

# 11

## Creación de Vistas

ORACLE

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

# Objetivos

**Al finalizar esta lección, debería estar capacitado para:**

- **Describir una vista**
- **Crear y borrar una vista, así como modificar su definición**
- **Recuperar datos a través de una vista**
- **Insertar, actualizar y suprimir datos a través de una vista**
- **Crear y utilizar una vista en línea**
- **Realizar análisis de los “N principales”**

ORACLE

11-2

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Objetivo de la Lección

En esta lección, aprenderá a crear y a utilizar vistas. También aprenderá a consultar el objeto del diccionario de datos que sea necesario para recuperar información sobre vistas. Por último, aprenderá a crear y a utilizar vistas en línea, y a realizar análisis de los N principales utilizando las vistas en línea.

## Objetos de Base de Datos

Objeto	Descripción
Tabla	Unidad básica de almacenamiento; está formada por filas y columnas.
Vista	Representa lógicamente subconjuntos de datos de una o más tablas.
Secuencia	Genera valores de clave primaria.
Índice	Mejora el rendimiento de algunas consultas.
Sinónimo	Nombre alternativo para un objeto.

ORACLE

## ¿Qué es una Vista?

**Tabla EMPLOYEES:**

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	5800
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	3500
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	3100
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600
149	Zlotkey				JUL-98	ST_CLERK	2500
174	Abel				JAN-00	SA_MAN	10500
176	Taylor				MAY-96	SA_REP	11000
176	Taylor				MAR-98	SA_REP	8600
176	Kimberely	Grant	KGRANT	611.44.1844.423203	24-MAY-99	SA_REP	7000
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	6000
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000
206	William	Gietz	WGIEZT	515.123.8181	07-JUN-94	AC_ACCOUNT	8300

20 rows selected.

ORACLE

## ¿Qué es una Vista?

Puede presentar sub juegos lógicos o combinaciones de datos mediante la creación de vistas de tablas. Una vista es una tabla lógica basada en una tabla u otra vista. Una vista no contiene datos propios, sino que es muy similar a una ventana a través de la cual se pueden visualizar o cambiar datos de tablas. Las tablas en las que se basan las vistas se llaman tablas base. La vista se almacena como una sentencia SELECT en el diccionario de datos.

## ¿Para qué se Utilizan las Vistas?

- **Para restringir el acceso a los datos**
- **Para simplificar las consultas complejas**
- **Para proporcionar independencia de datos**
- **Para presentar diferentes vistas de los mismos datos**

ORACLE

11-5

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### **Ventajas de las Vistas**

- Las vistas restringen el acceso a los datos debido a que pueden mostrar columnas selectivas desde las tablas.
- Las vistas se pueden utilizar para hacer que las consultas sencillas recuperen los resultados de consultas complicadas. Por ejemplo, las vistas se pueden utilizar para consultar información de varias tablas sin necesidad de que el usuario sepa cómo escribir una sentencia de unión.
- Las vistas proporcionan independencia de datos para programas y usuarios ad-hoc. Una vista se puede utilizar para recuperar datos de diversas tablas.
- Las vistas proporcionan a los grupos de usuarios acceso a los datos de acuerdo con sus criterios específicos.

Para obtener más información, consulte *Oracle9i SQL Reference*, “CREATE VIEW”.

## Vistas Simples y Vistas Complejas

Función	Vistas Simples	Vistas Complejas
Número de tablas	Una	Una o varias
Contiene funciones	No	Sí
Contiene grupos de datos	No	Sí
Operaciones DML a través de una vista	Sí	No siempre

ORACLE

11-6

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Vistas Simples frente a Vistas Complejas

Hay dos clasificaciones para vistas: simples y complejas. La diferencia básica está relacionada con las operaciones DML (INSERT, UPDATE y DELETE).

- Una vista simple es aquella que:
  - Deriva datos de sólo una tabla.
  - No contiene funciones ni grupos de datos.
  - Puede realizar operaciones DML a través de la vista.
- Una vista compleja es aquella que:
  - Deriva datos de muchas tablas.
  - Contiene funciones o grupos de datos.
  - No siempre permite operaciones DML a través de la vista.

# Creación de una Vista

- **Embeba una subconsulta dentro de la sentencia CREATE VIEW.**

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

- **La subconsulta puede contener sintaxis SELECT compleja.**

ORACLE

11-7

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Creación de una Vista

Puede crear una vista si embebe una subconsulta dentro de la sentencia CREATE VIEW.

En la sintaxis:

OR REPLACE	vuelve a crear la vista si ya existe
FORCE	crea la vista independientemente de si las tablas base existen o no.
NOFORCE	crea la vista sólo si las tablas base existen (es el valor por defecto).
<i>view</i>	es el nombre de la vista
<i>alias</i>	specifies names for the expressions selected by the view's (El número de alias debe coincidir con el número de expresiones seleccionadas en la vista.)
<i>subquery</i>	es una sentencia SELECT completa (puede utilizar alias para las columnas de la lista SELECT).
WITH CHECK OPTION	especifica que sólo las filas accesibles a la vista se pueden insertar o actualizar.
<i>constraint</i>	es el nombre asignado a la restricción CHECK OPTION.
WITH READ ONLY	asegura que no se pueda realizar ninguna operación DML en esta vista.

## Creación de una Vista

- Cree una vista, EMPVU80, que contenga detalles de los empleados del departamento 80.

```
CREATE VIEW empvu80
AS SELECT employee_id, last_name, salary
FROM employees
WHERE department_id = 80;
View created.
```

- Describa la estructura de la vista utilizando el comando DESCRIBE de iSQL\*Plus.

```
DESCRIBE empvu80
```

ORACLE

11-8

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de una Vista (continuación)

En el ejemplo de la transparencia se crea una vista que contiene el número de empleado, el apellido y el salario de cada empleado del departamento 80.

Puede visualizar la estructura de la vista utilizando el comando DESCRIBE de iSQL\*Plus.

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)

Instrucciones para la creación de una vista:

- La subconsulta que define una vista puede contener sintaxis SELECT compleja, incluyendo uniones, grupos y subconsultas.
- La subconsulta que define la vista no puede contener una cláusula ORDER BY. La cláusula ORDER BY se especifica cuando se recuperan datos de la vista.
- Si no especifica un nombre de restricción para una vista creada con WITH CHECK OPTION, el sistema asigna un nombre por defecto con el formato SYS\_Cn.
- Puede utilizar la opción OR REPLACE para cambiar la definición de la vista sin borrarla y volver a crearla o volver a otorgar privilegios de objeto previamente otorgados.



## Creación de una Vista

- Cree una vista utilizando alias de columna en la subconsulta.

```
CREATE VIEW salvu50
AS SELECT employee_id ID_NUMBER, last_name NAME,
          salary*12 ANN_SALARY
FROM employees
WHERE department_id = 50;
View created.
```

- Seleccione las columnas desde esta vista mediante los nombres de alias especificados.

ORACLE

11-9

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de una Vista (continuación)

Puede controlar los nombres de columna incluyendo alias de columna dentro de la subconsulta.

En el ejemplo de la transparencia se crea una vista que contiene el número de empleado (EMPLOYEE\_ID) con el alias ID\_NUMBER, nombre (LAST\_NAME) con el alias NAME y salario anual (SALARY) con el alias ANN\_SALARY para cada empleado del departamento 50.

