



Sub-Consultas y Vistas

1 SUB-CONSULTAS

- a) Muestre las empresas de envío que no fueron utilizadas por la librería para realizar entregas entre el 01-01-2020 y el 31-05-2020.

```
SELECT * FROM envio
WHERE id_envio NOT IN (SELECT DISTINCT id_envio
                      FROM venta
                      WHERE fecha_envio BETWEEN '2020-01-01' AND '2020-05-31');
```

-- Filas devueltas: 3

- b) Liste el nombre de las ciudades a las cuales no se envió nunca un artículo.

```
SELECT ciudad FROM ciudad
WHERE id_ciudad NOT IN (SELECT id_ciudad FROM venta
                      INNER JOIN socio USING(id_socio)
                      INNER JOIN ciudad USING(id_ciudad));
```

-- Filas devueltas: 139

- c) Liste todas las funciones que no fueron asignadas a ningún empleado.

```
SELECT funcion FROM funcion
WHERE id_funcion NOT IN (SELECT id_funcion FROM empleado);
```

-- Filas devueltas: 3

- d) Muestre las empresas de envío que fueron utilizadas por la librería para entregar las compras entre los días 01-01-2019 y 31-12-2019 a las provincias de 'La Rioja' y 'Mendoza'.

```
SELECT DISTINCT envio FROM envio
INNER JOIN venta USING(id_envio)
INNER JOIN socio USING(id_socio)
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
WHERE fecha_envio BETWEEN '2019-01-01' AND '2019-12-31'
AND provincia IN ('La Rioja', 'Mendoza');
```

-- Filas devueltas: 9

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- e) Obtenga el número de venta, el id y nombre del socio, el id y nombre del empleado, fecha y el importe de todas las ventas que superaron el importe promedio de venta, ordenado por importe de mayor a menor, fecha y socio.

```
SELECT id_venta, id_socio, ps.nombre AS socio, id_empleado,
       pe.nombre AS empleado, fecha, total
FROM venta
INNER JOIN socio USING(id_socio)
INNER JOIN persona ps ON id_socio = ps.id_persona
INNER JOIN empleado USING(id_empleado)
INNER JOIN persona pe ON id_empleado = pe.id_persona
WHERE total > (SELECT AVG(total)
              FROM venta)
ORDER BY total DESC, fecha, socio;
-- Filas devueltas: 908
```

- f) Liste el id, nombre, la ciudad y la provincia de todos los socios que hayan comprado artículos del género 'Acción' desde el 01-01-2019 en adelante, ordenados alfabéticamente.

```
SELECT id_socio, nombre, ciudad, provincia, genero, fecha
FROM articulo
INNER JOIN genero USING(id_genero)
INNER JOIN detalle_venta USING(id_articulo)
INNER JOIN venta USING(id_venta)
INNER JOIN socio USING(id_socio)
INNER JOIN persona ON id_socio = id_persona
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
WHERE genero LIKE 'Accion'
      AND fecha > '2019-01-01'
ORDER BY nombre;
-- Filas devueltas: 200
```

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- g) Obtenga el socio, el empleado, título y la fecha de todos los artículos que fueron vendidos en 'Tucumán', 'Salta' y 'Jujuy' y tienen el mismo autor del artículo de título 'Cien Años De Soledad'. Ordenado por título, fecha y empleado.

```
SELECT ps.nombre AS socio, pe.nombre AS empleado, titulo, fecha
FROM articulo
INNER JOIN autor USING(id_autor)
INNER JOIN detalle_venta USING(id_articulo)
INNER JOIN venta USING(id_venta)
INNER JOIN socio USING(id_socio)
INNER JOIN persona ps ON id_socio = ps.id_persona
INNER JOIN empleado USING(id_empleado)
INNER JOIN persona pe ON id_empleado = pe.id_persona
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
WHERE provincia IN ('Tucumán', 'Salta', 'Jujuy')
      AND id_autor = (SELECT id_autor
                      FROM autor
                      INNER JOIN articulo USING(id_autor)
                      WHERE titulo LIKE 'Cien Años De Soledad')
ORDER BY titulo, fecha, empleado;
-- Filas devueltas: 13
```

