



VERDAD, BELLEZA, PROBIIDAD

UAT

Facultad de Ingeniería
Arturo Narro SillerCoordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

ACADEMIA / LÍNEA DE INVESTIGACIÓN	MATERIA / PROYECTO
Tratamiento de la información	Administración de base datos corporativas

EXAMEN No.	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)	ALUMNOS POR GRUPO DE TRABAJO
1	Aplicación de Cursores, Proc. Alm y Pivot	1	1

1. INTRODUCCIÓN

El examen lo realizara en el lenguaje los administradores de sql server y mysql. Empleando el uso de operaciones de cursores, procedimientos almacenados y pivot.

2. OBJETIVO(S)

El alumno empleara las operaciones de manejo de Cursores, procedimientos almacenados y pivot para la obtención de información específica.

3. EQUIPO REQUERIDO

1 Computadora

4. MATERIAL REQUERIDO

Aplicaciones Netbeans

5. RECOMENDACIONES DE SEGURIDAD

1.- Trabajar en al base de datos AdventureWorks2014

2.- Realice una copia del procedimiento almacenado uspGetManagerEmployeesA, asignando el nombre de AdministradorEmp, agregue el campo de JobTitle.

3.- Genere la consulta resultante usando cursores, emplea las herramientas necesarias para obtener el siguiente resultado:

Name	SalesYTD	SalesLastYear	Bonus	Ganancia
Next-Door Bike Store	2315185.611	1849640.9418	6700.00	Ganacia
Professional Sales and Service	4251368.5497	1439156.0291	2000.00	No hay Ganancia
Riders Company	3189418.3662	1997186.2037	2500.00	No hay Ganancia
The Bike Mechanics	3763178.1787	1750406.4785	4100.00	No hay Ganancia
Nationwide Supply	1421810.9242	2278548.9776	5650.00	Ganacia

4.- Agregué una columna mas al resultado anterior, llamada Diferencia Diferida la cual se calcula de la siguiente manera:

- Para los que tienen una Ganancia: obtener la diferencia entre SalesLastYear y SALESYTD y multiplicar por 0.10.
- Para los que no tienen Ganancia: se obtiene la diferencia entre SalesLastYear y SALESYTD menos el Bonus y se multiplica el resultado por 0.03.

5.- Genere la consulta en pivot para mostrar el resultado de las sales.

Customer Enero Febrero (considere solo la fecha del OrderDate)

Nombre del Customer sumatoria de la venta.(totalDUE)

Los casos anteriores replicarlos en el administrador de MYSQL.



UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

6. PROCEDIMIENTO

7. BIBLIOGRAFÍA.

ELABORÓ / DIRECTOR DE LA INVESTIGACIÓN	FIRMA	FECHA DE ELABORACIÓN
Zubiri Valdez Hedson Leonardo		29-10-2021



VERDAD, BELLEZA, PROBIIDAD

UAT

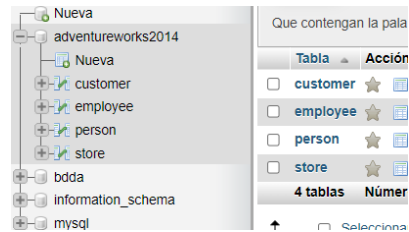
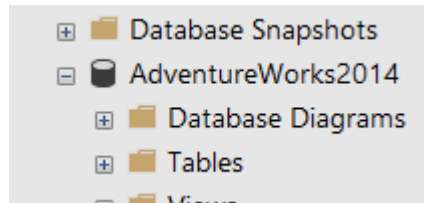


Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

REPORTE DE PRÁCTICA.

1.- Trabajar en la base de datos AdventureWorks2014



2.- Realice una copia del procedimiento almacenado uspGetManagerEmployees, asignando el nombre de AdministradorEmp, agregue el campo de JobTitle.