Otra alternativa consiste en utilizar un alias después de la sentencia CREATE y antes de la subconsulta SELECT. El número de alias enumerado debe coincidir con el número de expresiones seleccionadas en la subconsulta.

```
CREATE VIEW salvu50 (ID_NUMBER, NAME, ANN_SALARY)
AS SELECT employee_id, last_name, salary*12
FROM employees
WHERE department_id = 50;
View created.
```

## Recuperación de Datos de una Vista

```
SELECT *  
FROM salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

ORACLE

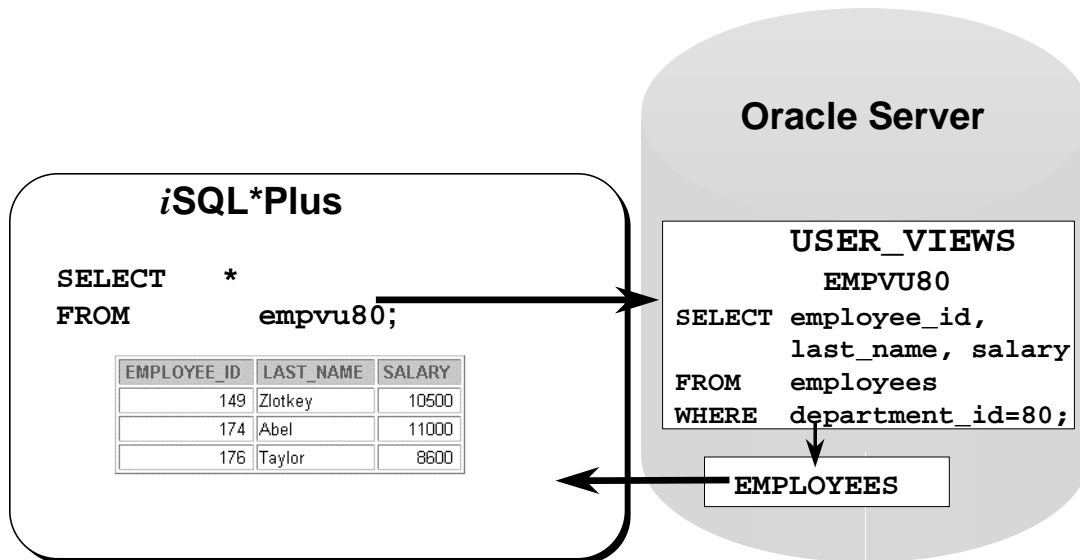
11-10

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Recuperación de Datos de una Vista

Puede recuperar datos de una vista de la misma forma que los recupera de una tabla. Puede visualizar los contenidos de toda la vista o sólo de columnas y filas específicas.

## Consulta de una Vista



ORACLE

11-11

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Vistas del Diccionario de Datos

Una vez que se haya creado la vista, puede consultar la vista del diccionario de datos llamada `USER_VIEWS` para ver el nombre y la definición de la vista. El texto de la sentencia `SELECT` que constituye la vista se almacena en una columna `LONG`.

### Acceso a Datos Utilizando Vistas

Cuando accede a datos utilizando una vista, Oracle Server realiza las siguientes operaciones:

1. Recupera la definición de la vista de la tabla `USER_VIEWS` del diccionario de datos.
2. Comprueba los privilegios de acceso para la tabla base de la vista.
3. Convierte la consulta de la vista en una operación equivalente en las tablas o tabla base subyacente. Dicho de otro modo, los datos se recuperan de las tablas base o se realiza una actualización de ellas.

## Modificación de una Vista

- **Modifique la vista EMPVU80 utilizando la cláusula CREATE OR REPLACE VIEW. Agregue un alias para cada nombre de columna.**

```
CREATE OR REPLACE VIEW empvu80
(id_number, name, sal, department_id)
AS SELECT  employee_id, first_name || ' ' || last_name,
           salary, department_id
FROM      employees
WHERE     department_id = 80;
View created.
```

- **Los alias de columna de la cláusula CREATE VIEW están enumerados en el mismo orden que las columnas de la subconsulta.**

ORACLE

### Modificación de una Vista

Con la opción OR REPLACE, se puede crear una vista incluso si ya existe otra con el mismo nombre, sustituyendo así la versión antigua de la vista para su propietario. Esto significa que la vista se puede modificar sin necesidad de borrar, volver a crear u otorgar, de nuevo, privilegios de objeto.

**Nota:** Cuando asigne alias de columna en la cláusula CREATE VIEW, recuerde que los alias se enumeran en el mismo orden que las columnas de la subconsulta.

## Creación de una Vista Compleja

**Cree una vista compleja que contenga funciones de grupo para visualizar los valores de dos tablas.**

```
CREATE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT    d.department_name, MIN(e.salary),
             MAX(e.salary),AVG(e.salary)
  FROM      employees e, departments d
  WHERE     e.department_id = d.department_id
  GROUP BY  d.department_name;
View created.
```

ORACLE

11-13

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de una Vista Compleja

En el ejemplo de la transparencia se crea una vista compleja de nombres de departamento, salarios mínimos, salarios máximos y salarios medios por departamento. Observe que se han especificado nombres alternativos para la vista. Esto es un requisito si alguna columna de la vista se deriva de una función o una expresión.

Puede visualizar la estructura de la vista utilizando el comando DESCRIBE de *iSQL\*Plus*. Visualice los contenidos de la vista emitiendo una sentencia SELECT.

```
SELECT  *
FROM    dept_sum_vu;
```

NAME	MINSAL	MAXSAL	AVGSAL
Accounting	8300	12000	10150
Administration	4400	4400	4400
Executive	17000	24000	19333.3333
IT	4200	9000	6400
Marketing	6000	13000	9500
Sales	8600	11000	10033.3333
Shipping	2500	5800	3500

7 rows selected.

## Reglas para la Realización de Operaciones DML en una Vista

- Puede realizar operaciones DML en vistas simples.
- No puede eliminar una fila si la vista contiene:
  - Funciones de grupo
  - Una cláusula `GROUP BY`
  - La palabra clave `DISTINCT`
  - La palabra clave `ROWNUM` de pseudocolumna

ORACLE

11-14

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Realización de Operaciones DML en una Vista

Puede realizar operaciones DML en datos a través de una vista si dichas operaciones siguen ciertas reglas.

Puede eliminar una fila de una vista a no ser que contenga:

- Funciones de grupo
- Una cláusula `GROUP BY`
- La palabra clave `DISTINCT`
- La palabra clave `ROWNUM` de pseudocolumna

## Reglas para la Realización de Operaciones DML en una Vista

No puede modificar datos en una vista si ésta contiene:

- Funciones de grupo
- Una cláusula `GROUP BY`
- La palabra clave `DISTINCT`
- La palabra clave `ROWNUM` de pseudocolumna
- Columnas definidas por expresiones

ORACLE

11-15

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Realización de Operaciones DML en una Vista (continuación)

Puede modificar los datos a través de una vista a no ser que ésta contenga alguna de las condiciones mencionadas en la transparencia anterior o columnas definidas por expresiones (por ejemplo, `SALARY * 1.2`).

## Reglas para la Realización de Operaciones DML en una Vista

No puede agregar datos a través de una vista si ésta incluye:

- Funciones de grupo
- Una cláusula `GROUP BY`
- La palabra clave `DISTINCT`
- La palabra clave `ROWNUM` de pseudocolumna
- Columnas definidas por expresiones
- Columnas `NOT NULL` en las tablas base que no estén seleccionadas por la vista

ORACLE

11-16

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Realización de Operaciones DML en una Vista (continuación)

Puede agregar datos a través de una vista a no ser que ésta contenga alguno de los elementos enumerados en la transparencia o haya columnas `NOT NULL` sin valores por defecto en la tabla base que no estén seleccionadas por la vista. Todos los valores requeridos deben estar presentes en la vista. Recuerde que está agregando valores directamente en la tabla subyacente *a través de* la vista.

Para obtener más información, consulte *Oracle9i SQL Reference*, “CREATE VIEW”.



## Uso de la Cláusula WITH CHECK OPTION

- Se puede asegurar de que las operaciones DML realizadas en la vista permanecen dentro del dominio de la misma utilizando la cláusula WITH CHECK OPTION.

