

Bases de datos relacionales y SQL

Jesús Arias Fisteus

Computación Web (2024/25)

uc3m

Universidad **Carlos III** de Madrid

Departamento de Ingeniería Telemática

Parte I

Introducción a las bases de datos relacionales

- ▶ Colección de datos almacenados en **una o más tablas**.
- ▶ Las tablas constan de **filas y columnas**.
- ▶ Las tablas pueden estar **relacionadas entre sí**.

Ejemplo: una única tabla

Tabla “Paises”

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	Europa	505370	46438422	monarquía	Madrid
2	Francia	Europa	643801	64590000	república	París
3	Canadá	América	9984670	36155487	monarquía	Ottawa
4	Alemania	Europa	357022	81770900	república	Berlín
5	Australia	Oceanía	7692024	23613193	monarquía	Canberra

Ejemplo: varias tablas relacionadas

Tabla “Países”

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Tabla “Continentes”

id	nombre	superficie	poblacion
1	Europa	10180000	731000000
2	América	42330000	910000000
3	Oceanía	9008458	38889988

Ejemplo: varias tablas relacionadas

Tabla “Ciudades”

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.20	381488

Un **gestor de bases de datos relacionales** es un programa que da soporte al uso de bases de datos relacionales.

Algunos ejemplos de gestores de bases de datos relacionales son: Oracle Database, Microsoft SQL Server, IBM DB2, SAP Advantage Database Server, MySQL, MariaDB, PostgreSQL, y SQLite.

SQL es el lenguaje estándar para utilizar y mantener bases de datos relacionales.

Todos los gestores de bases de datos relacionales dan soporte a SQL, aunque con pequeñas variaciones (*dialectos*) dependiendo del gestor.

Parte II

El lenguaje SQL: recuperación de datos

La sentencia **SELECT**

La sentencia **SELECT** se utiliza para recuperar datos de una tabla.

¿Qué países contiene la tabla y con qué datos?

```
SELECT *  
FROM Países
```

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué población tiene cada país?

```
SELECT nombre, poblacion  
FROM Países
```

nombre	poblacion
España	46438422
Francia	64590000
Canadá	36155487
Alemania	81770900
Australia	23613193

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué población y superficie tiene España?

```
SELECT superficie, poblacion  
FROM Países  
WHERE nombre= 'España'
```

superficie	poblacion
505370	46438422

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué población y superficie tiene el país con identificador 2?

```
SELECT superficie, poblacion  
FROM Países  
WHERE id=2
```

superficie	poblacion
643801	64590000

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

- ▶ Operadores Booleanos: **AND**, **OR**, **NOT**.
- ▶ Expresiones de comparación: **=**, **<>**, **<**, **>**, **<=**, **>=**, **BETWEEN**.
- ▶ Pertenencia a un conjunto de valores: **IN**.
- ▶ Comprobación de nulidad: **IS NULL**, **IS NOT NULL**.

¿Qué países tienen más de 50 millones de habitantes?

```
SELECT nombre, poblacion
FROM Países
WHERE poblacion > 50000000
```

nombre	poblacion
Francia	64590000
Alemania	81770900

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué países del continente con identificador 1 tienen menos de 50 millones de habitantes?

```
SELECT nombre, poblacion
FROM Países
WHERE poblacion < 50000000 AND continente = 1
```

nombre	poblacion
España	46438422

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué países tienen entre 50 y 70 millones de habitantes, ambos inclusive?

```
SELECT nombre, poblacion
FROM Países
WHERE poblacion BETWEEN 50000000 AND 70000000
```

nombre	poblacion
Francia	64590000

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué países están en el continente 1 o en el 3?

```
SELECT nombre
FROM Países
WHERE continente IN (1, 3)
```

nombre

España
Francia
Alemania
Australia

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

ORDER BY especifica la columna sobre la cual se deben ordenar las filas.

Se pueden especificar varias columnas separadas por comas, teniendo la primera de la lista mayor peso, y la última menor peso, en la ordenación.

El orden es ascendente por defecto. Se especifica orden descendente con la palabra **DESC** tras el nombre de la columna.

Ordena los países por población.

```
SELECT nombre, poblacion
FROM Países
ORDER BY poblacion
```

nombre	poblacion
Australia	23613193
Canadá	36155487
España	46438422
Francia	64590000
Alemania	81770900

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Ordena los países por población, de mayor a menor.

```
SELECT nombre, poblacion
FROM Países
ORDER BY poblacion DESC
```

nombre	poblacion
Alemania	81770900
Francia	64590000
España	46438422
Canadá	36155487
Australia	23613193

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

*Ordena los países de **menos de 40 millones de habitantes** por población, de mayor a menor.*

```
SELECT nombre, poblacion
FROM Países
WHERE poblacion < 40000000
ORDER BY poblacion DESC
```

nombre	poblacion
Canadá	36155487
Australia	23613193

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Con **LIMIT** se especifica el número máximo a obtener.

