

BASES DE DATOS

Desarrollo de Aplicaciones Multiplataforma

Desarrollo de Aplicaciones Web

GESTIÓN DE BASES DE DATOS

Administración de Sistemas Informáticos en Red

PRÁCTICA GUIADADML: CONSULTAS DE DATOS III



Práctica guiada

Luis Dorado Garcés

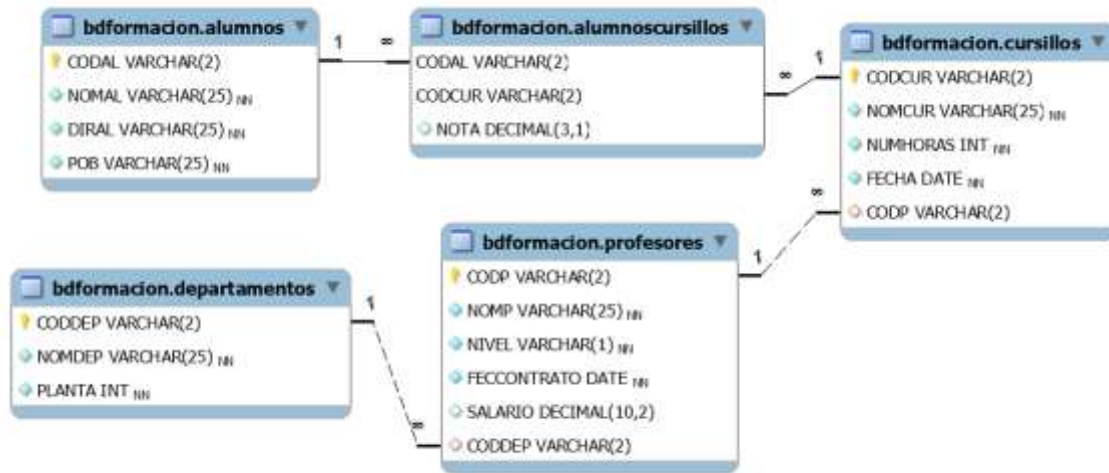
Basado en el trabajo de Alba Tortosa López

ÍNDICE

1	ESTRUCTURA Y DATOS BD FORMACIÓN	1
2	SUBCONSULTAS	1
2.1	Subconsultas básicas (Ej. 1)	2
2.2	Subconsultas ANY/ALL e IN/NOT IN (Ej. 2 - 5).....	2
2.3	Subconsultas correlacionadas (Ej. 6 y 7).....	6
2.4	Predicado EXISTS y consultas correlacionadas (Ej. 8 y 9).....	10

1 Estructura y datos BD Formación

Diagrama relacional



- ✓ La clave primaria de *alumnos cursillos* es compuesta por sus dos campos de clave ajena.
- ✓ Los campos *alumnos.diral*, *cursillos.nomcur*, *profesores.nomp* son claves alternativas.
- ✓ Las inserciones de los datos se entregan en un archivo SQL aparte.

2 Subconsultas

Una subconsulta es una sentencia *SELECT* que se incluye dentro de otra sentencia *SELECT* (que llamamos principal), en la condición de **WHERE** o de **HAVING**.

La **reformulación**:

- Cuando trabajamos con subconsultas muchas veces debemos reformular el texto del enunciado para resolver de diferentes maneras una consulta.
- Por ejemplo, decir que “queremos el alumno cuya nota es igual a la nota máxima” es lo mismo que decir que “queremos el alumno cuya nota es igual o mayor a todas las notas”.
- Aunque ambas formulaciones nos muestran el mismo resultado, se resuelven de manera diferente.
- “Nombre del alumno cuya nota **es iguala** la nota máxima”.

```

SELECT DISTINCT ALUMNOS.NOMAL
FROM ALUMNOSCURSILLOS INNERJOIN ALUMNOS
ON ALUMNOSCURSILLOS.CODAL = ALUMNOS.CODAL
WHERE NOTA = (SELECT MAX(NOTA)
              FROM ALUMNOSCURSILLOS);
  
```

- “Nombre del alumno cuya nota **es mayor o iguala todas** las notas de la BD”.

```

SELECT DISTINCT ALUMNOS.NOMAL
FROM ALUMNOSCURSILLOS INNERJOIN ALUMNOS
ON ALUMNOSCURSILLOS.CODAL = ALUMNOS.CODAL
WHERE NOTA >= ALL (SELECT NOTA
                   FROM ALUMNOSCURSILLOS);
  
```

2.1 Subconsultas básicas (Ej. 1)

1. Visualizar el código de cada alumno, código de cursillo y nota siempre que dicha nota sea mayor que la media de notas de todos los alumnos en todos los cursillos.