```
CREATE OR REPLACE VIEW empvu20
AS SELECT *
   FROM   employees
  WHERE   department_id = 20
  WITH CHECK OPTION CONSTRAINT empvu20_ck ;
View created.
```

- Cualquier intento de cambiar el número de departamento para una fila en la vista falla porque viola la restricción WITH CHECK OPTION.

ORACLE

11-17

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Uso de la Cláusula WITH CHECK OPTION

Es posible realizar comprobaciones de integridad referencial a través de las vistas. También puede forzar las restricciones en el nivel de la base de datos. La vista se puede utilizar para proteger la integridad de datos, pero el uso está muy limitado.

La cláusula WITH CHECK OPTION especifica que las inserciones (INSERT) y las actualizaciones (UPDATE) realizadas a través de la vista no pueden crear filas que la vista no pueda seleccionar y, por tanto, permite que las restricciones de integridad y las comprobaciones de validación de datos se fuercen en los datos que se insertan o se actualizan.

Si hay un intento de realizar operaciones DML en filas que la vista no haya seleccionado, se muestra un mensaje de error junto con el nombre de la restricción, si se ha especificado.

```
UPDATE empvu20
   SET   department_id = 10
  WHERE  employee_id = 201;
UPDATE empvu20
   *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

**Nota:** No se actualiza ninguna fila ya que si el número de departamento tuviera que cambiar a 10, la vista ya no podría mostrar a ese empleado. Por lo tanto, con la cláusula WITH CHECK OPTION, la vista sólo puede mostrar a los empleados del departamento 20 y no permite que el número de departamento de dichos empleados cambie a través de la vista.

## Denegación de Operaciones DML

- Puede asegurarse de que no se produce ninguna operación DML al agregar la opción `WITH READ ONLY` a la definición de la vista.
- Cualquier intento de realizar un DML en una fila de la vista da como resultado un error de Oracle Server.

ORACLE

### Denegación de Operaciones DML

Puede asegurarse de que no se produce ninguna operación DML en la vista si la crea con la opción `WITH READ ONLY`. En el ejemplo de la transparencia se modifica la vista `EMPVU10` para evitar cualquier operación DML en la vista.

## Denegación de Operaciones DML

```
CREATE OR REPLACE VIEW empvu10
  (employee_number, employee_name, job_title)
AS SELECT  employee_id, last_name, job_id
  FROM      employees
  WHERE     department_id = 10
  WITH READ ONLY;
View created.
```

ORACLE

11-19

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Denegación de Operaciones DML

Cualquier intento de eliminar una fila de una vista con una restricción de sólo lectura da como resultado un error .

```
DELETE FROM empvu10
WHERE  employee_number = 200;
DELETE FROM empvu10
      *
ERROR at line 1:
ORA-01752: cannot delete from view without exactly one key-
preserved table
```

Cualquier intento de insertar o modificar una fila utilizando la vista con una restricción de sólo lectura da como resultado un error de Oracle Server:

```
01733: virtual column not allowed here.
```

## Eliminación de una Vista

**Puede eliminar una vista sin perder datos ya que la vista se basa en tablas subyacentes de la base de datos.**

```
DROP VIEW view;
```

```
DROP VIEW empvu80;  
View dropped.
```

ORACLE

11-20

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Eliminación de una Vista

Puede utilizar la sentencia DROP VIEW para eliminar una vista. Esta sentencia elimina la definición de la vista de la base de datos. La eliminación de vistas no tiene ningún efecto en las tablas en las que se basaba la vista. Las vistas u otras aplicaciones basadas en vistas suprimidas se invalidan. Solamente pueden eliminar una vista el creador o un usuario con el privilegio DROP ANY VIEW.

En la sintaxis:

*view*                      es el nombre de la vista.

## Vistas en Línea

- Una vista en línea es una subconsulta con un alias (o nombre de correlación) que puede utilizar dentro de una sentencia SQL.
- Una subconsulta especificada en la cláusula **FROM** de la consulta principal es un ejemplo de vista en línea.
- Una vista en línea no es un objeto de esquema.

ORACLE

11-21

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Vistas en Línea

Una vista en línea se crea colocando una subconsulta en la cláusula **FROM** y asignando un alias a la subconsulta. Ésta define un origen de datos al que se puede hacer referencia en la consulta principal. En el siguiente ejemplo, la vista en línea **b** devuelve los detalles de todos los números de departamento y el salario máximo de cada departamento de la tabla **EMPLOYEES**. La cláusula **WHERE** **a.department\_id = b.department\_id AND a.salary < b.maxsal** de la consulta principal muestra nombres de empleado, salarios, números de departamento y salarios máximos para todos los empleados que ganan menos que el salario máximo de su departamento

```
SELECT  a.last_name, a.salary, a.department_id, b.maxsal
FROM    employees a, (SELECT  department_id, max(salary) maxsal
                      FROM    employees
                      GROUP BY department_id) b
WHERE   a.department_id = b.department_id
AND     a.salary < b.maxsal;
```

LAST_NAME	SALARY	DEPARTMENT_ID	MAXSAL
Fay	6000	20	13000
Rajs	3500	50	5800
Davies	3100	50	5800
Matos	2600	50	5800
Vargas	2500	50	5800

■ ■ ■

12 rows selected.

## Análisis de los N Principales

- Las consultas de N principales solicitan los  $n$  valores más grandes o más pequeños de una columna. Por ejemplo:
  - ¿Cuáles son los diez productos más vendidos?
  - ¿Cuáles son los diez productos menos vendidos?
- Tanto los juegos de valores más grandes como los más pequeños se consideran consultas de N principales.

ORACLE

11-22

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Análisis de los “N Principales”

Las consultas de N principales son útiles en supuestos en los que es necesario mostrar solamente los  $n$  registros principales o los  $n$  menos importantes de una tabla basada en una condición. Este juego de resultados se puede utilizar para otros análisis. Por ejemplo, utilizando análisis de los N principales puede realizar los siguientes tipos de consulta:

- Los tres empleados de la compañía que más ganan
- Las cuatro contrataciones más recientes de la compañía
- Los dos representantes de ventas que han vendido el máximo número de productos
- Los tres productos que han tenido las máximas ventas en los últimos seis meses

## Ejecución de Análisis de los “N Principales”

La estructura de nivel superior de una consulta de análisis de N principales es:

```
SELECT [column_list], ROWNUM
FROM   (SELECT [column_list]
        FROM table
        ORDER BY Top-N_column)
WHERE  ROWNUM <= N;
```

ORACLE

11-23

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Ejecución de Análisis de los “N Principales”

Las consultas de N principales utilizan una estructura de consulta anidada consistente con los elementos que se describen a continuación:

- Una subconsulta o una vista en línea para generar la lista ordenada de datos. La subconsulta o la vista en línea incluye la cláusula `ORDER BY` para asegurar que la clasificación está en el orden deseado. Para que los resultados recuperen los valores más grandes es necesario un parámetro `DESC`.
- Una consulta externa para limitar el número de filas en el juego de resultados final. La consulta externa incluye los siguientes componentes:
  - La pseudocolumna `ROWNUM`, que asigna un valor secuencial que comienza por 1 a cada una de las filas devuelta en la subconsulta.
  - Una cláusula `WHERE`, que especifica las *n* filas que se van a devolver. La cláusula externa `WHERE` debe utilizar un operador `<` o `<=`.