Código SQL Server

```
USE AdventureWorks2014
IF object_id('AdministradorEmp') is not null
    drop procedure AdministradorEmp;
GO
CREATE PROCEDURE AdministradorEmp
    (@BusinessEntityID int)
AS
BEGIN
    SET NOCOUNT ON;
    -- Use recursive query to list out all Employees required for a particular Manager
    WITH [EMP_cte]([BusinessEntityID],
        [OrganizationNode],
        [FirstName],
        [LastName],
        [RecursionLevel],
        [JobTitle]
    )
    AS (
        SELECT e.[BusinessEntityID],
            e.[OrganizationNode],
            p.[FirstName],
            p.[LastName], 0,
            e.[JobTitle]
        FROM [HumanResources].[Employee] e
            INNER JOIN [Person].[Person] p
            ON p.[BusinessEntityID] = e.[BusinessEntityID]
        WHERE e.[BusinessEntityID] = @BusinessEntityID
        UNION ALL
        SELECT e.[BusinessEntityID],
            e.[OrganizationNode],
            p.[FirstName],
            p.[LastName],
    
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

```
[RecursionLevel] + 1,  
e.[JobTitle]  
FROM [HumanResources].[Employee] e  
    INNER JOIN [EMP_cte]  
    ON e.[OrganizationNode].GetAncestor(1) = [EMP_cte].[OrganizationNode]  
    INNER JOIN [Person].[Person] p  
    ON p.[BusinessEntityID] = e.[BusinessEntityID]  
)  
  
SELECT [EMP_cte].[RecursionLevel],  
    [EMP_cte].[OrganizationNode].ToString() as [OrganizationNode],  
    p.[FirstName] AS 'ManagerFirstName',  
    p.[LastName] AS 'ManagerLastName',  
    e.[JobTitle],  
    [EMP_cte].[BusinessEntityID],  
    [EMP_cte].[FirstName],  
    [EMP_cte].[LastName], -- Outer select from the CTE  
    [EMP_cte].[JobTitle]  
FROM [EMP_cte]  
  
    INNER JOIN [HumanResources].[Employee] e  
    ON [EMP_cte].[OrganizationNode].GetAncestor(1) = e.[OrganizationNode]  
    INNER JOIN [Person].[Person] p  
    ON p.[BusinessEntityID] = e.[BusinessEntityID]  
ORDER BY [RecursionLevel], [EMP_cte].[OrganizationNode].ToString()  
OPTION (MAXRECURSION 25)  
END;  
  
GO  
EXEC AdministradorEmp 7
```

Resultado Procedimiento original

EXEC uspGetEmployeeManagers 10							
Results							
	RecursionLevel	BusinessEntityID	FirstName	LastName	OrganizationNode	ManagerFirstName	ManagerLastName
1	0	10	Michael	Raheem	/1/1/4/3/	Dylan	Miller
2	1	7	Dylan	Miller	/1/1/4/	Roberto	Tamburello
3	2	3	Roberto	Tamburello	/1/1/	Terri	Duffy

Resultado SQL Server

EXEC AdministradorEmp 7								
Results								
	RecursionLevel	OrganizationNode	ManagerFirstName	ManagerLastName	JobTitle	BusinessEntityID	FirstName	LastName
1	0	/1/1/4/	Roberto	Tamburello	Engineering Manager	7	Dylan	Miller
2	1	/1/1/4/1/	Dylan	Miller	Research and Development Manager	8	Diane	Margheim
3	1	/1/1/4/2/	Dylan	Miller	Research and Development Manager	9	Gigi	Matthew
4	1	/1/1/4/3/	Dylan	Miller	Research and Development Manager	10	Michael	Raheem



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

Código MySQL

DELIMITER \$\$

```
CREATE PROCEDURE AdministradorEmp(IN BusinessEntityID INT)
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    -- Use recursive query to list out all Employees required for a particular
    Manager
```

```
    WITH EMP_cte (BusinessEntityID,
        OrganizationNode,
        FirstName,
        LastName,
        RecursionLevel,
        JobTitle)
```

```
AS (
```

```
    SELECT e.BusinessEntityID,
        e.OrganizationNode,
        p.FirstName,
        p.LastName, 0,
        e.JobTitle
```