En este caso hay que comparar cada nota de cada alumno con un valor (nota media) que desconocemos previamente; **utilizaremos una subconsulta para calcular ese valor** y luego se compara cada nota de cada alumno (de la SELECT principal) con dicho valor.

```
USEBDFORMACION;
```

```
SELECT CODAL, CODCUR, NOTA
FROM ALUMNOSCURSILLOS
WHERE NOTA > (SELECT AVG(NOTA) FROM ALUMNOSCURSILLOS);
```

CODAL	CODCUR	NOTA
A1	C1	9
A3	C1	9
A3	C2	9

Cuando se ejecuta la subconsulta (`SELECT AVG(NOTA) FROM ALUMNOSCURSILLOS`) el valor obtenido es **8,75** de forma que la consulta principal equivale a:

```
USEBDFORMACION;
```

```
SELECT CODAL, CODCUR, NOTA
FROM ALUMNOSCURSILLOS
WHERE NOTA > (8.75);
```

2.2 Subconsultas ANY/ALL e IN/NOT IN (Ej. 2 - 5)

A veces necesitamos que la **subconsulta devuelva un conjunto de valores**; en ese caso utilizaremos los operadores *ANY* y *ALL* combinados *con* =, >, <, >=, <=, y también los operadores *IN* y *NOT IN*.

2. Visualizar los nombres de los alumnos que no están apuntados a ningún cursillo:

	NOMAL
1	Antonio García Pérez
2	Carlos Pérez Ruiz
3	José Navarro Lard
4	Elisa Ébeda Sansón
5	Eva San Martín
6	Alfredo Hernández Luis

Alternativa con NOT IN

Reformulación: "Quiero los alumnos cuyo código **no está en** la tabla *ALUMNOSCURSILLOS*".

```
USEBDFORMACION;
```

```
SELECT ALUMNOS.NOMAL
FROM ALUMNOS
WHERE ALUMNOS.CODAL NOT IN (SELECT ALUMNOSCURSILLOS.CODAL
                             FROM ALUMNOSCURSILLOS);
```

La subconsulta devuelve los códigos de alumnos de la tabla ALUMNOSCURSILLOS, que son (A1, A2, A3, A3). Por cada código de alumno de la tabla ALUMNOS (SELECT principal) se comprueba si consta en la lista de valores devuelta por la subconsulta (A1, A2, A3, A3) y si es así se visualiza en pantalla el nombre correspondiente a dicho código (A4-A9).

Alternativa con ALL

Reformulación: “Quiero los alumnos cuyo código **es diferente a todos** los códigos de la tabla ALUMNOSCURSILLOS”

```
USE BDFORMACION;
```

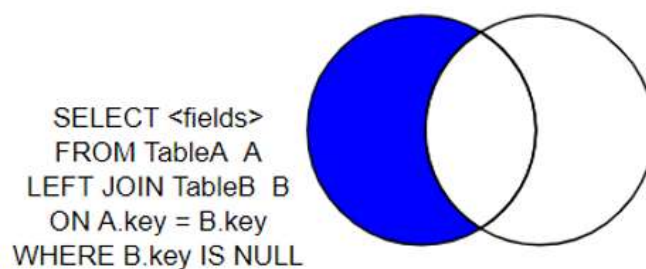
```
SELECT ALUMNOS.NOMAL
FROM ALUMNOS
WHERE ALUMNOS.CODAL != ALL (SELECT ALUMNOSCURSILLOS.CODAL
                             FROM ALUMNOSCURSILLOS);
```

