

## Resultados Unidad 3

### Guía 3 Página 4 Ejercicio 2 Resultados 228

Cree una vista que liste la fecha, la factura, el código y nombre del vendedor, el artículo, la cantidad e importe, para lo que va del año. Rotule como FECHA, NRO\_FACTURA, CODIGO\_VENDEDOR, NOMBRE\_VENDEDOR, ARTICULO, CANTIDAD, IMPORTE.

```
create view vis_facturas_este_anio
as select fecha FECHA, f.nro_factura NRO_FACTURA, v.cod_vendedor
CODIGO_VENDEDOR, nom_vendedor NOMBRE_VENDEDOR,
descripcion ARTICULO, cantidad CANTIDAD, df.pre_unitario IMPORTE
from facturas f join detalle_facturas df on
df.nro_factura=f.nro_factura
join vendedores v on v.cod_vendedor=f.cod_vendedor
join articulos a on a.cod_articulo=df.cod_articulo
where year(fecha)=year(GETDATE())
```

```
select * from vis_facturas_este_anio
```

### Guía 3 Página 4 Ejercicio 3 Resultados 16

Modifique la vista creada en el punto anterior, agréguele la condición de que solo tome el mes pasado (mes anterior al actual) y que también muestre la dirección del vendedor.

```
alter view vis_facturas_este_anio as
select fecha FECHA, f.nro_factura NRO_FACTURA, v.cod_vendedor
CODIGO_VENDEDOR, v.nom_vendedor NOMBRE_VENDEDOR, a.descripcion ARTICULO,
df.cantidad CANTIDAD, sum(df.pre_unitario*df.cantidad) IMPORTE, v.calle
DIRECCION
from facturas f join vendedores v on v.cod_vendedor=f.cod_vendedor join
detalle_facturas df on df.nro_factura=f.nro_factura
join articulos a on a.cod_articulo=df.cod_articulo
where year(f.fecha) = year(getdate()) and month(f.fecha) =
month(getdate()) -3
group by fecha, f.nro_factura, v.cod_vendedor, v.nom_vendedor,
a.descripcion, df.cantidad, v.calle
```

```
select * from vis_facturas_este_anio
```

### Guía 3 Página 4 Ejercicio 4 Resultados a 4 b 4 c 13

Consulta las vistas según el siguiente detalle:

- Llame a la vista creada en el punto anterior pero filtrando por importes inferiores a \$120.
- Llame a la vista creada en el punto anterior filtrando para el vendedor Miranda.
- Llama a la vista creada en el punto 4 filtrando para los importes menores a 10.000.

```
Select *  
from vis_facturas_este_anio  
where IMPORTE < 1200
```

```
Select *  
from vis_facturas_este_anio  
where APELLIDO_VENDEDOR= 'Miranda'
```

```
Select *  
from vis_facturas_este_anio  
where IMPORTE < 10000
```

### Guía 3 Página 4 Ejercicio 5 Resultados 0

Elimine las vistas creadas en el punto 3

```
drop view vis_facturas_este_anio
```

### Guía 2 Página 7 Ejercicio 1 Resultados a 2

#### 1. Cree los siguientes SP:

- Detalle\_Ventas:** liste la fecha, la factura, el vendedor, el cliente, el artículo, cantidad e importe. Este SP recibirá como parámetros de E un rango de fechas.
- CantidadArt\_Cli :** este SP me debe devolver la cantidad de artículos o clientes (según se pida) que existen en la empresa.
- INS\_Vendedor:** Cree un SP que le permita insertar registros en la tabla vendedores.
- UPD\_Vendedor:** cree un SP que le permita modificar un vendedor cargado.
- DEL\_Vendedor:** cree un SP que le permita eliminar un vendedor ingresado.

```
a  
CREATE PROC Detalle_Ventas  
@E datetime  
AS  
SELECT fecha 'Fecha', f.nro_factura 'Nro de Factura', ape_vendedor+'  
' + nom_vendedor 'Vendedor', ape_cliente+' ' + nom_cliente 'Cliente',
```

```

        a.descripcion 'Articulo',df.cantidad 'Cantidad',
df.pre_unitario 'Importe'
FROM facturas f
JOIN detalle_facturas df ON f.nro_factura=df.nro_factura
JOIN articulos a ON df.cod_articulo=a.cod_articulo
JOIN vendedores v ON f.cod_vendedor=v.cod_vendedor
JOIN clientes c ON f.cod_cliente=c.cod_cliente
WHERE fecha=@E;

```

```
exec Detalle_Ventas '05/31/2021';
```

b

```

CREATE PROC CantidadArt_Cli
@tipo varchar(50)
AS
BEGIN
SELECT COUNT(cod_cliente) 'Cantidad', 'Clientes' Tipo
FROM clientes
WHERE 'Clientes'=@tipo
SELECT COUNT(cod_articulo) 'Cantidad', 'Articulos' Tipo
FROM articulos
WHERE 'Articulos'=@tipo
END;

```

```
EXEC CantidadArt_Cli 'Articulos';
```

c

```

CREATE PROC INS_Vendedor
@cod_vendedor int,
@nom_vendedor varchar(50),
@ape_vendedor varchar(50),
@calle varchar(50),
@altura int,
@cod_barrio int,
@e_mail varchar(50),
@fec_nac datetime
AS
SET @cod_vendedor=(SELECT MAX(cod_vendedor)+1 FROM vendedores)
INSERT INTO
vendedores(cod_vendedor,nom_vendedor,ape_vendedor,calle,altura,cod_barrio,[e-mail],fec_nac)

VALUES(@cod_vendedor,@nom_vendedor,@ape_vendedor,@calle,@altura,@cod_barrio,@e_mail,@fec_nac
);

