

FACULTAD DE INGENIERIAS

MÓDULO: COM32 – ADMINISTRACION DE BASE DE DATOS

DOCENTE: ALVENIS ALVAREZ PINTO

ACTIVIDAD: PROGRAMACIÓN AVANZADA EN SQL: PROCEDIMIENTOS,
FUNCIONES, CICLOS Y CONDICIONALES

OBJETIVO DE LA ACTIVIDAD

- Aplicar conocimientos de programación avanzada en SQL para resolver problemas reales en el contexto de un sistema de gestión de ventas.
- Desarrollar procedimientos almacenados, funciones y scripts utilizando estructuras de control como ciclos, condicionales y casos.
- Manipular datos de forma eficiente en escenarios que simulan procesos empresariales reales.

CONTEXTO

Continuando con el caso práctico "**Venta de Productos**" que hemos trabajado de gestión de ventas implementado en una base de datos MySQL. Base de datos que almacena información sobre provincias, clientes, productos, precios, facturas, recibos de pago y sus respectivos ítems.

Para fortalecer la administración del negocio y mantener la integridad de sus datos, se requieren soluciones más avanzadas que permitan automatizar procesos, realizar cálculos complejos y mantener actualizada la información de manera eficiente.

CASOS Y ENUNCIADOS

EJERCICIO 1: ACTUALIZACIÓN DE PRECIOS SEGÚN RANGOS

CASO: La dirección comercial de la empresa decide ajustar los precios de los productos según el siguiente criterio:

- Si el precio actual es menor a \$500.000 → incrementar un 5%.
- Si está entre \$500.000 y \$1.500.000 → incrementar un 3%.
- Si es mayor a \$1.500.000 → mantener el mismo precio.

Se necesita automatizar este ajuste y conocer cuántos productos quedaron en cada rango tras la actualización.

ENUNCIADO

Desarrollar un procedimiento almacenado en MySQL que:

- Recorra todos los productos registrados en la tabla precioproducto.
- Aplique el ajuste de precios según los rangos indicados.
- Muestre al final la cantidad de productos actualizados en cada rango.

EJERCICIO 2: VERIFICACIÓN DE NIVELES DE STOCK

CASO: El área de logística requiere un mecanismo para supervisar los niveles de stock mínimo de los productos y emitir alertas sobre su estado.

ENUNCIADO

Desarrollar un procedimiento almacenado en MySQL que:

- Recorra todos los registros de la tabla producto.
- Para cada producto, muestre un mensaje:
 - Stock Crítico = si el stock mínimo es menor o igual a 10.
 - Stock Moderado = si está entre 11 y 20.
 - Stock Alto = si es mayor a 20.

EJERCICIO 3: CÁLCULO DE DESCUENTOS POR ANTIGÜEDAD

CASO: La empresa quiere incentivar el pago de facturas antiguas aplicando descuentos escalonados según su antigüedad:

- Más de 60 días → 15% de descuento.
- Entre 31 y 60 días → 10% de descuento.
- Hasta 30 días → sin descuento.

ENUNCIADO

Desarrollar una función en MySQL que:

- Reciba como parámetro la fecha de emisión de una factura.
- Devuelva el porcentaje de descuento aplicable según la antigüedad de la factura.

EJERCICIO 4: ELIMINACIÓN DE FACTURAS HUÉRFANAS

CASO: Se ha detectado que algunas facturas no tienen ítems asociados, lo que genera registros huérfanos en el sistema. Para mantener la integridad de la base de datos, es necesario eliminarlas junto a cualquier pago registrado sobre ellas.

ENUNCIADO

Desarrollar un procedimiento almacenado en MySQL que:

- Recorra todas las facturas en la tabla factura.
- Identifique aquellas que no tienen ítems en itemfactura.
- Elimine:
 - Los registros relacionados en itemrecibo.
 - La factura huérfana.
 - Los recibos que queden sin ítems asociados.

EJERCICIO 5: REPORTE DE VENTAS POR PROVINCIA

CASO: La gerencia de ventas necesita un reporte dinámico que permita consultar las ventas totales y la cantidad de facturas generadas por provincia.

ENUNCIADO

Desarrollar un procedimiento almacenado en MySQL que:

- Reciba como parámetro el nombre de una provincia.
- Busque todas las ventas (facturas) de clientes pertenecientes a esa provincia.
- Si no existen registros, devuelva un mensaje de error personalizado.
- Si existen, devuelva:
 - El número total de facturas emitidas.
 - El monto total de las ventas (sumando importes de itemrecibo).

INDICACIONES FINALES

- Los estudiantes deben formar grupos de 3 estudiantes **MÁXIMO**.
- Entregar los scripts SQL con el código de cada procedimiento o función.
- Por buenas prácticas se debe comentar el código para explicar el funcionamiento de cada bloque.
- Probar cada solución con datos reales de la base de datos.
- Se debe sustentar la solución planteada a todo el curso.