

	UNIVERSIDAD DON BOSCO FACULTAD DE ESTUDIOS TECNOLÓGICOS ESCUELA DE COMPUTACIÓN
CICLO 1-2016	TAREA PERIODO 3
	Temas: Creación de vistas e implementación de Procedimientos almacenados y Triggers Seguridad de Base de datos Materia: Base de datos Docente: Blanca Iris Cañas

Indicaciones:

1. Trabajar en los grupos de trabajo (máximo 4 personas)
2. Mandar los script .sql con la solución el día **Viernes 13 de Mayo a las 12:00 (medianoche)**, no se aceptarán archivos después de la hora máxima de envío
3. Colocar en el correo:
 - a. Asunto: Tarea Periodo 3 - Base de datos
 - b. En el correo: Todos los nombres de los integrantes del grupo
4. Ejercicios idénticos o casi idénticos quedan anulados y tendrá como nota 0.0

Parte 1

Ejercicio:

1. Creación de base de datos
2. Nombre de la base de datos: **Control_Biblioteca**
3. Crear las relaciones entre las tablas indicar cuáles son las claves primarias y foráneas.

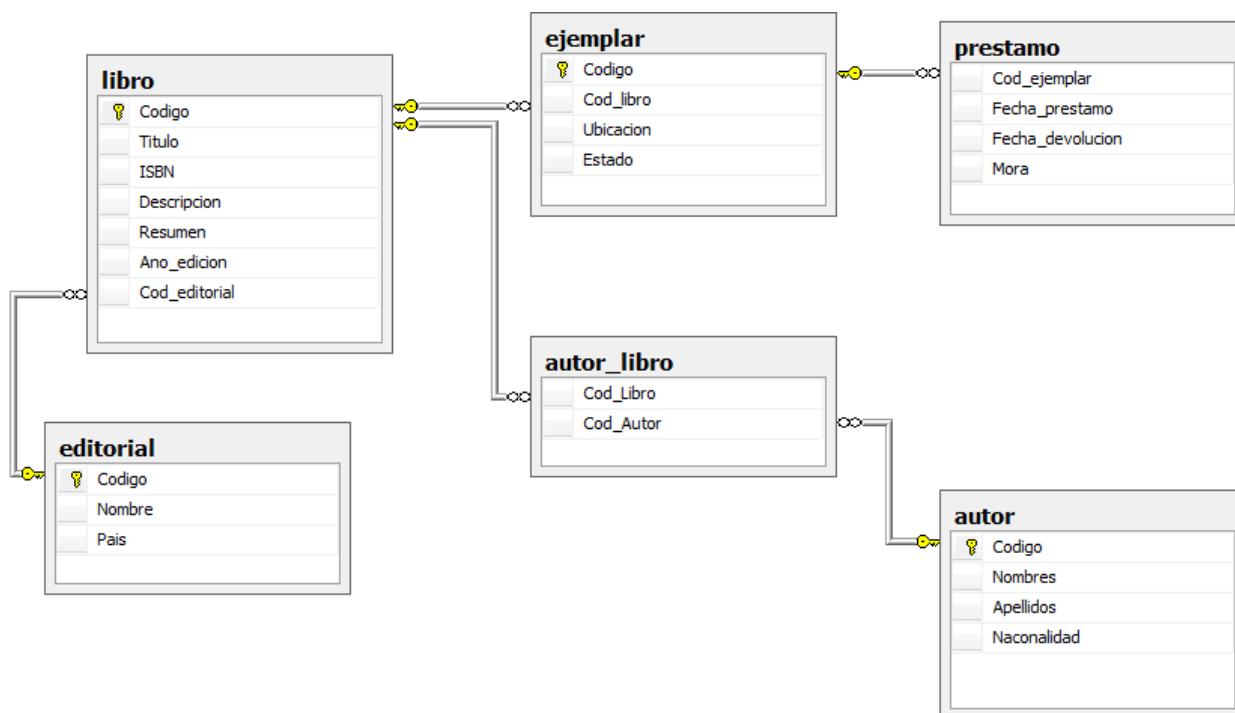


Tabla: Libro

- El campo ISBN se compone de una serie de números separados por guiones (no tiene un formato específico), el campo descripción almacena una breve descripción de lo que trata el libro, el campo resumen almacena información como por ejemplo: cuantas páginas tiene el libro, nombres de los capítulos etc.

Tabla: Ejemplar

- El campo ubicación almacenará en donde estará ubicado el libro, por ejemplo Estante1_puesto1, Estante1_puesto2 y así sucesivamente.

