

TRABAJO PRÁCTICO – Modulo 5 | Parte 2

JOINS – Uniendo Tablas

Alumno: Ignacio Figueroa – 45.406.120

Tecnicatura Universitaria en Programación – UTN

Materia: Base de Datos I

Comisión: 8

Introducción:

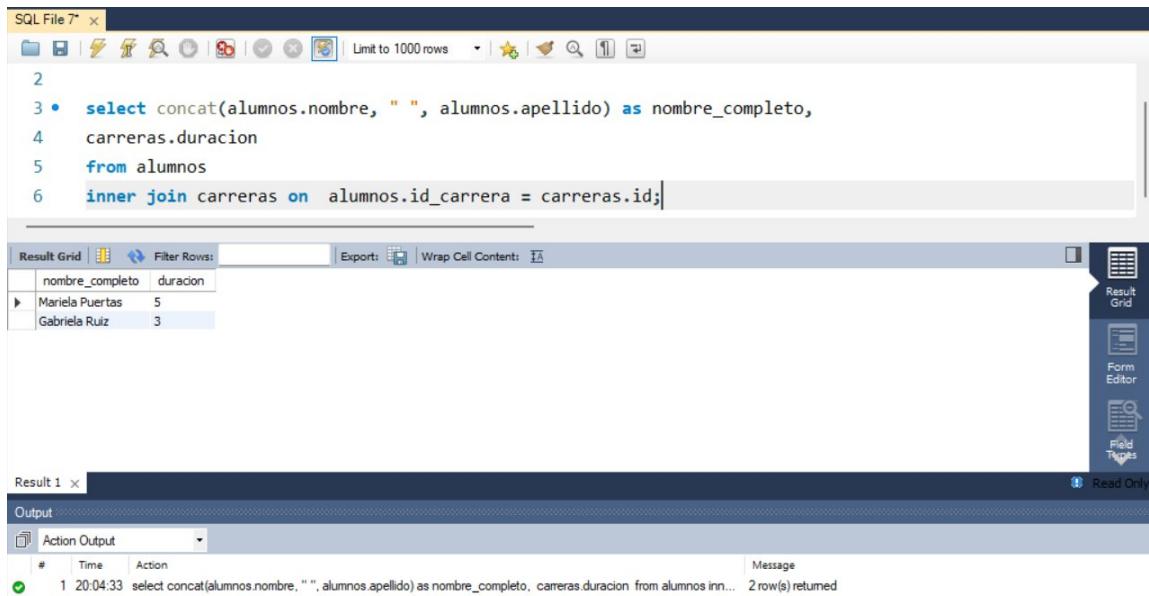
En este trabajo se aplican los distintos tipos de **JOINS** en MySQL:

INNER JOIN, LEFT JOIN y RIGHT JOIN, que permiten unir datos de varias tablas relacionadas entre sí dentro de una base de datos relacional.

Además, se realizarán consultas con **múltiples JOINS** para obtener información combinada de varias tablas.

Todas las consultas se realizan sobre la base de datos **gestionacademica** utilizada en los ejercicios anteriores.

Ejercicio 1: Mostrar el nombre completo del alumno y la duración de su carrera.



The screenshot shows the MySQL Workbench interface with two tabs: 'SQL File 7' and 'Result Grid'. The SQL tab contains the following query:

```

2
3 • select concat(alumnos.nombre, " ", alumnos.apellido) as nombre_completo,
4 carreras.duracion
5 from alumnos
6 inner join carreras on alumnos.id_carrera = carreras.id;

