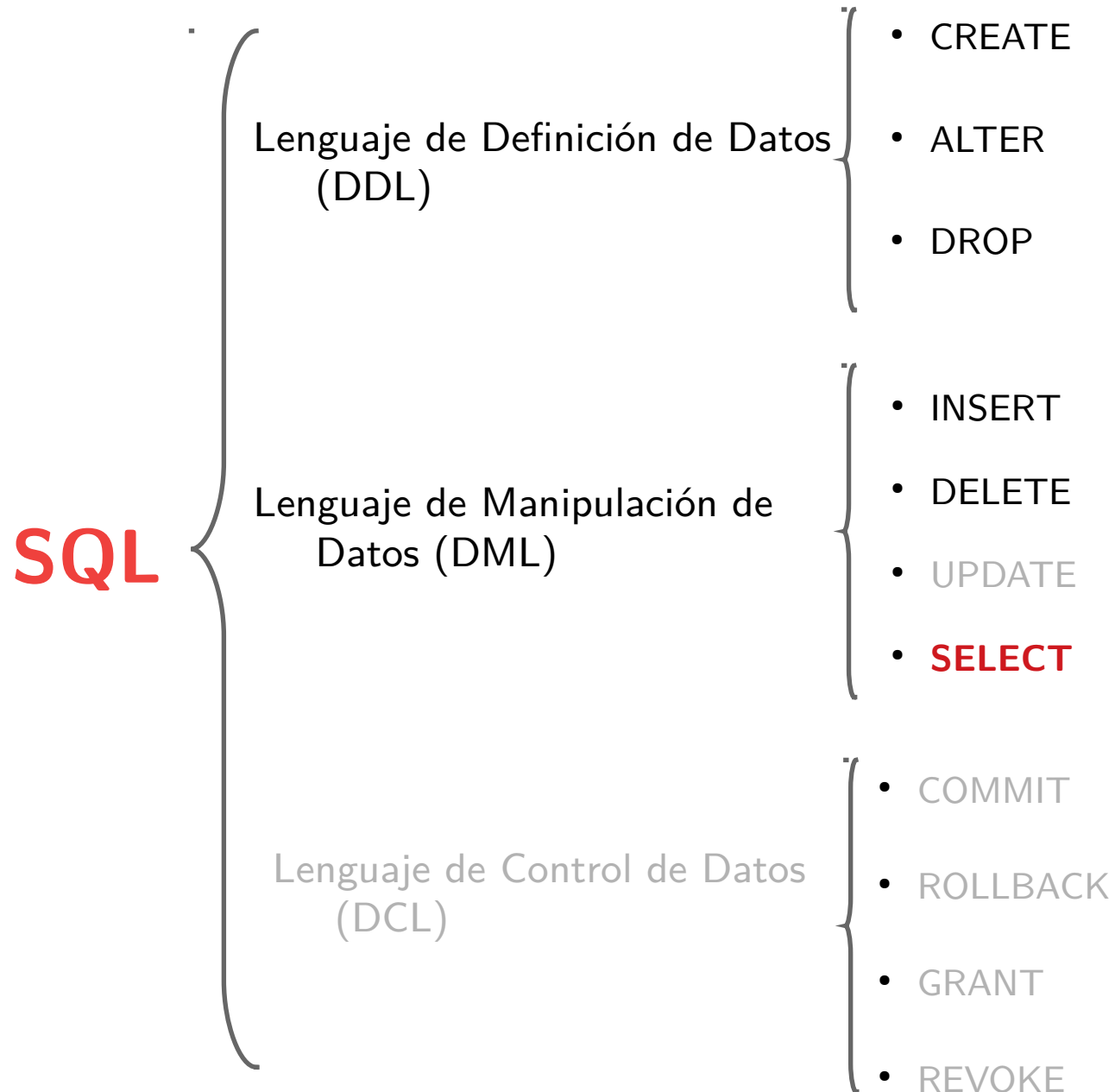


# Bases de Datos

## Structured Query Language (SQL)

Ciclo 2024

# Introducción





# Uso de Alias



# SELECT: Asignación de Alias

- También es posible definir **ALIAS** para nombres de relaciones y/o atributos usando la palabra clave **AS**

```
SELECT  C-Cod-Tomo D-Cod-Tomo AS D
FROM    Tomo
WHERE   D=10 OR D=30
```

- Utilidad:
  - Economizar escritura.
  - Mostrar resultados de manera mas amigable al usuario
- Cuando veamos ensambles, veremos la utilidad de usar alias para nombres de relaciones.

# SELECT: Asignación de Alias



- Los alias no modifican el esquema → los atributos en el bases de datos conservan su nombre.
- En práctico verán cómo modificar los nombres de atributos en el esquema.



