

BASES DE DATOS

Desarrollo de Aplicaciones Multiplataforma

Desarrollo de Aplicaciones Web

GESTIÓN DE BASES DE DATOS

Administración de Sistemas Informáticos en Red

PRÁCTICA GUIADA DML: CONSULTAS DE DATOS II

Práctica guiada

Luis Dorado Garcés

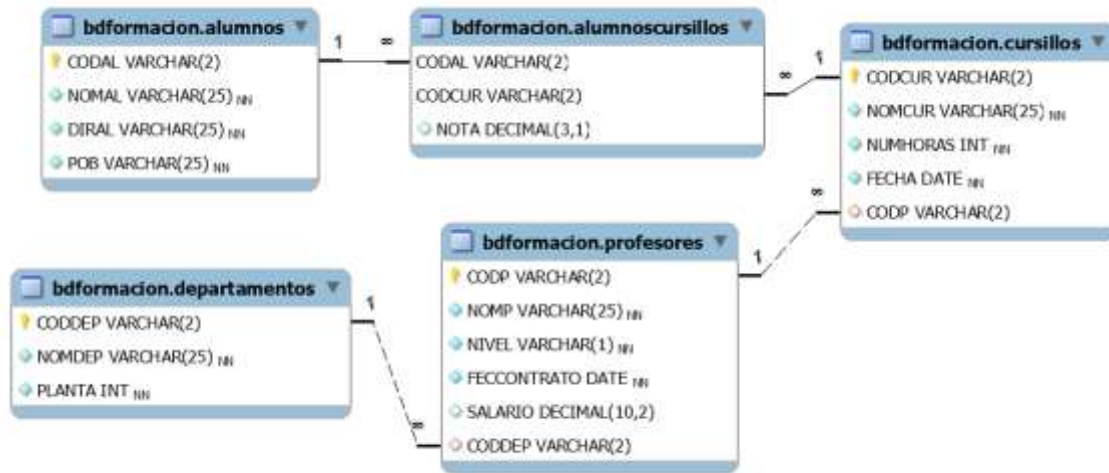
Basado en el trabajo de Alba Tortosa López

ÍNDICE

1	ESTRUCTURA Y DATOS BD FORMACIÓN	1
2	CONSULTAS SOBRE MÁS DE UNA TABLA	1
2.1	Introducción.....	1
2.1.1	Ejemplos con WHERE y con INNER JOIN (Ejemplos 1, 2 y 3)	2
2.2	Ampliación: Orden de ejecución de cláusulas l.....	7
2.3	LEFT JOIN y RIGHT JOIN (Ejemplo 4).....	4
2.4	Consultas de agrupación multitable	4
2.4.1	Cláusula GROUP BY en consultas multitable (Ejemplos 5 - 7)	4
2.4.2	Cláusula HAVING en consultas multitable (Ejemplos 8-10).....	5

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 Consultas sobre más de una tabla

Cuando queremos obtener información de más de una tabla hay que incluir dichas tablas en la cláusula *FROM* teniendo en cuenta que las **relacionaremos por un campo común** (normalmente **clave principal-clave ajena**), bien utilizando la cláusula *WHERE* o bien la cláusula *INNER JOIN*.

2.1 Introducción

Visualizar el nombre de los cursillos con el nombre del profesor que lo imparte.

```
USE BDFORMACION;
```

```
SELECT NOMCUR, NOMP FROM CURSILLOS, PROFESORES;
```

NOMCUR	NOMP		
SO	Patricia Pérez	EIEM	Paula Puerta
HW	Patricia Pérez	FOL	Paula Puerta
HTML	Patricia Pérez	SO	Peio Pagoaga
EIEM	Patricia Pérez	HW	Peio Pagoaga
FOL	Patricia Pérez	HTML	Peio Pagoaga
SO	Paula Puerta	EIEM	Peio Pagoaga
HW	Paula Puerta	FOL	Peio Pagoaga
HTML	Paula Puerta		

