Proyecto Final

Segunda entrega

# Vistas

Vista 1: productos

Descripción: es una vista que permite visualizar todos los productos que ingresamos incluyendo descripción y marca

Objetivo: Visualizar los productos en stock por marca

Tablas/Datos: Las tablas utilizadas son:

* producto
* marca

Realizamos un Join con la tabla “marca”, para poder obtener la marca solicitada.

Los datos utilizados son los siguientes:

* p.descripcion
* m.id\_marca
* p.marca

Vista 2: productos\_SPEC

Descripción: es una vista creada para poder visualizar los productos cuya marca es SPEC

Objetivo: poder visualizar los productos de dicha marca

Tablas/Datos: Las tablas utilizadas son:

* producto
* marca

Realizamos un Join con la tabla “marca”, para poder obtener la marca solicitada, donde mediante un where condicionamos que nos arroje solo de la marca SPEC

Los datos utilizados son los siguientes:

* p.descripcion
* m.id\_marca
* p.marca

Vista3: movimientos\_usuarios

Descripción: Es una lista que permite visualizar que usuario realizo el movimiento especifico.

Objetivo: Generar una vista que rápidamente permita ver los movimientos por usuario y en que deposito.

Tablas/Datos: Las tablas utilizadas son:

* movimiento\_usuario
* usuario

Realizamos un Join con la tabla “usuario”, para poder obtener donde realizo el movimiento el usuario, donde mediante un where condicionamos que nos arroje solo los movimientos de ingreso

Los datos utilizados son los siguientes:

* mu.usuario
* mu.tipo\_movimiento
* u.id\_usuario

Vista 4: proveedores

Descripción: Es una vista que permite visualizar un detalle de los proveedores ingresados.

Objetivo: tener un detalle de los proveedores ingresados con descripción y cuil\_cuit.

Tablas/Datos: La tabla utilizada es proveedor

Los datos utilizados son los siguientes:

* descripcion
* cuil\_cuit

Vista 4: ciudades

Descripción: Es una vista que permite visualizar un detalle de las ciudades ingresadas.

Objetivo: tener un detalle de las ciudades ingresados con descripción y código postal.

Tablas/Datos: La tabla utilizada es ciudad.

Los datos utilizados son los siguientes:

* descripción
* cp

1. Funciones

Función 1: get\_producto

Descripción: En este caso realicé una función que me permite obtener un producto en particular mediante un id

Objetivo: Permitirme obtener la descripción del producto pasando un id en particular

Tablas/Datos: La tabla utilizada es producto.

Los datos utilizados son los siguientes:

* descripción
* id\_producto

Función 2: count\_product

Descripción: En este caso realicé una función que me permite obtener la cantidad de un tipo de producto en particular mediante un id

Objetivo: Permitirme obtener la cantidad del tipo de producto pasando un id en particular

Tablas/Datos: Las tablas utilizadas son:

* producto
* tipo\_producto

Realizamos un Join con la tabla “tipo\_producto”, para poder obtener a que tipo de producto corresponde, donde mediante un where condicionamos que nos arroje solo la cantidad de ese tipo de producto

1. Stored Procedures

Procedure1: ordenar\_tabla\_marcas

Descripción: Permite ordenar las marcas productos según el campo que le pasemos y lo va hacer de manera ascendente o Descendente, dependiendo del segundo valor que pasemos siendo “asc” para Ascendente y “desc” para Descendente, de compartir cualquier otro valor tomara e valor por defecto de la consulta. Resuelto el stored procedures mostrara la tabla marca según los valores que hayamos compartido.

Objetivo: poder obtener de manera rápida los datos de la tabla marca ordenados según la necesidad que tengamos.

Tablas/Datos: Utilizamos la tabla marca y solicitamos los siguientes campos en el stored procedures para poder hacer el ordenamiento:

* Campo\_tabla (compo de la tabla de productos por el cual queremos ordenar)
* Orden (tipo de ordenamiento A o D)

Procedure2: insertar\_ciudad

Descripción: permite poder realizar un Insert a la tabla ciudad, donde exige para su carga que los campos indicados sean distintos de vacíos, de poder ejecutarse con éxito la carga, devolverá una variable que tendrá un mensaje indicando que la carga fue exitosa y además se ejecuta una consulta a la tabla "ciudad” ordenada por id\_cliente de manera Desc con lo cual podremos ver el registro cargado. Ahora bien si algunos de los campos ingresados está vacío, obtendremos en la variable de salida un mensaje indicando que la carga no fue exitosa y que tienen que completar al menos 1 ciudad.

Objetivo: Poder realizar insert en la tabla ciudad de manera mas simple.

Tablas/Datos:

Utilizamos la tabla clientes y solicitamos los siguientes campos en el stored procedure para le **Insert** de nueva ciudad:

* **IN** nombre varchar(20),
* **IN** código\_postal varchar(50),
* **OUT** output varchar(50)

1. Archivos SQL

Script creación de objetos y Script inserción de datos:

Link de Gut Hub, donde podrán encontrar todos los Script:

<https://github.com/yamilschopp/SQL-CODER.git>