

## **2.- Lenguaje de consulta SQL**

SQL es un lenguaje de consultas hacia bases de datos relacionales. Entre las características de SQL se tiene:

- Procesa conjuntos de datos como grupos, más que como unidades individuales.
- Provee navegación automática sobre los datos.
- Usa instrucciones que son complejas y poderosas individualmente. Las instrucciones de flujo de control de los otros lenguajes de programación están ausentes en SQL. Sin embargo ellas se proveen por extensiones al SQL, dependientes de cada fabricante de bases de datos.

SQL provee comandos para una variedad de tareas incluyendo:

- consulta sobre los datos
- inserción, actualización y borrado de registros en una tabla.
- creación, reemplazo, alteración y borrado de objetos.
- control de acceso a los objetos en la base de datos.
- garantiza consistencia e integridad en la base de datos.

### **2.1.- Recuperación de datos (select)**

Es usado para recuperar datos desde la base de datos.

sintaxis:   select   [lista\_de\_campos]  
              from     [tabla]

Ejemplos:

1) Listar todos los datos almacenados en la tabla Tiendas

SQL:               select \* from tiendas

	<b>id_tienda</b>	<b>nombre_tienda</b>	<b>direcc_tienda</b>	<b>ciudad</b>	<b>estado</b>	<b>pais</b>	<b>cod_postal</b>	<b>terminos</b>
1	7066	Barnum's	567 Pasadena Ave.	Tustin	CA	USA	92789	Net 30
2	7067	News & Brews	577 First St.	Los Gatos	CA	USA	96745	Net 30
3	7131	Doc-U-Mat: Quality Laundry and Books	24-A Avrogado Way	Remulade	WA	USA	98014	Net 60
4	8042	Bookbeat	679 Carson St.	Portland	OR	USA	89076	Net 30
5	6380	Eric the Read Books	788 Catamagus Ave.	Seattle	WA	USA	98056	Net 60
6	7896	Fricative Bookshop	89 Madison St.	Fremont	CA	USA	90019	Net 60
7	5023	Thoreau Reading Discount Chain	20435 Walden Expressway	Concord	MA	USA	1776	Net 60

- 2) Lista únicamente el nombre de las tiendas de la tabla Tiendas

SQL:       select nombre\_tienda from tiendas

	<b>nombre_tienda</b>
1	Barnum's
2	News & Brews
3	Doc-U-Mat: Quality Laundry and Books
4	Bookbeat
5	Eric the Read Books
6	Fricative Bookshop
7	Thoreau Reading Discount Chain

- 3) Lista la ciudad, nombre y estado de las tiendas (notar que la posición de los campos, no concuerda con los de la tabla)

SQL:       select ciudad, nombre\_tienda, estado from tiendas

	<b>ciudad</b>	<b>nombre_tienda</b>	<b>estado</b>
1	Tustin	Barnum's	CA
2	Los Gatos	News & Brews	CA
3	Remulade	Doc-U-Mat: Quality Laundry and Books	WA
4	Portland	Bookbeat	OR
5	Seattle	Eric the Read Books	WA
6	Fremont	Fricative Bookshop	CA
7	Concord	Thoreau Reading Discount Chain	MA

## **2.2.- Eliminación de registros repetidos**

La cláusula DISTINCT elimina filas duplicadas en el despliegue de los datos.

sintaxis:   select   distinct [lista\_de\_campos]  
              from     [tabla]

Ejemplos:

- 1) Liste los estados que están en la tabla Tiendas

SQL:       select estado from tiendas

	<b>estado</b>
1	CA
2	CA

3	WA
4	OR
5	WA
6	CA
7	MA

2) Liste una sola ocurrencia de los estados de la tabla Tiendas

SQL:                select distinct estado from tiendas

estado	
1	CA
2	MA
3	OR
4	WA

### **2.3.- Recuperación condicionada.**

Cláusula WHERE es usada para determinar exactamente que filas recuperar.

sintaxis:    select    [lista\_de\_campos]  
                  from    [tabla]  
                  where    [condición\_busqueda]

Los operadores de comparación y búsqueda son:

Operador	Significado
=	Igual a
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que
!=	distinto (no igual)
Between	rangos
in	selecciona de una lista

Observaciones:

