

CFGSS

***Desarrollo de Aplicaciones
Multiplataforma/Web
Administración de Sistemas
Informáticos en Red***

MÓDULOS

Bases de Datos

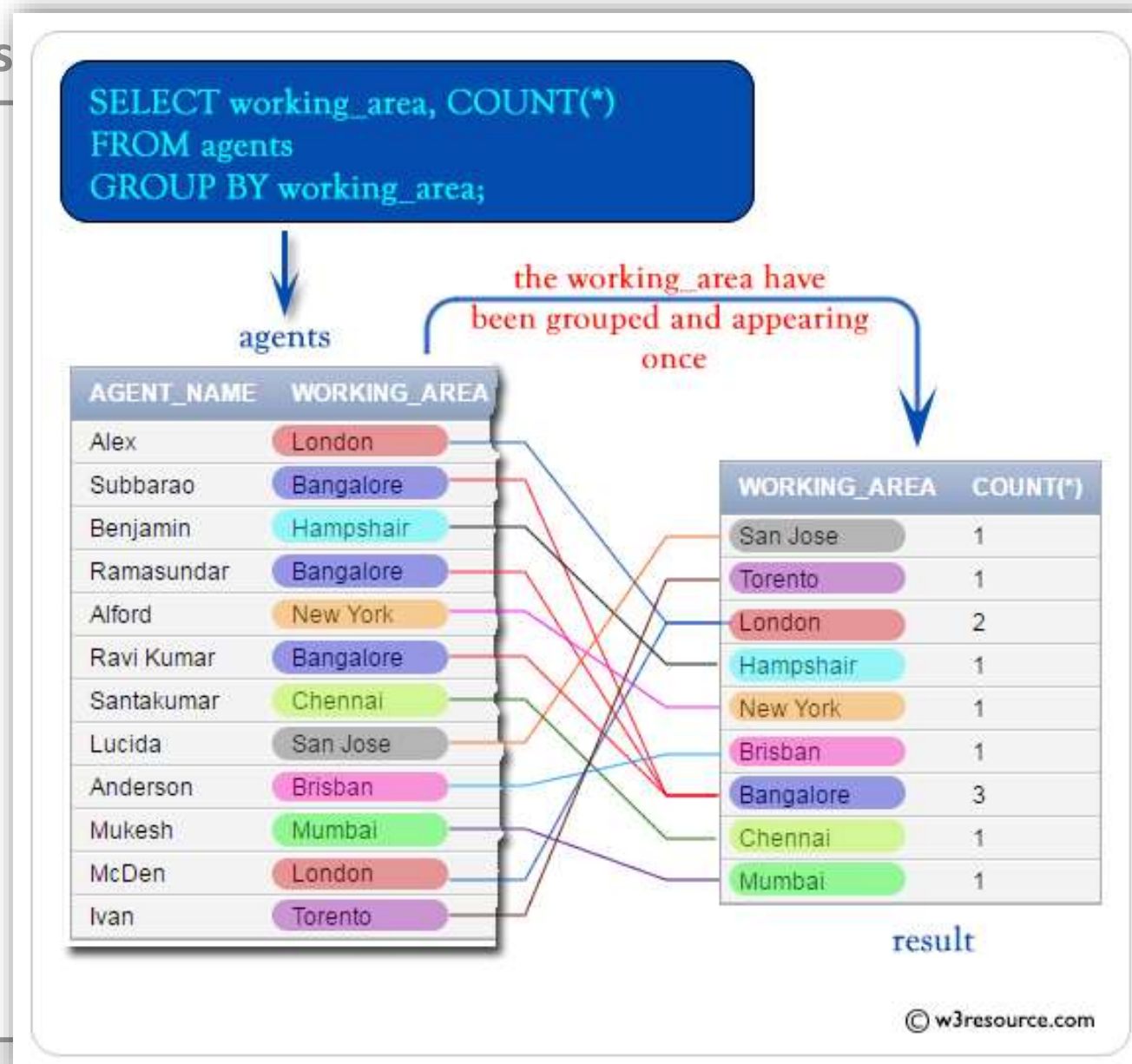
Gestión de Bases de Datos

RESUMEN GROUP BY, JOIN y subconsultas

Índice

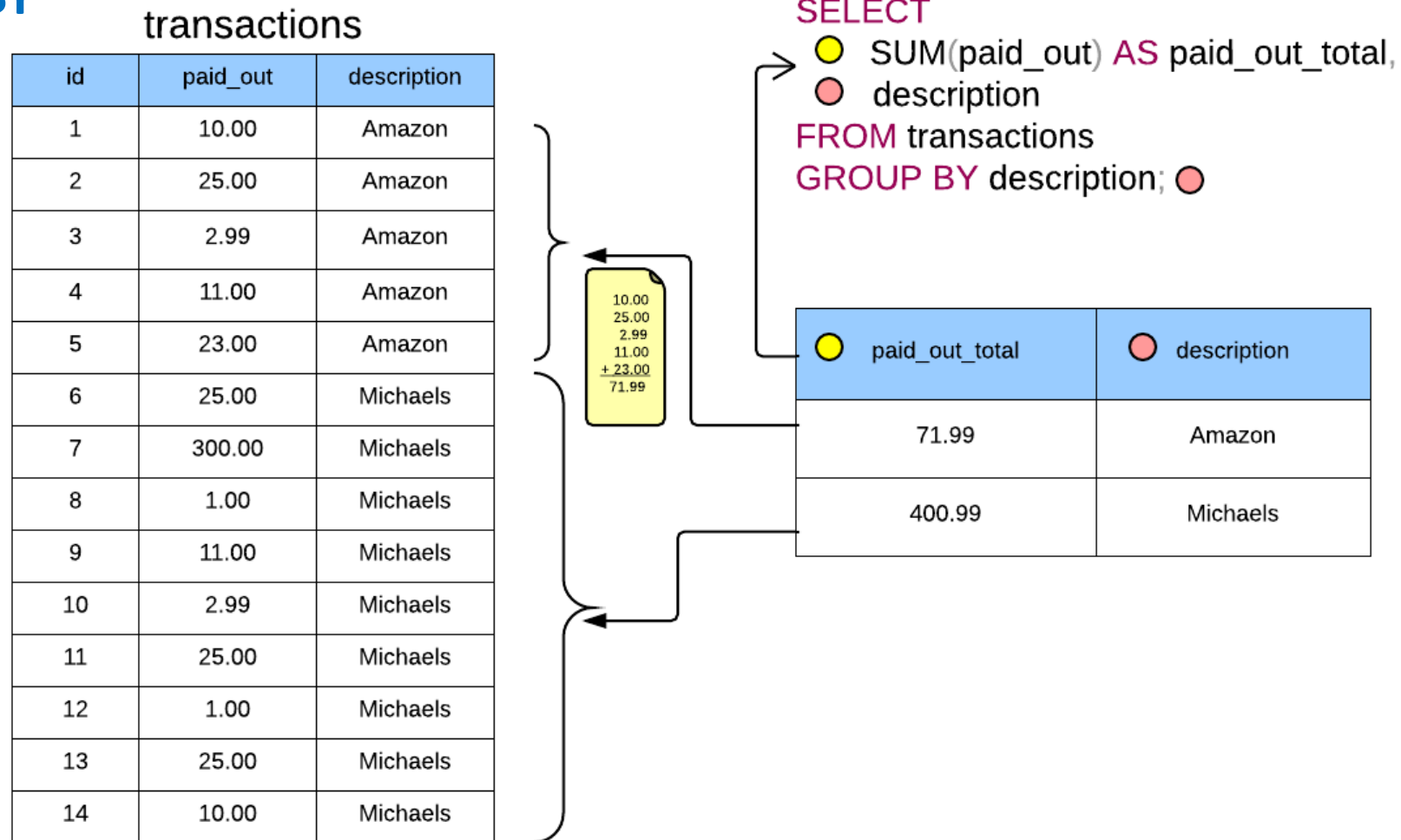
1. Agrupaciones
2. Tipos de combinaciones de tablas
3. Subconsultas
4. Orden de ejecución de cláusulas

1. Agrupaciones: GROUP BY



Resumen GROUP BY, JOIN y subconsultas

1. Agrupaciones: GROUP BY



1. Agrupaciones: GROUP BY + HAVING

```
SELECT commission, COUNT (*)  
FROM agents  
GROUP BY commission  
HAVING COUNT ( * ) > 3;
```

agents

AGENT_NAME	COMMISSION
Alex	.13
Subbarao	.14
Benjamin	.11
Ramasundar	.15
Alford	.12
Ravi Kumar	.15
Santakumar	.14
Lucida	.12
Anderson	.13
Mukesh	.11
McDen	.15
Ivan	.15

GROUP BY commission

COMMISSION	COUNT(*)
.15	4
.11	2
.14	2
.13	2
.12	2

HAVING COUNT (*) > 3;

COMMISSION	COUNT(*)
.15	4

COMMISSION	COUNT(*)
.15	4
.11	2
.14	2
.13	2
.12	2

1. Agrupaciones: GROUP BY + HAVING

Employee

EmployeeID	Ename	DeptID	Salary
1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID;

