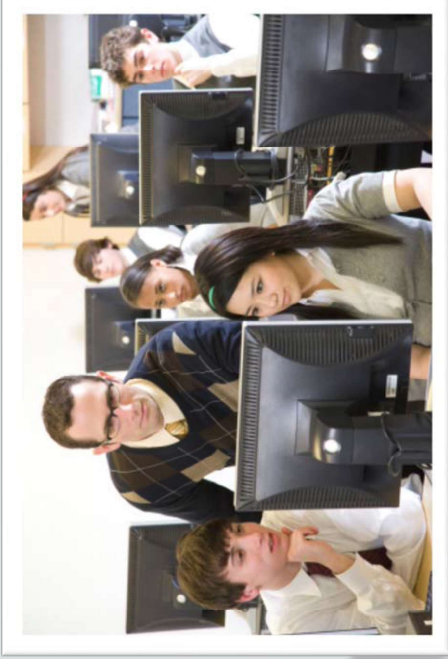


Programación de bases de datos con SQL

15-1

Creación de Vistas



Objetivos

En esta lección se abordan los siguientes objetivos:

- Enumerar tres usos de las vistas desde el punto de vista de un administrador de base de datos
- Explicar, desde una perspectiva empresarial, por qué es importante tener la capacidad de crear y utilizar subjuegos lógicos de datos derivados de una o más tablas
- Crear una vista con y sin alias de columna en la subconsulta utilizando una única tabla base

Objetivos

En esta lección se abordan los siguientes objetivos:

- Crear una vista compleja que contenga funciones de grupo para visualizar valores de dos tablas
- Recuperación de Datos de una Vista

Objetivo

- Dedique un minuto a ver lo que ha aprendido hasta ahora como alumno de Oracle Academy.
- ¿Hasta qué punto le resultaría fácil explicar lo que sabe a alguien que no ha realizado esta clase?
- Se merece una palmadita.
- El nivel de conocimiento que ha adquirido solo lo comprenden unos pocos elegidos.

Objetivo

- Imagínese ahora a sí mismo como el administrador de bases de datos de una empresa.
- ¿Qué hace cuando un gestor le pide que haga posible que pueda recuperar e introducir datos con la base de datos de la compañía?
- "Que no sea demasiado complicado. Solo quiero poder preparar informes sobre todas nuestras operaciones."

Objetivo

- ¿Deben tener estos empleados acceso a todos los datos de la compañía?
- ¿Cómo ejecutarán comandos que necesitan condiciones de unión?
- ¿Es conveniente permitir la entrada de datos de todos?
- Estas son las preguntas que como administrador de bases de datos tiene que saber responder.
- En esta sección, aprenderá a crear "vistas": representaciones virtuales de tablas personalizadas para cumplir los requisitos de usuario específicos.

Ver

- Una vista, como una tabla, es un objeto de base de datos.
- Sin embargo, las vistas no son tablas "reales".
- Son representaciones lógicas de tablas existentes o de otra vista.
- Las vistas no contienen datos propios.
- Funcionan como una ventana por la que se pueden ver o cambiar los datos de las tablas.

Ver

- Las tablas en las que se basa la vista se denominan tablas "base".
- La vista es una consulta almacenada como una sentencia SELECT en el diccionario de datos.

```
CREATE VIEW view_employees  
AS SELECT employee_id,first_name,  
last_name, email  
FROM employees  
WHERE employee_id BETWEEN 100 and 124;
```

```
SELECT *  
FROM view_employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
100	Steven	King	SKING
101	Neena	Kochhar	NKOCHHAR
102	Lex	De Haan	LDEHAAN
124	Kevin	Mourgos	KMOURGOS
103	Alexander	Hunold	AHUNOLD
104	Bruce	Ernst	BERNST
107	Diana	Lorentz	DLORENTZ

¿Por Qué Utilizar Vistas?

- Las vistas restringen el acceso a la tabla base porque la vista puede mostrar las columnas selectivas de la tabla.
- Las vistas se pueden utilizar para reducir la complejidad de la ejecución de las consultas basadas en sentencias SELECT complicadas.
- Por ejemplo, el creador de la vista puede construir sentencias de unión que recuperen datos de varias tablas.
- El usuario de la vista no ve el código subyacente ni cómo crearlo.
- El usuario, mediante la vista interactúa con la base de datos con consultas simples.

¿Por Qué Utilizar Vistas?

- Las vistas se pueden utilizar para recuperar datos de varias tablas, proporcionando independencia de datos a los usuarios.
- Los usuarios pueden ver los mismos datos de distintas formas.
- Las vistas proporcionan a los grupos de usuarios acceso a los datos según unos permisos o criterios concretos.

Creación de Vistas

- Para crear una vista, embeba una subconsulta en la sentencia CREATE VIEW.
- La sintaxis de una sentencia view es la siguiente:

```
CREATE [OR REPLACE] [FORCE| NOFORCE] VIEW view [(alias [, alias]...)] AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

Creación de Vistas

OR REPLACE	Vuelve a crear la vista si ya existe.
FORCE	Crea la vista independientemente de si existen o no las tablas base.
NOFORCE	Crea la vista solo si existe la tabla base (opción por defecto).
view_name	Especifica el nombre de la vista.
alias	Especifica un nombre para cada expresión seleccionada por la consulta de la vista.
subconsulta	Es una sentencia SELECT completa. Puede utilizar alias para las columnas de la lista SELECT. La subconsulta puede contener la sintaxis compleja SELECT.

Creación de Vistas

WITH CHECK OPTION	Especifica que las filas siguen estando accesibles para la vista después de las operaciones de inserción o actualización.
CONSTRAINT	Es el nombre asignado a la restricción CHECK OPTION.
WITH READ ONLY	Garantiza que no se pueda realizar ninguna operación DML en esta vista.