- "<>" es equivalente a "!="
- Para las variables tipo texto, usar comillas simples como delimitador.
- La cláusula BETWEEN es usada para especificar un rango inclusive (los bordes están considerados en la búsqueda)

Ejemplos:

1) Encontrar las tiendas localizadas en California

SQL:       select   nombre\_tienda, ciudad, estado  
              from    tiendas  
              where   estado = 'CA'

	nombre_tienda	ciudad	estado
1	Barnum's	Tustin	CA
2	News & Brews	Los Gatos	CA
3	Fricative Bookshop	Fremont	CA

Ejercicios:

- a) Encontrar las tiendas que no están localizadas en California ('CA').
- b) Encuentre el código y el nombre de las tiendas que tiene código mayor que '7066'

2) Encuentre aquellas tiendas cuyos códigos estén entre '7067' y '8000'

SQL:       select   id\_tienda, nombre\_tienda, ciudad, estado  
              from    tiendas  
              where   id\_tienda between '7067' and '8000'

	id_tienda	nombre_tienda	ciudad	estado
1	7067	News & Brews	Los Gatos	CA
2	7131	Doc-U-Mat: Quality Laundry and Books	Remulade	WA
3	7896	Fricative Bookshop	Fremont	CA

3) Encuentre aquellas tiendas cuyos códigos están bajo '7067' o sobre '8000'

SQL:       select   id\_tienda, nombre\_tienda, ciudad, estado  
              from    tiendas  
              where   id\_tienda not between '7067' and '8000'

	id_tienda	nombre_tienda	ciudad	estado
1	7066	Barnum's	Tustin	CA
2	8042	Bookbeat	Portland	OR
3	6380	Eric the Read Books	Seattle	WA
4	5023	Thoreau Reading Discount Chain	Concord	MA

4) Encuentre las tiendas en California ('CA') y Washington ('WA')

SQL:       select   nombre\_tienda , ciudad, estado  
              from    tiendas  
              where   estado in ('CA','WA')

	nombre_tienda	ciudad	estado
1	Barnum's	Tustin	CA
2	News & Brews	Los Gatos	CA
3	Doc-U-Mat: Quality Laundry and Books	Remulade	WA
4	Eric the Read Books	Seattle	WA
5	Fricative Bookshop	Fremont	CA

## 2.4.- Búsquedas con patrones de caracteres.

Existe la cláusula LIKE, para búsqueda de patrones en textos. El símbolo "%" indica una secuencia de cero o más caracteres. El símbolo "\_" indica un único carácter.

Ejemplos:

- 1) Encuentre las tiendas cuyo nombre comienza con "B" (nota: se distingue entre mayúsculas y minúsculas)

SQL:           select   nombre\_tienda  
                  from    tiendas  
                  where   nombre\_tienda like 'B%'

	nombre_tienda
1	Barnum's
2	Bookbeat

- 2) Encuentre aquellas tiendas que no comienzan con 'B'

SQL:           select   nombre\_tienda  
                  from    tiendas  
                  where   nombre\_tienda not like 'B%'

	nombre_tienda
1	News & Brews
2	Doc-U-Mat: Quality Laundry and Books
3	Eric the Read Books
4	Fricative Bookshop
5	Thoreau Reading Discount Chain

- 3) Encuentre aquellas tiendas que tienen un código de 4 componentes (dígitos) de los cuales los primeros dos son 70

SQL:           select   id\_tienda, nombre\_tienda  
                  from    tiendas  
                  where   id\_tienda like '70\_\_'

	<b>id_tienda</b>	<b>nombre_tienda</b>
1	7066	Barnum's
2	7067	News & Brews

## **2.5.- Búsquedas con condiciones lógicas**

Para conectar condiciones usar los operadores lógicos AND , OR y NOT

Ejemplos:

- 1) Encontrar aquellos libros que están en la categoría (tipo) de "business", además de aquellos libros en la categoría de "psychology", los cuales tengan un avance de sobre \$5500.

```
SQL:      select  id_titulo, tipo, avance
           from    titulos
           where    tipo = 'business' or tipo='psychology'
           and      avance > 5500
```

	<b>id_titulo</b>	<b>tipo</b>	<b>avance</b>
1	BU1032	business	5000
2	BU1111	business	5000
3	BU2075	business	10125
4	BU7832	business	5000
5	PS1372	psychology	7000
6	PS2106	psychology	6000

¿Que paso? ¿Porque trajo el registro 1,2,4, si tienen avances menores a lo solicitado?

