



En esta sesión se continuará trabajando con la misma base de datos de la sesión anterior dedicada a “Funciones y Procedimientos”, y representada en la Fig. 1, así como con los ficheros “universidad\_estructura.sql” y “universidad\_datos.sql”.

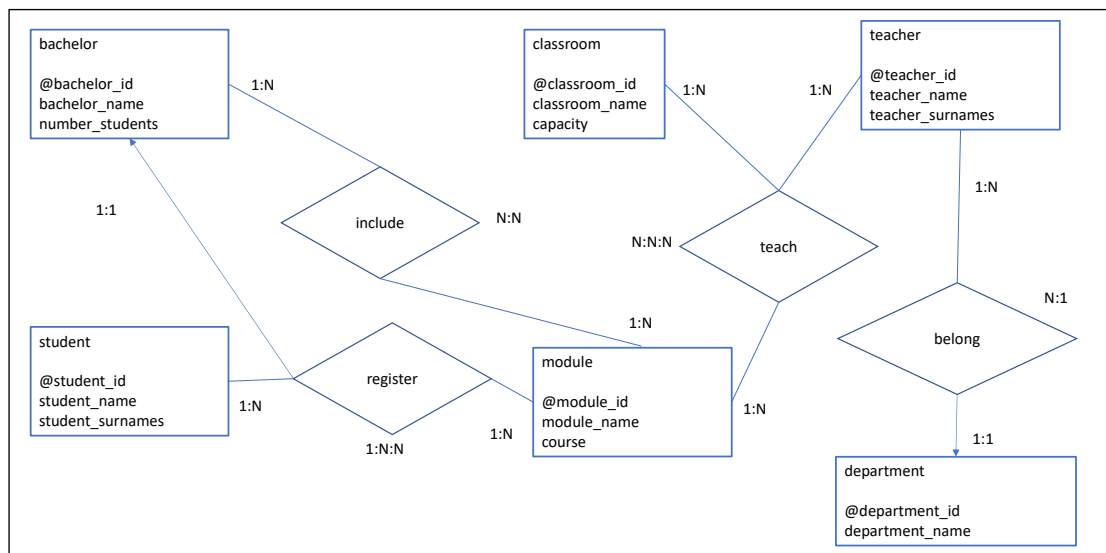


Figura 1. Modelo Entidad-Relación de la base de datos universidad.

El contenido se refleja en la Tabla 1.

Tabla 1. Datos de la base de datos universidad.

Tablas	Contenido
<b>grado</b>	1, Ingeniería Informática en Tecnologías de la Información, 115
	2, Ingeniería en Tecnologías y Servicios de Telecomunicación, 110
	3, Ingeniería Mecánica, 240
	4, Ingeniería en Tecnologías Industriales, 145
	5, Ingeniería Química Industrial, 60
	6, Ingeniería Electrónica Industrial y Automática, 110
	7, Ingeniería Eléctrica, 80
	8, Ingeniería Náutica y Transporte Marítimo, 35
	9, Ingeniería de Organización Industrial, 60
	10, Marina, 35
<b>modulo</b>	1, Cálculo, 1
	2, Estadística, 1
	3, Álgebra Lineal, 1
	4, Introducción a la Programación, 1
	5, Estructura de Datos, 2
	6, Ingeniería del Software, 3
	7, Sistemas de Información, 3
	8, Tecnologías Web, 4
	9, Algoritmia, 2
	10, Fundamentos de Informática 1



