



SERVICIO NACIONAL DE APRENDIZAJE SENA
SISTEMA INTEGRADO DE GESTIÓN
Procedimiento Ejecución de la Formación Profesional Integral
GUÍA DE APRENDIZAJE

Versión: 02

Código: GFPI-F-019

GUÍA DE APRENDIZAJE N° 17

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

Programa de Formación: Técnico en programación de software	Código: Versión:	228120 100		
Nombre del Proyecto: Sistema de información para la gestión Empresarial V.1.3	Código:	704330		
Fase del proyecto: No. 2 Desarrollo del proyecto				
Actividad (es) del Proyecto: Desarrollo de interfaces y Validación de campos	Actividad (es) de Aprendizaje: Diseñar consultas.(Create,Read, Update, Delete) según los requerimientos de su proyecto para ser ejecutadas desde la interfaz.	Ambiente de formación Aula de informática dotada con computadores, conexión a internet, videobeam, tablero, salida de emergencia, extintor.	MATERIALES DE FORMACIÓN	
			DEVOLUTIVO Computadores, videobeam, tablero.	CONSUMIBLE Marcadores, hojas tamaño carta
Resultados de Aprendizaje: 22050100701 Construir las tablas que hacen parte del diseño del diagrama relacional en el motor de base de datos empleando las cuatro formas de normalización	Competencia: 220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática			
22050100702 Relacionar las tablas construidas para presentar la información solicitada en el diseño.	220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática			
22050100704 Construir la matriz CRUD en el lenguaje de programación seleccionado para verificar la funcionalidad del sistema de acuerdo con el diseño entregado.	220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática			
22050103204 Interpreta el diagrama relacional para identificar el modelo de datos.	220501007 Desarrollar el sistema que cumpla con los requerimientos de la solución informática			
Duración de la guía (en horas):	Presenciales: 20			

2. INTRODUCCIÓN

Una vista es una alternativa para mostrar datos de varias tablas. Una vista es como una tabla virtual que almacena una consulta. Los datos accesibles a través de la vista no están almacenados en la base de datos como un objeto.

Entonces, una vista almacena una consulta como un objeto para utilizarse posteriormente. Las tablas consultadas en una vista se llaman tablas base. En general, se puede dar un nombre a cualquier consulta y almacenarla como una vista.

Una vista suele llamarse también tabla virtual porque los resultados que retorna y la manera de referenciarlas es la misma que para una tabla.

Ventajas

- ocultar información: permitiendo el acceso a algunos datos y manteniendo oculto el resto de la información que no se incluye en la vista. El usuario opera con los datos de una vista como si se tratara de una tabla, pudiendo modificar tales datos.
- simplificar la administración de los permisos de usuario: se pueden dar al usuario permisos para que solamente pueda acceder a los datos a través de vistas, en lugar de concederle permisos para acceder a ciertos campos, así se protegen las tablas base de cambios en su estructura.
- mejorar el rendimiento: se puede evitar tipear instrucciones repetidamente almacenando en una vista el resultado de una consulta compleja que incluya información de varias tablas.

Podemos crear vistas con: un subconjunto de registros y campos de una tabla; una unión de varias tablas; una combinación de varias tablas; un resumen estadístico de una tabla; un subconjunto de otra vista, combinación de vistas y tablas.

Una vista se define usando un "select".

La sintaxis básica parcial para crear una vista es la siguiente:

```
create view NOMBREVISTA as
  SENTENCIASSELECT
  from TABLA;
```

El contenido de una vista se muestra con un "select":

```
select *from NOMBREVISTA;
```

En el siguiente ejemplo creamos la vista "vista_empleados", que es resultado de una combinación en la cual se muestran 4 campos:

```
create view vista_empleados as
  select (apellido+' '+e.nombre) as nombre,sexo,
  s.nombre as seccion, cantidadhijos
  from empleados as e
  join secciones as s
  on código=seccion
```

Para ver la información contenida en la vista creada anteriormente tipeamos:

```
select *from vista_empleados;
```

Podemos realizar consultas a una vista como si se tratara de una tabla:

```
select seccion,count(*) as cantidad  
from vista_empleados;
```

Los nombres para vistas deben seguir las mismas reglas que cualquier identificador. Para distinguir una tabla de una vista podemos fijar una convención para darle nombres, por ejemplo, colocar el sufijo "vista" y luego el nombre de las tablas consultadas en ellas.