GROUP BY
Employee Table
using DeptID

DeptID	AVG(Salary)
1	3000.00
2	4000.00
3	4250.00

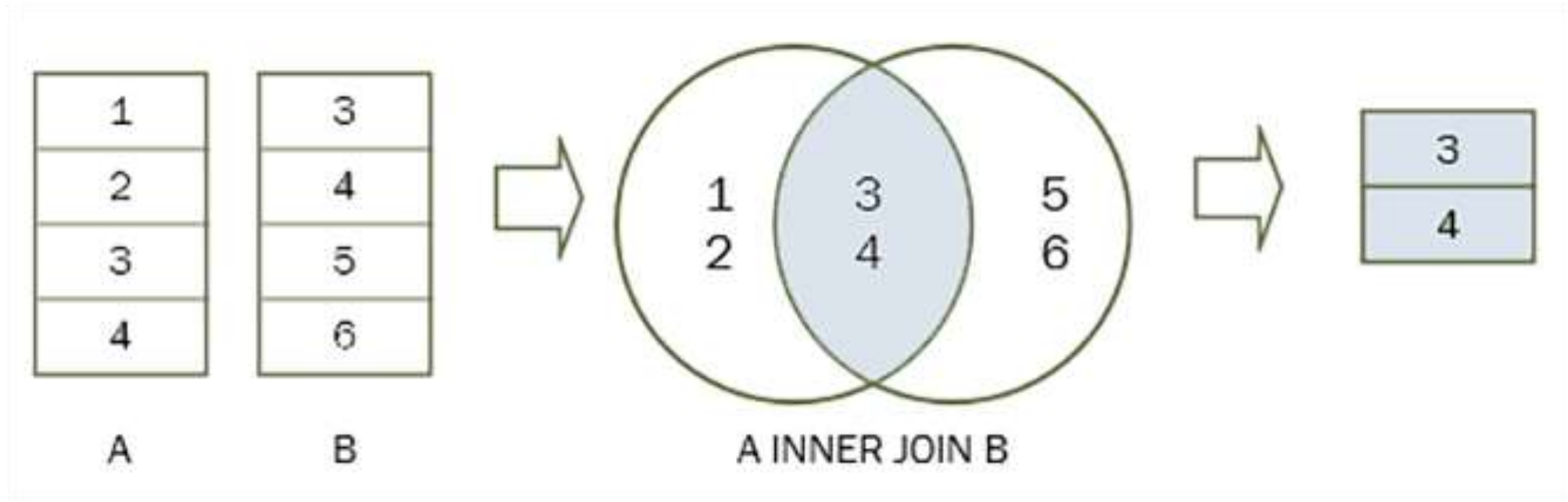
SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID
HAVING AVG(Salary) > 3000;

HAVING

DeptID	AVG(Salary)
2	4000.00
3	4250.00

2. Tipos de combinaciones de tablas: INNER JOIN o combinación interna

Un **INNER JOIN** te entrega la intersección de ambas tablas, es decir, las filas que ambas tablas tienen en común.



2. Tipos de combinaciones de tablas: INNER JOIN o combinación interna

Tabla: Artículos

Id	Título	Fecha	IdAutor
1	El SQL es adorable	02/06/2002	1
2	El psicoanálisis del JOIN	01/04/2002	2
3	Fenomenología del GROUP BY	25/04/2002	2
4	Teoría de conjuntos	14/05/2002	1
5	Lógica difusa para torpes	26/07/2003	NULL

Tabla: Autores

Id	Nombre
1	Horario Gómez
2	Hortensia Ramírez
3	Otilio Marchanete

SELECT titulo, fecha, nombre
FROM articulos INNER JOIN autores
ON articulos.IdAutor = autores.id;

Título	Fecha	Nombre
El SQL es adorable	02/06/2002	Horario Gómez
El psicoanálisis del JOIN	01/04/2002	Hortensia Ramírez
Fenomenología del GROUP BY	25/04/2002	Hortensia Ramírez
Teoría de conjuntos	14/05/2002	Horario Gómez

2. Tipos de combinaciones de tablas: INNER JOIN o combinación interna

EMPLEADOS

DNI	NOMBRE	LOCALIDAD	COD_SUCURSAL
11111111A	ANA	ALMERÍA	0001
22222222B	BERNARDO	GRANADA	0001
33333333C	CARLOS	GRANADA	NULL
44444444D	DAVID	JEREZ	0003