- 2) Encontrar aquellos libros en la categoría "business" y "psychology" que tengan un avance de sobre \$5500

```
SQL:      select  id_titulo, tipo, avance
           from    titulos
           where    (tipo = 'business' or tipo='psychology')
           and      avance > 5500
```

	<b>id_titulo</b>	<b>tipo</b>	<b>avance</b>
1	BU2075	business	10125
2	PS1372	psychology	7000
3	PS2106	psychology	6000

## **2.6.- Ejercicios Propuestos**

- 1) Que almacenes están localizados en el estado de Washington.
- 2) Liste los tipos de libros disponibles.
- 3) Liste todos los títulos existentes y la fecha de publicación.
- 4) Liste los libros con un precio mayor a \$10
- 5) Liste todos los tipos de libros de "business"
- 6) Liste todos los libros que tengan la palabra "Computer" en su título.
- 7) Encuentre un libro de "psychology" que cueste mas de \$20
- 8) Encuentre aquellos libros que tiene un precio entre \$10 y \$20
- 9) Encuentre las categorías de libros que el publicador '0877' publica.
- 10) Encuentre el título del libro que está en la categoría "psychology", que cueste mas de \$20 y que tiene la palabra "Computer" en el título.
- 11) Encuentre aquellos libros que tienen un avance superior a \$5000 y han vendido menos de 4000 copias.
- 12) Cuales son las categorías de libros que han vendido 4000 a 5000 copias en total.
- 13) Cuales libros han vendido mas de 1000 copias y cuestan menos de \$10 en la categoría de "business" o en la categoría de cocina moderna ("mod\_cook")

## **2.7.- Renombrando columnas**

Se permite usar un nombre alternativo para desplegar en el resultado de la consulta, con sólo indicarlo después del campo.

```
SQL:      select  id_autor  Identificacion_del_autor,
                apellido  Apellido_Paterno
          from    autores
          where   estado='CA'
```

	<b>Identificacion_del_autor</b>	<b>Apellido_Paterno</b>
1	172-32-1176	White
2	213-46-8915	Green
3	238-95-7766	Carson
4	267-41-2394	O'Leary
5	274-80-9391	Straight
6	409-56-7008	Bennet
7	427-17-2319	Dull
8	472-27-2349	Gringlesby
9	486-29-1786	Locksley
10	672-71-3249	Yokomoto
11	724-08-9931	Stringer

12	724-80-9391	MacFeather
13	756-30-7391	Karsen
14	846-92-7186	Hunter
15	893-72-1158	McBadden

## **2.8.- Expresiones numéricas**

Operadores aritméticos ('+', '-', '\*', '/') se pueden usar en cualquier columna numérica o cláusula que permita una expresión.

Ejemplos:

- 1) 

```
select avance, precio, avance+precio, id_titulo
from titulos
where tipo = 'business'
```

	<b>avance</b>	<b>precio</b>	<b>avance+precio</b>	<b>id_titulo</b>
1	5000	19,99	5019,99	BU1032
2	5000	11,95	5011,95	BU1111
3	10125	2,99	10127,99	BU2075
4	5000	19,99	5019,99	BU7832

- 2) 

```
select titulo, precio* total_ventas ingresos
from titulos
where precio*total_ventas > 80000
```

	<b>titulo</b>	<b>ingresos</b>
1	The Busy Executive's Database Guide	81859,05
2	Straight Talk About Computers	81859,05
3	But Is It User Friendly?	201501
4	Secrets of Silicon Valley	81900
5	Prolonged Data Deprivation: Four Case Studies	81399,28
6	Fifty Years in Buckingham Palace Kitchens	180397,2

- 3) 

```
select titulo, precio* total_ventas ingresos
from titulos
where precio*total_ventas > 80000
and tipo='business'
```

	<b>titulo</b>	<b>ingresos</b>
1	The Busy Executive's Database Guide	81859,05
2	Straight Talk About Computers	81859,05



## **2.9.- Valores nulos (NULL)**

Un valor Null (nulo) implica un valor desconocido. Un valor Null, no significa Cero o blanco; no hay ningun valor explicitamente asignado.

