Unidad 5
Data Query Language (II)

DQL

Contenido:

- Consultas multitabla
- Subconsultas
- Vistas

Consultas multitabla

- La potencia de las bases de datos relacionales radica en que los datos se encuentran almacenados en tablas que se **relacionan entre sí a través de las claves foráneas**.
- Para definir una consulta sobre varias tablas, se indican en la cláusula FROM todas las tablas implicadas, separadas por coma ‘,’
- En este tipo de consultas se hace necesario **distinguir las columnas de cada tabla**, ya que es habitual que distintas tablas llamen a los campos relacionados de la misma manera, o que simplemente distintas tablas tengan campos con el mismo nombre .

- Se utiliza el nombre de la tabla para referenciar un campo concreto: **nombretabla.campo**
  - Ejemplos: *asignatura.idCiclo*
    ciclo.idCiclo
    alumno.nombre
    asignatura.nombre

Consultas multitabla

- Podemos realizar las consultas sobre varias tablas utilizando:
  - El operador de conjuntos **UNION**
    - UNION : excluye registros duplicados
    - UNION ALL : incluye registros duplicados
  - El operador **JOIN**, con tablas relacionadas
    - {LEFT |INNER | RIGHT} JOIN
    - ON

Consultas multitabla

- **UNION / UNION ALL**:
  - Al utilizar UNION se unifican (concatenan) los resultados obtenidos en consultas independientes.
  - Dichas consultas pueden ser sobre una o varias tablas.
  - Se debe garantizar que:
    - El número y orden de las columnas en todas las consultas es el mismo
    - Los tipos de datos con compatibles

```
select campo1, campo2 from tabla1
UNION
select campo3, campo4 from tabla2;
```

```
select 'ASIG', nombre from asignatura where codciclo = 1
UNION
select 'CICLO', nombre from cicloform where codciclo = 1;
```

Consultas multitabla

- Cuando se quiera obtener datos de varias tablas relacionadas, será necesario incluirlas en la cláusula FROM.
- Cuando se indican dos tablas en el FROM, se obtiene el **producto cartesiano** de los registros de la primera tabla con la segunda.
- El producto cartesiano es el resultado de combinar cada registro de la primera tabla con cada uno de los registros de la segunda (todas las combinaciones posibles)

Consultas multitabla

a) Dado el siguiente esquema:

ASIGNATURA

P· CodAsig	Nombre	NH	B	F· CodCF
1	Bases de datos	165	S	1
2	Lenguaje de marcas	120	N	1
3	Seguridad informática	90	S	2
4	Despliegue aplicaciones webs	110	N	1
5	Fundamentos de hardware	90	N	2
6	Acceso a datos	180	N	1

CICLO

P· CodCF	Nombre	Siglas
1	Desarrollo de aplicaciones webs	DAW
2	Administración de sistemas informáticos en red	ASIR
3	Desarrollo de aplicaciones multiplataforma	DAM

b) SELECT * FROM asignatura, ciclo;

c) Salida:

ASIGNATURA x CICLO

P· CodAsig	Nombre	NH	B	F· CodCF	P· CodCF	Nombre	Siglas
1	Bases de datos	165	S	1	1	Desarrollo de aplicaciones webs	DAW
1	Bases de datos	165	S	1	2	Administración de sistemas informáticos en red	ASIR
1	Bases de datos	165	S	1	3	Desarrollo de aplicaciones multiplataforma	DAM
2	Lenguaje de marcas	120	N	1	1	Desarrollo de aplicaciones webs	DAW
2	Lenguaje de marcas	120	N	1	2	Administración de sistemas informáticos en red	ASIR
2	Lenguaje de marcas	120	N	1	3	Desarrollo de aplicaciones multiplataforma	DAM
3	Seguridad informática	90	S	2	1	Desarrollo de aplicaciones webs	DAW
3	Seguridad informática	90	S	2	2	Administración de sistemas informáticos en red	ASIR
3	Seguridad informática	90	S	2	3	Desarrollo de aplicaciones multiplataforma	DAM
4	Despliegue aplicaciones webs	110	N	1	1	Desarrollo de aplicaciones webs	DAW
4	Despliegue aplicaciones webs	110	N	1	2	Administración de sistemas informáticos en red	ASIR
4	Despliegue aplicaciones webs	110	N	1	3	Desarrollo de aplicaciones multiplataforma	DAM
5	Fundamentos de hardware	90	N	2	1	Desarrollo de aplicaciones webs	DAW
5	Fundamentos de hardware	90	N	2	2	Administración de sistemas informáticos en red	ASIR
5	Fundamentos de hardware	90	N	2	3	Desarrollo de aplicaciones multiplataforma	DAM
6	Acceso a datos	180	N	1	1	Desarrollo de aplicaciones webs	DAW
6	Acceso a datos	180	N	1	2	Administración de sistemas informáticos en red	ASIR
6	Acceso a datos	180	N	1	3	Desarrollo de aplicaciones multiplataforma	DAM

