



Área personal ▶ Mis cursos ▶ [2021-1] TECNOLOGIA DE BASE DE DATOS II [1505AR,5304AR] ▶
Examen Parcial – 15/05 ▶ Examen Parcial de Base de Datos 2

Comenzado el	sábado, 15 de mayo de 2021, 18:16
Estado	Finalizado
Finalizado en	sábado, 15 de mayo de 2021, 19:48
Tiempo empleado	1 hora 31 minutos
Calificación	Sin calificar aún

Pregunta 1

Finalizado

3,00 puntos

Haz un procedimiento almacenado que reciba el nombre de un producto y muestre un listado de cuántas unidades de dicho producto compró cada cliente. Mostrar el nombre del cliente y la sumatoria de unidades compradas (Quantity), los registros deben estar ordenados descendientemente por cantidad

```
create or
alter procedure Pregunta1 @NombreProducto varchar(max)
as
begin
    select distinct c.ContactName,
                   sum(od.Quantity) over (partition by
c.ContactName,p.ProductName order by c.ContactName)
CantidadComprado
    from Customers c
         inner join Orders O on c.CustomerID = O.CustomerID
         inner join [Order Details] OD on O.OrderID =
OD.OrderID
         inner join Products P on P.ProductID = OD.ProductID
    where p.ProductName = @NombreProducto
end

execute Pregunta1 'CAMEMBERT PIERROT'
```



Pregunta **2**
Finalizado
4,00 puntos

Haz un trigger que envíe un correo al administrador del sistema cada vez que se modifiquen o eliminen funciones, procedimientos o tablas. El correo debe contener información de qué se hizo, quién lo hizo y cuándo.

- 
- 
- 
- 
- 
- 
-
- 
- 
- 
- 
- 





```
sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO

sp_configure 'Database Mail XPs', 1;
GO
RECONFIGURE
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name = 'Notificaciones',
    @description = 'Perfil para mandar correos con correo del
isur';
GO
EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name = 'Notificaciones',
    @principal_name = 'public',
    @is_default = 1;
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name = 'Isur Email',
    @description = 'Correo que se manda para hacer
notificaciones',
    @email_address = 'Correo dele Isur',
    @display_name = 'Correo Automatico',
    @mailserver_name = 'smtp.office365.com',
    @port = 587,
    @enable_ssl = 1,
    @username = 'Email utilizado para enviar el correo',
    @password = 'Contraseña del dicho correo anteriormente
nombrado';
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name = 'Notificaciones',
    @account_name = 'Isur Email',
    @sequence_number =1;
GO

-- PARA BORRAR ANTERIORES CONFIGURACIONES
-- EXECUTE msdb.dbo.sysmail_delete_profileaccount_sp @profile_name
= 'Notificaciones'
-- EXECUTE msdb.dbo.sysmail_delete_principalprofile_sp
@profile_name = 'Notificaciones'
-- EXECUTE msdb.dbo.sysmail_delete_account_sp @account_name =
'Isur Email'
-- EXECUTE msdb.dbo.sysmail_delete_profile_sp @profile_name =
'Notificaciones'

CREATE or alter TRIGGER Enviar_EmailPregunta2
    ON database for DDL_DATABASE_LEVEL_EVENTS
    AS
BEGIN
    SET NOCOUNT ON;
    declare @body varchar(max);
    set @body =
        'Se a modificado o Eliminado una
Funcion,Procedimiento o tabla para mas Informacion lea el
siguiente reporte: Se Efectuo un ' +
        (select
EVENTDATA().value('/EVENT_INSTANCE/EventType)[1]',
'nvarchar(max)')) + ' Efectuado por: ' +
        (select
EVENTDATA().value('/EVENT_INSTANCE/LoginName)[1]',
'nvarchar(max)')) + ' Se hizo en la fecha: ' +
        (select
EVENTDATA().value('/EVENT_INSTANCE/PostTime)[1]',
'nvarchar(max)'))
        EXEC msdb.dbo.sp_send_dbmail
            @profile_name = 'Notificaciones',
            @recipients = 'Correo al que se va enviar',
            @subject = 'Evento Ocacionado',
            @body = @body;
END
GO;
```