SUCURSALES

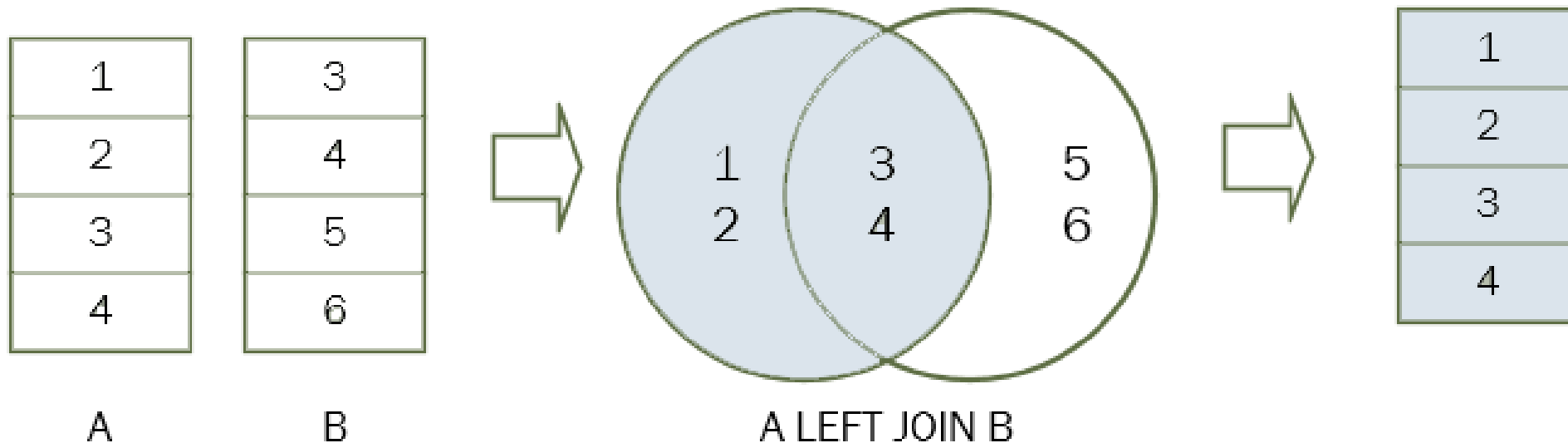
COD_SUCURSAL	DIRECCIÓN	LOCALIDAD
0001	C/ ANCHA, 1	ALMERÍA
0002	C/ NUEVA, 1	GRANADA
0003	C/ CORTÉS, 33	CÁDIZ

```
SELECT EMPLEADOS.*, SUCURSALES.LOCALIDAD
FROM EMPLEADOS JOIN SUCURSALES
ON EMPLEADOS.COD_SUCURSAL = SUCURSALES.COD_SUCURSAL;
```

DNI	NOMBRE	LOCALIDAD	COD_SUCURSAL	LOCALIDAD
22222222B	BERNARDO	GRANADA	0001	ALMERÍA
11111111A	ANA	ALMERÍA	0001	ALMERÍA
44444444D	DAVID	JEREZ	0003	CÁDIZ

2. Tipos de combinaciones de tablas: LEFT JOIN o combinación externa izquierda

Un **LEFT JOIN**, te dará todas las filas de A, incluyendo las filas comunes entre A y B.



2. Tipos de combinaciones de tablas: LEFT JOIN o combinación externa izquierda

Tabla: Artículos

Id	Título	Fecha	IdAutor
1	El SQL es adorable	02/06/2002	1
2	El psicoanálisis del JOIN	01/04/2002	2
3	Fenomenología del GROUP BY	25/04/2002	2
4	Teoría de conjuntos	14/05/2002	1
5	Lógica difusa para torpes	26/07/2003	NULL

Tabla: Autores

Id	Nombre
1	Horario Gómez
2	Hortensia Ramírez
3	Otilio Marchanete

SELECT titulo, fecha, nombre
FROM articulos **LEFT JOIN** autores
ON articulos.IdAutor = autores.id;

Título	Fecha	Nombre
El SQL es adorable	02/06/2002	Horario Gómez
El psicoanálisis del JOIN	01/04/2002	Hortensia Ramírez
Fenomenología del GROUP BY	25/04/2002	Hortensia Ramírez
Teoría de conjuntos	14/05/2002	Horario Gómez
Lógica difusa para torpes	26/07/2003	

2. Tipos de combinaciones de tablas: LEFT JOIN o combinación externa izquierda

EMPLEADOS

DNI	NOMBRE	LOCALIDAD	COD_SUCURSAL
11111111A	ANA	ALMERÍA	0001
22222222B	BERNARDO	GRANADA	0001
33333333C	CARLOS	GRANADA	NULL
44444444D	DAVID	JEREZ	0003

