

Base de Datos

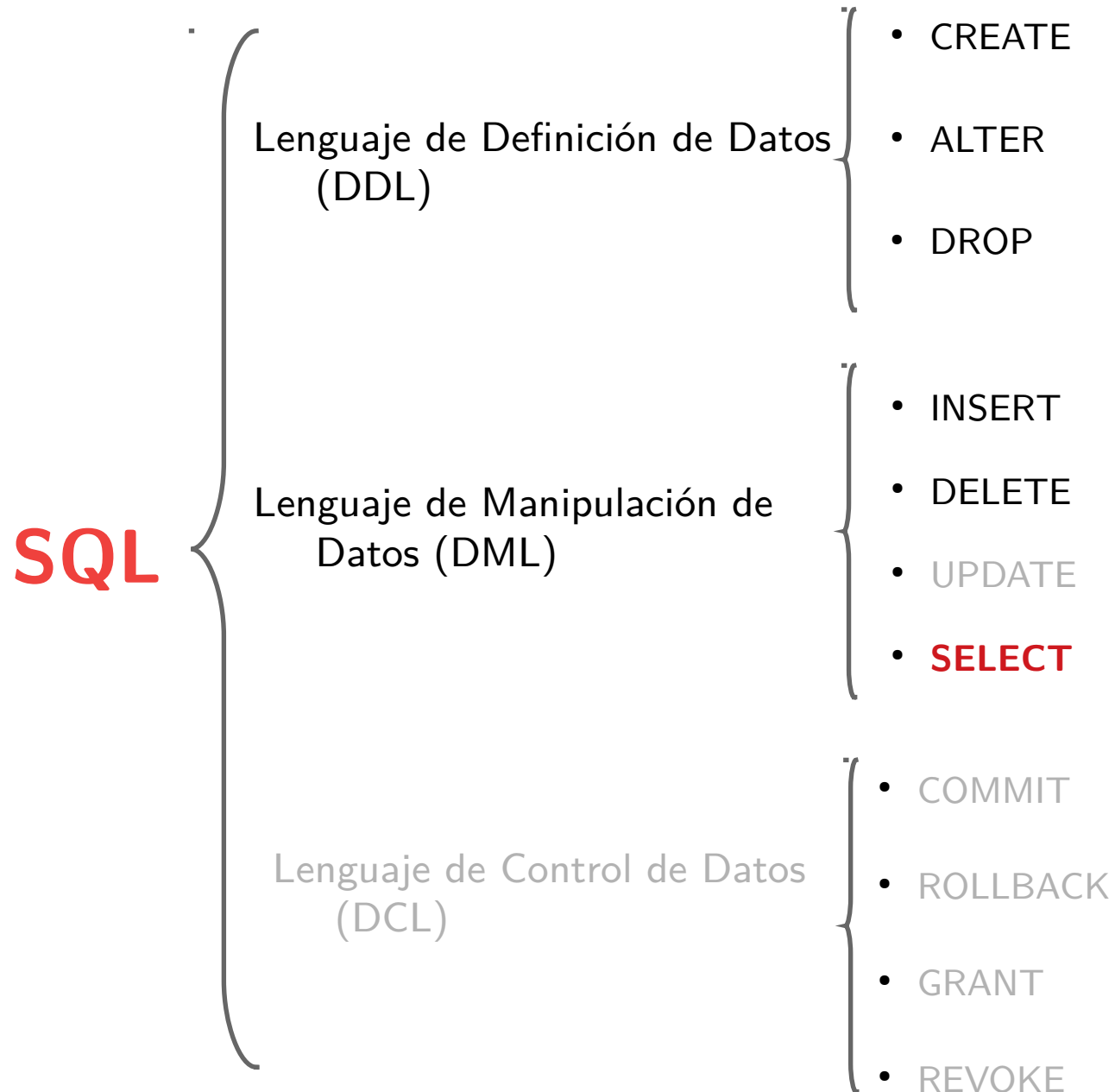
Structured Query Language (SQL)



