

PRÁCTICA: 7. VISTAS**OBJETIVOS:**

Al final de esta práctica el alumno será capaz de describir y crear una vista, obtener datos de una vista, alterar la definición de una vista, insertar, borrar y modificar datos de una vista y eliminar una vista.

MATERIAL:

ORACLE9i versión para WINDOWS XP

BIBLIOGRAFIA:

ORACLE: Introduction to ORACLE 9i:SQL. Student Guide. Volume 2. 2001.

ORACLE 9I: GUIA DE APRENDIZAJE
ABBEY, MICHAEL Y COREY, MIKE Y ABRAMSON, IAN
MCGRAW-HILL / INTERAMERICANA DE ESPAÑA, S.A.

CONTENIDO:

Concepto de Vista
Vistas Simples vs Complejas
Creando una Vista
Recuperando datos desde una Vista
Modificando la Definición de una Vista
Creando Vistas Complejas
Operaciones DML sobre una Vista
Empleo de la cláusula WITH CHECK OPTION
Eliminando una Vista

Después de completar esta práctica, el alumno debería ser capaz de:

- Describir una vista,
- Crear, modificar su definición y eliminar una vista.
- Recuperar datos a través de una vista.
- Insertar, modificar y eliminar datos a través de una vista.

Esquema de Trabajo

Los ejemplos empleados en esta práctica están basados en el siguiente esquema relacional:

Empleados (id, apellidos, nombre, salario, dep)

Departamentos (nombre, edificio)

Empleados.dep -> Departamentos

En SQL:

```
CREATE TABLE Departamentos (  
    nombre      CHAR(20) PRIMARY KEY,  
    edificio     CHAR(50));
```

```
CREATE TABLE Empleados (  
    id           SMALLINT PRIMARY KEY,  
    apellidos    CHAR(40),  
    nombre       CHAR(20),  
    salario      INTEGER,  
    dep          CHAR(20) REFERENCES departamentos);
```

1. Concepto de Vista

Una vista (VIEW) es una tabla lógica (no física) que se basa en otras tablas o vistas.

Funciona como una ventana a través de la cual pueden visualizarse o modificarse datos de las tablas.

Una vista no contiene datos en si misma.

Las tablas sobre las que se define una vista se llaman tablas base.

La vista se almacena como una sentencia SELECT nominada en el diccionario de datos.

¿Por qué utilizar vistas?

- Conseguir un acceso restringido a la base de datos, ya que la vista puede mostrar sólo una porción específica.
- Sustituir consultas complejas por varias consultas más simples que emplean vistas. Por ejemplo, una selección sobre una vista definida sobre varias tablas será más fácil de realizar que si tuviéramos que utilizar la combinación (JOIN) de esas tablas para realizarla.

- Proveer independencia de datos.
- Proveer distintas maneras de ver los mismos datos, adaptados a cada usuario o aplicación.
- Permitir el acceso a grupos de usuarios de acuerdo con unos criterios concretos.

2. Vistas Simples vs Complejas

En función de las operaciones de manipulación de datos (DML), es decir, INSERT, UPDATE y DELETE, que se pueden realizar a través de ellas, existen dos categorías de vistas:

Simples:

- Extraen los datos de una sola tabla
- No contienen funciones ni grupos de datos (GROUP BY)
- Siempre pueden realizarse operaciones DML a través de ellas.

Complejas:

- Extraen los datos de múltiples tablas
- Contienen funciones o grupos de datos (GROUP BY)
- Es frecuente que no permitan operaciones DML.

3. Creando una Vista

Se emplea la sentencia CREATE VIEW, que incluye una subconsulta (subquery) para determinar los datos a ser mostrados a través de la vista.

Sintaxis:

