educación IT
CODE YOUR FUTURE

Introducción a Bases de Datos y SQL

Resolución de la etapa 5

Resolución de la etapa 5.1

1. Utilizando la tabla ***PRODUCTOS_NEPTUNO***, obtener una lista de todos aquellos productos cuyo **precio supere el precio promedio**. Esta debe contener todos los campos de la tabla. Por último, ordenar **alfabéticamente** el resultado según los nombres de los productos.

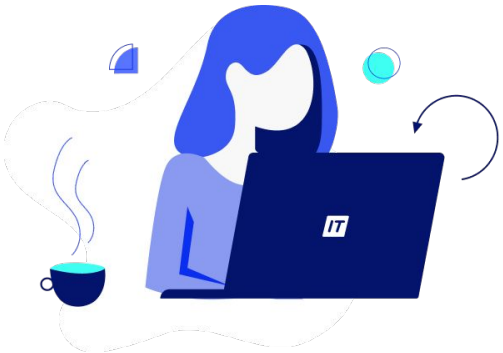
```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE PRECIOUNIDAD >
(SELECT AVG(PRECIOUNIDAD) FROM PRODUCTOS_NEPTUNO)
ORDER BY NOMBREPRODUCTO;
```

2. Toma la tabla ***PRODUCTOS_NEPTUNO*** y obtén una lista de todos aquellos productos cuyo **precio** sea superior al producto **más caro** de la tabla ***PRODUCTOS_SUSPENDIDOS***. Esta debe contener **todos los campos** de la tabla. Luego, ordenar el resultado de **mayor a menor** de acuerdo con los precios obtenidos.

```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE PRECIOUNIDAD >
(SELECT MAX(PRECIOUNIDAD) FROM PRODUCTOS_SUSPENDIDOS)
ORDER BY PRECIOUNIDAD DESC;
```

3. Utilizando la tabla ***VARONES***, obtener una lista de todos aquellos bebés que hayan nacido con una cantidad de **semanas de gestación menor** que el bebé de **sexo indeterminado con menor gestación**. La lista debe mostrar **todos los campos** de la tabla.

```
SELECT * FROM VARONES
WHERE SEMANAS <
(SELECT MIN(SEMANAS) FROM INDETERMINADOS);
```

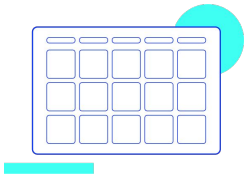


4. Dada la tabla ***PRODUCTOS_NEPTUNO***, obtener una lista de todos los productos cuyo nombre comience con la **inicial del apellido del empleado cuyo *IDEMPLEADO*** es el número ***8***. Esta debe mostrar **todos los campos** de la tabla ***PRODUCTOS_NEPTUNO*** y se debe ordenar **alfabéticamente** según los nombres de los **productos**.

```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE LEFT(NOMBREPRODUCTO, 1) =
(SELECT LEFT(NOMBRE_EMPLEADO, 1) FROM EMPLEADOS
WHERE IDEMPLEADO = 8)
ORDER BY NOMBREPRODUCTO;
```

5. Utilizando la tabla ***PRODUCTOS_NEPTUNO***, obtener una lista de todos los productos que pertenezcan al **proveedor con el ID más alto**. La lista debe mostrar todos los campos de la tabla ***PRODUCTOS_NEPTUNO*** y debe ordenarse **alfabéticamente** por los nombres de los **productos**.

```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE IDPROVEEDOR =
(SELECT MAX(IDPROVEEDOR) FROM PROVEEDORES)
ORDER BY NOMBREPRODUCTO;
```

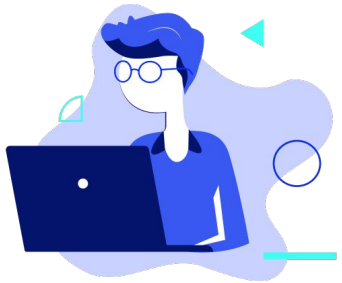


6. Dada la tabla ***PRODUCTOS_NEPTUNO***, extraer una lista de todos aquellos productos que pertenezcan a la categoría ***BEBIDAS*** y cuyo **precio sea superior al producto más caro de** la categoría ***CONDIMENTOS***. La lista debe mostrar **todos los campos** de la tabla.

```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE NOMBRECATEGORIA = 'BEBIDAS' AND
PRECIOUNIDAD >
(SELECT MAX(PRECIOUNIDAD) FROM PRODUCTOS_NEPTUNO
WHERE NOMBRECATEGORIA = 'CONDIMENTOS');
```

