ORACLE

Academy

Database Programming with SQL

11-1

Asegurando la Calidad de los Resultados de las Consultas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Crear una consulta para producir datos especificados
  - Modificar una consulta para producir datos especificados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Ha aprendido las reglas de sintaxis para generar una consulta SQL pero, ¿está seguro de que produce los datos deseados?
- Observar la salida deseada y, a continuación, averiguar la consulta para generar dicha salida le ayudará a ganar confianza en que los resultados de la consulta son los esperados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de todas las tablas cuyos dos primeros caracteres en el nombre de la tabla sean JO
  - Las tablas deben ser propiedad del usuario de Oracle actual
- Tablas Utilizadas:
  - User_tables

Resultado de la Consulta:

TABLE_NAME
JOBS
JOB_GRADES
JOB_HISTORY

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista que incluya la inicial del nombre de cada empleado, un espacio y el apellido del empleado
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Employee Names
E Abel
C Davies
L De Haan
B Ernst
P Fay
W Gietz
K Grant
M Hartstein
S Higgins
A Hunold
S King
N Kochhar
D Lorentz
R Matos
K Mourgos
T Rajs
J Taylor
P Vargas
J Whalen
E Zlotkey

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de todos los empleados concatenados a un espacio y el apellido del empleado y el correo electrónico de todos los empleados donde la dirección de correo electrónico contenga la cadena "IN"
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Employee Name	Email
Shelley Higgins	SHIGGINS
Steven King	SKING

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista del apellido "más pequeño" y el apellido "más grande" de la tabla employees
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

First Last Name	Last Last Name
Abel	Zlotkey

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de salarios semanales a partir de la tabla employees donde el salario semanal esté entre 700 y 3000
  - Los salarios deben estar formateados para incluir un signo $ y tener dos decimales como: $9999,99
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Weekly Salary
$1015.38
$2769.23
$1915.38
$2423.08
$2538.46
$1984.62
$1615.38
$1338.46
$807.69
$715.38
$2076.92
$1384.62
$969.23
$3000
$1384.62

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de todos los empleados y su cargo relacionado ordenada por job_title
- Tablas Utilizadas:
  - Employees, Jobs

Resultado de la Consulta:

Employee Name	Job
S Higgins	Accounting Manager
J Whalen	Administration Assistant
L De Haan	Administration Vice President
N Kochhar	Administration Vice President
M Hartstein	Marketing Manager
P Fay	Marketing Representative
S King	President
A Hunold	Programmer
B Ernst	Programmer
D Lorentz	Programmer
W Gietz	Public Accountant
E Zlotkey	Sales Manager
J Taylor	Sales Representative
K Grant	Sales Representative
E Abel	Sales Representative
T Rajs	Stock Clerk
C Davies	Stock Clerk
P Vargas	Stock Clerk
R Matos	Stock Clerk
K Mourgos	Stock Manager

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de cargos de todos los empleados, rangos de salarios del cargo y salario del empleado
  - Enumere el rango de salarios menor y mayor de cada cargo con un guión para separar los salarios de la siguiente forma: 100 – 200
- Tablas Utilizadas:
  - Employees, Jobs

Resultado de la Consulta:

Employee Name	Job	Salary Range	Employee's Salary
S Higgins	Accounting Manager	8200 - 16000	12000
J Whalen	Administration Assistant	3000 - 6000	4400
L De Haan	Administration Vice President	15000 - 30000	17000
N Kochhar	Administration Vice President	15000 - 30000	17000
M Hartstein	Marketing Manager	9000 - 15000	13000
P Fay	Marketing Representative	4000 - 9000	6000
S King	President	20000 - 40000	24000
A Hunold	Programmer	4000 - 10000	9000
B Ernst	Programmer	4000 - 10000	6000
D Lorentz	Programmer	4000 - 10000	4200
W Gietz	Public Accountant	4200 - 9000	8300
E Zlotkey	Sales Manager	10000 - 20000	10500
E Abel	Sales Representative	6000 - 12000	11000
K Grant	Sales Representative	6000 - 12000	7000
J Taylor	Sales Representative	6000 - 12000	8600
T Rajs	Stock Clerk	2000 - 5000	3500
C Davies	Stock Clerk	2000 - 5000	3100
P Vargas	Stock Clerk	2000 - 5000	2500
R Matos	Stock Clerk	2000 - 5000	2600
K Mourgos	Stock Manager	5500 - 8500	5800

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Mediante un método de unión de ANSI, cree una lista de iniciales y apellidos de todos los empleados y nombres de departamento
  - Asegúrese de que las tablas se unen en todas las claves ajenas declaradas entre las dos tablas
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Employee Name	Department Name
N Kochhar	Executive
L De Haan	Executive
W Gietz	Accounting
E Abel	Sales
J Taylor	Sales
T Rajs	Shipping
C Davies	Shipping
R Matos	Shipping
P Vargas	Shipping
B Ernst	IT
D Lorentz	IT
P Fay	Marketing

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cambie la lista anterior a unión solo en la columna department_id
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Employee Name	Department Name
J Whalen	Administration
M Hartstein	Marketing
P Fay	Marketing
C Davies	Shipping
P Vargas	Shipping
T Rajs	Shipping
K Mourgos	Shipping
R Matos	Shipping
A Hunold	IT
B Ernst	IT
D Lorentz	IT
J Taylor	Sales
E Zlotkey	Sales
E Abel	Sales
L De Haan	Executive
S King	Executive
N Kochhar	Executive
S Higgins	Accounting
W Gietz	Accounting

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de apellidos de todos los empleados y la palabra nobody o somebody en función de si el empleado tiene un jefe
  - Utilice la función DECODE de Oracle para crear la lista
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Works for	Last Name
Nobody	King
Somebody	Kochhar
Somebody	De Haan
Somebody	Whalen
Somebody	Higgins
Somebody	Gietz
Somebody	Zlotkey
Somebody	Abel
Somebody	Taylor
Somebody	Grant
Somebody	Mourgos
Somebody	Rajs
Somebody	Davies
Somebody	Matos
Somebody	Vargas
Somebody	Hunold
Somebody	Ernst
Somebody	Lorentz
Somebody	Hartstein
Somebody	Fay

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de iniciales y apellidos de todos los empleados, salarios y un sí o no para mostrar si un empleado tiene una comisión
  - Corrija esta consulta para producir el resultado.
- CONSULTA:

```
SELECT SUBSTR(first_name,1 1)||' '|last_name,
  "Employee Name", salary "Salary",
  DEC(commission_pct NULL, 'No', 'Yes')'Commission'
FROM   employees;
```

Resultado de la Consulta:

Employee Name	Salary	Commission
S King	24000	No
N Kochhar	17000	No
L De Haan	17000	No
J Whalen	4400	No
S Higgins	12000	No
W Gietz	8300	No
E Zlotkey	10500	Yes
E Abel	11000	Yes
J Taylor	8600	Yes
K Grant	7000	Yes
K Mourgos	5800	No
T Rajs	3500	No
C Davies	3100	No
R Matos	2600	No
P Vargas	2500	No
A Hunold	9000	No
B Ernst	6000	No
D Lorentz	4200	No
M Hartstein	13000	No
P Fay	6000	No

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de apellidos de todos los empleados, nombres de departamento, ciudades y estados/provincias
  - Incluya los departamentos sin los empleados
  - Es necesaria una unión externa
- Tablas Utilizadas:
  - Employees, Departments, Locations

Resultado de la Consulta:

LAST_NAME	DEPARTMENT_NAME	CITY	STATE_PROVINCE
Abel	Sales	Oxford	Oxford
Davies	Shipping	South San Francisco	California
De Haan	Executive	Seattle	Washington
Ernst	IT	Southlake	Texas
Fay	Marketing	Toronto	Ontario
Gietz	Accounting	Seattle	Washington
Hartstein	Marketing	Toronto	Ontario
Higgins	Accounting	Seattle	Washington
Hunold	IT	Southlake	Texas
King	Executive	Seattle	Washington
Kochhar	Executive	Seattle	Washington
Lorentz	IT	Southlake	Texas
Matos	Shipping	South San Francisco	California
Mourgos	Shipping	South San Francisco	California
Rajs	Shipping	South San Francisco	California
Taylor	Sales	Oxford	Oxford
Vargas	Shipping	South San Francisco	California
Whalen	Administration	Seattle	Washington
Zlotkey	Sales	Oxford	Oxford
-	Contracting	Seattle	Washington

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres y apellidos de todos los empleados y la primera incidencia de: commission_pct, manager_id o -1
  - Si un empleado obtiene una comisión, muestre la columna commission_pct; si no hay ninguna comisión, muestre su manager_id; si no tiene ni comisión ni jefe, el número -1
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

First Name	Last Name	Which function???
Steven	King	-1
Neena	Kochhar	100
Lex	De Haan	100
Jennifer	Whalen	101
Shelley	Higgins	101
William	Gietz	205
Eleni	Zlotkey	.2
Ellen	Abel	.3
Jonathon	Taylor	.2
Kimberely	Grant	.15
Kevin	Mourgos	100
Trenna	Rajs	124
Curtis	Davies	124
Randall	Matos	124
Peter	Vargas	124
Alexander	Hunold	102
Bruce	Ernst	103
Diana	Lorentz	103
Michael	Hartstein	100
Pat	Fay	201

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de apellidos de todos los empleados, salarios y job_grade para todos los empleados que trabajan en departamentos con department_id superior a 50
- Tablas Utilizadas:
  - Employees, job_grades

Resultado de la Consulta:

LAST_NAME	SALARY	GRADE_LEVEL
Lorentz	4200	B
Ernst	6000	C
Gietz	8300	C
Taylor	8600	C
Hunold	9000	C
Zlotkey	10500	D
Abel	11000	D
Higgins	12000	D
Kochhar	17000	E
De Haan	17000	E
King	24000	E

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Elabore una lista del apellido de todos los empleados y el nombre de departamento
  - Incluya los empleados sin departamentos y los departamentos sin empleados
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

LAST_NAME	DEPARTMENT_NAME
Whalen	Administration
Fay	Marketing
Hartstein	Marketing
Mourgos	Shipping
Vargas	Shipping
Matos	Shipping
Davies	Shipping
Rajs	Shipping
Hunold	IT
Ernst	IT
Lorentz	IT
Zlotkey	Sales
Abel	Sales
Taylor	Sales
De Haan	Executive
Kochhar	Executive
King	Executive
Gietz	Accounting
Higgins	Accounting
Grant	-
-	Contracting

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de recorrido de árboles de apellidos de todos los empleados, apellido de su jefe y su posición en la compañía
  - El jefe de nivel superior tiene la posición 1, los subordinados de este jefe tienen la posición 2, sus subordinados tienen la posición 3, y así sucesivamente
  - Inicie la lista con el número de empleado 100
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

POSITION	LAST_NAME	MANAGER_NAME
1	King	-
2	Kochhar	King
3	Whalen	Kochhar
3	Higgins	Kochhar
4	Gietz	Higgins
2	De Haan	King
3	Hunold	De Haan
4	Ernst	Hunold
4	Lorentz	Hunold
2	Mourgos	King
3	Rajs	Mourgos
3	Davies	Mourgos
3	Matos	Mourgos
3	Vargas	Mourgos
2	Zlotkey	King
3	Abel	Zlotkey
3	Taylor	Zlotkey
3	Grant	Zlotkey
2	Hartstein	King
3	Fay	Hartstein

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de la primera fecha de contratación, la última fecha de contratación y el número de empleados de la tabla employees
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Lowest	Highest	No of Employees
17-Jun-1987	29-Jan-2000	20

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de departamento y los costos de departamento (sumados los salarios)
  - Incluya solo los departamentos cuyos costos de salarios sean de entre 15000 y 31000, y ordene la lista por el costo
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

DEPARTMENT_NAME	SALARIES
Shipping	17500
Marketing	19000
IT	19200
Accounting	20300
Sales	30100

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de departamento, el ID de jefe, el nombre del jefe (apellido del empleado) de dicho departamento y el salario medio de cada departamento
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

DEPARTMENT_NAME	MANAGER_ID	MANAGER_NAME	AVG_DEPT_SALARY
Shipping	124	Mourgos	3500
Administration	200	Whalen	4400
IT	103	Hunold	6400
Marketing	201	Hartstein	9500
Sales	149	Zlotkey	10033
Accounting	205	Higgins	10150
Executive	100	King	19333

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Muestre el mayor salario medio para los departamentos en la tabla employees
  - Redondee el resultado al número entero más cercano
- Tablas Utilizadas:
  - Employees

Resultado de la Consulta:

Highest Avg Sal for Depts
19333

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de departamento y sus costos mensuales (sumados los salarios)
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Department Name	Monthly Cost
Administration	4400
Accounting	20300
IT	19200
Executive	58000
Shipping	17500
Sales	30100
Marketing	19000

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de departamento y job_ids
  - Calcule el costo de salarios mensuales para cada job_id dentro de un departamento, para cada departamento y para todos los departamentos sumados juntos
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Department Name	Job Title	Monthly Cost
Accounting	AC_ACCOUNT	8300
Accounting	AC_MGR	12000
Accounting	-	20300
Administration	AD_ASST	4400
Administration	-	4400
Executive	AD_PRES	24000
Executive	AD_VP	34000
Executive	-	58000
IT	IT_PROG	19200
IT	-	19200
Marketing	MK_MAN	13000
Marketing	MK_REP	6000
Marketing	-	19000
Sales	SA_MAN	10500
Sales	SA_REP	19600
Sales	-	30100
Shipping	ST_CLERK	11700
Shipping	ST_MAN	5800
Shipping	-	17500
-	-	168500

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de departamento y job_ids
  - Calcule el costo de salarios mensuales para cada job_id dentro de un departamento, para cada departamento, para cada grupo de job_ids independientemente del departamento y para todos los departamentos sumados juntos (Indicación: cubo)
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Department Name	Job Title	Monthly Cost
Accounting	AC_ACCOUNT	8300
Accounting	AC_MGR	12000
Accounting	-	20300
Administration	AD_ASST	4400
Administration	-	4400
Executive	AD_PRES	24000
Executive	AD_VP	34000
Executive	-	58000
IT	IT_PROG	19200
IT	-	19200
Marketing	MK_MAN	13000
Marketing	MK_REP	6000
Marketing	-	19000
Sales	SA_MAN	10500
Sales	SA_REP	19600
Sales	-	30100
Shipping	ST_CLERK	11700
Shipping	ST_MAN	5800
Shipping	-	17500
-	AC_ACCOUNT	8300
-	AC_MGR	12000
-	AD_ASST	4400
-	AD_PRES	24000
-	AD_VP	34000
-	IT_PROG	19200
-	MK_MAN	13000
-	MK_REP	6000
-	SA_MAN	10500
-	SA_REP	19600
-	ST_CLERK	11700
-	ST_MAN	5800
-	-	168500

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Amplíe la lista anterior para mostrar también si el valor de department_id o job_id se ha utilizado para crear los subtotales mostrados en la salida. (Indicación: cubo, agrupación)
- Tablas Utilizadas:
  - Employees, Departments

Resultado de la Consulta:

Department Name	Job Title	Monthly Cost	Department ID Used	Job ID Used
Accounting	AC_ACCOUNT	8300	Yes	Yes
Accounting	AC_MGR	12000	Yes	Yes
Accounting	-	20300	Yes	No
Administration	AD_ASST	4400	Yes	Yes
Administration	-	4400	Yes	No
Executive	AD_PRES	24000	Yes	Yes
Executive	AD_VP	34000	Yes	Yes
Executive	-	58000	Yes	No
IT	IT_PROG	19200	Yes	Yes
IT	-	19200	Yes	No
Marketing	MK_MAN	13000	Yes	Yes
Marketing	MK_REP	6000	Yes	Yes
Marketing	-	19000	Yes	No
Sales	SA_MAN	10500	Yes	Yes
Sales	SA_REP	19600	Yes	Yes
Sales	-	30100	Yes	No
Shipping	ST_CLERK	11700	Yes	Yes
Shipping	ST_MAN	5800	Yes	Yes
Shipping	-	17500	Yes	No
-	AC_ACCOUNT	8300	No	Yes
-	AC_MGR	12000	No	Yes
-	AD_ASST	4400	No	Yes
-	AD_PRES	24000	No	Yes
-	AD_VP	34000	No	Yes
-	IT_PROG	19200	No	Yes
-	MK_MAN	13000	No	Yes
-	MK_REP	6000	No	Yes
-	SA_MAN	10500	No	Yes
-	SA_REP	19600	No	Yes
-	ST_CLERK	11700	No	Yes
-	ST_MAN	5800	No	Yes
-	-	168500	No	No

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista que incluya los costos de salarios mensuales de cada cargo dentro de un departamento
  - En la misma lista, muestre el costo de salarios mensuales por ciudad. (Indicación: juegos de agrupamiento)
- Tablas Utilizadas:
  - Employees, Departments, Locations

Resultado de la Consulta:

DEPARTMENT_NAME	JOB_ID	CITY	SUM(SALARY)
Accounting	AC_MGR	-	12000
Accounting	AC_ACCOUNT	-	8300
Administration	AD_ASST	-	4400
Executive	AD_VP	-	34000
Executive	AD_PRES	-	24000
IT	IT_PROG	-	19200
Marketing	MK_REP	-	6000
Marketing	MK_MAN	-	13000
Sales	SA_REP	-	19600
Sales	SA_MAN	-	10500
Shipping	ST_MAN	-	5800
Shipping	ST_CLERK	-	11700
-	-	Oxford	30100
-	-	Seattle	82700
-	-	South San Francisco	17500
-	-	Southlake	19200
-	-	Toronto	19000

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de nombres de empleados como se muestran e ID de departamento
  - En el mismo informe, muestre los ID y nombres de departamento. Por último, enumere las ciudades
  - Las filas no deben estar unidas, solo aparecer en el mismo informe (Indicación: unión)
- Tablas Utilizadas:
  - Employees, Departments, Locations

Resultado de la Consulta:

Employee Name	Department Id	Department Name	City
A Hunold	60	-	-
B Ernst	60	-	-
C Davies	50	-	-
D Lorentz	60	-	-
E Abel	80	-	-
E Ziotkey	80	-	-
J Taylor	80	-	-
J Whalen	10	-	-
K Grant	-	-	-
K Mourgos	50	-	-
L De Haan	90	-	-
M Hartstein	20	-	-
N Kochhar	90	-	-
P Fay	20	-	-
P Vargas	50	-	-
R Matos	50	-	-
S Higgins	110	-	-
S King	90	-	-
T Rajs	50	-	-
W Gietz	110	-	-
-	10	Administration	-
-	20	Marketing	-
-	50	Shipping	-
-	60	IT	-
-	80	Sales	-
-	90	Executive	-
-	110	Accounting	-
-	190	Contracting	-
-	-	-	Oxford
-	-	-	Seattle
-	-	-	South San Francisco
-	-	-	Southlake
-	-	-	Toronto

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Escriba la Consulta

- Problema:
  - Cree una lista de inicial y apellido de cada empleado, salario y nombre de departamento de cada empleado que gane más de la media de su departamento
- Tablas Utilizadas:
  - Departments, Employees

Resultado de la Consulta:

Employee	Salary	Department Name
M Hartstein	13000	Marketing
K Mourgos	5800	Shipping
A Hunold	9000	IT
E Zlotkey	10500	Sales
E Abel	11000	Sales
S King	24000	Executive
S Higgins	12000	Accounting

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Crear una consulta para producir datos especificados
  - Modificar una consulta para producir datos especificados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

12-1

Sentencias INSERT

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Explicar la importancia de poder modificar los datos de una base de datos
  - Construir y ejecutar sentencias INSERT que insertan una única fila con una cláusula VALUES
  - Construir y ejecutar sentencias INSERT que utilizan valores especiales, valores nulos y valores de fecha
  - Construir y ejecutar sentencias INSERT que copian filas de una tabla a otra mediante una subconsulta

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Hasta ahora, ha aprendido cómo acceder a datos de una base de datos
- Ha llegado el momento de aprender a realizar cambios en los datos de la base de datos
- En la empresa, las bases de datos son dinámicas
- Están constantemente en el proceso de inserción, actualización y supresión de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Piense en cuántas veces cambia la base de datos de alumnos de la escuela de un día a otro y de un año a otro
- Si no se realizaran cambios, la base de datos perdería rápidamente su utilidad
- En esta lección, comenzará a utilizar sentencias de lenguaje de manipulación de datos (DML) para realizar cambios en una base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Copia de Tablas antes de la Inserción

- Será el responsable de la modificación de tablas en el esquema
- También será responsables de restaurarlos al igual que un administrador de base de datos real asume esta responsabilidad
- Para mantener las tablas de esquema en su estado original, realizará una copia de cada tabla antes de terminar las actividades prácticas de esta lección y de las posteriores
- En cada actividad práctica, utilizará la copia de la tabla que cree, pero no la original
- Si por accidente modificara una copia de la tabla, podrá utilizar la tabla original para restaurar la copia

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Copia de Tablas antes de la Inserción

- Debe asignar un nombre a cada tabla copiada: copy_tablename
- Las copias de tablas no heredarán las reglas de integridad de clave primaria a clave ajena asociadas (restricciones de relación) de las tablas originales
- Sin embardo, los tipos de dato de columna se heredan de las tablas copias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis para Crear una Copia de una Tabla

- Cree la sintaxis de tabla:

```
CREATE TABLE copy_tablename
   AS (SELECT * FROM tablename);
```

- Por ejemplo:

```
CREATE TABLE copy_employees
AS (SELECT * FROM employees);
```

```
CREATE TABLE copy_departments
AS (SELECT * FROM departments);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis para Crear una Copia de una Tabla

- Para verificar y ver la copia de la tabla, utilice las siguientes sentencias DESCRIBE y SELECT:

```
DESCRIBE copy_employees;
```

```
SELECT * FROM copy_employees;
```

```
DESCRIBE copy_departments;
```

```
SELECT * FROM copy_departments;
```

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

9

La estructura y el contenido de la tabla también se pueden ver con el Explorador de Objetos en APEX.

INSERT

- La sentencia INSERT se utiliza para agregar una nueva fila a una tabla. La sentencia necesita tres valores:
  - el nombre de la tabla
  - los nombres de las columnas de la tabla que se va a rellenar
  - los valores correspondientes para cada columna
- ¿Cómo podemos INSERTAR los datos siguientes para crear un nuevo departamento en la tabla copy_departments?

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
200	Human Resources	205	1500

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

INSERT

- La siguiente sintaxis utiliza INSERT para agregar un nuevo departamento a la tabla copy_departments
- Esta sentencia muestra explícitamente cada columna tal y como aparece en la tabla
- Los valores para cada columna se muestran en el mismo orden
  - Tenga en cuenta que los valores de número no están entre comillas simples

```
INSERT INTO copy_departments
   (department_id, department_name, manager_id, location_id)
VALUES (200,'Human Resources', 205, 1500);
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
200	Human Resources	205	1500

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

11

Los nombres de columna se pueden mostrar en cualquier orden (aunque no supone ninguna ventaja hacerlo así) mientras que los valores de columna se muestran en el mismo orden que los nombres de columna.

INSERT

- Otra forma de insertar valores en una tabla es agregarlos implícitamente omitiendo los nombres de columna
- Una precaución: los valores de cada columna deben coincidir exactamente con el orden por defecto en el que aparecen en la tabla (como se muestra en una sentencia DESCRIBE) y se debe proporcionar un valor para cada columna

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

12

Nota: en el momento de la escritura, la función DESCRIBE no funciona como se esperaba en APEX. Mientras sea así, utilice el Explorador de Objetos para ver la información detallada de la tabla.

INSERT

- La sentencia INSERT en este ejemplo se ha escrito sin nombrar explícitamente las columnas
- Sin embargo, para mayor claridad, es mejor utilizar los nombres de columna en una cláusula INSERT

```
INSERT INTO copy_departments
VALUES
  (210,'Estate Management', 102, 1700);
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
210	Estate Management	102	1700

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Comprobación de la Tabla en Primer Lugar

- Antes de insertar datos en una tabla, debe comprobar varios detalles de la tabla
- La sentencia de nombre de tabla DESCRIBE devolverá una descripción de la estructura de la tabla y el gráfico de resumen de la tabla
- RESUMEN DE LA TABLA COPY_DEPARTMENTS:

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
DEPARTMENT_NAME	VARCHAR2	30	-	-	-	-
MANAGER_ID	NUMBER	-	6	0	-	✓
LOCATION_ID	NUMBER	-	4	0	-	✓

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

14

Nota: como esta tabla se creó como copia de la tabla de departamentos, la restricción de clave primaria no se ha copiado de la original.

Resumen de la Tabla

- Como se muestra en el ejemplo, el resumen de la tabla proporciona información sobre cada columna de la tabla, como:
  - permiso de valores duplicados
  - tipo de dato permitido
  - cantidad de datos permitida
  - permiso de valores NULL

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen de la Tabla

- Observe que la columna Tipo de Dato para los tipos de dato de caracteres especifica entre paréntesis el número máximo de caracteres permitidos

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen de la Tabla

- First_name tiene un tipo de dato VARCHAR2(20), esto significa que se pueden introducir un máximo de 20 caracteres para esta columna

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen de la Tabla

- Para los tipos de dato Number, los paréntesis especifican la precisión y la escala
- La precisión es el número total de dígitos y la escala es el número de dígitos a la derecha de la posición decimal

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

18

Los tipos de dato se tratan más detalladamente más adelante en el curso.

Resumen de la Tabla

- La columna SALARY permite números con una precisión de 8 y una escala de 2
- El valor máximo permitido en esta columna es 999999,99

Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
EMPLOYEE_ID	NUMBER	-	6	0	1	-
FIRST_NAME	VARCHAR2	20	-	-	-	✓
LAST_NAME	VARCHAR2	25	-	-	-	-
EMAIL	VARCHAR2	25	-	-	-	-
PHONE_NUMBER	VARCHAR2	20	-	-	-	✓
HIRE_DATE	DATE	7	-	-	-	-
JOB_ID	VARCHAR2	10	-	-	-	-
SALARY	NUMBER	-	8	2	-	✓
COMMISSION_PCT	NUMBER	-	2	2	-	✓
MANAGER_ID	NUMBER	-	6	0	-	✓
DEPARTMENT_ID	NUMBER	-	4	0	-	✓
BONUS	VARCHAR2	5	-	-	-	✓

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Filas con Valores Nulos

- La sentencia INSERT no necesita especificar todas las columnas: se pueden excluir las columnas con valores nulos
- Si a todas las columnas que necesitan un valor se les asigna un valor, la inserción funciona

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Filas con Valores Nulos

- En nuestro ejemplo, la columna EMAIL se define como una columna NOT NULL
- Un intento implícito de agregar valores a la tabla como el que se muestra generará un error

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, phone_number,
hire_date,
   job_id, salary)
VALUES
  (302,'Grigorz','Polanski', '8586667641', '15-Jun-2017',
   'IT_PROG',4200);
```

```
ORA-01400: cannot insert NULL into
("US_A009EMEA815_PLSQL_T01"."COPY_EMPLOYEES"."EMAIL")
```

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

21

Se ha omitido la columna EMAIL en la inserción, por lo que la inserción falla ya que no se permiten valores NULL para la columna EMAIL.

Inserción de Filas con Valores Nulos

- Una inserción implícita insertará automáticamente un valor nulo en las columnas que permiten valores nulos
- Para agregar explícitamente un valor nulo a una columna que permite valores nulos, utilice la palabra clave NULL en la lista VALUES

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Filas con Valores Nulos

- Para especificar cadenas vacías y/o fechas que faltan, utilice comillas simples vacía (sin espacios entre ellas de esta forma") para los datos que faltan

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number,
   hire_date, job_id, salary)
VALUES
  (302,'Grigorz','Polanski', 'gpolanski', '', '15-Jun-2017',
   'IT_PROG',4200);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
302	Grigorz	Polanski	gpolanski	-	15-Jun-2017	IT_PROG	4200

COMM_PCT	MGR_ID	DEPT_ID	BONUS
-	-	-	-

.....

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

23

NOTA: los nombres de columna en la salida se han abreviado para que quepan en la diapositiva.

Inserción de Valores Especiales

- Los valores especiales como SYSDATE y USER se pueden introducir en la lista VALUES de una sentencia INSERT
- SYSDATE colocará la fecha y hora actuales en una columna
- USER insertará el nombre de usuario de la sesión actual, que es OAE_PUBLIC_USER en Oracle Application Express

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Valores Especiales

- En este ejemplo se agrega USER como apellido y SYSDATE para la fecha de contratación

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number,
hire_date,
   job_id, salary)
VALUES
  (304,'Test',USER, 't_user', 4159982010, SYSDATE,
'ST_CLERK',2500);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
304	Test	APEX_PUBLIC_USER	t_user	4159982010	15-Jun-2017	ST_CLERK	2500

COMM_PCT	MGR_ID	DEPT_ID	BONUS
-	-	-	-

.....

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

25

Algunas columnas se han omitido de la salida por motivos de visualización.