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW <vista>
  [(<alias>[, <alias>] ... )]
  AS <subconsulta>
  [WITH CHECK OPTION [CONSTRAINT <restricción>]]
  [WITH READ ONLY [CONSTRAINT <restricción>]];
```

donde:

OR REPLACE	Se utiliza por si la vista ya estuviera creada anteriormente. En ese caso, la sustituye por la nueva definición.
FORCE	Crea la vista sin comprobar si las tablas base existen.
NO FORCE	Crea la vista sólo si las tablas base de donde se extraen los datos existen realmente (es la opción por defecto).
<vista>	Es el nombre de la vista.
<alias>	Especifica alias para las expresiones/columnas seleccionadas por la subconsulta. El número de alias debe coincidir con el número de expresiones seleccionadas por la vista.

<subconsulta>	Es una sentencia SELECT completa. Se pueden emplear alias para las columnas en la lista que sigue al SELECT.
WITH CHECK OPTION	Especifica que solo las filas accesibles para la vista pueden ser insertadas o modificadas (ver apartado 8).
<restricción>	Nombre asignado a la restricción CHECK OPTION o a la restricción READ ONLY.
WITH READ ONLY	Asegura que no podrán ejecutarse operaciones de DML a través de la vista. La vista sólo permite consultas.

Ejemplo:

Crear una vista que contiene los apellidos y nombre de los empleados del departamento "Ventas". Asegurarse que a través de ella sólo pueden modificarse, eliminarse o añadirse los empleados de dicho departamento.

```
CREATE VIEW EmpDepVentas
AS SELECT apellidos, nombre
FROM Empleados
WHERE dep="Ventas"
WITH CHECK OPTION;
```

Visualizar la estructura de una vista:

```
DESCRIBE <vista>;
```

donde:

<vista> Es el nombre de la vista.

Listar las vistas existentes:

```
SELECT * FROM USER_VIEWS;
```

Indicaciones y restricciones de uso:

- La subconsulta puede contener una sentencia SELECT de sintaxis compleja, incluyendo combinaciones (JOIN), agrupamientos (GROUP BY), y subconsultas internas.
- Pero no puede incluir una cláusula ORDER BY. Si se desea ordenar, esto deberá hacerse mediante una consulta posterior que utilice la vista (ver ejemplo en apartado 4).

Empleo de Alias

Los alias se pueden definir dentro de la subconsulta o fuera de ella, justo a continuación del nombre la vista.

Ejemplo:

Crear una vista que contiene los apellidos y nombre de los empleados del departamento “Ventas” renombrando dichas propiedades como “Last_name” y “First_name”.

Opción a) Alias dentro de la subconsulta

```
CREATE VIEW EmpDepVentas
  AS SELECT apellidos Last_name, nombre First_name
     FROM Empleados
     WHERE dep='Ventas';
```

Opción b) Alias fuera de la subconsulta

```
CREATE VIEW EmpDepVentas (Last_name, First_name)
  AS SELECT apellidos, nombre
     FROM Empleados
     WHERE dep='Ventas';
```

4. Recuperando datos desde una Vista

Una Vista se puede emplear exactamente igual que una tabla base para recuperar datos de ella mediante consultas (SELECT).

Cuando se ejecuta una vista, el ORACLE SERVER recupera la definición de la vista y ejecuta la subconsulta correspondiente.

Ejemplo:

Listar los apellidos y nombre de los empleados del departamento “Ventas” ordenados alfabéticamente.

```
SELECT apellidos, nombre
  FROM EmpDepVentas
  ORDER BY apellidos, nombre;
```

5. Modificando la Definición de una Vista

Para modificar la definición de una vista no es necesario eliminarla previamente, basta con ejecutar una nueva sentencia CREATE VIEW que incluya la opción “OR REPLACE” y la subconsulta modificada.

Ejemplo:

Cambiar la vista EmpDepVentas creada en el apartado 3 para que incluya también el salario.

```
CREATE OR REPLACE VIEW EmpDepVentas
AS SELECT apellidos, nombre, salario
FROM Empleados
WHERE dep='Ventas'
WITH CHECK OPTION;
```

6. Creando Vistas Complejas

En las vistas complejas es necesario emplear alias cuando alguna columna de la vista se deriva de una función o una expresión.

Ejemplo:

Crear una vista que muestre los salarios mínimo, máximo y medio de los empleados de cada departamento.