## Ejemplo de Análisis de los N Principales

Para visualizar los salarios y los nombres de los tres empleados que más ganan desde la tabla **EMPLOYEES**:

```
SELECT ROWNUM as RANK, last_name, salary
FROM (SELECT last_name,salary FROM employees
      ORDER BY salary DESC)
WHERE ROWNUM <= 3;
```

RANK	LAST_NAME	SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000

ORACLE

11-24

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Ejemplo de Análisis de los "N Principales"

El ejemplo de la transparencia muestra cómo visualizar los nombres y los salarios de los tres empleados que más ganan desde la tabla **EMPLOYEES**. La subconsulta devuelve los detalles de los salarios y los nombres de todos los empleados de la tabla **EMPLOYEES**, en orden descendente de salarios. La cláusula **WHERE ROWNUM < 3** de la consulta principal asegura que sólo se mostrarán los tres primeros registros de este juego de resultados.

Éste es otro ejemplo de análisis de los N principales que utiliza una vista en línea. En el ejemplo siguiente se utiliza la vista en línea **E** para mostrar los cuatro empleados más antiguos de la compañía.

```
SELECT ROWNUM as SENIOR,E.last_name, E.hire_date
FROM (SELECT last_name,hire_date FROM employees
      ORDER BY hire_date)E
WHERE rownum <= 4;
```

SENIOR	LAST_NAME	HIRE_DATE
1	King	17-JUN-87
2	Whalen	17-SEP-87
3	Kochhar	21-SEP-89
4	Hunold	03-JAN-90



# Resumen

**En esta lección, debería haber aprendido que una vista se deriva de los datos de otras tablas o vistas y proporciona las siguientes ventajas:**

- **Restringe el acceso a la base de datos.**
- **Simplifica las consultas.**
- **Proporciona independencia de datos.**
- **Proporciona varias vistas de los mismos datos.**
- **Se puede borrar sin eliminar los datos subyacentes.**
- **Una vista en línea es una subconsulta con un nombre de alias.**
- **Los análisis de los N principales se pueden realizar utilizando consultas externas y subconsultas.**

ORACLE

11-25

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## ¿Qué es una Vista?

Una vista se basa en una tabla u otra vista y actúa como una ventana a través de la cual se pueden visualizar o cambiar datos de tabla. Una vista no contiene datos. La definición de la vista se almacena en el diccionario de datos. Puede visualizar la definición de la vista en la tabla USER\_VIEWS del diccionario de datos.

## Ventajas de las Vistas

- Restringen el acceso a la base de datos.
- Simplifican las consultas.
- Proporcionan independencia de datos.
- Proporcionan varias vistas de los mismos datos.
- Se pueden eliminar sin afectar a los datos subyacentes.

## Opciones de Vista

- Puede ser una vista simple basada en una tabla.
- Puede ser una vista compleja basada en más de una tabla o puede contener grupos de funciones.
- Puede sustituir a otras vistas con el mismo nombre.
- Puede contener una restricción de control.
- Puede ser de sólo lectura.

# Visión General de la Práctica 11

**Esta práctica cubre los siguientes temas:**

- **Creación de una vista simple**
- **Creación de una vista compleja**
- **Creación de una vista con una restricción de control**
- **Intento de modificar datos de la vista**
- **Visualización de definiciones de vista**
- **Eliminación de vistas**

ORACLE

11-26

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Visión General de la Práctica 11

En esta práctica, creará vistas simples y complejas e intentará ejecutar en ellas sentencias DML.

## Práctica 11

1. Cree una vista llamada EMPLOYEES\_VU basada en los números y los nombres de empleado y los números de departamento de la tabla EMPLOYEES. Cambie la cabecera para el nombre de empleado a EMPLOYEE.
2. Visualice los contenidos de la vista EMPLOYEES\_VU.

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60
...		
206	Gietz	110

20 rows selected.

3. Seleccione el nombre de vista y el texto de la vista del diccionario de datos USER\_VIEWS.

**Nota:** Ya existe otra vista. EMP\_DETAILS\_VIEW se creó como parte del esquema.

**Nota:** Para ver más contenidos de una columna LONG, utilice el comando SET LONG n, de iSQL\*Plus, donde n es el valor del número de caracteres de la columna LONG que desea ver.

VIEW_NAME	TEXT
EMPLOYEES_VU	SELECT employee_id, last_name employee, department_id FROM employees
EMP_DETAILS_VIEW	SELECT e.employee_id, e.job_id, e.manager_id, e.department_id, d.location_id, l.country_id, e.first_name, e.last_name, e.salary, e.commission_pct, d.department_name, j.job_title, l.city, l.state_province, c.country_name, r.region_name FROM employees e, departments d, jobs j, locations l, countries c, regions r WHERE e.department_id = d.department_id AND d.location_id = l.location_id AND l.country_id = c.country_id AND c.region_id = r.region_id AND j.job_id = e.job_id WITH READ ONLY

4. Utilizando la vista EMPLOYEES\_VU, introduzca una consulta para mostrar todos los nombres de empleado y los números de departamento.

EMPLOYEE	DEPARTMENT_ID
King	90
Kochhar	90
...	
Gietz	110

20 rows selected.

### Práctica 11 (continuación)

5. Cree una vista llamada DEPT50 que contenga los números y los apellidos de los empleados y los números de departamento para todos los empleados del departamento 50. Etiquete las columnas de la vista como EMPNO, EMPLOYEE y DEPTNO. No permita que se reasigne un empleado a otro departamento a través de la vista.
6. Visualice la estructura y los contenidos de la vista DEPT50.

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(6)
EMPLOYEE	NOT NULL	VARCHAR2(25)
DEPTNO		NUMBER(4)

EMPNO	EMPLOYEE	DEPTNO
124	Mourgos	50
141	Rajs	50
142	Davies	50
143	Matos	50
144	Vargas	50

7. Intente reasignar a Matos al departamento 80.

Si tiene tiempo, complete el siguiente ejercicio:

8. Cree una vista llamada SALARY\_VU basándose en los apellidos de los empleados, los nombres de departamento, los salarios y los grados de salario para todos los empleados. Utilice las tablas EMPLOYEES, DEPARTMENTS y JOB\_GRADES. Etiquete las columnas Employee, Department, Salary y Grade, respectivamente.

# 12

## Otros Objetos de Base de Datos

ORACLE

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

# Objetivos

**Al finalizar esta lección, debería estar capacitado para:**

- **Crear, mantener y utilizar secuencias**
- **Crear y mantener índices**
- **Crear sinónimos públicos y privados**

ORACLE®

12-2

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Objetivo de la Lección

En esta lección, aprenderá a crear y mantener algunos de los otros objetos de la base de datos utilizados habitualmente. Entre estos objetos se incluyen secuencias, índices y sinónimos

## Objetos de Base de Datos

Objeto	Descripción
Tabla	Unidad básica de almacenamiento; está formada por filas y columnas.
Vista	Representa lógicamente subconjuntos de datos de una o más tablas.
Secuencia	Genera valores de clave primaria.
Índice	Mejora el rendimiento de algunas consultas.
Sinónimo	Nombre alternativo para un objeto.

