

Tarea 3

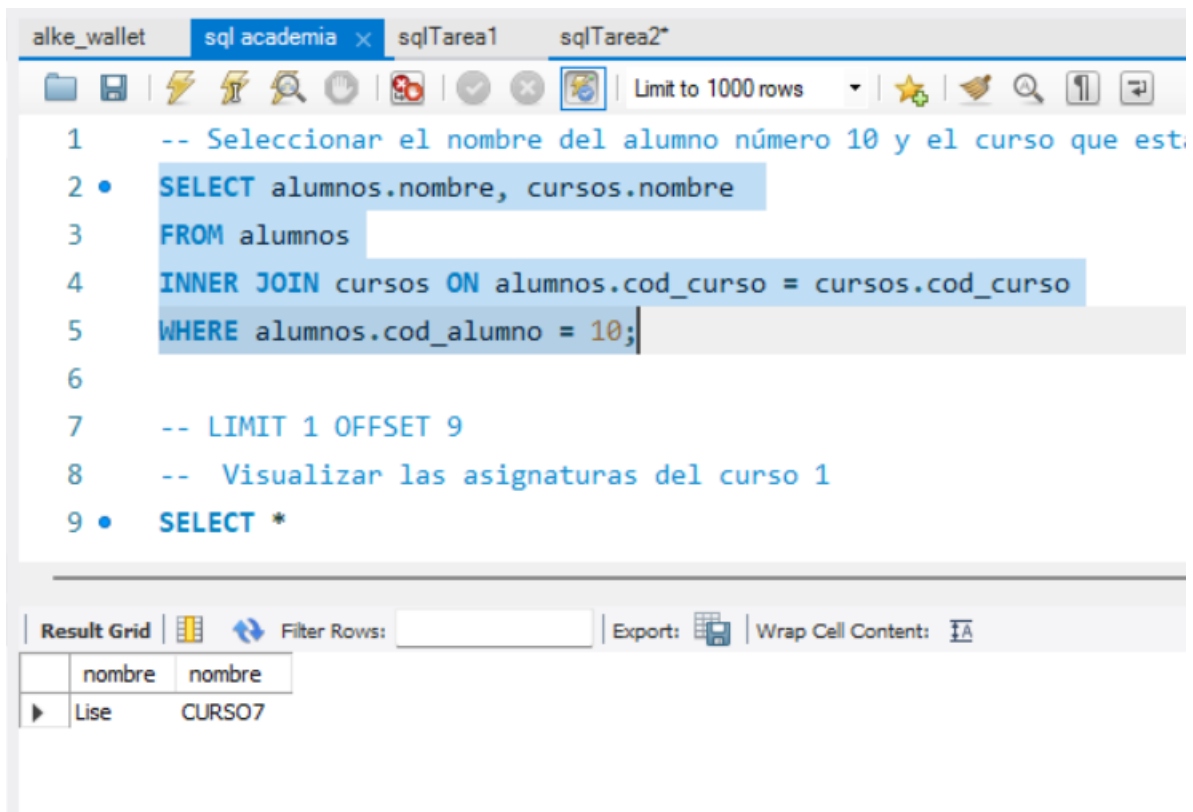
-- Seleccionar el nombre del alumno número 10 y el curso que está haciendo

SELECT alumnos.nombre, cursos.nombre

FROM alumnos

INNER JOIN cursos ON alumnos.cod_curso = cursos.cod_curso

WHERE alumnos.cod_alumno = 10;



The screenshot shows a SQL IDE interface with multiple tabs: 'alke_wallet', 'sql academia', 'sqlTarea1', and 'sqlTarea2*'. The 'sql academia' tab is active, displaying a SQL query. The query is as follows:

```
1  -- Seleccionar el nombre del alumno número 10 y el curso que est
2  • SELECT alumnos.nombre, cursos.nombre
3  FROM alumnos
4  INNER JOIN cursos ON alumnos.cod_curso = cursos.cod_curso
5  WHERE alumnos.cod_alumno = 10;
6
7  -- LIMIT 1 OFFSET 9
8  -- Visualizar las asignaturas del curso 1
9  • SELECT *
```

Below the query editor, there is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'nombre' and 'nombre'. The first row shows the name 'Lise' and the course 'CURSO7'.

	nombre	nombre
▶	Lise	CURSO7

-- LIMIT 1 OFFSET 9

-- Visualizar las asignaturas del curso 1

SELECT *

FROM asignaturas

WHERE asignaturas.cod_curso=1;

```
7  -- LIMIT 1 OFFSET 9
8  -- Visualizar las asignaturas del curso 1
9  • SELECT *
10 FROM asignaturas
11 WHERE asignaturas.cod_curso=1;
12
13 -- Visualizar el número de asignaturas de
14 • SELECT cursos.cod_curso, cursos.nombre, cur:
15 FROM cursos
```

Result Grid				
Filter Rows:				
Edit:				
cod_asignatura	nombre	cod_curso	cod_profesor	duracion
3	historia	1	5	11
7	matematicas	1	2	35
NULL	NULL	NULL	NULL	NULL

-- Visualizar el número de asignaturas de cada curso.

