

## TAREA EVALUABLE 2

Bases de Datos  
CFGs DAW

*Las soluciones no son absolutas y pueden ser distintas en algunos casos.  
La mayoría de las soluciones propuestas se muestran en nomenclatura explícita e implícita.*

## Parte I

1. Listado de profesores con nombre, apellido1 y el departamento al que pertenecen. (13 rows)

```
SELECT nombre,apellido1,departamento
FROM usuarios u INNER JOIN profesores p
ON u.dni=p.dni;
```

```
SELECT nombre,apellido1,departamento
FROM usuarios u , profesores p
WHERE u.dni=p.dni;
```

nombre	apellido1	departamento
ROSARIO	IBÁÑEZ	AFI
LUIS	PARRA	AFI
BEATRIZ	SORIANO	COM
ALBERTO	DOMINGUEZ	COM
JAVIER	MONTES	COM
JULIA	SANCHEZ	COM
SARA	ALVAREZ	FOL
VICENTE	LARA	FOL
VICENTE	RIVERA	INF
CARMEN	GALLEGO	INF
SUSANA	ALONSO	INF
ROSARIO	ROBLES	TUR
NURIA	GARRIDO	TUR

2. Listado de reservas de aulas (con el nombre del profesor que hace la reserva) ordenados por aula. (31 rows)

```
SELECT cod_aula,nombre,apellido1, fecha
FROM usuarios u INNER JOIN profesores p ON u.dni=p.dni
INNER JOIN reserva_aula ON p.dni=dni_profesor
ORDER BY cod_aula;
```

```
SELECT cod_aula,nombre,apellido1, fecha
FROM usuarios u, profesores p, reserva_aula
WHERE u.dni=p.dni AND p.dni=dni_profesor
ORDER BY cod_aula;
```

cod_aula	nombre	apellido1	fecha
A01	NURIA	GARRIDO	2020-10-06 16:00:00
A02	NURIA	GARRIDO	2020-10-13 16:00:00
A02	VICENTE	RIVERA	2020-10-19 11:10:00
A02	VICENTE	RIVERA	2020-10-20 11:10:00
A02	ALBERTO	DOMINGUEZ	2020-11-12 09:55:00
A02	ALBERTO	DOMINGUEZ	2020-12-10 09:55:00
A02	BEATRIZ	SORIANO	2021-01-25 11:10:00
A03	SARA	ALVAREZ	2021-01-21 12:05:00
A11	NURIA	GARRIDO	2020-10-20 16:00:00
A11	NURIA	GARRIDO	2020-10-27 16:00:00
A12	ROSARIO	ROBLES	2020-11-10 19:05:00
A12	BEATRIZ	SORIANO	2021-01-18 11:10:00
A13	SARA	ALVAREZ	2020-10-05 09:00:00
A13	JULIA	SANCHEZ	2020-11-30 18:10:00
A14	BEATRIZ	SORIANO	2020-11-10 12:05:00
A14	ROSARIO	IBÁÑEZ	2021-01-18 11:10:00
A14	JULIA	SANCHEZ	2021-02-10 13:00:00
A15	ROSARIO	IBÁÑEZ	2020-02-10 18:10:00
A15	JAVIER	MONTES	2020-12-16 09:00:00
B01	ROSARIO	IBÁÑEZ	2020-11-26 16:55:00
B02	NURIA	GARRIDO	2020-11-03 16:00:00
B02	ROSARIO	ROBLES	2021-01-13 18:10:00
B03	BEATRIZ	SORIANO	2020-10-06 09:55:00
B03	NURIA	GARRIDO	2020-11-10 16:00:00
B04	JAVIER	MONTES	2021-02-03 11:10:00
B04	BEATRIZ	SORIANO	2021-02-03 18:10:00
B11	SARA	ALVAREZ	2020-10-07 18:10:00
B12	NURIA	GARRIDO	2020-11-17 16:00:00
B12	JULIA	SANCHEZ	2021-01-20 11:10:00
B13	VICENTE	RIVERA	2021-01-27 16:55:00
B14	ROSARIO	IBÁÑEZ	2020-12-14 11:10:00