Consultas multitabla

- Para filtrar el producto cartesiano y obtener únicamente los registros de la primera tabla, que están relacionados con la segunda, es necesario utilizar (comparar) las claves foráneas vs primarias

```
SELECT *
  FROM asignatura, ciclo
WHERE asignatura.codCF = ciclo.codCF
```

ASIGNATURA x CICLO

P· CodAsig	Nombre	NH	B	F· *CodCF*	P· CodCF	Nombre	Siglas
1	Bases de datos	165	S	1	1	Desarrollo de aplicaciones webs	DAW
2	Lenguaje de marcas	120	N	1	1	Desarrollo de aplicaciones webs	DAW
3	Seguridad informática	90	S	2	2	Administración de sistemas informáticos en red	ASIR
4	Despliegue aplicaciones webs	110	N	1	1	Desarrollo de aplicaciones webs	DAW
5	Fundamentos de hardware	90	N	2	2	Administración de sistemas informáticos en red	ASIR
6	Acceso a datos	180	N	1	1	Desarrollo de aplicaciones webs	DAW

Consultas multitabla

- En la consulta anterior se ha aplicado un filtro sobre el producto cartesiano, es decir, se establece una condición sobre la relación: "*asignatura.codCF = ciclo.codCF*".
- También se puede aplicar como hasta ahora, un filtro a los registros seleccionados, por ejemplo, "*ciclo.siglas = 'DAM'* ", este filtro nada tiene que ver con la relación entre ambas tablas.

```
SELECT *
  FROM asignatura, ciclo
WHERE ciclo.siglas = 'DAM'
    AND asignatura.codCF = ciclo.codCF
```

- El operador **JOIN** surge para **separar las condiciones** que aplican a la relación de las condiciones que se aplican a la selección de registros.

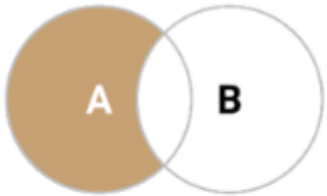
Consultas multitabla

- **JOIN**: Operador para la combinación de los registros de dos o más tablas a partir de la relación establecida mediante claves foráneas.
  - **ON**: cláusula que introduce las condiciones de la relación, aquellas que filtran el producto cartesiano resultado de la asociación de las tablas.

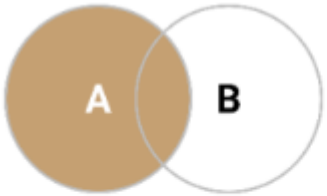
```
SELECT *
  FROM asignatura
  INNER JOIN  ciclo ON asignatura.codCF = ciclo.codCF
WHERE ciclo.siglas = 'DAM';
```

Consultas multitabla

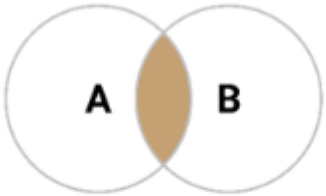
- **Tipos de JOIN**.