SELECT cursos.cod_curso, cursos.nombre, cursos.precio, COUNT(*) AS 'N° de asignaturas'

FROM cursos

INNER JOIN asignaturas ON asignaturas.cod_curso = cursos.cod_curso

GROUP BY cursos.cod_curso, cursos.nombre, cursos.precio;

```
13 -- Visualizar el número de asignaturas de cada curso.
14 • SELECT cursos.cod_curso, cursos.nombre, cursos.precio, COUNT(*) AS 'N° de asignaturas'
15 FROM cursos
16 INNER JOIN asignaturas ON asignaturas.cod_curso = cursos.cod_curso
17 GROUP BY cursos.cod_curso, cursos.nombre, cursos.precio;
18
```







Result Grid Filter Rows: Export: Wrap Cell Content:				
	cod_curso	nombre	precio	N° de asignaturas
▶	1	CURSO1	194	2
	2	CURSO2	104	6
	3	CURSO3	121	5
	4	CURSO4	121	5
	5	CURSO5	105	8
	6	CURSO6	177	3
	7	CURSO7	128	8
	8	CURSO8	160	2
	9	CURSO9	136	3
	10	CURSO10	119	8

-- Visualizar las asignaturas del profesor con el código 2

SELECT * FROM asignaturas

WHERE asignaturas.cod_profesor = 2;

```
19  -- Visualizar las asignaturas del profesor con el código 2
20 • SELECT * FROM asignaturas
21   WHERE asignaturas.cod_profesor = 2;
22
23  -- Visualizar las notas del alumno 10
24 • SELECT nombre, nota
```

Result Grid  Filter Rows: <input type="text"/> Edit:    Export/Import:   Wrap C					
	cod_asignatura	nombre	cod_curso	cod_profesor	duracion
▶	7	matematicas	1	2	35
	9	dibujo	10	2	12
	10	fisica	3	2	13
	23	laboratorio	2	2	13
	34	dibujo	10	2	37
	49	Informatica	4	2	25
*	NULL	NULL	NULL	NULL	NULL

-- Visualizar las notas del alumno 10

SELECT nombre, nota

FROM alumnos

INNER JOIN notas_alumnos ON alumnos.cod_alumno = notas_alumnos.cod_alumno

WHERE alumnos.cod_alumno = 10;

```
23 -- Visualizar las notas del alumno 10
24 • SELECT nombre, nota
25 FROM alumnos
26 INNER JOIN notas_alumnos ON alumnos.cod_alumno = notas_alumnos.cod_alumno
27 WHERE alumnos.cod_alumno = 10;
28
29 -- Mostrar el curso y el promedio de notas de los alumnos
30 • SELECT cursos.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	nombre	nota			
►	Lise	4			
	Lise	2			
	Lise	6			
	Lise	9			
	Lise	1			
	Lise	7			
	Lise	9			
	Lise	5			

-- Mostrar el curso y el promedio de notas de los alumnos

SELECT cursos.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'

FROM cursos

INNER JOIN notas_alumnos ON cursos.cod_curso = notas_alumnos.cod_curso

GROUP BY cursos.nombre;

```
29  -- Mostrar el curso y el promedio de notas de los alumnos
30 • SELECT cursos.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'
31 FROM cursos
32 INNER JOIN notas_alumnos ON cursos.cod_curso = notas_alumnos.cod_curso
33 GROUP BY cursos.nombre;
34
35  -- Sacar el promedio, pero en este caso por asignatura.
36 • SELECT asignaturas.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	curso	notas			
▶	CURSO1	5.6000			
	CURSO2	5.8667			
	CURSO3	6.0571			
	CURSO4	5.2667			
	CURSO5	5.1500			
	CURSO6	5.6364			
	CURSO7	5.6429			
	CURSO8	4.8750			
	CURSO9	5.0000			
	CURSO10	5.3482			

-- Sacar el promedio, pero en este caso por asignatura.

```
SELECT asignaturas.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'
```

```
FROM asignaturas
```

```
INNER JOIN notas_alumnos ON asignaturas.cod_curso = notas_alumnos.cod_curso
```

```
GROUP BY asignaturas.nombre;
```

```
35 -- Sacar el promedio, pero en este caso por asignatura.  
36 • SELECT asignaturas.nombre AS 'curso', AVG(notas_alumnos.nota) AS 'notas'  
37 FROM asignaturas  
38 INNER JOIN notas_alumnos ON asignaturas.cod_curso = notas_alumnos.cod_curso  
39 GROUP BY asignaturas.nombre;  
40  
41 -- Inserta un par de filas en la tabla CURSOS. Esto permite tener dos cursos  
42 • insert into cursos values(11,'curso11',100);
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	curso	notas			
▶	laboratorio	5.5376			
	Informatica	5.4529			
	historia	5.4549			
	plastica	5.5474			
	dibujo	5.4033			
	matematicas	5.3314			
	ciencias naturales	5.4340			
	fisicia	5.5187			
	literatura	5.6031			

-- Inserta un par de filas en la tabla CURSOS. Esto permite tener dos cursos que no tienen curso