ORACLE

12-3

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Objetos de Base de Datos

Muchas aplicaciones requieren el uso de números únicos como valores de clave primaria. Puede crear un código en la aplicación para manejar este requisito o utilizar una secuencia para generar números únicos.

Si desea mejorar el rendimiento de algunas consultas, debería considerar la creación de un índice. También puede utilizar índices para forzar la unicidad en una columna o en una recopilación de columnas.

Puede proporcionar nombres alternativos a los objetos utilizando sinónimos.

# ¿Qué es una Secuencia?

## **Una secuencia:**

- **Genera automáticamente números únicos.**
- **Es un objeto compartible.**
- **Normalmente se utiliza para crear un valor de clave primaria.**
- **Sustituye al código de aplicación.**
- **Acelera la eficacia del acceso a los valores de secuencia al almacenarse en memoria caché.**

ORACLE

12-4

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## **¿Qué es una Secuencia?**

Una secuencia es un objeto de base de datos creado por un usuario que se puede compartir con varios usuarios para generar enteros únicos.

Un uso típico de las secuencias es la creación de un valor de clave primaria, que debe ser único para cada fila. Una rutina interna de Oracle genera y aumenta (o disminuye) la secuencia. Esto puede ser un objeto que ahorre tiempo ya que puede reducir la cantidad de código de aplicación necesario para escribir una rutina generadora de secuencias.

Los números de secuencia se almacenan y se generan independientemente de las tablas. Por lo tanto, la misma secuencia se puede utilizar para varias tablas.



# Sintaxis de la Sentencia

## CREATE SEQUENCE

Defina una secuencia para generar números secuenciales automáticamente:

```
CREATE SEQUENCE sequence
  [INCREMENT BY n]
  [START WITH n]
  [{MAXVALUE n | NOMAXVALUE}]
  [{MINVALUE n | NOMINVALUE}]
  [{CYCLE | NOCYCLE}]
  [{CACHE n | NOCACHE}];
```

ORACLE

12-5

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de una Secuencia

Genere automáticamente números secuenciales utilizando la sentencia CREATE SEQUENCE.

En la sintaxis:

<i>sequence</i>	es el nombre del generador de secuencias.
INCREMENT BY <i>n</i>	especifica el intervalo entre números de secuencia donde <i>n</i> es un entero (si esta cláusula se omite, la secuencia aumenta en 1).
START WITH <i>n</i>	especifica el primer número de secuencia que se va a generar (si esta cláusula se omite, la secuencia comienza por 1).
MAXVALUE <i>n</i>	especifica el valor máximo que la secuencia puede generar.
NOMAXVALUE	especifica un valor máximo de $10^{27}$ para una secuencia ascendente y $-1$ para una secuencia descendente (ésta es la opción por defecto).
MINVALUE <i>n</i>	especifica el valor de secuencia mínimo.
NOMINVALUE	especifica un valor mínimo de 1 para una secuencia ascendente y $(10^{26})$ para una secuencia descendente (ésta es la opción por defecto).
CYCLE   NOCYCLE	especifica si la secuencia continua generando valores después de alcanzar su valor máximo o mínimo (NOCYCLE es la opción por defecto).
CACHE <i>n</i>   NOCACHE	especifica cuántos valores preasigna Oracle Server y cuántos mantiene en memoria (por defecto, Oracle Server almacena en caché 20 valores).

## Creación de una Secuencia

- Cree una secuencia llamada DEPT\_DEPTID\_SEQ para utilizarla para la clave primaria de la tabla DEPARTMENTS.
- No utilice la opción CYCLE.

```
CREATE SEQUENCE dept_deptid_seq
            INCREMENT BY 10
            START WITH 120
            MAXVALUE 9999
            NOCACHE
            NOCYCLE;
Sequence created.
```

ORACLE

12-6

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de una Secuencia (continuación)

En el ejemplo de la transparencia se crea una secuencia llamada DEPT\_DEPTID\_SEQ para utilizarla para la columna DEPARTMENT\_ID de la tabla DEPARTMENTS. La secuencia comienza en 120, no permite el almacenamiento en caché y no tiene ciclos.

No utilice la opción CYCLE si la secuencia se usa para generar valores de clave primaria, a menos que disponga de un mecanismo seguro que depure las filas antiguas más rápidamente que los ciclos de secuencia.

Para obtener más información, consulte *Oracle9i SQL Reference*, “CREATE SEQUENCE”.

**Nota:** La secuencia no está unida a una tabla. Por lo general, debería asignar un nombre a la secuencia según el uso para el que se creó; no obstante, la secuencia se puede utilizar en cualquier lugar, independientemente de su nombre.

## Confirmación de Secuencias

- Verifique los valores de secuencia en la tabla `USER_SEQUENCES` del diccionario de datos.

```
SELECT  sequence_name, min_value, max_value,
        increment_by, last_number
FROM    user_sequences;
```

- La columna `LAST_NUMBER` muestra el siguiente número de secuencia disponible si se especifica `NOCACHE`.

ORACLE

12-7

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Confirmación de Secuencias

Una vez que haya creado la secuencia, ésta se documenta en el diccionario de datos. Puesto que una secuencia es un objeto de base de datos, puede identificarla en la tabla `USER_OBJECTS` del diccionario de datos.

También puede confirmar la definición de la secuencia seleccionando desde la vista del diccionario de datos `USER_SEQUENCES`.

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPARTMENTS_SEQ	1	9990	10	280
DEPT_DEPTID_SEQ	1	9999	10	120
EMPLOYEES_SEQ	1	1.0000E+27	1	207
LOCATIONS_SEQ	1	9900	100	3300

## Pseudocolumnas NEXTVAL y CURRVAL

- **NEXTVAL devuelve el siguiente valor de secuencia disponible. Devuelve un valor único cada vez que se hace referencia a ella, incluso para usuarios distintos.**
- **CURRVAL obtiene el valor de la secuencia actual.**
- **NEXTVAL se debe emitir para dicha secuencia antes de que CURRVAL contenga un valor.**

ORACLE

12-8

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Uso de una Secuencia

Después de crear la secuencia, ésta genera números secuenciales para el uso en tablas. Haga referencia a los valores de secuencia utilizando las pseudocolumnas NEXTVAL y CURRVAL.

#### Pseudocolumnas NEXTVAL y CURRVAL

La pseudocolumna NEXTVAL se utiliza para extraer números de secuencia sucesivos de una secuencia especificada. Debe cualificar NEXTVAL con el nombre de la secuencia. Al hacer referencia a *sequence*.NEXTVAL, se genera un nuevo número de secuencia y el número actual se coloca en CURRVAL.

La pseudocolumna CURRVAL se utiliza para hacer referencia a un número de secuencia que el usuario actual acaba de generar. NEXTVAL se debe utilizar para generar un número de secuencia en la sesión del usuario actual antes de que se pueda hacer referencia a CURRVAL. Debe cualificar CURRVAL con el nombre de la secuencia. Al hacer referencia a *sequence*.CURRVAL, se muestra el último valor devuelto al proceso de dicho usuario.

## Reglas para el Uso de NEXTVAL y CURRVAL

Puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:

- La lista SELECT de una sentencia SELECT que no forme parte de una subconsulta
- La lista SELECT de una subconsulta en una sentencia INSERT
- La cláusula VALUES de una sentencia INSERT
- La cláusula SET de una sentencia UPDATE

No puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:

- La lista SELECT de una vista
- Una sentencia SELECT con la palabra clave DISTINCT
- Una sentencia SELECT con cláusulas GROUP BY, HAVING u ORDER BY
- Una subconsulta en una sentencia SELECT, DELETE o UPDATE
- La expresión DEFAULT en una sentencia CREATE TABLE o ALTER TABLE

Para obtener más información, consulte *Oracle9i SQL Reference*, sección “Pseudocolumns” y “CREATE SEQUENCE”.