```

d

```

create proc UPD_VENDEDOR
@cod_vendedor int,
@nom_vendedor varchar(20),
@ape_vendedor varchar(20),
@calle varchar(20),
@altura int,
@cod_barrio int,
@nro_tel int,
@e_mail varchar(20),
@fec_nac datetime
as
update vendedores
set nom_vendedor = @nom_vendedor,
    ape_vendedor = @ape_vendedor,
    calle = @calle,
    altura = @altura,

```

```

        cod_barrio = @cod_barrio,
        nro_tel = @nro_tel,
        [e-mail] = @e_mail,
        fec_nac = @fec_nac where cod_vendedor = @cod_vendedor

```

```

EXEC    UPD_VENDEDOR
        @cod_vendedor = 2,
        @nom_vendedor = N'Pablo',
        @ape_vendedor = N'Ortiz',
        @calle = N'9 de Julio',
        @altura = 512,
        @cod_barrio = 1,
        @nro_tel = 1111111,
        @e_mail = N'mail@mail.com',
        @fec_nac = N'1/1/2000'

```

e

```

create proc DEL_VENDEDOR
@cod_vendedor int
as
begin
delete from vendedores where cod_vendedor = @cod_vendedor
end

exec DEL_VENDEDOR 1

```

### Guía 3 Página 8 Ejercicio 2 Resultados

Modifique el SP 1-a, permitiendo que los resultados del SP puedan filtrarse por una fecha determinada, por un rango de fechas y por un rango de vendedores; según se pida.

```

ALTER PROC Detalle_Ventas
@fecha_desde date = '01/01/0001',
@fecha_hasta date = GETDATE,
@vendedor_desde int = 1,
@vendedor_hasta int = 100
AS SELECT f.fecha 'FECHA', f.nro_factura 'No FACTURA', c.ape_cliente + ' ' + c.ape_cliente
'CLIENTE', v.ape_vendedor + ' ' + v.nom_vendedor 'VENDEDOR', a.descripcion 'ARTICULO',
df.cantidad 'CANTIDAD', df.cantidad*df.pre_unitario 'IMPORTE'
FROM facturas f
JOIN detalle_facturas df ON df.nro_factura = f.nro_factura
JOIN clientes c ON c.cod_cliente = f.cod_cliente
JOIN vendedores v ON v.cod_vendedor = f.cod_vendedor
JOIN articulos a ON a.cod_articulo = df.cod_articulo
WHERE f.fecha BETWEEN @fecha_desde AND @fecha_hasta
AND v.cod_vendedor BETWEEN @vendedor_desde AND @vendedor_hasta

```

### Guía 3 Página 8 Ejercicio 3 Resultados 6

Ejecute los SP creados en el punto 1 (todos).

```

EXEC Detalle_Ventas '08/01/2022', '08/30/2022', 1, 3

```

### Guía 3 Página 8 Ejercicio 4 Resultados

Elimine los SP creados en el punto 1.

```
DROP PROC Detalle_Ventas
```

### Guía 3 Página 8 Ejercicio 5 Resultados

Cree las siguientes funciones:

- Hora: una función que les devuelva la hora del sistema en el formato HH:MM:SS (tipo carácter de 8).
- Fecha: una función que devuelva la fecha en el formato AAAMMDD (en carácter de 8), a partir de una fecha que le ingresa como parámetro (ingresa como tipo fecha).
- Dia\_Habil: función que devuelve si un día es o no hábil (considere como días no hábiles los sábados y domingos). Debe devolver 1 (hábil), 0 (no hábil)

a

```
CREATE FUNCTION f_Hora
(
)
RETURNS varchar(25)
BEGIN
    DECLARE @hora varchar(25)
    SET @hora = DATENAME(HOUR, GETDATE()) + ':' + DATENAME(MINUTE,
GETDATE()) + ':' + DATENAME(SECOND, GETDATE())
    RETURN @hora
END;

SELECT dbo.hora()
```

b

```
CREATE FUNCTION f_fecha_YMD
(@fecha date)
RETURNS varchar(25)
BEGIN
    DECLARE @f varchar(25)
    SET @f = CONVERT(varchar(25), YEAR(@fecha)) + '/' +
CONVERT(varchar(25), MONTH(@fecha)) + '/' +
CONVERT(varchar(25), DAY(@fecha))
    RETURN @f
END;

SET DATEFORMAT DMY
SELECT dbo.fecha_YMD('25/09/2022')
```

c

```
CREATE FUNCTION dia_habil
(@dia VARCHAR(50))
RETURNS BIT
BEGIN
    DECLARE @aux bit = 1
    IF @dia IN ('sábado', 'domingo')
    SET @aux = 0
    RETURN @aux
END;
```

```
SELECT dbo.dia_habil('domingo')
SELECT dbo.dia_habil('miércoles')
```

### Guía 3 Página 8 Ejercicio 6 Resultados

Modifique la f(x) 1.c, considerando solo como día no hábil el domingo.

```
ALTER FUNCTION dia_habil
(@dia VARCHAR(50))
RETURNS BIT
BEGIN
DECLARE @aux bit = 1
IF @dia IN ('domingo')
SET @aux = 0
RETURN @aux
END;

SELECT dbo.dia_habil('domingo')
SELECT dbo.dia_habil('miércoles')
```

### Guía 3 Página 8 Ejercicio 7 Resultados

Ejecute las funciones creadas en el punto 1 (todas).

```
SELECT dbo.dia_habil('domingo')
SELECT dbo.dia_habil('miércoles')
```

### Guía 3 Página 8 Ejercicio 8 Resultados

Elimine las funciones creadas en el punto 1.

```
drop function dbo.f_hora
drop function dbo.f_fecha
drop function dbo.f_esdiahabil
```