SUCURSALES

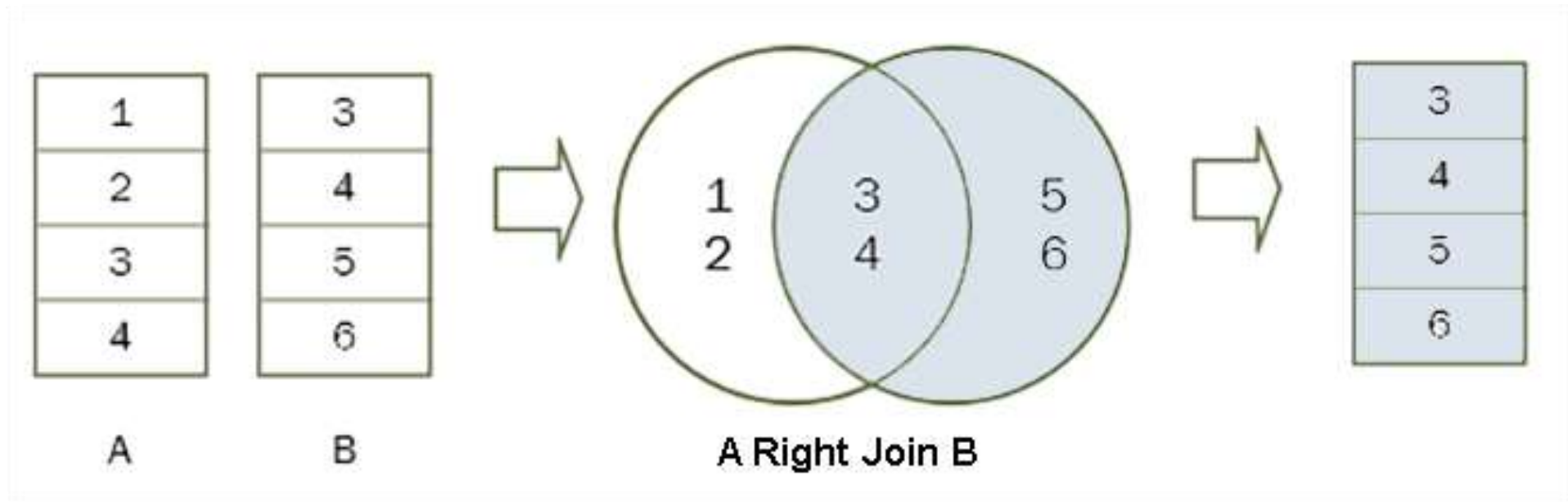
COD_SUCURSAL	DIRECCIÓN	LOCALIDAD
0001	C/ ANCHA, 1	ALMERÍA
0002	C/ NUEVA, 1	GRANADA
0003	C/ CORTÉS, 33	CÁDIZ

```
SELECT E.*, S.LOCALIDAD
FROM EMPLEADOS E
LEFT JOIN SUCURSALES S
ON
EMPLEADOS.COD_SUCURSAL =
SUCURSALES.COD_SUCURSAL;
```

DNI	NOMBRE	LOCALIDAD	<u>COD_SUCURSAL</u>	LOCALIDAD
11111111A	ANA	ALMERÍA	0001	ALMERÍA
22222222B	BERNARDO	GRANADA	0001	ALMERÍA
33333333C	CARLOS	GRANADA	<u>NULL</u>	<u>NULL</u>
44444444D	DAVID	JEREZ	0003	CÁDIZ

2. Tipos de combinaciones de tablas: **RIGHT JOIN** o combinación externa derecha

Un **RIGHT JOIN**, te dará todas las filas de B, incluyendo las filas comunes entre A y B.



2. Tipos de combinaciones de tablas: RIGHT JOIN o combinación externa derecha

Tabla: Artículos

Id	Título	Fecha	IdAutor
1	El SQL es adorable	02/06/2002	1
2	El psicoanálisis del JOIN	01/04/2002	2
3	Fenomenología del GROUP BY	25/04/2002	2
4	Teoría de conjuntos	14/05/2002	1
5	Lógica difusa para torpes	26/07/2003	NULL

Tabla: Autores

Id	Nombre
1	Horario Gómez
2	Hortensia Ramírez
3	Otilio Marchanete

SELECT titulo, fecha, nombre
FROM articulos RIGHT JOIN autores
ON articulos.IdAutor = autores.id;

Título	Fecha	Nombre
El SQL es adorable	02/06/2002	Horario Gómez
Teoría de conjuntos	14/05/2002	Horario Gómez
El psicoanálisis del JOIN	01/04/2002	Hortensia Ramírez
Teoría de conjuntos	14/05/2002	Hortensia Ramírez
		Otilio Marchanete

2. Tipos de combinaciones de tablas: RIGHT JOIN o combinación externa derecha

EMPLEADOS

DNI	NOMBRE	LOCALIDAD	COD_SUCURSAL
11111111A	ANA	ALMERÍA	0001
22222222B	BERNARDO	GRANADA	0001
33333333C	CARLOS	GRANADA	NULL
44444444D	DAVID	JEREZ	0003