Al poner las dos tablas en el *FROM*, el sistema obtiene una tabla resultado de realizar el **producto cartesiano** de dichas tablas, es decir, todas las combinaciones de los grupos de campos de todas las filas de una tabla con los de la otra. Pero observamos que, de todas esas filas resultado, algunas no son "ciertas". Las únicas relaciones verdaderas entre cursillos y profesores son aquellas en las que el código del profesor en la tabla cursillos coincida con el código del profesor en la tabla profesores; esto lo indicaremos en la cláusula *WHERE* o bien en la cláusula *INNER JOIN* de *FROM* (combinación interna).

```
USE BDFORMACION;
```

```
SELECT CODCUR, NOMCUR, CURSILLOS.CODP, NOMP
FROM CURSILLOS, PROFESORES
WHERE CURSILLOS.CODP=PROFESORES.CODP
```

O también:

```
USE BDFORMACION;
```

```
SELECT CODCUR, NOMCUR, CURSILLOS.CODP, NOMP
FROM CURSILLOS INNER JOIN PROFESORES ON
CURSILLOS.CODP=PROFESORES.CODP;
```

CODCUR	NOMCUR	CODP	NOMP
C1	SO	P1	Patricia Pérez
C2	HW	P3	Peio Pagoaga
C3	HTML	P1	Patricia Pérez
C4	EIEM	P2	Paula Puerta
C5	FOL	P2	Paula Puerta

- ✓ **Nosotros usaremos esta última sintaxis para hacer las combinaciones de tablas.**
- ✓ En el caso de que alguno de los campos de la combinación esté indexado es más eficiente *INNER JOIN*.
- ✓ Cuando dos campos de diferentes tablas se llaman igual hay que indicar a cuál nos referimos poniendo el nombre de la tabla antes del nombre campo y separados por un punto.
- ✓ Podemos utilizar alias para los nombres de tabla en el *FROM*:

```
USE BDFORMACION;
```

```
SELECT CODCUR, NOMCUR, C.CODP, NOMP
FROM CURSILLOS C INNER JOIN PROFESORES P ON C.CODP=P.CODP;
```

2.1.1 Ejemplos con *WHERE* y con *INNER JOIN* (Ejemplos 1, 2 y 3)

1. Visualizar el nombre de cada alumno con el nombre de cada cursillo al que está apuntado y la nota correspondiente.

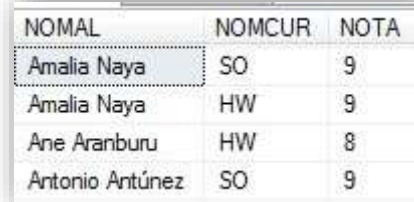
```
USE BDFORMACION;
```

```
SELECT NOMAL, NOMCUR, NOTA
FROM ALUMNOS, CURSILLOS, ALUMNOSCURSILLOS
WHERE (ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL) AND
      (CURSILLOS.CODCUR=ALUMNOSCURSILLOS.CODCUR);
```

Con INNER JOIN:

```
USE BDFORMACION;
```

```
SELECT NOMAL, NOMCUR, NOTA
FROM ALUMNOS INNER JOIN ALUMNOSCURSILLOS
      ON ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL
      INNER JOIN CURSILLOS
      ON CURSILLOS.CODCUR=ALUMNOSCURSILLOS.CODCUR;
```



NOMAL	NOMCUR	NOTA
Amalia Naya	SO	9
Amalia Naya	HW	9
Ane Aranburu	HW	8
Antonio Antúnez	SO	9

- Visualizar el nombre de cada alumno con el nombre de cada cursillo al que está apuntado, el nombre del profesor que lo imparte y la nota correspondiente.

```
USE BDFORMACION;
```

```
SELECT NOMAL, NOMCUR, NOMP, NOTA
FROM ALUMNOS, CURSILLOS, PROFESORES, ALUMNOSCURSILLOS
WHERE (ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL) AND
      (CURSILLOS.CODCUR=ALUMNOSCURSILLOS.CODCUR) AND
      (CURSILLOS.CODP=PROFESORES.CODP);
```

Con INNER JOIN:

```
USE BDFORMACION;
```

```
SELECT NOMAL, NOMCUR, NOMP, NOTA
FROM ALUMNOS INNER JOIN ALUMNOSCURSILLOS
      ON ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL
      INNER JOIN CURSILLOS ON CURSILLOS.CODCUR=ALUMNOSCURSILLOS.CODCUR
      INNER JOIN PROFESORES ON CURSILLOS.CODP=PROFESORES.CODP;
```

NOMAL	NOMCUR	NOMP	NOTA
Antonio Antúnez	SO	Patricia Pére	9
Ane Aranburu	HW	Peio Pagoaga	8
Amalia Naya	SO	Patricia Pére	9
Amalia Naya	HW	Peio Pagoaga	9

3. Ídem que el ejemplo 2 pero de aquellos **cursillos que comiencen en marzo** (con INNER JOIN).