# Comparación de Subcadenas



# SELECT: Comparación de Subcadenas

- SQL permite especificar condiciones de comparación de partes de una cadena (subcadenas) empleando el operador LIKE.
- Las cadenas parciales se especifican por medio de **dos caracteres reservados**:
  - %** se sustituye por un número arbitrario de caracteres
  - \_** se sustituye por un único carácter

# SELECT: Comparación de Subcadenas

## Cursos:

C-Cod	C-Nbre	C-Duracion	D-Cod-Supervisa
1000	Introducción a Lógica	40	10
2000	Introducción a Minería de Datos	30	10
3000	Minería de Datos	60	10
4000	Base de Datos	50	10
5000	Lógica de Primer Orden	50	10
6000	Estructuras de Datos I	50	10

## Consulta:

**SELECT** C-Cod, C-Nbre  
**FROM** Cursos  
**WHERE** C-Nbre **LIKE** '%Minería%'

## Consulta:

**SELECT** C-Cod, C-Nbre  
**FROM** Cursos  
**WHERE** C-Nbre **LIKE** '% Datos'



# SELECT: Comparación de Subcadenas

## Cursos:

C-Cod	C-Nbre	C-Duracion	D-Cod-Supervisa
1000	Introducción a Lógica	40	10
2000	Introducción a Minería de Datos	30	10
3000	Minería de Datos	60	10
4000	Base de Datos	50	10
5000	Lógica de Primer Orden	50	10
6000	Estructuras de Datos I	50	10

## Consulta:

```
SELECT C-Cod, C-Nbre
FROM Cursos
WHERE C-Nbre LIKE 'Lógica%'
```

## Consulta:

```
SELECT C-Cod, C-Nbre
FROM Cursos
WHERE C-Nbre LIKE 'Lógica _ _ %'
```



# Consultas Multitablas



# 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> **]**

# Introducción

- Vimos que la consulta más sencilla que podemos tener es:

**SELECT** <lista de atributos>

**FROM** < lista de relaciones(tablas) >

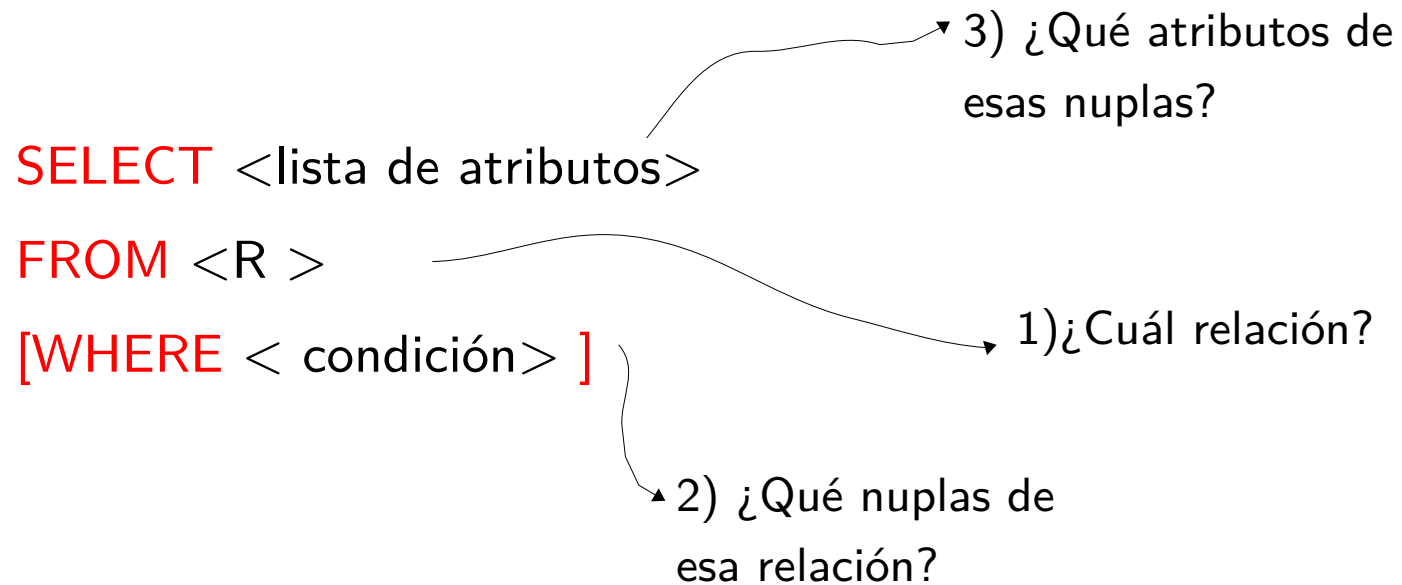
- Estudiamos consultas sobre una única relación(tabla) de la base de datos:

**SELECT** <lista de atributos>

**FROM** R

# Introducción

- Analizamos el efecto de la cláusula WHERE:



En el resultado sólo intervienen las nuplas de R que hacen verdadera la condición especificada en el WHERE.

# Select Multitabla

- Volvamos a la consulta mas sencilla que podemos tener:

**SELECT** <lista de atributos>

**FROM** <lista de relaciones>

- Cuando en un SELECT interviene mas de una relación

**SELECT** <lista de atributos>

**FROM**  $R_1 R_2 \dots R_n$

El efecto es realizar el producto cartesiano de las relaciones involucradas.