Se puede usar "IS NULL" ó "= NULL" para seleccionar columnas teniendo nulos.

Algunas columnas se definen para permitir o no permitir valores Null (llave primaria)

Ejemplos.

- 3) `select titulo, precio`  
`from titulos`  
`where tipo = 'popular_comp'`  
`and precio = null` (Obs: equivalente a: "and precio is null")

	<b>titulo</b>	<b>precio</b>
1	Net Etiquette	

- 4) `select id_titulo, avance`  
`from titulos`  
`where avance < 5000`

	<b>id_titulo</b>	<b>avance</b>
1	MC2222	0
2	PS2091	2275
3	PS3333	2000
4	PS7777	4000
5	TC4203	4000

`select id_titulo, avance`  
`from titulos`  
`where avance < 5000 or avance is null`

	<b>id_titulo</b>	<b>avance</b>
1	MC2222	0
2	MC3026	
3	PC9999	

4	PS2091	2275
5	PS3333	2000
6	PS7777	4000
7	TC4203	4000

- 5) Las operaciones (ej, cálculos aritméticos) involucrando nulos dan como resultado nulos

```
select id_titulo, avance, precio, avance/precio
from    titulos
where   tipo not in ('business','psychology','trad_cook')
```

	id_titulo	avance	precio	avance/precio
1	MC2222	0	19,99	0
2	MC3021	15000	2,99	5016,72241
3	MC3026			
4	PC1035	7000	22,95	305,010893
5	PC8888	8000	20	400
6	PC9999			

## **2.10.- Ordenando los resultados**

La cláusula ORDER BY ordena los resultados (en orden ascendente por omisión). Los campos nombrados en la cláusula ORDER BY, no necesitan estar en la lista del select. Cuando se usa ORDER BY, los valores nulos son listados primero.

```
sintaxis:  select  [lista_de_campos]
            from    [tabla]
            where   [condición_busqueda]
            order by [columna | expresion] [ASC | DESC ], .....
```

Ejemplos:

- 1) select nombre\_tienda, estado  
from tiendas  
order by estado, nombre\_tienda

	nombre_tienda	estado
1	Barnum's	CA
2	Fricative Bookshop	CA
3	News & Brews	CA
4	Thoreau Reading Discount Chain	MA
5	Bookbeat	OR

6	Doc-U-Mat: Quality Laundry and Books	WA
7	Eric the Read Books	WA

- 2) select estado, nombre\_tienda  
from tiendas  
order by estado, nombre\_tienda desc (¿en "estado" asume asc o desc? )

	estado	nombre_tienda
1	CA	News & Brews
2	CA	Fricative Bookshop
3	CA	Barnum's
4	MA	Thoreau Reading Discount Chain
5	OR	Bookbeat
6	WA	Eric the Read Books
7	WA	Doc-U-Mat: Quality Laundry and Books

- 3) Projete cuantos libros de sicología en promedio necesitan ser vendidos cada mes del proximo año, asumiendo un 10% de incremento en ventas.

SQL: select ((total\_ventas/12)\*1.10),id\_titulo  
from titulos  
where tipo = 'psychology'  
order by 1

	((total_ventas/12)*1,10)	id_titulo
1	10,175	PS2106
2	34,375	PS1372
3	187,4583333	PS2091
4	305,8	PS7777
5	373,2666667	PS3333

- 4) select id\_titulo, tipo, precio (ojo: con los nulos)  
from titulos  
where tipo= 'popular\_comp'  
order by precio

	id_titulo	tipo	precio
1	PC9999	popular_comp	
2	PC8888	popular_comp	20
3	PC1035	popular_comp	22,95

## 2.11.-Funciones de totalización.

Funcion	Valor calculado
SUM	total
AVG	valor promedio
MIN	valor mínimo
MAX	valor máximo
COUNT(*)	número de registros

- Las funciones anteriores ignoran los valores nulos (excepto count(\*))
- SUM y AVG trabajan sólo con campos numéricos
- Sólo una fila es retornada para estas funciones (si no se usa cláusula GROUP BY)
- No pueden ser usadas en cláusula WHERE
- Pueden ser aplicadas a toda la tabla o un subconjunto de ellas.