NOMAL	NOMCUR	NOMP	NOTA
Ane Aranburu	HW	Peio Pagoaga	8
Amalia Naya	HW	Peio Pagoaga	9

```
USE BDFORMACION;
```

```
SELECT NOMAL, NOMCUR, NOMP, NOTA
FROM
ALUMNOS INNER JOIN ALUMNOSCURSILLOS
ON ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL
INNER JOIN CURSILLOS ON CURSILLOS.CODCUR=ALUMNOSCURSILLOS.CODCUR
INNER JOIN PROFESORES ON CURSILLOS.CODP=PROFESORES.CODP
WHERE MONTH(FECHA)=3;
```

2.2 Consultas de agrupación multitable

La cláusula *GROUP BY* en la sentencia *SELECT* permite agrupar filas producto del *FROM* o (del *FROM* con *WHERE*) por algún campo/s de la tabla, y hacer algún cálculo sobre otro/s campo.

2.2.1 Cláusula GROUP BY en consultas multitable (Ejemplos 4 - 6)

4. Visualizar el código de aquellos alumnos de la calle “La Ermita” con su nota media.

```
USE BDFORMACION;
```

```
SELECT ALUMNOSCURSILLOS.CODAL, AVG(ALUMNOSCURSILLOS.NOTA) AS 'NOTA
MEDIA'
FROM ALUMNOS INNER JOIN ALUMNOSCURSILLOS
ON ALUMNOS.CODAL = ALUMNOSCURSILLOS.CODAL
WHERE ALUMNOS.DIRAL LIKE '%La Ermita%'
GROUP BY ALUMNOSCURSILLOS.CODAL;
```

CODAL	NOTA MEDIA
A1	9.00000

5. Visualizar cada nombre de cursillo de aquellos que comiencen por H con el número de alumnos apuntados.

```
USE BDFORMACION;
```

```
SELECT NOMCUR, COUNT(*) AS 'Nº ALUMNOS APUNTADOS'
FROM ALUMNOSCURSILLOS INNER JOIN CURSILLOS
ON ALUMNOSCURSILLOS.CODCUR=CURSILLOS.CODCUR
```

```
WHERE NOMCUR LIKE 'H%'
GROUP BY NOMCUR;
```

NOMCUR	Nº ALUMNOS APUNTADOS
HW	2

6. Ejercicio: Visualizar cada nombre de cursillo con el número de alumnos apuntados.

```
USE BDFORMACION;
SELECT NOMCUR, COUNT(*) AS 'Nº ALUMNOS APUNTADOS'
FROM ALUMNOSCURSILLOS INNER JOIN CURSILLOS
ON ALUMNOSCURSILLOS.CODCUR=CURSILLOS.CODCUR
GROUP BY NOMCUR;
```

NOMCUR	Nº ALUMNOS APUNTADOS
HW	2
SO	2

2.2.2 Cláusula HAVING en consultas multitabla (Ejemplos 7-9)

Así como la cláusula *WHERE* permite retener las filas de *FROM* que cumplen una determinada condición, la cláusula **HAVING** permite, si la condición asociada es cierta, retener filas del *GROUP BY* previamente realizado.

7. Visualizar el código de aquellos alumnos cuya dirección contiene 'C/' con su nota media siempre que esa nota media sea mayor o igual a 8.

```
USE BDFORMACION;

SELECT ALUMNOSCURSILLOS.CODAL,
AVG(ALUMNOSCURSILLOS.NOTA) AS 'NOTA MEDIA'
FROM ALUMNOS INNER JOIN ALUMNOSCURSILLOS
ON ALUMNOS.CODAL = ALUMNOSCURSILLOS.CODAL
WHERE ALUMNOS.DIRAL LIKE '%C/%'
GROUP BY ALUMNOSCURSILLOS.CODAL
HAVING AVG(NOTA) >= 8;
```