Con **OFFSET** se especifica un número de resultados a descartar (contando desde el primer resultado). Se suele combinar con **LIMIT** para recorrer los resultados por rangos.

¿Cuáles son los **dos países** con mayor población?

```
SELECT nombre, poblacion
FROM Países
ORDER BY poblacion DESC
LIMIT 2
```

nombre	poblacion
Alemania	81770900
Francia	64590000

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Y los **dos** siguientes?

```
SELECT nombre, poblacion
FROM Países
ORDER BY poblacion DESC
LIMIT 2
OFFSET 2
```

nombre	poblacion
España	46438422
Canadá	36155487

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

- ▶ Práctica 3:
 - ▶ Ejercicios 1 y 2.

Parte III

Consultas de agregación y agrupamiento

¿Qué identificadores de continente aparecen en la tabla?

```
SELECT continente  
FROM Países
```

continente

1
1
1
2
3

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

*¿Qué identificadores de continente aparecen en la tabla, **sin repetir valores?***

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Eliminación de duplicados

Si entre los resultados de una consulta aparecen filas iguales, el modificador **DISTINCT** elimina todas excepto una.

```
SELECT DISTINCT continente  
FROM Países
```

continente

1

2

3

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

*¿Cuál es la **suma agregada**
de la superficie y la población
de los países del continente con identificador 1?*

1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
4	Alemania	1	357022	81770900	república	6
			1506193	192799322		

Agregación de filas: sumas, medias, etc.

Las funciones **SUM**, **AVG**, **MIN** y **MAX** agregan los datos de las filas seleccionadas.

```
SELECT SUM(superficie), SUM(poblacion)
FROM Países
WHERE continente=1
```

SUM(superficie)	SUM(poblacion)
1506193	192799322

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Cuántos países de la tabla pertenecen al continente con identificador 1?

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Cuenta del número de resultados

La función **COUNT** devuelve el número de filas seleccionadas.

```
SELECT COUNT(*)  
FROM Países  
WHERE continente=1
```

COUNT(*)

3

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

*¿De **cuántos** monumentos
conocemos en qué ciudad se encuentran?*

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

Cuenta del número de resultados

Aplicada a una columna, la función **COUNT** devuelve el número de valores no nulos en la misma.

```
SELECT COUNT(ciudad)
FROM Monumentos
```

COUNT(ciudad)

3

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

¿A cuántas ciudades distintas pertenecen los monumentos de la tabla?

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

Cuenta del número de resultados

También es posible contar valores no nulos, sin tener en cuenta sus repeticiones, combinando **COUNT** con **DISTINCT**.

```
SELECT COUNT(DISTINCT ciudad)
FROM Monumentos
```

COUNT(DISTINCT ciudad)

2

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

Para cada uno de los continentes,
*¿cuál es la suma de la superficie
de los países que pertenecen al mismo?*

1	1506193
2	9984670
3	7692024

GROUP BY permite hacer cálculos agregados (suma, media, etc.) sobre grupos de filas.

```
SELECT continente, SUM(superficie)
FROM Países
GROUP BY continente
```

continente	SUM(superficie)
1	1506193
2	9984670
3	7692024

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Cuánta población está sujeta
en cada continente
a cada una de las formas de gobierno?

GROUP BY permite también agrupar por el valor de una combinación de varias columnas.

```
SELECT continente, gobierno, SUM(poblacion)
FROM Países
GROUP BY continente, gobierno
```

continente	gobierno	SUM(poblacion)
1	república	146360900
1	monarquía	46438422
2	monarquía	36155487
3	monarquía	23613193

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

**¿Para qué continentes es la suma agregada
de la superficie de sus países
mayor de 5 millones de kilómetros cuadrados?**

HAVING filtra los grupos resultantes.

```
SELECT continente, SUM(superficie)
FROM Países
GROUP BY continente
HAVING SUM(superficie) > 5000000
```

continente	SUM(superficie)
2	9984670
3	7692024

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

No se debe confundir **WHERE** con **HAVING**: el primero filtra filas antes de agrupar, el segundo filtra grupos.

```
SELECT continente, SUM(superficie)
FROM Países
WHERE poblacion < 70000000
GROUP BY continente
HAVING SUM(superficie) < 9000000
```

continente	SUM(superficie)
1	1149171
3	7692024

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Se puede establecer el nombre que tomarán las columnas resultantes de una consulta.

```
SELECT
  continente,
  SUM(superficie) AS 'superficie agregada'
FROM Países
GROUP BY continente
```

continente	superficie agregada
1	1506193
2	9984670
3	7692024

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

*¿Cuál es la **densidad de población** de cada país?*

Se pueden obtener columnas resultantes de realizar cálculos sobre los valores de otras columnas.

```
SELECT
nombre,
poblacion/superficie AS densidad
FROM Países
```

nombre	densidad
España	91.88994598017294
Francia	100.3260324230624
Canadá	3.6210998460640162
Alemania	229.0360257911277
Australia	3.0698283052678983

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué países tienen una densidad de población mayor de 100 habitantes por kilómetro cuadrado?

```
SELECT nombre  
FROM Países  
WHERE poblacion/superficie > 100
```

nombre

Francia
Alemania

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

- ▶ Práctica 3:
 - ▶ Ejercicio 3

Parte IV

Combinación de tablas

Tabla "Países"

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Tabla "Ciudades"

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

¿Cuál es la capital de cada país?