Pregunta **3**
Finalizado
2,00 puntos

Haz una TVF que reciba el nombre de una categoría y devuelva todos los productos de la categoría (nombre y precio) Usa la TVF en una consulta

```
create or
alter function Pregunta3(@NombreCategoria as varchar(max))
    returns TABLE
    as
    return
    (
        select p.ProductName, p.UnitPrice
        from Products P
            inner join Categories C2 on C2.CategoryID
= P.CategoryID
        where C2.CategoryName = @NombreCategoria
    )
go;

select * from dbo.Pregunta3( 'Beverages');
```

Pregunta **4**
Finalizado
2,00 puntos

Haz una función escalar que reciba un cliente y devuelva la fecha de la última compra de dicho cliente. Luego haz una consulta donde uses la función.

```
create or
alter function Pregunta4(@NombreCliente varchar(max))
    returns date
as
begin
    declare @FechaUltimo date;
    select @FechaUltimo= max(o.OrderDate)
    from Customers C
        inner join Orders O on C.CustomerID = O.CustomerID
    where c.ContactName = @NombreCliente
    group by c.ContactName;
    return @FechaUltimo;
end

select dbo.Pregunta4('Paul Henriot');
```



Pregunta **5**
Finalizado
3,00 puntos

Haz una TVF que reciba dos fechas y devuelva un listado de todos los días laborables (de lunes a viernes) que hay entre las dos fechas. Haz una consulta donde uses la función.

```
create or
alter function Pregunta5(@fecha_inicio DateTime,
                        @fecha_fin DateTime)

    returns table
    return
    (
        WITH FECHAS(fecha) AS (
            SELECT @fecha_inicio fecha
            UNION ALL
            SELECT DATEADD(day, 1, fecha) fecha
            FROM FECHAS
            WHERE fecha < @fecha_fin
        )
        select fecha, format(fecha, 'dddd', 'es-pe')Dia
        from FECHAS
        where format(fecha, 'dddd', 'es-pe') = 'lunes'
           or format(fecha, 'dddd', 'es-pe') = 'martes'
           or format(fecha, 'dddd', 'es-pe') = 'miércoles'
           or format(fecha, 'dddd', 'es-pe') = 'jueves'
           or format(fecha, 'dddd', 'es-pe') = 'viernes'
    );

go
select  * from dbo.Pregunta5('20210509','20210515')
```

Pregunta **6**
Finalizado
3,00 puntos

Haz un procedimiento almacenado que reciba los datos de un producto como parámetros e inserte un registro en la tabla de productos con los valores suministrados

```
create or
alter procedure Pregunta6(@NombreProducto nvarchar(80),
@Discontinuado bit, @Proveedor int=null, @Categoria int =null,
                        @CantidadPorUnidad nvarchar(40)=null,
@precio money =null, @UnidadesEnStock smallint=null,
                        @UnidadesEnOrden smallint=null,
@ReordenarNivel smallint=null)
as
begin
    insert into Products
    values (@NombreProducto, @Proveedor, @Categoria,
@CantidadPorUnidad, @precio, @UnidadesEnStock, @UnidadesEnOrden,
        @ReordenarNivel, @Discontinuado)
end

exec dbo.Pregunta6 'Papitas',false;
```

Pregunta **7**
Finalizado
3,00 puntos

Haz un trigger que se dispare cuando se eliminan ítems de órdenes de compra, si se han eliminado todos los ítems entonces SQL debe borrar automáticamente la cabecera

```
create or alter trigger Pregunta7
on [Order Details]
for delete
as
begin
declare @itemsOrderDetails bigint,@orderdetailelimado int;
select @itemsOrderDetails = count(*)
from [Order Details],
deleted
where [Order Details].OrderID = deleted.OrderID
select @orderdetailelimado=deleted.OrderID from deleted;
if (@itemsOrderDetails = 0)
begin
delete from Orders where Orders.OrderID =
@orderdetailelimado;
end
else
begin
print 'Se elimino la orden '+ @orderdetailelimado
+'Todavia quedan '+@itemsOrderDetails +' de ordenes';
end
end;
```

Actividad previa

◀ Consultas

Ir a...

Siguiente actividad

Clase en Teams ▶

📁 Resumen de retención de datos

📱 Descargar la app para dispositivos móviles

