

## Ejercicio interactivo de consultas sql

Evidencia:

### Creación de la base de datos y tablas

The screenshot shows a SQL IDE interface. On the left, a tree view displays the database structure, including a 'universidad' database with tables like 'departamento', 'persona', and 'profesor'. The main editor displays the following SQL code:

```
1 DROP DATABASE IF EXISTS universidad;
2 CREATE DATABASE universidad CHARACTER SET utf8mb4;
3 USE universidad;
4
5 CREATE TABLE departamento (
6     id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
7     nombre VARCHAR(50) NOT NULL
8 );
9
10 CREATE TABLE persona (
11     id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
12     nif VARCHAR(9) UNIQUE,
13     nombre VARCHAR(25) NOT NULL,
14     apellido1 VARCHAR(50) NOT NULL,
15     apellido2 VARCHAR(50),
16     ciudad VARCHAR(25) NOT NULL,
17     direccion VARCHAR(50) NOT NULL,
18     telefono VARCHAR(9),
19     fecha_nacimiento DATE NOT NULL,
20     sexo ENUM('H', 'M') NOT NULL,
21     tipo ENUM('profesor', 'alumno') NOT NULL
22 );
23
24 CREATE TABLE profesor (
25     id_profesor INT UNSIGNED PRIMARY KEY,
26     id_departamento INT UNSIGNED NOT NULL,
27     FOREIGN KEY (id_departamento) REFERENCES departamento(id)
28 );
```

Below the code editor, a filter box is set to 'Expresión regular'. At the bottom, a list of SQL commands is shown:

```
223 SHOW TABLE STATUS FROM `universidad`;
224 SHOW FUNCTION STATUS WHERE `Db`='universidad';
225 SHOW PROCEDURE STATUS WHERE `Db`='universidad';
226 SHOW TRIGGERS FROM `universidad`;
227 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='universidad';
```

### Integración de datos a las tablas

The screenshot shows the same SQL IDE interface, but now with data being inserted into the tables. The main editor displays the following SQL code:

```
64 /* Departamento */
65
66 INSERT INTO departamento VALUES (1, 'Informática');
67 INSERT INTO departamento VALUES (2, 'Matemáticas');
68 INSERT INTO departamento VALUES (3, 'Economía y Empresa');
69 INSERT INTO departamento VALUES (4, 'Educación');
70 INSERT INTO departamento VALUES (5, 'Agronomía');
71 INSERT INTO departamento VALUES (6, 'Química y Física');
72 INSERT INTO departamento VALUES (7, 'Filología');
73 INSERT INTO departamento VALUES (8, 'Derecho');
74 INSERT INTO departamento VALUES (9, 'Biología y Geología');
75
76 /* Persona */
77 INSERT INTO persona VALUES (1, '26902806M', 'Salvador', 'Sánchez', 'Pérez', 'Almería', 'C/ Real del bar
78 INSERT INTO persona VALUES (2, '89542419S', 'Juan', 'Saez', 'Vega', 'Almería', 'C/ Mercurio', '61825387
79 INSERT INTO persona VALUES (3, '11105554G', 'Zoe', 'Ramirez', 'Gea', 'Almería', 'C/ Marte', '618223876'
80 INSERT INTO persona VALUES (4, '17105885A', 'Pedro', 'Heller', 'Pagac', 'Almería', 'C/ Estrella fugaz',
81 INSERT INTO persona VALUES (5, '38223286T', 'David', 'Schmidt', 'Fisher', 'Almería', 'C/ Venus', '67851
82 INSERT INTO persona VALUES (6, '04233869Y', 'José', 'Koss', 'Bayer', 'Almería', 'C/ Júpiter', '62834959
83 INSERT INTO persona VALUES (7, '97258166K', 'Ismael', 'Strosin', 'Turcotte', 'Almería', 'C/ Neptuno', N
84 INSERT INTO persona VALUES (8, '79503962T', 'Cristina', 'Lemke', 'Rutherford', 'Almería', 'C/ Saturno',
85 INSERT INTO persona VALUES (9, '82842571K', 'Ramón', 'Herzog', 'Tremblay', 'Almería', 'C/ Urano', '6263
86 INSERT INTO persona VALUES (10, '61142000L', 'Esther', 'Spencer', 'Lakin', 'Almería', 'C/ Plutón', NULL
87 INSERT INTO persona VALUES (11, '46900725E', 'Daniel', 'Herman', 'Pacocho', 'Almería', 'C/ Andarax', '6
88 INSERT INTO persona VALUES (12, '85366986W', 'Carmen', 'Streich', 'Hirthe', 'Almería', 'C/ Alanzora', '6
89 INSERT INTO persona VALUES (13, '73571384L', 'Alfredo', 'Stiedemann', 'Morissette', 'Almería', 'C/ Guad
90 INSERT INTO persona VALUES (14, '82937751G', 'Manolo', 'Hamill', 'Kozey', 'Almería', 'C/ Duero', '95026
```