Ejemplos:

1) select count(\*) total from titulos

	<b>total</b>
1	8

select count(avances) total from titulos

	<b>total</b>
1	16

2) select   max(precio) maximo,  
              min(precio) minimo,  
              avg(precio) promedio,  
              sum(precio) total  
from titulos

	<b>maximo</b>	<b>minimo</b>	<b>promedio</b>	<b>total</b>
1	22,95	2,99	14,76625	236,26

## **2.12.- Ejercicios propuestos.**

- 1) Cuantos libros están disponibles en la base de datos.
- 2) Cuantos publicadores han publicado libros.
- 3) Cuantos libros en la categoría "business" han sido vendidos.

### 2.13.- Resultados Agrupados (group by)

- La cláusula GROUP BY divide los datos en grupos.
- Usualmente es usada con funciones de totalización en la lista del select.
- Todos los valores nulos, son tratados como un grupo más.
- La cláusula WHERE elimina las filas antes de agruparlas.
- La cláusula HAVING aplica condiciones a los grupos.

sintaxis:   select   [lista\_de\_campos]  
               from    [tabla]  
               where   [condición\_búsqueda]  
               group by [campos]  
               order by [columna | expresion] [ASC | DESC ], .....

Ejemplos:

1) select tipo, avg(precio)               (¿que tiene de malo?)  
     from titulos

select tipo, avg(precio) promedio  
     from titulos  
     group by tipo

	tipo	promedio
1	business	13,73
2	mod_cook	11,49
3	popular_comp	21,475
4	psychology	13,504
5	trad_cook	15,96333333
6	UNDECIDED	

2) select tipo, avg(precio) promedio  
     from titulos  
     group by tipo  
     order by avg(precio)

	tipo	promedio
1	UNDECIDED	
2	mod_cook	11,49
3	psychology	13,504
4	business	13,73
5	trad_cook	15,96333333
6	popular_comp	21,475

3) Cuantos libros han sido vendidos por cada título.

SQL:       select id\_titulo, sum(cantidad) total  
              from detalle\_venta  
              group by id\_titulo

	<b>id_titulo</b>	<b>total</b>
1	BU1032	4095
2	BU1111	3876
3	BU2075	15722
4	BU7832	4095
5	MC2222	2032
6	MC3021	22246
7	PC1035	8780
8	PC8888	3795
9	PS1372	375
10	PS2091	2045
11	PS2106	111
12	PS3333	4072
13	PS7777	3336
14	TC3218	375
15	TC4203	15096
16	TC7777	4095

4) cuantos libros han sido vendidos por cada tienda.

SQL:       select id\_tienda, sum(cantidad) total  
              from detalle\_venta  
              group by id\_tienda

	<b>id_tienda</b>	<b>total</b>
1	5023	79374
2	6380	2430
3	7066	2430
4	7067	2900
5	7131	2097
6	7896	1182
7	8042	3733

¿Cómo obtenemos el nombre de la Tienda que esta en otra tabla?

5) Cuantos libros han sido vendidos que tengan un descuento mayor a 50%

SQL:       select id\_titulo, sum(cantidad) total  
              from detalle\_venta  
              where descuento > 50       (obs: saca los registros antes de agrupar)  
              group by id\_titulo

	<b>id_titulo</b>	<b>total</b>
1	BU1032	875
2	BU1111	390
3	BU2075	4810
4	BU7832	300
5	MC3021	1527
6	PC1035	533
7	PS7777	180
8	TC4203	3856

- 6) `select tipo, avg(precio) promedio`  
`from titulos`  
`where tipo in ('business','popular_comp')`  
`group by tipo`

	<b>tipo</b>	<b>promedio</b>
1	business	13,73
2	popular_comp	21,475

- 7) Cuantos libros han sido vendidos de los títulos mas vendido (>10.000)

`select id_titulo, sum(cantidad) total`  
`from detalle_venta`  
`group by id_titulo`