3. Listado de profesores y los módulos que imparten, ambos con los nombres. (29 rows)

```
SELECT nombre,apellido1,module
FROM usuarios u
INNER JOIN profesores p ON u.dni=p.dni
INNER JOIN imparten i ON p.dni=dni_profesor
INNER JOIN modulos m ON i.cod_mod=m.cod_mod;
```

```
SELECT nombre,apellido1,module
FROM usuarios u, profesores p, imparten i, modulos m
WHERE u.dni=p.dni AND p.dni=dni_profesor AND i.cod_mod=m.cod_mod;
```

nombre	apellido1	modulo
ROSARIO	IBAÑEZ	Gestion financiera
ROSARIO	IBAÑEZ	Simulacion empresarial
LUIS	PARRA	Comunicacion y atencion al cliente
LUIS	PARRA	Gestion de recursos humanos
BEATRIZ	SORIANO	Comercio digital internacional
BEATRIZ	SORIANO	Investigacion comercial
BEATRIZ	SORIANO	Politicas de marketing
ALBERTO	DOMINGUEZ	Investigacion comercial
ALBERTO	DOMINGUEZ	Organizacion de equipos de ventas
ALBERTO	DOMINGUEZ	Tecnicas de venta y negociacion
JAVIER	MONTES	Logistica de almacenamiento
JAVIER	MONTES	Negociacion Internacional
JULIA	SANCHEZ	Comercio digital internacional
JULIA	SANCHEZ	Marketing internacional
SARA	ALVAREZ	Desarrollo web en entorno servidor
SARA	ALVAREZ	Empresa e iniciativa emprendedora
VICENTE	LARA	Formacion y orientacion laboral
VICENTE	RIVERA	Desarrollo web en entorno cliente
VICENTE	RIVERA	Lenguajes de marcas
VICENTE	RIVERA	Programacion
CARMEN	GALLEGO	Bases de datos
CARMEN	GALLEGO	Despliegue de aplicaciones web
CARMEN	GALLEGO	Programacion
SUSANA	ALONSO	Desarrollo web en entorno servidor
ROSARIO	ROBLES	Estructura del mercado turistico
ROSARIO	ROBLES	Protocolo y relaciones publicas
NURIA	GARRIDO	Estructura del mercado turistico
NURIA	GARRIDO	Gestion del departamento de pisos
NURIA	GARRIDO	Recepcion y reservas

29 rows in set (0.01 sec)

**4. Nombre de los módulos que imparte el profesor con dni 67538464E. (3 rows)**

```
SELECT modulo FROM profesores p
INNER JOIN imparten i ON p.dni=dni_profesor
INNER JOIN modulos m ON i.cod_mod=m.cod_mod
WHERE dni='67538464E';
```

```
+-----+
| modulo |
+-----+
| Bases de datos |
| Despliegue de aplicaciones web |
| Programacion |
+-----+
3 rows in set (0.00 sec)
```

```
SELECT modulo
FROM profesores p, imparten i, modulos m
WHERE dni='67538464E' AND p.dni=dni_profesor AND i.cod_mod=m.cod_mod;
```

**5. Nombre completo en un único campo (se puede usar la función concat(campo1,campo2..) para unir campos varchar, también acepta texto directamente entre comillas) y dni de ALUMNOS que empiezan con la letra S. (2 rows)**

```
SELECT CONCAT(nombre,' ',apellido1,' ',apellido2) AS 'Nombre Completo', u.dni
FROM usuarios u NATURAL JOIN alumnos a
WHERE nombre LIKE 'S%';
```

```
SELECT CONCAT(nombre,' ',apellido1,' ',apellido2) AS 'Nombre Completo', u.dni
FROM usuarios u, alumnos a
WHERE nombre LIKE 'S%' AND a.dni=u.dni;
```

```
+-----+-----+
| Nombre Completo | dni |
+-----+-----+
| SERGIO VIDAL MENDOZA | 05005100R |
| SONIA HERRERO CARRASCO | 52704314K |
+-----+-----+
2 rows in set (0.00 sec)
```