Relaciones entre tablas

Tabla "Países"

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Tabla "Ciudades"

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

Las **relaciones entre tablas** se explicitan por medio de:

- ▶ **Claves primarias**: columna o combinación de columnas que identifican unívocamente a las filas de una tabla.
- ▶ **Claves ajenas** (también conocidas como claves externas o foráneas): columna o combinación de columnas en una tabla que hacen referencia a la clave primaria de otra tabla.

Relaciones entre tablas

Tabla "Países"

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

clave
primaria

Tabla "Ciudades"

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

clave
ajena

¿Cuál es la capital de cada país?

1	España	1	505370	46438422	monarquía	1	1	Madrid	605.77	3141991
2	Francia	1	643801	64590000	república	3	3	París	105.40	2229621
3	Canadá	2	9984670	36155487	monarquía	4	4	Ottawa	2778.64	1083391
4	Alemania	1	357022	81770900	república	6	6	Berlín	891.68	3469849
5	Australia	3	7692024	23613193	monarquía	7	7	Canberra	814.2	381488

5	Nueva York	1214.00	8491079
2	Barcelona	102.15	1604555

El operador **INNER JOIN** combina dos tablas: se forman todas las combinaciones de filas de una tabla con filas de la otra para las cuales la expresión de la cláusula **ON** se evalúe con valor *cierto*.

Combinación interna

```
SELECT *  
FROM Países  
INNER JOIN Ciudades  
ON capital=Ciudades.id
```

id	nombre	cont.	sup.	pob.	gob.	cap.	id	nombre	sup.	pob.
1	España	1	505370	46438422	monarquía	1	1	Madrid	605.77	3141991
2	Francia	1	643801	64590000	república	3	3	París	105.40	2229621
3	Canadá	2	9984670	36155487	monarquía	4	4	Ottawa	2778.64	1083391
4	Alemania	1	357022	81770900	república	6	6	Berlín	891.68	3469849
5	Australia	3	7692024	23613193	monarquía	7	7	Canberra	814.2	381488

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Combinación interna

```
SELECT
Países.nombre ,
Ciudades.nombre
FROM Países
INNER JOIN Ciudades
ON capital=Ciudades.id
```

nombre	nombre
España	Madrid
Francia	París
Canadá	Ottawa
Alemania	Berlín
Australia	Canberra

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Combinación interna

El orden en que se combinan tablas con **INNER JOIN** es indiferente.

```
SELECT
Países.nombre,
Ciudades.nombre
FROM Ciudades
INNER JOIN Países
ON capital=Ciudades.id
```

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

nombre	nombre
España	Madrid
Francia	París
Canadá	Ottawa
Alemania	Berlín
Australia	Canberra

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Combinación interna

```
SELECT
Países.nombre AS país,
Ciudades.nombre AS capital
FROM Países
INNER JOIN Ciudades
ON capital=Ciudades.id
```

país	capital
España	Madrid
Francia	París
Canadá	Ottawa
Alemania	Berlín
Australia	Canberra

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

¿Qué países tienen una capital con más de tres millones de habitantes?

1	España	1	505370	46438422	monarquía	1	1 Madrid	605.77	3141991
2	Francia	1	643801	64590000	república	3	3 París	105.40	2229621
3	Canadá	2	9984670	36155487	monarquía	4	4 Ottawa	2778.64	1083391
4	Alemania	1	357022	81770900	república	6	6 Berlín	891.68	3469849
5	Australia	3	7692024	23613193	monarquía	7	7 Canberra	814.2	381488

¿Qué países tienen una capital con más de tres millones de habitantes?

```
SELECT
Países.nombre AS país
FROM Países
INNER JOIN Ciudades
ON capital=Ciudades.id
WHERE Ciudades.poblacion>3000000
```

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlin	891.68	3469849
7	Canberra	814.2	381488

país

España
Alemania

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Alias de nombres de tablas

```
SELECT
P.nombre AS país
FROM Países AS P
INNER JOIN Ciudades AS C
ON capital=C.id
WHERE C.poblacion>3000000
```

país

España
Alemania

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlin	891.68	3469849
7	Canberra	814.2	381488

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

Combinación interna: filas seleccionadas

Las filas de cualquiera de las dos tablas que no cumplan el criterio para ningún posible par no aparecerán entre los resultados.

En el ejemplo, las filas de Nueva York y Barcelona.