7. A partir de la tabla ***MUJERES***, obtener una lista de todas aquellas **bebas** que hayan nacido de **madres con una edad superior a la madre más longeva** que figure en la tabla ***VARONES***. La lista debe mostrar **todos los campos** de la tabla ***MUJERES***.

```
SELECT * FROM MUJERES
WHERE EDAD_MADRE >
(SELECT MAX(EDAD_MADRE) FROM VARONES);
```



8. Utilizando la tabla ***CLIENTES_NEPTUNO***, extraer una lista de todos los clientes que hayan realizado **compras por un cargo superior a 500 dólares**. La lista debe mostrar los campos ***NOMBRECOMPANIA***, ***CIUDAD*** y ***PAÍS*** y debe estar ordenada **alfabéticamente** por los nombres de las **compañías**.

```
SELECT NOMBRECOMPANIA, CIUDAD, PAIS
FROM CLIENTES_NEPTUNO
WHERE NOMBRECOMPANIA IN
(SELECT NOMBRECOMPANIA FROM PEDIDOS_NEPTUNO
WHERE CARGO > 500);
```

Resolución de la etapa 5.2

1. Utiliza la tabla ***CLIENTES_NEPTUNO***, generar una consulta que muestre los campos ***IDCLIENTE***, ***NOMBRECOMPANÍA***, ***CIUDAD*** y ***PAÍS***. Luego, agregar una columna llamada ***CONTINENTE***, en la que se muestren los valores definidos en las condiciones.

```
SELECT IDCLIENTE, NOMBRECOMPANIA, CIUDAD, PAIS,
CASE
WHEN PAIS IN ('ARGENTINA', 'BRASIL', 'VENEZUELA') THEN 'AMERICA DEL SUR'
WHEN PAIS IN ('MÉXICO', 'USA', 'CANADÁ') THEN 'AMERICA DEL NORTE'
ELSE 'EUROPA'
END AS CONTINENTE
FROM CLIENTES_NEPTUNO
ORDER BY CONTINENTE, PAIS;
```

2. Utilizar la tabla ***PEDIDOS_NEPTUNO***, generar una consulta que muestre los campos ***IDPEDIDO, NOMBRECOMPAÑÍA, FECHAPEDIDO y CARGO***.

Luego, agregar una **columna** llamada ***EVALUACIÓN*** en la que se muestren los valores definidos en las condiciones.

```
SELECT IDPEDIDO, NOMBRECOMPANIA, FECHAPEDIDO, CARGO,
CASE
WHEN CARGO > 700 THEN 'EXCELENTE'
WHEN CARGO > 500 THEN 'MUY BUENO'
WHEN CARGO > 250 THEN 'BUENO'
WHEN CARGO > 50 THEN 'REGULAR'
ELSE 'MALO'
END AS 'EVALUACION'
FROM PEDIDOS_NEPTUNO
ORDER BY CARGO DESC;
```

3. Utilizando la tabla ***PRODUCTOS_NEPTUNO***, generar una consulta que muestre los campos ***IDPRODUCTO, NOMBREPRODUCTO, NOMBRECATEGORÍA y PRECIOUNIDAD***.

Agregar una columna con el nombre ***TIPO*** en la que se muestren los valores definidos en las condiciones.

```
SELECT IDPRODUCTO, NOMBREPRODUCTO, NOMBRECATEGORIA, PRECIOUNIDAD,
CASE
WHEN PRECIOUNIDAD > 100 THEN 'DELUXE'
WHEN PRECIOUNIDAD > 10 THEN 'REGULAR'
ELSE 'ECONOMICO'
END AS TIPO
FROM PRODUCTOS_NEPTUNO
ORDER BY PRECIOUNIDAD DESC;
```

Resolución de la etapa 5.3

1. Obtener una lista de todos aquellos bebés nacidos con **menos de 20 semanas de gestación**. La lista debe mostrar los bebés de **cualquier sexo**, por lo tanto, la consulta se debe llevar a cabo en las tablas ***VARONES***, ***MUJERES*** e ***INDETERMINADOS***.

```
SELECT * FROM VARONES WHERE SEMANAS < 20
UNION
SELECT * FROM MUJERES WHERE SEMANAS < 20
UNION
SELECT * FROM INDETERMINADOS WHERE SEMANAS < 20;
```

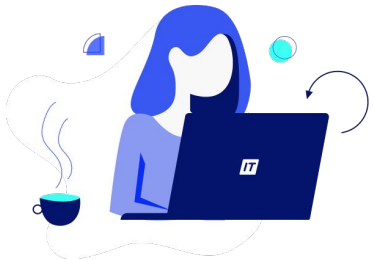
2. Luego, obtener una lista de todos aquellos bebés nacidos durante el mes de **septiembre**, con **más de 40 semanas de gestación** y nacidos de **madres chilenas casadas**.