	<b>id_titulo</b>	<b>total</b>
1	BU1032	4095
2	BU1111	3876
3	BU2075	15722
4	BU7832	4095
5	MC2222	2032
6	MC3021	22246
7	PC1035	8780
8	PC8888	3795
9	PS1372	375
10	PS2091	2045
11	PS2106	111
12	PS3333	4072
13	PS7777	3336
14	TC3218	375
15	TC4203	15096
16	TC7777	4095

```
select id_titulo, sum(cantidad) total
from detalle_venta
group by id_titulo
having sum(cantidad) > 10000
```

	id_titulo	total
1	BU2075	15722
2	MC3021	22246
3	TC4203	15096

## **2.14.- Ejercicio propuestos.**

- 1) Cuantos de los libros mas vendidos tienen un descuento mayor al 50%
- 2) Cual es el precio promedio de un libro por tipo y por publicador.
- 3) Cual es el costo del libro mas barato y más caro de cada categoría.
- 4) Cuantos libros de cada categoría ha vendido cada publicador
- 5) Para publicadores teniendo libros a menos de \$10, calcule el precio promedio de todos sus libros.
- 6) Cuantos libros hay en cada categoría

## **2.15.- Operaciones entre mas de una tabla (join)**

La operación "join" recupera datos de mas de una tabla. Se tienen las siguientes observaciones:

- SELECT. Las columnas deben ser precedidas por el nombre de la tabla, porque si en mas una tabla de las usadas, hay coincidencia en el nombre de las columnas, se produce ambigüedad.
- FROM. Al indicar tablas en la sentencia "from", se asume que el join es deseado (ojo, con los productos cruz)
- WHERE. Las columnas son comparadas; ellas deben ser del mismo tipo de datos.
- Los valores nulos nunca son usados para el "join"
- Las columnas en la condición de join, no necesitan estar en la cláusula select.

Ejemplos:

- 1) select nombre\_pub, publicadores.id\_pub, titulos.id\_titulo  
from publicadores, titulos  
where publicadores.id\_pub = titulos.id\_pub (join basado en igualdad )

	nombre_pub	id_pub	id_titulo
1	Algodata Infosystems	1389	BU1032
2	Algodata Infosystems	1389	BU1111



3	New Age Books	736	BU2075
4	Algodata Infosystems	1389	BU7832
5	Binnet & Hardley	877	MC2222
6	Binnet & Hardley	877	MC3021
7	Binnet & Hardley	877	MC3026
8	Algodata Infosystems	1389	PC1035
9	Algodata Infosystems	1389	PC8888
10	Algodata Infosystems	1389	PC9999
11	Binnet & Hardley	877	PS1372
12	New Age Books	736	PS2091
13	New Age Books	736	PS2106
14	New Age Books	736	PS3333
15	New Age Books	736	PS7777
16	Binnet & Hardley	877	TC3218
17	Binnet & Hardley	877	TC4203
18	Binnet & Hardley	877	TC7777

2) Cuantos libros han sido vendidos por cada tienda.

SQL:      select id\_tienda, sum(cantidad) total  
              from detalle\_venta  
              group by id\_tienda  
              order by sum(cantidad)

	<b>id_tienda</b>	<b>total</b>
1	7896	1182
2	7131	2097
3	7066	2430
4	6380	2430
5	7067	2900
6	8042	3733
7	5023	79374

¿Cómo obtenemos el nombre de la Tienda que esta en otra tabla?

SQL:      select nombre\_tienda, sum(cantidad) total  
              from detalle\_venta, tiendas  
              where tiendas.id\_tienda = id\_tienda                      (join basado en igualdad)  
              group by nombre\_tienda  
              order by sum(cantidad)

	<b>nombre_tienda</b>	<b>total</b>
1	Fricative Bookshop	1182
2	Doc-U-Mat: Quality Laundry and Books	2097
3	Eric the Read Books	2430
4	Barnum's	2430
5	News & Brews	2900