Las filas de cualquiera de las dos tablas que cumplan el criterio para varios pares aparecerán varias veces entre los resultados, una por cada par.

Tabla “Ciudades”

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

Tabla “Monumentos”

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

*Obtener una lista de monumentos
junto con el nombre de la ciudad
en que se encuentran.*

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlin	891.68	3469849
7	Canberra	814.2	381488

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

Obtener una lista de monumentos junto con el nombre de la ciudad en que se encuentran.

```
SELECT
Monumentos.nombre AS monumento,
Ciudades.nombre AS ciudad
FROM Monumentos
INNER JOIN Ciudades
ON ciudad=Ciudades.id
```

monumento	ciudad
Estatua de la Libertad	Nueva York
Torre Eiffel	París
Empire State Building	Nueva York

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlin	891.68	3469849
7	Canberra	814.2	381488

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

*Obtener una lista de monumentos
junto con el nombre de la ciudad y país
en que se encuentran.*

2 Torre Eiffel	3	3 París	105.40	2229621	2 Francia	1	643801	64590000	república	3
----------------	---	---------	--------	---------	-----------	---	--------	----------	-----------	---

1 Estatua de la Libertad	5	5 Nueva York	1214.00	8491079	1 España	1	505370	46438422	monarquía	1
3 Empire State Building	5	5 Nueva York	1214.00	8491079	3 Canadá	2	9984670	36155487	monarquía	4
					4 Alemania	1	357022	81770900	república	6
					5 Australia	3	7692024	23613193	monarquía	7

Combinación interna de más de dos tablas

Obtener una lista de monumentos junto con el nombre de la ciudad y país en que se encuentran.

```
SELECT
Monumentos.nombre AS monumento,
Ciudades.nombre AS ciudad,
Países.nombre AS país
FROM Monumentos
INNER JOIN Ciudades
ON ciudad=Ciudades.id
INNER JOIN Países
ON capital=Ciudades.id
```

monumento	ciudad	país
-----------	--------	------

Torre Eiffel	París	Francia
--------------	-------	---------

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlin	891.68	3469849
7	Canberra	814.2	381488

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

id	nombre	cont.	superficie	poblacion	gobierno	cap.
1	España	1	505370	46438422	monarquía	1
2	Francia	1	643801	64590000	república	3
3	Canadá	2	9984670	36155487	monarquía	4
4	Alemania	1	357022	81770900	república	6
5	Australia	3	7692024	23613193	monarquía	7

- ▶ Práctica 3:
 - ▶ Ejercicio 4

*¿Qué **pares de monumentos**
están ubicados en la misma ciudad?*

Es posible unir una tabla consigo misma mediante el uso de alias de nombres de tablas: se toman dos “copias” de la misma tabla, cada una con un alias que la diferencia de la otra.

1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

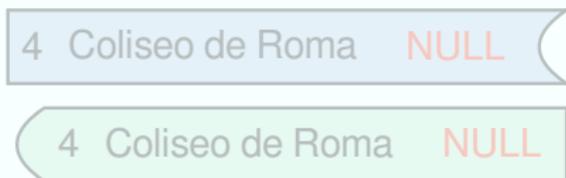
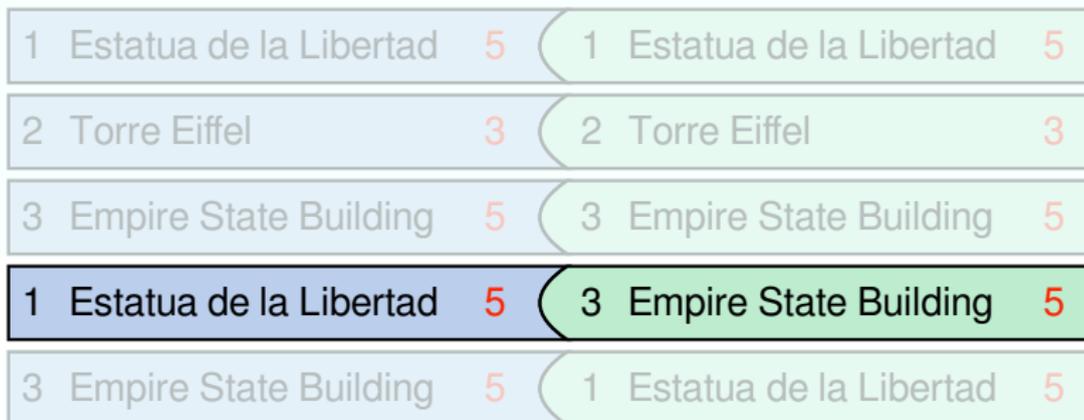
¿Qué **pares de monumentos** están ubicados en la misma ciudad?

```
SELECT
A.nombre, B.nombre
FROM Monumentos AS A
INNER JOIN Monumentos AS B
ON A.ciudad=B.ciudad
```

nombre	nombre
Estatua de la Libertad	Estatua de la Libertad
Empire State Building	Estatua de la Libertad
Torre Eiffel	Torre Eiffel
Estatua de la Libertad	Empire State Building
Empire State Building	Empire State Building