Planificación

MES	SEMANA del CUATR.	LUNES	JUEVES
O C T U B R E	Semana 9 del 31/9 al 4/10	Práctico 6: Pasaje al MR	Práctico 6: Pasaje al MR
	Semana 10 del 7 al 11	Teoría: SQL	Práctico 7: SQL
	Semana 11 del 14 al 18	Teoría: DF y FN Práctico 8: DF y FN	Práctico 7: SQL
	Semana 12 del 21 al 25	Práctico 8: DF y FN	Consulta para el Parcial. Práctico de Promoción: Inicia
	Semana 13 del 28 al 31	SEGUNDO PARCIAL	Resolución y muestra de parcial 2 y Consultas para recuperaciones
N O V.	Semana 14 del 4 al 8	RECUPERACIÓN 1 del PARCIAL 1	RECUPERACIÓN 1 del PARCIAL 2
	Semana 15 del 28 al 31	RECUPERACIÓN 2 Práctico de Promoción: Entrega	

Introducción



Introducción

■ Sintaxis

SELECT <lista de atributos a seleccionar>

FROM < lista de relaciones(tablas) >

[WHERE < condición> **]**

[GROUP BY < atributo de agrupación> **]**

[HAVING < condición de agrupación> **]**

[ORDER BY < lista de atributos> **]**



Operador INNER JOIN



Ensamblajes: Repaso

- Si en una consulta interviene mas de una relación, hay que poner la condición de ensamble para que la consulta tenga una semántica acorde a la realidad sobre la que se trabaja.
- Si no se pone la condición bajo la cual ensamblar nuplas \rightarrow producto cartesiano.

Ensamblas: Repaso

Docentes

D-Cod	D-Nbre	D-Dir	D-CantHijos
10	Ana	d1	0
20	Juan	d2	2
30	Cecilia	d3	1

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2

SELECT *

FROM Docentes, Tomo

D-Cod	D-Nbre	D-Dir	D-CantHijos	D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	Ana	d1	0	10	1000	f1
10	Ana	d1	0	30	2000	f2
20	Juan	d2	2	10	1000	f1
20	Juan	d2	2	30	2000	f2
30	Cecilia	d3	1	10	2000	f1
30	Cecilia	d3	1	30	2000	f2

Select Multitabla

Docentes

D-Cod	D-Nbre	D-Dir	D-CantHijos
10	Ana	d1	0
20	Juan	d2	2
30	Cecilia	d3	1

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2

SELECT *

FROM Docentes, Tomo

D-Cod	D-Nbre	D-Dir	D-CantHijos	D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	Ana	d1	0	10	1000	f1
10	Ana	d1	0	30	2000	f2
20	Juan	d2	2	10	1000	f1
20	Juan	d2	2	30	2000	f2
30	Cecilia	d3	1	10	1000	f1
30	Cecilia	d3	1	30	2000	f2

Ensamblas: Repaso

Docentes

D-Cod	D-Nbre	D-Dir	D-CantHijos
10	Ana	d1	0
20	Juan	d2	2
30	Cecilia	d3	1

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2

SELECT *

FROM Docentes, Tomo

WHERE D-Cod = D-Cod-Tomo

*Condición de
ensamble*

D-Cod	D-Nbre	D-Dir	D-CantHijos	D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	Ana	d1	0	10	1000	f1
30	Cecilia	d3	1	30	2000	f2

Operador INNER JOIN

```
SELECT *  
FROM Docentes, Tomo  
WHERE D-Cod =D-CodTomo
```

```
SELECT *  
FROM Docentes INNER JOIN Tomo ON D-Cod=D-Cod-Tomo
```

- Los dos SELECT resuelven la misma consulta.
- El uso de INNER JOIN no tiene efecto en la eficiencia de la consulta → el SGBD ejecutará a las dos de la misma manera (plan de ejecución).
- El uso de INNER JOIN aporta quizás más claridad sobre la semántica de la consulta que se está resolviendo.

Operador INNER JOIN

Docentes

D-Cod	D-Nbre	D-Dir	D-CantHijos
10	Ana	d1	0
20	Juan	d2	2
30	Cecilia	d3	1
40	Pedro	d4	0

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2
20	1000	f1
30	3000	f3

Consulta :

```
SELECT D-Cod- D-Nbre, C-Cod-Tomo
FROM Docentes, Tomo
WHERE D-Cod=D-Cod-Tomo AND D-Cod =10
```

*Condición de
ensamble*

D-Cod	D-Nbre	C-Cod-Tomo
10	Ana	1000

¿Semántica?