Los campos y expresiones de la consulta que define una vista DEBEN tener un nombre. Se debe colocar nombre de campo cuando es un campo calculado o si hay 2 campos con el mismo nombre. Note que en el ejemplo, al concatenar los campos "apellido" y "nombre" colocamos un alias; si no lo hubiésemos hecho aparecería un mensaje de error porque dicha expresión DEBE tener un encabezado, SQL Server no lo coloca por defecto.

Los nombres de los campos y expresiones de la consulta que define una vista DEBEN ser únicos (no puede haber dos campos o encabezados con igual nombre). Note que en la vista definida en el ejemplo, al campo "s.nombre" le colocamos un alias porque ya había un encabezado (el alias de la concatenación) llamado "nombre" y no pueden repetirse, si sucediera, aparecería un mensaje de error.

Otra sintaxis es la siguiente:

```
create view NOMBREVISTA (NOMBRESDEENCABEZADOS)  
as  
SENTENCIASSELECT  
from TABLA;
```

Creamos otra vista de "empleados" denominada "vista_empleados_ingreso" que almacena la cantidad de empleados por año:

```
create view vista_empleados_ingreso (fecha,cantidad)  
as  
select datepart(year,fechaingreso),count(*)  
from empleados  
group by datepart(year,fechaingreso)
```

La diferencia es que se colocan entre paréntesis los encabezados de las columnas que aparecerán en la vista. Si no los colocamos y empleamos la sintaxis vista anteriormente, se emplean los nombres de los campos o alias (que en este caso habría que agregar) colocados en el "select" que define la vista. Los nombres que se colocan entre paréntesis deben ser tantos como los campos o expresiones que se definen en la vista.

Las vistas se crean en la base de datos activa.

Al crear una vista, SQL Server verifica que existan las tablas a las que se hacen referencia en ella.

Se aconseja probar la sentencia "select" con la cual definiremos la vista antes de crearla para asegurarnos que el resultado que retorna es el imaginado.

Existen algunas restricciones para el uso de "create view", a saber:

- no puede incluir las cláusulas "compute" ni "compute by" ni la palabra clave "into";
- no se pueden crear vistas temporales ni crear vistas sobre tablas temporales.
- no se pueden asociar reglas ni valores por defecto a las vistas.
- no puede combinarse con otras instrucciones en un mismo lote.

Se pueden construir vistas sobre otras vistas.

Ejemplos:

--CREACION DE VISTA PROVEEDORES Y SUS PEDIDOS

```
CREATE VIEW VIEW_SUPPLIER_PRODUCS
AS
SELECT S.SUPPLIERID,S.COMPANYNAME,S.CONTACTNAME
,P.PRODUCTID,P.PRODUCTNAME, P.UNITPRICE
FROM SUPPLIERS AS S INNER JOIN PRODUCTS AS P
ON
S.SUPPLIERID=P.SUPPLIERID
GO
```

--CREACION DE UNA VISTA CON INSTRUCCION GROUP

```
CREATE VIEW VIEW_SUBTOTALES(CODIGO_ORDEN,SUB_TOTAL)
AS
SELECT OD.ORDERID,SUM(CONVERT(MONEY,(OD.UNITPRICE* QUANTITY*(1-DISCOUNT)/100))*100)
FROM [ORDER DETAILS] OD
GROUP BY OD.ORDERID
GO
```

--LLAMAR A LA CONSULTA ANTERIOR

```
SELECT * FROM VIEW_SUBTOTALES
```

--VER VISTAS

```
SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE='VIEW'
```

3. ESTRUCTURACION DIDACTICA DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de Reflexión inicial.

Las vistas en Bases de Datos nos permiten agrupar información de varias tablas ocultando así información que no es relevante para las consultas ¿Cree usted que esta función permite tener mayor seguridad en la información almacenada en la base de datos?

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje

Tarea de Aprendizaje

1. Qué es una vista?
2. Cuáles son los beneficios de implementar vistas en Bases de Datos?
3. En qué casos es más común la implementación de vistas?
4. Analizando la base de datos de su proyecto cree al menos 3 vistas que cumplan con los requerimientos de información analizados en su proyecto.

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

1. Adjunte su base de datos en SQL server.
2. Utilice el editor de consultas de SQL Server para implementar los ejemplos propuestos en la guía.

3.4 Actividades de transferencia del conocimiento.

Actividad 1: Cree al menos 3 vistas que cumplan con los requerimientos de información de su proyecto.

Actividad 2: Implemente dentro de las consultas de su proyecto las vistas que creo.