6	Bookbeat	3733
7	Thoreau Reading Discount Chain	79374

- 3) Cual es el total de ventas para cada uno de los libros vendidos.

```
select titulos.id_titulo, cantidad, precio, precio*cantidad total_precio
from titulos, detalle_venta
where titulos.id_titulo = detalle_venta.id_titulo
order by precio*cantidad
```

	id_titulo	cantidad	precio	total_precio
1	BU2075	30	2,99	89,7
2	BU2075	35	2,99	104,65
3	BU2075	42	2,99	125,58
4	MC3021	69	2,99	206,31
5	PS2106	30	7	210
6	PS2106	31	7	217
	.....	.....	.....	.....
109	MC2222	2032	19,99	40619,68
110	TC4203	3500	11,95	41825
111	PC1035	2000	22,95	45900
112	PC1035	2000	22,95	45900
113	PC1035	2138	22,95	49067,1
114	PS3333	2687	19,99	53713,13

- 4) Cuales tiendas han ordenado menos de 70 copias de un libro

```
select tiendas.id_tienda, cantidad, id_titulo, nombre_tienda
from detalle_venta, tiendas
where detalle_venta.id_tienda = tiendas.id_tienda
and cantidad < 70
```

	id_tienda	cantidad	id_titulo	nombre_tienda
1	7131	50	TC3218	Doc-U-Mat: Quality Laundry and Books
2	8042	40	TC3218	Bookbeat
3	8042	30	PS2106	Bookbeat
4	8042	50	PS2106	Bookbeat
5	8042	31	PS2106	Bookbeat
6	8042	69	MC3021	Bookbeat
7	7896	42	BU2075	Fricative Bookshop
8	7896	25	PC1035	Fricative Bookshop
9	7131	35	BU2075	Doc-U-Mat: Quality Laundry and Books
10	7067	34	PC1035	News & Brews
11	7067	53	TC4203	News & Brews
12	8042	30	BU2075	Bookbeat

5) Que autores viven en la misma ciudad que Livia Karsen

```
select  a1.apellido, a1.nombre, a1.ciudad
from    autores a1, autores a2          (ojo, que es la misma tabla)
where   a2.apellido='Karsen'
and     a2.nombre='Livia'
and     a1.ciudad=a2.ciudad
```

	apellido	nombre	ciudad
1	Stringer	Dirk	Oakland
2	MacFeather	Stearns	Oakland
3	Karsen	Livia	Oakland
4	Straight	Dick	Oakland
5	Green	Marjorie	Oakland

6) Despliegue para cada Tienda, los libros vendidos

```
select  nombre_tienda, titulo
from    tiendas, detalle_venta, titulos
where   tiendas.id_tienda = detalle_venta.id_tienda
and     detalle_venta.id_titulo = titulos.id_titulo
```

	nombre_tienda	titulo
1	Fricative Bookshop	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean
2	Fricative Bookshop	Sushi, Anyone?
3	Doc-U-Mat: Quality Laundry and Books	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean
4	Doc-U-Mat: Quality Laundry and Books	Sushi, Anyone?
5	Thoreau Reading Discount Chain	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean
6	Bookbeat	Prolonged Data Deprivation: Four Case Studies
7	Bookbeat	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean
8	Bookbeat	Life Without Fear
9	Bookbeat	Life Without Fear
10	Bookbeat	Life Without Fear
11	Bookbeat	The Gourmet Microwave
12	Thoreau Reading Discount Chain	But Is It User Friendly?
13	Thoreau Reading Discount Chain	You Can Combat Computer Stress!
14	Thoreau Reading Discount Chain	The Busy Executive's Database Guide
15	Thoreau Reading Discount Chain	Straight Talk About Computers
16	Thoreau Reading Discount Chain	Emotional Security: A New Algorithm
17	Thoreau Reading Discount Chain	Sushi, Anyone?
18	Thoreau Reading Discount Chain	The Busy Executive's Database Guide
19	Thoreau Reading Discount Chain	But Is It User Friendly?
	.....	.....
107	Eric the Read Books	The Gourmet Microwave
108	Eric the Read Books	Prolonged Data Deprivation: Four Case Studies
109	Eric the Read Books	Emotional Security: A New Algorithm
110	Eric the Read Books	Onions, Leeks, and Garlic: Cooking Secrets of the Mediterranean

111	Eric the Read Books	You Can Combat Computer Stress!
112	Eric the Read Books	The Busy Executive's Database Guide
113	Eric the Read Books	Fifty Years in Buckingham Palace Kitchens
114	Eric the Read Books	The Gourmet Microwave

