

Instituto Tecnológico de Estudios Superiores de Monterrey

Campus Querétaro

ISC

**Laboratorio 22: Manipulación de datos usando Stored Procedures**

Desarrollo de Aplicaciones Web y Bases de Datos

Eduardo Juárez y Ricardo Cortés

Jesús Olmos Larios - A01275595

Quinto Semestre

A modo de referencia, incluimos los esquemas de las tablas que creaste en la práctica anterior y que serán con las que trabajaremos en esta práctica:

**Materiales**(Clave, Descripción, Costo, PorcentajeImpuesto)

**Proveedores**(RFC, RazonSocial)

**Proyectos**(Numero, Denominacion)

**Entregan**(Clave, RFC, Numero, Fecha, Cantidad)

Con base en lo que se explica en la lectura sobre funciones agregadas, plantea y ejecuta las siguientes consultas, agregando los alias de columna necesarios para que los resultados resulten legibles:

* La suma de las cantidades e importe total de todas las entregas realizadas durante el 97.

SELECT E.Clave, SUM(E.Cantidad) AS 'Suma de Cantidades', SUM((E.Cantidad \* M.Costo)+((M.PorcentajeImpuesto\*(E.Cantidad \* M.Costo))/100)) AS 'Importe Total'

FROM ENTREGAN E, Materiales M

WHERE E.Clave = M.Clave AND E.Fecha >= '1997-01-01' AND E.FECHA < '1998-01-01'

GROUP BY E.Clave; 1

 1

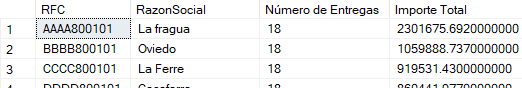
* Para cada proveedor, obtener la razón social del proveedor, número de entregas e importe total de las entregas realizadas.

SELECT E.RFC, P.RazonSocial, COUNT(E.Cantidad) AS 'Número de Entregas', SUM((E.Cantidad \* M.Costo)+((M.PorcentajeImpuesto\*(E.Cantidad \* M.Costo))/100)) AS 'Importe Total'

FROM ENTREGAN E, PROVEEDORES P, MATERIALES M

WHERE E.RFC = P.RFC AND E.Clave = M.Clave

GROUP BY E.RFC, P.RazonSocial;

 8

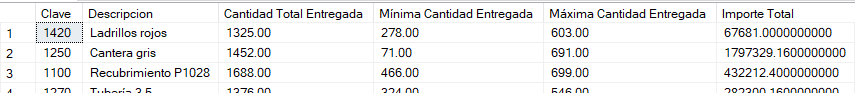
* Por cada material obtener la clave y descripción del material, la cantidad total entregada, la mínima cantidad entregada, la máxima cantidad entregada, el importe total de las entregas de aquellos materiales en los que la cantidad promedio entregada sea mayor a 400.

SELECT E.Clave, M.Descripcion, SUM(E.Cantidad) AS 'Cantidad Total Entregada', MIN(E.Cantidad) AS 'Mínima Cantidad Entregada', MAX(E.Cantidad) AS 'Máxima Cantidad Entregada', SUM((E.Cantidad \* M.Costo)+((M.PorcentajeImpuesto\*(E.Cantidad \* M.Costo))/100)) AS 'Importe Total'

FROM ENTREGAN E, MATERIALES M

GROUP BY E.Clave, M.Descripcion

HAVING AVG(E.Cantidad) > 400

630

* Para cada proveedor, indicar su razón social y mostrar la cantidad promedio de cada material entregado, detallando la clave y descripción del material, excluyendo aquellos proveedores para los que la cantidad promedio sea menor a 500.

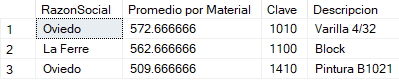
SELECT P.RazonSocial, AVG(E.Cantidad) AS 'Promedio por Material', M.Clave, M.Descripcion

FROM PROVEEDORES P, MATERIALES M, ENTREGAN E

WHERE P.RFC = E.RFC AND M.Clave = E.Clave

GROUP BY P.RazonSocial, M.Clave, M.Descripcion

HAVING AVG(E.Cantidad) >= 500

3

* Mostrar en una solo consulta los mismos datos que en la consulta anterior pero para dos grupos de proveedores: aquellos para los que la cantidad promedio entregada es menor a 370 y aquellos para los que la cantidad promedio entregada sea mayor a 450.

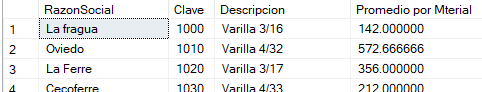
SELECT P.RazonSocial, M.Clave, M.Descripcion, AVG(E.Cantidad) AS 'Promedio por Mterial'

FROM PROVEEDORES P, MATERIALES M, ENTREGAN E

WHERE P.RFC = E.RFC AND M.Clave = E.Clave

GROUP BY P.RazonSocial, M.Clave, M.Descripcion

HAVING AVG(E.Cantidad) < 370 OR AVG(E.Cantidad) > 450

34

Utilizando la sentencia

**INSERT INTO tabla VALUES (valorcolumna1, valorcolumna2, [...] , valorcolumnan) ;**

Considerando que los valores de tipos CHAR y VARCHAR deben ir encerrados entre apóstrofes, los valores numéricos se escriben directamente y los de fecha, como '1-JAN-00' para 1o. de enero del 2000, inserta cinco nuevos materiales.