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

Autocombinación



¿Qué pares de monumentos están ubicados en la misma ciudad?

```
SELECT
A.nombre, B.nombre
FROM Monumentos AS A
INNER JOIN Monumentos AS B
ON A.ciudad=B.ciudad
WHERE A.id<B.id
```

nombre	nombre
Estatua de la Libertad	Empire State Building

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

*¿Qué monumentos hay y,
cuando dicha información esté disponible,
en qué ciudad están ubicados?*

Mediante **combinación externa** se recuperan todas las filas de la **tabla primaria** aunque no cumplan el criterio con ninguna fila de la otra tabla.

Tipos de combinación externa:

- ▶ **LEFT JOIN**: la tabla primaria es la especificada antes del operador **JOIN** (a la izquierda).
- ▶ **RIGHT JOIN**: la tabla primaria es la especificada tras el operador **JOIN** (a la derecha).
- ▶ **FULL JOIN**: ambas tablas son primarias.

¿Qué monumentos hay y, cuando dicha información esté disponible, en qué ciudad están ubicados?

```
SELECT
Monumentos.nombre, Ciudades.nombre
FROM Monumentos
LEFT JOIN Ciudades
ON ciudad=Ciudades.id
```

nombre	nombre
Estatua de la Libertad	Nueva York
Torre Eiffel	París
Empire State Building	Nueva York
Coliseo de Roma	NULL

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

*¿Qué ciudades hay y,
cuando dicha información esté disponible,
qué monumentos hay en cada ciudad?*

Combinación externa

¿Qué ciudades hay y, cuando dicha información esté disponible, qué monumentos hay en cada ciudad?

```
SELECT
Monumentos.nombre, Ciudades.nombre
FROM Monumentos
RIGHT JOIN Ciudades
ON ciudad=Ciudades.id
```

nombre	nombre
NULL	Madrid
NULL	Barcelona
Torre Eiffel	París
NULL	Ottawa
Estatua de la Libertad	Nueva York
Empire State Building	Nueva York
NULL	Berlín
NULL	Canberra

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

Combinación externa

Se puede cambiar el orden de las tablas utilizando **LEFT JOIN** en vez de **RIGHT JOIN**.

```
SELECT
Monumentos.nombre, Ciudades.nombre
FROM Ciudades
LEFT JOIN Monumentos
ON ciudad=Ciudades.id
```

nombre	nombre
Madrid	NULL
Barcelona	NULL
París	Torre Eiffel
Ottawa	NULL
Nueva York	Estatua de la Libertad
Nueva York	Empire State Building
Berlín	NULL
Canberra	NULL

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

Consultas FULL JOIN (no disponibles en MySQL)

```
SELECT
Monumentos.nombre, Ciudades.nombre
FROM Ciudades
FULL JOIN Monumentos
ON ciudad=Ciudades.id
```

nombre	nombre
Madrid	NULL
Barcelona	NULL
París	Torre Eiffel
Ottawa	NULL
Nueva York	Estatua de la Libertad
Nueva York	Empire State Building
Berlín	NULL
Canberra	NULL
NULL	Coliseo de Roma

id	nombre	ciudad
1	Estatua de la Libertad	5
2	Torre Eiffel	3
3	Empire State Building	5
4	Coliseo de Roma	NULL

id	nombre	superficie	poblacion
1	Madrid	605.77	3141991
2	Barcelona	102.15	1604555
3	París	105.40	2229621
4	Ottawa	2778.64	1083391
5	Nueva York	1214.00	8491079
6	Berlín	891.68	3469849
7	Canberra	814.2	381488

Parte V

El lenguaje SQL: inserción y modificación de datos

- ▶ Se crean tablas con **CREATE TABLE**.
- ▶ Al crear una tabla, se especifican aspectos como:
 - ▶ Nombre de la tabla.
 - ▶ Para cada columna:
 - ▶ Nombre y tipo de datos
 - ▶ Valor por defecto
 - ▶ Si puede tomar valor `NULL`.
 - ▶ Si es un campo de auto-incremento.
 - ▶ Claves primarias y ajenas.
 - ▶ Índices.

```
CREATE TABLE Continentes (  
    id INT NOT NULL auto_increment,  
    nombre VARCHAR(255) NOT NULL,  
    superficie DOUBLE NOT NULL,  
    poblacion LONG NOT NULL,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Ciudades (  
    id INT NOT NULL auto_increment,  
    nombre VARCHAR(255) NOT NULL,  
    superficie DOUBLE NOT NULL,  
    poblacion LONG NOT NULL,  
    PRIMARY KEY(id)  
)
```

```
CREATE TABLE Países (  
  id INT NOT NULL auto_increment,  
  nombre VARCHAR(255) NOT NULL,  
  continente INT NOT NULL,  
  superficie DOUBLE NOT NULL,  
  poblacion LONG NOT NULL,  
  gobierno ENUM('monarquia', 'republica'),  
  capital INT NOT NULL,  
  PRIMARY KEY(id),  
  CONSTRAINT FOREIGN KEY (continente)  
    REFERENCES Continentes(id),  
  CONSTRAINT FOREIGN KEY (capital)  
    REFERENCES Ciudades(id)  
)
```