Below the code editor, a filter box is set to 'Expresión regular'. At the bottom, a list of SQL commands is shown:

```
223 SHOW TABLE STATUS FROM `universidad`;
224 SHOW FUNCTION STATUS WHERE `Db`='universidad';
225 SHOW PROCEDURE STATUS WHERE `Db`='universidad';
226 SHOW TRIGGERS FROM `universidad`;
227 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='universidad';
```

1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre

SQL 1\*

```
1 /*1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre*/
2 SELECT apellido1, apellido2, nombre
3 FROM persona
4 WHERE tipo = 'alumno'
5 ORDER BY apellido1, apellido2, nombre;
6
7
```

	apellido1	apellido2	nombre
1	Dominguez	Guerrero	Antonio
2	Gea	Ruiz	Sonia
3	Gutiérrez	López	Juan
4	Heller	Pagac	Pedro
5	Herman	Pacocha	Daniel
6	Hernández	Martínez	Irene
7	Herrera	Tramblay	Ramón

Execution finished without errors.  
Result: 12 rows returned in 73ms  
At line 1:  
/\*Listado ordenado de alumnos por primer apellido, segundo apellido y nombre\*/  
SELECT apellido1, apellido2, nombre

2) Alumnos que no han dado de alta su número de teléfono

SQL 1\*

```
3 FROM persona
4 WHERE tipo = 'alumno'
5 ORDER BY apellido1, apellido2, nombre;*/
6
7 /*2. Alumnos que no han dado de alta su número de teléfono*/
8 SELECT nombre, apellido1, apellido2
9 FROM persona
10 WHERE tipo = 'alumno' AND telefono IS NULL;
11 $$$
12
13
```

	nombre	apellido1	apellido2
1	Pedro	Heller	Pagac
2	Ismael	Strosin	Turcotte

Execution finished without errors.  
Result: 2 rows returned in 14ms  
At line 1:  
/\*1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre\*/  
/\*SELECT apellido1, apellido2, nombre

### 3) Alumnos nacidos en 1999

```
11
12      /*Alumnos nacidos en 1999*/
13      SELECT nombre, apellido1, apellido2
14      FROM persona
15      WHERE tipo = 'alumno' AND strftime('%Y', fecha_nacimiento) = '1999';
16
```

nombre	apellido1	apellido2
--------	-----------	-----------

Execution finished without errors.  
Result: 0 rows returned in 9ms  
At line 1:  
/\*1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre\*/  
/\*SELECT apellido1, apellido2, nombre  
FROM

### 4) Profesores sin número de teléfono y con NIF terminado en "K"

```
SQL 1*
12      /*Alumnos nacidos en 1999*/
13      /*SELECT nombre, apellido1, apellido2
14      FROM persona
15      WHERE tipo = 'alumno' AND strftime('%Y', fecha_nacimiento) = '1999';*/
16
17
18      /*Profesores sin número de teléfono y con NIF terminado en "K"*/
19      SELECT nombre, apellido1, apellido2
20      FROM persona
21      WHERE tipo = 'profesor' AND telefono IS NULL AND nif LIKE '%K';
22
23
```

	nombre	apellido1	apellido2
1	Antonio	Fahey	Considine
2	Guillermo	Ruecker	Upton

Execution finished without errors.  
Result: 2 rows returned in 11ms  
At line 1:  
/\*1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre\*/  
/\*SELECT apellido1, apellido2, nombre  
FROM

5) Asignaturas del primer cuatrimestre, tercer curso, grado con ID 7

SQL 1\*

```

19  /*SELECT nombre, apellidol, apellido2
20  FROM persona
21  WHERE tipo = 'profesor' AND telefono IS NULL AND nif LIKE '%K';*/
22
23  /*Asignaturas del primer cuatrimestre, tercer curso, grado con ID 7*/
24  SELECT nombre
25  FROM asignatura
26  WHERE cuatrimestre = 1 AND curso = 3 AND id_grado = 7;
27
28
29

```

	id	nombre	creditos	tipo	curso	cuatrimestre	id_profesor	id_grado
55	55	Química general	6.0	básica	1	1	NULL	7
56	56	Química orgánica	6.0	básica	1	1	NULL	7
57	57	Biología vegetal y animal	6.0	básica	1	2	NULL	7
58	58	Bioquímica	6.0	básica	1	2	NULL	7
59	59	Genética	6.0	básica	1	2	NULL	7
60	60	Matemáticas II	6.0	básica	1	2	NULL	7
61	61	Microbiología	6.0	básica	1	2	NULL	7

Execution finished without errors.  
Result: 83 rows returned in 21ms  
At line 1:  
/\*1) Listado ordenado de alumnos por primer apellido, segundo apellido y nombre\*/  
/\*SELECT apellidol, apellido2, nombre