## Uso de una Secuencia

- Inserte un nuevo departamento llamado “Support” en el identificador de ubicación 2500.

```
INSERT INTO departments(department_id,
                        department_name, location_id)
VALUES      (dept_deptid_seq.NEXTVAL,
            'Support', 2500);
1 row created.
```

- Visualice el valor actual para la secuencia DEPT\_DEPTID\_SEQ.

```
SELECT  dept_deptid_seq.CURRVAL
FROM    dual;
```

ORACLE

12-10

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Uso de una Secuencia

En el ejemplo de la transparencia se inserta un departamento nuevo en la tabla DEPARTMENTS. Se utiliza la secuencia DEPT\_DEPTID\_SEQ para generar un nuevo número de departamento como se indica a continuación:

Puede visualizar el valor actual de la secuencia:

```
SELECT dept_deptid_seq.CURRVAL
FROM    dual;
```

CURRVAL	
	120

Suponga ahora que desea contratar a empleados para el nuevo departamento. La sentencia INSERT que va a ejecutar para todos los empleados nuevos puede incluir el siguiente código:

```
INSERT INTO employees (employee_id, department_id, ...)
VALUES (employees_seq.NEXTVAL, dept_deptid_seq.CURRVAL, ...);
```

**Nota:** En el ejemplo anterior se asume que ya se ha creado una secuencia llamada EMPLOYEE\_SEQ para generar nuevos números de empleado.

## Uso de una Secuencia

- El almacenamiento en memoria caché de valores de secuencia proporciona un acceso más rápido a dichos valores.
- Los intervalos en los valores de secuencia se pueden producir cuando:
  - Se realiza un rollback.
  - El sistema falla.
  - Se está utilizando una secuencia en otra tabla.
- Si la secuencia se creó con NOCACHE, visualice el siguiente valor disponible, consultando la tabla `USER_SEQUENCES`.

ORACLE

12-11

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Almacenamiento en Caché de Valores de Secuencia

Almacene en memoria caché las secuencias para proporcionar un acceso más rápido a dichos valores de secuencia. El caché se rellena la primera vez que se hace referencia a la secuencia. Cada solicitud para el siguiente valor de secuencia se recupera desde la secuencia almacenada en caché. Después de que se haya utilizado el último valor de secuencia, la siguiente solicitud introduce otro caché de secuencias en la memoria.

#### Intervalos en la Secuencia

Aunque los generadores de secuencias emiten números secuenciales sin intervalos, esta acción se produce independientemente de una validación o rollback. Por lo tanto, si realiza rollback en una sentencia que contenga una secuencia, se pierde el número.

Otro evento que puede producir intervalos en la secuencia es un fallo del sistema. Si la secuencia almacena en memoria caché valores, estos se pierden si el sistema falla.

Como las secuencias no están unidas a tablas directamente, la misma secuencia se puede utilizar para varias tablas. De este modo, cada tabla puede contener intervalos en los números secuenciales.

#### Visualización del Siguiete Valor de Secuencia Disponible sin Aumento

Si la secuencia se creó con NOCACHE, es posible visualizar el siguiente valor de secuencia disponible sin aumentarlo mediante una consulta de la tabla `USER_SEQUENCES`.

## Modificación de una Secuencia

**Cambie el valor de aumento, el valor máximo, el valor mínimo, la opción de ciclo o la opción de caché.**

```
ALTER SEQUENCE dept_deptid_seq
        INCREMENT BY 20
        MAXVALUE 999999
        NOCACHE
        NOCYCLE;
Sequence altered.
```

ORACLE

12-12

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Modificación de una Secuencia

Si alcanza el límite MAXVALUE para la secuencia, no se asignará ningún valor adicional de la secuencia y recibirá un mensaje de error que indicando que la secuencia excede el límite MAXVALUE. Para continuar utilizando la secuencia, puede modificarla utilizando la sentencia ALTER SEQUENCE.

#### Sintaxis

```
ALTER SEQUENCE sequence
    [ INCREMENT BY n ]
    [ { MAXVALUE n | NOMAXVALUE } ]
    [ { MINVALUE n | NOMINVALUE } ]
    [ { CYCLE | NOCYCLE } ]
    [ { CACHE n | NOCACHE } ] ;
```

En la sintaxis:

*sequence* es el nombre del generador de secuencias.

Para obtener más información, consulte *Oracle9i SQL Reference*, “ALTER SEQUENCE”.



## Instrucciones para la Modificación de una Secuencia

- Debe ser el propietario o tener el privilegio **ALTER** para la secuencia.
- Sólo se ven afectados los números de secuencia futuros.
- Se debe borrar y volver a crear la secuencia para reiniciarla en un número diferente.
- Se realiza alguna validación.

ORACLE

12-13

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Instrucciones para la Modificación de Secuencias

- Debe ser el propietario o tener el privilegio **ALTER** para la secuencia para modificarla.
- Sólo se ven afectados los números de secuencia futuros por la sentencia **ALTER SEQUENCE**.
- La opción **START WITH** no se puede cambiar utilizando **ALTER SEQUENCE**. La secuencia se debe borrar y volver a crear para reiniciarla en un número diferente.
- Se realiza alguna validación. Por ejemplo, no se puede imponer un nuevo **MAXVALUE** que sea menor que el número de secuencia actual.

```
ALTER SEQUENCE dept_deptid_seq
    INCREMENT BY 20
    MAXVALUE 90
    NOCACHE
    NOCYCLE;
```

```
ALTER SEQUENCE dept_deptid_seq
```

```
*
```

```
ERROR at line 1:
```

```
ORA-04009: MAXVALUE cannot be made to be less than the current
value
```

## Eliminación de una Secuencia

- Elimine una secuencia del diccionario de datos utilizando la sentencia `DROP SEQUENCE`.
- Una vez que se ha eliminado, ya no se puede hacer referencia a la secuencia.

```
DROP SEQUENCE dept_deptid_seq;  
Sequence dropped.
```

ORACLE

12-14

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Eliminación de una Secuencia

Para eliminar una secuencia del diccionario de datos, utilice la sentencia `DROP SEQUENCE`. Debe ser el propietario de la secuencia o tener el privilegio `DROP ANY SEQUENCE` para eliminarla.

#### Sintaxis

```
DROP SEQUENCE sequence ;
```

En la sintaxis:

*sequence* es el nombre del generador de secuencias.

Para obtener más información, consulte *Oracle9i SQL Reference*, “`DROP SEQUENCE`”.

# ¿Qué es un Índice?

## Un índice:

- Es un objeto de esquema.
- Oracle Server lo utiliza para acelerar la recuperación de filas utilizando un puntero.
- Puede reducir la E/S de disco mediante un método de ruta de acceso rápido para encontrar datos.
- Es independiente de su tabla indexada.
- Oracle Server lo utiliza y mantiene automáticamente.

ORACLE

12-15

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Índices

Un índice de Oracle Server es un objeto de esquema que puede acelerar la recuperación de filas utilizando un puntero. Los índices se pueden crear explícitamente o automáticamente. Si no tiene un índice en la columna, se realiza una exploración completa de la tabla.