Creación de Vistas

- Ejemplo:

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id, region_id, country_name, capitol
FROM wf_countries
WHERE location LIKE '%Europe';
```

```
SELECT * FROM view_euro_countries
ORDER BY country_name;
```

COUNTRY_ID	REGION_ID	COUNTRY_NAME	CAPITOL
22	155	Bailiwick of Guernsey	Saint Peter Port
203	155	Bailiwick of Jersey	Saint Helier
387	39	Bosnia and Herzegovina	Sarajevo
420	151	Czech Republic	Prague
298	154	Faroe Islands	Torshavn
49	155	Federal Republic of Germany	Berlin
33	155	French Republic	Paris
...

Instrucciones para la Creación de Vistas

- La subconsulta que define la vista no puede contener ninguna sintaxis SELECT compleja.
- Por razones de rendimiento, la subconsulta que define la vista no debe contener ninguna cláusula ORDER BY. La cláusula ORDER BY se mejor especifica al recuperar datos de la vista.
- Puede utilizar la opción OR REPLACE para cambiar la definición de la vista sin tener que borrarla o sin necesidad de volver a otorgarle los privilegios de objeto otorgados previamente.
- Se pueden utilizar alias para los nombres de columna en la subconsulta.

Funciones de CREATE VIEW

- Se utilizan dos clasificaciones de vistas: simples y complejas.
- En la tabla se resumen las funciones de cada vista.

Función	Vistas Simples	Vistas Complejas
Número de tablas utilizadas para derivar datos	Una	Una o más
Puede contener funciones	No	Sí
Pueden contener grupos de datos	No	Sí
Puede realizar operaciones DML (INSERT, UPDATE, DELETE) en una vista	Sí	No siempre

Vista Simple

- La vista que se muestra a continuación es un ejemplo de una vista simple.
- La subconsulta deriva datos a partir de una única tabla y no contiene ninguna función de unión ni ninguna función de grupo.
- Puesto que es una vista simple, las operaciones INSERT, UPDATE, DELETE y MERGE que afectan a la tabla base se podrían realizar en la vista.

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id, country_name, capitol
FROM wf_countries
WHERE location LIKE '%Europe';
```

Vista Simple

- Los nombres de columna en la sentencia SELECT pueden tener alias como se muestra a continuación.
- Tenga en cuenta que los alias también se pueden enumerar después de la sentencia CREATE VIEW y antes de la subconsulta SELECT.

```
CREATE OR REPLACE VIEW view_euro_countries
AS SELECT country_id AS "ID", country_name AS "Country",
capitol AS "Capitol City"
FROM wf_countries
WHERE location LIKE '%Europe';
```

```
CREATE OR REPLACE VIEW view_euro_countries("ID", "Country",
"Capitol City")
AS SELECT country_id, country_name, capitol
FROM wf_countries
WHERE location LIKE '%Europe';
```

Vista Simple

- Es posible crear una vista independientemente de si existen o no las tablas base.
- Al agregar la palabra FORCE a la sentencia CREATE VIEW, se crea la vista.
- Como DBA, esta opción puede ser útil durante el desarrollo de una base de datos, especialmente si está esperando que en breve se le otorguen los privilegios necesarios para el objeto al que se hace referencia.
- La opción FORCE creará la vista a pesar de no ser válida.
- La opción NOFORCE es el valor por defecto al crear una vista.

Vista Compleja

- Las vistas complejas son vistas que pueden contener funciones de grupo y uniones.
- El siguiente ejemplo crea una vista que deriva datos de dos tablas.

```
CREATE OR REPLACE VIEW view_euro_countries  
("ID", "Country", "Capitol City", "Region")  
AS SELECT c.country_id, c.country_name, c.capitol, r.region_name  
FROM wf_countries c JOIN wf_world_regions r  
USING (region_id)  
WHERE location LIKE '%Europe';
```

```
SELECT *  
FROM view_euro_countries;
```

Vista Compleja

ID	Country	Capitol City	Region
375	Republic of Belarus	Minsk	Eastern Europe
48	Republic of Poland	Warsaw	Eastern Europe
421	Slovak Republic	Bratislava	Eastern Europe
36	Republic of Hungary	Budapest	Eastern Europe
90	Republic of Turkey	Ankara	Eastern Europe
40	Romania	Bucharest	Eastern Europe
373	Republic of Moldova	Chisinau	Eastern Europe
370	Republic of Lithuania	Vilnius	Eastern Europe
371	Republic of Latvia	Riga	Eastern Europe
372	Republic of Estonia	Tallinn	Eastern Europe
...

Vista Compleja

- Las funciones de grupo también se pueden agregar a sentencias de vistas complejas.

```
CREATE OR REPLACE VIEW view_high_pop  
("Region ID", "Highest population")  
AS SELECT region_id, MAX(population)  
FROM wf_countries  
GROUP BY region_id;
```

```
SELECT * FROM view_high_pop;
```

Region ID	Highest population
5	188078227
9	20264082
11	131859731
13	107449525
14	74777981
15	78887007
17	62660551
18	44187637
21	298444215
...	...

Modificación de Vistas

- Para modificar una vista existente sin tener que borrarla y, a continuación, volver a crearla, utilice la opción **OR REPLACE** en la sentencia **CREATE VIEW**.
- La vista antigua se sustituye por la nueva versión.
- Por ejemplo:

```
CREATE OR REPLACE VIEW view_euro_countries  
AS SELECT country_id, region_id, country_name, capital  
FROM wf_countries  
WHERE location LIKE '%Europe';
```