Creación de tablas

```
1 CREATE TABLE AreasGeograficas (  
2     id INT NOT NULL auto_increment,  
3     nombre VARCHAR(255) NOT NULL,  
4     tipo ENUM('continente', 'pais',  
5              'region', 'ciudad') NOT NULL,  
6     area DOUBLE NOT NULL,  
7     habitantes LONG NOT NULL,  
8     PRIMARY KEY(id)  
9 )
```

Cada columna tiene un **tipo de datos** asociado (numérico, cadena de texto, fecha/hora, etc.). El valor especial **NULL** indica ausencia de valor en la columna.

Los tipos de datos concretos disponibles varían según el gestor de bases de datos.

Tipos de datos numéricos en MySQL

- ▶ Bits: **BIT** (p.e. cuatro bits: **BIT(4)**).
- ▶ Enteros con signo (opcional **UNSIGNED**):
 - ▶ 1 byte: **TINYINT** / **BOOL** / **BOOLEAN**.
 - ▶ 2 bytes: **SMALLINT**.
 - ▶ 3 bytes: **MEDIUMINT**.
 - ▶ 4 bytes: **INT** / **INTEGER**.
 - ▶ 8 bytes: **BIGINT**.
- ▶ Punto flotante:
 - ▶ 4 bytes (precisión aprox. 7 decimales): **FLOAT**.
 - ▶ 8 bytes (precisión aprox. 15 decimales): **DOUBLE**.
- ▶ Punto fijo: **DECIMAL** / **NUMERIC**
 - ▶ P.e. **DECIMAL(6, 2)** representa 6 dígitos, de los cuales 2 son decimales.

Tipos de datos de cadenas

- ▶ Cadenas de caracteres:
 - ▶ Tamaño fijo: **CHAR** (p.e. **CHAR(8)**).
 - ▶ Tamaño variable: **VARCHAR** (p.e. **VARCHAR(255)**).
 - ▶ Tamaño grande: **TEXT**.
- ▶ Cadenas de bytes:
 - ▶ Tamaño fijo: **BINARY** (p.e. **BINARY(8)**).
 - ▶ Tamaño variable: **VARBINARY** (p.e. **VARBINARY(255)**).
 - ▶ Tamaño grande: **BLOB**.
- ▶ Enumerados: **ENUM**
 - ▶ P.e.: `ENUM('profesor', 'alumno', 'administrativo')`.
- ▶ Conjuntos: **SET**
 - ▶ P.e.: `SET('rojo', 'verde', 'azul')`.

Tipos de datos de fechas y horas en MySQL

- ▶ Fecha: **DATE**.
- ▶ Fecha y hora (con minutos y segundos, sin zona horaria): **DATETIME**.
- ▶ Hora: **TIME**.
- ▶ Año: **YEAR** (para año con dos dígitos, `YEAR(2)`).
- ▶ Sello temporal (número de segundos desde 1-1-1970): **TIMESTAMP**.
 - ▶ Por defecto, la primera columna de tipo `TIMESTAMP` se actualiza automáticamente con `INSERT` y `UPDATE`, salvo que se le asigne un valor o se especifique lo contrario.

- ▶ Mostrar todas las tablas de una base de datos: **SHOW TABLES**
- ▶ Mostrar la estructura de una tabla: **DESCRIBE** (p.e.: DESCRIBE Países).
- ▶ Eliminar una tabla: **DROP TABLE** (p.e.: DROP TABLE Países).
- ▶ Añadir, modificar o eliminar columnas en una tabla: **ALTER TABLE**.

Se insertan filas nuevas en tablas mediante **INSERT INTO**.

Se puede especificar qué columnas se proporcionan y en qué orden. De lo contrario, por defecto, se proporcionan todas y en el mismo orden en que se definieron al crear la tabla.

Se puede insertar una o más filas en una única sentencia.

```
INSERT INTO Continentes  
VALUES  
(4, 'Asia', 44579000.0, 4164252000)
```

Cuando se omite el valor de alguna columna:

- ▶ Columnas autoincrementales:
 - ▶ Reciben automáticamente un valor numérico único que se incrementa a medida que se insertan filas en la tabla.
 - ▶ Se suelen utilizar como clave primaria.
- ▶ Otros tipos de columnas:
 - ▶ Reciben el valor por defecto definido para dicha columna, o **NULL** si no se ha definido dicho valor.