```

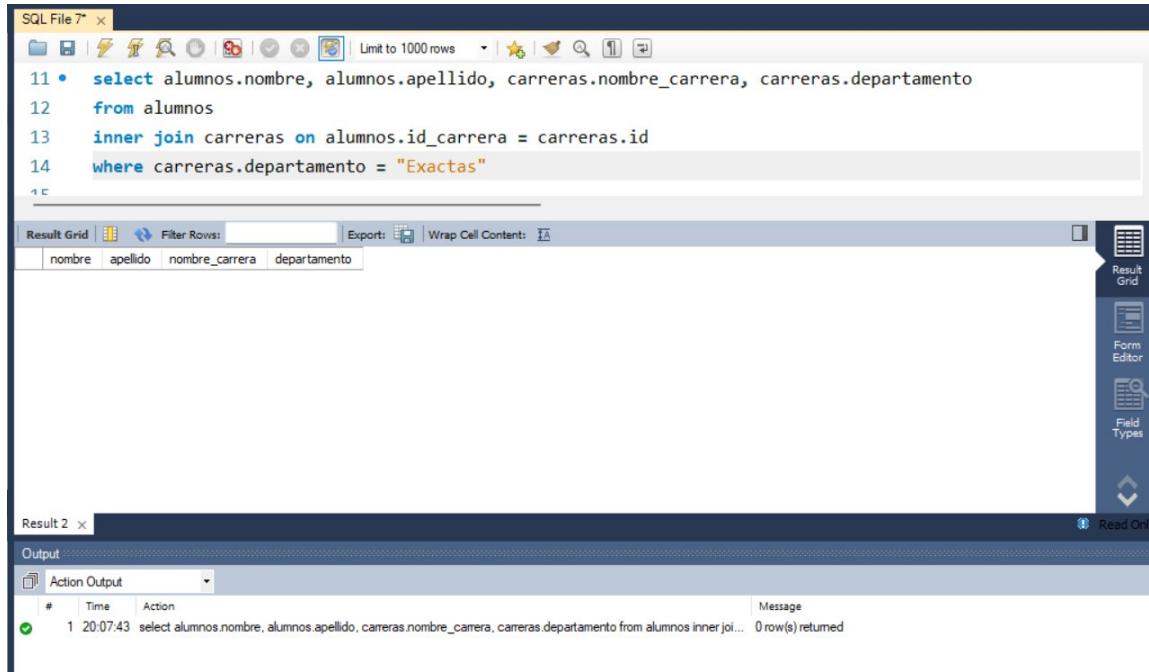
The Result Grid shows the output:

nombre_completo	duracion
Mariela Puertas	5
Gabriela Ruiz	3

The Output tab shows the execution log:

#	Time	Action	Message
1	20:04:33	select concat(alumnos.nombre, " ", alumnos.apellido) as nombre_completo, carreras.duracion from alumnos inn...	2 row(s) returned

Ejercicio 2: Listar alumnos que pertenecen a carreras del departamento 'Exactas'.



The screenshot shows the MySQL Workbench interface with two tabs: 'SQL File 7' and 'Result Grid'. The SQL tab contains the following query:

```

11 • select alumnos.nombre, alumnos.apellido, carreras.nombre_carrera, carreras.departamento
12 from alumnos
13 inner join carreras on alumnos.id_carrera = carreras.id
14 where carreras.departamento = "Exactas"

```

The Result Grid shows the output:

