos ciclistas tiene cada uno.

| NOMEQ | CICLISTAS |
|------------------------|-----------|
| ----- | ----- |
| Amore Vita | 3 |
| Artiach | 7 |
| Banesto | 11 |
| Bresciali-Refin | 4 |
| Carrera | 3 |
| Castorama | 2 |
| Euskadi | 2 |
| Gatorade | 4 |
| Gewiss | 8 |
| Jolly Club | 2 |
| Kelme | 7 |
| Lotus Festina | 3 |
| Mapei-Clas | 7 |
| Mercatone Uno | 8 |
| Motorola | 3 |
| Navigare | 5 |
| ONCE | 5 |
| PDM | 0 |
| Seguros Amaya | 3 |
| TVM | 6 |
| Telecom | 4 |
| Wordperfect | 3 |
| 22 filas seleccionadas | |

```
select nomeq, count(dorsal)
from equipo natural left join ciclista
group by nomeq
order by nomeq;
```

34. Obtener el director y el nombre de los equipos que tengan más de 3 ciclistas y cuya edad media sea inferior o igual a 30 años.

| DIRECTOR | NOMEQ |
|------------------------|---------------|
| ----- | ----- |
| Ettore Romano | Mercatone Uno |
| José Pérez | Artiach |
| Lonrenzo Sciacchi | Navigare |
| Manuel Sainz | ONCE |
| Moreno Argentin | Gewiss |
| Morgan Reikcard | Telecom |
| 6 filas seleccionadas. | |

```
select e.director, e.nomeq
from equipo e, ciclista c
where c.nomeq = e.nomeq
group by e.nomeq, e.director
having avg(c.edad) <= 30
and count(c.dorsal)>3
order by e.director;
```

35. Obtener el nombre de los ciclistas que pertenezcan a un equipo que tenga más de cinco corredores y que hayan ganado alguna etapa indicando cuántas etapas ha ganado.

| NOMBRE | ETAPAS |
|----------------------|--------|
| ----- | ----- |
| Bo Hamburger | 1 |
| Gert-Jan Theunisse | 1 |
| Gian Matteo Fagnini | 1 |
| Giorgio Furlan | 1 |
| Hernan Buenahora | 1 |
| Juan Martinez Oliver | 1 |
| Mario Cipollini | 1 |
| Miguel Induráin | 2 |

| | |
|------------------|---|
| Pedro Delgado | 3 |
| Tony Rominger | 1 |
| Vladislav Bobrik | 1 |

11 filas seleccionadas.

```
select c.nombre, count(e.netapa) ETAPAS
from ciclista c, etapa e, equipo t
where c.nomeq = t.nomeq
      and c.dorsal = e.dorsal
      and 5 < (select count(dorsal)
               from ciclista
               where nomeq = c.nomeq)
group by c.nombre
order by c.nombre;
```

36. Obtener el nombre de los equipos y la edad media de sus ciclistas de aquellos equipos que tengan la media de edad máxima de todos los equipos.

| NOMEQ | MEDIA |
|------------|-------|
| Amore Vita | 32 |
| Gatorade | 32 |

```
Select nomeq, avg(edad) MEDIA
From ciclista
Group by nomeq
Having avg(edad) >= ALL(select max(avg(edad))
                       From ciclista
                       Group by nomeq);
```

37. Obtener el director de los equipos cuyos ciclistas han llevado, entre todos, más días maillots de cualquier tipo. Nota: cada tupla de la relación *Llevar* indica que un ciclista ha llevado un maillot un día

| DIRECTOR |
|-------------------|
| Miguel Echevarria |

```
Select director
From equipo e, ciclista c, llevar l
Where e.nomeq = c.nomeq
      and c.dorsal = l.dorsal
      and l.codigo is not null
group by director
having count(l.codigo) >= ALL (select count(t.codigo)
                              From equipo p, ciclista i, llevar t
                              Where p.nomeq = i.nomeq
                                    and i.dorsal = t.dorsal
                                    and t.codigo is not null
                              group by director);
```

11.6 Consultas generales

38. Obtener el código y el color del maillot que ha sido llevado por algún ciclista que no ha ganado ninguna etapa.

| COD | COLOR |
|-----|---------------------|
| MGE | Amarillo |
| MMO | Blanco y Rojo |
| MMS | Estrellitas moradas |
| MMV | Rojo |
| MRE | Verde |
| MSE | Rosa |

6 filas seleccionadas.

```
Select distinct m.codigo, color
From maillot m, llevar l, etapa e
Where m.codigo = l.codigo
      and l.netapa <> e.netapa
order by m.codigo;
```