```
    FROM employee e
```

```
        INNER JOIN person p
```

```
        ON p.BusinessEntityID = e.BusinessEntityID
```

```
    WHERE e.BusinessEntityID = @BusinessEntityID
```

```
    UNION ALL
```

```
    SELECT e.BusinessEntityID,
        e.OrganizationNode,
        p.FirstName,
        p.LastName,
        RecursionLevel + 1,
        e.JobTitle
```

```
    FROM employee e
```

```
        INNER JOIN EMP_cte
```

```
        ON e.OrganizationNode.GetAncestor(1) = EMP_cte.OrganizationNode
```

```
        INNER JOIN person p
```

```
        ON p.BusinessEntityID = e.BusinessEntityID
```

```
)
```

```
    -- Join back to Employee to return the manager name
```

```
    SELECT EMP_cte.RecursionLevel,
        EMP_cte.OrganizationNode.ToString() as OrganizationNode,
        p.FirstName AS 'ManagerFirstName',
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

```
p.LastName AS 'ManagerLastName',
e.JobTitle,
EMP_cte.BusinessEntityID,
EMP_cte.FirstName,
EMP_cte.LastName, -- Outer select from the CTE
EMP_cte.JobTitle
FROM EMP_cte

INNER JOIN employee e
ON EMP_cte.OrganizationNode.GetAncestor(1) = e.OrganizationNode
INNER JOIN person p
ON p.BusinessEntityID = e.BusinessEntityID
ORDER BY RecursionLevel, EMP_cte.OrganizationNode.ToString()
OPTION (MAXRECURSION 25)
END$$
DELIMITER ;

CALL AdministradorEmp(3);
```

3.- Genere la consulta resultante usando cursores, emplea las herramientas necesarias para obtener el siguiente resultado:

Name	SalesYTD	SalesLastYear	Bonus	Ganancia
Next-Door Bike Store	2315185.611	1849640.9418	6700.00	Ganacia
Professional Sales and Service	4251368.5497	1439156.0291	2000.00	No hay Ganancia
Riders Company	3189418.3662	1997186.2037	2500.00	No hay Ganancia
The Bike Mechanics	3763178.1787	1750406.4785	4100.00	No hay Ganancia
Nationwide Supply	1421810.9242	2278548.9776	5650.00	Ganacia

Código SQL Server

```
IF OBJECT_ID('tempdb.dbo.#Consulta', 'U') IS NOT NULL
DROP TABLE #Consulta;
CREATE TABLE #Consulta(
Name NVARCHAR(100),
SalesYTD MONEY,
SalesLastYear MONEY,
Bonus MONEY,
Ganancia VARCHAR(20));
DECLARE
@Name NVARCHAR(100),
@SalesYTD MONEY,
@SalesLastYear MONEY,
```



VERDAD, BELLEZA, PROBIIDAD

UAT

```
@Bonus MONEY,  
@Ganancia VARCHAR(20);  
DECLARE Ej3Ganancia CURSOR FOR  
SELECT sp.SalesYTD, sp.SalesLastYear, sp.Bonus, @Ganancia, ss.Name  
FROM Sales.Store ss  
INNER JOIN Sales.SalesPerson sp ON ss.SalesPersonID = sp.BusinessEntityID  
OPEN Ej3Ganancia  
FETCH NEXT FROM Ej3Ganancia  
INTO @SalesYTD, @SalesLastYear, @Bonus, @Ganancia, @Name  
WHILE @@FETCH_STATUS = 0 BEGIN  
    IF(@Bonus > 5000) BEGIN  
        SET @Ganancia = 'Ganancia';  
    END  
    ELSE  
        SET @Ganancia = 'No hay Ganancia';  
    INSERT INTO #Consulta VALUES (@Name, @SalesYTD, @SalesLastYear, @Bonus, @Ganancia)  
    FETCH NEXT FROM Ej3Ganancia  
    INTO @SalesYTD, @SalesLastYear, @Bonus, @Ganancia, @Name  
END  
--Mostrar los resultados  
SELECT * FROM #Consulta  
CLOSE Ej3Ganancia  
DEALLOCATE Ej3Ganancia
```