SUCURSALES

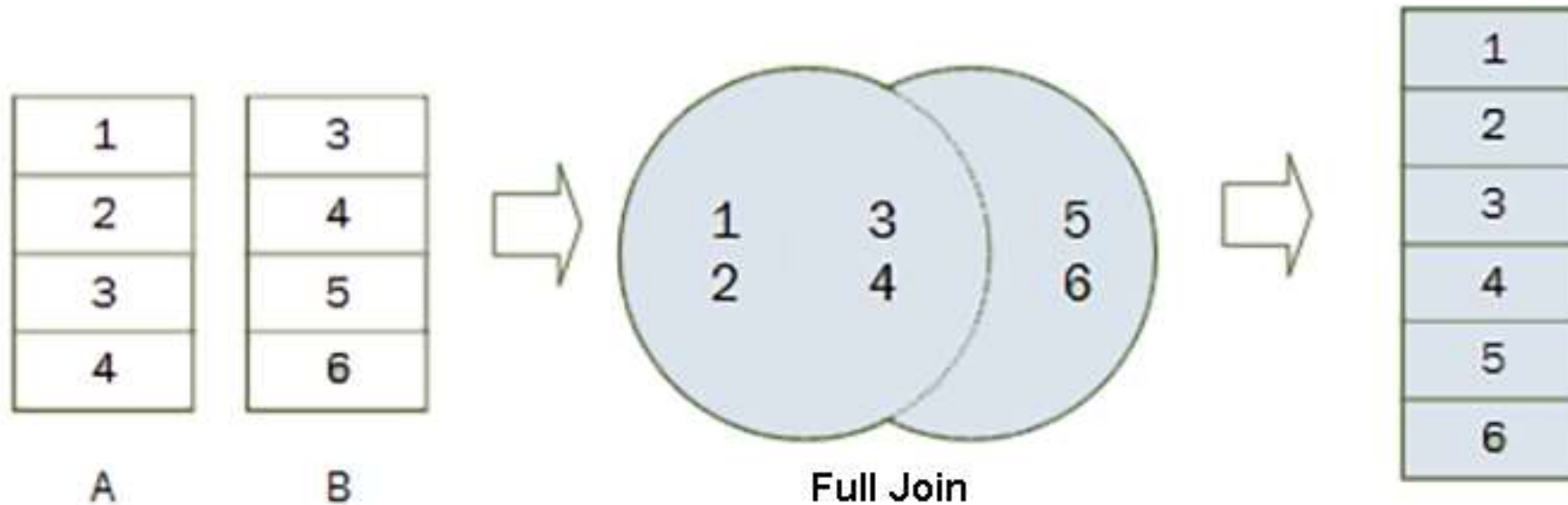
COD_SUCURSAL	DIRECCIÓN	LOCALIDAD
0001	C/ ANCHA, 1	ALMERÍA
0002	C/ NUEVA, 1	GRANADA
0003	C/ CORTÉS, 33	CÁDIZ

```
SELECT E.DNI, E.NOMBRE, S.* FROM EMPLEADOS E RIGHT JOIN SUCURSALES ON
EMPLEADOS.COD_SUCURSAL = SUCURSALES.COD_SUCURSAL;
```

DNI	NOMBRE	<u>COD_SUCURSAL</u>	DIRECCIÓN	LOCALIDAD
11111111A	ANA	0001	C/ ANCHA, 1	ALMERÍA
22222222B	BERNARDO	0001	C/ ANCHA, 1	ALMERÍA
44444444D	DAVID	0003	C/ CORTÉS, 33	CÁDIZ
<u>NULL</u>	<u>NULL</u>	0002	C/ NUEVA, 1	GRANADA

2. Tipos de combinaciones de tablas: OUTER JOIN o combinación externa

Un **OUTER JOIN** te entregará la unión de A y B; es decir, todas las filas de A y todas las filas de B. Si una fila en A no tiene una fila correspondiente en B, la porción de B es null, y viceversa.



2. Tipos de combinaciones de tablas: OUTER JOIN o combinación externa

EMPLEADOS

DNI	NOMBRE	LOCALIDAD	COD_SUCURSAL
11111111A	ANA	ALMERÍA	0001
22222222B	BERNARDO	GRANADA	0001
33333333C	CARLOS	GRANADA	NULL
44444444D	DAVID	JEREZ	0003

SUCURSALES

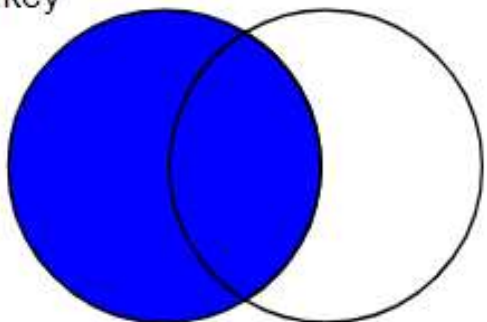
COD_SUCURSAL	DIRECCIÓN	LOCALIDAD
0001	C/ ANCHA, 1	ALMERÍA
0002	C/ NUEVA, 1	GRANADA
0003	C/ CORTÉS, 33	CÁDIZ

SELECT E.DNI, E.NOMBRE, E.LOCALIDAD, S.COD_SUCURSAL, S.LOCALIDAD **FROM** EMPLEADOS E **OUTER JOIN** SUCURSALES **ON** EMPLEADOS.COD_SUCURSAL = SUCURSALES.COD_SUCURSAL;

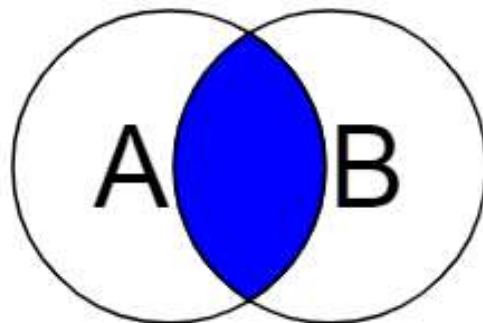
DNI	NOMBRE	LOCALIDAD	<u>COD_SUCURSAL</u>	LOCALIDAD
11111111A	ANA	ALMERÍA	0001	ALMERÍA
22222222B	BERNARDO	GRANADA	0001	ALMERÍA
<u>NULL</u>	<u>NULL</u>	<u>NULL</u>	0002	GRANADA
33333333C	CARLOS	GRANADA	<u>NULL</u>	<u>NULL</u>
44444444D	DAVID	JEREZ	0003	CÁDIZ