Inserción de Valores de Fecha Específicos

- El modelo de formato por defecto para tipos de dato de fecha es DD-Mes-AAAA
- Con este formato de fecha, la hora por defecto de medianoche (00:00:00) también se incluye
- En la sección anterior, hemos aprendido cómo utilizar la función TO_CHAR para convertir una fecha en una cadena de caracteres cuando queremos recuperar y mostrar un valor de fecha con un formato que no es el formato por defecto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Valores de Fecha Específicos

- Este es un recordatorio de TO_CHAR:

```
SELECT first_name, TO_CHAR(hire_date,'Month, fmdd, yyyy')
FROM employees
WHERE employee_id = 101;
```

FIRST_NAME	TO_CHAR(HIRE_DATE,'MONTH,FMDD,YYYY')
Neena	September, 21, 1989

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Valores de Fecha Específicos

- Del mismo modo, si deseamos INSERTAR una fila con un formato que no sea el formato por defecto para una columna de fecha, debemos utilizar la función TO_DATE para convertir el valor de fecha (cadena de caracteres) en una fecha

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number,
hire_date,job_id, salary)
VALUES
   (301,'Katie','Hernandez', 'khernandez','8586667641',
    TO_DATE('July 8, 2017', 'Month fmdd, yyyy'),
'MK_REP',4200);
```

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción de Valores de Fecha Específicos

- Un segundo ejemplo de TO_DATE permite la inserción de una hora del día concreta, sustituyendo el la hora de medianoche por defecto

```
INSERT INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary)
VALUES
  (303,'Angelina','Wright', 'awright','4159982010',
   TO_DATE('July 10, 2017 17:20', 'Month fmdd, yyyy HH24:MI'),
   'MK_REP', 3600);
```

```
SELECT first_name, last_name,
   TO_CHAR(hire_date, 'dd-Mon-YYYY HH24:MI') As "Date and Time"
FROM copy_employees
WHERE employee_id = 303;
```

FIRST_NAME	LAST_NAME	Date and Time
Angelina	Wright	10-Jul-2017 17:20

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Subconsulta para Copiar Filas

- Cada sentencia INSERT que hemos visto hasta el momento solo agrega una fila a la tabla
- Sin embargo, suponga que deseamos copiar 100 filas de una tabla a otra
- No queremos tener que escribir y ejecutar 100 sentencias INSERT independientes, una tras otra
- Eso llevaría mucho tiempo
- Afortunadamente, SQL nos permite utilizar una subconsulta dentro de una sentencia INSERT

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

30

NOTA: en la lección 3 de esta sección, veremos la sentencia INSERT ALL. Esta permite agregar varias filas en una sentencia SQL.

Uso de una Subconsulta para Copiar Filas

- Todos los resultados de la subconsulta se insertan en la tabla
- Por lo tanto podamos copiar 100 filas (o 1000 filas) con una subconsulta de varias filas en INSERT
- Como era de esperar, no necesita una cláusula VALUES al utilizar una subconsulta para copiar filas porque los valores insertados serán exactamente los valores devueltos por la subconsulta

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Subconsulta para Copiar Filas

- En el ejemplo que se muestra, una nueva tabla denominada SALES_REPS se está rellenando con copias de algunas de las filas y columnas de la tabla EMPLEADOS
- La cláusula WHERE selecciona aquellos empleados que tengan identificadores de trabajo como "%REP%"

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
    SELECT employee_id, last_name, salary, commission_pct
    FROM   employees
    WHERE  job_id LIKE '%REP%';
```

ORACLE Academy

DP 12-1
Sentencias INSERT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

32

NOTA: para ejecutar la sentencia INSERT anterior, se tendrá que crear primero la tabla sales_reps.

Uso de una Subconsulta para Copiar Filas

- El número de columnas y sus tipos de dato de la lista de columnas de la cláusula INSERT deben coincidir con el número de columnas y sus tipos de dato de la subconsulta
- La subconsulta no está incluida entre paréntesis como se hace con las subconsultas en la cláusula WHERE de una sentencia SELECT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Subconsulta para Copiar Filas

- Si deseamos copiar todos los datos: todas las filas y todas las columnas: la sintaxis es aún más sencilla
- Para seleccionar todas las filas de la tabla EMPLEADOS e insertarlos en la tabla SALES_REPS, la sentencia se escribe de la siguiente forma:

```
INSERT INTO sales_reps
   SELECT *
   FROM   employees;
```

- De nuevo, esto solo funcionará si ambas tablas tienen el mismo número de columnas que coincida con los tipos de dato y que estén en el mismo orden

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - INSERT INTO
  - USER
  - Transaction
  - Explícito

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Explicar la importancia de poder modificar los datos de una base de datos
  - Construir y ejecutar sentencias INSERT que insertan una única fila con una cláusula VALUES
  - Construir y ejecutar sentencias INSERT que utilizan valores especiales, valores nulos y valores de fecha
  - Construir y ejecutar sentencias INSERT que copian filas de una tabla a otra mediante una subconsulta

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

Programación de Bases de Datos con SQL
12-2: Actualización de Valores de Columna y Supresión de Filas
Actividades de Práctica

Objetivos

- Crear y ejecutar una sentencia UPDATE
- Crear y ejecutar una sentencia DELETE
- Crear y ejecutar una consulta que utilice una subconsulta para actualizar y suprimir datos de una tabla
- Crear y ejecutar una consulta que utilice una subconsulta correlacionada para actualizar y suprimir datos de una tabla
- Explicar cómo las restricción de integridad de clave ajena y clave primaria afectan a las sentencias UPDATE y DELETE
- Explicar el objetivo de la cláusula FOR UPDATE en una sentencia SELECT

Terminología

Identifique el término para cada una de las siguientes definiciones.

	Modifica filas existentes en una tabla
	Recupera información de una tabla y utiliza esta información para actualizar otra tabla
	Se asegura de que los datos cumplen un juego de reglas predefinido
	Suprime la información en una tabla enlazada en función de lo que se ha suprimido en la otra tabla
	Elimina filas existentes en una tabla

Inténtelo/Resuélvalo

NOTA: Las tablas de copia no existen en esta sección

Si no es posible realizar algún cambio, explique el motivo.

1. Monique Tuttle, la jefa de Global Fast Foods, ha enviado una circular solicitando un cambio inmediato de los precios. El precio de un batido de fresa subirá de 3,59 a 3,75 $, y el precio de las patatas fritas subirá a 1,20 $. Realice los siguientes cambios en la tabla copy_f_food_items.

Copyright © 2020 Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

2. Bob Miller y Sue Doe han sido empleados destacados de Global Fast Foods. La dirección ha decidido recompensarlos aumentando su paga por horas extra. Bob Miller recibirá 0,75 $ adicionales por hora y Sue Doe recibirá 0,85 $ adicionales por hora. Actualice la tabla copy_f_staffs con estos nuevos valores. (Nota: Bob Miller actualmente no recibe paga por horas extra. ¿Qué función debe utilizar para convertir un valor nulo en 0?)

3. Agregue los pedidos que se muestran a continuación a la tabla copy_f_orders de Global Fast Foods:

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5680	June 12, 2004	159.78	145	9
5691	09-23-2004	145.98	225	12
5701	July 4, 2004	229.31	230	12

4. Agregue los nuevos clientes que se muestran a continuación a la tabla copy_f_customers. Es posible que ya haya agregado a Katie Hernandez. ¿Podrá agregar todos estos registros correctamente?

ID	FIRST_ NAME	LAST_ NAME	ADDRESS	CITY	STATE	ZIP	PHONE_NUMBER
145	Katie	Hernandez	92 Chico Way	Los Angeles	CA	98008	8586667641
225	Daniel	Spode	1923 Silverado	Denver	CO	80219	7193343523
230	Adam	Zurn	5 Admiral Way	Seattle	WA		4258879009

5. Sue Doe ha sido un miembro destacado del personal de Global Fast Foods y se le ha concedido un aumento salarial. Ahora recibirá la misma paga que Bob Miller. Actualice su registro en copy_f_staffs.

6. Global Fast Foods está ampliando su personal. La jefa, Monique Tuttle, ha contratado a Kai Kim. No está disponible toda la información en este momento, pero agregue la información que se muestra aquí.

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	STAFF_TYPE
25	Kai	Kim	3-Nov-1988	6,75	Order Taker

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios

7. Ahora que está disponible toda la información de Kai Kim, actualice su registro en Global Fast Foods para incluir lo siguiente: Kai tendrá el mismo jefe que Sue Doe. No podrá trabajar horas extra. Deje los valores de formación, presupuesto del jefe y objetivo del jefe como null.

8. Ejecute la siguiente sentencia SQL. Registre los resultados.

   ```
   DELETE from departments
   WHERE department_id = 60;
   ```

9. Kim Kai ha decidido volver a la universidad y no tiene tiempo para trabajar y asistir a las clases. Suprímalo de la plantilla de Global Fast Foods. Verifique que el cambio ha sido correcto.

10. Cree una copia de la tabla de empleados y denomínela lesson7_emp;
    Una vez creada esta tabla, escriba una sentencia delete correlacionada que suprima cualquier empleado de la tabla lesson7_employees que también exista en la tabla job_history.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y/o sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios

ORACLE

Academy

Database Programming with SQL

12-3

Valores DEFAULT, MERGE e Inserciones en Varias Tablas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Comprender cuándo especificar un valor DEFAULT
  - Crear y ejecutar una sentencia MERGE
  - Construir y ejecutar sentencias DML utilizando subconsultas
  - Construir y ejecutar inserciones en varias tablas

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Hasta ahora, ha estado actualizando los datos mediante una única sentencia INSERT
- Ha sido relativamente fácil al agregar registros uno a uno, ¿y si la compañía es muy grande y utiliza un almacén de datos para almacenar los datos de registros de ventas, de clientes, de nóminas, de contabilidad y de personal?
- En este caso, es probable que los datos procedan de varios orígenes y estén gestionados por varias personas
- La gestión de datos registro a registro puede ser muy confuso y llevar mucho tiempo

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

4

Un almacén de datos es una recopilación de datos diseñada para soportar la toma de decisiones relacionada con la gestión empresarial. Los almacenes de datos contienen una amplia variedad de datos como, por ejemplo, datos de ventas, de clientes, de nóminas, de contabilidad y de personal, que presentan una imagen coherente de las condiciones empresariales en un solo punto en el tiempo.

Objetivo

- ¿Cómo determina lo que se ha insertado nuevo o qué se ha cambiado recientemente?
- En esta lección, conocerá un método más eficaz para actualizar e insertar datos utilizando una secuencia de comandos INSERT y UPDATE condicionales en una única sentencia atómica
- También aprenderá a recuperar datos de una única subconsulta e INSERTAR las filas devueltas en más de una tabla de destino
- A medida que amplíe sus conocimientos en SQL, apreciará formas eficaces para realizar su trabajo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

DEFAULT

- Cada columna de una tabla puede tener un valor por defecto especificado para él
- En el caso de que se inserte una fila nueva y no haya ningún valor asignado para la columna, se asignará el valor por defecto en lugar de un valor nulo
- El uso de valores por defecto permite controlar dónde y cuándo se debe aplicar el valor por defecto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

DEFAULT

- El valor por defecto puede ser un valor literal, una expresión o una función SQL como SYSDATE o USER, pero el valor no puede ser el nombre de otra columna
- El valor por defecto debe coincidir con el tipo de dato de la columna
- Se puede especificar DEFAULT para una columna al crear o modificar la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de DEFAULT

- En el ejemplo siguiente, se muestra un valor por defecto especificado para la columna hire_date al crear la tabla:

```
CREATE TABLE my_employees (
      hire_date       DATE DEFAULT SYSDATE,
      first_name      VARCHAR2(15),
      last_name       VARCHAR2(15));
```

- Al agregar filas a esta tabla, SYSDATE se asignará a cualquier fila que no especifique explícitamente un valor hire_date

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

8

La creación y la modificación de tablas se tratarán más detalladamente más adelante en el curso.

DEFAULT explícito con INSERT

- Se pueden utilizar valores por defecto explícitos en las sentencias INSERT y UPDATE
- El ejemplo de INSERT con la tabla my_employees muestra el uso explícito de DEFAULT:

```
INSERT INTO my_employees
   (hire_date, first_name, last_name)
VALUES
   (DEFAULT, 'Angelina','Wright');
```

- Uso implícito de DEFAULT

```
INSERT INTO my_employees
   (first_name, last_name)
VALUES
   ('Angelina','Wright');
```

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

9

Si se ha especificado un valor por defecto para la columna hire_date al crear la tabla, Oracle asigna el valor por defecto a la columna. Sin embargo, si no se ha especificado ningún valor por defecto, Oracle asigna un valor nulo.

DEFAULT explícito con UPDATE

- Se pueden utilizar valores por defecto explícitos en las sentencias INSERT y UPDATE
- El ejemplo de UPDATE con la tabla my_employees muestra el uso explícito de DEFAULT:

```
UPDATE my_employees
SET hire_date = DEFAULT
WHERE last_name = 'Wright';
```

- Si se ha especificado un valor por defecto para la columna hire_date, se asigna a la columna el valor por defecto
- Sin embargo, si no se ha especificado ningún valor por defecto al crear la columna, se asigna un valor nulo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

MERGE

- El uso de la sentencia MERGE logra dos tareas al mismo tiempo. MERGE INSERTARÁ y ACTUALIZARÁ simultáneamente. Si falta un valor, se inserta uno nuevo
- Si existe un valor pero se debe cambiar, MERGE lo actualizará
- Para realizar estos tipos de cambios en tablas de base de datos, debe tener privilegios INSERT y UPDATE en la tabla de destino y privilegios SELECT en la tabla de origen
- Los alias se pueden utilizar con la sentencia MERGE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de MERGE

- Se lee una fila cada vez de la tabla de origen y se compara con las filas de la tabla de destino utilizando la condición de coincidencia
- Si existe una fila coincidente en la tabla de destino, la fila de origen se utiliza para actualizar una o más columnas de la fila de destino coincidente
- Si no existe una fila coincidente, los valores de la fila de origen se utilizan para insertar una nueva fila en la tabla de destino

```
MERGE INTO destination-table USING source-table
  ON matching-condition
WHEN MATCHED THEN UPDATE
    SET ……
WHEN NOT MATCHED THEN INSERT
    VALUES (……);
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de MERGE

- En este ejemplo, se utiliza la tabla EMPLOYEES (alias e) como origen de datos para insertar y actualizar filas en una copia de la tabla denominada COPY_EMP (alias c)

```
MERGE INTO copy_emp c  USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN UPDATE
     SET
         c.last_name   = e.last_name,
         c.department_id   = e.department_id
WHEN NOT MATCHED THEN INSERT
     VALUES   (e.employee_id, e.last_name, e.department_id);
```

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

13

create table copy_emp as (Select * from employees where department_id > 100);

select * from copy_emp;

update copy_emp
set last_name = 'Cotton'
where employee_id = 205;

select * from copy_emp;

MERGE INTO copy_emp c USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN UPDATE
  SET
  c.last_name = e.last_name,
  c.department_id = e.department_id
WHEN NOT MATCHED THEN INSERT
  VALUES (e.employee_id, e.first_name, e.last_name, e.email, e.phone_number, e.hire_date, e.job_id, e.salary, e.commission_pct, e.manager_id, e.department_id, e.bonus);

select * from copy_emp;

drop table copy_emp;

Ejemplo de MERGE

- Las filas 100 y 103 de EMPLOYEES tienen filas coincidentes en COPY_EMP y, por lo tanto, se actualizaron las filas coincidentes de COPY_EMP
- EMPLOYEE 142 no tiene ninguna fila coincidente y, por lo tanto, se insertó en COPY_EMP

EMPLOYEES (tabla de origen)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
103	Hunold	60
142	Davies	50

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

14

```
create table copy_emp as (Select * from employees where department_id > 100);

select * from copy_emp;

update copy_emp
set last_name = 'Cotton'
where employee_id = 205;

select * from copy_emp;

MERGE INTO copy_emp c  USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN UPDATE
   SET
   c.last_name = e.last_name,
   c.department_id = e.department_id
WHEN NOT MATCHED THEN INSERT
   VALUES (e.employee_id, e.first_name, e.last_name, e.email, e.phone_number, e.hire_date, e.job_id, e.salary,
e.commission_pct, e.manager_id, e.department_id, e.bonus);

select * from copy_emp;

drop table copy_emp;
```

Ejemplo de MERGE

EMPLOYEES (tabla de origen)

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
103	Hunold	60
142	Davies	50

COPY_EMP antes de ejecutar MERGE

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	Smith	40
103	Chang	30

COPY_EMP después de ejecutar MERGE

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
103	Hunold	60
142	Davies	50

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

15

```
create table copy_emp as (Select * from employees where department_id > 100);

select * from copy_emp;

update copy_emp
set last_name = 'Cotton'
where employee_id = 205;

select * from copy_emp;

MERGE INTO copy_emp c  USING employees e
  ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN UPDATE
   SET
   c.last_name = e.last_name,
   c.department_id = e.department_id
WHEN NOT MATCHED THEN INSERT
   VALUES (e.employee_id, e.first_name, e.last_name, e.email, e.phone_number, e.hire_date, e.job_id, e.salary,
e.commission_pct, e.manager_id, e.department_id, e.bonus);

select * from copy_emp;

drop table copy_emp;
```

Inserciones en Varias Tablas

- Las inserciones en varias tablas se utilizan cuando se deben insertar los mismos datos de origen en más de una tabla de destino
- Esta funcionalidad resulta útil cuando trabaja en un entorno de almacén de datos donde es común mover con regularidad datos de los sistemas operativos a un almacén de datos para la generación de informes analíticos y análisis
- La creación y la gestión de almacenes de datos es una forma de gestionar el a veces muy elevado número de filas insertadas en sistemas operativos durante un día laborable normal

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserciones en Varias Tablas

- Imagine, por ejemplo, ¿cuántas filas de datos debe crear su proveedor de telefonía diariamente para todas sus llamadas o mensajes de texto realizados en todos los dispositivos a los que tiene acceso a?
- A continuación, agregue a eso la navegación por Internet y las descargas de tonos, fondos de escritorio, juegos y otras aplicaciones móviles
- Multiplique ese número por el número total de clientes y eso puede darle una idea de la cantidad de datos que las compañías de telecomunicaciones tienen que gestionar

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserciones en Varias Tablas

- Puede que estas filas se tengan que insertar en más de una tabla del almacén de datos, por lo que si simplemente tenemos que SELECCIONARLAS una vez y, a continuación, replicarlas, se mejorará el rendimiento
- Las inserciones en varias pueden ser in condicionales o condicionales. En una inserción incondicional en varias tablas, Oracle insertará todas las filas devueltas por la subconsulta en todas las cláusulas de inserción de tabla encontradas en la sentencia
- En una inserción condicional en varias tablas, puede especificar ALL o FIRST

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserciones en Varias Tablas

- Especificación de ALL:
  - Si especifica ALL, valor por defecto, la base de datos evalúa cada cláusula WHEN independientemente de los resultados de la evaluación de cualquier otra cláusula WHEN
  - Para cada cláusula WHEN cuya condición se evalúe como true, la base de datos ejecuta la lista de cláusulas INTO correspondientes
- Especificación de FIRST:
  - Si especifica FIRST, la base de datos evalúa cada cláusula WHEN en el orden en el que aparece en la sentencia
  - Para la primera cláusula WHEN que se evalúe como true, la base de datos ejecuta la cláusula INTO correspondiente y omite las cláusulas WHEN posteriores de la fila determinada

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

19

Nota: hay ocasiones en que la cláusula WHEN no es necesaria, consulte el ejemplo de la diapositiva siguiente.

Inserciones en Varias Tablas

- Especificación de la cláusula ELSE:
  - Para una fila determinada, si la cláusula WHEN no se evalúa como true, la base de datos ejecuta la lista de cláusulas INTO asociada con la cláusula ELSE
  - Si no especifica una cláusula ELSE, la base de datos no realiza ninguna acción para dicha fila

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Para obtener ejemplos más detallados incluido el uso de FIRST y ELSE, visite http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_9014.htm#i2125362

Sintaxis de Inserciones en Varias Tablas

- La sintaxis de la sentencia de inserción en varias tablas es la siguiente:

```
INSERT ALL INTO clause VALUES clause SUBQUERY
```

- Un ejemplo de la sentencia de inserción en varias tablas es el siguiente:

```
INSERT ALL
   INTO my_employees
      VALUES (hire_date, first_name, last_name)
   INTO copy_my_employees
      VALUES (hire_date, first_name, last_name)
SELECT hire_date, first_name, last_name
FROM employees;
```

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

21

Esta inserción en varias tablas recuperará las filas de la sentencia SELECT e INSERTARÁ las filas en la tabla my_employees y copy_my_employees.

INSERT ALL necesita una sentencia SELECT para ejecutarse. Sin embargo, podemos utilizar la tabla ficticia DUAL para realizar varias sentencias INSERT en una única tabla, por ejemplo:

```
INSERT ALL
 INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary)
 VALUES
  (304,'James','Parrot', 'jparrot', '912-699-4656', '15/Jul/2015', 'IT_PROG',4200)
 INTO copy_employees
  (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary)
 VALUES
  (305,'Rebecca','Graham', 'rgraham', '480-678-4259', '15/Jun/2015', 'IT_PROG',4200)
SELECT * FROM dual;
```

Inserciones Condicionales en Varias Tablas

```
INSERT ALL
   WHEN call_ format IN ('tlk','txt','pic') THEN
   INTO all_calls
      VALUES (caller_id, call_timestamp, call_duration, call_format)
   WHEN call_ format IN ('tlk','txt') THEN
   INTO police_record_calls
      VALUES (caller_id, call_timestamp, recipient_caller)
   WHEN call_duration < 50 AND call_type = 'tlk' THEN
   INTO short_calls
      VALUES (caller_id, call_timestamp, call_duration)
   WHEN call_duration > = 50 AND call_type = 'tlk' THEN
   INTO long_calls
      VALUES (caller_id, call_timestamp, call_duration)
SELECT  caller_id, call_timestamp, call_duration, call_format,
        recipient_caller
FROM calls
WHERE TRUNC(call_timestamp ) = TRUNC(SYSDATE);
```

ORACLE Academy

DP 12-3
Valores DEFAULT, MERGE e Inserciones en Varias Tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

22

Esta inserción en varias tablas utiliza la condición WHEN. Si una fila de la sentencia SELECT coincide con los criterios de la condición WHEN, la fila se agrega a la tabla en la cláusula INTO.

Nota: este ejemplo se utiliza para demostrar un uso en el mundo real para inserciones en varias tablas, pero como no existen tablas esta consulta no se ejecutarán en APEX.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - DEFAULT
  - MERGE
  - Inserciones en Varias Tablas
  - ALL, FIRST y ELSE

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Comprender cuándo especificar un valor DEFAULT
  - Crear y ejecutar una sentencia MERGE
  - Construir y ejecutar sentencias DML utilizando subconsultas
  - Construir y ejecutar inserciones en varias tablas

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

13-1

Creación de tablas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Enumerar y clasificar los principales objetos de base de datos
  - Revisar la estructura de una tabla
  - Describir cómo la base de datos Oracle utiliza los objetos de esquema
  - Crear una tabla mediante el tipo de dato adecuado para cada columna
  - Explicar el uso de tablas externas
  - Consultar el diccionario de datos para obtener los nombres y otros atributos de objetos de la base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Hasta ahora, ha seleccionado, actualizado, insertado y suprimido información en las tablas existentes de una base de datos
- Como administrador de base de datos (DBA), se espera que sepa también cómo crear tablas
- En esta lección, conocerá los objetos de base de datos que más se utilizan, cómo mirar la estructura de la tabla y cómo crear tablas nuevas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Las tablas serán pequeñas en comparación con las tablas que contienen millones de filas y cientos de columnas, pero la creación de una tabla pequeña implica las mismas sentencias SQL y sintaxis que la creación de una muy grande
- También aprenderá sobre las tablas externas, tablas que son similares en cuanto a estructura a las tablas de base de datos Oracle normales, pero las filas de datos reales se almacenan de forma externa un archivo plano y solo se acceder a ellas cuando es necesario

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- Una base de datos Oracle puede contener muchos tipos distintos de objetos
- En esta sección se presentan los objetos más utilizados y también se describe cómo Oracle Server utiliza la información almacenada en el diccionario de datos cuando realiza el trabajo como resultado de las sentencias SQL que emita

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- Los principales tipos de objetos de base de datos son:
  - Tabla
  - Índice de
  - ¿La restricción
  - Ver
  - Secuencia
  - Sinónimo
- Algunos de estos tipos de objetos pueden existir de forma independiente y otros no pueden

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- Algunos de los tipos de objetos ocupan espacio, lo que se denomina almacenamiento, en la base de datos y otros no
- Los objetos de base de datos que ocupan mucho espacio de almacenamiento se denominan segmentos
- Las tablas y los índices son ejemplos de segmentos, ya que los valores almacenados en las columnas de cada fila ocupan mucho espacio en el disco físico
- Las vistas, restricciones, secuencias y sinónimos también son objetos, pero el único espacio que necesitan en la base de datos está en la definición del objeto: ninguno de ellos tiene filas de datos asociadas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- La base de datos almacena las definiciones de todos los objetos de base de datos en el diccionario de datos y estas definiciones están accesibles para todos los usuarios de la base de datos así como para la base de datos en sí
- ¿Se ha preguntado alguna vez cómo sabe Oracle qué columnas devolver de una consulta?
- Por ejemplo, si especifica SELECT * FROM jobs en lugar de SELECT job_id, job_title FROM jobs, ¿cómo sabe Oracle qué columnas devolver?
- La base de datos busca la definición de la tabla utilizada en la consulta, traduce el "*" en la lista completa de columnas, y devuelve el resultado

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- La base de datos utiliza el diccionario de datos para todas las sentencias que emita, incluso aunque enumere los nombres de columna en lugar de utilizar "*"
- Comprueba que las tablas a las que hace referencia en la sentencia existen en la base de datos, comprueba que los nombres de columna son correctos, comprueba si tiene los privilegios correctos para realizar la acción que está solicitando y, finalmente, utiliza el diccionario de datos para decidir el plan de ejecución, cómo realizará en realidad la solicitud

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- Todos los usuarios de la base de datos pueden consultar el diccionario de datos
- En Application Express, se puede acceder a él mediante sentencias SQL en la interfaz SQL Workshop> SQL Commands

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- En la ventana SQL Commands, debe saber los nombres de la tabla que consulta y en la interfaz Object Browser, solo tiene que hacer clic en los objetos mostrados para ver sus detalles
- Si desea ver los detalles de una tabla, simplemente haga clic en su título en la lista de la tabla
- Con el Explorador de objetos, puede ver los detalles de las tablas, así como las opciones para ver los datos, índices, restricciones, concesiones y otros detalles de la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetos de Esquema de la Base de Datos

- Mediante la ventana SQL Commands, debe solicitar una DESCripción de la tabla
- Ninguna de las opciones adicionales que ofrece Object Browser está disponible en esta interfaz

```
DESCRIBE  jobs;
```

Results | Explain | **Describe** | Saved SQL | History

Object Type **TABLE** Object **JOBS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOBS	JOB_ID	VARCHAR2	10	-	-	1	-	-	-
	JOB_TITLE	VARCHAR2	35	-	-	-	-	-	-
	MIN_SALARY	NUMBER	-	6	0	-	✓	-	-
	MAX_SALARY	NUMBER	-	6	0	-	✓	-	-

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Tabla

- Todos los datos de una base de datos relacional se almacenan en tablas
- Al crear una nueva tabla, utilice las siguientes reglas para los nombres de tabla y columnas:
  - Deben empezar por una letra
  - Deben tener entre 1 y 30 caracteres
  - Solo deben contener A - Z, a - z, 0 - 9, _ (carácter de subrayado), $ y #
  - No deben ser un duplicado de otro nombre de objeto propiedad del mismo usuario
  - No deben ser una palabra reservada de Oracle Server

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Reglas de Nomenclatura

- Es mejor utilizar nombres descriptivos para las tablas y otros objetos de la base de datos
- Si una tabla va a almacenar información sobre los alumnos, asígnele el nombre ALUMNOS, no PERSONAS o NIÑOS
- Los nombres de tablas no son sensibles a mayúsculas/minúsculas
- Por ejemplo, ALUMNOS se trata igual que ALuMnoS o alumnos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Reglas de Nomenclatura

- Los nombres de tabla deben ser en plural, por ejemplo, ALUMNOS, no alumno
- La creación de tablas forma parte del lenguaje de definición de datos (DDL) de SQL
- En las sentencias DDL que se utilizan para configurar, cambiar y eliminar estructuras de datos de tablas se incluyen ALTER, DROP, RENAME y TRUNCATE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

CREATE TABLE

- Para crear una nueva tabla, debe tener el privilegio CREATE TABLE y un área de almacenamiento para él
- El administrador de la base datos utiliza sentencias de lenguaje de control de datos (DCL) para otorgar este privilegio a los usuarios y asignar un área de almacenamiento
- Las tablas que pertenecen a otros usuarios no se incluyen en su esquema
- Si desea utilizar una tabla que no está en su esquema, utilice el nombre del propietario de la tabla como prefijo para el nombre de tabla:

```
SELECT *
FROM mary.students;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

CREATE TABLE

- Si desea utilizar una tabla que no está en su esquema, utilice el nombre del propietario de la tabla como prefijo para el nombre de tabla:

```
SELECT *
FROM mary.students;
```

  - Se le debe otorgar acceso a una tabla para poder realizar selecciones en ella

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de CREATE TABLE

- Para crear una nueva tabla, utilice los siguientes detalles de sintaxis:
  - table es el nombre de la tabla
  - column es el nombre de la columna
  - datatype es el tipo de dato y la longitud de la columna
  - La expresión DEFAULT especifica un valor por defecto si se omite un valor en la sentencia INSERT

```
CREATE TABLE table
(column data type [DEFAULT expression],
(column data type [DEFAULT expression],
(……[ ] );
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de CREATE TABLE

- Los ejemplos siguientes muestran la sentencia CREATE TABLE:

```
CREATE TABLE my_cd_collection
 (cd_number NUMBER(3),
  title VARCHAR2(20),
  artist VARCHAR2(20),
  purchase_date DATE DEFAULT SYSDATE);
```

```
CREATE TABLE my_friends
 (first_name VARCHAR2(20),
  last_name VARCHAR2(30),
  email VARCHAR2(30),
  phone_num VARCHAR2(12),
  birth_date DATE);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tablas Externas

- Oracle también soporta otro tipo de tabla: tabla externa
- En una tabla externa, las filas de datos no están en los archivos de la base de datos, sino que en su lugar se encuentran en un archivo plano, almacenado de forma externa a la base de datos
- Normalmente, una tabla externa se utiliza para almacenar datos migrados de versiones anteriores de las bases de datos utilizadas por una compañía

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tablas Externas

- Cuando una compañía implanta una nueva aplicación y base de datos, suele tener que importar la mayoría de los datos de los sistemas antiguos al nuevo sistema para el acceso de lectura y escritura normal, pero puede haber algunos datos que no se utilicen con frecuencia y, por lo tanto, solo se tenga que acceder a ellos para lectura
- Este tipo de dato podría alojarse en una tabla externa

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tablas Externas

- Una de las principales ventajas de Oracle es que solo se tiene que hacer una copia de seguridad una vez de los datos alojados en tablas externas y, a continuación, nunca más a menos que el contenido del archivo cambie
- La sintaxis para crear una tabla externa es muy similar a la de crear una tabla estándar, excepto por el hecho de que tiene sintaxis adicional al final

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tablas Externas

- La nueva sintaxis (que aparece en rojo) de las siguientes diapositivas no se utiliza en sentencias SQL estándar para la creación de tablas
- ORGANIZATION EXTERNAL: indica a Oracle que cree una tabla externa
- TYPE ORACLE_LOADER: de tipo Oracle Loader (producto de Oracle)
- DEFAULT DIRECTORY def_dir1: nombre del directorio para el archivo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tablas Externas

- ACCESS PARAMETERS: cómo leer el archivo
- RECORDS DELIMITED BY NEWLINE: cómo identificar el principio de una nueva fila
- FIELDS: especificaciones de nombre de campo y tipo de dato
- LOCATION: nombre del archivo real que contiene los datos
- Un ejemplo de la nueva sintaxis se muestra en rojo en la siguiente diapositiva

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Tablas Externas

```
CREATE TABLE emp_load
  (employee_number CHAR(5),
   employee_dob CHAR(20),
   employee_last_name CHAR(20),
   employee_first_name CHAR(15),
   employee_middle_name CHAR(15),
   employee_hire_date DATE)
ORGANIZATION EXTERNAL
   (TYPE ORACLE_LOADER
    DEFAULT DIRECTORY def_dir1
    ACCESS PARAMETERS
   (RECORDS DELIMITED BY NEWLINE
    FIELDS (employee_number CHAR(2),
                    employee_dob CHAR(20),
                    employee_last_name CHAR(18),
                    employee_first_name CHAR(11),
                    employee_middle_name CHAR(11),
                    employee_hire_date CHAR(10) date_format DATE mask
"mm/dd/yyyy"))
   LOCATION ('info.dat'));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Existen dos tipos de tablas en una base de datos Oracle: tablas de usuario y tablas de diccionario de datos
- Se pueden emitir sentencias SQL para acceder a ambos tipos de tablas: puede seleccionar, insertar, actualizar y suprimir datos de las tablas de usuario y puede seleccionar datos de la tabla de diccionario de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Las tablas de usuario creadas por el usuario que contiene sus datos:
  - empleados, departamentos, cargos, etc.
- Tablas de diccionario de datos:
  - DICTIONARY, USER_OBJECTS, USER_TABLES, USER_SEGMENTS, USER_INDEXES, etc.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Las tablas de diccionario de datos son propiedad de un usuario de Oracle especial llamado SYS y solo se deben utilizar sentencias SELECT al trabajar con cualquiera de estas tablas
- Para proteger a estas tablas del acceso accidental de usuario, todas tienen vistas creadas a través de las que los usuarios de la base de datos acceden al diccionario de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Si cualquier usuario de Oracle intenta realizar inserciones, actualizaciones o supresiones en una de las tablas de diccionario de datos, no se permite la operación ya que puede poner en peligro la integridad de toda la base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Al utilizar las vistas del diccionario de datos en la interfaz SQL Commands, debe saber los nombres de las vistas del diccionario con las que está trabajando
- En Oracle, esto es bastante sencillo: ponga como prefijo al tipo de objeto que está buscando USER_xxx o ALL_xxx,
  donde xxx es el tipo de objeto

```
SELECT table_name, status
FROM USER_TABLES;
```

```
SELECT table_name, status
FROM ALL_TABLES;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Data Dictionary

- Por lo tanto, si desea investigar los índices, simplemente seleccione USER_INDEXES; si desea información acerca de las secuencias, la tabla es USER_SEQUENCES, etc.

```
SELECT *
FROM user_indexes;
```

```
SELECT *
FROM user_objects
WHERE object_type = 'SEQUENCE';
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - CREATE TABLE
  - Diccionario de datos
  - Tabla
  - Tabla externa
  - Esquema
  - DEFAULT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Clasificación de los Principales Objetos de Base de Datos
  - Revisar la estructura de una tabla
  - Describir cómo se utilizan los objetos de esquema
  - Crear una tabla mediante el tipo de dato adecuado para cada columna
  - Explicar el uso de tablas externas
  - Utilizar el diccionario de datos para obtener los nombres y otros atributos de objetos de la base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

13-2

Uso de Tipos de Datos

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Crear una tabla utilizando los tipos de dato de columna TIMESTAMP y TIMESTAMP WITH TIME ZONE
  - Crear una tabla utilizando los tipos de dato de columna INTERVAL YEAR TO MONTH e INTERVAL DAY TO SECOND
  - Dar ejemplos de organizaciones y situaciones personales en las que es importante conocer a qué zona horaria se refiere un valor de fecha y hora
  - Enumerar y dar un ejemplo de cada uno de los tipos de dato de número, fecha y carácter

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Si alguna vez viaja a otro país, descubrirá rápidamente que el dinero en su bolsillo puede no ser el de la moneda local
- Si desea comprar algo, será necesario convertir el dinero en la moneda del país local
- Este proceso de conversión se parece mucho a tratar con tipos de dato en SQL
- Los distintos tipos de dato tienen tipos diferentes de características, cuyo objetivo es almacenar de manera eficaz los datos
- En esta lección, aprenderá más sobre los tipos de dato y sus usos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visión General de Tipo de Dato

- Cada valor manipulado por Oracle tiene un tipo de dato
- El tipo de dato de un valor asocia un juego fijo de propiedades al valor
- Estas propiedades hacen que la base de datos trate los valores de un tipo de dato de forma diferente a valores de otro tipo de dato

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visión General de Tipo de Dato

- Los distintos tipos de dato ofrecen varias ventajas:
  - Las columnas de un solo tipo producen resultados consistentes
  - Por ejemplo, las columnas del tipo de dato DATE siempre producen valores de fecha
  - No se puede insertar el tipo de dato incorrecto en una columna. Por ejemplo, las columnas del tipo de dato DATE impedirán que se inserten datos de tipo NUMBER
- Por este motivo, cada columna de una base de datos relacional solo puede contener un tipo de dato
- No puede combinar los tipos de dato en una columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato Comunes

- A continuación se muestran los tipos de dato de columna más utilizados para los valores de carácter y número
- Para valores de carácter:
  - CHAR (tamaño fijo, máximo de 2000 caracteres)
  - VARCHAR2 (tamaño variable, máximo de 4000 caracteres)
  - CLOB (tamaño variable, máximo de 128 terabytes)
- Para valores de número:
  - NUMBER (tamaño variable, precisión máxima de 38 dígitos)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato Comunes

- A continuación se muestran los tipos de dato de columna más utilizados para los valores de fecha, hora y binario
- Para valores de fecha y hora:
  - DATE
  - TIMESTAMP ….
  - INTERVAL
- Para valores binarios (p. ej., multimedia: JPG, WAV, MP3, etc.):
  - RAW (tamaño variable, máximo de 2000 bytes)
  - BLOB (tamaño variable, máximo de 128 terabytes)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato Comunes

- Para valores de caracteres, normalmente es mejor utilizar VARCHAR2 o CLOB que CHAR, porque se ahorra espacio
- Por ejemplo, el apellido de un empleado es "Chang"
- En una columna VARCHAR2(30), solo se almacenan los 5 caracteres importantes: C h a n g
- Pero en una columna CHAR(30), se almacenarán también 25 espacios finales, para formar un tamaño fijo de 30 caracteres
- Los valores de número pueden ser negativos, así como positivos. Por ejemplo, NUMBER(6,2) puede almacenar cualquier valor de +9999,99 hasta -9999,99

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato DATE-TIME

- El tipo de dato DATE almacena un valor de siglos hasta segundos enteros, pero no se pueden almacenar las fracciones de un segundo
- "21-Aug-2003 17:25:30" es un valor válido, pero "21-Aug-2003 17:25:30.255" no lo es
- El tipo de dato TIMESTAMP es una extensión del tipo de dato DATE que permite fracciones de un segundo
- Por ejemplo, TIMESTAMP(3) permite 3 dígitos después de los segundos completos, lo que permite almacenar valores de hasta milisegundos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato DATE-TIME

- Ejemplo de TIMESTAMP:

```
CREATE TABLE time_ex1
(exact_time TIMESTAMP);
```

```
INSERT INTO time_ex1
VALUES ('10-Jun-2017 10:52:29.123456');
```

```
INSERT INTO time_ex1
VALUES (SYSDATE);
```

```
INSERT INTO time_ex1
VALUES (SYSTIMESTAMP);
```

```
SELECT *
FROM time_ex1;
```

EXACT_TIME
10-JUN-15 10.52.29.123456 AM
16-JUL-15 08.17.08.000000 AM
16-JUL-15 08.17.16.610293 AM

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TIMESTAMP…With [LOCAL] Time Zone

- Piense en el valor de hora "17:30". Por supuesto, significa "cinco y media de la tarde"
- ¿Pero en qué zona horaria?
- ¿Es cinco y media de la hora de Nueva York, de Pekín, de Estambul tiempo... ?
- En las organizaciones globalizadas actuales que operan en muchos países distintos, es importante saber a qué zona horaria hacer referencia un valor de fecha-hora

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TIMESTAMP…With [LOCAL] Time Zone

- TIMESTAMP WITH TIME ZONE almacena un valor de zona horaria como un desplazamiento de la Hora Universal Coordinada o UCT (también conocida como Hora Media de Greenwich o GMT)
- Un valor "21-Aug-2003 08:00:00 –5:00" significa 8:00 a.m. 5 horas por debajo de UTC
- Esta es Hora Oficial Oriental de EE. UU. (EST)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TIMESTAMP…With [LOCAL] Time Zone

- Ejemplo de TIMESTAMP WITH TIME ZONE:

```
CREATE TABLE time_ex2
(time_with_offset TIMESTAMP WITH TIME ZONE);
```

```
INSERT INTO time_ex2
VALUES (SYSTIMESTAMP);
```

```
INSERT INTO time_ex2
VALUES ('10-Jun-2017 10:52:29.123456 AM +2:00');
```

```
SELECT *
FROM time_ex2;
```

TIME_WITH_OFFSET
16-JUL-15 08.49.47.126056 AM -07:00
10-JUN-15 10.52.29.123456 AM +02:00

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TIMESTAMP…With [LOCAL] Time Zone

- TIMESTAMP WITH LOCAL TIME ZONE es similar, pero con una diferencia: si esta columna se seleccionado en una sentencia SQL, la hora se convierte automáticamente a la zona horaria del usuario que realiza la selección
- Ejemplo de TIMESTAMP With…Time Zone:

```
CREATE TABLE time_ex3
( first_column TIMESTAMP WITH TIME ZONE,
  second_column TIMESTAMP WITH LOCAL TIME ZONE);
```

```
INSERT INTO time_ex3
  (first_column, second_column)
VALUES
  ('15-Jul-2017 08:00:00 AM -07:00', '15-Nov-2007 08:00:00');
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de TIMESTAMP…With Time Zone

- Ambos valores se almacenan con un desplazamiento de hora de –07:00 horas (PST)
- Pero ahora un usuario en Estambul ejecuta:

```
SELECT *
FROM time_ex3;
```

FIRST_COLUMN	SECOND_COLUMN
15-JUL-15 08.00.00.000000 AM -07:00	15-NOV-07 05.00.00.000000 PM

- La hora de Estambul es de 9 horas por delante de PST; cuando son las 8:00 a.m. en Los Ángeles, son las 5:00 p.m. en Estambul

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Dato INTERVAL

- Estos almacenan el tiempo o intervalo de tiempo transcurrido entre dos valores de fecha y hora
- INTERVAL YEAR TO MONTH almacena un período de tiempo medido en años y meses
- INTERVAL DAY TO SECOND almacena un período de tiempo medido en días, horas, minutos y segundos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

INTERVAL YEAR…TO MONTH

- Sintaxis:

```
INTERVAL YEAR [(year_precision)] TO MONTH
```

- year_precision es el número máximo de dígitos del elemento YEAR
- El valor por defecto de year_precision es 2

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

INTERVAL YEAR...TO MONTH

- En este ejemplo se muestra INTERVAL YEAR TO MONTH:

```
CREATE TABLE time_ex4
(loan_duration1 INTERVAL YEAR(3) TO MONTH,
 loan_duration2 INTERVAL YEAR(2) TO MONTH);
```

```
INSERT INTO time_ex4 (loan_duration1, loan_duration2)
VALUES (INTERVAL '120' MONTH(3),
        INTERVAL '3-6' YEAR TO MONTH);
```

Suponga que la fecha de hoy es: 17-Jul-2017

```
SELECT SYSDATE + loan_duration1 AS "120 months from now",
       SYSDATE + loan_duration2 AS "3 years 6 months from
now"
FROM time_ex4;
```

120 months from now (120 meses a partir de ahora)	3 years 6 months from now (3 años y 6 meses a partir de ahora)
17-Jul-2027	17-Jan-2021

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

INTERVAL DAY…TO SECOND

- Utilice esta opción cuando necesite una diferencia más precisa entre dos valores de fecha-hora
- Sintaxis:

```
INTERVAL DAY [day_precision)] TO SECOND
[(fractional_seconds_precision)]
```

- Day_precision es el número máximo de dígitos del elemento DAY
- El valor por defecto de day_precision es 2
- fractional_seconds_precision es el número de dígitos de la parte fraccionaria del campo de fecha/hora SECOND
- El valor por defecto es 6

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

INTERVAL DAY…TO SECOND

- En este ejemplo se muestra INTERVAL DAY TO SECOND:

```
CREATE TABLE time_ex5
(day_duration1 INTERVAL DAY(3) TO SECOND,
 day_duration2 INTERVAL DAY(3) TO SECOND);
```


```
CREATE TABLE time_ex5
(day_duration1 INTERVAL DAY(3) TO SECOND,
 day_duration2 INTERVAL DAY(3) TO SECOND);
```


```
SELECT SYSDATE + day_duration1 AS  "25 Days from now",
   TO_CHAR(SYSDATE + day_duration2, 'dd-Mon-yyyy hh:mi:ss')
   AS  "precise days and time from now"
FROM time_ex5;
```

25 Days from now (25 días a partir de ahora)	precise days and time from now (Días y hora precisos a partir de ahora)
11-Aug-2017	21-Jul-2017 01:13:17

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - CLOB
  - BLOB
  - TIMESTAMP
  - TIMESTAMP WITH TIMEZONE
  - TIMESTAMP WITH LOCAL TIMEZONE
  - INTERVAL DAY TO SECOND
  - INTERVAL DAY TO MONTH

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, ha aprendido lo siguiente:
  - Crear una tabla utilizando los tipos de dato de columna TIMESTAMP y TIMESTAMP WITH TIME ZONE
  - Crear una tabla utilizando los tipos de dato de columna INTERVAL YEAR TO MONTH e INTERVAL DAY TO SECOND
  - Dar ejemplos de organizaciones y situaciones personales en las que es importante conocer a qué zona horaria se refiere un valor de fecha y hora
  - Enumerar y dar un ejemplo de cada uno de los tipos de dato de número, fecha y carácter

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

13-3

Modificación de una Tabla

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Explicar por qué es importante poder modificar una tabla
  - Explicar y proporcionar un ejemplo para cada una de las sentencias DDL (ALTER, DROP, RENAME y TRUNCATE) y el efecto que tiene cada una en las tablas y columnas
  - Construir una consulta y ejecutar los comandos ADD, MODIFY y DROP de ALTER TABLE
  - Explicar y ejecutar FLASHBACK QUERY en una tabla
  - Explicar y realizar operaciones FLASHBACK table

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Seguimiento de los Cambios en los Datos a lo Largo de un Período de Tiempo
  - Explicar los motivos para utilizar TRUNCATE frente a DELETE en las tablas
  - Agregar un comentario a una tabla utilizando el comando COMMENT ON TABLE
  - Enumerar los cambios que se pueden y no se pueden realizar en una columna
  - Explicar cuándo y por qué la sentencia SET UNUSED es ventajosa

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Recuerde la frase "No hay nada permanente excepto el cambio"?
- ¿No sería fantástico si nunca cometiéramos errores ni necesitáramos cambiar nada?
- Como sabe hasta ahora, las bases de datos son entidades dinámicas
- Probablemente no serían muy útiles si no se pudieran cambiar

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Hasta ahora, ha creado tablas y realizado los cambios en los datos de fila de tablas, ¿pero cómo realiza cambios en las propias tablas?
- En esta lección se presentan los comandos DDL que se utilizan para modificar, renombrar, vaciar o simplemente eliminar una tabla por completo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE

- Las sentencias ALTER TABLE se utilizan para:
  - Agregar una nueva columna
  - Modificar una columna existente
  - Definir un valor DEFAULT para una columna
  - Borrar una columna
- Puede agregar o modificar una columna de una tabla, pero no puede especificar donde aparece la columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE

- Una columna recién agregada siempre se convierte en la última columna de la tabla
- Además, si una tabla ya tiene filas de datos y agrega una nueva columna a la tabla, la nueva columna inicialmente será nula para todas las filas ya existentes

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE: Adición de una Columna

- Para agregar una nueva columna, utilice la sintaxis SQL que se muestra a continuación:

```
ALTER TABLE tablename
ADD (column name data type [DEFAULT expression],
column name data type [DEFAULT expression], ...
```

- Por ejemplo:

```
ALTER TABLE my_cd_collection
ADD (release_date DATE DEFAULT SYSDATE);
```

```
ALTER TABLE my_friends
ADD (favorite_game VARCHAR2(30));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE: Modificación de una Columna

- La modificación de una columna puede incluir cambios en el tipo de dato, tamaño y valor DEFAULT de una columna
- Las reglas y las restricciones al modificar una columna son las siguientes:
  - Puede aumentar el ancho o la precisión de una columna numérica
  - Puede aumentar el ancho de una columna de caracteres
  - Puede reducir el ancho de una columna NUMBER si la columna contiene únicamente valores nulos o si la tabla no tiene filas
  - Para tipos VARCHAR, puede reducir el ancho hasta el mayor valor incluido en la columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE: Modificación de una Columna

- Puede cambiar el tipo de dato solo si la columna contiene valores nulos
- Solo puede convertir una columna CHAR a VARCHAR2 o una columna VARCHAR2 a CHAR si la columna contiene valores nulos, o bien si no cambia el tamaño a algo más pequeño que cualquier valor de la columna
- El cambio del valor DEFAULT de una columna silo afecta a las inserciones posteriores en la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de ALTER TABLE: Modificación de una Columna

- Ejemplo: se ha creado una tabla con dos columnas:

```
CREATE TABLE mod_emp
  (last_name VARCHAR2(20),
   salary NUMBER(8,2));
```

- ¿Cuál de estas modificaciones estaría permitida y cuál no?
- Considere las respuestas con y sin filas de datos en la tabla

```
ALTER TABLE mod_emp
     MODIFY (last_name VARCHAR2(30));
ALTER TABLE mod_emp
     MODIFY (last_name VARCHAR2(10));
ALTER TABLE mod_emp
     MODIFY (salary NUMBER(10,2));
ALTER TABLE mod_emp
     MODIFY (salary NUMBER(8,2) DEFAULT 50);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de ALTER TABLE: Modificación de una Columna

- Solo estaría permitida si las columnas estuvieran vacías o el nombre más largo tuviera 10 caracteres o menos

```
ALTER TABLE mod_emp
     MODIFY (last_name VARCHAR2(10));
```

- Estaría permitida con o sin datos ya que aumenta el ancho de columna

```
ALTER TABLE mod_emp
     MODIFY (last_name VARCHAR2(30));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de ALTER TABLE: Modificación de una Columna

- Estaría permitida con o sin datos ya que aumenta la precisión de columna

```
ALTER TABLE mod_emp
     MODIFY (salary NUMBER(10,2));
```

- Se permitiría con o sin datos, ya que solo se agrega un valor DEFAULT

```
ALTER TABLE mod_emp
     MODIFY (salary NUMBER(8,2) DEFAULT 50);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE: Borrado de una Columna

- Al borrar una columna se aplican las siguientes reglas:
  - Se puede borrar una columna que contenga datos
  - Solo se puede borrar una columna cada vez
  - No puede borrar todas las columnas de una tabla; debe permanecer al menos una columna
  - Una vez borrada la columna, los valores de datos de esta no se pueden recuperar

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ALTER TABLE: Borrado de una Columna

- Sintaxis SQL:

```
ALTER TABLE tablename
DROP COLUMN column name;
```

- Por ejemplo:

```
ALTER TABLE my_cd_collection
DROP COLUMN release_date;
```

```
ALTER TABLE my_friends
DROP COLUMN favorite_game;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SET UNUSED para Columnas

- Borrar una columna de una tabla grande puede llevar mucho tiempo
- Una alternativa más rápida es marcar la columna como no utilizable
- Los valores de columna permanecen en la base de datos, pero no se puede acceder a ellos de ninguna forma, por lo que el efecto es el mismo que borrar la columna
- De hecho, puede agregar una nueva columna a la base de datos con el mismo nombre que la columna no utilizada
- Las columnas no utilizadas existen, pero son invisibles
- Sintaxis: `ALTER TABLE tablename SET UNUSED (column name);`

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de SET UNUSED para Columnas

- Ejemplo:

```
ALTER TABLE copy_employees
   SET UNUSED (email);
```

- DROP UNUSED COLUMNS elimina todas las columnas marcadas actualmente como no utilizadas
- Utilice esta sentencia cuando desee reclamar el espacio en disco adicional de las columnas no utilizadas en una tabla
- Ejemplo:

```
ALTER TABLE copy_employees
   DROP UNUSED COLUMNS;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen de ALTER TABLE

- En este gráfico se resumen los usos del comando ALTER TABLE:

Sintaxis	Resultados	Problemas
ALTER TABLE tablename ADD (column name data type [DEFAULT expression], column name data type [DEFAULT expression], …	Agrega una nueva columna a la tabla	No puede especificar dónde debe aparecer la columna en la tabla. Se convierte en la última columna.
ALTER TABLE tablename MODIFY (column name data type [DEFAULT expression], column name data type, …	Se utiliza para cambiar el tipo de dato, tamaño y valor por defecto de una columna.	El cambio del valor por defecto de una columna solo afecta a las inserciones posteriores en la tabla.
ALTER TABLE tablename DROP COLUMN column name;	Se utiliza para borrar una columna de una tabla	Debe quedar al menos una columna en la tabla después de modificarla. Una vez borrada, la columna no se puede recuperar.
ALTER TABLE tablename SET UNUSED (column name);	Se utiliza para marcar una o más columnas para que se puedan borrar más tarde	No restaura espacio en disco. Las columnas se tratan como si se hubieran borrado.
ALTER TABLE tablename DROP UNUSED COLUMNS	Elimina de la tabla todas las columnas marcadas actualmente como no utilizadas.	Una vez definidas como no utilizadas, no hay acceso a las columnas; no se muestra ningún dato con DESCRIBE. Eliminación permanente; sin rollback.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

DROP TABLE

- La sentencia DROP TABLE elimina la definición de una tabla de Oracle
- La base de datos pierde todos los datos de la tabla y los índices asociados a los mismos
- Cuando se emite una sentencia DROP TABLE:
  - Se suprimen todos los datos de la tabla
  - Se elimina la descripción de la tabla del diccionario de datos
- Oracle Server no cuestiona la decisión y borra la tabla inmediatamente

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

DROP TABLE

- En la siguiente diapositiva, verá que puede restaurar una tabla después borrarla, pero no está garantizado
- Solo el creador de la tabla o un usuario con el privilegio DROP ANY TABLE (normalmente solo el DBA) puede eliminar una tabla
- Sintaxis:

```
DROP TABLE tablename;
```

- Ejemplo:

```
DROP TABLE copy_employees;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK TABLE

- Si borra una tabla por error, puede recuperar esa tabla y sus datos
- Cada usuario de base de datos tiene su propia papelera de reciclaje a la que se mueven los objetos borrados y que se pueden recuperar de aquí con el comando FLASHBACK TABLE
- Este comando se puede utilizar para restaurar una tabla, una vista o un índice que se haya borrado por error
- La sintaxis es la siguiente:

```
FLASHBACK TABLE tablename TO BEFORE DROP;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK TABLE

- Por ejemplo, si borra la tabla EMPLOYEES por error, puede recuperarla simplemente emitiendo el comando:

```
FLASHBACK TABLE copy_employees TO BEFORE DROP;
```

- Como propietario de una tabla, puede emitir el comando de flashback y si la tabla que está restaurando tenía índices, estos también se restauran
- Es posible ver los objetos que se pueden restaurar consultando la vista del diccionario de datos USER_RECYCLEBIN

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK TABLE

- Se puede consultar la vista USER_RECYCLEBIN como todas las demás vistas del diccionario de datos:

```
SELECT original_name, operation, droptime
FROM user_recyclebin
```

ORIGINAL_NAME	OPERATION	DROPTIME
EMPLOYEES	DROP	2007-12-05:12.34.24
EMP_PK	DROP	2007-12-05:12.34.24

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK TABLE

- Una vez restaurada una tabla con el comando FLASHBACK TABLE, ya no está visible en la vista USER_RECYCLEBIN
- Cualquier índice que se borrara con la tabla original también se restaurará
- Puede que sea necesario (por motivos de seguridad) borrar una tabla completamente, omitiendo la papelera de reciclaje
- Esto se puede hacer agregando la palabra clave PURGE

```
DROP TABLE copy_employees PURGE;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

RENAME

- Para cambiar el nombre de una tabla, utilice la sentencia RENAME
- Esto solo lo puede hacer el propietario del objeto o el DBA
- Sintaxis:

```
RENAME old_name to new_name;
```

- Ejemplo:

```
RENAME my_cd_collection TO my_music;
```

- Veremos más adelante que podemos renombrar otros tipos de objetos como, por ejemplo, vistas, secuencias y sinónimos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TRUNCATE

- Al truncar una tabla, se eliminan todas las filas de una tabla y se libera el espacio de almacenamiento utilizado por dicha tabla
- Al utiliza la sentencia TRUNCATE TABLE:
  - No se puede realizar rollback de la eliminación de filas
  - Debe ser el propietario de la tabla o tener privilegios de sistema DROP ANY TABLE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

TRUNCATE

- Sintaxis:

```
TRUNCATE TABLE tablename;
```

- Con la sentencia DELETE también se eliminan todas las filas de una tabla, pero no se libera el espacio de almacenamiento
- TRUNCATE es más rápido que DELETE porque no genera información de rollback

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMENT ON TABLE

- Puede agregar un comentario de hasta 2.000 caracteres sobre una columna, tabla o vista mediante la sentencia COMMENT
- Sintaxis:

```
COMMENT ON TABLE tablename | COLUMN table.column
     IS 'place your comment here';
```

- Ejemplo:

```
COMMENT ON TABLE employees
     IS 'Western Region only';
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMENT ON TABLE

- Para ver los comentarios de la tabla en el diccionario de datos:

```
SELECT table_name, comments
FROM user_tab_comments;
```