Operador INNER JOIN

```
SELECT    D-Cod- D-Nbre, C-Cod-Tomo
FROM      Docentes, Tomo
WHERE      D-Cod=D-Cod-Tomo AND D-Cod =10
```

```
SELECT    D-Cod- D-Nbre, C-Cod-Tomo
FROM      Docentes INNER JOIN Tomo ON D-Cod=D-Cod-Tomo
WHERE      D-Cod =10
```



Consultas sumarias: GROUP BY y HAVING



Consultas Sumarias

- Ya hemos visto como resolver consultas del tipo: cantidad de docentes que tomaron el curso de código 1000.
- A veces es necesario mostrar esta información pero no para un curso en particular, sino para todos los cursos: “obtener la cantidad de docentes que tomó cada uno de los cursos”.
- Estas consultas se conocen con el nombre de consultas sumarias e implica obtener un **resumen** de los datos que tenemos en la instancia por ejemplo:
 - Dar la cantidad de docentes que tomaron cada curso.
 - Dar la cantidad de cursos que tomó cada docente.

Consultas Sumarias

- Dar la cantidad de docentes que tomaron cada curso.
- Dar la cantidad de cursos que tomó cada docente

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2
20	1000	f1
10	3000	f3
30	3000	f3

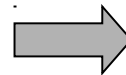
- Primero identificamos sobre qué atributo se quiere resumir la información para luego agrupar las nuplas de la instancia por los valores de ese atributo:
 - Dar la cantidad de docentes que tomaron cada curso → C-Cod-Tomo
 - Dar la cantidad de cursos que tomó cada docente → D-Cod-Tomo

Consultas Sumarias: GROUP BY

```
SELECT C-Cod-Tomo , COUNT (D-Cod-Tomo)
FROM Tomo
GROUP BY (C-Cod-Tomo)
```

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
d1	c1	f1
d2	c3	f2
d1	c2	f3
d3	c1	f3
d3	c3	f2
d4	c3	f4

GB



C-Cod-Tomo	D-Cod-Tomo	Fecha-Tomo
c1	d1 d3	f1 f3
c3	d2 d3 d4	f2 f2 f4
c2	d1	f3

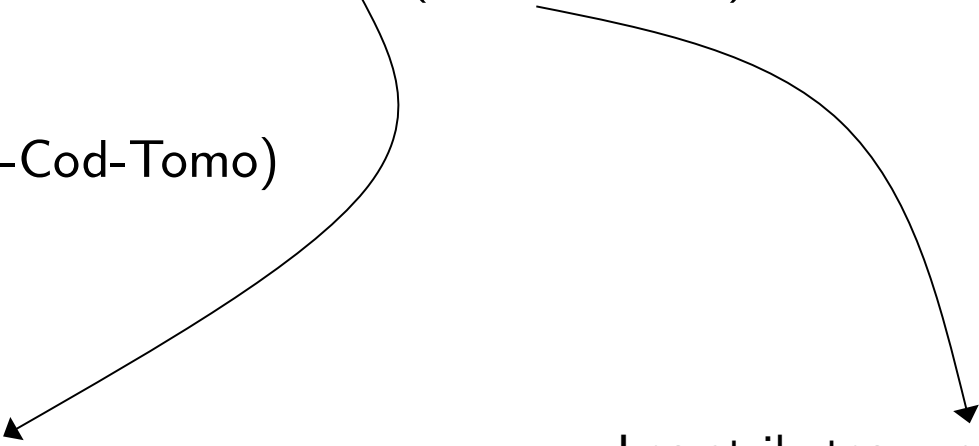
Resultado:

C-Cod-Tomo	COUNT (D-Cod-Tomo)
c1	2
c3	3
c2	1

Consultas Sumarias: GROUP BY

Importante

```
SELECT C-Cod-Tomo , COUNT (D-Cod-Tomo)
FROM Tomo
GROUP BY (C-Cod-Tomo)
```



Cuando usamos *group by*, las funciones agregadas se aplican únicamente dentro de los grupos

Los atributos que no se mencionan en *group by* **DEBEN** aparecer con una función agregada.
¿Por qué?