Un índice proporciona acceso directo y rápido a las filas de una tabla. Su objetivo es reducir la necesidad de E/S de disco utilizando una ruta indexada para encontrar los datos rápidamente. Oracle Server utiliza y mantiene el índice automáticamente. Una vez que se ha creado el índice, no se requiere ninguna actividad directa por parte del usuario.

Los índices son independientes lógicamente y físicamente de su tabla indexada. Esto significa que se pueden crear o borrar en cualquier momento y que no tienen efecto en las tablas base ni en otros índices.

**Nota:** Al borrar una tabla, los índices correspondientes también se borran.

Para obtener más información, consulte *Oracle9i Concepts*, sección “Schema Objects”, tema “Indexes”.

## ¿Cómo Se Crean los Índices?

- **Automáticamente:** Un índice único se crea automáticamente al definir una restricción `UNIQUE` o `PRIMARY KEY` en una definición de tabla.
- **Manualmente:** Los usuarios pueden crear índices no únicos en columnas para acelerar el acceso a las filas.

ORACLE

12-16

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Tipos de Índices

Se pueden crear dos tipos de índices. Uno de ellos es un índice único: Oracle Server lo crea automáticamente al definir una columna en una tabla para obtener una restricción `PRIMARY KEY` o de clave `UNIQUE`. El nombre del índice es el nombre que se ha asignado a la restricción.

El otro tipo de índice es un índice no único, que puede crear un usuario. Por ejemplo, puede crear un índice de columna `FOREIGN KEY` para una unión en una consulta para mejorar la velocidad de recuperación.

**Nota:** Puede crear manualmente un índice único, pero se recomienda que cree una restricción única, lo cual crea implícitamente un índice único.

## Creación de un Índice

- Cree un índice en una o varias columnas.

```
CREATE INDEX index  
ON table (column[, column]...);
```

- Mejore la velocidad de acceso de consulta a la columna `LAST_NAME` de la tabla `EMPLOYEES`.

```
CREATE INDEX emp_last_name_idx  
ON      employees(last_name);  
Index created.
```

ORACLE

12-17

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Creación de un Índice

Cree un índice en una o varias columnas emitiendo la sentencia `CREATE INDEX`.

En la sintaxis:

<i>index</i>	es el nombre del índice.
<i>table</i>	es el nombre de la tabla.
<i>column</i>	es el nombre de la columna en la tabla que se va a indexar.

Para obtener más información, consulte *Oracle9i SQL Reference*, “`CREATE INDEX`”.

## Cuándo Se Crea un Índice

**Debe crear un índice si:**

- Una columna contiene un amplio rango de valores.
- Una columna contiene un gran número de valores nulos.
- Una o más columnas se utilizan juntas frecuentemente en una cláusula `WHERE` o en una condición de unión.
- La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2-4 por ciento de las filas.

ORACLE

12-18

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Más No Siempre Es Mejor

Más índices en una tabla no significa consultas más rápidas. Cada operación DML que se valida en una tabla con índices significa que los índices se deben actualizar. Cuantos más índices tenga asociados a una tabla, más esfuerzo tiene que realizar Oracle Server para actualizar todos los índices después de una operación DML.

### Cuándo Se Crea un Índice

Por lo tanto, debe crear índices sólo si:

- La columna contiene un amplio rango de valores.
- La columna contiene un gran número de valores nulos.
- Una o más columnas se utilizan juntas frecuentemente en una cláusula `WHERE` o en una condición de unión.
- La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2-4 % de las filas.

Recuerde que si desea forzar la unicidad, debe definir una restricción única en la definición de tabla. A continuación, se crea automáticamente un índice único.

## **Cuándo No Se Crea un Índice**

**Normalmente no merece la pena crear un índice si:**

- **La tabla es pequeña.**
- **Las columnas no se suelen utilizar como condición en la consulta.**
- **Se espera que la mayoría de las consultas recuperen más del 2-4 por ciento de las filas de la tabla.**
- **La tabla se actualiza frecuentemente.**
- **Se hace referencia a las columnas indexadas como parte de una expresión.**

**ORACLE**

## Confirmación de Índices

- La vista **USER\_INDEXES** del diccionario de datos contiene el nombre del índice y su unicidad.
- La vista **USER\_IND\_COLUMNS** contiene el nombre del índice, el nombre de la tabla y el nombre de la columna.

```
SELECT    ic.index_name, ic.column_name,  
          ic.column_position col_pos, ix.uniqueness  
FROM      user_indexes ix, user_ind_columns ic  
WHERE     ic.index_name = ix.index_name  
AND       ic.table_name = 'EMPLOYEES';
```

ORACLE

12-20

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### Confirmación de Índices

Confirme la existencia de los índices desde la vista **USER\_INDEXES** del diccionario de datos. También puede verificar las columnas afectadas por un índice consultando la vista **USER\_IND\_COLUMNS**.

El ejemplo de la transparencia muestra todos los índices creados previamente, con los nombres de la columna afectada y la unicidad del índice, en la tabla **EMPLOYEES**.

INDEX_NAME	COLUMN_NAME	COL_POS	UNIQUENES
EMP_EMAIL_UK	EMAIL	1	UNIQUE
EMP_EMP_ID_PK	EMPLOYEE_ID	1	UNIQUE
EMP_DEPARTMENT_IX	DEPARTMENT_ID	1	NONUNIQUE
EMP_JOB_IX	JOB_ID	1	NONUNIQUE
EMP_MANAGER_IX	MANAGER_ID	1	NONUNIQUE
EMP_NAME_IX	LAST_NAME	1	NONUNIQUE
EMP_NAME_IX	FIRST_NAME	2	NONUNIQUE
EMP_LAST_NAME_IDX	LAST_NAME	1	NONUNIQUE

8 rows selected.



# Índices Basados en Funciones

- Un índice basado en función es un índice basado en expresiones.
- La expresión de índice se crea desde columnas de tabla, constantes, funciones SQL y funciones definidas por el usuario.

```
CREATE INDEX upper_dept_name_idx  
ON departments(UPPER(department_name));
```

Index created.

```
SELECT *  
FROM   departments  
WHERE  UPPER(department_name) = 'SALES';
```

ORACLE

12-21

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Índices Basados en Funciones

Los índices basados en funciones definidos con las palabras clave `UPPER(column_name)` o `LOWER(column_name)` permiten búsquedas no sensibles a mayúsculas/minúsculas. Por ejemplo, el siguiente índice:

```
CREATE INDEX upper_last_name_idx ON employees (UPPER(last_name));
```

facilita el procesamiento de consultas como:

```
SELECT * FROM employees WHERE UPPER(last_name) = 'KING';
```

Para asegurarse de que Oracle Server utiliza el índice en lugar de realizar una exploración completa de la tabla, compruebe que el valor de la función no es nulo en consultas posteriores. Por ejemplo, se garantiza que la siguiente sentencia utiliza el índice, pero sin la cláusula `WHERE` es posible que Oracle Server realice una exploración completa de la tabla:

```
SELECT *  
FROM   employees  
WHERE  UPPER (last_name) IS NOT NULL  
ORDER BY UPPER (last_name);
```

Oracle Server trata los índices con columnas marcadas `DESC` como índices basados en funciones. Las columnas marcadas `DESC` se ordenan en orden descendente.

## Eliminación de un Índice

- Elimine un índice del diccionario de datos utilizando el comando `DROP INDEX`.

```
DROP INDEX index;
```

- Elimine el índice `UPPER_LAST_NAME_IDX` del diccionario de datos.

```
DROP INDEX upper_last_name_idx;  
Index dropped.
```

- Para borrar un índice, debe ser el propietario del mismo o tener el privilegio `DROP ANY INDEX`.