- h) Muestre el nombre y la función de todos los empleados y, en caso de tener familiares, debe listar junto a los datos de cada empleado, el nombre y el parentesco del mismo. Ordenar del modo adecuado para mostrar primero los empleados que no tienen familiares.

```
SELECT p.nombre AS empleado, funcion, f.nombre AS familiar, parentesco
FROM funcion
INNER JOIN empleado USING(id_funcion)
INNER JOIN persona p ON id_persona = id_empleado
LEFT JOIN familiar f USING(id_empleado)
ORDER BY familiar;
-- Filas devueltas: 2285
```

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- i) Muestre género, el origen, el título y el precio de todos los artículos; y en caso no tener información de origen o género, debe mostrar el título. El listado sólo debe mostrar los artículos que no tiene información del año de creación. NOTA: Debe usar **LEFT Y RIGHT JOIN**.
`SELECT` genero, origen, titulo, precio `FROM` articulo
`LEFT JOIN` genero `USING`(id_genero)
`LEFT JOIN` origen `USING`(id_origen)
`WHERE` anio `IS NULL`;
-- Filas devueltas: 220
- j) Liste todas las ciudades y la provincia a la que pertenece, y, en caso de que haya socios registrados en las mismas, también mostrar el socio, ordenado por ciudad en forma descendente y por socio alfabéticamente.
`SELECT` ciudad, provincia, nombre `AS` socio
`FROM` provincia
`INNER JOIN` ciudad `USING`(id_provincia)
`LEFT JOIN` (`SELECT` * `FROM` socio
 `INNER JOIN` persona `ON` id_socio = id_persona) `AS` socios `USING`(id_ciudad)
`ORDER BY` ciudad `DESC`, socio;
-- Filas devueltas: 4662
- k) Muestre el id, nombre, función y el monto total de ventas realizadas, de los empleados que no hayan superado en monto a "Carrascosa Laura Fabiola".
`SELECT` id_id_empleado, nombre `AS` socio, funcion, `SUM`(total) `AS` monto
`FROM` funcion
`INNER JOIN` empleado `USING`(id_funcion)
`INNER JOIN` persona p `ON` id_persona = id_empleado
`INNER JOIN` venta `USING`(id_empleado)
`GROUP BY` id_empleado
`HAVING` monto < (`SELECT SUM`(total)
 `FROM` empleado
 `INNER JOIN` persona p `ON` id_persona = id_empleado
 `INNER JOIN` venta `USING`(id_empleado)
 `WHERE` nombre `LIKE` 'Carrascosa Laura Fabiola');
-- Filas devueltas: 162

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- l) Muestre el id, nombre, la ciudad y la cantidad de artículos comprados por los socios que no sean de 'Chaco', 'Formosa', 'Corrientes' ni 'Misiones' y que hayan comprado más artículos que la cantidad total vendida del artículo 'Harry Potter Y La Orden del Fénix'.

```
SELECT id_socio, nombre AS socio, ciudad, SUM(cantidad) AS cantidad FROM socio
INNER JOIN persona ON id_socio = id_persona
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
INNER JOIN venta USING(id_socio)
INNER JOIN detalle_venta USING(id_venta)
WHERE provincia NOT IN ('Chaco', 'Formosa', 'Corrientes', 'Misiones')
GROUP BY id_socio
HAVING cantidad > (SELECT SUM(cantidad) FROM articulo
INNER JOIN detalle_venta USING(id_articulo)
WHERE titulo LIKE 'Harry Potter Y La Orden Del Fenix');
```