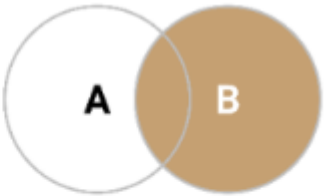

```
SELECT <select list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key=B.Key
WHERE B.Key IS NULL
```



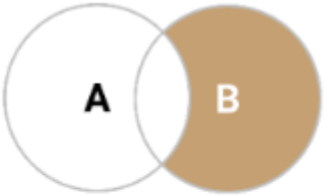

```
SELECT <select list>
FROM TableA
LEFT JOIN TableB
ON A.Key=B.Key
```




```
SELECT <select list>
FROM TableA A
INNER JOIN TableB B
ON A.Key=B.Key
```



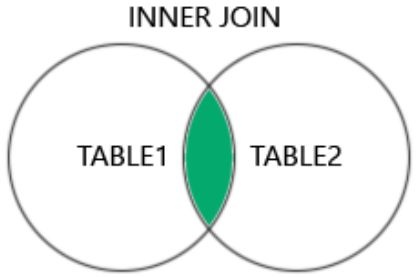

```
SELECT <select list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key=B.Key
```




```
SELECT <select list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key=B.Key
WHERE A.Key IS NULL
```

Consultas multitabla

- **INNER JOIN**: Selecciona los **registros que tienen el mismo valor** en las columnas que las relacionan.



INNER JOIN de 2 TABLAS

```
SELECT columnas
  FROM tabla1
 INNER JOIN tabla2
    ON tabla1.campo = tabla2.campo;
```

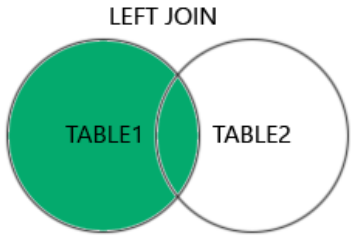
INNER JOIN de 3 TABLAS

```
SELECT columnas
  FROM tabla1
 INNER JOIN tabla2 ON tabla1.campo1 = tabla2.campo2
 INNER JOIN tabla3 ON tabla2.campo3 = tabla3.campo4;
```

- En la cláusula ON se cruzan las tablas por el campo que las relaciona.
- En el WHERE se aplican los filtros sobre el resto de columnas.

Consultas multitabla

- **LEFT JOIN**: Devuelve todos los registros de la tabla de la izquierda y los registros de la tabla de la derecha relacionados por las claves foráneas.

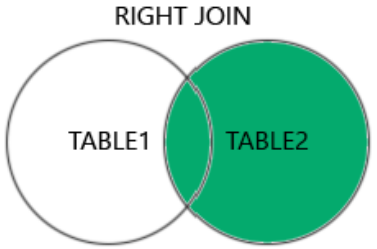



```
SELECT Cliente.Nombre, Pedido.IDPedido
FROM Cliente
LEFT JOIN Peiddo ON  Cliente.IdCliente = Pedido.IdCliente
ORDER BY  Cliente.Nombre;
```

- La consulta del ejemplo devuelve el nombre de todos los clientes, tengan o no pedidos relacionados. Para aquellos clientes que sí tienen un pedido, se muestra el IdPedido, para los que no, la 2ª columna se devuelve con un NULL

Consultas multitabla

- **RIGHT JOIN**: Devuelve todos los registros de la tabla de la derecha y los registros de la tabla de la izquierda relacionados por las claves foráneas.




```
SELECT Pedido.IdPedido, Empleado.Nombre
FROM  Pedido
RIGHT JOIN Empleado ON  Pedido.IdEmpleado = Empleado.IdEmpleado
ORDER BY  Pedido.IdPedido;
```

- La consulta del ejemplo devuelve el nombre de todos los empleados, tengan o no pedidos relacionados. Para aquellos empleados que sí tienen un pedido, se muestra el IdPedido, para los que no, la 1ª columna se devuelve con un NULL

Consultas multitabla

- **Filtros en left/right join.** Se pueden aplicar filtros tanto en la cláusula **ON** como en el **WHERE**:
  - Para filtrar los registros de la tabla unida sin afectar al número de registros devueltos de la tabla principal, se añaden las condiciones en el ON.

```
SELECT Pedido.IdPedido, Empleado.Nombre
FROM  Pedido
LEFT JOIN Empleado ON  Pedido.IdEmpleado = Empleado.IdEmpleado
                   AND Empleado.ciudad = 'Madrid'
ORDER BY  Pedido.IdPedido;
```

En el ejemplo, se devuelven todos los registros de Pedido, incluso si no hay coincidencias en Empleado con el filtro de ciudad = Madrid.