Resultado SQL Server

Results		Messages			
	Name	SalesYTD	SalesLastYear	Bonus	Ganancia
1	Next-Door Bike Store	2315185.611	1849640.9418	6700.00	Ganancia
2	Professional Sales and Service	4251368.5497	1439156.0291	2000.00	No hay Ganancia
3	Riders Company	3189418.3662	1997186.2037	2500.00	No hay Ganancia
4	The Bike Mechanics	3763178.1787	1750406.4785	4100.00	No hay Ganancia
5	Nationwide Supply	1421810.9242	2278548.9776	5650.00	Ganancia
6	Area Bike Accessories	2458535.6169	2073505.9999	3550.00	No hay Ganancia
7	Bicycle Accessories and Kits	1573012.9383	1371635.3158	3500.00	No hay Ganancia
8	Clamps & Brackets Co.	3763178.1787	1750406.4785	4100.00	No hay Ganancia
9	Valley Bicycle Specialists	3189418.3662	1997186.2037	2500.00	No hay Ganancia
10	New Bikes Company	2315185.611	1849640.9418	6700.00	Ganancia
11	Vinyl and Plastic Goods Corporation	2604540.7172	2038234.6549	5000.00	No hay Ganancia
12	Top of the Line Bikes	1827066.7118	1307949.7917	75.00	No hay Ganancia
13	Fun Toys and Bikes	2458535.6169	2073505.9999	3550.00	No hay Ganancia
14	Great Bikes	1573012.9383	1371635.3158	3500.00	No hay Ganancia
15	Metropolitan Sales and Rental	3763178.1787	1750406.4785	4100.00	No hay Ganancia
16	Irregulars Outlet	1827066.7118	1307949.7917	75.00	No hay Ganancia
17	Valley Toy Store	2604540.7172	2038234.6549	5000.00	No hay Ganancia
18	Worthwhile Activity Store	2315185.611	1849640.9418	6700.00	Ganancia
19	Purchase Mart	3763178.1787	1750406.4785	4100.00	No hay Ganancia
20	Major Sport Suppliers	1573012.9383	1371635.3158	3500.00	No hay Ganancia
21	Family's Favorite Bike Shop	1453719.4653	1620276.8966	500.00	No hay Ganancia
22	Global Plaza	2315185.611	1849640.9418	6700.00	Ganancia
23	Imported and Domestic Cycles	4251368.5497	1439156.0291	2000.00	No hay Ganancia
24	Systematic Sales	2458535.6169	2073505.9999	3550.00	No hay Ganancia
25	eCommerce Bikes	2315185.611	1849640.9418	6700.00	Ganancia
26	Mountain Toy Store	3189418.3662	1997186.2037	2500.00	No hay Ganancia
27	Retail Sales and Service	3763178.1787	1750406.4785	4100.00	No hay Ganancia
28	Designated Distributors	2604540.7172	2038234.6549	5000.00	No hay Ganancia
29	Bold Bike Accessories	4251368.5497	1439156.0291	2000.00	No hay Ganancia
30	Twin Cycles	1421810.9242	2278548.9776	5650.00	Ganancia



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

Código MySQL

```
CREATE TEMPORARY TABLE Consulta(  
    Name VARCHAR(100),  
    SalesYTD DECIMAL(13,2),  
    SalesLastYear DECIMAL(13,2),  
    Bonus DECIMAL(13,2),  
    Ganancia VARCHAR(20));  
  
DROP PROCEDURE IF EXISTS CursorTest;  
DELIMITER $$  
CREATE PROCEDURE CursorTest()  
BEGIN  
    DECLARE V_Name VARCHAR(100);  
    DECLARE V_SalesYTD DECIMAL(13,2);  
    DECLARE V_SalesLastYear DECIMAL(13,2);  
    DECLARE V_Bonus DECIMAL(13,2);  
    DECLARE V_Ganancia VARCHAR(20);  
  
    DELETE FROM Consulta;  
  
    DECLARE FIN INTEGER DEFAULT 0;  
  
    DECLARE Run_CTres CURSOR FOR  
        SELECT sp.SalesYTD, sp.SalesLastYear, sp.Bonus, V_Ganancia, s.Name  
        FROM Store s  
        INNER JOIN SalesPerson sp ON s.SalesPersonID = sp.BusinessEntityID;  
  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET FIN = 1;  
  
    OPEN Run_CTres;  
    Get_RunCT: LOOP  
        FETCH Run_CTres  
        INTO V_SalesYTD, V_SalesLastYear, V_Bonus, V_Ganancia, V_Name;  
  
        IF(V_Bonus > 5000) BEGIN  
            SET V_Ganancia = 'Ganancia';  
        END  
        ELSE  
            SET V_Ganancia = 'No hay Ganancia';  
  
        INSERT INTO Consulta VALUES (V_Name, V_SalesYTD, V_SalesLastYear,  
V_Bonus, V_Ganancia);  
  
        IF FIN = 1 THEN  
            LEAVE Get_RunCT;
```




VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

```
END IF;  
END LOOP Get_RunCT;  
SELECT * FROM Consulta  
CLOSE CursorTest  
END  
$$ DELIMITER ;
```

- 4.- Agregue una columna más al resultado anterior, llamada Diferencia Diferida la cual se calcula de la siguiente manera:
- Para los que tienen una Ganancia: obtener la diferencia entre SalesLastYear y SalesYTD y multiplicar por 0.10.
 - Para los que no tienen Ganancia: se obtiene la diferencia entre SalesLastYear y SalesYTD menos el Bonus y se multiplica el resultado por 0.03.

Código SQL Server

```
IF OBJECT_ID('dbo.Consulta2', 'U') IS NOT NULL  
DROP TABLE Consulta2;  
  
CREATE TABLE Consulta2 (  
    Name NVARCHAR(100),  
    SalesYTD MONEY,  
    SalesLastYear MONEY,  
    Bonus MONEY,  
    Ganancia VARCHAR(20),  
    DiferenciaDiferida MONEY  
)  
  
DECLARE  
    @Name NVARCHAR(100),  
    @SalesYTD MONEY,  
    @SalesLastYear MONEY,  
    @Bonus MONEY,  
    @Ganancia VARCHAR(20),  
    @DD MONEY;  
  
DECLARE Ej4DiferenciaDiferida CURSOR FOR  
    SELECT Name, SalesYTD, SalesLastYear, Bonus, Ganancia, @DD  
    FROM Consulta;  
OPEN Ej4DiferenciaDiferida  
FETCH NEXT FROM Ej4DiferenciaDiferida  
INTO @Name, @SalesYTD, @SalesLastYear, @Bonus, @Ganancia, @DD  
  
WHILE @@FETCH_STATUS = 0 BEGIN  
    IF(@Ganancia = 'Ganancia') BEGIN  
        SET @DD = (@SalesLastYear - @SalesYTD) * 0.10;  
    END  
    ELSE  
        SET @DD = (@SalesLastYear - @SalesYTD - @Bonus) * 0.03;  
  
    INSERT INTO Consulta2  
    VALUES (@Name, @SalesYTD, @SalesLastYear, @Bonus, @Ganancia, @DD)
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