INSERT INTO MATERIALES VALUES (1440,'Tela Comex', 150, 2.88);

INSERT INTO MATERIALES VALUES (1450,'Telas Rayo', 150, 2.9);

INSERT INTO MATERIALES VALUES (1460,'Tela Prieto', 150, 2.92);

INSERT INTO MATERIALES VALUES (1470,'Telas Obo', 150, 2.94);

INSERT INTO MATERIALES VALUES (1480,'Telas Poncho', 150, 2.96);

Con base en lo que se explica en la lectura sobre consultas con roles y subconsultas, plantea y ejecuta las siguientes consultas:

* Clave y descripción de los materiales que nunca han sido entregados.

SELECT Clave, Descripcion

FROM MATERIALES

WHERE Clave NOT IN (SELECT Clave FROM ENTREGAN)

5

* Razón social de los proveedores que han realizado entregas tanto al proyecto 'Vamos México' como al proyecto 'Querétaro Limpio'.

(SELECT PV.RazonSocial

FROM PROVEEDORES PV, PROYECTOS P, ENTREGAN E

WHERE PV.RFC = E.RFC AND P.Numero = E.Numero AND E.Numero IN (SELECT Numero FROM PROYECTOS WHERE Denominacion = 'Vamos Mexico'))

UNION

(SELECT PV.RazonSocial

FROM PROVEEDORES PV, PROYECTOS P, ENTREGAN E

WHERE PV.RFC = E.RFC AND P.Numero = E.Numero AND E.Numero IN (SELECT Numero FROM PROYECTOS WHERE Denominacion = 'Queretaro limpio'))

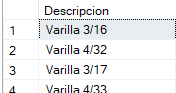
4

* Descripción de los materiales que nunca han sido entregados al proyecto 'CIT Yucatán'.

SELECT Descripcion

FROM Materiales

WHERE Clave NOT IN (SELECT E.Clave FROM PROYECTOS P, ENTREGAN E WHERE P.Numero = E.Numero AND P.Denominacion = 'CIT Yucatan')

45

* Razón social y promedio de cantidad entregada de los proveedores cuyo promedio de cantidad entregada es mayor al promedio de la cantidad entregada por el proveedor con el RFC 'VAGO780901'.

SELECT PV.RazonSocial, AVG(E.Cantidad) AS 'Promedio de la Cantidad Entregada'

FROM Entregan E, Proyectos P, Proveedores PV

WHERE P.Numero = E.Numero AND PV.RFC = E.RFC

GROUP BY PV.RazonSocial

HAVING AVG(E.Cantidad) > (SELECT AVG(Cantidad) FROM ENTREGAN WHERE RFC='VAGO780901' GROUP BY RFC)



* RFC, razón social de los proveedores que participaron en el proyecto 'Infonavit Durango' y cuyas cantidades totales entregadas en el 2000 fueron mayores a las cantidades totales entregadas en el 2001.

SET DATEFORMAT dmy

SELECT PV.RFC, PV.RazonSocial

FROM ENTREGAN E, PROYECTOS P, PROVEEDORES PV

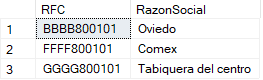
WHERE P.Numero = E.Numero AND PV.RFC = E.RFC AND P.Denominacion ='Infonavit Durango'

GROUP BY PV.RFC, PV.RazonSocial HAVING

(SELECT SUM(E.Cantidad) FROM ENTREGAN E, PROYECTOS P, PROVEEDORES AS PV WHERE P.Numero = E.Numero AND PV.RFC = E.RFC AND (E.Fecha BETWEEN '01/01/2000' AND '31/12/2000'))

>

(SELECT SUM(E.Cantidad) FROM ENTREGAN E, PROYECTOS P, PROVEEDORES AS PV WHERE P.Numero = E.Numero AND PV.RFC = E.RFC AND (E.Fecha BETWEEN '01/01/2001' AND '31/12/2001'))

3

* ¿Qué hace el primer bloque del código (bloque del IF)?

Seleccionará el name de sysobjects que se llame “creaMaterial” y sea de tipo “P”, posteriormente será eliminado.

* ¿Para qué sirve la instrucción GO?

Para detener la ejecución del script completo hasta que el código previo al GO se cumpla.

* ¿Explica que recibe como parámetro este Procedimiento y qué tabla modifica?

Recibe como parámetros establecidos por el mismo usuario: Clave, Descripción, Costo e Impuesto

Por analogía crea procedimientos almacenados con los siguientes objetivos:

* **modificaMaterial** que permite modificar un material que reciba como parámetros las columnas de la tabla materiales y actualice las columnas correspondientes con los valores recibidos, para el registro cuya llave sea la clave que se recibe como parámetro.