Alternativa con JOINS

Ojo: Fijaos en que podríamos haber resuelto esta consulta usando LEFT JOIN en el que se seleccionan todas las filas de ALUMNOS que no participan en una relación con ALUMNOSCURSILLOS.

```
USE BDFORMACION;
```

```
SELECT ALUMNOS.NOMAL
FROM ALUMNOS
LEFT JOIN ALUMNOSCURSILLOS ON
  ALUMNOS.CODAL = ALUMNOSCURSILLOS.CODAL
WHERE ALUMNOSCURSILLOS.CODAL IS NULL;
```



- Visualizar el nombre y nivel de aquellos profesores que imparten cursillos que comiencen con H:

NOMP	NIVEL
Patricia Pére	A
Peio Pagoaga	A

Alternativa con JOINS

```
SELECT PROFESORES.NOMP, PROFESORES.NIVEL
FROM PROFESORES
INNERJOIN CURSILLOS
ON CURSILLOS.CODP =PROFESORES.CODP
WHERE CURSILLOS.NOMCUR LIKE 'H%';
```

Alternativa con IN

Reformulación: “Quiero el nombre y nivel de los profesores cuyo código **esté dentro de la** columna CODP (clave ajena) de los cursillos que empiezan por H”

```
SELECT NOMP, NIVEL
FROM PROFESORES
WHERE CODP IN (SELECT CODP
FROM CURSILLOS
WHERE NOMCUR LIKE 'H%');
```

Alternativa con ANY

Reformulación: “Quiero el nombre y nivel de los profesores cuyo código **sea igual a**

algún valor de la columna CODP (clave ajena) de los cursillos que empiezan por H”.

```
SELECT NOMP, NIVEL
FROM PROFESORES
WHERE CODP = ANY (SELECT CODP
FROM CURSILLOS
WHERE NOMCUR LIKE 'H%');
```

4. Visualizar el código de los alumnos, de los cursillos y la nota siempre que ésta sea mayor que **todas** las notas después de restarles 0.5.
 - a. Notas después de restarles 0.5 (Subconsulta):

```
USE BDFORMACION;
```

```
SELECT NOTA-0.5 AS 'NOTA-0.5' FROM ALUMNOSCURSILLOS;
```

	NOTA-0.5
1	8,5
2	7,5
3	8,5
4	8,5

- b. Alumnos con nota mayor que todas las anteriores (Consulta completa):

```
SELECT CODAL, CODCUR, NOTA
FROM ALUMNOSCURSILLOS
WHERE NOTA > ALL (SELECT NOTA-0.5 AS 'NOTA-0.5'
FROM ALUMNOSCURSILLOS);
```

CODAL	CODCUR	NOTA
A1	C1	9
A3	C1	9
A3	C2	9

En este caso la subselección (`SELECT NOTA-0.5 FROM ALUMNOSCURSILLOS`) devuelve los valores (8.5, 7.5, 8.5, 8.5). La SELECT principal recorre cada fila de la tabla ALUMNOSCURILOS **evaluando** si la nota correspondiente es mayor que **todas** las de la lista.

- Para la primera fila, A1 C1 9, la nota 9 es mayor que todos los valores de la lista por lo tanto se visualiza el código, el curso y la nota.
- Para la segunda fila, A2 C2 8, la nota 8 **no** es mayor que todos los valores de la lista por lo tanto no se visualiza el código, el curso y la nota.
- Para la tercera A3 C1 9, el valor de la nota 9 **sí** es mayor que todos los valores de la lista y se visualiza el código del alumno, el del cursillo y el de la nota.
- Para la cuarta fila A3 C2 9, el valor de la nota 9 **sí** es mayor que todos los valores de la lista y se visualiza el código del alumno, el del cursillo y el de la nota.

Ojo: Si no incluimos el ALL nos dará error porque estaremos comparando, para cada fila de la consulta principal, con varios valores. Y eso no está permitido.

Error: La subconsulta ha devuelto más de un valor, lo que no es correcto cuando va a continuación de =, !=, <, <=, >, >= o cuando se utiliza como expresión.

5. Visualizar el nombre aquellos alumnos **matriculados en un solo cursillo** y cuya nota es **menor que alguna** de las demás notas.