```

FETCH NEXT FROM Ej4DiferenciaDiferida
INTO @Name, @SalesYTD, @SalesLastYear, @Bonus, @Ganancia, @DD
END
SELECT * FROM Consulta2
CLOSE Ej4DiferenciaDiferida
DEALLOCATE Ej4DiferenciaDiferida

```

Resultado SQL Server

	Name	SalesYTD	SalesLastYear	Bonus	Ganancia	DiferenciaDiferida
1	Next-Door Bike Store	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
2	Professional Sales and Service	4251368.5497	1439156.0291	2000.00	No hay Ganancia	-84426.3756
3	Riders Company	3189418.3662	1997186.2037	2500.00	No hay Ganancia	-35841.9649
4	The Bike Mechanics	3763178.1787	1750406.4785	4100.00	No hay Ganancia	-60506.151
5	Nationwide Supply	1421810.9242	2278548.9776	5650.00	Ganancia	85673.8053
6	Area Bike Accessories	2458535.6169	2073505.9999	3550.00	No hay Ganancia	-11657.3885
7	Bicycle Accessories and Kits	1573012.9383	1371635.3158	3500.00	No hay Ganancia	-6146.3287
8	Clamps & Brackets Co.	3763178.1787	1750406.4785	4100.00	No hay Ganancia	-60506.151
9	Valley Bicycle Specialists	3189418.3662	1997186.2037	2500.00	No hay Ganancia	-35841.9649
10	New Bikes Company	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
11	Vinyl and Plastic Goods Corp...	2604540.7172	2038234.6549	5000.00	No hay Ganancia	-17139.1819
12	Top of the Line Bikes	1827066.7118	1307949.7917	75.00	No hay Ganancia	-15575.7576
13	Fun Toys and Bikes	2458535.6169	2073505.9999	3550.00	No hay Ganancia	-11657.3885
14	Great Bikes	1573012.9383	1371635.3158	3500.00	No hay Ganancia	-6146.3287
15	Metropolitan Sales and Rental	3763178.1787	1750406.4785	4100.00	No hay Ganancia	-60506.151
16	Irregulars Outlet	1827066.7118	1307949.7917	75.00	No hay Ganancia	-15575.7576
17	Valley Toy Store	2604540.7172	2038234.6549	5000.00	No hay Ganancia	-17139.1819
18	Worthwhile Activity Store	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
19	Purchase Mart	3763178.1787	1750406.4785	4100.00	No hay Ganancia	-60506.151
20	Major Sport Suppliers	1573012.9383	1371635.3158	3500.00	No hay Ganancia	-6146.3287
21	Family's Favorite Bike Shop	1453719.4653	1620276.8966	500.00	No hay Ganancia	4981.7229
22	Global Plaza	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
23	Imported and Domestic Cycles	4251368.5497	1439156.0291	2000.00	No hay Ganancia	-84426.3756
24	Systematic Sales	2458535.6169	2073505.9999	3550.00	No hay Ganancia	-11657.3885
25	eCommerce Bikes	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
26	Mountain Toy Store	3189418.3662	1997186.2037	2500.00	No hay Ganancia	-35841.9649
27	Retail Sales and Service	3763178.1787	1750406.4785	4100.00	No hay Ganancia	-60506.151
28	Designated Distributors	2604540.7172	2038234.6549	5000.00	No hay Ganancia	-17139.1819
29	Bold Bike Accessories	4251368.5497	1439156.0291	2000.00	No hay Ganancia	-84426.3756
30	Twin Cycles	1421810.9242	2278548.9776	5650.00	Ganancia	85673.8053
31	Tiny Bike Boutique	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
32	Acclaimed Bicycle Company	2315185.611	1849640.9418	6700.00	Ganancia	-46554.4669
33	Serious Cycles	1453719.4653	1620276.8966	500.00	No hay Ganancia	4981.7229

Query executed successfully.

Código MySQL

```

CREATE TEMPORARY TABLE Consulta2 (
    Name VARCHAR(100),
    SalesYTD DECIMAL(13,2),
    SalesLastYear DECIMAL(13,2),
    Bonus DECIMAL(13,2),
    Ganancia VARCHAR(20),
    DiferenciaDiferida DECIMAL(13,2)

```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio
R-OP-33-15-01

```
)  
DROP PROCEDURE IF EXISTS CursorTest2;  
DELIMITER $$  
CREATE PROCEDURE CursorTest2()  
BEGIN  
    DECLARE V_Name VARCHAR(100);  
    DECLARE V_SalesYTD DECIMAL(13,2);  
    DECLARE V_SalesLastYear DECIMAL(13,2);  
    DECLARE V_Bonus DECIMAL(13,2);  
    DECLARE V_Ganancia VARCHAR(20);  
    DECLARE V_DifDif DECIMAL(13,2);  
  
    DECLARE FIN INTEGER DEFAULT 0;  
  
    DECLARE Run_CCuatro CURSOR FOR  
        SELECT Name, SalesYTD, SalesLastYear, Bonus, Ganancia, V_DifDif  
        FROM Consulta2;  
  
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET FIN = 1;  
  