nombre	apellido	nombre_carrera	departamento

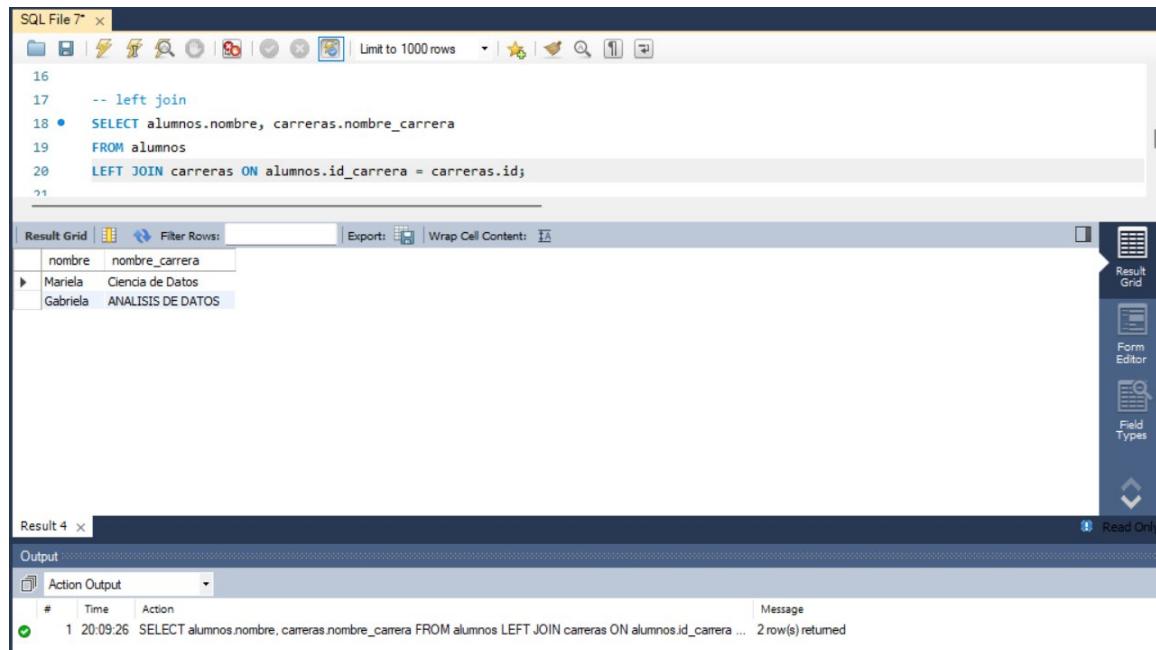
The Output tab shows the execution log:

#	Time	Action	Message
1	20:07:43	select alumnos.nombre, alumnos.apellido, carreras.nombre_carrera, carreras.departamento from alumnos inner joi...	0 row(s) returned

En mi caso no aparece ninguno porque no agregue ninguno en el trabajo anterior.

LEFT JOIN:

El **LEFT JOIN** muestra todos los registros de la tabla izquierda (alumnos) aunque no tengan coincidencia en la derecha (carreras).



```

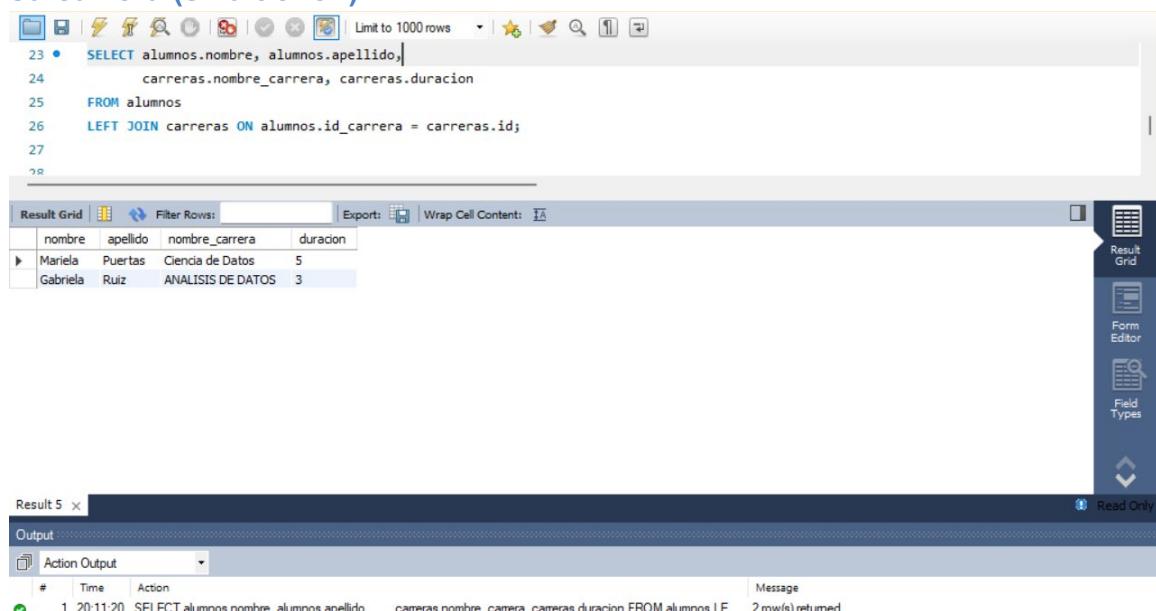
SQL File 7* x
16
17  -- left join
18 •  SELECT alumnos.nombre, carreras.nombre_carrera
19   FROM alumnos
20   LEFT JOIN carreras ON alumnos.id_carrera = carreras.id;
21

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
nombre      nombre_carrera
▶ Mariela    Ciencia de Datos
▶ Gabriela   ANALISIS DE DATOS

Result 4 x
Output
Action Output
# Time Action
1 20:09:26 SELECT alumnos.nombre, carreras.nombre_carrera FROM alumnos LEFT JOIN carreras ON alumnos.id_carrera ... 2 row(s) returned