La lista debe mostrar los bebés de **cualquier sexo**, por lo tanto, debe llevar adelante la consulta en las tablas ***VARONES***, ***MUJERES*** e ***INDETERMINADOS***.

```
SELECT * FROM VARONES WHERE FECHA LIKE '%/09/%' AND NACIONALIDAD =
'CHILENA' AND ESTADO_CIVIL_MADRE = 'CASADA' AND SEMANAS > 40
UNION
SELECT * FROM MUJERES WHERE FECHA LIKE '%/09/%' AND NACIONALIDAD =
'CHILENA' AND ESTADO_CIVIL_MADRE = 'CASADA' AND SEMANAS > 40
UNION
SELECT * FROM INDETERMINADOS WHERE FECHA LIKE '%/09/%' AND NACIONALIDAD =
'CHILENA' AND ESTADO_CIVIL_MADRE = 'CASADA' AND SEMANAS > 40;
```

3. Obtener una lista de todos aquellos **productos** (a la venta y suspendidos) cuyo **precio supere los 80 dólares**. La búsqueda se debe llevar a cabo en las tablas ***PRODUCTOS_NEPTUNO*** y ***PRODUCTOS_SUSPENDIDOS***. Después, ordenar el resultado **alfabéticamente** según los nombres de los productos.

```
SELECT *FROM PRODUCTOS_NEPTUNO
WHERE PRECIOUNIDAD > 80
UNION
SELECT * FROM PRODUCTOS_SUSPENDIDOS
WHERE PRECIOUNIDAD > 80
ORDER BY NOMBREPRODUCTO;
```

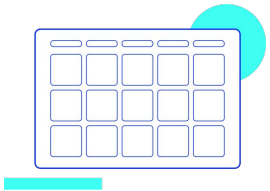


4. Modificar la consulta anterior para **agregar una columna** llamada ***CONDICIÓN*** en la que se muestre el texto ***"A LA VENTA"*** en el caso de que el registro provenga de la tabla ***PRODUCTOS_NEPTUNO***; o el texto ***SUSPENDIDO*** si el registro proviene de la tabla ***PRODUCTOS_SUSPENDIDOS***.

```
SELECT *, 'A LA VENTA' AS CONDICION FROM PRODUCTOS_NEPTUNO
WHERE PRECIOUNIDAD > 80
UNION
SELECT *, 'SUSPENDIDO' AS CONDICION FROM PRODUCTOS_SUSPENDIDOS
WHERE PRECIOUNIDAD > 80
ORDER BY NOMBREPRODUCTO;
```

5. Generar una lista de todos los productos que pertenezcan a la categoría ***BEBIDAS*** sin importar si los mismos se encuentran a la venta o suspendidos (la búsqueda se debe hacer en las tablas ***PRODUCTOS_NEPTUNO*** y la tabla ***PRODUCTOS_SUSPENDIDOS***). Luego, ordenar la lista **alfabéticamente** según los nombres de los productos.

```
SELECT * FROM PRODUCTOS_NEPTUNO
WHERE NOMBRECATEGORIA = 'BEBIDAS'
UNION
SELECT * FROM PRODUCTOS_SUSPENDIDOS
WHERE NOMBRECATEGORIA = 'BEBIDAS'
ORDER BY NOMBREPRODUCTO;
```



6. **Duplicar** el producto cuyo **ID** es el número ***43*** de la tabla ***PRODUCTOS_NEPTUNO*** en la tabla ***PRODUCTOS_SUSPENDIDOS*** a través de una *consulta de datos anexados*.

```
INSERT INTO PRODUCTOS_SUSPENDIDOS
(IDPRODUCTO, NOMBREPRODUCTO, NOMBRECONTACTO, NOMBRECATEGORIA, PRECIOUNIDAD,
SUSPENDIDO, IDPROVEEDOR)
SELECT IDPRODUCTO, NOMBREPRODUCTO, NOMBRECONTACTO, NOMBRECATEGORIA,
PRECIOUNIDAD, SUSPENDIDO, IDPROVEEDOR
FROM PRODUCTOS_NEPTUNO
WHERE IDPRODUCTO = 43 ;
```