<p><b>profesor</b></p> <p>Atributos:</p> <ul style="list-style-type: none"> <li>• <i>profesor_id</i> (PK)</li> <li>• <i>departamento_id</i> (PK)</li> <li>• <i>profesor_nombre</i></li> <li>• <i>profesor_apellidos</i></li> </ul> <p>“profesor_id” y “departamento_id” forman la clave primaria (PK) en la tabla profesore, y además “departamento_id” es una clave ajena (FK) de la tabla departamento.</p>	<p>1, 1, Jorge Diez Pelaez</p> <p>2, 1, Pedro Hernandez Arauzo</p> <p>3, 1, Alfredo Santiago Alguero Garcia</p> <p>4, 1, Cristian Augusto Alonso</p> <p>5, 1, Pablo Suarez-Otero Gonzalez</p> <p>6, 1, Pablo Javier Tuya Gonzalez</p> <p>7, 2, Maria Zulima Fernandez Muñiz</p> <p>8, 2, Virginia Selgas Buznego</p> <p>9, 3, Hortensia Lopez Garcia</p> <p>10, 1, Maria Jose Suarez Cabal</p> <p>11, 1, Jose Garcia Fanjul</p> <p>12, 1, Maria Teresa Gonzalez Aparicio</p> <p>13, 2, Matias Pico Serrano</p>
<p><b>departamento</b></p> <p>Atributos:</p> <ul style="list-style-type: none"> <li>• <i>departamento_id</i> (PK)</li> <li>• <i>departamento_nombre</i></li> </ul>	<p>1, Ciencias de la Computacion</p> <p>2, Matematicas</p> <p>3, Estadistica</p> <p>4, Telecomunicacion</p>
<p><b>aula</b></p> <p>Atributos:</p> <ul style="list-style-type: none"> <li>• <i>aula_id</i> (PK)</li> <li>• <i>aula_nombre</i></li> <li>• <i>capacidad</i></li> </ul>	<p>1, AN-A, 60</p> <p>2, PL1, 20</p> <p>3, PL2, 20</p> <p>4, PL3, 30</p> <p>5, PL4, 30</p> <p>6, DO-9, 50</p> <p>7, DE-3, 15</p> <p>8, AN-E, 20</p> <p>9, AN-B, 60</p>
<p><b>estudiante</b></p> <p>Atributos:</p> <ul style="list-style-type: none"> <li>• <i>estudiante_id</i> (PK)</li> <li>• <i>estudiante_nombre</i></li> <li>• <i>estudiante_apellidos</i></li> <li>• <i>erasmus</i></li> </ul>	<p>1, 'Diego', 'Ramos Mendez', true</p> <p>2, 'Mercedes', 'Garcia Menendez', false</p> <p>3, 'Maria', 'Alvarez Gomez', true</p> <p>4, 'Sara', 'Prendes Pardo', true</p> <p>5, 'Juan', 'Prieto Vazquez', true</p> <p>6, 'Alberto', 'Palacio Gomez', false</p> <p>7, 'Pedro', 'Gancedo Alvarez', true</p>
<p><b>estudiante_grado_modulo</b></p> <p>Atributos:</p> <ul style="list-style-type: none"> <li>• <i>estudiante_id</i> (PK)</li> <li>• <i>grado_id</i> (PK)</li> <li>• <i>modulo_id</i> (PK)</li> </ul> <p>Los 3 forman la clave PK, siendo claves ajenas a sus respectivas tablas.</p>	<p>*Diego (1), Ingenieria Informatica en Tecnologias de la Informacion (1), Calculo (1)</p> <p>*Diego (1), Ingenieria Informatica en Tecnologias de la Informacion (1), Introduccion a la Programacion (4)</p> <p>*Diego(1), Ingenieria Informatica en Tecnologias de la Informacion (1), Ingenieria del Software (6)</p> <p>*Diego(1), Ingenieria Informatica en Tecnologias de la Informacion (1), Sistemas de Informacion (7)</p> <p>*Diego(1), Ingenieria Informatica en Tecnologias de la Informacion (1), Tecnologias Web (8)</p> <p>*Mercedes(2), Ingenieria Informatica en Tecnologias de la Informacion (1), Calculo (1)</p>