## **2.16.- Ejercicios propuestos.**

- 1) Despliegue el nombre y precio de los libros tienen el mismo valor.
- 2) Cual es el nombre del autor que escribió el libro con id\_titulo = 'MC2222'
- 3) Que descuentos han sido dados a la tienda 'Eric the Read Books'
- 4) Que autores viven en la misma ciudad que un publicador.
- 5) Cuales autores viven en la misma ciudad (Cuales ciudades tiene mas de un autor viviendo ahí)
- 6) Cuales libros fueron publicados el mismo día.
- 7) Cuantos libros ha vendido cada autor.
- 8) Despliegue los libros con mas de un autor.
- 9) Un cliente llamó ordenando un libro, solo conoce que es un libro de sicología ("psychology") escrito por alguien llamado Ann ó Anne. ¿Cuál es el nombre completo de la autora y el título del libro?

## **2.17.- Subquery**

Los subquery o consultas anidadas, tiene relación con la posibilidad de ocupar cláusulas “select” dentro de otros select. Estas consultas anidadas, pueden tener uno o más select en su interior.

- Un subquery es una instrucción “select”, usada como expresión, como parte de otra instrucción select, update, insert o delete.
- El subquery (select anidado) es ejecutado y los resultados sustituidos dentro del query que lo invoco.
- Si la cláusula where de la consulta “padre” incluye un nombre de columna, esta debe ser del mismo tipo que la retornada por el subquery, es decir, deben ser del mismo dominio.
- Los “select” anidados, no pueden tener la cláusula “order by” en su sintaxis.
- La lista de campos a retornar por el subquery, sólo debe ser uno (excepto para la clausula “exist” que no se tratan en estos apuntes)
- La cláusula distinct no puede ser usada por subqueries que incluyen un cláusula “group by”
- El subquery, puede ir en la cláusula WHERE o HAVING.
- subqueries usados con IN o NOT IN pueden retornar cero o mas valores (múltiples filas)
- subqueries usados con operadores de comparación (=,>,<, ...) deben retornar sólo un valor, sino ocurrirá un error.

Ejemplos.

1.- Que titulos de libros son publicados por el publicador “New Age Book”

a) Usando dos “select” separados.

```
select id_pub
from publicadores
where nombre_pub = 'New Age Books'
```

	<b>id_pub</b>
1	736

```
select titulo
from titulos
where id_pub = '0736'
```

	<b>titulo</b>
1	You Can Combat Computer Stress!
2	Is Anger the Enemy?
3	Life Without Fear
4	Prolonged Data Deprivation: Four Case Studies
5	Emotional Security: A New Algorithm

b) Usando un “select” anidado

```
select titulo
from titulos
where id_pub = ( select id_pub
                  from publicadores
                  where nombre_pub='New Age Books'
                )
```

	<b>titulo</b>
1	You Can Combat Computer Stress!
2	Is Anger the Enemy?
3	Life Without Fear
4	Prolonged Data Deprivation: Four Case Studies
5	Emotional Security: A New Algorithm

c) opcionalmente se puede resolver con un join

```
select titulo
from titulos
where titulos.id_pub = publicadores.id_pub
and publicadores.nombre_pub = 'New Age Books'
```

2.- Que libros ha escrito “Blotchet-Halls”

```
select titulo
from titulos
where id_titulo = (select id_titulo
                  from titulo_autor
                  where id_autor = (select id_autor
                                   from autores
                                   where apellido = 'Blotchet-Halls'
                                   )
                  )
```

**titulo**

1 

Fifty Years in Buckingham Palace Kitchens
---

3.- Que publicadores imprimen los libros mas caros

```
select nombre_pub
from publicadores
where id_pub = ( select id_pub
                from titulos
                where precio = ( select max(precio)
                                from titulos
                                )
                )
```

**nombre\_pub**

1 

Algodata Infosystems
----------------------

4.- Que libros han sido vendidos en California ('CA')

```
select distinct id_tienda, id_titulo
from detalle_venta
where id_tienda in ( select id_tienda
                    from tiendas
                    where estado = 'CA'
                    )
```

	id_tienda	id_titulo
1	7066	BU2075
2	7066	BU7832
3	7066	MC3021
4	7066	PC1035
5	7066	PC8888
6	7066	PS7777
7	7066	TC4203
8	7066	TC7777