-- Filas devueltas: 3

- m) Liste todas las funciones existentes, y en caso de haber sido asignada a algún empleado, también debe mostrar el nombre y el sueldo del mismo.

```
SELECT funcion, nombre AS empleado, sueldo FROM funcion
LEFT JOIN (SELECT * FROM empleado
INNER JOIN persona ON id_persona = id_empleado) AS empleados USING(id_funcion);
```

-- Filas devueltas: 889

- n) Obtenga una lista con el id, nombre, función, sueldo, antigüedad y año de ingreso de todos los empleados que posean por lo menos 1 año de antigüedad y cuyo año de ingreso sea posterior al año de ingreso de 'Vega Miranda Americo Raul'. Ordenado por antigüedad de mayor a menor.

```
SELECT id_empleado, nombre AS empleado, funcion, sueldo, fecha_alta, timestampdiff(year,
fecha_alta, curdate()) AS antiguedad FROM funcion
INNER JOIN empleado USING(id_funcion)
INNER JOIN persona ON id_persona = id_empleado
WHERE timestampdiff(year, fecha_alta, curdate()) > 0
AND fecha_alta > (SELECT fecha_alta FROM empleado
INNER JOIN persona ON id_persona = id_empleado
WHERE nombre LIKE 'Vega Miranda Americo Raul')
```

```
ORDER BY antiguedad DESC;
```

-- Filas devueltas: 608

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- o) Muestre el id y el nombre de TODAS las empresas de envío, y también mostrar la cantidad de entregas realizadas por c/empresa a cada provincia (de haberlo hecho). Ordenar de mayor a menor por cantidad de envíos.

```
SELECT envio, provincia, COUNT(id_envio) AS cantidad
FROM envio
INNER JOIN venta USING(id_envio)
INNER JOIN socio USING(id_socio)
INNER JOIN ciudad USING(id_ciudad)
INNER JOIN provincia USING(id_provincia)
GROUP BY id_envio, provincia
ORDER BY cantidad DESC;
-- Filas devueltas: 222
```

- p) Muestre el id, nombre, función y monto total de ventas de los empleados que hayan vendido menos que el 10% del monto total de las ventas enviadas por “FedEx”.

```
SELECT id_empleado, nombre as empleado, funcion, SUM(total) AS monto
FROM funcion
INNER JOIN empleado USING(id_funcion)
INNER JOIN persona ON id_persona = id_empleado
INNER JOIN venta USING(id_empleado)
GROUP BY id_empleado
HAVING monto < (SELECT SUM(total) * 0.1
FROM envio
INNER JOIN venta USING(id_envio)
WHERE envio LIKE 'FedEx');
-- Filas devueltas: 610
```

- q) Modifique el gasto de envío, aumentando un 5%, a las ventas enviadas por ‘UPS’, ‘DHL’ o ‘FedEx’.

```
UPDATE envio
INNER JOIN venta USING(id_envio)
SET gasto_envio = gasto_envio * 1.05
WHERE envio IN ('UPS', 'DHL', 'FedEx');
-- Filas modificadas: 569
```

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

- r) Modifique el monto de la multa, aumentando un 10%, de todos los préstamos, de aquellos artículos que tienen origen España, Colombia o Irlanda y el autor no sea 'Gustavo Adolfo Bécquer'.

```
UPDATE prestamo
INNER JOIN articulo USING(id_articulo)
INNER JOIN origen USING(id_origen)
INNER JOIN autor USING(id_autor)
SET multa = multa * 1.1
WHERE origen IN ('España', 'Colombia', 'Irlanda')
      AND autor <> 'Gustavo Adolfo Bécquer'
-- Filas modificadas: 172
```

- s) Elimine las empresas de envío que nunca fueron utilizadas.

```
DELETE FROM envio
WHERE id_envio NOT IN (SELECT distinct id_envio
                      FROM venta);
-- Filas eliminadas: 3
```