TABLE_NAME	COMMENTS
EMPLOYEES	Western Region Only

- Si desea borrar un comentario realizado anteriormente en una tabla o columna, utilice la cadena vacía ('):

```
COMMENT ON TABLE employees IS ' ' ;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- Puede que detecte que los datos en una tabla se han cambiado, de algún modo, de forma incorrecta
- Afortunadamente, Oracle tiene una función que permite ver los datos de fila en momentos concretos en el tiempo, por lo que puede comparar diferentes versiones de una fila a lo largo del tiempo
- Esta función es muy útil
- Imagine, por ejemplo, que alguien por error ha realizado algún DML en una tabla y, a continuación, ejecuta COMMIT en dichos cambios
- Oracle Application Express confirma automáticamente, por lo que se cometen fácilmente errores

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- Puede utilizar la función FLASHBACK QUERY para examinar qué aspecto tenían las filas ANTES de que se aplicaran esos cambios
- Cuando Oracle cambia datos, siempre mantiene una copia del aspecto de los datos corregidos antes de que se realizara cualquier cambio
- Por lo que mantiene una copia del valor anterior de la columna para una actualización de columna, mantiene toda la fila para una supresión y no mantiene nada para una sentencia de inserción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- Estas copias antiguas se mantienen en un lugar especial denominado el tablespace UNDO
- Los usuarios pueden acceder a esta zona especial de la base de datos mediante una consulta de flashback
- Puede consultar las versiones anteriores de los datos utilizando la cláusula VERSIONS en una sentencia SELECT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- Por ejemplo:

```
SELECT employee_id,first_name ||' '|| last_name AS "NAME",
    versions_operation AS "OPERATION",
    versions_starttime AS "START_DATE",
    versions_endtime  AS "END_DATE", salary
FROM employees
  VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  employee_id = 1;
```

- El número SCN al que se hacía referencia en la consulta anterior significa que el número de cambio del sistema es una identificación precisa de tiempo en la base de datos
- Se trata de un número secuencial que incrementa y mantiene la propia base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- La mejor forma de demostrar FLASHBACK QUERY es con un ejemplo
- El contenido es el siguiente para employee_id 1 en la tabla de empleados

```
SELECT employee_id,first_name ||' '|| last_name AS "NAME",
    versions_operation AS "OPERATION",
    versions_starttime AS "START_DATE",
    versions_endtime  AS "END_DATE", salary
FROM employees
  VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  employee_id = 1;
```

no se han encontrado datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- A continuación, creamos el empleado:

```
INSERT INTO copy_employees
VALUES (1, 'Natacha', 'Hansen', 'NHANSEN', '4412312341234',
        '07-SEP-1998', 'AD_VP', 12000, null, 100, 90, NULL);
```

```
SELECT employee_id,first_name ||' '|| last_name AS "NAME",
    versions_operation AS "OPERATION",
    versions_starttime AS "START_DATE",
    versions_endtime  AS "END_DATE", salary
FROM employees
  VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  employee_id = 1;
```

EMPLOYEE_ID	NAME	OPERATION	START_DATE	END_DATE	SALARY
1	Natacha Hansen	I	07-SEP-1998 06.51.58 AM	-	12000

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- A continuación, puede actualizar la fila:

```
UPDATE copy_employees
SET salary = 1
WHERE employee_id = 1;
```

```
SELECT employee_id,first_name ||' '|| last_name AS "NAME",
    versions_operation AS "OPERATION",
    versions_starttime AS "START_DATE",
    versions_endtime  AS "END_DATE", salary
FROM employees
  VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  employee_id = 1;
```

EMPLOYEE_ID	NAME	OPERATION	START_DATE	END_DATE	SALARY
1	Natacha Hansen	U	07-SEP-1998 06.57.01 AM	-	1
1	Natacha Hansen	I	07-SEP-1998 06.51.58 AM	07-SEP-1998 06.57.01 AM	12000

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- A continuación, puede suprimir la fila:

```
DELETE from copy_employees
WHERE employee_id = 1;
```

```
SELECT employee_id,first_name ||' '|| last_name AS "NAME",
    versions_operation AS "OPERATION",
    versions_starttime AS "START_DATE",
    versions_endtime  AS "END_DATE", salary
FROM employees
  VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  employee_id = 1;
```

EMPLOYEE_ID	NAME	OPERATION	START_DATE	END_DATE	SALARY
1	Natacha Hansen	D	07-SEP-1998 07.00.10 AM	-	1
1	Natacha Hansen	U	07-SEP-1998 06.57.01 AM	07-SEP-1998 07.00.10 AM	1
1	Natacha Hansen	I	07-SEP-1998 06.51.58 AM	07-SEP-1998 06.57.01 AM	12000

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FLASHBACK QUERY

- El resultado de la última consulta de la diapositiva anterior solo está disponible si se utiliza Flashback query, es decir, la cláusula VERSIONS
- Si intenta realizar una búsqueda normal de employee_id = 1 tras la sentencia delete, recibirá el error normal, No Data Found

```
SELECT employee_id, salary
FROM copy_employees
WHERE employee_id = 1;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - ALTER TABLE
    - ADD
    - MODIFY
    - DROP
  - DROP TABLE
  - RENAME
  - TRUNCATE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - COMMENT ON TABLE
  - FLASHBACK TABLE
  - FLASHBACK QUERY
  - SET UNUSED

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, ha aprendido lo siguiente:
  - Explicar por qué es importante poder modificar una tabla
  - Explicar y proporcionar un ejemplo para cada una de las sentencias DDL (ALTER, DROP, RENAME y TRUNCATE) y el efecto que tiene cada una en las tablas y columnas
  - Construir una consulta y ejecutar los comandos ADD, MODIFY y DROP de ALTER TABLE
  - Explicar y ejecutar FLASHBACK QUERY en una tabla
  - Explicar y realizar operaciones FLASHBACK table

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, ha aprendido lo siguiente:
  - Seguimiento de los Cambios en los Datos a lo Largo de un Período de Tiempo
  - Explicar los motivos para utilizar TRUNCATE frente a DELETE en las tablas
  - Agregar un comentario a una tabla utilizando el comando COMMENT ON TABLE
  - Enumerar los cambios que se pueden y no se pueden realizar en una columna
  - Explicar cuándo y por qué la sentencia SET UNUSED es ventajosa

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

14-1

Introducción a las Restricciones; Restricciones NOT NULL y UNIQUE

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Definir el término "restricción" relacionado con el modelado de datos
  - Establecer cuándo es posible definir una restricción a nivel de columna y cuándo es posible a nivel de tabla
  - Definir por qué es importante proporcionar nombres significativos a las restricciones
  - Definir qué reglas de integridad de datos aplican las restricciones NOT NULL y UNIQUE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Escribir una sentencia CREATE TABLE que incluya las restricciones NOT NULL y UNIQUE en los niveles de tabla y columna
  - Explicar cómo crear restricciones en el momento de la creación de la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Si lo piensa, ¿cómo funcionaría la sociedad sin reglas?
- Es una regla parar en un semáforo en rojo
- ¿Sería seguro conducir sin esta regla?
- En el caso de las bases de datos, es una regla que no se puede introducir un valor de clave ajena sin primero introducir un valor de clave primaria
- ¿Qué cree que le sucedería a una base de datos si esta regla no se aplicara?
- Una base de datos solo es tan fiable como lo son los datos que tiene
- Las restricciones se utilizan para evitar la entrada de datos no válidos en las tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Tendría sentido tener valores de salario negativos, seis alumnos con el mismo identificador de alumno o dos tablas que ya no hacen referencia la una a la otra?
- Sin reglas, ¿cómo podría confía en la integridad de la base de datos?
- En las tres próximas lecciones, estudiará cómo crear las restricciones que fuerzan las "reglas"
- También aprenderá cómo gestionarlas y ver las definiciones de restricciones en el diccionario de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en General

- Por lo tanto, ¿qué es exactamente una restricción?
- Piense en las restricciones como reglas de la base de datos
- Todas las definiciones de restricciones se almacenan en el diccionario de datos
- Las restricciones evitan la supresión de una tabla si hay dependencias de otras tablas
- Las restricciones aplican reglas a los datos cuando se inserta, actualiza o suprime una fila de la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en General

- Las restricciones son importantes y asignarles un nombre también es importante
- Aunque puede asignar a una restricción el nombre "constraint_1" o "constraint_2", pronto le resultará difícil distinguirlas entre sí y acabará rehaciendo mucho trabajo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Restricciones

- Recuerde la sintaxis SQL para crear una tabla
- En la sentencia CREATE TABLE mostrada en la siguiente diapositiva, se define cada columna y su tipo de dato
- Utilice la sentencia CREATE TABLE para establecer restricciones para cada columna de la tabla
- Hay dos ubicaciones diferentes en la sentencia CREATE TABLE en los que puede especificar los detalles de la restricción:
  - En el nivel de columna situado junto al nombre y tipo de dato
  - En el nivel de tabla una vez que se muestran todos los nombres de columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Restricciones

```
CREATE TABLE clients
(client_number NUMBER(4),
 first_name        VARCHAR2(14),
 last_name        VARCHAR2(13));
```

- Tenga en cuenta que el nivel de columna simplemente se refiere al área de la sentencia CREATE TABLE en el que las columnas están definidas
- El nivel de tabla hace referencia a la última línea de la sentencia debajo de la lista de nombres de columna individuales

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en el Nivel de Columna

- Una restricción de nivel de columna hace referencia a una única columna
- Para establecer una restricción de nivel de columna, la restricción se debe definir en la sentencia CREATE TABLE como parte de la definición de columna
- Examine la siguiente sentencia SQL que establece una restricción de nivel de columna

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk
                                                PRIMARY KEY,
 first_name        VARCHAR2(14),
 last_name        VARCHAR2(13));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en el Nivel de Columna

- CREATE TABLE clients:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk
                                                PRIMARY KEY,
 first_name          VARCHAR2(14),
 last_name          VARCHAR2(13));
```

- El nombre de la restricción es clients_client_num_pk
- Se aplica la regla de negocio por la que client_number es la clave primaria de la tabla de clientes

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de Nomenclatura

- Todas las restricciones de la base de datos tienen un nombre. Cuando se crea una restricción, se puede asignar un nombre, como clients_client_num_pk, o no asignar ningún nombre, en cuyo caso el sistema asignará un nombre a la restricción, como, por ejemplo, SYS_C00585417
- Una convención de nomenclatura puede ser la combinación del nombre de tabla abreviado y un nombre de columna abreviado seguido de la abreviatura de la restricción: nombre-tabla_nombre-columna_tipo-restricción
- Si se utiliza la palabra reservada CONSTRAINT en la definición de CREATE TABLE, debe asignar un nombre a la restricción. Los nombres de restricción tienen un límite de 30 caracteres

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de Nomenclatura en el Nivel de Columna

- Es mejor que asigne un nombre a las restricciones porque los nombres generados por el sistema no son fáciles de reconocer
- Observe esta definición de tabla:

```
CREATE TABLE clients
(client_number   NUMBER(4),
 last_name       VARCHAR2(13),
 email           VARCHAR2(80));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de Nomenclatura en el Nivel de Columna

- Según nuestra convención de nomenclatura:
  - Una restricción de clave primaria en client_number tendría el nombre clients_client_number_pk
  - Una restricción not null en last_name tendría el nombre clients_last_name_nn
  - Una restricción única en una dirección de correo electrónico tendría el nombre clients_email_uk

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_cient_num_pk PRIMARY KEY,
 last_name     VARCHAR2(13) CONSTRAINT clients_last_name_nn NOT NULL,
 email         VARCHAR2(80) CONSTRAINT clients_emil_uk UNIQUE);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Restricción de Nomenclatura

- En este ejemplo se muestra una restricción a la que ha asignado nombre el usuario y una restricción a la que ha asignado nombre el sistema:

```
CREATE TABLE clients
 (client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
  last_name     VARCHAR2(13) NOT NULL,
  email         VARCHAR2(80));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Restricción de Nomenclatura

- Restricción a la que ha asignado nombre el sistema:

```
CREATE TABLE clients
 (client_number NUMBER(4) CONSTRAINT clients_client_num_pk PRIMARY KEY,
  last_name     VARCHAR2(13) NOT NULL,
  email         VARCHAR2(80));
```

- Se han creado dos restricciones:
  - Una restricción a la que el usuario ha asignado el nombre clients_client_num_pk, para aplicar la regla de que client_number es la clave primaria
  - Una restricción a la que el sistema ha asignado el nombre SYS_Cn (donde n es un entero único) para aplicar la regla de que last_names no puede ser nulo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en el Nivel de Tabla

- Las restricciones de nivel de tabla se muestran por separado de las definiciones de columna en la sentencia CREATE TABLE
- Las definiciones de restricciones de nivel de tabla se muestran después de que se hayan definido todas las columnas de la tabla
- En el ejemplo siguiente, la única restricción aparece en último lugar en la sentencia CREATE TABLE

```
CREATE TABLE clients (
 client_number NUMBER(6) NOT NULL,
 first_name    VARCHAR2(20),
 last_name     VARCHAR2(20),
 phone         VARCHAR2(20),
 email         VARCHAR2(10) NOT NULL,
CONSTRAINT clients_phone_email_uk UNIQUE (email,phone));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Reglas Básicas para Restricciones

- Las restricciones que hacen referencia a más de una columna se deben especificar en el nivel de tabla
- La restricción NOT NULL se puede especificar solo en el nivel de columna, no en el nivel de tabla
- Las restricciones UNIQUE, PRIMARY KEY, FOREIGN KEY y CHECK se pueden definir en el nivel de columna o de tabla
- Si se utiliza la palabra CONSTRAINT en la sentencia CREATE TABLE, debe asignar un nombre a la restricción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Examen de las Violaciones

```
CREATE TABLE clients(
        client_number  NUMBER(6),
        first_name     VARCHAR2(20),
        last_name      VARCHAR2(20),
        phone          VARCHAR2(20) CONSTRAINT phone_email_uk
                              UNIQUE(email,phone),
         email          VARCHAR2(10) CONSTRAINT NOT NULL,
     CONSTRAINT emailclients_email NOT NULL,
     CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Examen de las Violaciones

VIOLACIÓN DE CLAVE ÚNICA COMPUESTA
Las claves compuestas se deben definir en nivel de tabla

```
CREATE TABLE clients(
        client_number  NUMBER(6),
        first_name     VARCHAR2(20),
        last_name      VARCHAR2(20),
        phone          VARCHAR2(20) CONSTRAINT phone_email_uk
                              UNIQUE(email,phone),
         email         VARCHAR2(10) CONSTRAINT NOT NULL,
     CONSTRAINT emailclients_email NOT NULL,
     CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

VIOLACIÓN DE NOT NULL
Las restricciones NOT NULL solo se pueden definir en el nivel de columna

VIOLACIÓN DE NOMBRE
Cuando se utiliza el término CONSTRAINT, debe ir seguido de un nombre de restricción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cinco Tipos de Restricciones

- Existen cinco tipos de restricciones en una base de datos de Oracle
- Cada tipo aplica una regla diferente
- Los tipos son:
  - Restricciones NOT NULL
  - Restricciones UNIQUE
  - Restricciones PRIMARY KEY
  - Restricciones FOREIGN KEY
  - Restricciones CHECK

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cinco Tipos de Restricciones

- En el resto de esta lección, conocerá las restricciones NOT NULL y UNIQUE
- En la siguiente lección se le enseñarán los otros tres tipos de restricciones

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción NOT NULL

- Una columna definida con una restricción NOT NULL necesita que para cada fila introducida en la tabla, debe existir un valor para dicha columna
- Por ejemplo, si la columna de correo electrónico de una tabla de empleados se ha definido como NOT NULL, cada empleado introducido en la tabla DEBE tener un valor en la columna de correo electrónico
- Al definir columnas NOT NULL, es habitual utilizar el sufijo _nn en el nombre de restricción
- Por ejemplo, el nombre de restricción para la columna de correo electrónico NOT NULL de la tabla de empleados podría ser emp_email_nn

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción UNIQUE

- Una restricción UNIQUE necesita que todos los valores de una columna o juego de columnas (clave compuesta) sean únicos; es decir, que dos filas de una tabla no pueden tener valores duplicados
- Por ejemplo, puede ser importante para un negocio garantizar que no hay dos personas que tengan la misma dirección de correo electrónico
- La columna de correo electrónico se puede definir con una restricción UNIQUE
- La columna o juego de columnas que se define como UNIQUE se denomina una clave única

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción UNIQUE

- Si la combinación de dos o más columnas debe ser única para cada entrada, se dice que la restricción es una clave única compuesta
- La indicación de que todas las combinaciones de correo electrónico y apellido deben ser ÚNICAS es un ejemplo de clave única compuesta
- La palabra "clave" hace referencia a las columnas, no a los nombres de restricciones

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Restricción UNIQUE

- Si la columna de correo electrónico de la tabla se define con una restricción UNIQUE, ninguna otra entrada de cliente puede tener el mismo correo electrónico
- ¿Qué ocurre si dos clientes viven en el mismo domicilio y comparten una dirección de correo electrónico?

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net

```
INSERT INTO clients (client_number, first_name, Last_name, phone,
email)
   VALUES (7234, 'Lonny', 'Vigil', 4072220091, 'lbv@lbv.net');
```

```
ORA-00001: unique constraint
(USWA_SKHS_SQL01_T01.CLIENT_EMAIL_UK) violated
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Definición de Restricciones UNIQUE

- Al definir restricciones UNIQUE, es habitual utilizar el sufijo _uk en el nombre de restricción
- Por ejemplo, el nombre de restricción para la columna de correo electrónico UNIQUE de la tabla de empleados podría ser emp_email_uk
- Para definir una clave única compuesta, debe definir la restricción en el nivel de tabla en lugar del nivel de columna
- Un ejemplo de nombre de restricción de clave única es:

```
CONSTRAINT clients_phone_email_uk UNIQUE(email,phone)
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Clave Única Compuesta

- Las restricciones UNIQUE permiten la entrada de valores nulos a menos que la columna también tenga definida una restricción NOT NULL
- Un valor nulo en una columna (o en todas las columnas de una clave única compuesta) cumple siempre una restricción UNIQUE porque los valores nulos no se consideran igual a nada

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net

Esta combinación de columnas debe ser **ÚNICA**

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Clave Única Compuesta

- Para cumplir una restricción que designa una clave única compuesta, dos filas de la tabla no pueden tener la misma combinación de valores en las columnas de clave
- Además, cualquier fila que contenga valores nulos en todas las columnas de clave automáticamente cumple la restricción

CLIENT_NUMBER	FIRST_NAME	LAST_NAME	PHONE	EMAIL
5922	Hiram	Peters	3715832249	hpeters@yahoo.com
5857	Serena	Jones	7035335900	serena.jones@jones.com
6133	Lauren	Vigil	4072220090	lbv@lbv.net
7234	Lonny	Vigil	4072220091	lbv@lbv.net

Esta combinación de columnas debe ser **ÚNICA**

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones Creadas en la Creación de la Tabla

- Al agregar una restricción NOT NULL como parte de la sentencia de creación de una tabla, la base de datos Oracle creará una restricción Check en la base de datos para aplicar un valor en la columna NOT NULL
- La creación de esta restricción puede ser casi invisible para el usuario al crear la tabla: Oracle solo lo hace
- Al final de la sentencia de creación de tabla, se muestra el mensaje "Table created", pero no se proporcionan detalles sobre el número o tipos de restricciones que también se han creado

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Restricción CHECK
  - ¿La restricción
  - Restricciones de nivel de columna
  - FOREIGN KEY
  - Restricciones NOT NULL
  - PRIMARY KEY

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Restricción de nivel de tabla
  - Restricciones UNIQUE
  - UNIQUE KEY

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Definir el término "restricción" relacionado con el modelado de datos
  - Establecer cuándo es posible definir una restricción a nivel de columna y cuándo es posible a nivel de tabla
  - Definir por qué es importante proporcionar nombres significativos a las restricciones
  - Definir qué reglas de integridad de datos aplican las restricciones NOT NULL y UNIQUE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Escribir una sentencia CREATE TABLE que incluya las restricciones NOT NULL y UNIQUE en los niveles de tabla y columna
  - Explicar cómo crear restricciones en el momento de la creación de la tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

14-2

Restricciones PRIMARY KEY, FOREIGN KEY y CHECK

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Definir y dar un ejemplo de una restricción PRIMARY KEY, FOREIGN KEY y CHECK
  - Explicar el objetivo de definir las restricciones PRIMARY KEY, FOREIGN KEY y CHECK
  - Demostrar la creación de las restricciones a nivel de columna y a nivel de tabla en una sentencia CREATE TABLE
  - Evaluar un problema de negocio que necesita la adición de una restricción PRIMARY KEY y FOREIGN KEY y escribir el código para ejecutar el cambio

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Como se ha explicado en la última sección, las restricciones se utilizan para evitar la entrada de datos no válidos en las tablas de la base de datos
- ¿Qué sucedería si, a escondidas o solo por un error negligente, su identificación única personal se proporcionara a otra persona?
- ¿Qué sucedería si mañana en la escuela se atribuyeran a otra persona sus clases de graduación o esta pudiera almorzar con su número de tarjeta para el almuerzo?
- Garantizar la integridad de datos es de lo que tratan las. Después de todo, cada uno es único

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones PRIMARY KEY

- Una restricción PRIMARY KEY es una regla de que los valores de una columna o una combinación de columnas deben identificar de forma única cada fila de una tabla
- No puede aparece ningún valor de clave primaria en más de una fila de la tabla
- Para cumplir una restricción PRIMARY KEY, las dos condiciones siguientes deben ser verdaderas:
  - Ninguna columna que forme parte de la clave primaria puede contener un valor nulo
  - Una tabla solo puede tener una clave primaria

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones PRIMARY KEY

- Las restricciones PRIMARY KEY se pueden definir en nivel de columna o de tabla
- Sin embargo, si se crea una CLAVE PRIMARIA compuesta, se debe definir en el nivel de tabla
- Al definir columnas PRIMARY KEY, es una buena práctica utilizar el sufijo _pk en el nombre de restricción
- Por ejemplo, el nombre de restricción para la columna PRIMARY KEY denominada client_number de la tabla llamada CLIENTS podría ser clients_client_num_pk

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones PRIMARY KEY

- En una sentencia CREATE TABLE, la sintaxis de restricción PRIMARY KEY en el nivel de columna se indica de la siguiente forma:

```
CREATE TABLE clients
(client_number NUMBER(4) CONSTRAINT clients_client_num_pk
                           PRIMARY KEY,
first_name VARCHAR2(14),
last_name VARCHAR2(13));
```

- Tenga en cuenta que el nivel de columna simplemente se refiere al área de la sentencia CREATE TABLE en el que las columnas están definidas
- El nivel de tabla hace referencia a la última línea de la sentencia debajo de la lista de nombres de columna individuales

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones PRIMARY KEY

- Para crear la restricción PRIMARY KEY en el nivel de tabla, la sintaxis es:

```
CREATE TABLE clients
(client_number NUMBER(4),
 first_name VARCHAR2(14),
 last_name VARCHAR2(13),
CONSTRAINT clients_client_num_pk PRIMARY KEY
(client_number));
```

- Tenga en cuenta que el nombre de la columna PRIMARY KEY sigue el tipo de restricción y está entre paréntesis

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones PRIMARY KEY

- Para definir una CLAVE PRIMARIA compuesta, debe definir la restricción en el nivel de tabla en lugar del nivel de columna
- A continuación se muestra un ejemplo de restricción de clave primaria compuesta:

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
CONSTRAINT copy_jhist_id_st_date_pk PRIMARY KEY(employee_id,
start_date));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

FOREIGN KEY (INTEGRIDAD REFERENCIAL) Restricciones

- Las restricciones FOREIGN KEY también se denominan restricciones de "integridad referencial"
- Las restricciones Foreign Key designan una columna o una combinación de columnas como una clave ajena
- Un clave ajena enlaza a la clave primaria (o una clave única) en otra tabla y este enlace es la base de la relación entre las tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visualización de una Clave Ajena

- La tabla que contiene la clave ajena se denomina tabla "secundaria" y la tabla que contiene la clave de referencia se denomina tabla "principal"

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visualización de una Clave Ajena

- En las tablas mostradas, la clave primaria de la tabla DEPARTMENTS, department_id, también aparece en la tabla EMPLOYEES como una columna de clave ajena

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción de Integridad Referencial

- Para cumplir una restricción de integridad referencial, el valor de clave ajena debe coincidir con un valor existente de la tabla principal o ser un valor NULL

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción de Integridad Referencial

- Puede existir un valor de clave primaria sin un valor de clave ajena correspondiente; sin embargo, una clave ajena debe tener una clave primaria correspondiente

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Regla de Restricción de Integridad Referencial

- La regla es: antes de definir una restricción de integridad referencial en la tabla secundaria, ya debe estar definida la restricción UNIQUE o PRIMARY a la que se hace referencia

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Regla de Restricción de Integridad Referencial

- Es decir, primero debe tener una clave primaria principal definida para que pueda crear una clave ajena en una tabla secundaria

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricción FOREIGN KEY

- Para definir una restricción FOREIGN KEY, es una buena práctica utilizar el sufijo _fk en el nombre de restricción
- Por ejemplo, el nombre de restricción para la columna de CLAVE AJENA department_id de la tabla de empleados podría ser emps_dept_id_fk

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de Restricción FOREIGN KEY

- La sintaxis para definir una restricción FOREIGN KEY necesita una referencia a la tabla y columna de la tabla principal
- Una restricción FOREIGN KEY en una sentencia CREATE TABLE se puede definir de la siguiente forma
- Ejemplo de sintaxis de nivel de columna:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0) CONSTRAINT c_emps_dept_id_fk
                           REFERENCES departments(department_id),
 email VARCHAR2(25));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de Restricción FOREIGN KEY

- La sintaxis para definir una restricción FOREIGN KEY necesita una referencia a la tabla y columna de la tabla principal
- Una restricción FOREIGN KEY en una sentencia CREATE TABLE se puede definir de la siguiente forma
- Ejemplo de sintaxis de nivel de tabla:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT c_emps_dept_id_fk FOREIGN KEY (department_id)
                              REFERENCES departments(department_id));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ON DELETE CASCADE: Mantenimiento de Integridad Referencial

- El uso de la opción ON DELETE CASCADE al definir una clave ajena permite la supresión de las filas dependientes en la tabla secundaria cuando se suprime una fila de la tabla principal
- Si la clave ajena no tiene una opción ON DELETE CASCADE, las filas a las que se hace referencia de la tabla principal no se pueden suprimir
- Es decir, la restricción FOREIGN KEY de la tabla secundaria incluye el permiso ON DELETE CASCADE que permite a su principal suprimir las filas a las que se hace referencia

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ON DELETE CASCADE: Mantenimiento de Integridad Referencial

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ON DELETE CASCADE

- Si la columna department_id de empleados se ha creado con la opción ON DELETE CASCADE especificada, se ejecutará la sentencia DELETE emitida en la tabla de departamentos
- Si no se ha especificado la opción ON DELETE CASCADE al crear FOREIGN KEY, fallará el intento de supresión de un departamento de la tabla de departamentos que tiene entradas en la tabla de empleados

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de ON DELETE CASCADE

- Tabla creada sin ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
       REFERENCES copy_departments(department_id));
```

- Fallará un intento de supresión de department_id 110 de la tabla de departamentos ya que hay filas dependientes en la tabla de empleados

```
ORA-02292: integrity constraint (US_A009EMEA815_PLSQL_T01.CDEPT_DEPT_ID_FK)
violated - child record found
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de ON DELETE CASCADE

- Tabla creada con ON DELETE CASCADE:

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
    REFERENCES copy_departments(department_id) ON DELETE CASCADE);
```

- El intento de supresión de department_id 110 de la tabla de departamentos se realizará correctamente y las filas dependientes en la tabla de empleados también se suprimirán
- 1 fila suprimida

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ON DELETE SET NULL

- En lugar de que se supriman las filas de la tabla secundaria al utilizar una opción ON DELETE CASCADE, las filas secundarias se pueden rellenar con valores nulos mediante la opción ON DELETE SET NULL

```
CREATE TABLE copy_employees
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
 first_name VARCHAR2(20),
 last_name VARCHAR2(25),
 department_id NUMBER(4,0),
 email VARCHAR2(25),
 CONSTRAINT cdept_dept_id_fk FOREIGN KEY (department_id)
    REFERENCES copy_departments(department_id) ON DELETE SET NULL);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ON DELETE SET NULL

- Esto podría resultar útil cuando el valor de la tabla principal se va a cambiar a un nuevo número como, por ejemplo, al convertir números de inventario a números de código de barras
- No desea suprimir las filas de la tabla secundaria
- Cuando se introduzcan los nuevos números de código de barras en la tabla principal, se podrán insertar en la tabla secundaria sin tener que volver a crear totalmente cada una de las filas de la tabla secundaria

