



 **ACADEMY**

# Programación de bases de datos con SQL

16-2

Índices y Sinónimos



**ORACLE** ACADEMY

Copyright © 2017, 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

# 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.

# 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. (Suenan como alias, ¿no?)

# 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.

# 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.

# Índices

- Los índices son lógicos 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	Canadá	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
...	...	...	...



# 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	Canadá	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
...	...	...	...

# 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.

# 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.

# 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);
```

# 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

# 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.

# 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

# 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);
```



# 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.

# 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).

# 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

# 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.

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';
```

# 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.

# 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%';
```

# 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.



# 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.

# 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.

# 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.

# 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.

# 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;
```

# 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.

# 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.

# 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.



# 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;
```

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 ;
```

# 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.

# 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

# 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



 **ACADEMY**