

SQL

Durante la clase trabajaremos con el esquema dado por las siguientes tres tablas:

- Capitanes(cid, cnombre, rating, edad);
- Botes(bid, bnombre, bcolor);
- Reserva(cid, bid, fecha)

Y la instancia dada por

Capitanes	cid	cnombre	rating	edad	Reserva	cid	bid	fecha
	22	Pepe	7	45		22	101	10/10/13
	29	Juan	1	33		22	102	10/10/13
	31	Andy	8	55.5		22	103	8/10/13
	32	Ana	8	25.5		22	104	7/10/13
	58	Oscar	10	35		31	102	10/11/13
	64	Javi	7	35		31	103	6/11/13
	71	Pedro	10	16		31	104	12/11/13
	74	Javi	9	35		64	101	5/9/13
	85	Rosa	3	25.5		64	102	8/9/13
	95	Nico	3	63.5		74	103	8/9/13

Botes	bid	bnombre	color
	101	catamaran	azul
	102	catamaran	rojo
	103	laser	verde
	104	yate	rojo

1. Accede a una nueva base de datos en sqlite3
2. Crea las tablas de arriba, recuerda usar tipos de datos que hagan sentido
3. Llena las tablas con los valores de arriba. Es un poco tedioso, pero puedes usar la tecla “flecha hacia arriba” para repetir los comandos que ya has ingresado. Es también una buena práctica para familiarizarte con la interfaz.

1. Primeras Consultas

La forma básica de SQL es la siguiente:

```
SELECT  atributos
FROM    relaciones (pueden ser varias)
WHERE   condiciones - selecciones
```

1. Prueba usando

```
SELECT cid,cnombre,rating,edad
FROM  Capitanes
```

¿Qué ves? También puedes insertar un `*` si lo que quieres es la tabla entera. Prueba reemplazando `cid,cnombre,rating,edad` por `*`, y mira también las otras dos tablas de tu esquema. Revisa que hayas ingresado todos los datos.

2. ¿Qué crees que retorna?

```
SELECT  *
FROM    Capitanes
WHERE   cnombre = 'Juan'
```

3. Busca los capitanes con rating más de 4 (prueba seleccionando `*` y luego solo el atributo `cnombre`). **Ayuda:** Puedes usar $x = y$, $x < y$, $x \leq y$, $<>$, etc, para los números y fechas el significado es obvio, para los strings es orden lexicográfico.
4. Similarmente, busca los capitanes con rating más de 4 y cuya edad está entre 25 y 40 años. **Ayuda:** Combina condiciones en el `WHERE` usando `AND`, `OR`, `NOT`
5. Busca la edad de los capitanes que no tienen una 'a' en su nombre (recuerda usar `LIKE`)
6. Escribe una consulta que entregue los nombres y rating de los capitanes, pero ordenado por rating. Para ordenar agrega un `ORDER BY (<atri1>, . . . , <atriN>) DESC` al final de tu consulta. Eso ordena los resultados por el atributo `<atri1>`, después `<atri2>`, etc. Prueba cambiando `DESC` por `ASC`.

1.1. Joins, uniones, intersecciones, diferencia

Imaginemos que queremos encontrar los nombres de los capitanes que reservaron el bote 103.

1. Escribe primero esta consulta en Álgebra relacional.

Una forma directa para traspasarlo a SQL es:

```
SELECT  cnombre
FROM    Capitanes INNER JOIN Reservas
ON      Capitanes.cid = Reservas.cid
WHERE   Reservas.bid = 103
```

Ayuda: Este es un buen minuto para repasar los joins. Puedes correr la consulta de arriba sin el WHERE para ver lo que sería el resultado de `Capitanes ⋈cid=cid Reservas`. ¿Qué pasa al agregar la condición del WHERE?

Sin embargo, una forma más clara (o que a nosotros nos gusta más) es explotando el producto cartesiano:

```
SELECT  *
FROM    Capitanes, Reservas
WHERE   Capitanes.cid = Reservas.cid
AND     Reservas.bid = 103
```

En este caso estamos aplicando el WHERE al Producto Cartesiano, y por eso incluimos la condición `WHERE Capitanes.cid = Reservas.cid`. ¿Entiendes por qué ambas consultas son equivalentes?

Intenta escribir más consultas con joins.

2. Los nombres de los capitanes que reservaron un bote rojo
3. Los nombres de los capitanes que reservaron un bote rojo o un bote verde
4. La misma anterior, pero usando UNION para unir el resultado de dos consultas.
5. Prueba usando UNION y luego UNION ALL. ¿Cuál es la diferencia?
6. Usa AS para cambiar de nombres a los atributos (esto es útil no solo para la unión!). No puedes ver el resultado en sqllite, por que no muestra los nombres de los atributos, pero muchos sistemas si los muestran. La siguiente entrega todos los id's del sistema:

```
SELECT cid AS id
FROM    Capitanes
UNION
SELECT bid AS id
FROM    Botes
```

7. Los nombres de los capitanes que reservaron un bote rojo y un bote verde. Puedes usar INTERSECT para intersectar el resultado de dos consultas. ¿Puedes escribir también esta consulta usando AND?