**6. Id, fecha y alumno de las tutorías pedidas en el mes de diciembre. (80 rows).**

```
SELECT * FROM reserva_ti
WHERE fecha_ti BETWEEN '2020-12-01' AND '2020-12-31';
```

```
+-----+-----+-----+
| fecha_ti | id_ti | dni_alumno |
+-----+-----+-----+
| 2020-12-01 | C015 | 30447095V |
| 2020-12-01 | IT03 | 17179535F |
| 2020-12-01 | TU04 | 77629130J |
+-----+-----+-----+
```

No se muestra la lista completa

```
+-----+-----+-----+
| 2020-12-22 | F004 | 05448421C |
| 2020-12-22 | TU03 | 89665193S |
| 2020-12-23 | IT16 | 41707956D |
+-----+-----+-----+
80 rows in set (0.00 sec)
```

**7. Lista de Id de los horarios de tutoría de los que se ha reservado alguna sesión. (84 rows)**

3 Soluciones, la primera mas sencilla ya que no se piden datos de la TI:

```
SELECT DISTINCT id_ti FROM reserva_ti;
```

```
Select distinct rti.id_ti
FROM ti,reserva_ti rti
WHERE ti.id_ti=rti.id_ti;
```

```
Select distinct rti.id_ti
FROM ti NATURAL JOIN reserva_ti rti
```

id_ti
C011
C012
C013

TU15
TU16
TU18

84 rows in set

**8. Id, día hora y dni del profesor de los horarios de tutoría que no se han pedido todavía en ninguna fecha. (3 rows)**

```
SELECT * FROM ti
WHERE NOT EXISTS (
    SELECT rti.id_ti FROM reserva_ti rti
    WHERE rti.id_ti=ti.id_ti);
```

id_ti	dia	hora	dni_profesor
C032	1	19:05:00	31422905Y
TU17	3	12:05:00	99935360S
F016	3	13:00:00	29318262R

3 rows in set (0.00 sec)

**9. Listado de módulos(nombre) con el total de alumnos que tienen matriculados. (24 rows)**

```
SELECT modulo, count(dni_alumno) AS matriculados
FROM modulos mo INNER JOIN matriculados ma ON mo.cod_mod=ma.cod_mod
GROUP BY mo.cod_mod;
```

modulo	matriculados
Bases de datos	11
Comunicacion y atencion al cliente	5
Comercio digital internacional	6
Despliegue de aplicaciones web	7
Desarrollo web en entorno cliente	8
Desarrollo web en entorno servidor	7
Empresa e iniciativa emprendedora	22
Estructura del mercado turistico	10
Formacion y orientacion laboral	32
Gestion del departamento de pisos	10
Gestion financiera	5
Gestion de recursos humanos	4
Investigacion comercial	10
Logistica de almacenamiento	10
Lenguajes de marcas	10
Marketing internacional	6
Negociacion Internacional	9
Organizacion de equipos de ventas	5
Politicas de marketing	12
Programacion	8
Protocolo y relaciones publicas	10
Recepcion y reservas	10
Simulacion empresarial	6
Tecnicas de venta y negociacion	5

24 rows in set (0.00 sec)

**10. Número de alumnos del módulo con más alumnos.**

```
SELECT MAX(al_mod) AS Maximo_Alumnos FROM
  (SELECT count(dni_alumno) AS al_mod
   FROM matriculados GROUP BY cod_mod) alumxmod;
```

```
SELECT count(dni_alumno) AS Maximo_Alumnos
FROM matriculados GROUP BY cod_mod
HAVING Maximo_Alumnos >= ALL
  (SELECT count(dni_alumno)
   FROM matriculados GROUP BY cod_mod);
```

```
+-----+
| Maximo_Alumnos |
+-----+
|              32 |
+-----+
1 row in set (0.00 sec)
```

**11. Nombre y número de alumnos del módulo con menos alumnos.**

```
SELECT modulo, count(dni_alumno) AS N_Alumnos
FROM matriculados ma, modulos mo
WHERE ma.cod_mod=mo.cod_mod
GROUP BY ma.cod_mod HAVING N_Alumnos =
  (SELECT min(total) FROM
   (SELECT count(dni_alumno) as total
    FROM matriculados GROUP BY cod_mod) alumxmod);
```

```
+-----+-----+
| modulo | N_Alumnos |
+-----+-----+
| Gestion de recursos humanos | 4 |
+-----+-----+
1 row in set (0.00 sec)
```

```
SELECT modulo, count(dni_alumno) AS N_Alumnos
FROM matriculados ma, modulos mo
WHERE ma.cod_mod=mo.cod_mod
GROUP BY ma.cod_mod HAVING N_Alumnos <= ALL
  (SELECT count(dni_alumno)
   FROM matriculados GROUP BY cod_mod);
```