ORACLE

### Eliminación de un Índice

Los índices no se pueden modificar. Para cambiar un índice, debe borrarlo y volver a crearlo. Elimine una definición de índice del diccionario de datos emitiendo la sentencia `DROP INDEX`. Para borrar un índice, debe ser el propietario del mismo o tener el privilegio `DROP ANY INDEX`.

En la sintaxis:

*index* es el nombre del índice.

**Nota:** Si borra una tabla, los índices y las restricciones se borran automáticamente, pero las vistas y las secuencias se mantienen.

# Sinónimos

**Simplifique el acceso a los objetos mediante la creación de un sinónimo (un objeto con otro nombre). Con los sinónimos, puede:**

- **Facilitar la referencia a una tabla propiedad de otro usuario**
- **Reducir nombres largos de objeto.**

```
CREATE [PUBLIC] SYNONYM synonym  
FOR      object;
```

ORACLE

12-23

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Creación de un Sinónimo para un Objeto

Para consultar una tabla propiedad de otro usuario, es necesario que escriba en el nombre de la tabla el nombre del usuario que la creó como prefijo, seguido de un punto. Al crear un sinónimo se elimina la necesidad de cualificar el nombre de objeto con el esquema y se proporciona el nombre alternativo para una tabla, vista, secuencia, procedimiento u otros objetos. Este método puede ser especialmente útil con nombres largos de objeto, como por ejemplo, las vistas.

En la sintaxis:

<i>PUBLIC</i>	crea un sinónimo accesible para todos los usuarios.
<i>synonym</i>	es el nombre del sinónimo que se va a crear.
<i>object</i>	identifica el objeto para el que se crea el sinónimo.

### Instrucciones

- No se puede contener el objeto en un paquete.
- El nombre de un sinónimo privado debe ser distinto de todos los demás objetos propiedad del mismo usuario.

Para obtener más información, consulte *Oracle9i SQL Reference*, “CREATE SYNONYM”.

# Creación y Eliminación de Sinónimos

- Cree una abreviatura para la vista DEPT\_SUM\_VU.

```
CREATE SYNONYM d_sum  
FOR dept_sum_vu;  
Synonym Created.
```

- Borre un sinónimo.

```
DROP SYNONYM d_sum;  
Synonym dropped.
```

ORACLE

12-24

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Creación de un Sinónimo para un Objeto (continuación)

En el ejemplo de la transparencia se crea un sinónimo para la vista DEPT\_SUM\_VU para lograr una referencia más rápida.

El administrador de la base de datos puede crear un sinónimo público que sea accesible para todos los usuarios. En el siguiente ejemplo se crea un sinónimo público llamado DEPT para la tabla DEPARTMENTS de Alice:

```
CREATE PUBLIC SYNONYM dept  
FOR alice.departments;  
Synonym created.
```

## Eliminación de un Sinónimo

Para borrar un sinónimo, utilice la sentencia DROP SYNONYM. Solamente el administrador de la base de datos puede borrar un sinónimo público.

```
DROP PUBLIC SYNONYM dept;  
Synonym dropped.
```

Para obtener más información, consulte *Oracle9i SQL Reference*, “DROP SYNONYM”.

# Resumen

**En esta lección, debería haber aprendido a:**

- **Generar números de secuencias automáticamente utilizando un generador de secuencias**
- **Visualizar información de secuencia en la tabla `USER_SEQUENCES` del diccionario de datos**
- **Crear índices para mejorar la velocidad de recuperación de consulta**
- **Visualizar información de índice en la tabla `USER_INDEXES` del diccionario**
- **Utilizar sinónimos para proporcionar nombres alternativos para objetos**

ORACLE

12-25

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

## Resumen

En esta lección, debería haber aprendido acerca de algunos de los otros objetos de la base de datos, incluidos índices, secuencias y vistas.

### Secuencias

El generador de secuencias se puede utilizar para generar automáticamente números de secuencia para las filas de tablas. Esto puede ahorrar tiempo y reducir la cantidad necesaria de código de aplicación.

Una secuencia es un objeto de base de datos que se puede compartir con otros usuarios. Se puede encontrar información acerca de la secuencia en la tabla `USER_SEQUENCES` del diccionario de datos.

Para utilizar una secuencia, haga referencia a ella mediante las pseudocolumnas `NEXTVAL` o `CURRVAL`.

- Recupere el siguiente número de la secuencia haciendo referencia a `sequence .NEXTVAL`.
- Devuelva el número disponible actual haciendo referencia a `sequence .CURRVAL`.

### Índices

Los índices se utilizan para mejorar la velocidad de recuperación de consulta. Los usuarios pueden visualizar las definiciones de los índices en la vista `USER_INDEXES` del diccionario de datos. El creador de un índice o un usuario con el privilegio `DROP ANY INDEX` pueden borrar un índice utilizando la sentencia `DROP INDEX`.

### Sinónimos

Los administradores de la base de datos pueden crear sinónimos públicos y los usuarios pueden crear sinónimos privados, según convenga, utilizando la sentencia `CREATE SYNONYM`. Los sinónimos permiten abreviaturas o nombres alternativos para los objetos. Elimine los sinónimos utilizando la sentencia `DROP SYNONYM`.

## **Visión General de la Práctica 12**

**Esta práctica cubre los siguientes temas:**

- **Creación de secuencias**
- **Uso de secuencias**
- **Creación de índices no únicos**
- **Visualización de información del diccionario de datos acerca de secuencias e índices**
- **Borrado de índices**

ORACLE

12-26

Copyright © Oracle Corporation, 2001. Todos los Derechos Reservados.

### **Visión General de la Práctica 12**

En esta práctica creará una secuencia para utilizarla al rellenar la tabla. También creará índices implícitos y explícitos.

## Práctica 12

1. Cree una secuencia para utilizarla con la columna de clave primaria de la tabla DEPT. La secuencia debe comenzar en 200 y tener un valor máximo de 1000. Haga que la secuencia aumente en diez números. Asigne a la secuencia el nombre DEPT\_ID\_SEQ.
2. Escriba una consulta en un archivo de comandos para mostrar la siguiente información acerca de las secuencias: nombre de secuencia, valor máximo, tamaño de aumento y último número. Asigne al archivo de comandos el nombre lab12\_2.sql. Ejecute la sentencia en el archivo de comandos.

SEQUENCE_NAME	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
DEPARTMENTS_SEQ	9990	10	280
DEPT_ID_SEQ	1000	10	200
EMPLOYEES_SEQ	1.0000E+27	1	207
LOCATIONS_SEQ	9900	100	3300

3. Escriba un archivo de comandos para insertar dos filas en la tabla DEPT. Asigne al archivo de comandos el nombre lab12\_3.sql. Asegúrese de utilizar la secuencia que creó para la columna ID. Agregue dos departamentos llamados Education y Administration. Confirme las adiciones. Ejecute los comandos en el archivo de comandos.
4. Cree un índice no único en la columna de clave ajena (DEPT\_ID) en la tabla EMP.
5. Visualice los índices y la unicidad que existe en el diccionario de datos para la tabla EMP. Guarde la sentencia en un archivo de comandos llamado lab12\_5.sql.

INDEX_NAME	TABLE_NAME	UNIQUENES
EMP_DEPT_ID_IDX	EMP	NONUNIQUE
MY_EMP_ID_PK	EMP	UNIQUE