2. Tipos de combinaciones de tablas

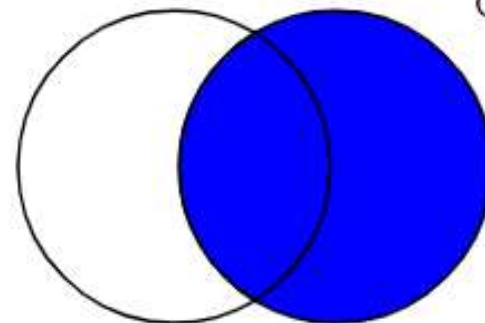
```
SELECT <fields>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.key = B.key
```



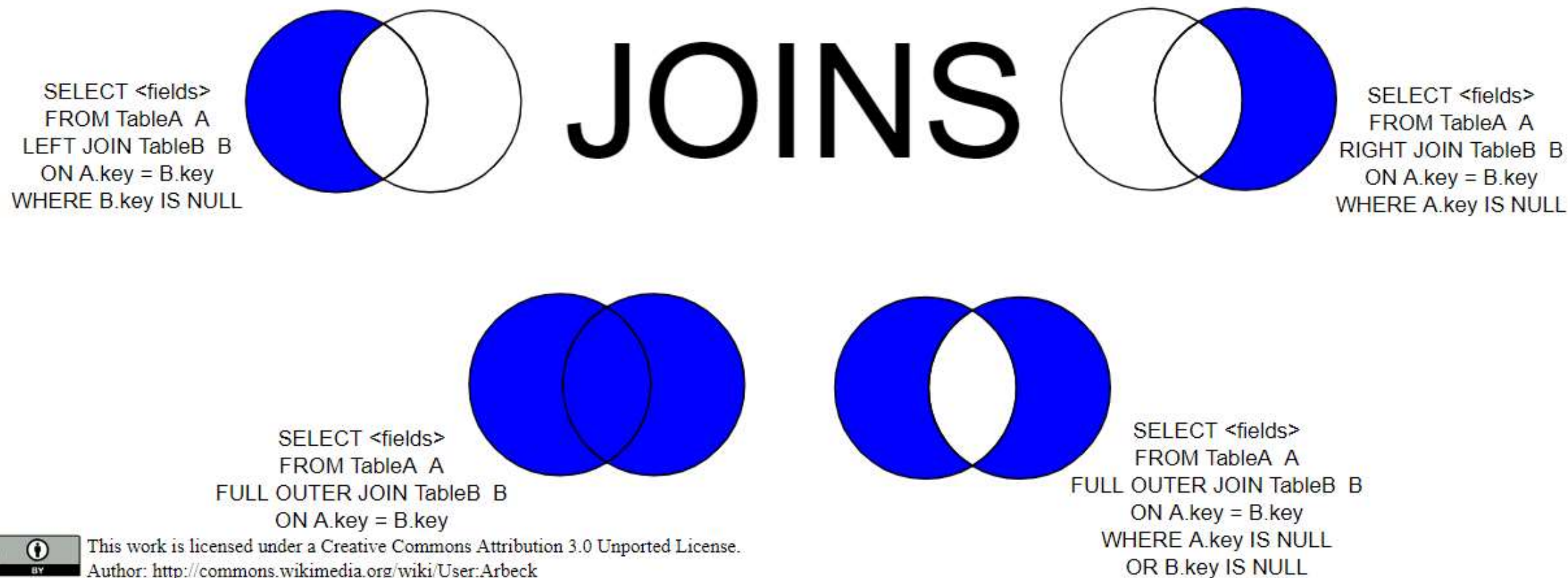
```
SELECT <fields>  
FROM TableA A  
INNER JOIN TableB B  
ON A.key = B.key
```



```
SELECT <fields>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.key = B.key
```



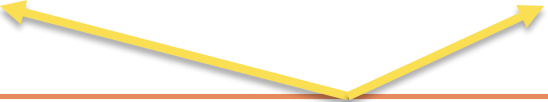
2. Tipos de combinaciones de tablas



3. Subconsultas: Subconsultas básicas de comparación

Enunciado 1: Visualizar el código de cada alumno, código de cursillo y nota siempre que **dicha nota sea mayor que la media** de notas de todos los alumnos en todos los cursillos.

```
SELECT CODAL, CODCUR, NOTA  
FROM ALUMNOSCURSILLOS  
WHERE NOTA > (SELECT AVG(NOTA) FROM ALUMNOSCURSILLOS);
```




Se compara **un valor** (la nota de cada alumno) con **otro valor** (la media de todas las notas de todos los alumnos) **para cada una de las filas**

3. Subconsultas: Subconsultas básicas de pertenencia (IN y NOT IN)

Enunciado 2: Visualizar los nombres de los alumnos que **no están apuntados a ningún** cursillo.