```
insert into cursos values(11,'curso11',100);
```

```
insert into cursos values(12,'curso12',200);
```

-- Averiguar ahora el número de alumnos por curso. Debes usar la cláusula

-- USING para hacer la join (Investiga sobre la clausaula USING y aplícala).

```
SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'
```

```
FROM alumnos
```

```
JOIN cursos USING (cod_curso)
```

```
GROUP BY cursos.nombre;
```

```
45 -- Averiguar ahora el número de alumnos por curso. Debes usar la cláusula
46 -- USING para hacer la join ( Investiga sobre la clausaula USING y aplícala).
47 • SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'
48 FROM alumnos
49 JOIN cursos USING (cod_curso)
50 GROUP BY cursos.nombre;
51
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	nombre	total alumnos			
+	CURSO1	11			
	CURSO2	5			
	CURSO3	7			
	CURSO4	12			
	CURSO5	5			
	CURSO6	12			
	CURSO7	15			
	CURSO8	9			
	CURSO9	10			
	CURSO10	14			

-- Comprueba si aparecen los nuevos cursos. Hazlo con Left Join y deben aparecer 0 alumnos.

SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'

FROM cursos

JOIN alumnos USING (cod_curso)

GROUP BY cursos.nombre;

```
52 -- Comprueba si aparecen los nuevos cursos. Hazlo con Left Join y deben aparecer 0 alumnos.
53 • SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'
54 FROM cursos
55 JOIN alumnos USING (cod_curso)
56 GROUP BY cursos.nombre;
57
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	nombre	total alumnos			
▶	CURSO1	11			
	CURSO2	5			
	CURSO3	7			
	CURSO4	12			
	CURSO5	5			
	CURSO6	12			
	CURSO7	15			
	CURSO8	9			
	CURSO9	10			
	CURSO10	14			

-- Comprueba si aparecen los nuevos cursos. Hazlo con Left Join y deben aparecer 0 alumnos

SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'

FROM cursos

LEFT JOIN alumnos USING (cod_curso)

GROUP BY cursos.nombre;

```
58 -- Comprueba si aparecen los nuevos cursos. Hazlo con Left Join y deben aparecer 0 alumnos
59 • SELECT cursos.nombre, count(alumnos.cod_curso) AS 'total alumnos'
60 FROM cursos
61 LEFT JOIN alumnos USING (cod_curso)
62 GROUP BY cursos.nombre;
63
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	nombre	total alumnos			
▶	CURSO1	11			
	CURSO2	5			
	CURSO3	7			
	CURSO4	12			
	CURSO5	5			
	CURSO6	12			
	CURSO7	15			
	CURSO8	9			
	CURSO9	10			
	CURSO10	14			
	curso11	0			
	curso12	0			

/*

Obtén cada asignatura con sus profesores. Hay algunos profesores que no aparecen ya que no tienen asignaturas todavía. Los debes incluir con un

RIGHT JOIN

*/

SELECT profesores.nombre AS 'Nombre Profesor',

profesores.apellidos AS 'Apellido Profesor',

asignaturas.nombre AS 'Asignatura'

FROM asignaturas

RIGHT JOIN profesores USING (cod_profesor);

```
64  /*
65  Obtén cada asignatura con sus profesores. Hay algunos profesores que no
66  aparecen ya que no tienen asignaturas todavía. Los debes incluir con un
67  RIGHT JOIN
68  */
69  • SELECT profesores.nombre AS 'Nombre Profesor',
70      profesores.apellidos AS 'Apellido Profesor',
71      asignaturas.nombre AS 'Asignatura'
72  FROM asignaturas
```

Result Grid Filter Rows: Export: Wrap Cell Content:			
	Nombre Profesor	Apellido Profesor	Asignatura
►	Cindelyn	Dawson	literatura
	Cindelyn	Dawson	plastica
	Cindelyn	Dawson	laboratorio
	Dreddy	Rigg	matematicas
	Dreddy	Rigg	dibujo
	Dreddy	Rigg	fisica
	Dreddy	Rigg	laboratorio
	Dreddy	Rigg	dibujo
	Dreddy	Rigg	Informatica
	Georgena	Chown	NULL
	Euphemia	Lavrinov	literatura
	Euphemia	Lavrinov	dibujo
	Euphemia	Lavrinov	plastica