	<p>*Mercedes(2), Ingeniería Informática en Tecnologías de la Información (1), Ingeniería del Software (6)</p> <p>*Mercedes(2), Ingeniería Informática en Tecnologías de la Información (1), Tecnologías Web (8)</p> <p>*Mercedes(2), Ingeniería Informática en Tecnologías de la Información (1), Algoritmia (9)</p> <p>*Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Cálculo (1)</p> <p>*Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Estadística (2)</p> <p>*Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Ingeniería del Software (6)</p> <p>* Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Sistemas de Información (7)</p> <p>*Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Tecnologías Web (8)</p> <p>*Maria (3), Ingeniería Informática en Tecnologías de la Información (1), Algoritmia (9)</p> <p>* Sara (4), Ingeniería Informática en Tecnologías de la Información (1), Cálculo (1)</p> <p>* Sara (4) Ingeniería Informática en Tecnologías de la Información (1), Estructura de Datos (5)</p> <p>*Sara (4), Ingeniería Informática en Tecnologías de la Información (1), Ingeniería del Software (6)</p> <p>*Sara (4), Ingeniería Informática en Tecnologías de la Información (1), Algoritmia (9)</p> <p>*Juan (5), Ingeniería Química Industrial (5), Cálculo (1)</p> <p>*Juan (5), Ingeniería Química Industrial (5), Estadística (2)</p> <p>*Alberto (6), Ingeniería Química Industrial (5), Cálculo (1)</p> <p>*Alberto (6), Ingeniería Química Industrial (5), Estadística (2)</p> <p>*Alberto (6), Ingeniería Química Industrial (5), Álgebra Lineal (3)</p>
<p><b>grado_modulo</b></p> <p><b>Atributos:</b></p> <ul style="list-style-type: none"> <li>• bachelor_id (PK)</li> <li>• module_id (PK)</li> </ul> <p><b>Ambos forman la clave primaria en grado_modulo, pero además son claves ajenas en relación a las tablas grado y modulo respectivamente.</b></p>	<p>*Todos los grados tienen Cálculo, Álgebra Lineal y Estadística.</p> <p>*Todos los grados, excepto el grado Ingeniería Informática en Tecnologías de la Información, tienen Fundamentos de Informática</p>
<p><b>modulo_profesor_aula</b></p> <p><b>Atributos:</b></p> <ul style="list-style-type: none"> <li>• modulo_id (PK)</li> <li>• profesor_id (PK)</li> <li>• aula_id (PK)</li> </ul>	<p>Estadística (2), Hortensia (9), AN-A (1), DE-3 (7)</p> <p>Álgebra Lineal (3), Zulima (7), AN-A (1), DE-3 (7)</p> <p>Cálculo (1), Virginia (8), AN-A (1), DE-3 (7)</p> <p>Tecnologías Web (8), Maria Jose (10), AN-E (8), AN-A (1)</p> <p>Algoritmia (9), Maria Teresa (12), DE-3 (7), AN-A (1)</p> <p>Sistemas de Información (7), Pablo Javier (6), DO-9 (6), AN-A (1)</p> <p>Estructura de Datos (5), Pedro Hernandez (2), PL1 (2), PL2 (3), AN-A (1)</p>



Los tres atributos forman la clave primaria, y además son claves ajenas en las tablas modulo, profesor y aula respectivamente

Introduccion a la Programacion (4), Jorge Diez (1), AN-A (1), PL3(4)  
Ingenieria del Software (6), Jose Garcia Fanjul (11), DO-9 (6), AN-A (1)  
Fundamentos de Informatica (10), Alfredo (3), Pablo (5) y Cristian (4),  
PL1(2), PL2 (3), PL3(4), PL4 (5), AN-A (1)

El principal objetivo es trabajar con triggers. Alguna documentación se puede encontrar en el siguiente enlace:

<https://w3resource.com/PostgreSQL/postgresql-triggers.php#UFT>

En el fichero “Estudiante-Triggers.doc”, los estudiantes deberán de escribir la respuesta en PL/SQL a cada una de las cuestiones planteadas.

#### Cuestiones:

1. Crear un trigger que cada vez que se inserte una fila en la tabla ‘estudiante\_grado\_modulo’ se presente el mensaje ‘Se ha realizado una inserción’.

```
create or replace function presenta_insercion() returns trigger as
$$
begin
if tg_op = 'INSERT' then
    raise notice 'Se ha realizado una insercion';
    return new;
end if;
return null;
end;
$$ language plpgsql;

create trigger tg_presenta_insercion
after insert on estudiante_grado_modulo
for each row execute procedure presenta_insercion();
```

Probar el funcionamiento del trigger:

```
insert into estudiante_grado_modulo values(4, 1, 8);
```

```
NOTICE: Se ha realizado una insercion
INSERT 0 1
```

2. Crear un trigger llamado ‘presenta\_InsertDeleteUpdateOperacion’, donde cada vez que se realiza una operación de inserción (‘INSERT’), un borrado (‘DELETE’) o una actualización (‘UPDATE’) se presente un mensaje (**REALIZAR POR LOS ESTUDIANTES**).



**Nota:** Debes realizar una función similar a ‘presenta\_insercion’, pero añadiendo dos operaciones más. Para tal fin, usa la estructura de sentencia condicional `if .. elsif ..` (sintaxis en la transparencia 13 de la sesión de “Funciones y Procedimientos”). Además, cuando se crea el trigger, necesitarás cambiar ‘after insert’ por ‘after insert or update or delete’.

**Ejemplo.** Para comprobar el funcionamiento del trigger, ejecuta el siguiente conjunto de instrucciones SQL en el mismo orden que se indica.

- `insert into estudiante_grado_modulo values(3, 1, 5);`
  - `update estudiante_grado_modulo set modulo_id=3 where estudiante_id=3 and modulo_id=5;`
  - `delete from estudiante_grado_modulo where estudiante_id=3 and modulo_id=3;`
3. Añadir dos columnas a la table ‘estudiante\_grado\_modulo’ llamadas ‘nota’ (int) y ‘grado’ (varchar(10)), con un valor por defecto de 0 e ‘indefinido’ respectivamente. **(REALIZAR POR LOS ESTUDIANTES)**
4. Cada vez que se realiza una inserción, actualización o borrado en la tabla ‘estudiante\_grado\_modulo’ se necesita registrar que dicha operación ha sido ejecutada. Para tal fin, se creará una table denominada ‘operacion\_notas\_log’ que deberá contener los siguientes campos: **(REALIZAR POR LOS ESTUDIANTES)**.