Columnas para las que no se proporciona valor

```
INSERT INTO Continentes
(nombre, superficie, poblacion)
VALUES
('Africa', 30370000, 1100000000),
('Antártida', 14000000, 135)
```

Se pueden insertar filas resultantes de una consulta **SELECT**.

```
INSERT INTO Ciudades  
(nombre, superficie, poblacion)  
SELECT  
nombre, area, habitantes  
FROM AreasGeograficas  
WHERE tipo='ciudad'
```

También es posible insertar el resultado de una consulta **SELECT** en una columna concreta.

```
INSERT INTO Países
(nombre, capital, continente, superficie,
 poblacion)
VALÚES
('Italia',
 (SELECT id
  FROM Ciudades
  WHERE nombre='Roma'),
 (SELECT id
  FROM Continentes
  WHERE nombre='Europa'),
301340.0,
60600000
)
```

- ▶ Práctica 4:
 - ▶ Ejercicio 1
 - ▶ Ejercicio 2
 - ▶ Ejercicio 3

Se eliminan filas en una tabla mediante la sentencia **DELETE**.

Se eliminan las filas que cumplan la condición dada, o todas si no se establece condición.

```
DELETE  
FROM Países  
WHERE poblacion < 4000000
```

Se actualizan valores mediante la sentencia **UPDATE**.

Se indica qué columnas se desea cambiar y su nuevo valor. El resto de columnas mantendrán su valor.

Se modifican las filas que cumplan la condición dada, o todas si no se establece condición.

```
UPDATE Países  
SET  
poblacion=46438500,  
superficie=505371  
WHERE id=1
```

```
UPDATE Países  
SET  
poblacion=poblacion-1  
WHERE continente=1
```

```
UPDATE Países
SET
poblacion=
(SELECT habitantes
 FROM AreasGeograficas
 WHERE Países.nombre=AreasGeograficas.nombre)
WHERE EXISTS
(SELECT *
 FROM AreasGeograficas
 WHERE Países.nombre=AreasGeograficas.nombre)
```

Parte VI

Consistencia de datos

Consistencia en las claves ajenas

- ▶ Una clave ajena debe corresponderse con una clave primaria que exista en la tabla a la que dicha clave ajena hace referencia.
 - ▶ Ejemplo: si `continente=2` en la fila con `id=3` en la tabla `Países`, en la tabla `Continentes` debe existir una fila con `id=2`.
- ▶ De lo contrario, la base de datos está en un estado inconsistente.
- ▶ La columna de clave ajena puede admitir valor **NULL** si se establece así al crear la tabla. Esto no supondría una inconsistencia.

Existe el riesgo de introducir inconsistencias en claves ajenas cuando:

- ▶ Se inserta una nueva fila:
 - ▶ Ejemplo: se inserta una fila en Países con continente=27, pero no existe esta fila en Continentes.
- ▶ Se elimina una fila:
 - ▶ Ejemplo: se elimina la fila con id=1 en Continentes, pero se mantienen filas con continente=1 en Países.
- ▶ Se modifica una columna en una fila:
 - ▶ Ejemplo: se establece el valor continente=27 en una fila de Países, pero no existe esta fila en Continentes.
 - ▶ Ejemplo: se modifica id de 1 a 7 en una fila de Continentes, pero se mantienen filas con continente=1 en Países.

Ejemplo

```
1 DELETE FROM Continentes WHERE id=1;
2
3 INSERT INTO Paises
4 (nombre, continente, capital,
5  superficie, poblacion)
6 VALUES
7 ('Estados Unidos', 27, 5, 0.0, 0);
8
9 UPDATE Paises SET continente=27 WHERE id=1;
10
11 UPDATE Continentes SET id=7 WHERE id=1;
```

Consistencia en las claves ajenas

- ▶ El gestor de bases de datos puede detectar sentencias que introducirían inconsistencias y evitarlo.
- ▶ Para cada clave ajena se puede configurar el comportamiento deseado en estas situaciones:
 - ▶ No ejecutar la sentencia y notificar el error (**RESTRICT** o **NO ACTION**). Es la opción por defecto.
 - ▶ Establecer valor NULL en la clave ajena afectada si esta lo permite (**SET NULL**).
 - ▶ Propagar el cambio a la clave ajena (**CASCADE**):
 - ▶ Ejemplo: si se elimina la fila con `id=1` en `Continentes`, se eliminan automáticamente todas las filas con `continente=1` en `Países`.
 - ▶ Ejemplo: si se modifica `id` de 1 a 7 en una fila de `Continentes`, se cambia `continente` de 1 a 7 en todas las filas afectadas de `Países`.
 - ▶ Establecer el valor por defecto de la clave ajena (**SET DEFAULT**).

```
CREATE TABLE Paises (  
    id INT NOT NULL auto_increment,  
    nombre VARCHAR(255) NOT NULL,  
    continente INT,  
    superficie DOUBLE NOT NULL,  
    poblacion LONG NOT NULL,  
    capital INT NOT NULL,  
    PRIMARY KEY(id),  
    CONSTRAINT FOREIGN KEY (continente)  
        REFERENCES Continentes(id)  
        ON UPDATE CASCADE  
        ON DELETE SET NULL,  
    CONSTRAINT FOREIGN KEY (capital)  
        REFERENCES Ciudades(id)  
)
```