7. **Repetir la consulta** generada en el **paso 5** (cinco) para observar que la cantidad de productos obtenida siga siendo la misma.

```
SELECT * FROM PRODUCTOS_NEPTUNO WHERE NOMBRECATEGORIA = 'BEBIDAS'
UNION
SELECT * FROM PRODUCTOS_SUSPENDIDOS WHERE NOMBRECATEGORIA = 'BEBIDAS'
ORDER BY NOMBREPRODUCTO;
```

8. Modificar la consulta del **paso 5** (cinco) para **mostrar el producto duplicado**.

```
SELECT * FROM PRODUCTOS_NEPTUNO WHERE NOMBRECATEGORIA = 'BEBIDAS'
UNION ALL
SELECT * FROM PRODUCTOS_SUSPENDIDOS WHERE NOMBRECATEGORIA = 'BEBIDAS'
ORDER BY NOMBREPRODUCTO;
```

9. **Eliminar** el producto cuyo **ID** es el número ***43*** de la tabla ***PRODUCTOS_SUSPENDIDOS***.

```
SET SQL_SAFE_UPDATES = 0;
DELETE FROM PRODUCTOS_SUSPENDIDOS WHERE IDPRODUCTO = 43;
```

Resolución de la etapa 5.4

1. Generar una tabla con el nombre ***EQUIPOS*** en la que **sólo se cree un campo** cuyo nombre sea ***EQUIPO***. Este campo debe ser de tipo ***VARCHAR***, almacenar **hasta 20 caracteres** y debe ser ***PRIMARY KEY*** de la tabla.

```
CREATE TABLE EQUIPOS (EQUIPO VARCHAR(20) PRIMARY KEY);
```

2. Cargar los nombres de las siguientes selecciones en la tabla ***EQUIPOS***: ***ARGENTINA***, ***BRASIL***, ***PARAGUAY***, ***CHILE***, ***URUGUAY***, ***COLOMBIA***, ***ECUADOR***, ***PERÚ***, ***BOLIVIA***, ***VENEZUELA***.

```
INSERT INTO EQUIPOS
VALUES ('ARGENTINA'), ('BRASIL'), ('CHILE'), ('PARAGUAY'), ('URUGUAY'),
('COLOMBIA'), ('ECUADOR'), ('PERÚ'), ('BOLIVIA'), ('VENEZUELA');
```

3. **Generar un *producto cartesiano*** en base a la misma tabla para lograr un *fixture* en el cual **cada selección juegue con las otras selecciones** (un partido como local y otro como visitante). Ordenar **alfabéticamente** el resultado, según el nombre del equipo local.

```
SELECT * FROM EQUIPOS L CROSS JOIN EQUIPOS V
WHERE L.EQUIPO <> V.EQUIPO
ORDER BY L.EQUIPO;
```

Resolución de la etapa 5.5

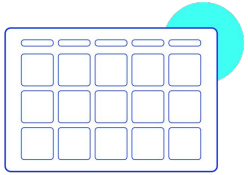
1. Generar un listado en la que se muestre el campo ***NOMBRECONTACTO*** de la tabla ***PROVEEDORES*** y los campos ***IDPRODUCTO***, ***NOMBREPRODUCTO*** y ***PRECIOUNIDAD*** de la tabla ***PRODUCTOS NEPTUNO***. Generar el ***JOIN*** a través de la cláusula ***FROM***.