- a. Notas (Subconsulta):

`USEBDFORMACION;`

`SELECT NOTA FROM ALUMNOSCURSILLOS;`

Resultados	
	NOTA
1	9
2	8
3	9
4	9

- b. Alumnos matriculados a un solo cursillo (Consulta principal):

	NOMAL
1	Ane Aranburu
2	Antonio Antúnez

```
USEBDFORMACION;
```

```
SELECT ALUMNOS.NOMAL
FROM ALUMNOS
INNERJOIN ALUMNOSCURSILLOS
ONALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL
GROUP BY ALUMNOS.NOMAL
HAVING COUNT(*)= 1;
```

c. Consulta completa:

```
USEBDFORMACION;
```

```
SELECT ALUMNOS.NOMAL
FROM ALUMNOS INNERJOINALUMNOSCURSILLOS
ONALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL
WHEREALUMNOSCURSILLOS.NOTA<ANY(SELECT NOTA
FROMALUMNOSCURSILLOS)
```

```
GROUPBYALUMNOS.NOMAL
HAVINGCOUNT(*)= 1;
```

Si. porque en el where se descarta el segundo registro de ese alumno(con la nota superior) pero en el having va a considerar a ese alumno ya que se descarto un registro de el previamente y a la condicion count(*) sería true. Aunque no se pide ello.

	NOMAL
1	Ane Aranburu

¿Te parece correcta la solución propuesta? ¿Qué pasaría si un alumno estuviese matriculado en dos cursillos, y en uno de los dos cursillos tuviese la nota mayor de las contenidas? Obviamente ese alumno no debería obtenerse en la salida ya que está matriculado en dos cursillos, pero ¿lo obtiene la solución propuesta? ¿Por qué?

Examinar esta otra solución:

```
SELECT NOMAL
FROM ALUMNOS
WHERE CODAL IN
(SELECT CODAL
FROM ALUMNOSCURSILLOS
WHERE NOTA < ANY
(SELECT NOTA
FROM ALUMNOSCURSILLOS))
AND CODAL IN (SELECT CODAL
FROMALUMNOSCURSILLOS
GROUP BYCODAL
HAVING COUNT (CODCUR) = 1);
```

¿En qué se diferencia de la anterior? ¿Qué hace primero, filtrar las filas que tienen notas superiores o filtrar los alumnos que sólo están en un curso? ¿Y en la anterior solución? ¿Influye esto en el resultado? ¿Con cuál nos quedamos entonces...?

con la segunda

2.3 Subconsultas correlacionadas (Ej. 6 y 7)

Se le denomina así a la subconsulta, cuando es necesario ejecutar la subconsulta pero

enlazando con algún valor de cada fila de la consulta principal.

cuando para ejecutar una subconsulta se necesita algún valor de cada fila de la consulta principal

- Visualizar el nombre, sueldo y código de departamento de los profesores con un sueldo mayor o igual que la media del sueldo de su departamento.

	NOMP	SALARIO	CODDEP
1	Patricia Pérez	1500,00	D1
2	Paula Puerta	1300,00	D2

Reformulación: "Quiero el nombre de los profesores cuyo sueldo sea mayor a la media de sueldos de los profesores que tienen el mismo CODDEP que él"

```
USE BDFORMACION;
```

```
SELECT P1.NOMP, SALARIO, CODDEP
FROM PROFESORES P1
WHERE P1.SALARIO >= (SELECT AVG(P2.SALARIO)
FROM PROFESORES P2
WHERE P2.CODDEP=P1.CODDEP)
```

- Visualizar el código, nombre y salario máximo en cada departamento junto con el nombre del profesor que lo cobra. El resultado que queremos obtener es el siguiente:

	CODDEP	NOMDEP	SALARIO MÁXIMO POR DEPARTAMENTO	NOMP
1	D1	INFORMÁTICA	1500,00	Patricia Pérez
2	D2	ADMINISTRATIVO	1300,00	Paula Puerta