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones CHECK

- La restricción CHECK define explícitamente una condición que se debe cumplir
- Para cumplir la restricción, cada una de las filas de la tabla debe hacer que condición sea True o desconocida (debido a un valor nulo)
- La condición de una restricción de CHECK puede hacer referencia a cualquier columna de la tabla especificada, pero no a columnas de otras tablas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Restricción CHECK

- Esta restricción CHECK garantiza que un valor introducido para end_date sea posterior a start_date

```
CREATE TABLE copy_job_history
(employee_id NUMBER(6,0),
 start_date DATE,
 end_date DATE,
 job_id VARCHAR2(10),
 department_id NUMBER(4,0),
 CONSTRAINT cjhist_emp_id_st_date_pk
           PRIMARY KEY(employee_id, start_date),
 CONSTRAINT cjhist_end_ck CHECK (end_date > start_date));
```

- Como esta RESTRICCIÓN CHECK hace referencia a dos columnas de la tabla, se DEBE definir en el nivel de tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Condiciones de la Restricción CHECK

- Una restricción CHECK solo debe estar en la fila en la que se define la restricción
- Una restricción CHECK no se puede utilizar en consultas que hacen referencia a valores de otras filas
- La restricción CHECK no puede contener llamadas a las funciones SYSDATE, UID, USER o USERENV
- La sentencia CHECK(SYSDATE >'05-May-1999') no está permitida

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Condiciones de la Restricción CHECK

- La restricción USER no puede utilizar las pseudocolumnas CURRVAL, NEXTVAL, LEVEL o ROWNUM
- La sentencia CHECK(NEXTVAL > 0) no está permitida
- Una sola columna puede tener varias restricciones CHECK que hagan referencia a la columna en su definición
- No hay ningún límite en cuanto al número de restricciones CHECK que puede definir en una columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de Restricción CHECK

- Las restricciones CHECK se pueden definir en el nivel de tabla o de columna.
- La sintaxis para definir una restricción CHECK es la siguiente:
  - Sintaxis de nivel de columna:

```
salary NUMBER(8,2) CONSTRAINT employees_min_sal_ck CHECK
                                        (salary > 0)
```

  - Sintaxis de nivel de tabla:

```
CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Restricción CHECK
  - Restricción FOREIGN KEY
  - REFERENCES
  - NOT NULL
  - ON DELETE CASCADE
  - ON DELETE SET NULL
  - Restricción PRIMARY KEY

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Definir y dar un ejemplo de una restricción PRIMARY KEY, FOREIGN KEY y CHECK
  - Explicar el objetivo de definir las restricciones PRIMARY KEY, FOREIGN KEY y CHECK
  - Demostrar la creación de las restricciones a nivel de columna y a nivel de tabla en una sentencia CREATE TABLE
  - Evaluar un problema de negocio que necesita la adición de una
  - restricción PRIMARY KEY y
  - FOREIGN KEY y escribir el
  - código para ejecutar el cambio

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

14-3

Gestión de Restricciones

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Enumerar cuatro funciones diferentes que puede realizar la sentencia ALTER en las restricciones
  - Escribir sentencias ALTER TABLE para agregar, borrar, desactivar y activar restricciones
  - Nombrar una función de negocio que necesitaría un DBA para borrar, activar y/o desactivar una restricción o utilizar la sintaxis CASCADE
  - Consultar USER_CONSTRAINTS en el diccionario de datos e interpretar
  - la información devuelta

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Supondría alguna diferencia que un nuevo número de identificación de alumno se introdujera en la base de datos de la escuela cuando no hubiera ningún alumno real inscrito?
- ¿Es probable que una compañía de tarjeta de crédito emita el mismo número de tarjeta de crédito a más de una cuenta o que una empresa contrate a un empleado para un departamento que no existe?
- ¿Qué predice que sucedería si una empresa no pudiera confiar en la fiabilidad de la información de su base de datos?

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Un sistema de base tiene que poder aplicar reglas de negocio y, al mismo tiempo, evitar la adición, la modificación o la supresión de datos que pueda dar como resultado una violación de la integridad referencial de la base de datos
- En esta sección, aprenderá a realizar cambios en las restricciones de tabla para que la integridad referencial y, a su vez, la fiabilidad de la base de datos se mantengan cuando sea necesario cambiar los datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Gestión de Restricciones

- La sentencia ALTER TABLE se utiliza para realizar cambios en las restricciones de tablas existentes
- Estos cambios pueden incluir agregar o borrar restricciones, activar o desactivar restricciones, así como agregar una restricción NOT NULL a una columna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Gestión de Restricciones

- Las directrices para realizar cambios en las restricciones son:
  - Puede agregar, borrar, activar o desactivar una restricción, pero no puede modificar su estructura
  - Puede agregar una restricción NOT NULL a una columna existente mediante la cláusula MODIFY de la sentencia ALTER TABLE
  - MODIFY se utiliza porque NOT NULL es un cambio de nivel de columna
  - Solo puede definir una restricción NOT NULL si la tabla está vacía o si la columna tiene un valor para cada fila

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sentencia ALTER

- La sentencia ALTER necesita lo siguiente:
  - nombre de la tabla
  - nombre de la restricción
  - tipo de restricción
  - nombre de la columna a la que afecta la restricción
- En el ejemplo de código que se muestra a continuación, con la tabla de empleados, la restricción de clave principal se puede haber agregado después de que se creara originalmente la tabla

```
ALTER TABLE employees
ADD CONSTRAINT  emp_id_pk PRIMARY KEY (employee_id);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Adición de Restricciones

- Para agregar una restricción a una tabla existente, utilice la siguiente sintaxis SQL:

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name] type of constraint
                                  (column_name);
```

- Si la restricción es una restricción FOREIGN KEY, la palabra clave REFERENCES se debe incluir en la sentencia
- Sintaxis:

```
ALTER TABLE tablename
ADD CONSTRAINT constraint_name FOREIGN KEY(column_name)
              REFERENCES tablename(column_name);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Adición de Restricciones

- Considere la base de datos de empleados
- La clave primaria de la tabla DEPARTMENTS se introduce en la tabla EMPLOYEES como una clave ajena

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Adición de Restricciones

- En el siguiente ejemplo se muestra la sintaxis para agregar esta clave ajena a la tabla EMPLOYEES:

```
ALTER TABLE employees
ADD CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
   REFERENCES departments (department_id) ON DELETE CASCADE;
```

DEPARTMENTS: principal

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE: secundaria

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Condiciones para la Adición de Restricciones

- Si la restricción es una restricción NOT NULL, la sentencia ALTER TABLE utiliza MODIFY en lugar de ADD
- Las restricciones NOT NULL solo se pueden agregar si la tabla está vacía o si la columna tiene un valor para cada fila:

```
ALTER TABLE table_name
MODIFY (column_name CONSTRAINT constraint_name NOT NULL);
```

```
ALTER TABLE employees
MODIFY (email CONSTRAINT emp_email_nn NOT NULL);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿Por Qué Activar y Desactivar Restricciones?

- Para aplicar las reglas definidas por restricciones de integridad, las restricciones deben estar siempre activadas
- En determinadas situaciones, se recomienda desactivar temporalmente las restricciones de integridad de una tabla por motivos de rendimiento, por ejemplo:
  - Al cargar grandes cantidades de datos en una tabla
  - Al ejecutar operaciones por lotes que realizan cambios masivos en una tabla (como puede ser cambiar el número de empleado de todas las personas agregando 1.000 al número existente)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Borrado de Restricciones

- Para borrar una restricción, debe saber el nombre de la restricción
- Si no lo sabe, puede buscar el nombre de la restricción en USER_CONSTRAINTS y USER_CONS_COLUMNS en el diccionario de datos
- La opción CASCADE de la cláusula DROP provoca que se borren también las restricciones dependientes
- Tenga en cuenta que al borrar una restricción de integridad, Oracle Server ya no aplica esa restricción y deja de estar disponible en el diccionario de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Borrado de Restricciones

- No se borra ninguna fila ni ningún dato en ninguna de las tablas afectadas al borrar una restricción

```
ALTER TABLE table_name
DROP CONSTRAINT  name [CASCADE}
```

```
ALTER TABLE copy_departments
DROP CONSTRAINT  c_dept_dept_id_pk CASCADE;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Desactivación de Restricciones

- Por defecto, siempre que una restricción de integridad está definida en una sentencia CREATE o ALTER TABLE, Oracle activa (aplica) automáticamente la restricción, a menos que se cree específicamente con un estado desactivado con la cláusula DISABLE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Desactivación de Restricciones

- Puede desactivar una restricción sin borrarla o volver a crearla mediante DISABLE de la opción ALTER TABLE
- DISABLE permite datos entrantes, tanto si se ajustan a la restricción como si no
- Esta función permite agregar datos una tabla secundaria sin tener los valores correspondientes en la tabla principal
- DISABLE simplemente desactiva la restricción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de la Cláusula DISABLE

- Puede utilizar la cláusula DISABLE tanto en la sentencia ALTER TABLE como en la sentencia CREATE TABLE

```
CREATE TABLE copy_employees
( employee_id NUMBER(6,0) PRIMARY KEY DISABLE,
   ...
   ...);

ALTER TABLE copy_employees
DISABLE CONSTRAINT c_emp_dept_id_fk;
```

- La desactivación de una restricción UNIQUE o PRIMARY KEY elimina el índice único

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de la Cláusula CASCADE

- La cláusula CASCADE desactiva las restricciones de integridad dependientes. Si la restricción se activa posteriormente, las restricciones dependientes no se activan automáticamente
- Sintaxis y ejemplo:

```
ALTER TABLE table_name
DISABLE CONSTRAINT constraint_name [CASCADE];
```

```
ALTER TABLE copy_departments
DISABLE CONSTRAINT c_dept_dept_id_pk CASCADE;;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Activación de Restricciones

- Para activar una restricción de integridad actualmente desactivada, utilice la cláusula ENABLE en la sentencia ALTER TABLE
- ENABLE garantiza que todos los datos entrantes se ajustan a la restricción
- Sintaxis y ejemplo:

```
ALTER TABLE table_name
ENABLE CONSTRAINT constraint_name;
```

```
ALTER TABLE copy_departments
ENABLE CONSTRAINT c_dept_dept_id_pk;
```

- Puede utilizar la cláusula ENABLE tanto en la sentencia CREATE TABLE como en la sentencia ALTER TABLE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consideraciones sobre la Activación de Restricciones

- Si activa una restricción, se aplica a todos los datos de la tabla
- Todos los datos de la tabla deben cumplir la restricción
- Si activa una CLAVE UNIQUE o una restricción PRIMARY KEY, se crea un índice UNIQUE o PRIMARY KEY automáticamente
- La activación de una restricción PRIMARY KEY desactivada con la opción CASCADE no activa ninguna clave ajena dependiente de la clave primaria
- ENABLE vuelve a activar la restricción después de desactivarla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en Cascada

- Las restricciones de integridad referencial en cascada definen las acciones que lleva a cabo el servidor de base de datos cuando un usuario intenta suprimir o actualizar una clave a la que apuntan las claves ajenas existentes
- La cláusula CASCADE CONSTRAINTS se utiliza junto con la cláusula DROP COLUMN
- Borra todas las restricciones de integridad referencial que hacen referencia a las claves primarias y únicas definidas en las columnas borradas
- Borra también todas las restricciones de varias columnas definidas en las columnas borradas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones en Cascada

- Si una sentencia ALTER TABLE no incluye la opción CASCADE CONSTRAINTS, cualquier intento de borrar una restricción de clave primaria o varias columnas fallará
- Recuerde que no puede suprimir un valor principal si existen valores secundarios en otras tablas

```
ALTER TABLE table_name
DROP(column name(s)) CASCADE CONSTRAINTS;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cuando No Es Necesario CASCADE

- Si las columnas a las que hacen referencia las restricciones definidas en las columnas borradas también se borran, CASCADE CONSTRAINTS no es necesario
- Por ejemplo, si ninguna restricción referencial de otras tablas hace referencia a la columna PK, es válido ejecutar la siguiente sentencia sin la cláusula CASCADE CONSTRAINTS:

```
ALTER TABLE tablename DROP
(pk_column_name(s));
```

- Sin embargo, si las columnas de otras tablas o las columnas que quedan en la tabla de destino hacen referencia a las restricciones, se debe especificar CASCADE CONSTRAINTS

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visualización de Restricciones

- Después de crear una tabla, puede confirmar su existencia emitiendo un comando DESCRIBE
- La única restricción que se puede verificar con el comando DESCRIBE es la restricción NOT NULL
- La restricción NOT NULL también aparece en el diccionario de datos como una restricción CHECK

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Visualización de Restricciones

- Para ver todas las restricciones de la tabla, consulte la tabla USER_CONSTRAINTS

```
SELECT constraint_name, table_name, constraint_type, status
FROM USER_CONSTRAINTS
WHERE table_name ='COPY_EMPLOYEES';
```

CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE	STATUS
COPY_EMP_PK	COPY_EMPLOYEES	P	ENABLED
CDEPT_DEPT_ID_FK	COPY_EMPLOYEES	R	ENABLED

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consulta USER_CONSTRAINTS

- Los tipos de restricción que aparecen en el diccionario de datos son:
  - P – PRIMARY KEY; R – REFERENCES (clave ajena);
  - C – restricción CHECK (incluida NOT NULL);
  - U – UNIQUE

CONSTRAINT_NAME	TABLE_NAME	CONSTRAINT_TYPE	STATUS
COPY_EMP_PK	COPY_EMPLOYEES	P	ENABLED
CDEPT_DEPT_ID_FK	COPY_EMPLOYEES	R	ENABLED

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - ALTER TABLE
  - Cláusula CASCADE
  - Cláusula CASCADE CONSTRAINT
  - DISABLE CONSTRAINT
  - DROP COLUMN
  - DROP CONSTRAINT
  - ENABLE CONSTRAINT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Enumerar cuatro funciones diferentes que puede realizar la sentencia ALTER en las restricciones
  - Escribir sentencias ALTER TABLE para agregar, borrar, desactivar y activar restricciones
  - Nombrar una función de negocio que necesitaría un DBA para borrar, activar y/o desactivar una restricción o utilizar la sintaxis CASCADE
  - Consultar USER_CONSTRAINTS en el diccionario de datos e interpretar la información devuelta

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

15-1

Creación de Vistas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

ORACLE Academy

DP 15-1
Creación de Vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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."

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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]];
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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
...	...	...	...

ORACLE Academy

DP 15-1
Creación de Vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

15

Una vez que el administrador de base de datos crea una VISTA, el usuario simplemente consulta la VISTA como si se tratase de una tabla.

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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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';
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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';
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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;
```

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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
...	...	...	...

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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
...	...

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

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, capitol
   FROM wf_countries
   WHERE location LIKE '%Europe';
```

ORACLE Academy

DP 15-1
Creación de Vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Alias
  - Vista compleja
  - CREATE VIEW
  - FORCE
  - NOFORCE
  - REPLACE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Vista simple
  - Subconsulta
  - Ver
  - VIEW_NAME

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - 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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Crear una vista compleja que contenga funciones de grupo para visualizar valores de dos tablas
  - Recuperación de Datos de una Vista

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

15-2

Operaciones DML y Vistas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

En esta lección se abordan los siguientes objetivos:

- Escribir y ejecutar una consulta que realice operaciones DML en una vista simple
- Nombrar las condiciones que restringen la capacidad de modificar una vista mediante las operaciones DML
- Escribir y ejecutar una consulta utilizando la cláusula WITH CHECK OPTION

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

En esta lección se abordan los siguientes objetivos:

- Explicar el uso de WITH CHECK OPTION que se aplica a las restricciones de integridad y la validación de datos
- Aplicar la opción WITH READ ONLY a una vista para restringir las operaciones DML

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Como ha aprendido en la última lección, las vistas simplifican el acceso de usuario a datos incluidos en una o más tablas de la base de datos
- Sin embargo, las vistas también permiten a los usuarios realizar cambios en las tablas subyacentes
- Como DBA y persona cuyo trabajo es mantener la integridad de la base de datos, puede que desee poner restricciones en determinadas vistas de datos
- En esta lección, aprenderá a permitir el acceso a datos y al mismo tiempo garantizar la seguridad de los datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sentencias DML y Vistas

- Las operaciones DML INSERT, UPDATE y DELETE se pueden realizar en vistas simples
- Estas operaciones se pueden utilizar para cambiar los datos en las tablas base subyacentes
- Si crea una vista que permita a los usuarios ver información restringida mediante la cláusula WHERE, los usuarios seguirán pudiendo realizar operaciones DML en todas las columnas de la vista

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sentencias DML y Vistas

- Por ejemplo, la vista que se muestra a la derecha se ha creado para que los jefes del departamento 50 la base de datos de empleados
- La intención de esta vista es permitir a los jefes del departamento 50 ver la información sobre sus empleados

```
CREATE VIEW view_dept50
AS SELECT department_id,
employee_id,first_name, last_name,
salary
 FROM copy_employees
 WHERE department_id = 50;
```

```
SELECT * FROM
view_dept50;
```

DEPARTMENT_ID	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
50	124	Kevin	Mourgos	5800
50	141	Trenna	Rajs	3500
50	142	Curtis	Davies	3100
50	143	Randall	Matos	2600
50	144	Peter	Vargas	2500

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Control de Vistas

- Mediante la vista como se ha indicado, es posible INSERTAR, ACTUALIZAR y SUPRIMIR información de todas las filas de la vista, incluso aunque el resultado sea que una fila ya no forme parte de la vista
- Puede que esto no sea lo que deseaba el DBA cuando se creó la vista
- Para controlar el acceso a los datos, se pueden agregar dos opciones a la sentencia CREATE VIEW:
  - WITH CHECK OPTION
  - WITH READ ONLY

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas con la Opción CHECK

- La vista se define WITH CHECK OPTION

```
CREATE VIEW view_dept50
AS SELECT department_id, employee_id, first_name, last_name,
salary
   FROM copy_employees
   WHERE department_id = 50;
```

- Mediante la vista, se ha cambiado el departamento de employee_id 124 a dept_id 90

```
UPDATE view_dept50
SET department_id = 90
WHERE employee_id = 124;
```

```
1 row(s) updated.
```

- La actualización se realiza correctamente, aunque este empleado ahora no forma parte de la vista

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas con la Opción CHECK

- WITH CHECK OPTION garantiza que las operaciones DML realizadas en la vista se mantengan en el dominio de la vista
- Cualquier intento de cambiar el número de departamento de una fila de la vista fallará porque viola la restricción WITH CHECK OPTION
- Observe en el ejemplo siguiente que a la restricción WITH CHECK OPTION se le asignó el nombre view_dept50_check

```
CREATE OR REPLACE VIEW view_dept50
AS SELECT department_id, employee_id, first_name, last_name, salary
   FROM employees
   WHERE department_id = 50
WITH CHECK OPTION CONSTRAINT view_dept50_check;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas con la Opción CHECK

- Ahora, si intenta modificar una fila en la vista que la sacará del dominio de la vista, se devuelve un error

```
UPDATE view_dept50
SET department_id = 90
WHERE employee_id = 124;
```

```
ORA-01402: view WITH CHECK OPTION where-clause violation
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas con READ ONLY

- La opción WITH READ ONLY garantiza que no se produce ninguna operación DML en la vista
- Cualquier intento de ejecutar una sentencia INSERT, UPDATE o DELETE producirá un error de Oracle Server

```
CREATE OR REPLACE VIEW view_dept50
AS SELECT department_id, employee_id, first_name, last_name,
salary
 FROM employees
 WHERE department_id = 50
WITH READ ONLY;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de DML

- Las vistas simples y complejos difieren en su capacidad para permitir operaciones DML en una vista
- Para las vistas simples, las operaciones DML se pueden realizar en la vista
- Para las vistas complejas, las operaciones DML no siempre están permitidas
- Las tres reglas siguientes se deben tener en cuenta al realizar operaciones DML en las vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de DML

- No puede eliminar una fila desde una tabla base subyacente si la vista contiene algo de lo siguiente:
  - Funciones de grupo
  - Una cláusula GROUP BY
  - La palabra clave DISTINCT
  - La palabra clave ROWNUM de pseudocolumna

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de DML

- No puede modificar datos de una vista si la vista contiene:
  - Funciones de grupo
  - Una cláusula GROUP BY
  - La palabra clave DISTINCT
  - La palabra clave ROWNUM de pseudocolumna
  - Columnas definidas por expresiones

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones de DML

- No puede agregar datos en una vista si esta:
  - incluye funciones de grupo
  - incluye una cláusula GROUP BY
  - incluye la palabra clave DISTINCT
  - incluye palabra clave ROWNUM de pseudocolumna
  - incluye columnas definidas por expresiones
  - no incluye columnas NOT NULL en las tablas base

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- WITH CHECK OPTION
- WITH READ ONLY

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Escribir y ejecutar una consulta que realice operaciones DML en una vista simple
- Nombrar las condiciones que restringen la capacidad de modificar una vista mediante las operaciones DML
- Escribir y ejecutar una consulta utilizando la cláusula WITH CHECK OPTION

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Explicar el uso de WITH CHECK OPTION que se aplica a las restricciones de integridad y la validación de datos
- Aplicar la opción WITH READ ONLY a una vista para restringir las operaciones DML

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

15-3

Gestión de Vistas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Crear y ejecutar una sentencia SQL que elimina una vista
  - Crear y ejecutar una consulta mediante una vista en línea
  - Crear y ejecutar una consulta para el análisis de N principales

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Aprender a crear y sustituir vistas no sería completo a menos que también supiera cómo eliminarlas
- Las vistas se crean con fines específicos
- Cuando la vista ya no es necesaria o se debe modificar, existen medios para realizar los cambios necesarios
- Si un empleado que ha tenido acceso a información financiera deja la compañía, es probable que no desee que su vista siga estando accesible
- En esta lección, aprenderá a suprimir una vista, crear una vista en línea y construir una sentencia SELECT para producir una lista ordenada de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Supresión de una Vista

- Puesto que una vista no contiene datos propios, su eliminación no afecta a los datos de las tablas subyacentes
- Si la vista se ha utilizado para INSERTAR, ACTUALIZAR o SUPRIMIR datos en el pasado, esos cambios en las tablas base se mantienen
- La supresión de una vista simplemente elimina la definición de vista de la base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Supresión de una Vista

- Recuerde que las vistas se almacenan como una sentencia SELECT en el diccionario de datos
- Solo el creador o usuarios con el privilegio DROP ANY VIEW pueden eliminar una vista
- La sintaxis SQL para eliminar una vista es:

```
DROP VIEW viewname;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas en Línea

- Las vistas en línea también se denominan subconsultas en la cláusula FROM
- Inserte una subconsulta en la cláusula FROM como si la subconsulta fuera un nombre de tabla
- Las vistas en línea se utilizan normalmente para simplificar las complejas consultas mediante la eliminación de operaciones de unión y la condensación de varias consultas en una sola

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas en Línea

- Como se muestra en el siguiente ejemplo, la cláusula FROM contiene una sentencia SELECT que recupera datos como cualquier sentencia SELECT
- A los datos devueltos por la subconsulta se les asigna un alias (d), que, a continuación, se utiliza junto con la consulta principal para devolver columnas seleccionadas de ambos orígenes de consulta

```
SELECT e.last_name, e.salary, e.department_id, d.maxsal
FROM employees e,
       (SELECT department_id, max(salary) maxsal
        FROM employees
        GROUP BY department_id) d
WHERE e.department_id = d.department_id
AND e.salary = d.maxsal;
```

ORACLE Academy

DP 15-3
Gestión de Vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

8

Esta vista en línea busca el salario más alto para cada departamento y, a continuación, la consulta muestra el nombre del empleado con ese salario.

Una vista en línea debe tener un alias (en este ejemplo, "d") ya que funciona como un nombre de tabla en la cláusula FROM y "SELECT department_id, max(salary) …" no es un nombre de tabla válido.

ANÁLISIS DE N PRINCIPALES

- El análisis de N principales es una operación SQL utilizada para clasificar resultados
- El uso del análisis de N principales resulta útil cuando desea recuperar los 5 registros principales o los n registros principales de un juego de resultados devuelto por una consulta

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM employees
WHERE ROWNUM <=5
ORDER BY hire_date;
```

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
4	Whalen	17-Sep-1987
2	Kochhar	21-Sep-1989
3	De Haan	13-Jan-1993
5	Higgins	07-Jun-1994

ORACLE Academy

DP 15-3
Gestión de Vistas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

9

Sin embargo, los resultados de esta consulta no son lo que espera. El motivo es que la cláusula ORDER BY ejecute siempre se ejecuta la última, por lo que las filas se ordenan después de que se les asigne un número. La siguiente diapositiva muestra cómo resolver este problema.

ANÁLISIS DE N PRINCIPALES

- El análisis de N principales utiliza una vista en línea (subconsulta) para devolver un juego de resultados
- Puede utilizar ROWNUM en su consulta para asignar un número de fila al juego de resultados
- A continuación, la consulta principal utiliza ROWNUM para ordenar los datos y devolver los cinco principales

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM (SELECT last_name, hire_date
      FROM employees
      ORDER BY hire_date)
WHERE ROWNUM <=5;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Mediante una vista en línea, la cláusula ORDER BY se realiza antes de que se agregue ROWNUM.

ANÁLISIS DE N PRINCIPALES

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
2	Whalen	17-Sep-1987
3	Kochhar	21-Sep-1989
4	Hunold	03-Jan-1990
5	Ernst	21-May-1991

- En el ejemplo anterior, la vista en línea selecciona primero la lista de last_names y hire_dates de los empleados:

```
(SELECT last_name, hire_date FROM employees...
```

- A continuación, la vista en línea ordena los años del más antiguo al más reciente

```
...ORDER BY hire_date)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ANÁLISIS DE N PRINCIPALES

- La cláusula WHERE de la consulta externa se utiliza para restringir el número de filas devueltas y debe utilizar un operador < o <=

```
SELECT ROWNUM AS "Longest employed", last_name, hire_date
FROM (SELECT last_name, hire_date
      FROM employees
      ORDER BY hire_date)
WHERE ROWNUM <=5;
```

Longest employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
2	Whalen	17-Sep-1987
3	Kochhar	21-Sep-1989
4	Hunold	03-Jan-1990
5	Ernst	21-May-1991

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - DROP VIEW
  - VISTA EN LÍNEA
  - ANÁLISIS DE N PRINCIPALES
  - ROWNUM

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Crear y ejecutar una sentencia SQL que elimina una vista
  - Crear y ejecutar una consulta mediante una vista en línea
  - Crear y ejecutar una consulta para el análisis de N principales

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

16-1

Trabajar con Secuencias

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Enumerar al menos tres características útiles de una secuencia
  - Escribir y ejecutar una sentencia SQL que crea una secuencia
  - Consultar el diccionario de datos mediante USER_SEQUENCES para confirmar una definición de secuencia
  - Aplicar las reglas para utilizar NEXTVAL a fin de generar números únicos secuenciales en una tabla
  - Enumerar las ventajas y desventajas del almacenamiento en caché de los valores de secuencia
  - Nombrar tres motivos por los que se pueden producir intervalos en una secuencia

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Puede imaginar lo tedioso que puede ser tener que introducir los nombres de las 30.000 personas que participan en el Maratón de Londres en una base de datos, asegurándose al mismo tiempo de que no se le ha dado el mismo número de identificación a nadie?
- ¿Qué sucedería si se fuera a almorzar y al volver, otra persona hubiera introducido algunas de las solicitudes de los corredores?
- ¿Cómo sabría dónde volver a empezar?

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Afortunadamente, SQL tiene un proceso para generar automáticamente números únicos que elimina la preocupación por los detalles de números duplicados
- El proceso de numeración se maneja a través de un objeto de base de datos denominado SEQUENCE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objeto Sequence

- Ya sabe cómo crear dos tipos de objetos de base de datos, TABLE y VIEW
- Un tercer objeto de base de datos es SEQUENCE
- SEQUENCE es un objeto que se puede compartir utilizado para generar automáticamente números únicos
- Puesto que es un objeto que se puede, varios usuarios pueden acceder a él
- Normalmente, se utilizan secuencias para crear un valor de clave primaria

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objeto Sequence

- Como recordará, claves primarias deben ser únicas para cada fila. La secuencia se genera y aumenta (o disminuye) mediante una rutina interna de Oracle
- Este objeto ahorra tiempo porque reduce la cantidad de código que tiene que escribir

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objeto Sequence

- Los números de secuencia se almacenan y generan independientemente de las tablas
- Por lo tanto, la misma secuencia se puede utilizar para varias tablas
- Para crear SEQUENCE:

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

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

8

Aunque una secuencia se puede utilizar para varias tablas, no se recomienda, ya que cada número de la secuencia se utiliza solo una vez, a cada tabla le faltarán valores (los utilizados por las demás tablas).

Sintaxis de Sequence

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

secuencia	es el nombre del generador de secuencias (objeto).
INCREMENT BY n	especifica el intervalo entre los números de secuencia, donde n es un número entero (si esta cláusula se omite, la secuencia se va incrementando en 1).
START WITH n	especifica el primer número de secuencia que se va a generar (si se omite esta cláusula, la secuencia empieza con 1).

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

9

El nombre de la secuencia debe estar incluido en la sentencia CREATE SEQUENCE, todas las demás cláusulas son opcionales, pero se recomienda incluir todas las cláusulas.