Consultas multitabla

- **Filtros en left/right join** (continuación)
  - Los filtros que se aplican en el where afectan también a las filas de la tabla principal.

```
SELECT Pedido.IdPedido, Empleado.Nombre
FROM  Pedido
LEFT JOIN Empleado ON  Pedido.IdEmpleado = Empleado.IdEmpleado
                   AND Empleado.ciudad = 'Madrid'
WHERE Empleado.Nombre is null
ORDER BY Pedido.IdPedido;
```

En el ejemplo, se filtran también los registros de Pedido, y se muestran únicamente aquellos que no tienen coincidencia con la tabla secundaria (también filtrada)

Subconsultas

- Las subconsultas son una técnica aplicada para mejorar el rendimiento en la ejecución de las consultas:
  - la lista de valores que devuelve una consulta se convierte en valores de entrada para otra consulta de un nivel superior.
  - Por este motivo se dice que las consultas están 'anidadas'
- Se consigue minimizar el coste de proceso del producto cartesiano que se realiza al unir varias tablas.
- Podemos clasificar las subconsultas en 2 tipos:
  - Escalonadas
  - De lista

Subconsultas

- Las **subconsultas escalonadas** son aquellas que **devuelven un único valor** que se va a utilizar para aplicar un filtro en la cláusula WHERE y/o HAVING de la consulta principal:

- Ejemplo, obtener los datos del empleado/s que tiene/n el mayor sueldo:

```
SELECT *
  FROM Empleado
WHERE empleado.sueldo = (SELECT max(empleado.sueldo)
                           FROM empleado);
```

Subconsultas

- **Subconsultas de lista**, en este caso la subconsulta **devuelve más de un registro** y se utilizan los operadores ALL, ANY, EXISTS, IN para conectar la consulta principal con la secundaria
  - **ALL y ANY** : permiten comparar un único valor de una columna con un rango de otros valores.
    - ANY devuelve verdadero si alguno de los valores de la subconsulta cumple la condición.
    - ALL devuelve verdadero si todos los valores de la subconsulta cumplen la condición. Se utiliza en las cláusulas select, where y having.
  - **IN** : permite especificar varios valores en la cláusula where. Es una forma abreviada para cuando se necesita utilizar varias condiciones OR.

Subconsultas

- **EXISTS**: se utiliza para analizar la existencia de algún registro en la subconsulta. Devuelve TRUE si la subconsulta devuelve uno o más registros.

- Ejemplo, ¿hay un conserje que tenga el suelo mayor al de **todos** los directores?:

```
SELECT *
  FROM empleado
 WHERE empleado.oficio = 'conserje'
   AND empleado.sueldo > ALL (SELECT sueldo
                                FROM empleado
                                WHERE oficio = 'director');
```

Subconsultas

- Ejemplo, ¿hay un conserje que tenga el suelo mayor que **alguno** de los directores?:

```
SELECT *
  FROM empleado
 WHERE empleado.oficio = 'conserje'
   AND empleado.sueldo > ANY (SELECT sueldo
                              FROM empleado
                              WHERE oficio = 'director');
```

Subconsultas

- Ejemplo, ¿hay algún hotel en el que todas las habitaciones tengan más de 1 cama?:

```
SELECT *
  FROM hotel h
 WHERE NOT EXISTS (SELECT * FROM habitación hab
                   WHERE hab.idhotel = h.idhotel
                     AND hab.camas < 2);
```

Se establece relación entre la consulta principal y la subconsulta:'hab.idhotel = h.idhotel'

Nota: que todas las habitaciones tengan más de 1 cama, es equivalente a decir 'no existe habitación con menos de 2 camas'.

Subconsultas

- Ejemplo, ¿qué empleados trabajan en los hoteles de Sueca?:

```
SELECT *
  FROM empleado
 WHERE idhotel in (SELECT idhotel FROM hotel
                   WHERE poblacion = 'Sueca');
```