CODAL	NOTA MEDIA
A1	9.00000
A3	9.00000

8. Visualizar el código de aquellos alumnos cuya dirección contiene 'C/' con su nota media siempre que esa nota media sea mayor o igual a LA NOTA MEDIA DE TODOS LOS ALUMNOS.

```
USE BDFORMACION;
SELECT A.CODAL, AVG(NOTA) AS 'NOTA MEDIA' FROM
(ALUMNOSCURSILLOS AC JOIN ALUMNOS A ON A.CODAL=AC.CODAL)
WHERE A.DIRAL LIKE '%C/%'
GROUP BY A.CODAL
HAVING AVG(NOTA) >= (SELECT AVG(NOTA) FROM ALUMNOSCURSILLOS)
```

9. Visualizar cada nombre de cursillo de aquellos que comiencen por H y que tengan más de un alumno apuntado.

```
SELECT NOMCUR, COUNT(*) AS 'Nº ALUMNOS APUNTADOS'
FROM ALUMNOSCURSILLOS INNER JOIN CURSILLOS
ON ALUMNOSCURSILLOS.CODCUR=CURSILLOS.CODCUR
WHERE NOMCUR LIKE 'H%'
GROUP BY NOMCUR
HAVING COUNT(*) > 1;
```

2.3 LEFT JOIN y RIGHT JOIN (Ejemplo 10)

Cuando trabajamos con más de una tabla, a veces es interesante visualizar todas las filas de una de las tablas y de aquellas filas que tengan su correspondiente campo en común con la otra tabla, que se visualicen las filas correspondientes de esa segunda tabla. Esto lo hacemos con las cláusulas LEFT JOIN o RIGHT JOIN.

Queremos visualizar la información de todos los alumnos de la base de datos (CODAL, NOMAL, DIRAL, POB), pero de aquellos alumnos que estén apuntados a cursillos, que también se visualice el código de los cursillos y la nota.

Previamente añadimos la siguiente fila en la tabla **ALUMNOS**:

CODAL	NOMAL	DIRAL	POB
A4	Aitor Aguirre	C/Ronda 7	BURLADA

```
USE BDFORMACION;
```

```
SELECT ALUMNOS.*, ALUMNOSCURSILLOS.CODCUR, NOTA
FROM ALUMNOS LEFT JOIN ALUMNOSCURSILLOS ON
ALUMNOS.CODAL=ALUMNOSCURSILLOS.CODAL;
```

CODAL	NOMAL	DIRAL	POB	CODCUR	NOTA
A1	Antonio Antúnez	C/La Ermita 3	Pamplona	C1	9.0
A2	Ane Aranburu	Avda. Central 5	Barañáin	C2	8.0
A3	Amalia Naya	C/Magdalena s/n	Pamplona	C1	9.0
A3	Amalia Naya	C/Magdalena s/n	Pamplona	C2	9.0
A4	Antonio García Pérez	Astilleros 3	Valencia	NULL	NULL
A5	Carlos Pérez Ruiz	Magallanes 21	Utiel	NULL	NULL
A6	José Navarro Lard	Río Segura 14	Valencia	NULL	NULL
A7	Elisa Ébeda Sansón	Valencia 4	Sagunto	NULL	NULL
A8	Eva San Martín	Villafranca 34	Alzira	NULL	NULL
A9	Alfredo Hernández Luis	Salinas 8	Valencia	NULL	NULL

Observamos que para el alumno A4 se visualizan sus datos pero como no tiene registros asociados en la tabla ALUMNOSCURSILLOS, quedan a NULL los correspondientes al código del cursillo y la nota.

Al poner la tabla ALUMNOS a la izquierda de la cláusula FROM ponemos LEFT JOIN, pero también podríamos haber resuelto la sentencia con la cláusula RIGHT JOIN a condición de poner el nombre de la tabla ALUMNOS a la derecha:


```
USE BDFORMACION;
```

```
SELECT ALUMNOS.*, ALUMNOSCURSILLOS.CODCUR, NOTA
FROM ALUMNOSCURSILLOS RIGHT JOIN ALUMNOS ON
ALUMNOSCURSILLOS.CODAL=ALUMNOS.CODAL;
```

El resultado es el mismo que el anterior.

2.4 Ampliación: Orden de ejecución de cláusulas I

Orden de ejecución de las cláusulas de la instrucción SELECT vistas hasta ahora. Este orden es fundamental para interpretar y elaborar las consultas SQL.