   Luego, ordenar el resultado **alfabéticamente** por los nombres de los contactos y, cuando el nombre del contacto (nombre del **proveedor**) se repita, ordena los **productos** provistos por el proveedor, también alfabéticamente.

```
SELECT PN.NOMBRECONTACTO, IDPRODUCTO, NOMBREPRODUCTO, PRECIOUNIDAD
FROM PROVEEDORES P JOIN PRODUCTOS_NEPTUNO PN
ON P.IDPROVEEDOR = PN.IDPROVEEDOR
ORDER BY PN.NOMBRECONTACTO, NOMBREPRODUCTO;
```

2. **Modificar la consulta anterior** para generar el ***JOIN*** a través de la cláusula ***WHERE***.

```
SELECT PN.NOMBRECONTACTO, IDPRODUCTO, NOMBREPRODUCTO, PRECIOUNIDAD
FROM PROVEEDORES P, PRODUCTOS_NEPTUNO PN
WHERE P.IDPROVEEDOR = PN.IDPROVEEDOR
ORDER BY PN.NOMBRECONTACTO, NOMBREPRODUCTO;
```



3. Crear un listado en la que se muestre el campo ***EMPRESA*** de la tabla ***CLIENTES*** y los campos ***NUMERO_PEDIDO***, ***FECHA_PEDIDO*** y ***FORMA_PAGO*** de la tabla ***PEDIDOS***. Generar el ***JOIN*** a través de la cláusula ***FROM***. Luego, ordenar el listado **alfabéticamente** por los nombres de las empresas.

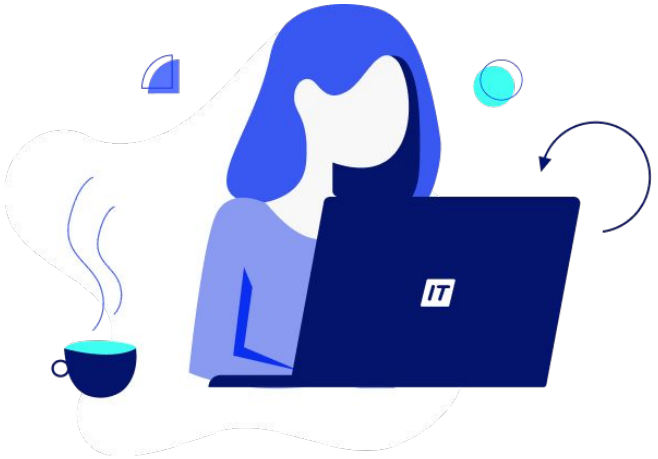
```
SELECT EMPRESA, NUMERO_PEDIDO, FECHA_PEDIDO, FORMA_PAGO
FROM CLIENTES C JOIN PEDIDOS P
ON C.COD_CLIENTE = P.CODIGO_CLIENTE
ORDER BY EMPRESA;
```

4. Modificar la consulta anterior para mostrar **únicamente** aquellos **clientes que no hayan efectuado ningún pedido**.

```
SELECT EMPRESA, NUMERO_PEDIDO, FECHA_PEDIDO, FORMA_PAGO
FROM CLIENTES C LEFT JOIN PEDIDOS P
ON C.COD_CLIENTE = P.CODIGO_CLIENTE
WHERE P.NUMERO_PEDIDO IS NULL
ORDER BY EMPRESA;
```

5. Luego, modificar la consulta anterior para **mostrar únicamente el campo *EMPRESA***.

```
SELECT EMPRESA
FROM CLIENTES C LEFT JOIN PEDIDOS P
ON C.COD_CLIENTE = P.CODIGO_CLIENTE
WHERE P.NUMERO_PEDIDO IS NULL
ORDER BY EMPRESA;
```



6. *¿Existe algún **proveedor** que en este momento no le esté vendiendo **ningún producto** a nuestra empresa?* Responder esta pregunta a través de una consulta en la que utilices las tablas ***PROVEEDORES*** y ***PRODUCTOS NEPTUNO***. Mostrar **todos los campos** de ambas tablas en el resultado de la consulta.

```
SELECT * FROM PROVEEDORES P LEFT JOIN PRODUCTOS_NEPTUNO PN
ON P.IDPROVEEDOR = PN.IDPROVEEDOR
WHERE PN.IDPRODUCTO IS NULL;
```

7. *¿Existe algún **producto** que **no se sepa** quién es **el proveedor** que lo provee a nuestra empresa?*
   Responder esta pregunta a través de una consulta en la que utilices las tablas ***PROVEEDORES*** y ***PRODUCTOS NEPTUNO***. Puedes mostrar **todos los campos** de ambas tablas en el resultado de la consulta.

```
SELECT * FROM PROVEEDORES P RIGHT JOIN PRODUCTOS_NEPTUNO PN
ON P.IDPROVEEDOR = PN.IDPROVEEDOR
WHERE P.IDPROVEEDOR IS NULL;
```

¡Sigamos trabajando!