4. Agregar las siguientes restricciones:

- a. El campo Estado de la tabla Ejemplar puede almacenar únicamente los siguientes datos: disponible, prestado y reservado.
- b. El campo ISBN de la tabla Libro tiene que ser un dato único.
- c. El campo año edición debe ser mayor o igual a 2000
- d. La fecha de préstamo no debe ser mayor a la fecha de devolución
- e. La fecha de préstamo por defecto debe ser la fecha actual
- f. La mora debe ser mayor o igual a cero
- g. Los códigos tienen el siguiente formato:
 - i. Tabla Editorial: Dos letras ED, seguidos de tres números, por ejemplo: ED001
 - ii. Tabla Libro: Tres letras LIB, seguidos de tres números, por ejemplo: LIB001
 - iii. Tabla Ejemplar: Dos letras EJ, seguidos de tres números, por ejemplo: EJ001
 - iv. Tabla Autor: Dos letras AU, seguidos de tres números, por ejemplo: AU001

5. Agregar 15 registros mínimos a cada una de las tablas.

6. Crear las siguientes vistas:

- a. Se debe mostrar los datos de los autores junto con los títulos de libros que han escrito
- b. Mostrar todos los libros que están prestados.
- c. Mostrar los libros que están prestados en un determinado rango de fechas de préstamo
- d. Se desea mostrar todos los libros que han sido editados entre un rango de años, considerar también que solo se mostraran aquellos libros donde el origen de la editorial sea española.
- e. Se desea mostrar la siguiente información: la fecha de préstamo, el título del libro y el nombre del autor, de aquellos préstamos que no posean mora.

7. Crear las siguientes los siguientes procedimientos almacenados:

- a. El procedimiento almacenado recibe como parámetro el nombre de una editorial, si esta se encuentra registrada en la base de datos, mostrar todos los autores que tienen libros que han sido publicados por esa editorial, sino existe mostrar un mensaje indicando tal situación.
- b. Mostrar todos los datos de los libros donde su estado sea Prestado o disponible o reservado.
- c. Mostrar los autores y la editorial donde en el título del libro se encuentre una palabra de búsqueda, como por ejemplo se desea mostrar la información de los libros donde en el titulo se encuentre la palabra "Base de datos".
- d. Calcular y mostrar la suma de la mora de un libro específico el cual es enviado como parámetro de entrada y la suma de la mora debe almacenarlo en un parámetro de salida.

8. Crear los siguientes triggers:

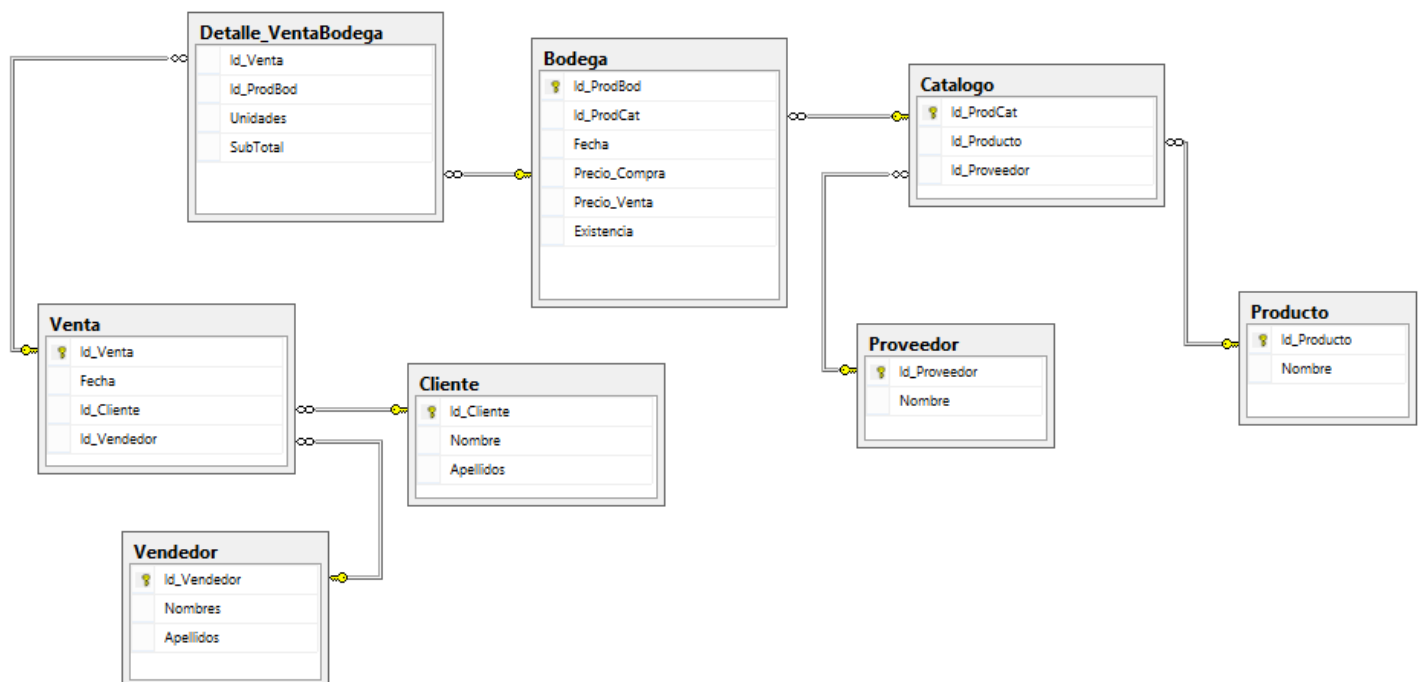
- Crear un trigger que controle la inserción de los libros, no se puede agregar libros con el mismo título y año de edición.
- Crear un trigger en donde se controle la mora del libro, si este se ha entregado después de la fecha de devolución, la mora por defecto es 0.35 ctvs por día, mostrar y guardar en la tabla la mora calculada.
- Crear un trigger que controle la ubicación del ejemplar del libro, ya que más de un ejemplar no pueden ser ubicados en el mismo lugar, también debe controlar que en un estante solo pueden ubicarse como máximo 50 ejemplares.
- Crear un trigger que controle los préstamos de los ejemplares, ya que un ejemplar en donde su estado sea prestado o reservado este no se puede prestar.