```

Ejercicio 3: Mostrar todos los alumnos junto con el nombre y duración de su carrera (si la tienen).



```

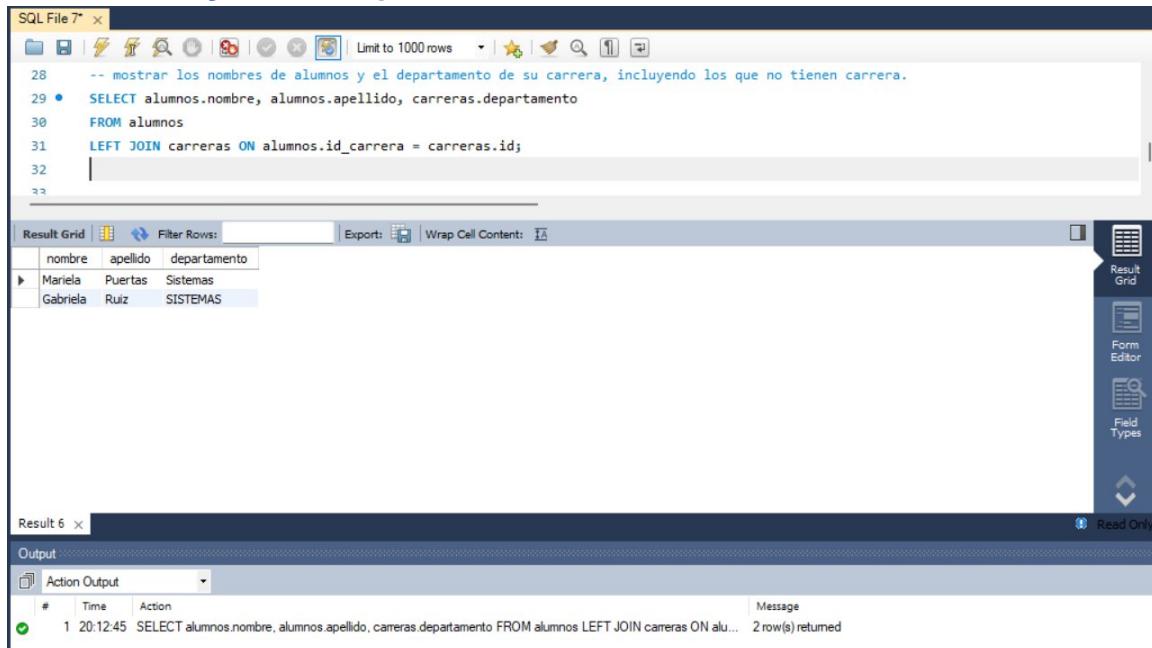
SQL File 8* x
23 •  SELECT alumnos.nombre, alumnos.apellido,
24           carreras.nombre_carrera, carreras.duracion
25   FROM alumnos
26   LEFT JOIN carreras ON alumnos.id_carrera = carreras.id;
27
28

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |
nombre      apellido     nombre_carrera  duracion
▶ Mariela    Puertas     Ciencia de Datos  5
▶ Gabriela   Ruiz        ANALISIS DE DATOS 3

Result 5 x
Output
Action Output
# Time Action
1 20:11:20 SELECT alumnos.nombre, alumnos.apellido,     carreras.nombre_carrera, carreras.duracion FROM alumnos LE... 2 row(s) returned