    OPEN Run_CCuatro;  
    Get_RunCC: LOOP  
        FETCH Run_CCuatro  
        INTO V_Name, V_SalesYTD, V_SalesLastYear, V_Bonus, V_Ganancia,  
        V_DifDif;  
        IF(V_Ganancia = 'Ganancia') BEGIN  
            SET V_DifDif = (V_SalesLastYear - V_SalesYTD) * 0.10;  
        END  
        ELSE  
            SET V_DifDif = (V_SalesLastYear - V_SalesYTD - V_Bonus) * 0.03;  
  
        INSERT INTO Consulta2  
        VALUES (V_Name, V_SalesYTD, V_SalesLastYear, V_Bonus, V_Ganancia,  
        V_DifDif);  
  
        IF FIN = 1 THEN  
            LEAVE Get_RunCC;  
        END IF;  
    END LOOP Get_RunCC;  
    SELECT * FROM Consulta2;  
    CLOSE CursorTest2;  
END  
$$ DELIMITER ;
```



VERDAD, BELLEZA, PROBIIDAD

UAT



Facultad de Ingeniería
Arturo Narro Siller

Coordinación de Carrera
Desarrollo de Prácticas de Laboratorio

R-OP-33-15-01

5.- Genere la consulta en pivot para mostrar el resultado de las sales.

Customer - Enero - Febrero (considere solo la fecha del OrderDate)

Nombre del Customer sumatoria de la venta.(totalDUE)

```
SELECT CustomerID, FirstName, LastName, OrderDate,
        ISNULL ([1], 0)[Enero],
        ISNULL ([2], 0)[Febrero]
FROM (
    SELECT soh.CustomerID, p.FirstName, p.LastName, soh.OrderDate, soh.TotalDue ,
    MONTH(soh.OrderDate) as m
    FROM Sales.SalesOrderHeader soh
    INNER JOIN Sales.Customer c ON soh.CustomerID = c.CustomerID
    INNER JOIN Person.Person p ON c.PersonID = p.BusinessEntityID
    WHERE MONTH(soh.OrderDate) = 1 or MONTH(soh.OrderDate) = 2
) AS results

PIVOT
(
    sum(TotalDue)
    FOR m
    IN ([1],[2])
) AS PivotTable;
```

Resultado SQL Server

82 %						
Results Messages						
	CustomerID	FirstName	LastName	OrderDate	Enero	Febrero
1	11019	Luke	Lal	2014-01-22 00:00:00.000	2.5305	0.00
2	11019	Luke	Lal	2014-02-08 00:00:00.000	0.00	27.614
3	11019	Luke	Lal	2014-02-19 00:00:00.000	0.00	26.2769
4	11032	Denise	Stone	2014-01-11 00:00:00.000	2651.1492	0.00
5	11033	Jaime	Nath	2014-01-01 00:00:00.000	2602.2529	0.00
6	11034	Ebony	Gonzalez	2014-01-06 00:00:00.000	2535.964	0.00
7	11035	Wendy	Dominguez	2014-02-10 00:00:00.000	0.00	2563.589
8	11038	Diana	Hernandez	2014-02-01 00:00:00.000	0.00	2563.589
9	11039	Marc	Martin	2014-02-20 00:00:00.000	0.00	2708.6534
10	11043	Nathan	Simmons	2014-01-18 00:00:00.000	38.675	0.00
11	11066	Grace	Butler	2014-02-19 00:00:00.000	0.00	43.0729
12	11067	Caleb	Carter	2014-01-18 00:00:00.000	6.9394	0.00
13	11073	Amy	Ye	2014-01-21 00:00:00.000	25.3819	0.00
14	11079	Donald	Gonzalez	2014-01-31 00:00:00.000	25.3819	0.00
15	11087	Tamara	Liang	2014-01-09 00:00:00.000	41.2055	0.00
16	11090	Trevor	Bryant	2012-01-07 00:00:00.000	3953.9884	0.00
17	11091	Dalton	Perez	2014-01-03 00:00:00.000	127.0308	0.00
18	11091	Dalton	Perez	2014-01-05 00:00:00.000	28.1554	0.00
19	11091	Dalton	Perez	2014-01-18 00:00:00.000	34.5534	0.00
20	11091	Dalton	Perez	2014-02-11 00:00:00.000	0.00	28.664