# 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	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** D-Nbre, C-CodTomo, Fecha-Tomo  
**FROM** Docentes, Tomo

D-Nbre	C-Cod-Tomo	Fecha-Tomo
Ana	1000	f1
Ana	2000	f2
Juan	1000	f1
Juan	2000	f2
Cecilia	1000	f1
Cecilia	2000	f2



# Select Multitabla

```
SELECT *
```

```
FROM Docentes, Tomo
```

```
SELECT *
```

```
FROM Docentes, Cursos, Tomo
```

- ¿Semántica?

# Select Multitabla

- El producto cartesiano carece de semántica: solo junta nuplas de distintas relaciones.
- Pero es el operador base a partir del cual se pueden realizar consultas que tengan una semántica acorde a la realidad que el MR representa.
- En el ejemplo anterior:

*Obtener información sobre los cursos que ha tomado cada docente.*

# 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

# Select Multitabla: Ensamble (Join)

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

# Select Multitabla: Ensamble (Join)

**SELECT** <lista de atributos>

**FROM** <R >

**[WHERE** < condición> **]**

3) ¿Qué atributos de esas nuplas?

1) ¿Cuál relación?

2) ¿Qué nuplas de esa relación?

**SELECT** <lista de atributos>

**FROM** <R1, R2, ..., Rn>

**[WHERE** < condición> **]**

3) ¿Qué atributos de esas nuplas?

1) Producto cartesiano de las n relaciones

2) ¿Qué nuplas de ese producto cartesiano?

# Select Multitabla: Ensamble (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
```

D-Cod	D-Nbre	C-Cod-Tomo
10	Ana	1000
20	Juan	2000
30	Cecilia	2000
30	Cecilia	3000

¿Semántica?

# Select Multitabla: Ensamble (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?

# Select Multitabla: Ensamble (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 =30
```

D-Cod	D-Nbre	C-Cod-Tomo
30	Cecilia	2000
30	Cecilia	3000



# Select Multitabla: Ensemble (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 =40
```

D-Cod	D-Nbre	C-Cod-Tomo

# Select Multitabla: Ensamble (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

Cursos

C-Cod	C-Nombre	C-Duracion	D-Cod-Sup
1000	Lógica	50	10
2000	Programación	40	20
3000	Álgebra	90	30
4000	Cálculo	60	20

Tomo

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

- Obtener información sobre los cursos que han realizado los docentes. En ese listado deberá figurar nombre y código del docente junto con nombre y código del curso que ha realizado.

# Select Multitabla: Ensamble (Join)

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

*Condición de  
ensamble entre  
Docentes y Tomo*

*Condición de  
ensamble entre  
Cursos y Tomo*

- **IMPORTANTE:** no estamos indicando el orden en que debe resolverse esta consulta. Eso lo decide el SGBD buscando la forma de resolución que sea mas eficiente (mas adelante veremos esto nuevamente)

# Select Multitabla: Ensamble (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

Cursos

C-Cod	C-Nombre	C-Duracion	D-Cod-Sup
1000	Lógica	50	10
2000	Programación	40	20
3000	Álgebra	90	30
4000	Cálculo	60	20

Tomo

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

- Obtener nombre y dirección de los docentes que han realizado el curso 'Lógica'

# Select Multitabla: Ensamble (Join)

```
SELECT  D-Cod- D-Nbre, C-Cod, C-Nbre
FROM    Docentes, Cursos, Tomo
WHERE   D-Cod=D-Cod-Tomo AND C-Cod =C-Cod-Tomo AND C-Nombre='Lógica'
```

*Condición de  
ensamble entre  
Docentes y Tomo*

*Condición de  
ensamble entre  
Cursos y Tomo*

- Cuando en un SELECT interviene más de una relación la condición estará formada por:
  - Condiciones de ensamble de las relaciones involucradas.
  - Condiciones propias de la consulta a resolver.

# Nombres de Atributos Ambiguos

- En SQL se puede usar el mismo nombre para dos atributos de distintas relaciones.
- Cuando una consulta hace referencia a dos atributos distintos pero con el mismo nombre, se deben calificar los mismos anteponiendo el nombre de la relación:

NombreRelación.NombreAtributo

# Nombres de Atributos Ambiguos

```
CREATE TABLE Docentes (  
    D-Cod INT NOT NULL,  
    Nombre VARCHAR (50) NOT NULL,  
    D-Domicilio VARCHAR (60) NOT NULL,  
    PRIMARY KEY (DCod )  
);
```

```
CREATE TABLE Cursos (  
    C-Cod INT NOT NULL,  
    Nombre VARCHAR (50) NOT NULL,  
    D-Duracion INT NOT NULL,  
    PRIMARY KEY (CCod)  
);
```

```
CREATE TABLE Tomo (  
    D-Cod INT NOT NULL,  
    C-Cod-Tomo INT NOT NULL,  
    Fecha-Tomo DATE NOT NULL,  
    PRIMARY KEY (D-Cod, C-Cod-Tomo )  
);
```

- Si Docentes y Cursos intervienen en una misma consulta → conflicto con Nombre
- Si Docentes y Tomo intervienen en la misma consulta → conflicto con D-Cod