```

Ejercicio 4: Mostrar los nombres de alumnos y el departamento de su carrera, incluyendo los que no tienen carrera.



The screenshot shows the SQL Developer interface with a query window titled "SQL File 7*". The code is:

```

28 -- mostrar los nombres de alumnos y el departamento de su carrera, incluyendo los que no tienen carrera.
29 • SELECT alumnos.nombre, alumnos.apellido, carreras.departamento
30   FROM alumnos
31  LEFT JOIN carreras ON alumnos.id_carrera = carreras.id;
32
33

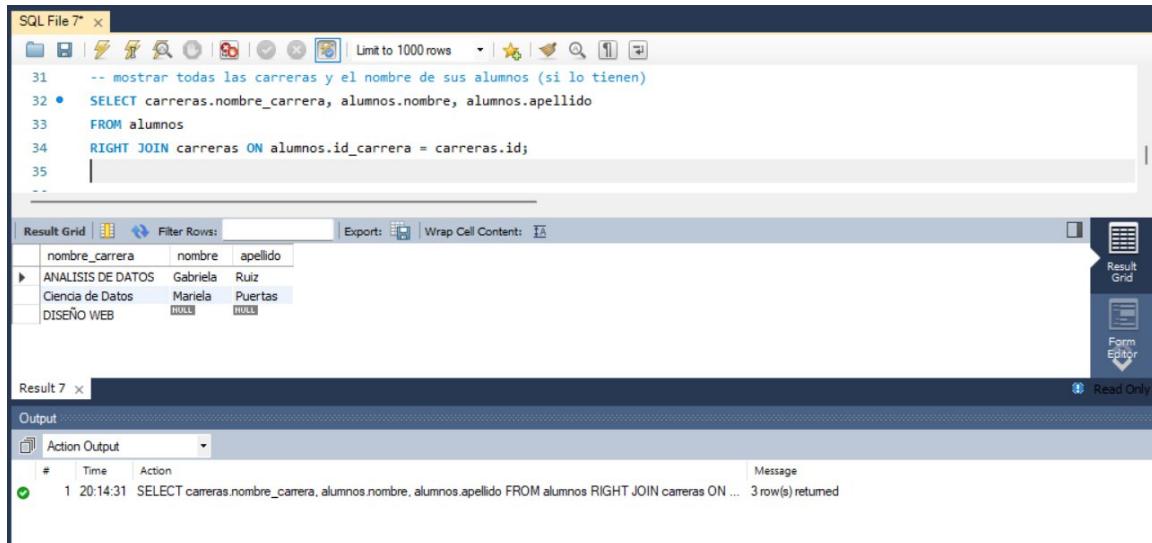
```

The results grid shows the following data:

nombre	apellido	departamento
Mariela	Puertas	Sistemas
Gabriela	Ruiz	SISTEMAS

The output window shows the message: "1 20:12:45 SELECT alumnos.nombre, alumnos.apellido, carreras.departamento FROM alumnos LEFT JOIN carreras ON alu... 2 row(s) returned".

Ejercicio 5: Mostrar todas las carreras y el nombre de sus alumnos (si los tienen).