Tabla: operacion_notas_log	
Atributo	Tipo
operacion	char(1)
hora	timestamp
estudiantetid	int
moduloid	int
nota	int

5. Crea un trigger para que cada vez que una nota es la tabla ‘estudiante\_grado\_modulo’, entonces se añada un registro en la tabla ‘operación\_notas\_log’ con la siguiente información: (‘U’, hora actual, estudiante id, modulo id, nota).

```
create or replace function operacion_log() returns trigger as
$$
begin
if tg_op = ‘UPDATE’ then
insert into operacion_notas_log(operacion, hora, estudianteid, moduloid,
nota) values (‘U’, now(), new.estudiante_id, new.modulo_id, new.nota);
return new;
```



```
end if;  
return null;  
end;  
$$ language plpgsql;
```

```
create trigger tg_operacion_log  
after update on estudiante_grado_modulo  
for each row execute procedure operacion_log();
```

Ejecutar el trigger realizamos una operación de actualización:

```
update estudiante_grado_modulo set nota = 5 where estudiante_id=1 and  
modulo_id=4;
```

```
select * from operacion_notas_log;
```

operacion	hora	estudianteid	moduloid	nota
U	2022-03-27 00:19:31.217417	1	4	5
(1 fila)				

5.1. Realizar otras modificaciones y comprueba los cambios en la tabla 'operacion\_notas\_log'. **(REALIZAR POR LOS ESTUDIANTES)**

6. Borrar la función anterior 'operacion\_notas\_log' (apartado 5) de la siguiente manera:

```
drop function operacion_notas_log cascade;
```

**Nota:** Se necesita escribir la palabra reservada 'cascade' con el fin de borrar también el trigger 'tg\_operacion\_log' vinculado a la función.

En el caso de que se quisiese borrar solo el trigger, entonces se debería de aplicar la siguiente sintaxis:

```
Drop trigger [if exists] trigger_name on table_name  
[ CASCADE | RESTRICT ]
```

Por lo tanto, en nuestro ejemplo sería:

```
drop trigger tg_operacion_log on estudiante_grado_modulo;
```



7. Crear un trigger similar al anterior pero que también registre las operaciones de inserción ('I') y borrado ('D'). Usa la estructura del condicional anidada. Escribe al menos un ejemplo para comprobar el funcionamiento de cada uno de los dos tipos de operaciones añadidas. (**REALIZAR POR LOS ESTUDIANTES**)
8. Borrar la tabla 'operacion\_notas\_log'. También será necesario borrar el trigger tg\_operacion\_log que insertaba en la tabla borrada. (**REALIZAR POR LOS ESTUDIANTES**)
9. Crear una tabla denominada 'operacion\_calificacion\_log' similar a la del apartado 4, pero añadiendo el atributo 'calificacion'. (**REALIZAR POR LOS ESTUDIANTES**)

Tabla: operacion_calificacion_log	
Atributo	Tipo
operacion	char(1)
stamp	timestamp
estudianteid	int
moduloid	int
nota	int
calificacion	varchar(10)

10. Crear un trigger denominado 'calificacion\_log' similar al del punto 5, pero antes de que la nota sea insertada o actualizada en la tabla 'estudiante\_grado\_modulo' se calculará una calificación no numérica de acuerdo al baremo indicado en la siguiente tabla:

nota	calificación
< 4	'Pobre'
4 <= x < 5	'No buena'
5 <= x < 7	'Buena'
7 <= x < 9	'Muy buena'
9 <= x <= 10	'Excelente'

Cada operación debe ser registrada en el tabla 'operacion\_calificacion\_log'. Escribe un ejemplo para algunos de ellos, y comprueba que las tablas se actualizan correctamente. (**REALIZAR POR LOS ESTUDIANTES**)

Al finalizar la sesión, renombra el fichero Estudiante-Triggers.doc con tu nombre y apellidos (nombre\_apellido1\_apellido2.doc), y súbelo al Campus Virtual.



UNIVERSIDAD DE OVIEDO  
ESCUELA POLITÉCNICA DE  
INGENIERIA DE GIJÓN

BASES DE DATOS (PostgreSQL)  
TRIGGERS  
Marzo 2022

**Objetivos de aprendizaje:**

- ✓ Crear triggers