- t) Elimine todos los préstamos realizados entre '01-01-2019' y 31-12-2020' y que hayan pagado multa, de aquellos artículos de los cuales no se tiene información del origen del mismo.

```
DELETE prestamo FROM prestamo
INNER JOIN articulo USING(id_articulo)
WHERE id_origen IS NULL AND inicio_prestamo BETWEEN '2019-01-01' AND '2020-12-31';
-- Filas eliminadas: 1052
```

2 VISTAS

- a) Genere una vista llamada info_venta que muestre el nombre del empleado, el nombre y la provincia del socio, el título del artículo, autor, género, origen, editorial y el precio (tabla detalle_venta) de todos los artículos vendidos, la fecha de envío cada uno de los artículos, la empresa por la que fue enviado, la fecha de entrega y si fue entregado, ordenado por artículo y la fecha de venta.

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

CREATE VIEW info_venta **AS**

SELECT pe.nombre **AS** empleado, ps.nombre **AS** socio, provincia, titulo, autor, genero, origen, editorial, dv.precio, fecha_envio, envio **AS** empresa_envio, fecha **AS** fecha_venta, entregado
FROM articulo

LEFT JOIN autor **USING**(id_autor)

LEFT JOIN genero **USING**(id_genero)

LEFT JOIN origen **USIN**(id_origen)

LEFT JOIN editorial **USING**(id_editorial)

INNER JOIN detalle_venta dv **USING**(id_articulo)

INNER JOIN venta **USING**(id_venta)

INNER JOIN envio **USING**(id_envio)

INNER JOIN socio **USING**(id_socio)

INNER JOIN persona ps **ON** ps.id_persona = id_socio

INNER JOIN empleado **USING**(id_empleado)

INNER JOIN persona pe **ON** pe.id_persona = id_empleado

INNER JOIN ciudad **USING**(id_ciudad)

INNER JOIN provincia **USING**(id_provincia)

ORDER BY id_articulo, fecha_venta;

-- Filas devueltas: 3660

- b) Genere una vista llamada info_prestamos que muestre el nombre, la ciudad y provincia del socio, el título del artículo, autor, género, origen, editorial, el monto y la multa de todos los artículos prestados, la fecha de inicio préstamo, fecha del fin, fecha de devolución, la cantidad de días agendada para el préstamo (de acuerdo a las fechas de inicio y fin) y la cantidad de días que duró cada préstamo realmente, considerando la fecha de inicio y devolución.

CREATE VIEW info_prestamos **AS**

SELECT nombre as socio, ciudad, provincia, titulo, autor, genero, origen, editorial, monto, multa, inicio_prestamo, fin_prestamo, fecha_devolucion, **timestampdiff**(day, inicio_prestamo, fin_prestamo) **AS** dias_prestamo, **timestampdiff**(day, inicio_prestamo, fecha_devolucion) **AS** dias_devolucion

FROM articulo

LEFT JOIN autor **USING**(id_autor)

LEFT JOIN genero **USING**(id_genero)

LEFT JOIN origen **USING**(id_origen)

LEFT JOIN editorial **USING**(id_editorial)

INNER JOIN prestamo **USING**(id_articulo)

CONCEPTOS DE BASES DE DATOS I

Programador Universitario y Lic. en Informática
Fac. de C. Exactas y Tecnología. – UNT



Trabajo Práctico Nro. 9 - Ciclo 2022

INNER JOIN socio **USING**(id_socio)

INNER JOIN persona **ON** id_persona = id_socio

INNER JOIN ciudad **USING**(id_ciudad)

INNER JOIN provincia **USING**(id_provincia);

-- Filas devueltas: 5146

- Una vez creada la vista muestre nombre del socio, título del artículo, y la diferencia en días de mora en la devolución.

SELECT socio, titulo, dias_devolucion - dias_prestamo **AS** mora

FROM info_prestamos;

-- Filas devueltas: 5146