The screenshot shows the SQL Developer interface with a query window titled "SQL File 7*". The code is:

```

31 -- mostrar todas las carreras y el nombre de sus alumnos (si lo tienen)
32 • SELECT carreras.nombre_carrera, alumnos.nombre, alumnos.apellido
33   FROM alumnos
34  RIGHT JOIN carreras ON alumnos.id_carrera = carreras.id;
35
36

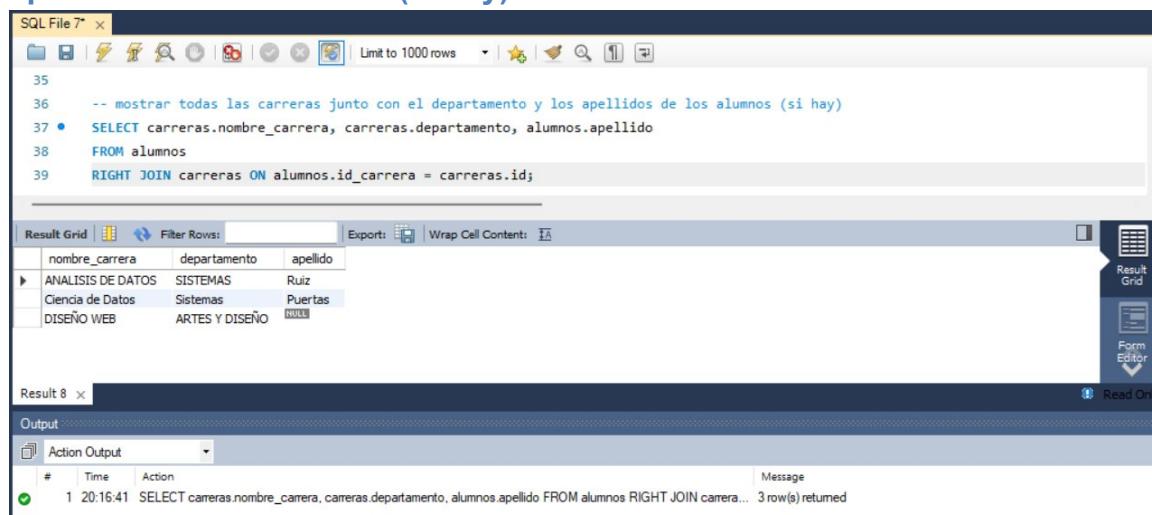
```

The results grid shows the following data:

nombre_carrera	nombre	apellido
ANALISIS DE DATOS	Gabriela	Ruiz
Ciencia de Datos	Mariela	Puertas
DISEÑO WEB	Maria	Maria

The output window shows the message: "1 20:14:31 SELECT carreras.nombre_carrera, alumnos.nombre, alumnos.apellido FROM alumnos RIGHT JOIN carreras ON ... 3 row(s) returned".

Ejercicio 6: Mostrar todas las carreras junto con el departamento y los apellidos de los alumnos (si hay).



The screenshot shows a SQL editor window titled "SQL File 7" with the following SQL code:

```
35
36 -- mostrar todas las carreras junto con el departamento y los apellidos de los alumnos (si hay)
37 • SELECT carreras.nombre_carrera, carreras.departamento, alumnos.apellido
38   FROM alumnos
39   RIGHT JOIN carreras ON alumnos.id_carrera = carreras.id;
```

The results grid displays the following data:

nombre_carrera	departamento	apellido
ANALISIS DE DATOS	SISTEMAS	Ruiz
Glencia de Datos	Sistemas	Puertas
DISEÑO WEB	ARTES Y DISEÑO	NULL

The output pane shows the following log entry:

#	Time	Action
1	20:16:41	SELECT carreras.nombre_carrera, carreras.departamento, alumnos.apellido FROM alumnos RIGHT JOIN carrera...

Message: 3 row(s) returned