**12. Listado de reserva de aulas que coincide con una tutoría de ese profesor. Es decir datos de la reserva de aula que coincida en fecha y hora con una tutoría reservada para ese mismo profesor. Para poder comparar fechas y horas con la fecha+hora completa de reserva\_aula se pueden usar funciones como DATE(fecha) o TIME(fecha) que extraen sólo esa parte de la información. (1 row)**

```
SELECT ra.* FROM reserva_aula ra, reserva_ti rti, ti
WHERE fecha_ti=DATE(fecha) AND rti.id_ti=ti.id_ti
AND hora=TIME(fecha) AND ra.dni_profesor=ti.dni_profesor;
```

```
+-----+-----+-----+
| cod_aula | dni_profesor | fecha |
+-----+-----+-----+
| B03      | 02165682W   | 2020-10-06 09:55:00 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Con NATURAL JOIN (que en este caso cuadra) se puede simplificar y además al seleccionar todo (\*) los campos iguales en varias tablas no se repiten, por eso esta vez se ha optado por mostrar todo y no sólo los campos de reserva aula (ra.\*) para comprobar mejor que se cumplen las condiciones:

```
SELECT * FROM reserva_aula ra NATURAL JOIN reserva_ti rti NATURAL JOIN ti
WHERE fecha_ti=DATE(fecha) AND hora=TIME(fecha);
```

dni_profesor	id_ti	cod_aula	fecha	fecha_ti	dni_alumno	día	hora
02165682W	C015	B03	2020-10-06 09:55:00	2020-10-06	32077097D	1	09:55:00

1 row in set (0.00 sec)

Otra opción es incluir la tabla de profesores también, ya que en la teoría une ti con reserva\_aula, pero en la práctica para este caso no aporta nada

### 13. Nombre y apellidos del alumno que ha pedido más tutorías y cuántas son.

```
SELECT nombre,apellido1,apellido2,COUNT(*) as Total_tutorias
FROM reserva_ti ri,usuarios u,alumnos a
WHERE a.dni=u.dni AND a.dni=dni_alumno GROUP BY dni_alumno
HAVING Total_tutorias>= ALL
(SELECT COUNT(id_ti) FROM reserva_ti GROUP BY dni_alumno);
```

```
SELECT nombre,apellido1,apellido2,COUNT(*) as Total_tutorias
FROM reserva_ti ri,usuarios u,alumnos a
WHERE a.dni=u.dni AND a.dni=dni_alumno GROUP BY dni_alumno
HAVING Total_tutorias=
(SELECT max(total) FROM
(SELECT COUNT(*) as total FROM reserva_ti
GROUP BY dni_alumno) N_tutorias);
```

nombre	apellido1	apellido2	Total_tutorias
RAMON	PASTOR	ESCUDERO	14

1 row in set (0.00 sec)

### 14. Crear una vista que muestre el dni del profesor y el código del módulo que incumplen la restricción de que un profesor no puede matricularse en módulos que se impartan en su departamento. (1 row)

```
CREATE VIEW prof_al_incumple AS
SELECT DISTINCT p.dni,m.cod_mod
FROM profesores p, profesores p2, alumnos a,matriculados m, imparten i
WHERE p.dni=a.dni AND p.dni=m.dni_alumno
AND p2.departamento=p.departamento AND p2.dni=i.dni_profesor
AND m.cod_mod=i.cod_mod;
```

```

+-----+-----+
| dni    | cod_mod |
+-----+-----+
| 31422905Y | CDI    |
+-----+-----+
1 row in set (0.00 sec)

```

- 15. Consulta para borrar la fila referida en la consulta anterior. Los datos del profesor y la columna deben introducirse directamente en la propia consulta no puede crearse una subconsulta de la vista anterior, puesto que las consultas DELETE no permiten incluir a la propia tabla en las condiciones del where. En caso de no haber resuelto el punto anterior simplemente elige un profesor+alumno y un módulo de los que está matriculado y borra esa fila.**