39. Obtener el valor del atributo netapa, la ciudad de salida y la ciudad de llegada de las etapas de más de 190 km. y que tengan por lo menos dos puertos.

| NETAPA | SALIDA | LLEGADA |
|--------|----------|--------------------|
| 10 | Igualada | Andorra |
| 11 | Andorra | Estación de Cerler |
| 18 | Ávila | Ávila |

```
select e.netapa, salida, llegada
from etapa e, puerto p
where km > 190 and e.netapa = p.netapa
group by e.netapa, salida, llegada
having count(p.netapa) >= 2
order by e.netapa;
```

40. Obtener el dorsal y el nombre de los ciclistas que no han llevado todos los maillots que ha llevado el ciclista de dorsal 20

```
DORSAL      NOMBRE
-----
... /*Salen todos menos el 1 y el 20*/
98 filas seleccionadas.
```

```
select c.dorsal, nombre
from ciclista c
where exists (select *
              from llevar
              where dorsal = 20
                 and codigo not in (select codigo
                                    from llevar
                                    where dorsal = c.dorsal))
order by c.dorsal;
```

41. Obtener el dorsal y el nombre de los ciclistas que han llevado al menos un maillot de los que ha llevado el ciclista de dorsal 20.

```
DORSAL      NOMBRE
-----
1 Miguel Induráin
16 Dimitri Konishev
17 Bruno Leali
27 Laurent Jalabert
33 Stefano della Santa
42 Davide Cassani
48 Marco Saligari
7 filas seleccionadas.
```

```
select c.dorsal, nombre
from ciclista c
where dorsal in (select dorsal
                from llevar
                where codigo in (select codigo
                                from llevar
                                where dorsal = 20))
and dorsal <> 20
order by c.dorsal;
```

42. Obtener el dorsal y el nombre de los ciclistas que no han llevado ningún maillot de los que ha llevado el ciclista de dorsal 20.

```
DORSAL      NOMBRE
-----
... /*Salen todos menos el 1,16,17,20,27,33,42 y 48*/
```

```
select c.dorsal, nombre
from ciclista c
where dorsal not in (select dorsal
                    from llevar
                    where codigo in (select codigo
                                    from llevar
                                    where dorsal = 20))

order by c.dorsal;
```

43. 9Obtener el dorsal y el nombre de los ciclistas que han llevado todos los maillots que ha llevado el ciclista de dorsal 20.

| DORSAL | NOMBRE |
|--------|-----------------|
| ----- | |
| 1 | Miguel Induráin |

44. Obtener el dorsal y el nombre de los ciclistas que han llevado exactamente los mismos maillots que ha llevado el ciclista de dorsal 20.

no se ha seleccionado ninguna fila

45. Obtener el dorsal y el nombre del ciclista que ha llevado durante más kilómetros un mismo maillot e indicar también el color de dicho maillot.

| DORSAL | NOMBRE | COLOR |
|--------|-------------------|-------|
| ----- | | |
| 20 | Alfonso Gutiérrez | Verde |

```
select c.dorsal, nombre, color
from ciclista c, llevar l, maillot m, etapa e
where c.dorsal = l.dorsal
      and l.codigo = m.codigo
      and l.netapa = e.netapa
group by c.dorsal, nombre, color
having sum(km) >= all(select max(sum(km))
                    from ciclista t, llevar p, maillot k, etapa n
                    where t.dorsal = p.dorsal
                      and p.codigo = k.codigo
                      and p.netapa = n.netapa
                    group by c.dorsal, nombre, color);
```

46. Obtener el dorsal y el nombre de los ciclistas que han llevado tres tipos de maillot menos de los que ha llevado el ciclista de dorsal 1.

| DORSAL | NOMBRE |
|--------|-------------------|
| ----- | |
| 20 | Alfonso Gutiérrez |
| 30 | Melchor Mauri |
| 26 | Mikel Zarrabeitia |
| 2 | Pedro Delgado |

```

select c.dorsal, nombre
from ciclista c, llevar l
where c.dorsal = l.dorsal
group by c.dorsal, nombre
having count(distinct codigo) + 3 = (select count(distinct codigo)
                                     from llevar
                                     where dorsal = '1')

order by nombre;

```

47. Obtener el valor del atributo netapa y los km de las etapas que tienen puertos de montaña.

| NETAPA | KM |
|--------|----|
|--------|----|

| | |
|----|-----|
| 2 | 180 |
| 10 | 200 |
| 11 | 195 |
| 15 | 207 |
| 16 | 160 |
| 18 | 195 |
| 19 | 190 |

7 filas seleccionadas.

```

select distinct e.netapa, km
from etapa e, puerto p
where e.netapa = p.netapa
order by e.netapa ;

```