Sintaxis de Sequence

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

MAXVALUE n	especifique el valor máximo que puede generar la secuencia.
NOMAXVALUE	especifica un valor máximo de 10^27 para una secuencia ascendente y –1 para una secuencia descendente (valor por defecto).
MINVALUE n	especifica el valor mínimo de secuencia.

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

10

MINVAUE se utilizará para una secuencia descendente.

Sintaxis de Sequence

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

NOMINVALUE	especifica un valor máximo de 1 para una secuencia ascendente y de – (10^26) para una secuencia descendente (por defecto).
CYCLE | NOCYCLE	especifica si la secuencia sigue generando valores después de alcanzar su valor máximo o mínimo (NOCYCLE es la opción por defecto).

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis de Sequence

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

CACHE n | NOCACHE	especifica cuántos valores asigna previamente y mantiene Oracle Server en la memoria. (Por defecto, Oracle Server almacena en caché 20 valores). Si el sistema falla, los valores se pierden.

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Secuencias

- En la SEQUENCE creada para los corredores del Maratón de Londres, los números se irán incrementando en 1, empezando por el número 1
- En este caso, empezar la secuencia con 1 es probablemente el mejor punto de partida
- Es tradicional que el mejor corredor del grupo de élite lleve el número 1

```
CREATE SEQUENCE runner_id_seq
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 50000
  NOCACHE
  NOCYCLE;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Secuencias

- Para otras situaciones, como, por ejemplo, identificadores de departamentos y de empleados se le puede asignar el número de inicio de forma diferente
- Puesto no será habrá por lo menos 30.000 corredores, el valor máximo de la secuencia se ha definido muy por encima del número esperado de corredores

```
CREATE SEQUENCE runner_id_seq
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 50000
  NOCACHE
  NOCYCLE;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Secuencias

- La opción NOCACHE evita que los valores de SEQUENCE se almacenan en la caché de la memoria, con lo que, en caso de fallo del sistema, se evitaría que se perdieran los números asignados previamente y retenidos en la memoria

```
CREATE SEQUENCE runner_id_seq
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 50000
  NOCACHE
  NOCYCLE;
```

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

15

El uso de la opción CACHE proporciona una ligera ventaja de rendimiento ya que los números se asignan previamente y almacenan en la memoria caché. Es más rápido de acceder que con NOCACHE, que llama al siguiente valor de la secuencia cuando se solicita.

Creación de Secuencias

- La opción NOCYCLE evita que la numeración vuelva a empezar en 1 si se excede el valor 50.000
- No utilice la opción CYCLE si la secuencia se utiliza para generar valores de clave primaria, a menos que disponga de un mecanismo fiable que suprima las filas antiguas más rápido que lo que se agregan las nuevas

```
CREATE SEQUENCE runner_id_seq
  INCREMENT BY 1
  START WITH 1
  MAXVALUE 50000
  NOCACHE
  NOCYCLE;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de Secuencias

- Para verificar que la secuencia se ha creado, consulte el diccionario de datos USER_OBJECTS
- Para ver toda la configuración de SEQUENCE, consulte el diccionario de datos USER_SEQUENCES como se muestra a continuación
- Visualice los nombres de valor en la sentencia SELECT, como se muestra a continuación

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

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de Secuencias

- Si se especifica NOCACHE, la columna last_number de la consulta muestra el siguiente número de secuencia disponible
- Si se especifica CACHE, la columna last_number muestra el siguiente número disponible de la secuencia que no se ha almacenado en caché en la memoria

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

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pseudocolumnas NEXTVAL y CURRVAL

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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pseudocolumnas NEXTVAL y CURRVAL

- En el ejemplo siguiente se inserta un nuevo departamento en la tabla DEPARTMENTS
- Utiliza la secuencia DEPARTMENTS_SEQ para generar un nuevo número de departamento como se muestra a continuación:

```
INSERT INTO departments
        (department_id, department_name, location_id)
VALUES  (departments_seq.NEXTVAL, 'Support', 2500);
```

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pseudocolumnas NEXTVAL y CURRVAL

- Supongamos ahora que desea contratar empleados para el nuevo departamento
- La sentencia INSERT que se debe ejecutar para los nuevos empleados 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 la secuencia denominada EMPLOYEE_SEQ ya se ha creado para generar los números de los nuevos empleados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pseudocolumnas NEXTVAL y CURRVAL

- La pseudocolumna CURRVAL del ejemplo siguiente se utiliza para hacer referencia al número de secuencia que acaba de generar el usuario actual
- 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 a CURRVAL con el nombre de secuencia

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pseudocolumnas NEXTVAL y CURRVAL

- Al hacer referencia a sequence.CURRVAL, se devuelve el último valor generado por el proceso de ese usuario

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

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- Después de crear una secuencia, se generan números secuenciales para utilizarlos en las tablas. Haga referencia a los valores de secuencia mediante las pseudocolumnas 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 de la sentencia INSERT
  - La cláusula VALUES de la sentencia INSERT
  - La cláusula SET de la sentencia UPDATE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- 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 las cláusulas GROUP BY, HAVING o ORDER BY
  - Una subconsulta en una sentencia SELECT, DELETE o UPDATE
  - La expresión DEFAULT en una sentencia CREATE TABLE o ALTER TABLE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- Para continuar con nuestro ejemplo del Maratón de Londres, se ha creado una tabla para los corredores:

```
CREATE TABLE runners
(runner_id NUMBER(6,0) CONSTRAINT runners_id_pk PRIMARY KEY,
 first_name VARCHAR2(30),
 last_name VARCHAR2(30));
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- A continuación, creamos la secuencia que generará valores para la columna de clave primaria runner_id

```
CREATE SEQUENCE runner_id_seq
   INCREMENT BY 1
   START WITH 1
   MAXVALUE 50000
   NOCACHE
   NOCYCLE;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- El uso de la siguiente sintaxis permitirá insertar nuevos participantes en la tabla de corredores
- El número de identificación del corredor se generará mediante la recuperación del NEXTVAL de la secuencia

```
INSERT INTO runners
        (runner_id, first_name, last_name)
VALUES  (runner_id_seq.NEXTVAL, 'Joanne', 'Everely');
```

```
INSERT INTO runners
        (runner_id, first_name, last_name)
VALUES  (runner_id_seq.NEXTVAL, 'Adam', 'Curtis');
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- Para confirmar que la secuencia funciona correctamente, se consulta la tabla:

```
SELECT runner_id, first_name, last_name
FROM runners;
```

RUNNER_ID	FIRST_NAME	LAST_NAME
1	Joanne	Everely
2	Adam	Curtis

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Uso de una Secuencia

- Para ver el valor actual de runners_id_seq, se utiliza CURRVAL
- Observe el uso de la tabla DUAL en este ejemplo
- Oracle Application Express no ejecutará esta consulta, pero debe comprender cómo funciona

```
SELECT runner_id_seq.CURRVAL
FROM dual;
```

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

30

Una vez que recupere NEXTVAL de la secuencia, ya no tiene ninguna "sesión" y APEX ya no sabe lo que está incluido en CURRVAL. Si una sentencia intenta recuperar CURRVAL, se generará un error de informe: ORA-08002: sequence SEQ_D_SONGS_SQ.CURRVAL is not yet defined in this session.

Uso de una Secuencia

- Las secuencias almacenadas en la caché de la memoria permiten un acceso más rápido a los valores de secuencia
- La caché se rellena la primera vez que hace referencia a la secuencia
- Las solicitudes del siguiente valor de secuencia se recuperan de la secuencia almacenada en caché
- Después de utilizar el último valor de secuencia, la siguiente solicitud de la secuencia introduce otra caché de secuencias en la memoria
- 20 es el número por defecto de los números de secuencia almacenados en caché

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Números no Secuenciales

- Aunque los generadores de secuencias emiten números secuenciales sin intervalos, esta acción se realiza independientemente de que se realice una confirmación o un rollback en la base de datos
- Los intervalos (números no secuenciales) se pueden generar:
  - Al realizar un rollback de una sentencia que contiene una secuencia, se pierde el número
  - Un fallo del sistema. Si la secuencia almacena en la caché los valores en la memoria y el sistema falla, dichos valores se pierden
  - El uso de la misma secuencia para varias tablas. Si lo hace así, cada tabla puede contener intervalos en los números secuenciales

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

32

COMMIT y ROLLBACK se tratarán más detalladamente más adelante en el curso.

Visualización del Siguiente Valor

- Si la secuencia se ha creado con NOCACHE, es posible ver el siguiente valor de secuencia disponible sin aumentarlo con la consulta de la tabla USER_SEQUENCES

```
SELECT sequence_name, min_value, max_value, last_number AS
"Next number"
FROM USER_SEQUENCES
WHERE sequence_name = 'RUNNER_ID_SEQ';
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	Next number (Siguiente número)
RUNNER_ID_SEQ	1	50000	3

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Modificación de una Secuencia

- Al igual que con otros objetos de base de datos que ha creado, SEQUENCE también se puede cambiar mediante la sentencia ALTER SEQUENCE
- ¿Qué ocurre si el Maratón de Londres ha excedido los 50.000 registros de corredores y necesita agregar más números?
- La secuencia se puede cambiar para aumentar el valor MAXVALUE sin cambiar el orden de números existente

```
ALTER SEQUENCE runner_id_seq
               INCREMENT BY 1
               MAXVALUE 999999
               NOCACHE
               NOCYCLE;
```

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Modificación de una Secuencia

- Se realiza alguna validación al modificar una secuencia
- Por ejemplo, no se puede ejecutar un nuevo valor MAXVALUE menor que el número de secuencia actual

```
ALTER SEQUENCE runner_id_seq
               INCREMENT BY 1
               MAXVALUE 90
               NOCACHE
               NOCYCLE;
```

```
ERROR at line 1:
ORA-04009: MAXVALUE cannot be made to be less than the current value
```

ORACLE Academy

DP 16-1
Trabajar con Secuencias

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Directrices de ALTER SEQUENCE

- Se aplican algunas directrices al ejecutar una sentencia ALTER SEQUENCE
- Tienen las siguientes características:
  - Debe ser el propietario o tener el privilegio ALTER para la secuencia para poder modificarla
  - Solo se ven afectados por la sentencia ALTER SEQUENCE los números de secuencia futuros
  - La opción START WITH no se puede cambiar mediante ALTER SEQUENCE. La secuencia se debe borrar y volver a crear para reiniciar la secuencia en un número diferente

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Eliminación de Secuencias

- 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
- Una vez eliminada, ya no se puede hacer referencia a la secuencia

```
DROP SEQUENCE runner_id_seq;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - CACHE/ NOCACHE
  - CREATE SEQUENCE
  - CURRVAL
  - CYCLE/ NOCYCLE
  - INCREMENT BY
  - MAXVALUE
  - MINVALUE

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - NEXTVAL
  - NOMAXVALUE
  - NOMINVALUE
  - Secuencias
  - START WITH

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Enumerar al menos tres características útiles de una secuencia
  - Escribir y ejecutar una sentencia SQL que crea una secuencia
  - Consultar el diccionario de datos mediante USER_SEQUENCES para confirmar una definición de secuencia
  - Aplicar las reglas para utilizar NEXTVAL a fin de generar números únicos secuenciales en una tabla
  - Enumerar las ventajas y desventajas del almacenamiento en caché de los valores de secuencia
  - Nombrar tres motivos por los que se pueden producir intervalos en una secuencia

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

16-2

Índices y Sinónimos

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

En esta lección se abordan los siguientes objetivos:

- Definir un índice y su uso como un objeto de esquema
- Nombrar las condiciones que causan que se cree un índice automáticamente
- Crear y ejecutar una sentencia CREATE INDEX y DROP INDEX
- Crear y ejecutar un índice basado en funciones
- Creación de Sinónimos Privados y Públicos

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Se puede imaginar ir a una librería o una biblioteca y encontrar todos los libros apilados en estanterías fila tras fila sin ningún orden real?
- La única forma de localizar lo que está buscando sería examinar todos los libros de todas las filas
- No es muy eficaz, ¿no está de acuerdo?
- Tal vez le sorprenda saber que la recuperación de datos de una base de datos es exactamente igual que las filas de libros
- Para cada consulta, se produce una exploración de la tabla completa

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Afortunadamente no existe una forma para que la búsqueda de datos en una base de datos sea más eficaz
- Oracle utiliza un índice para acelerar la recuperación de filas
- En esta lección, aprenderá cómo y cuándo crear un índice, además de cómo suprimir un índice
- Asimismo, en esta lección, aprenderá a crear nombres fáciles de recordar para los objetos de base de datos
- Al igual que las direcciones web de Internet que eliminan tener que conocer la dirección de protocolo de Internet, los sinónimos proporcionan una forma para asignar a un objeto un nombre más sencillo (Suena como alias, ¿no?)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices

- Un índice de Oracle Server es un objeto de esquema que puede acelerar la recuperación de filas mediante un puntero. Los índices se pueden crear explícita o automáticamente
- Si no hay un índice en la columna seleccionada, se produce una exploración de tabla completa
- Un índice proporciona acceso directo y rápido a las filas de una tabla
- Su finalidad es reducir la necesidad de E/S (entrada/salida) de disco mediante una ruta de acceso indexada para buscar datos de forma rápida

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices

- El índice lo utiliza y mantiene automáticamente Oracle Server. Una vez creado un índice, no será necesaria ninguna intervención directa por parte del usuario
- Un ROWID es una representación de cadena en base 64 de la dirección de fila que contiene el identificador de bloque, la ubicación de la fila en el bloque y el identificador de archivo de la base de datos
- Los índices utilizan los ROWID porque son la forma más rápida para acceder a cualquier fila concreta

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices

- Los índices son lógica y físicamente independientes de la tabla que indexan
- Esto significa que se pueden crear o borrar en cualquier momento sin que afecten a las tablas base o a otros índices

COUNTRY_ID	COUNTRY_NAME	CAPITOL	REGION_ID
1	United States of America	Washington, DC	21
2	Canada	Ottawa	21
3	Republic of Kazakhstan	Astana	143
7	Russian Federation	Moscow	151
12	Coral Sea Islands Territory	-	9
13	Cook Islands	Avarua	9
15	Europa Island	-	18
20	Arab Republic of Egypt	Cairo	15
...	...	...	...

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices

- Nota: al borrar una tabla, se borran también los índices correspondientes

COUNTRY_ID	COUNTRY_NAME	CAPITOL	REGION_ID
1	United States of America	Washington, DC	21
2	Canada	Ottawa	21
3	Republic of Kazakhstan	Astana	143
7	Russian Federation	Moscow	151
12	Coral Sea Islands Territory	-	9
13	Cook Islands	Avarua	9
15	Europa Island	-	18
20	Arab Republic of Egypt	Cairo	15
…	…	…	…

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Tipos de Índices

- Se pueden crear dos tipos de índices:
  - Índice único: Oracle Server crea automáticamente este índice al definir una restricción de clave PRIMARY KEY o UNIQUE en una columna de la tabla
    - El nombre del índice es el nombre proporcionado a la restricción
    - Aunque puede crear manualmente un índice único, se recomienda crear una restricción única en la tabla, que implícitamente crea el índice único
  - Índice no único: este es un índice que un usuario puede crear para acelerar el acceso a las filas
    - Por ejemplo, para optimizar las uniones, puede crear un índice en la columna FOREIGN KEY, que acelera la búsqueda de las filas coincidentes en la columna PRIMARY KEY

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Índices

- Para crear un índice en una o más columnas, emita la sentencia CREATE INDEX:

```
CREATE INDEX index_name
ON  table_name( column...,column)
```

- Para crear un índice en su propio, debe tener el privilegio CREATE TABLE
- Para crear un índice en cualquier esquema, necesita el privilegio CREATE ANY INDEX o el privilegio CREATE TABLE en la tabla en la que va a crear el índice. Los valores nulos no se incluyen en el índice

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Índices

- Por ejemplo, para mejorar la velocidad de acceso de consulta a la columna REGION_ID de la tabla WF_COUNTRIES:

```
CREATE INDEX wf_cont_reg_id_idx
ON wf_countries(region_id);
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cuándo Crear un Índice

- Se debe crear un índice solo si:
  - La columna contiene una amplia variedad de valores
  - Una columna contiene un gran número de valores nulos
  - Una o más columnas se utilizan con frecuencia en conjunto en una cláusula WHERE o una condición de unión
  - La tabla es grande y se espera que la mayoría de las consultas recuperen menos del 2 al 4% de las filas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cuándo no Crear un Índice

- Cuando decida si desea crear un índice, más no siempre es mejor
- Cada operación DML (INSERT, UPDATE, DELETE) que se realiza en una tabla con índices implica la actualización de los índices
- Cuanto mayor sea el número de índices asociados a una tabla, mayor será el esfuerzo para actualizar todos los índices después de la operación DML

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cuándo no Crear un Índice

- Por lo general, no merece la pena crear un índice si:
  - La tabla es pequeña
  - No se suelen utilizar las columnas como condición en la consulta
  - Se espera que la mayoría de las consultas recuperen más del 2 al 4% en la tabla
  - La tabla se actualiza con frecuencia
  - Se hace referencia a las columnas indexadas como parte de una expresión

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Compuestos

- Un índice compuesto (también denominado índice "concatenado") es un índice creado en varias columnas de una tabla
- Las columnas del índice compuesto pueden aparecer en cualquier orden y no es necesario que sean adyacentes en la tabla
- Los índices compuestos pueden acelerar la recuperación de datos para las sentencias SELECT en las que la cláusula WHERE hace referencia a todas o a la parte inicial de las columnas del índice compuesto

```
CREATE INDEX emps_name_idx
ON employees(first_name, last_name);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Compuestos

- Los valores nulos no se incluyen en el índice compuesto
- Para optimizar las uniones, puede crear un índice en la columna FOREIGN KEY, que acelera la búsqueda de las filas coincidentes en la columna PRIMARY KEY
- El optimizador no utiliza un índice si la cláusula WHERE contiene la expresión IS NULL

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de Índices

- Confirme la existencia de los índices de la vista del diccionario de datos USER_INDEXES
- También puede comprobar las columnas implicadas en un índice mediante la consulta de la vista USER_IND_COLUMNS
- La consulta que se muestran en la siguiente diapositiva es una unión entre la tabla USER_INDEXES (nombres de los índices y su unicidad) y la tabla USER_IND_COLUMNS (nombres de los índices, nombres de tabla y nombres de columna)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de Índices

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

INDEX NAME	COLUMN NAME	COL POS	UNIQUENESS
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

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Un índice basado en funciones almacena los valores indexados y utiliza el índice basado en una sentencia SELECT para recuperar los datos
- Un índice basado en funciones es un índice basado en expresiones
- La expresión de índice se genera a partir de las columnas de las tablas, restricciones, funciones SQL y funciones definidas por el usuario

ORACLE Academy

DP 16-2
Índices y Sinónimos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

20

La creación de funciones definidas por el usuario se trata en el curso Programación de Bases de Datos con PL/SQL.

Índices Basados en Funciones

- Los índices basados en funciones son útiles cuando no sabe en qué caso los datos se han almacenado en la base de datos
- Por ejemplo, puede crear un índice basado en funciones que se pueda utilizar con una sentencia SELECT utilizando UPPER en la cláusula WHERE
- El índice se utilizará en esta búsqueda

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

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

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. 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 realizar búsquedas no sensibles a mayúsculas y minúsculas
- Si no sabe la forma en que los apellidos del empleado se han introducido en la base de datos, puede seguir utilizando el índice introduciendo las mayúsculas en la sentencia SELECT
- Cuando se modifica una consulta mediante una expresión en la cláusula WHERE, el índice no se utilizará a menos que cree un índice basado en funciones para que coincida con la expresión

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Por ejemplo, la siguiente sentencia permite realizar búsquedas no sensibles a mayúsculas y minúsculas mediante el índice:

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

```
SELECT *
FROM employees
WHERE UPPER(last_name) LIKE 'KIN%';
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Para garantizar que Oracle Server utiliza el índice en lugar de realizar una exploración de tabla completa, asegúrese de que el valor de la función no sea nulo en consultas posteriores
- Por ejemplo, se garantiza que la siguiente sentencia utilice el índice, pero sin la cláusula WHERE, Oracle Server puede realizar una exploración de tabla completa:

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

- El servidor de Oracle trata los índices con columnas marcadas como DESC como índices basados en funciones

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Las columnas marcadas como DESC se ordenan en orden descendente
- Todos estos ejemplos usan las funciones UPPER y LOWER, pero cabe señalar que, aunque estas dos se utilizan con mucha frecuencia en los índices basados en funciones, la base de datos Oracle no se limita a ellos
- Se puede utilizar cualquier función incorporada de Oracle, por ejemplo, TO_CHAR

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Aquí se muestra otro ejemplo de índices basados en funciones
- Se realiza una consulta de la tabla de empleados para buscar cualquier empleado contratado en 1987

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE TO_CHAR(hire_date, 'yyyy') = '1987'
```

- Esta consulta da como resultado una exploración de tabla completa, que puede ser una operación muy cara si la tabla es grande
- Incluso aunque la columna hire_date esté indexada, no se utiliza el índice debido a la expresión TO_CHAR

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Índices Basados en Funciones

- Una vez que creamos el siguiente índice basado en funciones, podemos ejecutar la misma consulta, pero esta vez evitando la costosa exploración de tabla completa

```
CREATE INDEX emp_hire_year_idx
ON employees (TO_CHAR(hire_date, 'yyyy'));
```

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE TO_CHAR(hire_date, 'yyyy') = '1987'
```

FIRST_NAME	LAST_NAME	HIRE_DATE
Steven	King	17-Jun-1987
Jennifer	Whalen	17-Sep-1987

- Ahora, Oracle puede utilizar el índice en la columna hire_date

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Eliminación de Índices

- No puede modificar los índices
- Para cambiar un índice, debe borrarlo y volver a crearlo
- Elimine una definición de índice del diccionario de datos mediante la sentencia DROP INDEX

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Eliminación de Índices

- Para borrar un índice, debe ser el propietario del mismo o tener el privilegio DROP ANY INDEX
- Si borra una tabla, los índices y las restricciones se borran automáticamente, pero permanecen las vistas y las secuencias

```
DROP INDEX upper_last_name_idx;
```

```
DROP INDEX emps_name_idx;
```

```
DROP INDEX emp_hire_year_idx;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SYNONYM

- En SQL, como en el lenguaje, un sinónimo es una palabra o expresión que es un sustituto aceptado de otra palabra
- Los sinónimos se utilizan para simplificar el acceso a objetos creando otro nombre para el objeto
- Los sinónimos pueden hacer referencia a una tabla propiedad de otro usuario de forma más sencilla y reducir los nombres de objetos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SYNONYM

- Por ejemplo, para hacer referencia a la tabla amy_copy_employees del esquema del compañero de clase, puede poner como prefijo el nombre de la tabla con el nombre del usuario que la creó seguido de un punto y, a continuación, el nombre de la tabla, como en USMA_SBHS_SQL01_S04. amy_copy_employees
- Con la creación de un sinónimo se elimina la necesidad de cualificar el nombre del objeto con el esquema y se ofrecen nombres alternativos para tablas, vistas, secuencias, procedimientos u otros objetos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SYNONYM

- Este método es especialmente útil son nombres de objetos largos, como las vistas
- El administrador de la base de datos puede crear un sinónimo público accesible para todos los usuarios y puede otorgar específicamente el privilegio CREATE PUBLIC SYNONYM a cualquier usuario, y dicho usuario puede crear sinónimos públicos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SYNONYM

- Sintaxis:

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

- PUBLIC: crea un sinónimo al que pueden acceder todos los usuarios
- synonym: es el nombre del sinónimo que se va a crear
- object: identifica el objeto para el que se crea el sinónimo

```
CREATE SYNONYM amy_emps
FOR amy_copy_employees;
```

ORACLE Academy

DP 16-2
Índices y Sinónimos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

33

Nota: se produce un error de privilegios insuficientes al utilizar CREATE PUBLIC SYNONYM en APEX.
Se permite CREATE SYNONYM.

Para crear un SINÓNIMO para una tabla que no esté en el esquema, puede poner como prefijo de la tabla el nombre de esquema y un punto, por ejemplo:

CREATE SYNONYM amy_emps

FOR USMA_SBHS_SQL01_S04.amy_copy_employees;

Directrices de SYNONYM

- Instrucciones:
  - El objeto no puede estar en un paquete
  - Un nombre de sinónimo privado debe ser distinto de todos los demás objetos propiedad del mismo usuario
- Para eliminar un sinónimo:

```
DROP [PUBLIC] SYNONYM name_of_synonym
```

```
DROP SYNONYM amy_emps;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de SYNONYM

- La existencia de sinónimos se puede confirmar consultado la vista del diccionario de datos USER_SYNONYMS

Nombre de la Columna	Contenido
Synonym_name	Nombre del sinónimo.
Table_name	Propietario del objeto al que hace referencia el sinónimo.
Table_owner	Nombre del objeto al que hace referencia el sinónimo.
Db_link	Enlace de base de datos al que se hace referencia en un sinónimo remoto.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- Índice compuesto
- Confirmación de índice
- CREATE PUBLIC SYNONYM
- DROP INDEX
- Índice basado en funciones
- Índice no único
- Sinónimo
- Índice único

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Definir un índice y su uso como un objeto de esquema
- Nombrar las condiciones que causan que se cree un índice automáticamente
- Crear y ejecutar una sentencia CREATE INDEX y DROP INDEX
- Creación de Índices Basados en Funciones
- Creación de Sinónimos Privados y Públicos

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

17-1

Control del Acceso de los Usuarios

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Comparar la diferencia entre privilegios de objeto y privilegios del sistema
  - Construir los dos comandos necesarios para permitir que un usuario tenga acceso a una base de datos
  - Construir y ejecutar una sentencia GRANT… ON …TO para asignar privilegios a objetos de un esquema para otros usuarios y/o a PUBLIC
  - Consultar el diccionario de datos para confirmar los privilegios otorgados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Si comparte una computadora con otros usuarios, ya sea en la escuela o en casa, probablemente otra persona haya visto, cambiado o suprimido algo con lo que haya trabajado
- ¿No sería fantástico poder controlar los privilegios que otros usuarios tienen a sus archivos personales?
- En el caso de bases de datos, al igual que en la escuela o en casa, la seguridad de los datos es muy importante
- En esta lección, aprenderá a otorgar o quitar acceso a los objetos de base de datos como un medio para controlar quién puede modificar, suprimir, actualizar, insertar, indexar o hacer referencia a los objetos de base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Control del Acceso de los Usuarios

- En un entorno de varios usuarios, desea mantener la seguridad del uso y acceso a la base de datos
- Con la seguridad de la base de datos del servidor de Oracle, puede realizar lo siguiente:
  - Controlar el acceso a la base de datos
  - Proporcionar acceso a objetos específicos en la base de datos
  - Confirmar los privilegios asignados y recibidos en el diccionario de datos de Oracle
  - Crear sinónimos para objetos de bases de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Seguridad de la Base de Datos

- La seguridad de la base de datos se puede clasificar en dos categorías:
  - Seguridad del sistema
  - Seguridad de datos
- La seguridad del sistema abarca el acceso a la base de datos en el nivel del sistema, como la creación de usuarios, nombres de usuario y contraseñas, la asignación de espacio en disco a los usuarios y la concesión de los privilegios del sistema que los usuarios pueden llevar a cabo, como crear tablas, vistas y secuencias
- Existen más de 100 privilegios del sistema distintos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Seguridad de la Base de Datos

- La seguridad de datos (también denominada seguridad de objetos) está relacionada con los privilegios de objeto que abarca el acceso y el uso de los objetos de base de datos, así como las acciones que los usuarios pueden realizar sobre los objetos
- Estos privilegios incluyen poder ejecutar sentencias DML

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios y Esquemas

- Los privilegios son el derecho a ejecutar sentencias SQL determinadas
- El DBA es un usuario de alto nivel con capacidad para otorgar a los usuarios acceso a la base de datos y sus objetos
- Los usuarios necesitan privilegios del sistema para obtener acceso a la base de datos
- Necesitan privilegios de objeto para manipular el contenido de los objetos de la base de datos
- A los usuarios también se les da el privilegio de otorgar privilegios adicionales a otros usuarios o a los roles, que son grupos con nombres de privilegios relacionados

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios y Esquemas

- Un esquema es una recopilación de objetos, como tablas, vistas y secuencias
- El esquema es propiedad de un usuario de base de datos y tiene el mismo nombre que el usuario
- En este curso, su nombre de esquema es una combinación de su país/estado, escuela, curso y número de alumno
- Por ejemplo: uswa_skhs_sql01_s22

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Seguridad del Sistema

- Este nivel de seguridad abarca el acceso y uso de la base de datos en el nivel del sistema
- Existen más de 100 privilegios del sistema distintos
- Los privilegios del sistema, como la capacidad de crear o eliminar usuarios, eliminar tablas o realizar copias de seguridad de tablas, solo los suele tener el DBA

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Seguridad del Sistema

- En esta tabla se muestran algunos de los privilegios del sistema que el DBA normalmente no otorga a otros usuarios
- ¿Desea que otro usuario pueda para borrar sus tablas?

Privilegio del Sistema	Operaciones Autorizadas
CREATE USER	Los usuarios con privilegios pueden crear otros usuarios de Oracle (privilegio necesario para el rol DBA).
DROP USER	El usuario con privilegios puede borrar otro usuario.
DROP ANY TABLE	El usuario con privilegios puede borrar una tabla en cualquier esquema.
BACKUP ANY TABLE	El usuario con privilegios puede realizar una copia de seguridad de tablas en cualquier esquema con la utilidad de exportación.
SELECT ANY TABLE	El usuario con privilegios puede consultar las tablas, vistas o instantáneas en cualquier esquema.
CREATE ANY TABLE	El usuario con privilegios puede crear una tabla en cualquier esquema.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios del Sistema

- El DBA crea el usuario mediante la ejecución de la sentenciaCREATE USER
- El usuario no tiene ningún privilegio en este punto
- El DBA puede otorgar privilegios necesarios a dicho usuario
- Sintaxis:

