**PRACTICAS PROGRAMACIÓN**

**BLOQUES PL/PGSQL**

1. A partir de la vista que muestra las tablas de la BD: **pg**\_**tables**, imprimir el recuento de filas de cada tabla de la BD envíos. Imprimir el número de total de tablas y de registros.
2. **Crear una BD Empresa\_3, restaurar BK empresa3 y:**

Escribir un módulo de PL que realice las siguientes tareas:

Crear un esquema países.

0- Obtener una copia de la tabla de pedidos: tbpedidos\_copia

1- Crear una tabla de países (id serial PK y pais varchar).

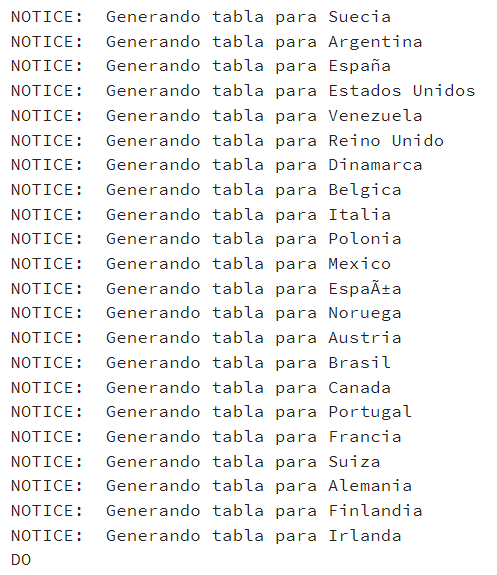
2- Rellenar la tabla de paises con los países que hay en pedidos pero sin repetidos

3- Crear una nueva columna en tbpedidos que se llame: idpais (FK), integer

4- Actualizar la columna idpais en la tabla de tbpedidos utilizando tbpaises.

5- Borrar la columna paisdestinatario de la tabla de tbpedidos;

1. En la BD empresa3, utilizamos la tabla de pedidos y generamos una tabla de pedidos por cada país. En cada tabla se almacenan los registros de ese país. Se imprime el nombre de la tabla generada.



1. Realizar otro script que se encargue de borrar las tablas creadas. Igual que antes que vaya informando por la consola.

**FUNCIONES**

1. Crear una función que devuelva el importe total de un pedido. El cargo de la tabla de pedidos y los detalles de pedido. En la BD empresa3
2. En la BD taller6 crear una función que proporcione la información para mostrar el siguiente mensaje (con las tablas movimientos y claseMovimiento). La función será **generarReporte**, imprimir todos los registros por la consola y en esta primera versión devolver un 0.

**fecha cod.producto clase mov clase tipo cantidad**

**2021-05-02 778045 compra a prov 101 E 100**

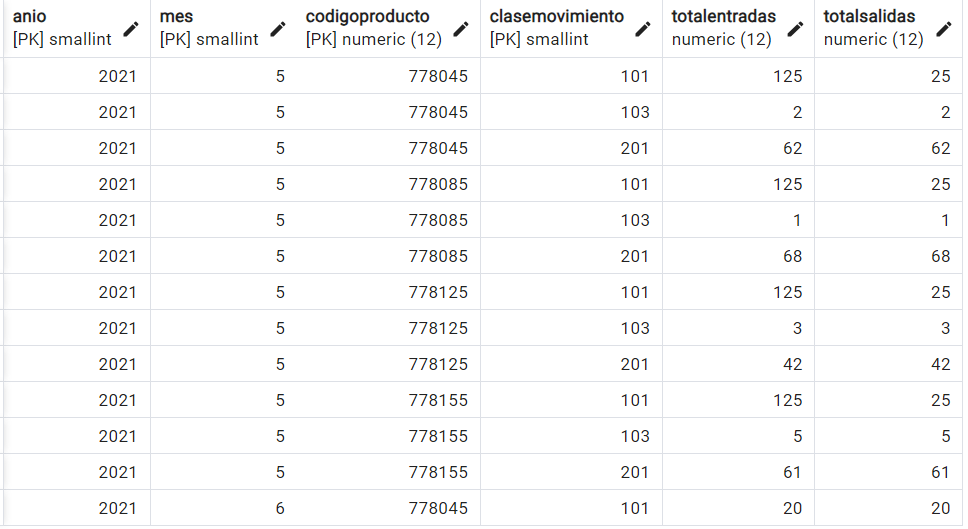
1. Una nueva versión de la función anterior que extraiga el mes y el año de la fecha de movimiento (ver el contenido de la tabla taller6.chComercial.reporteMovimiento). Utilizar case para determinar si estamos en una entrada o en una salida. Imprimir los datos. La función será: **generarReporte2**.
2. En la última versión hay que sumar el total de las entradas y de las salidas. Esta recibe 5 parámetros: año, mes, cod. Producto, clase de producto, cantidad. Tipos: integer, integer, numeric, integer, integer. Dentro de la función se puede realizar un update. Y la llamada puede ser:

select "schComercial"."sumarProducto"(2021, 5, 778045, 101, 100);

Como la tabla reporteMovimiento está vacía puede devolver false.

1. Modificar la función del paso 3 para llamar a la función del paso 4.

La tabla reporte de movimiento quedará:



**PROCEDIMIENTOS**

1. Crear la BD: dbtaller2 y ejecutar desde query tool el fichero:

practicas/ 6\_creacion-taller-002-procedimientos.sql

Crear un procedimiento almacenado para realizar una transferencia de un usuario a otro en la tabla de: "schFinanciero"."cuentasClientes"

1. Crear la BD: dbtaller3 y ejecutar desde query tool el fichero:

Practicas/ 6\_creacion-taller-003-procedimientos.sql

Crear un procedimiento almacenado que calcule e imprima los saldos de las cuentas de los clientes.

1. Crear un procedimiento almacenado para eliminar un pedido (tiene que eliminar primero los detalles y luego el pedido) el id del pedido lo recibe por parámetro.
2. Un procedimiento que inserte una nueva categoría calculando el nuevo id.
3. Crear la BD: dbtaller4 y ejecutar desde query tool el fichero:

Practicas/ 6\_creacion-taller-004-procedimientos.sql

Convertir en procedimiento la última versión de las funciones almacenadas.

**TRIGGERS**

1. Crear la BD dbtaller5 y ejecutar con query tool el script:

practicas/9\_creacion-taller-005-triggers.sql

Se proporciona una tabla cuentasclientes y un par de procedimientos: crearCliente y transferir.

Mediante el uso de triggers queremos saber:

* **A) Quien creo y cuando fue creado cada cliente**. La información se almacenará en logcuentasclientes. El trigger debe ejecutarse cada vez que se produzca un insert dentro de la tabla cuentas\_clientes y por cada fila se ejecutará una función que almacene los datos en la tabla logCuentasClientes y puede retornar el trigger. El usuario que será postgre se puede captura con:

select user into variable

* **B) Trigger de actualización**: Registrar en la tabla de log cuentas clientes las actualizaciones cuando se hacen transferencias.