Consultas Sumarias: GROUP BY

- Dar la cantidad de cursos que tomó cada docente

Tomo

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
30	2000	f2
20	1000	f1
10	3000	f3
30	3000	f3

```
SELECT  D-Cod-Tomo , COUNT (C-Cod-Tomo)
FROM    Tomo
GROUP BY (D-Cod-Tomo)
```

Consultas Sumarias: GROUP BY

- GROUP BY agrupa nuplas con idéntico valor en los campos especificados.
- Podemos especificar más de un campo de agrupación

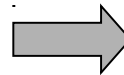
```
SELECT C-Cod-Tomo, Fecha, COUNT (D-Cod-Tomo),  
FROM Tomo  
GROUP BY C-Cod-Tomo, Fecha
```

Consultas Sumarias: GROUP BY

```
SELECT C-Cod-Tomo, Fecha, COUNT (D-Cod-Tomo)
FROM Tomo
GROUP BY C-Cod-Tomo, Fecha
```

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
d1	c1	f1
d2	c3	f2
d1	c2	f3
d3	c1	f3
d3	c3	f2
d4	c1	f1
d1	c3	f2

GB



C-Cod-Tomo	Fecha-Tomo	D-Cod-Tomo
c1	f1	d1 d4
c3	f2	d2 d3 d1
c2	f3	d1
c1	f3	d3

- ¿Resultado?
- ¿Semántica?

Consultas Sumarias: GROUP BY

- Un ejemplo más complicado:

Obtener la cantidad de cursos realizados por cada docente. En ese listado también debe aparecer el nombre y código de cada docente. El listado debe ordenarse por nombre del docente.

Docentes = { D-Cod, D-DNI, D-NbreApellido, D-Domicilio }

Cursos = { C-Cod, C-Nombre, C-Duración, D-Cod-Supervisa }

Tomo = { D-Cod-Tomo, C-Cod-Tomo, Fecha-Tomó }

```
SELECT  D-Cod-Tomo, COUNT (C-Cod-Tomo), D-NbreApellido
FROM    Tomo , Docentes
WHERE   D-Cod-Tomo = D-Cod
GROUP BY D-Cod-Tomo , D-NbreApellido
ORDER BY D-NbreApellido
```

Consultas Sumarias: GROUP BY

```
SELECT D-Cod-Tomo, COUNT (C-Cod-Tomo), D-NbreApellido  
FROM Tomo , Docentes  
WHERE D-Cod-Tomo = D-Cod  
GROUP BY D-Cod-Tomo , D-NbreApellido  
ORDER BY D-NbreApellido
```

```
SELECT D-Cod-Tomo, COUNT(C-Cod-Tomo), D-NbreApellidoa  
FROM Tomás INNER JOIN Docentes ON D-Cod-Tomo = D-Cod  
GROUP BY D-Cod-Tomo, D-NbreApellido  
ORDER BY D-NombreApellido
```

Consultas Sumarias: GROUP BY

Docentes

D-Cod	D-Nbre	D-Dir	D-CantHijos
10	Ana	d1	0
20	Juan	d2	2
30	Cecilia	d3	1
40	Pedro	d4	0

D-Cod-Tomo	C-Cod-Tomo	Fecha-Tomo
10	1000	f1
10	2000	f2
20	1000	f1
10	3000	f3

```
SELECT D-Cod-Tomo, COUNT(C-Cod-Tomo), D-NbreApellido
FROM Tomó INNER JOIN Docentes ON D-Cod-Tomo = D-Cod
GROUP BY D-Cod-Tomo, D-NbreApellido
ORDER BY D-NombreApellido
```

Consultas Sumarias: GROUP BY + HAVING

- La cláusula HAVING puede usarse para seleccionar y rechazar grupos.

- Ejemplo:

Obtener la cantidad de cursos realizados por cada docente, pero sólo para aquellos docentes que hayan realizado al menos dos cursos.

```
SELECT COUNT(C-Cod-Tomo), D-Cod-Tomo  
FROM Tomo  
GROUP BY D-Cod-Tomo  
HAVING COUNT (C-Cod-Tomo) >= 2
```