```
CREATE USER user
IDENTIFIED BY password;
```

- Ejemplo:

```
CREATE USER  scott
IDENTIFIED BY ur35scott;
```

ORACLE Academy

DP 17-1
Control del Acceso de los Usuarios

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios del Sistema

- Mediante la sentencia ALTER USER, un usuario puede cambiar su contraseña
- Ejemplo:

```
ALTER USER scott
IDENTIFIED BY imscott35;
```

ORACLE Academy

DP 17-1
Control del Acceso de los Usuarios

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

13

Los alumnos no podrán probar estos privilegios en APEX, debido a privilegios insuficientes.

Privilegios del Sistema de Usuario

- El DBA utiliza la sentencia GRANT para asignar privilegios del sistema al usuario
- Estos privilegios del sistema determinan lo que el usuario puede realizar en el nivel de base de datos
- Una vez que el usuario ha otorgado los privilegios, el usuario puede utilizar estos privilegios inmediatamente

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

```
GRANT  create session, create table, create sequence, create
view
TO scott;
```

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios del Sistema de Usuario

- Un usuario debe tener un privilegio CREATE SESSION y un identificador de usuario si debe poder acceder a una base de datos
- No puede emitir el comando CREATE SESSION en Oracle Application Express; esto se produce de forma automática en segundo plano

Privilegio del Sistema	Operaciones Autorizadas
CREATE SESSION	Conectar a la base de datos.
CREATE TABLE	Crear tablas en el esquema del usuario.
CREATE SEQUENCE	Crear una secuencia en el esquema del usuario.
CREATE VIEW	Crear una vista en el esquema del usuario.
CREATE PROCEDURE	Crear un paquete, función o procedimiento en el esquema del usuario.

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Seguridad de Objetos

- Este nivel de seguridad abarca el acceso y el uso de los objetos de la base de datos así como las acciones que los usuarios puedan realizar en dichos objetos

ORACLE Academy

DP 17-1
Control del Acceso de los Usuarios

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Objeto

- Cada objeto tiene un juego determinado de privilegios que se pueden otorgar
- En la tabla siguiente se muestran los privilegios para varios objetos

Privilegio de Objeto	Tabla	Ver	Secuencia	Procedimiento
ALTER	X		X	
DELETE	X	X		
EXECUTE				X
INDEX	X	X		
INSERT	X	X		
REFERENCES	X			
SELECT	X	X	X	
UPDATE	X	X		

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Objeto

- Es importante tener en cuenta los siguientes cuatro puntos sobre los privilegios de objeto:
  - Los únicos privilegios que se aplican a una secuencia son SELECT y ALTER
  - Recuerde que una secuencia utiliza ALTER para cambiar las opciones INCREMENT, MAXVALUE, CACHE/NOCACHE o CYCLE/NOCYCLE
  - La opción START WITH no se puede cambiar mediante ALTER

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Objeto

- Puede otorgar los privilegios UPDATE, REFERENCES e INSERT en columnas individuales de una tabla
- Por ejemplo:

```
GRANT UPDATE (salary)
      ON employees TO steven_king
```

- Un privilegio SELECT se puede restringir mediante la creación de una vista con un subjuego y otorgamiento del privilegio SELECT solo en la vista
- No puede otorgar SELECT en columnas individuales

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Objeto

- Un privilegio otorgado en un sinónimo se convierte en un privilegio en la tabla base a la que hace referencia el sinónimo
- Es decir, un sinónimo es simplemente un nuevo nombre más fácil de utilizar
- El uso de este nombre para otorgar un privilegio es lo mismo que otorga el privilegio en la propia tabla

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Palabra Clave PUBLIC

- Un propietario de una tabla puede otorgar acceso a todos los usuarios mediante la palabra clave PUBLIC
- En el segundo ejemplo que se muestra a continuación, se permite a todos los usuarios del sistema consultar datos de la tabla DEPARTMENTS de Alice

```
GRANT select
   ON alice.departments
TO PUBLIC;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Palabra Clave PUBLIC

- Si una sentencia no utiliza el nombre completo de un objeto, Oracle Server incluye implícitamente en el nombre de objeto el nombre del usuario (o esquema) actual como prefijo
- Si el usuario Scott consulta la tabla DEPARTMENTS, por ejemplo, el sistema selecciona en la tabla SCOTT.DEPARTMENTS
- Si una sentencia no utiliza el nombre completo de un objeto y el usuario actual no es propietario de un objeto con ese nombre, el sistema incluye como prefijo PUBLIC en el nombre de objeto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Palabra Clave PUBLIC

- Por ejemplo, si el usuario Scott consulta la vista USER_OBJECTS y Scott no es propietario de dicha tabla, el sistema realiza selecciones en la vista del diccionario de datos mediante el sinónimo público PUBLIC.USER_OBJECTS

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Confirmación de Privilegios Otorgados

- Si intenta realizar una operación no autorizada, como la supresión de una fila de una tabla para la que no tiene el privilegio DELETE, el servidor de Oracle no permite que se produzca la operación
- Si recibe el mensaje de error de Oracle Server "table or view does not exist", es porque ha realizado una de las siguientes acciones:
  - Asignar un nombre a una tabla o vista que no existe
  - Intentar realizar una operación en una tabla o vista para la que no tiene el privilegio adecuado

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Visualización

- Puede acceder al diccionario de datos para ver los privilegios que tiene
- En el gráfico mostrado se describen varias vistas del diccionario de datos
- Mediante Oracle Application Express Developer, acceda a SQL Workshop, Utilities, Object Reports
- Los privilegios del usuario se pueden ver en la sección Security Reports

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Privilegios de Visualización

Vista del Diccionario de Datos	Description
ROLE_SYS_PRIVS	Privilegios del sistema otorgados a roles
ROLE_TAB_PRIVS	Privilegios de tabla otorgados a roles
USER_ROLE_PRIVS	Roles a los que puede acceder el usuario
USER_TAB_PRIVS_MADE	Privilegios de objeto otorgados a objetos del usuario
USER_TAB_PRIVS_RECD	Privilegios de objeto otorgados al usuario
USER_COL_PRIVS_MADE	Privilegios de objeto otorgados a columnas de objetos del usuario
USER_COL_PRIVS_RECD	Privilegios de objeto otorgados al usuario en columnas específicas
USER_SYS_PRIVS	Muestra privilegios del sistema otorgados al usuario

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Privilegio CREATE SESSION
  - Privilegio GRANT
  - Privilegios de Objeto
  - Seguridad de objetos
  - de Objeto
  - Privilegio PUBLIC
  - Role

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Esquema
  - Privilegios del Sistema
  - Seguridad del sistema

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Comparar la diferencia entre privilegios de objeto y privilegios del sistema
  - Construir los dos comandos necesarios para permitir que un usuario tenga acceso a una base de datos
  - Construir y ejecutar una sentencia GRANT… ON …TO para asignar privilegios a objetos de un esquema para otros usuarios y/o a PUBLIC
  - Consultar el diccionario de datos para confirmar los privilegios otorgados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

17-2

Creación y Revocación de Privilegios de Objeto

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Explicar qué es un ROL y cuáles son sus ventajas
  - Crear una sentencia para crear un ROL y OTORGARLE privilegios
  - Crear una sentencia GRANT .. ON .. TO.. WITH GRANT OPTION para asignar privilegios a objetos de su esquema para otros usuarios y/o a PUBLIC
  - Crear y ejecutar una sentencia para REVOCAR los privilegios de objeto a otros usuarios y/o PUBLIC

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Distinción entre Privilegios y Roles
  - Explicar el objetivo de un enlace de base de datos

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Si comparte una computadora con otros usuarios, ya sea en la escuela o en casa, probablemente otra persona haya visto, cambiado o suprimido algo con lo que haya trabajado
- ¿No sería fantástico poder controlar los privilegios que otros usuarios tienen a sus archivos personales?
- En el caso de bases de datos, al igual que en la escuela o en casa, la seguridad de los datos es muy importante
- En esta lección, aprenderá a otorgar o quitar acceso a los objetos de base de datos como un medio para controlar quién puede modificar, suprimir, actualizar, insertar, indexar o hacer referencia a esos objetos de base de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones

- Un rol es un grupo con nombre de privilegios relacionados que se pueden otorgar a un usuario
- Este método facilita la revocación y el mantenimiento de los privilegios
- Un usuario puede tener acceso a diferentes roles y el mismo rol se puede asignar a diferentes usuarios
- Los roles normalmente se crean para una aplicación de base de datos

ORACLE Academy

DP 17-2
Creación y Revocación de Privilegios de Objeto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

6

El uso de roles hace que el proceso de otorgar privilegios sea mucho más fácil. En lugar de asignar privilegios individuales a cientos de usuarios, el DBA puede crear un rol, asignar privilegios al rol y, a continuación, otorgar el rol a los usuarios.

Funciones

- Para crear y asignar un rol, en primer lugar el DBA debe crear el rol
- A continuación, el DBA puede asignar privilegios al rol, y dicho rol a los usuarios

```
CREATE ROLE manager;
    Role created.
GRANT create table, create view TO manager;
    Grant succeeded.
GRANT manager TO jennifer_cho;
    Grant succeeded.
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones

- Utilice la sintaxis siguiente para crear un rol:

```
CREATE ROLE role_name;
```

- Después de crear el rol, el DBA puede utilizar la sentencia GRANT para asignar el rol a los usuarios, así como asignar privilegios al rol

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones

- En el ejemplo que se muestra se crea un rol manager y, a continuación, se permite a los gestores crear tablas y vistas
- A continuación, se otorga el rol a un usuario
- Ahora el usuario puede crear tablas y vistas

```
CREATE ROLE manager;
    Role created
GRANT create table, create view TO manager;
    Grant succeeded
GRANT manager TO jennifer_cho;
    Grant succeeded
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones

- Si los usuarios tiene varios roles otorgados, reciben todos los privilegios asociados a todos los roles
- Nota: CREATE ROLE es un privilegio del sistema que no se ha emitido para las clases de Academy

```
CREATE ROLE manager;
    Role created
GRANT create table, create view TO manager;
    Grant succeeded
GRANT manager TO jennifer_cho;
    Grant succeeded
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Características de los Roles

- Los roles son grupos con nombre de privilegios relacionados
- Se pueden otorgar a usuarios
- Simplifican el proceso de otorgamiento y revocación de privilegios
- Los crea un DBA

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Asignación de Privilegios de Objeto

- Utilice la siguiente sintaxis para otorgar privilegios de objeto:

```
GRANT object_priv [(column_list)]
  ON object_name
TO {user|role|PUBLIC}
  [WITH GRANT OPTION];
```

Sintaxis	Definida
object_priv	es un privilegio de objeto que se va a otorgar
column_list	especifica la columna de una tabla o vista en la que se otorgan los privilegios
ON object_name	es el objeto sobre el que se otorgan los privilegios
TO user|role	identifica el usuario o rol al que se le otorga el privilegio
PUBLIC	otorga privilegios de objeto a todos los usuarios
WITH GRANT OPTION	permite al usuario con privilegios otorgar privilegios de objeto a otros usuarios y roles

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Directrices de Privilegios de Objeto

- Para otorgar privilegios en un objeto, el objeto debe estar en el propio esquema o debe otorgar el privilegio con WITH GRANT OPTION
- Un propietario de objeto puede otorgar privilegios de objeto en el objeto a cualquier otro usuario o rol de la base de datos
- El propietario de un objeto adquiere automáticamente todos los privilegios de objeto en dicho objeto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de GRANT

- Scott King (nombre de usuario scott_king) ha creado una tabla de clientes
- En el ejemplo 1 de la derecha, a todos los usuarios se les ha otorgado permiso para SELECCIONAR en la tabla de clientes de Scott
- En el ejemplo 2 se otorgan privilegios UPDATE a Jennifer y al rol de gestor en columnas específicas de la tabla de clientes de Scott

```
1. GRANT SELECT
   ON clients
   TO PUBLIC;

2. GRANT UPDATE(first_name,
                last_name)
   ON clients
   TO jennifer_cho, manager;

3. SELECT  *
   FROM scott_king.clients;

4. CREATE SYNONYM clients
   FOR scott_king.clients;

5. SELECT *
   FROM clients;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de GRANT

- Si Jennifer ahora desea SELECCIONAR datos de la tabla de Scott, la sintaxis que debe utilizar se muestra en el ejemplo 3
- Asimismo, Jennifer puede crear un sinónimo para la tabla de Scott y SELECCIONAR en el sinónimo
- Consulte la sintaxis en los ejemplos 4 y 5

```
1. GRANT SELECT
   ON clients
   TO PUBLIC;

2. GRANT UPDATE(first_name,
                last_name)
   ON clients
   TO jennifer_cho, manager;

3. SELECT  *
   FROM scott_king.clients;

4. CREATE SYNONYM clients
   FOR scott_king.clients;

5. SELECT *
   FROM clients;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de GRANT

- Hay disponibles diferentes privilegios de objeto para distintos tipos de objetos de esquema
- Un usuario automáticamente tiene todos los privilegios de objeto para los objetos de esquema que están en su esquema
- Un usuario puede otorgar privilegios de objeto sobre cualquier objeto de esquema que el usuario posea a otro usuario o rol

```
1. GRANT SELECT
   ON clients
   TO PUBLIC;

2. GRANT UPDATE(first_name,
                last_name)
   ON clients
   TO jennifer_cho, manager;

3. SELECT  *
   FROM scott_king.clients;

4. CREATE SYNONYM clients
   FOR scott_king.clients;

5. SELECT *
   FROM clients;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

WITH GRANT OPTION

- El usuario con privilegios puede transferir un privilegio que se otorga con la cláusula WITH GRANT OPTION a otros usuarios y roles
- Los privilegios de objeto otorgados con la cláusula WITH GRANT OPTION se revocan si se revoca el privilegio del otorgante

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

WITH GRANT OPTION

- En el ejemplo siguiente se proporciona al usuario Scott acceso a la tabla de clientes con los privilegios para consultar la tabla y agregar filas a la tabla
- En el ejemplo también se permite a Scott otorgar a otros usuarios estos privilegios:

```
GRANT   SELECT, INSERT
ON      clients
TO      scott_king
WITH    GRANT OPTION;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Palabra Clave PUBLIC

- Un propietario de una tabla puede otorgar acceso a todos los usuarios mediante la palabra clave PUBLIC
- El segundo ejemplo de la diapositiva permite a todos los usuarios del sistema consultar datos de la tabla de clientes de Jason

```
GRANT   SELECT
ON      jason_tsang.clients
TO      PUBLIC;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objeto DELETE

- Si intenta realizar una operación no autorizada, como la supresión de una fila de una tabla en la que no tiene el privilegio DELETE, Oracle Server no permite que se produzca la operación
- Si recibe el mensaje de error de Oracle Server "table or view does not exist", es porque ha realizado una de las siguientes acciones:
  - Hacer referencia a una tabla o vista que no existe
  - Intentar realizar una operación en una tabla o vista para la que no tiene los privilegios adecuados

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Revocación de Privilegios de Objeto

- Puede eliminar los privilegios otorgados a otros usuarios mediante la sentencia REVOKE
- Al utilizar la sentencia REVOKE, los privilegios que especifique se revocarán a los usuarios que designe y a otros usuarios a los que se haya otorgado estos privilegios con las palabras clave WITH GRANT OPTION

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Revocación de Privilegios de Objeto

- Utilice la siguiente sintaxis para revocar privilegios de objeto:

```
REVOKE {privilege [, privilege...]|ALL}
ON  object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```

- CASCADE CONSTRAINTS es obligatorio para eliminar todas las restricciones de integridad referenciales realizadas sobre el objeto mediante el privilegio REFERENCES

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

With Grant Option

- En el ejemplo siguiente, se revocan los privilegios SELECT e INSERT proporcionados al usuario Scott en la tabla de clientes

```
REVOKE SELECT, INSERT
ON clients
FROM scott_king;
```

- Si se otorga a un usuario un privilegio con la cláusula WITH GRANT OPTION, dicho usuario también puede otorgar el privilegio mediante la cláusula WITH GRANT OPTION
- Esto significa que es posible una larga cadena de usuarios con privilegios, pero no se permiten otorgar permisos de forma circular

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

With Grant Option

- Si el propietario revoca un privilegio de un usuario que ha otorgado privilegios a otros usuarios, la sentencia de revocación tiene un efecto en cascada en todos los privilegios otorgados

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

With Grant Option

- Por ejemplo, si el usuario A otorga un privilegio SELECT en una tabla al usuario B, con la cláusula WITH GRANT OPTION, el usuario B puede otorgar al usuario C el privilegio SELECT también con la cláusula WITH GRANT OPTION

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

With Grant Option

- Ahora, el usuario C puede otorgar al usuario D el privilegio SELECT

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

With Grant Option

- Sin embargo, si el usuario A revoca los privilegios del usuario B, los privilegios otorgados a los usuarios C y D también se revocan

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sinónimos Privados y Públicos

- Como se ha mencionado anteriormente en esta lección, puede crear un sinónimo para eliminar la necesidad de cualificar el nombre del objeto con el esquema y se ofrecen nombres alternativos para tablas, vistas, secuencias, procedimientos u otros objetos
- Los sinónimos pueden ser privados (valor por defecto) o públicos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sinónimos Privados y Públicos

- Un sinónimo público lo pueden crear los administradores de base de datos o usuarios de base de datos a los que se haya dado los privilegios para hacerlo, pero no todo el mundo puede crear automáticamente sinónimos públicos
- Nota: el privilegio CREATE PUBLIC SYNONYM no se ha otorgado a los alumnos de Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Roles y Privilegios

- Los roles y privilegios difieren de varias formas:
  - Un privilegio de usuario es un derecho para ejecutar un tipo concreto de sentencia SQL o un derecho para acceder a un objeto de otro usuario
  - Oracle define todos los privilegios
  - Los roles, por otra parte, los crean los usuarios (normalmente administradores) y se utilizan para agrupar los privilegios u otros roles
  - Se crean para que sea más fácil gestionar el otorgamiento de varios privilegios o roles a los usuarios
  - Los privilegios se incluyen con la base de datos y los roles los realizan los administradores de base de datos o los usuarios de una base de datos concreta

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- Un enlace de base de datos es un puntero que define una ruta de acceso de comunicación unidireccional de una base de datos Oracle a otra base de datos
- El puntero del enlace se define realmente como una entrada en la tabla del diccionario de datos
- Para acceder al enlace, debe estar conectado a la base de datos local que contiene la entrada del diccionario de datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- Una conexión de enlace de base de datos es "unidireccional" en el sentido de que un cliente conectado a la base de datos local A puede utilizar un enlace almacenado en la base de datos A para acceder a información de la base de datos remota B, pero los usuarios conectados a la base de datos B no pueden utilizar el mismo enlace para acceder a los datos de la base de datos A

scott_king

Base de datos HQ_ACME.COM

```
SELECT * FROM
emp@HQ_ACME.COM
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- CREATE DATABASE LINK: en Oracle Application Express, no hay una conexión constante a la base de datos y, en consecuencia, esta función no está disponible
- Si los usuarios locales de la base de datos B desean acceder a los datos de la base de datos A, deben definir un enlace que se almacena en el diccionario de datos de la base de datos B
- Una conexión de enlace de base de datos proporciona a los usuarios locales acceso a los datos en una base de datos remota

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- Para que se produzca esta conexión, cada base de datos del sistema distribuido debe tener un nombre de base de datos global único
- El nombre de base de datos global identifica de forma única un servidor de base de datos en un sistema distribuido
- La gran ventaja de los enlaces de base de datos es que permiten a los usuarios acceder a objetos de otros usuarios en una base de datos remota, para que estén limitados por el juego de privilegios del propietario del objeto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- Es decir, un usuario local puede acceder a una base de datos remota sin tener que ser un usuario en la base de datos remota
- En el ejemplo aparece un usuario scott_king que accede a la tabla EMP en la base de datos remota con el nombre global HQ.ACME.COM
- Normalmente, el administrador de base de datos es el responsable de crear el enlace de la base de datos

scott_king

Base de datos HQ_ACME.COM

```
SELECT * FROM
emp@HQ_ACME.COM
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- La vista del diccionario USER_DB_LINKS contiene información sobre los enlaces a los que tiene acceso un usuario
- Una vez que se ha creado el enlace de base de datos, puede escribir sentencias SQL para los datos en el sitio remoto
- Si se configura un sinónimo, puede escribir sentencias SQL mediante el sinónimo

scott_king

Base de datos HQ_ACME.COM

```
SELECT * FROM
emp@HQ_ACME.COM
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Enlaces de Base de Datos

- Por ejemplo:

```
CREATE PUBLIC SYNONYM HQ_EMP
 FOR emp@HQ.ACME.COM;
```

- A continuación, escriba la sentencia SQL que utiliza el sinónimo:

```
SELECT *
FROM HQ_EMP;
```

- no puede otorgar privilegios en objetos remotos

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Privilegio CREATE ROLE
  - WITH GRANT OPTION
  - Privilegio REVOKE
  - Sentencia REVOKE
  - PUBLIC SYNONYM
  - PRIVATE SYNONYM
  - Enlaces de Base de Datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Explicar qué es un ROL y cuáles son sus ventajas
  - Crear una sentencia para crear un ROL y OTORGARLE privilegios
  - Crear una sentencia GRANT .. ON .. TO.. WITH GRANT OPTION para asignar privilegios a objetos de su esquema para otros usuarios y/o a PUBLIC
  - Crear y ejecutar una sentencia para REVOCAR los privilegios de objeto a otros usuarios y/o PUBLIC

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Distinción entre Privilegios y Roles
  - Explicar el objetivo de un enlace de base de datos

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

17-3

Expresiones regulares

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Describir las expresiones regulares
  - Utilizar expresiones regulares para buscar, hacer coincidir y sustituir cadenas en sentencias SQL
  - Construir y ejecutar expresiones regulares y restricciones de control

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- En ocasiones, puede que tenga que buscar o sustituir una parte concreta de texto en una columna, cadena de texto o documento
- Ya sabe cómo realizar una coincidencia de patrón simple mediante LIKE y comodines
- En ocasiones puede que deba buscar cadenas de texto muy complejas como, por ejemplo, la extracción de todas URL de un fragmento de texto
- Otras veces, es posible que se le solicite que realice una búsqueda más compleja como buscar todas las palabras cuyo segundo carácter es una vocal

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Las expresiones regulares son un método de descripción de patrones tanto simples como complejos para la búsqueda y la manipulación
- Se utilizan ampliamente en el sector informático y no están limitadas a Oracle
- La implantación de expresiones regulares de Oracle es una extensión de POSIX (Portable Operating System for UNIX) y, como tal, es completamente compatible con el estándar POSIX, controlado por Institute of Electrical and Electronics Engineers (IEEE)

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Las expresiones regulares son un tema muy detallado y profundo. El único objetivo de esta lección es proporcionar una breve visión general de su finalidad y su uso.

Expresiones regulares

- El uso de expresiones normales se basa en el uso de metacaracteres
  - Los metacaracteres son caracteres con un significado especial, como un carácter comodín, un carácter repetitivo, un carácter no coincidente o un rango de caracteres
  - Puede utilizar varios símbolos de metacaracteres predefinidos en la coincidencia de patrones
  - En las siguientes diapositivas se muestran algunos de los metacaracteres y se proporcione una breve explicación de cada uno de ellos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Metacaracteres

Símbolo	Description
. (punto)	Coincide con cualquier carácter en el juego de caracteres soportados, excepto NULL
?	Coincide con ninguna o una incidencia
*	Coincide con cero o más incidencias
+	Coincide con una o más incidencias
()	Expresión de agrupación, que se trata como una sola subexpresión
\	Carácter de escape
|	Operador de alternancia para especificar coincidencias alternativas
^/$	Coincide con el principio de línea/final de línea
[ ]	Expresión de corchetes para una lista coincidente que coincide con cualquier expresión representada en la lista

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

La lista completa de metacaracteres con ejemplos de su uso se puede encontrar en http://docs.oracle.com

Ejemplos de Expresiones Normales

- Una expresión regular simple es muy similar a la búsqueda con comodines con la que ya está familiarizado
- Veamos un ejemplo: vamos a utilizar el operador de punto para buscar la letra "a" seguida de cualquier carácter seguido de la letra "c"
- Como expresión regular, esto se haría como: 'a.c'
- La misma expresión como búsqueda con comodines SQL estándar sería: WHERE column LIKE 'a_c'

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Expresiones Normales

- ¿Cuáles de las siguientes cadenas coincidirá con 'a.c'?
  - 'ABC',
  - 'abc',
  - 'aqx',
  - 'axc',
  - 'aBc',
  - 'abC'
  - 'Amc'
  - 'amrc'

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Expresiones Normales

- Las cadenas en rojo coinciden con la cadena de búsqueda 'a.c.'
- 'ABC', 'abc', 'aqx', 'axc', 'aBc', 'abC', 'Amc', 'amrc'
- El resto de ejemplos falla debido a que tienen el carácter en la posición incorrecta o con las mayúsculas/minúsculas incorrectos (mayúsculas no minúsculas según lo especificado en la cadena de búsqueda)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones de Expresiones Regulares

- Oracle proporciona un juego de funciones SQL que puede utilizar para buscar y manipular cadenas mediante expresiones regulares
- Puede utilizar estas funciones en cualquier tipo de dato que contenga datos de caracteres como CHAR, CLOB y VARCHAR2
- Una expresión regular debe ir entre comillas simples

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones de Expresiones Regulares

Name	Description
REGEXP_LIKE	Similar to the LIKE operator, but performs regular expression matching instead of simple pattern matching
REGEXP_REPLACE	Searches for a regular expression pattern and replaces it with a replacement string
REGEXP_INSTR	Searches for a given string for a regular expression pattern and returns the position where the match is found
REGEXP_SUBSTR	Searches for a regular expression pattern within a given string and returns the matched substring
REGEXP_COUNT	Returns the number of times a pattern appears in a string. You specify the string and the pattern. You can also specify the start position and matching options (for example, c for case sensitivity).