Nota: en los procedimientos y triggers debe controlar las entradas de datos y si se da algún error deberá mostrar un mensaje indicando que error se dio y sino mostrar los resultados correctos.

Parte 2

Ejercicio:

Tomando en cuenta el script **control_ventas.sql** crear la siguiente base de datos



1. Crear las siguientes restricciones:

- El campo Nombre de las tablas: Proveedor, Producto deben ser únicos
- Los campos Nombres y Apellidos de las tablas Cliente y Vendedor no deben aceptar valores nulos
- El campo Fecha de la tabla Venta por defecto debe ser la fecha actual del sistema

2. Ingresar datos a las siguientes tablas:

- a. Producto
- b. Cliente
- c. Vendedor
- d. Proveedor

Como mínimo 5 registros a cada tabla

3. Crear los siguientes esquemas:

- a. Esquema: Ventas (colocar las tablas: Cliente, Venta, Vendedor, Detalle_VentaBodega)
- b. Esquema: Productos (colocar las tablas: Producto, Catalogo y Bodega)
- c. Esquema: Proveedores (colocar la tabla: Proveedor)

Nota: Implementar la opción **ALTER SCHEMA – TRANSFER**, para mover las tablas al esquema

4. Crear los siguientes triggers:

- a. Controlar que al momento de ingresar los datos a la tabla **Catalogo**, este debe verificar que el código del producto y proveedor existan en las tablas correspondientes, de lo contrario debe mandar un mensaje de error y no se debe ingresar el registro a la tabla, de lo contrario si se debe ingresar el registro.
- b. Controlar cuando se ingrese los datos a la tabla **Bodega** debe existir el código del producto en catálogo, la fecha no debe ser menor a la actual, el precio de compra no debe ser mayor al precio de venta y las existencias no deben ser negativas, si se da cualquiera de los errores anteriores debe mandar un mensaje de error y no se debe ingresar el registro a la tabla, de lo contrario si se debe ingresar el registro.

5. Crear los siguientes procedimientos almacenados:

- a. Mostrar el nombre del producto, nombre del proveedor y el precio de los productos, cuyo precio sea mayor a un valor enviado como parámetro
- b. Mostrar los datos de un cliente que ha realizado una compra o compras, junto con la fecha de dicha compra y las unidades compradas en dicha compra, el procedimiento recibe como parámetro el código del cliente
- c. Controlar las **ventas de los productos** (tablas Venta y Detalle_VentaBodega) tomando en cuenta: cuando se ingrese la venta de un producto, primero debe verificar que las unidades vendidas no superen a las existencias de ese producto almacenadas en bodega, la fecha de la venta debe ser la fecha y hora del sistema, el producto debe existir en la bodega de lo contrario debe mostrar un mensaje de error, también se debe calcular el subtotal de la venta el cual se calcula por medio de las unidades vendidas por el precio de venta y por ultimo debe actualizar las unidades almacenadas en bodega.
El procedimiento recibe como parámetros de entrada: el nombre del producto, el nombre del vendedor, el nombre del cliente y las unidades vendidas.

6. Creación de usuarios:

- a. Crear un usuario que tenga todos los permisos en la base de datos.
- b. Crear un usuario que tenga los permisos de seleccionar los datos del esquema Ventas.
- c. Crear un usuario que tenga los permisos de actualizar y agregar datos en las tablas producto, catalogo y proveedor.
- d. Crear un usuario que tenga los permisos de eliminar y actualizar los datos de las tablas bajo el esquema Proveedores
- e. Remover los permisos del usuario que creo en el punto d