Reformulación: “*Quiero los alumnos cuyo código **no está en la tabla ALUMNOSCURSILLOS***”.

```
SELECT ALUMNOS.NOMAL FROM ALUMNOS  
WHERE ALUMNOS.CODAL NOT IN (SELECT ALUMNOSCURSILLOS.CODAL  
                             FROM ALUMNOSCURSILLOS);
```



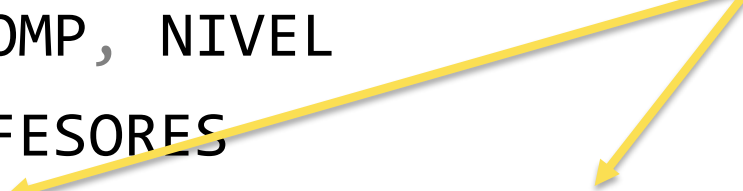
Se comprueba que **un valor** (el código de cada alumno) no está entre **varios valores** (todos los códigos de alumnos matriculados a algún curso)

3. Subconsultas: Subconsultas básicas de pertenencia (IN y NOT IN)

Enunciado 3: Visualizar el nombre y nivel de aquellos **profesores que imparten cursos** que comiencen con H.

Reformulación: “*Quiero el nombre y nivel de los profesores cuyo código esté dentro de la columna CODP (clave ajena) de los cursos que empiezan por H*”.

```
SELECT NOMP, NIVEL  
FROM PROFESORES  
WHERE CODP IN (SELECT CODP FROM CURSILLOS  
                WHERE NOMCUR LIKE 'H%');
```



Se comprueba que **un valor** (el código de cada profesor) está entre **varios valores** (todos los códigos de profesores que imparten cursos que empiezan por H)

3. Subconsultas: Subconsultas básicas de pertenencia (ALL y ANY)

Enunciado 2: Visualizar los nombres de los alumnos que **no están apuntados a ningún** cursillo.

Reformulación: “*Quiero los alumnos cuyo código **es diferente a todos los** códigos de la tabla ALUMNOSCURSILLOS*”

```
SELECT ALUMNOS.NOMAL FROM ALUMNOS
WHERE ALUMNOS.CODAL != ALL (SELECT ALUMNOSCURSILLOS.CODAL
                                FROM ALUMNOSCURSILLOS);
```



Se comprueba que **un valor** (el código de cada alumno) es diferente a **varios valores** (todos los códigos de alumnos matriculados a algún curso)

3. Subconsultas: Subconsultas ALL y ANY

Enunciado 3: Visualizar el nombre y nivel de aquellos **profesores que imparten cursillos** que comiencen con H.

Reformulación: “*Quiero el nombre y nivel de los profesores cuyo código sea igual a algún valor de la columna CODP (clave ajena) de los cursillos que empiezan por H*”.

```
SELECT NOMP, NIVEL  
FROM PROFESORES  
WHERE CODP = ANY (SELECT CODP FROM CURSILLOS  
                   WHERE NOMCUR LIKE 'H%');
```


Se comprueba que **un valor** (el código de cada profesor) es igual a alguno de **varios valores** (todos los códigos de profesores que imparten cursos que empiezan por H)

3. Subconsultas: Subconsultas correlacionadas

Enunciado 4: Visualizar el nombre, sueldo y código de departamento de los profesores con un sueldo mayor o igual que la media del sueldo de su **departamento**.

Reformulación: “Quiero el nombre de los profesores cuyo sueldo **sea mayor a la media de sueldos de los profesores que tienen el mismo CODDEP que él**”

```
SELECT P1.NOMP, SALARIO, CODDEP FROM PROFESORES P1
WHERE P1.SALARIO >= (SELECT AVG(P2.SALARIO) FROM PROFESORES P2
                     WHERE P2.CODDEP=P1.CODDEP)
```



Nos aseguramos que el sueldo de cada profesor es comparado con la media de profesores **que pertenecen al mismo departamento**.

3. Subconsultas: Subconsultas con predicado EXISTS

El predicado **EXISTS** devuelve *Verdadero* si la subconsulta devuelve alguna fila de resultados y *Falso* en caso contrario.

EXISTS siempre implica una consulta correlacionada.

Enunciado 2: Visualizar los nombres de los alumnos que **no están apuntados a ningún** cursillo.

Reformulación: “Quiero los alumnos para los cuales **no exista** ninguna matrícula en *ALUMNOSCURSILLOS* con su código de alumno”.

```
SELECT A.NOMAL FROM ALUMNOS A
WHERE NOT EXISTS (SELECT *
                  FROM ALUMNOSCURSILLOS AC
                  WHERE AC.CODAL=A.CODAL);
```

Se mostrarán los alumnos para los cuales **no exista ninguna fila** en *ALUMNOSCURSILLOS* con su código de alumno.

4. Orden de ejecución de cláusulas

PROCESAMIENTO DE CONSULTA LOGICO



TECLEADO EN ORDEN DE LA CONSULTA