Si realizamos la consulta con GROUPBY:

```
USE BDFORMACION;
```

```
SELECT D.CODDEP, D.NOMDEP, MAX(SALARIO)
AS 'SALARIO MÁXIMO POR DEPARTAMENTO'
FROM PROFESORES P INNER JOIN DEPARTAMENTOS D
ON P.CODDEP=D.CODDEP
GROUP BY D.CODDEP, D.NOMDEP;
```

	CODDEP	NOMDEP	SALARIO MÁXIMO POR DEPARTAMENTO
1	D1	INFORMÁTICA	1500,00
2	D2	ADMINISTRATIVO	1300,00

Obtenemos el salario máximo por cada departamento pero no hemos obtenido el nombre del profesor que lo cobra; si introducimos el nombre en la SELECT nos dará el siguiente error:

```
USEBDFORMACION;
```

```
SELECT D.CODDEP, D.NOMDEP, MAX(SALARIO)
      AS 'SALARIO MÁXIMO POR DEPARTAMENTO', P.NOMP
FROM PROFESORES P INNER JOIN DEPARTAMENTOS D
      ON P.CODDEP=D.CODDEP
GROUP BY D.CODDEP, D.NOMDEP
ORDER BY D.CODDEP;
```

Error: La columna 'PROFESORES.NOMP' de la lista de selección no es válida, porque no está contenida en una función de agregado ni en la cláusula GROUP BY.

Esto es porque una vez que hemos obtenido las filas de agrupamiento sólo podemos visualizar en la *SELECT* las columnas indicadas en la cláusula *GROUP BY* funciones de agregado. Y el nombre del profesor no es ninguna de las dos y si lo añadiéramos desharíamos el agrupamiento:

func de agregacion: resume filas de un grupo en un valor (count, min, avg, ...)

```
USEBDFORMACION;
```

```
SELECT D.CODDEP, D.NOMDEP, MAX(SALARIO)
      AS 'SALARIO MÁXIMO POR DEPARTAMENTO', P.NOMP
FROM PROFESORES P INNER JOIN DEPARTAMENTOS D
      ON P.CODDEP=D.CODDEP
GROUP BY D.CODDEP, D.NOMDEP, P.NOMP
ORDER BY D.CODDEP;
```

	CODDEP	NOMDEP	SALARIO MÁXIMO POR DEPARTAMENTO	NOMP
1	D1	INFORMÁTICA	1500,00	Patricia Pérez
2	D1	INFORMÁTICA	1000,00	Peio Pagoaga
3	D2	ADMINISTRATIVO	1300,00	Paula Puerta

Realizamos la consulta con una subconsulta correlacionada:

Reformulación: “Quiero el nombre de profesor junto con su sueldo, código y nombre de departamento de los profesores cuyo sueldo **coincida con el sueldo máximo de los profesores que tienen el mismo CODDEP que él**”.

```
USEBDFORMACION;
```

```
SELECT D.CODDEP, D.NOMDEP,
      P1.SALARIO AS 'SALARIO MÁXIMO POR DEPARTAMENTO',
      P1.NOMP
FROM PROFESORES P1 INNER JOIN DEPARTAMENTOS D
      ON P1.CODDEP=D.CODDEP
WHERE SALARIO = (SELECT MAX(SALARIO)
                  FROM PROFESORES P2
                  WHERE P2.CODDEP = P1.CODDEP)
ORDER BY D.CODDEP;
```

	CODDEP	NOMDEP	SALARIO MÁXIMO POR DEPARTAMENTO	NOMP
1	D1	INFORMÁTICA	1500.00	Patricia Pérez
2	D2	ADMINISTRATIVO	1300.00	Paula Puerta

Por cada fila de la SELECT principal se ejecuta la subconsulta eligiendo las filas de la subconsulta que coincidan en valor con algún campo de la fila de la SELECT principal que se está examinando. Es decir, por cada fila-profesor se comprobará si su sueldo coincide con el sueldo máximo de los profesores con un código de departamento igual al suyo (del mismo departamento).

PROFESORES (P) (SELECT principal)

Primera fila

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

PROFESORES (PR) (Subselección)

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

Filas devueltas por la subconsulta; el salario más alto entre 1000 y 1500 es 1500

¿El salario 1500 es igual a 1500? Sí; entonces se visualiza el SALARIO y NOMP del profesor P1 de la tabla P junto con la información de su departamento.