```
CREATE VIEW DepSalarios
(Departamento, Minimo, Maximo, Medio)
AS SELECT d.nombre, MIN(e.salario), MAX(e.salario), AVG(e.salario)
FROM Empleados e, Departamentos d
WHERE e.dep=d.nombre
GROUP BY d.nombre;
```

7. Operaciones DML sobre una Vista

Las vistas son una especie de ventanas que permiten “ver” los datos almacenados en las tablas base, pero estructurados, organizados o formateados de otra manera.

Para poder modificar datos (de las tablas base) a través de una vista es necesario que cada dato individual obtenido mediante la vista corresponda a un único dato real de una tabla base y que dicha correspondencia no se haya perdido a través de la definición de la subconsulta de la vista.

Las reglas que resumen esta situación son:

- Las vistas simples (apartado 2) siempre permiten operaciones DML (INSERT, UPDATE y DELETE).
- No es posible eliminar filas de una vista (sentencia DELETE) si su subconsulta tiene alguna de las siguientes características:
 - o Funciones de agrupamiento (SUM, AVG, etc.).
 - o Cláusula GROUP BY.
 - o La palabra clave DISTINCT.
 - o La pseudocolumna ROWNUM.

- No es posible modificar datos en una vista (sentencia UPDATE) si su subconsulta contiene:
 - o alguna de las opciones indicadas antes para restringir la operación DELETE, o
 - o Columnas formadas por expresiones (ej: salario*1'2).
- No es posible añadir datos a través de una vista (sentencia INSERT) si su subconsulta contiene:
 - o alguna de las opciones indicadas antes para restringir las operaciones DELETE y UPDATE, o
 - o Columnas con la restricción NOT NULL en su tabla base, y sin valor por defecto, que no son seleccionadas por la vista.

8. Empleo de la cláusula WITH CHECK OPTION

A través de las vistas es posible realizar controles de la integridad referencial. Por tanto, una vista puede servir de mecanismo para proteger la integridad de los datos.

La cláusula WITH CHECK OPTION garantiza que las operaciones INSERT y UPDATE realizadas a través de una vista no pueden crear filas que no serían seleccionadas por la propia vista.

Ejemplo:

Empleando la siguiente vista:

```
CREATE OR REPLACE VIEW EmpDepVentas
AS SELECT *
   FROM Empleados
   WHERE dep='Ventas'
WITH CHECK OPTION CONSTRAINT control_depVentas;
```

se generará un error si se intenta una operación INSERT o UPDATE a través de ella con un departamento que no sea "Ventas", por ejemplo, la siguiente:

```
UPDATE EmpDepVentas
   SET dep='Compras'
   WHERE id='123';
```

9. **Eliminando una Vista**

Cuando ya no se va a emplear más, una vista puede ser eliminada del esquema de la base de datos mediante la siguiente orden:

```
DROP VIEW <vista>;
```

donde:

<vista> Es el nombre de la vista.

Ejemplo:

```
DROP VIEW EmpDepVentas;
```

Restricciones de uso:

Solo el creador o un usuario con el privilegio “DROP ANY VIEW” puede eliminar una vista.

EJERCICIOS PRÁCTICA 7 - VISTAS

1. Crear una vista simple, llamada ProLSI, que contenga los datos de los profesores del área de 'Lenguajes y Sistemas Informáticos' (código 'LSI').
2. Crear una vista compleja, llamada AsigMX23, que contenga las siglas, nombre, créditos y curso de las asignaturas que se imparten en martes (día='2') o miércoles (día='3'), en segundo o tercer curso. Evitar que la vista contenga más de una vez la misma asignatura.
3. Empleando las vistas anteriores, obtener un listado alfabético de los apellidos y nombre de los profesores del área de 'Lenguajes y Sistemas Informáticos' cuya dedicación no es a tiempo completo (TC).
4. Crear una vista, llamada AreasDep, que contenga el código y nombre de cada área y el código y nombre del departamento al que pertenece. Renombrar estas columnas con las etiquetas 'codigo_area', 'nombre_area', 'codigo_departamento' y 'nombre_departamento' respectivamente.
5. Emplear la vista AreasDep para obtener el número de profesores de cada categoría, que pertenecen a cada una de las áreas del departamento cuyo nombre es 'Informática'. Encabezar el listado con las etiquetas 'area', 'categoria' y 'numero_profesores'
6. Crear una vista, llamada AsiTron, con la información de las asignaturas troncales (clase='T'), permitiendo que a través de ella se puedan hacer actualizaciones de asignaturas de cualquier clase (troncales o no).
7. Modificar la vista anterior para que a través de ella sólo se puedan realizar operaciones de actualización sólo si la clase sigue siendo troncal.
8. Modificar la vista anterior para que sólo sea de consulta y no pueda hacerse ninguna actualización a través de ella.
9. Emplear la vista anterior para crear otra vista, llamada MediaAlumnosTroncales, que muestre la media de alumnos en las asignaturas troncales, para cada curso.
10. Eliminar la vista MediaAlumnosTroncales.