```

DELETE FROM matriculados
WHERE dni_alumno='31422905y' AND cod_mod='CDI';

```

- 16. Actualizar la capacidad máxima de todas las aulas bajándola en 2, excepto el laboratorio y el taller (A15 y B15)**

```

UPDATE aulas
SET capacidad_max=capacidad_max-2
WHERE cod_aula NOT IN ('A15','B15');

```

## PARTE II (2 puntos)

```

CREATE TABLE libros(
    isbn VARCHAR(13) PRIMARY KEY,
    nombre VARCHAR(100),
    autor VARCHAR(50),
    editorial VARCHAR(30)
);

```

-- Se podría hacer un check del ISBN con expresiones regulares usando REGEXP().

```

CREATE TABLE ejemplares(
    isbn VARCHAR(13),
    n_copia INT,
    estado VARCHAR(10),
    PRIMARY KEY (isbn,n_copia),
    FOREIGN KEY (isbn) REFERENCES libros(isbn),
    CHECK (estado IN ('Bueno','Usado','Muy usado'))
);

```

-- Se podría usar un tipo de variable más pequeño para n\_copia, pero así vale para cualquier SGBD. Se restringen los posibles estados a 3.



```
CREATE TABLE reserva_libros(  
    id_reserva INT PRIMARY KEY,  
    fecha_reserva DATE,  
    hora TIME,  
    fecha_recogida DATE,  
    dni varchar(9),  
    FOREIGN KEY (dni) REFERENCES usuarios(dni),  
    UNIQUE(fecha_reserva,hora,dni),  
    CHECK(fecha_recogida>=fecha_reserva)  
);
```

-- La fecha de recogida del libro nunca puede ser anterior a la reserva, obviamente.

```
CREATE TABLE lineas_reserva(  
    id_reserva INT,  
    n_linea INT,  
    fecha_dev DATE,  
    isbn VARCHAR(13),  
    n_copia INT,  
    PRIMARY KEY (id_reserva,n_linea),  
    FOREIGN KEY (id_reserva) REFERENCES reserva_libros(id_reserva),  
    FOREIGN KEY (isbn,n_copia) REFERENCES ejemplares(isbn,n_copia)  
);
```

-- La restricción para que no se pueda prestar un libro mientras ya está prestado no se puede hacer aquí, pues no permite consultas de la propia tabla en el check. Debe ser mediante un trigger. Se debería restringir que un libro no puede incluirse en una línea de reserva si aparece en otra (de cualquier reserva) con una fecha de devolución posterior a la fecha de recogida de la reserva actual.

-- Pasamos a insertar los datos de ejemplo.

```
INSERT INTO libros VALUES ('9788499641003', 'Administración de sistemas gestores de  
bases de datos', 'Hueso Ibañez, Luis', 'RAMA'),  
('9786073208178', 'Redes de computadoras', 'Andrew S. Tanenbaum', 'Pearson'),  
('8448125991', 'Desarrollo de Aplicaciones en entornos de 4.Generación y con  
Herramientas Case', 'A. Montero Ramos', 'Mcgraw-hill');
```

```
INSERT INTO ejemplares VALUES  
('9788499641003', 1, 'Muy Usado'), ('9788499641003', 2, 'Usado'),  
('9788499641003', 3, 'Bueno'), ('9788499641003', 4, 'Bueno'),  
('9786073208178', 1, 'Bueno'), ('9786073208178', 2, 'Usado'),  
('9786073208178', 3, 'Bueno'), ('8448125991', 1, 'Bueno'),  
('8448125991', 2, 'Bueno'), ('8448125991', 3, 'Muy usado');
```

```
INSERT INTO reserva_libros VALUES
(2456,'2021-02-17', '09:37', '2021-02-18', '39540403E'),
(2512,'2021-03-25', '18:42', '2021-03-29', '13891090X');
INSERT INTO lineas_reserva VALUES
(2456, 1, '2021-03-18', '9788499641003',2),
(2512, 1, '2021-04-29', '8448125991', 3),
(2512, 2, '2021-04-29', '9786073208178', 1),
(2512, 3, '2021-04-29', '9786073208178', 2);
```