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Funciones de Expresiones Regulares

- Supongamos que se le ha pedido que enumere todos los empleados con el nombre Stephen o Steven

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE(first_name, '^Ste(v|ph)en$');
```

- Con expresiones regulares, solo tiene que utilizar la función REGEXP_LIKE y la cadena de búsqueda: '^Ste(v|ph)en$'
  - "^" especifica el inicio de la cadena que se está buscando
  - "S" mayúscula, seguida de
  - "t" minúscula, seguida de
  - "e" minúscula, seguida de

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Funciones de Expresiones Regulares

- Con expresiones regulares, solo tiene que utilizar la función REGEXP_LIKE y la cadena de búsqueda: '^Ste(v|ph)en$'
  - "(" inicia una subexpresión
  - "v" minúscula
  - "|" especifica un O
  - "p" minúscula seguida de "h" mayúscula
  - ")" termina con el grupo de opciones,
  - "e" minúscula
  - "n" minúscula
  - "$" especifica el final de la cadena que se está buscando

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Funciones de Expresiones Regulares

- Ejemplo:

```
SELECT first_name, last_name
FROM employees
WHERE REGEXP_LIKE(first_name, '^Ste(v|ph)en$');
```

- Resultado:

FIRST_NAME	LAST_NAME
Steven	King

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Funciones de Expresiones Regulares

- La función REPLACE de expresión regular sustituirá un patrón de cadena por otro
- En este ejemplo se busca una "H" seguida de cualquier vocal y se sustituyen por dos símbolos "*"

```
SELECT last_name, REGEXP_REPLACE(last_name, '^H(a|e|i|o|u)',
'**')
  AS "Name changed"
FROM employees;
```

LAST_NAME	"Name changed"
Abel	Abel
Davies	Davies
…	…
Hartstein	**rtstein
Higgins	**ggins
…	…

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplos de Funciones de Expresiones Regulares

- La función COUNT de expresión regular devuelve el número de veces que aparece un patrón en una cadena
- En este ejemplo se busca la subexpresión "ab"

```
SELECT country_name, REGEXP_COUNT(country_name, '(ab)') AS
"Count of 'ab'"
FROM wf_countries
WHERE REGEXP_COUNT(country_name, '(ab)')>0;
```

COUNTRY_NAME	Count of 'ab'
Republic of Zimbabwe	1
Arab Republic of Egypt	1
Great Socialist Peoples Libyan Arab Jamahiriya	1
Kingdom of Saudi Arabia	1
Syrian Arab Republic	1
Gabonese Republic	1
United Arab Emirates	1

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Subexpresión: expresión que forma parte de una expresión mayor.

Expresiones Regulares en Restricciones de Control

- Las expresiones regulares también se pueden utilizar como parte del código de la aplicación para garantizar que solo se almacenan los datos en la base de datos
- Se pueden incluir una llamada a una función de expresión regular, por ejemplo, una restricción CHECK

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Expresiones Regulares en Restricciones de Control

- Por lo tanto, si desea asegurarse de que no se captura ninguna dirección de correo electrónico sin "@" en una tabla en la base de datos, solo tiene que agregar las siguientes restricciones de control:

```
ALTER TABLE employees
ADD CONSTRAINT email_addr_chk
CHECK(REGEXP_LIKE(email,'@'));
```

- Esto garantizará que todas las direcciones de correo electrónico incluyan un signo "@"

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Expresiones Regulares en Restricciones de Control

- Mediante expresiones regulares, puede comprobar el formato de las direcciones de correo electrónico más a fondo para comprobar que son válidas
- Una dirección de correo electrónico válida tendrá uno o más caracteres y, a continuación, @, seguido de uno o más caracteres y, a continuación, un . (punto), seguido de uno o más caracteres

```
CREATE TABLE my_contacts
(first_name VARCHAR2(15),
 last_name VARCHAR2(15),
 email VARCHAR2(30) CHECK(REGEXP_LIKE(email, '.+@.+\..+')));
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Expresiones Regulares en Restricciones de Control

- Definiciones de sintaxis:
  - .+ significa uno o más caracteres
  - @ un símbolo @
  - \. a . (un punto) (aquí la barra invertida es un carácter de escape)

```
CREATE TABLE my_contacts
(first_name VARCHAR2(15),
 last_name VARCHAR2(15),
 email VARCHAR2(30) CHECK(REGEXP_LIKE(email, '.+@.+\..+')));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - EXPRESIONES REGULARES
  - Metacaracteres
  - Subexpresiones

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Describir las expresiones regulares
  - Utilizar expresiones regulares para buscar, hacer coincidir y sustituir cadenas en sentencias SQL
  - Construir y ejecutar expresiones regulares y restricciones de control

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

18-1

Transacciones de Base de Datos

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección, aprenderá a:
  - Definir los términos COMMIT, ROLLBACK y SAVEPOINT y su relación con las transacciones de datos
  - Enumerar tres ventajas de las sentencias COMMIT, ROLLBACK y SAVEPOINT
  - Explicar por qué es importante, desde una perspectiva de negocio, poder controlar el flujo de procesamiento de transacciones

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- ¿Qué ocurre si un banco no tenía ningún proceso sistemático para registrar los depósitos y retiradas?
- ¿Cómo sabe si un depósito se ha abonado en su cuenta antes de que tenga que retirar dinero?
- Puede imaginar la confusión que produciría

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- Afortunadamente, los bancos controlan los procesos de transacciones para garantizar la consistencia de los datos
- En esta lección aprenderá cómo se gestiona el proceso de cambio de datos y cómo se confirman o cancelan los cambios en una base de datos
- COMMIT, ROLLBACK y SAVEPOINT no están soportados en Oracle Application Express, debido a la forma en que Oracle Application Express gestiona las conexiones a la base de datos

ORACLE Academy

DP 18-1
Transacciones de Base de Datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

5

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

Transactions

- Las transacciones son un concepto fundamental de todos los sistemas de base de datos
- Las transacciones permiten a los usuarios realizar cambios en los datos y, a continuación, decidir si desean guardar o desechar el trabajo
- Las transacciones de base agrupar varios pasos en una unidad lógica de trabajo

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Transactions

- Una transacción consiste en una de las siguientes opciones:
  - Sentencias DML que constituyen un cambio consistente de los datos
  - Las sentencias DML incluyen INSERT, UPDATE, DELETE y MERGE
  - Una sentencia DDL como CREATE, ALTER, DROP, RENAME o TRUNCATE
  - Una sentencia DCL como GRANT o REVOKE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Analogía de Transacción

- La base de datos de un banco contiene saldos de varias cuentas de cliente, así como saldos de depósito totales para otras sucursales
- Supongamos que un cliente desea retirar y transferir dinero de su cuenta y depositarlo en otra cuenta del cliente en una sucursal diferente

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Analogía de Transacción

- Hay varios pasos independientes implicados para lograr esta operación bastante sencilla
- Las dos sucursales bancarias desean asegurarse de que se producen todos los pasos de la transacción, o no se produce ninguno de ellos, y si el sistema falla, la transacción no se deja parcialmente terminada
- La agrupación de los pasos de retirada y depósito en una transacción proporciona esta garantía
- Una transacción se produce completamente o no se produce en absoluto

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Control de Transacciones

- Las transacciones se controlan mediante las siguientes sentencias:
  - COMMIT: representa el punto en el tiempo en el que el usuario ha realizado todos los cambios que quería para agruparlos lógicamente y, puesto que no se ha cometido ningún error, el usuario está listo para guardar el trabajo
    - Cuando se emite una sentencia COMMIT, la transacción actual finaliza haciendo que todos los cambios pendientes sean permanentes
  - ROLLBACK: permite al usuario desechar los cambios realizados en la base de datos
    - Cuando se emite una sentencia ROLLBACK, se desechan todos los cambios pendientes

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Control de Transacciones

- Las transacciones se controlan mediante las siguientes sentencias:
  - SAVEPOINT: crea un marcador en una transacción, que divide la transacción en varias partes más pequeñas
  - ROLLBACK TO SAVEPOINT: permite al usuario realizar un rollback de la transacción actual hasta un punto de grabación especificado
    - Si se ha cometido un error, el usuario puede emitir una sentencia ROLLBACK TO SAVEPOINT desechando solo los cambios realizados después de establecer SAVEPOINT

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejemplo de Transacción

- En el ejemplo, el usuario ha emitido una sentencia UPDATE e inmediatamente creado SAVEPOINT one

```
UPDATE copy_departments
SET manager_id= 101
WHERE department_id = 60;
```

```
SAVEPOINT one;
```

- Después de una sentencia INSERT y una sentencia UPDATE (en la siguiente diapositiva), el usuario se ha dado cuenta de que no se ha incluido una cláusula WHERE en la última UPDATE
- Para solucionar el error, el usuario ha emitido ROLLBACK TO SAVEPOINT one

ORACLE Academy

DP 18-1
Transacciones de Base de Datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

12

Consulte la diapositiva 5

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

Ejemplo de Transacción

- Los datos se han restaurado a su estado en SAVEPOINT one

```
INSERT INTO copy_departments(department_id, department_name,
  manager_id, location_id)
VALUES(130, 'Estate Management', 102, 1500);
```

```
UPDATE copy_departments
SET department_id = 140;
```

← Cláusula WHERE omitida

```
ROLLBACK TO SAVEPOINT one;
```

```
COMMIT;
```

ORACLE Academy

DP 18-1
Transacciones de Base de Datos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

13

Consulte la diapositiva 5

Cada vez que envía una transacción en Oracle Application Express, se produce una CONFIRMACIÓN implícita. Puesto que se produce esta CONFIRMACIÓN, no se puede producir un ROLLBACK.

¿Cuándo Empieza y Termina una Transacción?

- Una transacción empieza con la primera sentencia DML (INSERT, UPDATE, DELETE o MERGE)
  - Una transacción termina cuando se produce una de las siguientes situaciones:
  - Se emite una sentencia COMMIT o ROLLBACK
  - Se emite una sentencia DDL (CREATE, ALTER, DROP, RENAME o TRUNCATE)
  - Se emite una sentencia DCL (GRANT o REVOKE)
  - Un usuario sale normalmente de la utilidad de la base de datos Oracle, lo que hace que la transacción actual que se confirme de forma implícita

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿Cuándo Empieza y Termina una Transacción?

- Tras la finalización de una transacción, la siguiente sentencia SQL ejecutable inicia automáticamente la próxima transacción
- Se confirma automáticamente una nueva sentencia DDL o DCL y, por lo tanto, finaliza de forma implícita una transacción
- Todos los cambios de datos realizados durante una transacción son temporales hasta que se confirma la transacción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consistencia de los Datos

- Imagine dedicar varias horas a realizar cambios en los datos de los empleados solo para averiguar que alguien más está introduciendo información que ha entrado en conflicto con sus cambios
- Para evitar estas interrupciones o conflictos y permitir que varios usuarios accedan a la base de datos al mismo tiempo, los sistemas de base de datos utilizan una implantación automática denominada "consistencia de lectura"

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consistencia de Lectura

- La consistencia de lectura garantiza una vista consistente de los datos para todos los usuarios en todo momento
- Que los lectores no visualicen datos en proceso de cambio
- Que los escritores tengan garantizado que los cambios en la base de datos se realizarán de forma consistente
- Que los cambios realizados por un escritor no destruyan ni entren en conflicto con los cambios que esté realizando otro escritor

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consistencia de Lectura

- La consistencia de lectura es una implementación automática
- Una copia parcial de la base de datos se mantiene en segmentos de deshacer. Cuando el usuario A emite una operación de inserción, actualización o supresión en la base de datos, Oracle Server toma una instantánea (copia) de los datos antes de cambiarlos y los escribe en un segmento de deshacer (rollback)
- El usuario B sigue viendo la base de datos como existía antes de que se iniciaran los cambios; ve la instantánea de los datos del segmento de deshacer

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Consistencia de Lectura

- Antes de confirmar los cambios en la base de datos, solo el usuario que está cambiando los datos ve los cambios; todos los demás ven la instantánea en el segmento de deshacer
- Esto garantiza que los lectores de los datos vean datos consistentes en los que no se esté realizando actualmente ningún cambio

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cambios Visibles

- Cuando se confirma una sentencia DML, cualquiera que ejecute la sentencia SELECT puede ver el cambio realizado
- Si se realiza un rollback de la transacción, los cambios se deshacen:
  - La versión original anterior de los datos del segmento de deshacer se vuelve a escribir en la tabla
  - Todos los usuarios ven la base de datos como existía antes de comenzar la transacción

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMIT, ROLLBACK y SAVEPOINT

- COMMIT y ROLLBACK garantizan la consistencia de los datos, posibilitando obtener una vista previa de los cambios en los datos antes de hacer que los cambios sean permanentes y para agrupar lógicamente operaciones relacionadas
- SAVEPOINT crea un punto en una transacción al que puede realizar un rollback sin tener que deshacer toda la transacción
- COMMIT, ROLLBACK y SAVEPOINT se denominan lenguaje de control de transacciones o TCL

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMIT, ROLLBACK y SAVEPOINT

- En la transacción que se muestra en el gráfico, se ha emitido una sentencia DELETE y, a continuación, se ha establecido SAVEPOINT A
- Este SAVEPOINT actúa como un marcador que permitirá al usuario para realizar un rollback de los cambios posteriores en los datos al estado de los datos tal como existían en este punto

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMIT, ROLLBACK y SAVEPOINT

- En el ejemplo, después de SAVEPOINT A, el usuario emite una sentencia INSERT y UPDATE y, a continuación, establece otro marcador de rollback a SAVEPOINT B

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMIT, ROLLBACK y SAVEPOINT

- Si, por algún motivo el usuario no desea que se produzcan estas sentencias INSERT y/o UPDATE, puede emitir una sentencia ROLLBACK TO SAVEPOINT A
- Esta acción realizará un rollback al estado de los datos tal como se encontraba en el marcador SAVEPOINT A

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

COMMIT, ROLLBACK y SAVEPOINT

- La adición de otros SAVEPOINTS crea otros marcadores para puntos de rollback
- Si un usuario emite una sentencia ROLLBACK sin una sentencia ROLLBACK TO SAVEPOINT, se finaliza toda la transacción y se desechan todos los cambios de datos pendientes

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

SAVEPOINTS no son objetos de esquema y no se puede hacer referencia a ellos en el diccionario de datos. Cuando una transacción termina (por COMMIT o ROLLBACK) todos los SAVEPOINTS definidos en la transacción se pierden.

Procesamiento de Transacciones Implícitas

- La confirmación automática de cambios en los datos se produce en las siguientes circunstancias:
  - se emite una sentencia DDL
  - se emite una sentencia DCL
  - un usuario sale normalmente de la utilidad de la base de datos Oracle, lo que hace que la transacción actual que se confirme de forma implícita
  - se emite explícitamente sentencias COMMIT o ROLLBACK

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Procesamiento de Transacciones Implícitas

- El rollback automático se produce tras una terminación anormal de la utilizad de base de datos Oracle o cuando se produce un fallo del sistema
- Esto evita que cualquier error en los datos provoque cambios no deseados en las tablas subyacentes
- Por lo tanto, la integridad de los datos está protegida

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Bloqueo

- Es importante evitar que más de un usuario cambie los datos a la vez
- Oracle utiliza bloqueos para evitar una interacción destructiva entre transacciones que acceden al mismo recurso, tanto objetos de usuarios (como tablas o filas) como objetos del sistema no visibles para los usuarios (como estructuras de datos compartidos y filas del diccionario de datos)

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Cómo Bloquea la Base de Datos Oracle los Datos

- El bloqueo de Oracle se realiza automáticamente y no necesita ninguna acción del usuario
- El bloqueo implícito se produce para sentencias SQL según sea necesario, según la acción solicitada
- El bloqueo implícito se produce para todas las sentencias SQL excepto SELECT
- Los usuarios también pueden bloquear los datos de forma manual, lo que se denomina bloqueo explícito
- Cuando se emite una sentencia COMMIT o ROLLBACK, se liberan los bloqueos en las filas afectadas

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Transaction
  - confirmación
  - Savepoint
  - Rollback
  - Lenguaje de control de transacciones
  - Consistencia de lectura
  - Locks

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, ha aprendido lo siguiente:
  - Definir los términos COMMIT, ROLLBACK y SAVEPOINT y su relación con las transacciones de datos
  - Enumerar tres ventajas de las sentencias COMMIT, ROLLBACK y SAVEPOINT
  - Explicar por qué es importante, desde una perspectiva de negocio, poder controlar el flujo de procesamiento de transacciones

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

19-1

Pruebas

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Desarrollar y aplicar una estrategia para comprobar que una base de datos funciona tal y como se diseñó

ORACLE
Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- La mayoría de los usuarios, cuando compran un vehículo, desean saber que es fiable y no se va a romper
- Por eso, los fabricantes hacen que el vehículo pase un número de pruebas antes de ponerlo en venta
- Lo mismo se aplica a las bases de datos; antes de venderlas a un cliente, se prueban para verificar que cumplen con los requisitos de negocio

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Pruebas de Unidades

- Si se prueban dos cosas a la vez y falla la prueba, es difícil o imposible averiguar qué ha provocado el fallo
- Por lo que es importante probar solo una cosa a la vez
- Esto suele conocerse como pruebas de unidades

ORACLE Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿Qué Se Puede Probar?

- Al probar una base de datos, necesita probar gran variedad de elementos
- Por ejemplo:
  - Se debe probar que las columnas contengan el tipo de dato correcto
  - Se debe probar que las columnas pueden albergar la mayor cantidad de datos que se puedan introducir
  - Debe comprobarse que las restricciones solo restringen o limitan los datos que se supone deben restringir, ni más ni menos

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

¿Qué Se Debe Probar?

- Con frecuencia resulta poco realista probar todas las columnas y todas las restricciones de cada tabla de una base de datos si se trata de una base de datos de gran tamaño
- Deben realizarse una serie de pruebas aleatorias, que comprueben algunas columnas y algunas restricciones

ORACLE Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Diseño de las Pruebas

- Antes de realizar una prueba, debe tener una buena idea de lo que espera ver como resultado si la base de datos funciona correctamente
- Esta debe documentarse antes de realizar la prueba en una tabla similar a la que se muestra:

Número de Prueba	Fecha	Descripción de la Prueba	Entrada	Salida Esperada	Resultado/ Discrepancia	Acción
22	19-Aug-2006	Confirmar restricción NOT NULL en JOB_TITLE en la tabla JOBS	INSERT INTO jobs (job_id, job_title, min_salary, max_salary) VALUES (222,NULL,100, 200)	No se puede introducir NULL...		

ORACLE Academy

DP 19-1 Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Ejecución de las Pruebas

- Una vez que haya diseñado la prueba, puede ejecutarla y registrar sus resultados

Número de Prueba	Fecha	Descripción de la Prueba	Entrada	Salida Esperada	Resultado/ Discrepancia	Acción
22	19-Aug-2006	Confirmar restricción NOT NULL en JOB_TITLE en la tabla JOBS	INSERT INTO jobs (job_id, job_title, min_salary, max_salary) VALUES (222,NULL,100, 200)	No se puede introducir NULL...	No se puede introducir NULL...	Ninguna

ORACLE Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Pruebas
  - Pruebas de unidades

ORACLE
Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Desarrollar y aplicar una estrategia para comprobar que una base de datos funciona tal y como se diseñó

ORACLE Academy

DP 19-1
Prueba

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy

ORACLE

Academy

Database Programming with SQL

19-3

Revisión del Examen Final

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Revisar los puntos clave sobre la manipulación de mayúsculas/minúsculas y caracteres
  - Revisar las funciones generales y de número, fecha, conversión
  - Revisar las expresiones condicionales
  - Revisar el producto cartesiano y las operaciones de unión
  - Revisar las uniones no igualitarias, uniones externas, autouniones, uniones cruzadas, uniones naturales y cláusulas de unión
  - Revisar las funciones de grupo, la sintaxis group by y las cláusulas having

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Escribir subconsultas de una y varias filas
  - Revisar las subconsultas pareadas y no pareadas
  - Revisar las subconsultas correlacionadas
  - Revisar las sentencias DML insert, update, delete, merge y las inserciones en varias tablas
  - Revisar las sentencias DDL CREATE, ALTER, RENAME, TRUNCATE, FLASHBACK TABLE, DROP y FLASHBACK QUERY
  - Revisar las sentencias DCL CREATE y los privilegios de objetos REVOKE

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Objetivo

- La revisión es la mejor preparación para la evaluación
- La evaluación le permite observar todo lo que ha aprendido y resalta las áreas en las que puede que desee mejorar
- La revisión de los temas aprendidos hasta este punto le ayudará a ser el mejor durante el examen final

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Revisión de la Sintaxis

- Se trata de una revisión de la sintaxis
- Asegúrese de que también revisa las reglas relativas a la sintaxis
- Estas se tratan con profundidad a lo largo del curso

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Manipulación de Mayúsculas/ Minúsculas y de Caracteres

- Mayúsculas/minúsculas

```
LOWER(column name|expression)
UPPER(column name|expression)
INITCAP(column name|expression)
```

- Caracteres

```
CONCAT(column name|expression, column name|expression)
SUBSTR(column name|expression,n,m)
LENGTH(column name|expression)
INSTR(column name|expression, string literal)
LPAD (column name|expression, n, character literal)
RPAD(column name|expression, n, character literal)
TRIM ( [leading | trailing | both]  char1 FROM char2)
REPLACE (column name|expression, string to be replaced,
      replacement string)
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones Numéricas

```
ROUND(column|expression,n)
TRUNC(column|expression,n)
MOD(column|expression, column|expression)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones de Fecha

```
ROUND(column|expression,string)
TRUNC(column|expression,string)

MONTHS_BETWEEN(column|expression, column|expression)
ADD_MONTHS(column|expression,n)
NEXT_DAY(column|expression,'day')
LAST_DAY(column|expression)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones de Conversión

```
TO_CHAR(number, 'format model')
TO_CHAR(date, 'format model')
TO_NUMBER(character string, 'format model')
TO_DATE(character string, 'format model')
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones NULL

```
NVL(column|expression, value)
```

```
NVL2(column|expression, column|expression,
            column|expression)
```

```
NULLIF(column|expression, column|expression)
```

```
COALESCE(column|expression, column|expression,
            column|expression…. column|expression)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Expresiones Condicionales

- Específico de Oracle

```
DECODE(columnl|expression, search1, result1
         [, search2, result2,...,]
         [, default])
```

- ANSI

```
CASE expr WHEN comparison_expr1 THEN return_expr1
         [WHEN comparison_expr2 THEN return_expr2
          WHEN comparison_exprn THEN return_exprn
          ELSE else_expr]
END
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis Estándar ANSI SQL

- Cross Join

```
SELECT last_name, department_name
FROM employees CROSS JOIN departments;
```

- Natural Join

```
SELECT employee_id, last_name, department_name
FROM employees NATURAL JOIN departments;
```

- Join .. On

```
SELECT e.employee_id, e.last_name, e.salary, j.grade_level
FROM employees e JOIN job_grades j
ON (e.salary BETWEEN j.lowest_sal AND j.highest_sal);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis Estándar ANSI SQL

- Joins .. Using

```
SELECT employee_id, last_name, department_name
FROM employees JOIN departments
USING (department_id);
```

- Join .. On

```
SELECT e.employee_id, e.last_name, d.department_id,
d.location_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis Estándar ANSI SQL

- Uniones Externas
- Right Outer Join

```
SELECT e.employee_id, e.last_name, e.department_id,
d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

- Left Outer Join

```
SELECT e.employee_id, e.last_name, e.department_id,
d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sintaxis Estándar ANSI SQL

- Uniones Externas
- Full Outer Join (no es comparable con la unión específica de Oracle)

```
SELECT e.employee_id, e.last_name, e.department_id,
d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id = d.department_id);
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Funciones de Grupo, Sintaxis de Group By y Cláusulas HAVING

```
AVG (column |expression)
COUNT (column |expression)
MIN (column |expression)
MAX (column |expression)
SUM (column |expression)
VARIANCE (column |expression)
STDDEV (column |expression)
```

```
SELECT column1, AVG (column |expression)
FROM table 1
GROUP BY (ROLLUP | CUBE)  (column1 | GROUPING SETS)
HAVING AVG (column |expression)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Subconsultas de Una o Varias Filas

```
SELECT column1..
FROM table 1
WHERE column2 = (SELECT column2
                 FROM table 1
                 WHERE column 3 = expression)
```

- Operadores de Una Sola Fila: = =,>,<,>=,<=,<>
- Operadores de Varias Filas: IN, ANY, ALL

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Subconsultas Pareadas y No Pareadas

- Pareadas

```
SELECT column1..
FROM table 1
WHERE (column2, column3) = (SELECT column2, column3
                            FROM table 1
                            WHERE column 4 = expression);
```

- No pareadas

```
SELECT column1..
FROM table 1
WHERE column2 = (SELECT column2
                 FROM table 1
                 WHERE column 4 = expression)
AND   column3 = (SELECT column3
              FROM table 2
              WHERE column 4 = expression);
```

ORACLE Academy

DP 19-3
Revisión del Examen Final

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Subconsultas Correlacionadas

```
SELECT o.column1..
FROM table_1 o
WHERE o.column2 = (SELECT i.column2
                   FROM table_2 i
                   WHERE i.column1 = o.column1)
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción, Actualización y Supresión de Datos

- INSERT explícito

```
INSERT INTO table (column1, column2…)
VALUES (value1, value2…) ;
```

- INSERT implícito

```
INSERT INTO table
VALUES (value1, value2, value3, value4);
```

```
UPDATE table1
SET column1 = value1,
         column2 = value2…
WHERE column1 = value;
```

```
DELETE FROM table1
WHERE column1 = value;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción, Actualización y Supresión de Datos

```
UPDATE table1
SET column1 = value1,
         column2 = value2…
WHERE column1 = value;
```

```
DELETE FROM table1
WHERE column1 = value;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Inserción, Actualización y Supresión de Datos

```
conditional_insert_clause
[ ALL | FIRST ]
WHEN condition THEN
    insert_into_clause [ values_clause ]
WHEN condition THEN
    insert_into_clause [ values_clause ]
ELSE insert_into_clause [ values_clause ]
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Valores por Defecto

```
CREATE TABLE table1 (
column1           DATE DEFAULT SYSDATE,…)

INSERT INTO table1
   (column1,….)
VALUES
   (DEFAULT,…);
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Sentencia Merge

- Inserción en varias tablas

```
MERGE INTO destination-table USING source-table
ON matching-condition
WHEN MATCHED THEN UPDATE
SET ……
WHEN NOT MATCHED THEN INSERT
VALUES (……);
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Tablas

```
CREATE TABLE table
(column data type [DEFAULT expression],
column data type [DEFAULT expression],
……[ ] );
```

```
CREATE TABLE tablename
[(column, column, …)]
AS subquery;
```

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Especificación de Tipos de Dato

```
NUMBER(p,s)
CHAR
VARCHAR2(n)
DATE
TIMESTAMP
TIMESTAMP WITH TIMEZONE
TIMESTAMP WITH LOCAL TIME ZONE
INTERVAL YEAR TO MONTH
INTERVAL DAY TO SECOND
CLOB
BLOB
RAW
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Modificación de una Tabla

```
ALTER TABLE tablename
ADD (column_name data type [DEFAULT expression]…);
```

```
ALTER TABLE tablename MODIFY (column_name VARCHAR2(30));
```

```
ALTER TABLE tablename DROP COLUMN column name;
```

```
ALTER TABLE tablename SET UNUSED (column name);
```

```
ALTER TABLE tablename DROP UNUSED COLUMNS;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Modificación de una Tabla

- Inserción en varias tablas

```
DROP TABLE tablename;
```

```
FLASHBACK TABLE tablename TO BEFORE DROP;
```

```
SELECT * FROM user_recyclebin;
```

```
SELECT versions_starttime "START_DATE",
       versions_endtime   "END_DATE",
       column, column......
FROM   table
   VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE  column = value;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones a Nivel de Columna

```
CREATE TABLE table
(col1 data type CONSTRAINT tab_col1_pk PRIMARY KEY,
 col2 data type CONSTRAINT tab_col2_nn NOT NULL,
 col3 data type CONSTRAINT tab_col3_uk UNIQUE,
 col4 data type CONSTRAINT tab_col4_ck CHECK (col4 > value),
 col5 data type CONSTRAINT tab_col5 REFERENCES table2 (col1));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Restricciones a Nivel de Tabla

```
CREATE TABLE table
(col1 data type,
 col2 data type,
 col3 data type,
 col4 data type,
 col5 data type,
CONSTRAINT tab_col1_pk PRIMARY(col1),
CONSTRAINT tab_col3_uk UNIQUE(col2),
CONSTRAINT tab_col4_ck CHECK (col4 > value),
CONSTRAINT tab1_col5_fk FOREIGN KEY (col5) REFERENCES table2 (col1));
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación y Gestión de Vistas

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

```
DROP VIEW viewname;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Análisis de N Principales

```
SELECT ROWNUM as RANK, col1, col2
FROM (SELECT col1, col2 FROM table1
ORDER BY col1)
WHERE ROWNUM <= n;
```

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Vistas en Línea

```
SELECT t1.col1, t2.col2…
FROM table 1 t1, (SELECT col1, col2..
                  FROM table2
                  WHERE …) t2
WHERE ……;
```

ORACLE
Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Secuencias

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

```
DROP SEQUENCE sequence_name;
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación de Índices y Sinónimos

```
CREATE INDEX index_name
ON  table_name( column...,column);
```

```
DROP INDEX index_name;
```

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

```
DROP [PUBLIC] SYNONYM name_of_synonym
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación y Revocación de Privilegios de Objeto

```
CREATE USER user
IDENTIFIED BY   password;
```

```
GRANT privilege [, privilege...]
TO user [, user| role, PUBLIC...];
```

```
ALTER USER user
IDENTIFIED BY password;
```

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Creación y Revocación de Privilegios de Objeto

```
CREATE ROLE role_name;
```

```
GRANT object_priv [(column_list)]
ON object_name
TO {user|role|PUBLIC}
[WITH GRANT OPTION];
```

```
REVOKE {privilege [, privilege...]|ALL}
ON  object
FROM   {user[, user...]|role|PUBLIC}
[CASCADE CONSTRAINTS];
```


Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber revisado lo siguiente:
  - Puntos clave sobre la manipulación de mayúsculas/minúsculas y caracteres
  - Funciones generales y de número, fecha, conversión
  - Expresiones condicionales
  - Producto cartesiano y las operaciones de unión
  - Uniones no igualitarias, uniones externas, autouniones, uniones cruzadas, uniones naturales y cláusulas de unión
  - Funciones de grupo, sintaxis de GROUP BY y Cláusulas HAVING

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

Resumen

- En esta lección, debe haber revisado lo siguiente:
  - Subconsultas de una o varias filas
  - Subconsultas pareadas y no pareadas
  - Subconsultas correlacionadas
  - Sentencias DML insert, update, delete, merge y las inserciones en varias tablas
  - Sentencias DDL, FLASHBACK TABLE, DROP y FLASHBACK QUERY

ORACLE Academy

Copyright © 2020, Oracle y/o sus filiales. Todos los derechos reservados.

ORACLE

Academy