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'modificaMaterial' AND type = 'P')

DROP PROCEDURE modificaMaterial

GO

CREATE PROCEDURE modificaMaterial

@uclave NUMERIC(5,0),

@udescripcion VARCHAR(50),

@ucosto NUMERIC(8,2),

@uimpuesto NUMERIC(6,2)

AS

UPDATE Materiales SET Clave = @uclave, Descripcion = @udescripcion, Costo = @ucosto, PorcentajeImpuesto = @uimpuesto

GO

* **eliminaMaterial** que elimina el registro de la tabla materiales cuya llave sea la clave que se recibe como parámetro.

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'eliminaMaterial' AND type = 'P')

DROP PROCEDURE eliminaMaterial

GO

CREATE PROCEDURE eliminaMaterial

@uclave NUMERIC(5,0)

AS

DELETE FROM Materiales WHERE Clave = @uclave

GO

* Desarrollar los procedimientos (almacenados) creaProyecto , modificaproyecto y eliminaproyecto, hacer lo mismo para las tablas proveedores y entregan.

--Proyectos

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'creaProyectos' AND type = 'P')

DROP PROCEDURE creaProyecto

GO

CREATE PROCEDURE creaProyecto

@uNumero NUMERIC(5),

@uDenominacion VARCHAR(50)

AS

INSERT INTO Proyectos VALUES(@uNumero, @uDenominacion)

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'modificaProyectos' AND type = 'P')

DROP PROCEDURE modificaMaterial

GO

CREATE PROCEDURE modificaProyectos

@uNumero NUMERIC(5),

@uDenominacion VARCHAR(50)

AS

UPDATE Proyectos SET Numero = @uNumero, Denominacion = @uDenominacion WHERE Numero = @uNumero

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'eliminaProyectos' AND type = 'P')

DROP PROCEDURE eliminaProyectos

GO

CREATE PROCEDURE eliminaProyectos

@uNumero NUMERIC(5)

AS

DELETE FROM Proyectos WHERE Numero = @uNumero

GO

--Proveedores

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'creaProveedores' AND type = 'P')

DROP PROCEDURE creaProveedores

GO

CREATE PROCEDURE creaProveedores

@uRFC CHAR(13),

@uRazonSocial VARCHAR(50)

AS

INSERT INTO Proveedores VALUES(@uRFC, @uRazonSocial)

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'modificaProveedores' AND type = 'P')

DROP PROCEDURE modificaProveedores

GO

CREATE PROCEDURE modificaProveedores

@uRFC CHAR(13),

@uRazonSocial VARCHAR(50)

AS

UPDATE Proveedores SET RFC = @uRFC, RazonSocial = @uRazonSocial WHERE RFC = @uRFC

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'eliminaProveedores' AND type = 'P')

DROP PROCEDURE eliminaProveedores

GO

CREATE PROCEDURE eliminaProveedores

@uRFC CHAR(13)

AS

DELETE FROM Proveedores WHERE RFC = @uRFC

GO

--Entregan

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'creaEntrega' AND type = 'P')

DROP PROCEDURE creaEntrega

GO

CREATE PROCEDURE creaEntrega

@uClave NUMERIC(5),

@uRFC CHAR(13),

@uNumero NUMERIC(5),

@uFecha DATETIME,

@uCantidad NUMERIC(8,2)

AS

INSERT INTO Entregan VALUES(@uClave, @uRFC, @uNumero, @uFecha, @uCantidad)

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'modificaEntrega' AND type = 'P')

DROP PROCEDURE modificaEntrega

GO

CREATE PROCEDURE modificaEntrega

@uClave NUMERIC(5),

@uRFC CHAR(13),

@uNumero NUMERIC(5),

@uFecha DATETIME,

@uCantidad NUMERIC(8,2)

AS

UPDATE Entregan SET Clave = @uclave, RFC = @uRFC, Numero = @uNumero, Fecha = @uFecha, Cantidad = @uCantidad WHERE Clave = @uclave AND RFC = @uRFC AND Numero = @uNumero AND Fecha = @uFecha

GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'eliminaEntrega' AND type = 'P')

DROP PROCEDURE eliminaEntrega

GO

CREATE PROCEDURE eliminaEntrega

@uClave NUMERIC(5),

@uRFC CHAR(13),

@uNumero NUMERIC(5),

@uFecha DATETIME

AS

DELETE FROM Entregan WHERE Clave = @uclave AND RFC = @uRFC AND Numero = @uNumero AND Fecha = @uFecha

GO

* ¿Qué ventajas tienen el utilizar Stored Procedures en una aplicación cliente-servidor?

Podemos realizar consultas de una forma muy rápida.

Mejoramos la usabilidad de nuestra aplicación.

Son fáciles de eliminar y modificar.

Ahorramos líneas de código.

* ¿Qué ventajas tiene utilizar SP en un proyecto?

Las mencionadas en la parte de arriba.

Permite la asignación de responsabilidades y roles como es la edición, inserción y eliminación sin tener que conocer especificaciones hechas en SQL