2. Sub consultas, consultas anidadas

Has observado que cada consulta entrega una tabla, los atributos de esa tabla son lo que pones en el SELECT. Esto significa que podemos poner cualquier consulta SQL en el FROM.

1. Escribe una consulta que entregue los nombres de los capitanes junto a los bid de los botes que han reservado, pero contando solo las reservas hechas después del primero de octubre del 2013.
2. Sea Q la consulta anterior. Qué retorna:

```
SELECT bid,bnombre,bcolor
FROM Botes, (Q) AS Res2015
WHERE Botes.bid = Res2015.bid
```

3. Ahora modifica Q para que solo retorne los bid de los botes, y no los nombres de los capitanes. Qué retorna:

```
SELECT bid,bnombre,bcolor
FROM Botes
WHERE Botes.bid IN Q
```

4. Prueba que pasa si Q entrega dos atributos en vez de uno (es decir, usando Q sin modificar). ¿Por qué pasa esto?

También podemos usar consultas anidadas en el WHERE. Además del IN podemos usar NOT IN, o >ANY, >ALL para cualquier comparación entre <,>,<=,>=,=, y también anteponiendo NOT.

Por ejemplo, si queremos encontrar los capitanes que no han reservado un bote verde:

- Primero escribe una consulta que entregue los cid de los capitanes que han reservado un bote verde
- Sea Q la consulta anterior. Los capitanes que no han reservado un bote verde se pueden extraer remplazando Q en:

```
SELECT cnombre
FROM capitanes
WHERE cid NOT IN (Q)
```

1. Encuentre los nombres de los capitanes que tienen más de 20 años y que no han reservado un bote rojo. ¿Puedes escribir esto sin usar consultas anidadas?
2. Encuentre los nombres de los capitanes que reservaron al menos dos botes.
3. Encuentre los nombres de los capitanes que han reservado todos los botes. ¿Puedes escribir esto sin usar consultas anidadas?

3. Agregación

¿Qué pasa si queremos el capitán con el mejor rating?

- Primero debemos sacar el mejor rating:

```
SELECT MAX(rating)
FROM Capitanes
```

- Ahora seleccionamos los nombres de los capitanes con ese rating:

```
SELECT cnombre
FROM Capitanes
WHERE rating = (SELECT MAX(rating)
                FROM Capitanes)
```

En general SQL soporta varias funciones de agregación: MAX, MIN (mínimo), AVG (promedio), SUM (suma).

1. Encuentra el rating promedio de los capitanes mayores a 45 años
2. Encuentra los capitanes cuya edad es menor que el promedio
3. Encuentra los botes que no han sido reservados por el capitán con peor rating

También podemos usar COUNT para contar tuplas. Hay dos versiones (pruébalas!):

- `SELECT COUNT(*) FROM Tabla` cuenta cuantas tuplas tiene la tabla
 - `SELECT COUNT(DISTINCT cnombre) FROM Capitanes` cuenta cuantos nombres hay en Capitanes
 - En general `SELECT COUNT(*) FROM Tabla` y `SELECT COUNT(atributo) FROM Tabla` es lo mismo, pero hay casos en los que es distinto: cuando las tablas tienen valores nulos o indefinidos. ¡Más de esto en Abril!
4. Encuentra cuantos capitanes menores de 40 años hay
 5. Cuenta la cantidad de reservas hechas a botes de color rojo
 6. Cuenta cuantos capitanes tienen rating mayor al promedio y luego cuantos capitanes tienen rating menor al promedio

Finalmente podemos usar GROUP BY para ir agrupando la base de datos en pedazos. Prueba poniendo:

```
SELECT cnombre, SUM(edad)
FROM Capitanes
GROUP BY cnombre
```

¿Qué pasó con Javi? Lo que hacemos es agrupar a las dos tuplas con nombre Javi en una, y sumamos la edad. Prueba ahora:

```
SELECT color, count(color)
FROM Botes
GROUP BY color
```

7. Encuentra para los Capitanes el promedio del ranking por cada edad distinta
8. Encuentra el número de reservas por cada color de bote
9. Encuentra el numero de reservas por capitán y por color de bote

4. Insertando y Removiendo Tuplas

En vez de poner valores explícitos, podemos insertar datos de una consulta.

1. Crea una tabla CapitanesTop(cid,cnombre)
2. Llénala con

```
INSERT INTO CapitanesTop
SELECT cid,cnombre
FROM Capitanes
WHERE Rating > 8
```

¿Qué queda?

3. Si solo quieres quedarte con los que tiene rating ≥ 9 :

```
DELETE FROM CapitanesTop
WHERE cid IN
    (SELECT cid
     FROM Capitanes
     WHERE Rating < 9)
```