Parte VII

Índices

- ▶ Los **índices** consisten en estructuras de datos adicionales cuyo objeto es agilizar la ejecución de determinadas búsquedas de datos en una tabla.
- ▶ Ventajas:
 - ▶ Localización más rápida de datos en la tabla en acceso aleatorio.
 - ▶ Acceso a datos en orden de forma más rápida.
- ▶ Desventajas:
 - ▶ Coste adicional en la inserción/modificación de datos.
 - ▶ Necesidad de más espacio de almacenamiento.
- ▶ Para la clave primaria se construye implícitamente un índice.

```
-- Crear un índice en una tabla existente
CREATE [UNIQUE] INDEX <nombre_de_índice>
ON <nombre_de_tabla> (<col_1>, <col_2>,...)
```

```
-- Crear un índice al mismo tiempo que la tabla:
CREATE TABLE <nombre_de_tabla> (
    (...)
    [UNIQUE] INDEX [<nombre_de_índice>]
        (<col_1>, <col_2>,...)
)
```

```
CREATE INDEX idx_ciudades_nombre  
ON Ciudades (nombre)
```

```
CREATE TABLE Ciudades (  
    id INT NOT NULL auto_increment,  
    nombre VARCHAR(255) NOT NULL,  
    superficie DOUBLE NOT NULL,  
    poblacion LONG NOT NULL,  
    PRIMARY KEY(id),  
    INDEX (nombre)  
)
```

- ▶ Práctica 4:
 - ▶ Ejercicio 4

Parte VIII

Transacciones en SQL

El gestor de bases de datos puede recibir sentencias desde varias conexiones concurrentes, que potencialmente podrían causar inconsistencias en los datos.

Una **transacción** es una secuencia de sentencias SQL que deben ser tratadas como una unidad.

Los gestores de bases de datos relacionales proporcionan el soporte necesario para que transacciones concurrentes cumplan los principios ACID: atomicidad, consistencia, aislamiento y durabilidad.

- ▶ **Atomicidad:** o se ejecutan con éxito todas las sentencias de la transacción, o la base de datos debe volver al estado previo al inicio de la transacción.
- ▶ **Consistencia:** una vez finalizada la transacción, la base de datos debe estar en un estado consistente (se deben cumplir todas las restricciones de consistencia de los datos).
- ▶ **Aislamiento:** durante la ejecución de una transacción, sus cambios no pueden ser visibles para el resto de transacciones.
- ▶ **Durabilidad:** una vez finaliza una transacción con éxito, se debe garantizar que los cambios perduren incluso ante situaciones de fallo en el sistema.

- ▶ Normalmente, cada sentencia SQL individual se ejecuta como una transacción separada.
 - ▶ Se puede deshabilitar de forma temporal para realizar una única transacción con varias sentencias mediante **START TRANSACTION**
 - ▶ Se puede deshabilitar en la sesión actual mediante **SET AUTOCOMMIT=0**
- ▶ Para finalizar la transacción:
 - ▶ Cancelándola: **ROLLBACK**
 - ▶ Confirmándola: **COMMIT**

- ▶ Los gestores de bases de datos relacionales suelen utilizar **cerros** para controlar el acceso concurrente a tablas.
- ▶ Se usan dos tipos de cerros principalmente:
 - ▶ **Cerros S** (*shared*)
 - ▶ **Cerros X** (*exclusive*)

- ▶ La adquisición de un cerrojo S es compatible con otros cerrojos S sobre la misma fila.
- ▶ La adquisición de un cerrojo X no es compatible con ningún otro cerrojo sobre la misma fila.
- ▶ El intento de adquisición de un cerrojo bloquea la operación hasta que sea posible. Si hay interbloqueo con otra transacción, la sentencia falla.

- ▶ En modificaciones y eliminaciones se adquiere X automáticamente, hasta el final de la transacción.
- ▶ Las lecturas se pueden realizar de tres formas:
 - ▶ *Consistent read*: no se adquiere cerrojo (por defecto).
 - ▶ Adquisición de S: **SELECT ... LOCK IN SHARE MODE.**
 - ▶ Adquisición de X: **SELECT ... FOR UPDATE.**

- ▶ *The Language of SQL, Second Edition*, Larry Rockoff. Course Technology PTR (2016).
 - ▶ Accesible en O'Reilly a través de Biblioteca.

- ▶ MySQL Transactional and Locking Statements:

- ▶ <http://dev.mysql.com/doc/refman/8.0/en/sql-syntax-transactions.html>

- ▶ The InnoDB Transaction Model:

- ▶ <http://dev.mysql.com/doc/refman/8.0/en/innodb-transaction-model.html>