

FINAL DE GESTION DE DATOS

Apellido y Nombre	Legajo	1a	1b	2a	2b	3a	3b

A efectos de aprobar este examen final deberá sumar como mínimo 6 (seis) puntos y al menos tener un ejercicio práctico correcto. Para los ejercicios que requieran codificación SQL o PL-SQL deberá especificar de qué motor se trata.

- 1.a y b) Respuesta correcta 1 punto, respuesta incorrecta resta 1 punto.
 2.a) 2 puntos 2.b) 2 puntos. 3 a) 2 puntos 3.b) 2 puntos

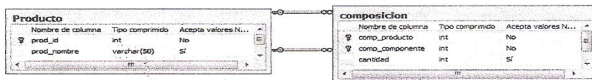
Contestar por Verdadero o Falso

- 1.a) Huffman – El árbol en el que se basa es un árbol principal derecho balanceado.
 1.b) ABB – Cuando un ABB se basa en un árbol AVL su orden de complejidad será el mejor, $n \log_2 n$.

Responder exactamente lo solicitado en la pregunta en no más de 15 renglones.

- 2.a) Describa las diferencias ente un Stored Procedure, una Función y un Trigger.
 2.b) Explique las principales características que diferencian a una Base de Datos Operacional de un Datawarehouse

Desarrollar los puntos 3.a y 3.b en base al siguiente modelo:



NOTA: las tablas no poseen triggers ni check constraint. No se conoce la calidad de los datos cargados

3.a Se le ha solicitado a un programador que desarrolle una función que dado un id de producto retorne el grado de composición del mismo. Si el elemento no es compuesto deberá retornar 0, si el elemento esta compuesto por otros/otros no compuestos deberá retornar 1 y así sucesivamente respetando los niveles del árbol de composición. El programador desarrollo la siguiente función.

```
create function dbo.Nivel_composicion (@idElem int)
returns int as
begin
    declare @id_composicion int, @id_componente int declare @grado int, @max_grado int
    declare composicion cursor
    for select c.comp_producto, c.comp_componente
        from COMPOSICION c
        where c.comp_producto=@idElem
    set @grado=0
    open composicion
    fetch composicion into @id_composicion, @id_componente
    if @@FETCH_STATUS!=0
    begin
        close composicion
        deallocate composicion
        return 0
    end
    set @max_grado=1
    while @@FETCH_STATUS=0
    begin
        if exists (select * from COMPOSICION c where c.comp_producto=@id_componente)
```

```

        set @grado=1+dbo.Nivel_composicion(@id_componente)
    end
    if(@grado>@max_grado)
    begin
        set @max_grado=@grado
    end

    fetch composicion into @id_composicion,@id_componente
end
close composicion
deallocate composicion
return @max_grado
end

```

Sabiendo que no hay errores de sintaxis y solo conociendo el DER informado, defina si la función cumple con lo solicitado. En caso que su respuesta sea positiva detalle como mejoraría la misma para que sea más performante. En caso que su rta. sea negativa detalle cual es la falla o potencial falla de la misma y como lo solucionaría.

3.b Describa detalladamente al menos dos formas en las que implementaría el siguiente control en el modelo dado "los valores comp_producto y comp_componente de una misma fila nunca pueden tener el mismo valor"