CONCLUSIONES

Una vista es una tabla virtual que representa los datos de una o más tablas de una forma alternativa. Para crear una nueva vista se emplea la sentencia CREATE VIEW, debe ser la primera instrucción en un lote de consultas.

Una vista sólo se puede crear en la base de datos actual.

Para ejecutar CREATE VIEW, se necesita, como mínimo, el permiso CREATE VIEW en la base de datos y el permiso ALTER en el esquema en el que se está creando la vista.

Sintaxis:

CREATE VIEW [nbEsquema.] nbVista

[(columna [,...n])]

AS (sentencia_select) [;]

nbEsquema Es el nombre del esquema al que pertenece la nueva tabla.

nbVista Es el nombre de la nueva vista. Los nombres de vistas deben seguir las reglas de los identificadores.

sentencia_select Es la instrucción SELECT que define la vista. Dicha instrucción puede utilizar más de una tabla y otras vistas. Se necesitan permisos adecuados para seleccionar los objetos a los que se hace referencia en la cláusula SELECT de la vista que se ha creado.

Una vista no tiene por qué ser un simple subconjunto de filas y de columnas de una tabla determinada. Es posible crear una vista que utilice más de una tabla u otras vistas mediante una cláusula SELECT de cualquier complejidad.

También se pueden utilizar funciones y varias instrucciones SELECT separadas por UNION o UNION ALL.

Una vista puede tener como máximo 1.024 columnas.

3.5 Actividades de evaluación.

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : ✓ Investigación	<ul style="list-style-type: none">• Plantea la necesidad de utilización de las vistas según los requerimientos del cliente.• Diseña vistas según la base d datos desarrollada.	Socialización de la investigación. Inspección resultados de la inyección de código. Juego de Roles – Verificación ejecución y verificación de triggers.
Evidencias de Desempeño: ✓ Implementación de vistas.	<ul style="list-style-type: none">• Genera ideas de cómo se pueden agrupar en vistas las consultas diseñadas.	Script creación de vistas. Lista de chequeo.

Evidencias de Producto:		
✓ Implementación vistas propuestos para su proyecto.		

4. RECURSOS PARA EL APRENDIZAJE

ACTIVIDADES DEL PROYECTO	DURACIÓN (Horas)	Materiales de formación devolutivos: (Equipos/Herramientas)		Materiales de formación (consumibles)		Talento Humano (Instructores)		AMBIENTES DE APRENDIZAJE TIPIFICADOS ESCENARIO (Aula, Laboratorio, taller, unidad productiva) y elementos y condiciones de seguridad industrial, salud ocupacional y medio ambiente
		Descripción	Cantidad	Descripción	Cantidad	Especialidad	Cantidad	
Diseñar consultas.(Create,Read, Update, Delete) para ser ejecutadas desde la interfaz.	15	Computadores, conexión a internet, Videobeam, tablero. Plataforma Blackboard	Según cantidad de aprendices	Marcadores	2	Ing. De Sistemas o Tecnólogo en Desarrollo de sistemas de información	1	Aula de informática dotada con salida de emergencia, extintor.

También cuenta con el material de apoyo documentos :

- ✓ Triggers en Transact SQL. docx

5. BIBLIOGRAFÍA/ WEBGRAFÍA

<ul style="list-style-type: none"> ✓ Bases de Datos: Enfoque práctico McGrawHill KORTH, F. Henry y SILBERCHATZ, Abraham ✓ FUNDAMENTO DE BASES DE DATOS Edit. Mc Graw Hill TechNet Microsoft

Guía de Aprendizaje

- ✓ MSDN SQL Server CREATE TRIGGER (Transact-SQL)
<http://msdn.microsoft.com/es-es/library/ms189799.aspx>
- ✓ MyGNET Triggers
<http://mygnet.net/articulos/sql/triggers.774>
- ✓ Blog Edison García Tecnologías Microsoft Sql Server
<http://mredison.wordpress.com/2008/10/26/sql-server-qu-es-un-procedimientoalmacenado/>
- ✓ Blog dedicado a SQL Server Maximiliano Damián Accotto
<http://blog.maxiacotto.com/category/TSQL.aspx>

También cuenta con el material de apoyo documentos :

- ✓ Triggers en Transact SQL. Docx

6. CONTROL DEL DOCUMENTO (ELABORADA POR)

OSCAR JAVIER ORTEGON REYES	Instructora Ingeniero de Sistemas	Integración con la media técnica	Marzo 15 de 2015	Distrito Capital
-------------------------------	---	-------------------------------------	------------------	------------------