PROFESORES (P) (SELECT principal)

Segunda fila

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

PROFESORES (PR) (Subselección)

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

Filas devueltas por la subconsulta (solo una); el salario más alto es 1300

¿El salario 1300 es igual a 1300? Sí; entonces se visualiza el SALARIO y NOMP del profesor

P2 de la tabla P junto con la información de su departamento.

Tercera fila

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

PROFESORES (PR) (Subselección)

CODP	NOMP	SALARIO	CODDEP
P1	Patricia Pérez	1500,0000	D1
P2	Paula Puerta	1300,0000	D2
P3	Peio Pagoaga	1000,0000	D1

Filas devueltas por la subconsulta; el salario más alto entre 1000 y 1500 es 1500

¿El salario 1000 es igual a 1500? No; entonces NO se visualiza el SALARIO y NOMBRE del profesor P3 de la tabla P (ya que no es el más alto de su departamento).

2.4 Predicado EXISTS y consultas correlacionadas (Ej. 8 y 9)

- El predicado **EXISTS** devuelve *Verdadero* si la subconsulta devuelve alguna fila de resultados y *Falso* en caso contrario.
 - Este predicado es interesante cuando no nos interesan los datos concretos devueltos por la subconsulta sino **sólo si la subconsulta devuelve filas de datos o no**.
 - Este predicado es usado frecuentemente en subconsultas correlacionadas para verificar si un valor recuperado por la consulta principal existe (o no) en el conjunto de resultados obtenidos por la subconsulta.
 - Si la subconsulta obtiene al menos una fila, el operador obtiene el valor Verdadero si no obtiene el valor Falso.
 - El predicado NOT EXISTS verifica si un valor recuperado por la consulta principal no existe en el conjunto de resultados obtenidos por la subconsulta.
8. Visualizar los nombres de los alumnos que no están apuntados a ningún cursillo (la hemos realizado antes con NOT IN), la hacemos ahora con EXISTS.
- a. Con NOT IN:
- Reformulación: “Quiero los alumnos cuyo código **no está en la tabla ALUMNOSCURSILLOS**”.

USEBDFORMACION;

```
SELECT ALUMNOS.NOMAL FROM ALUMNOS
WHERE ALUMNOS.CODAL NOT IN (SELECT ALUMNOSCURSILLOS.CODAL
                             FROM ALUMNOSCURSILLOS);
```

	NOMAL
1	Antonio García Pérez
2	Carlos Pérez Ruiz
3	José Navarro Lard
4	Elisa Ébeda Sansón
5	Eva San Martín
6	Alfredo Hernández Luis

b. Con EXISTS:

Reformulación: “Quiero los alumnos para los cuales **no exista** ninguna matrícula en ALUMNOSCURSILLOS con su código de alumno”.

USEBDFORMACION;

```
SELECT A.NOMAL FROM ALUMNOS A
WHERE NOT EXISTS (SELECT *
                  FROM ALUMNOSCURSILLOS AC
                  WHERE AC.CODAL = A.CODAL);
```

Si la subconsulta no devuelve datos (el caso de los seis alumnos cuyo código no aparece en alumnos cursillos) el predicado *NOT EXISTS* se hace cierto y entonces se visualiza su nombre.

9. Visualizar toda la información de aquellos departamentos que tengan algún profesor de nivel ‘A’. (La consulta deberá visualizar el departamento D1 que es el único que tiene algún profesor de nivel ‘A’).

USEBDFORMACION;

```
SELECT D.* FROM DEPARTAMENTOS D
WHERE EXISTS (SELECT *
              FROM PROFESORES P
              WHERE P.NIVEL = 'A' AND P.CODDEP = D.CODDEP);
```

CODDEP	NOMDEP	PLANTA
D1	INFORMÁTICA	1

Esta consulta también se podría haber hecho con *INNER JOIN*:

USEBDFORMACION;

```
SELECT DISTINCT D.*
FROM DEPARTAMENTOS D INNER JOIN PROFESORES P ON P.CODDEP = D.CODDEP
WHERE P.NIVEL = 'A';
```