

Bases de Datos y Sistemas de Información

Grado en Ingeniería Informática

**Práctica 2: Lenguaje de definición de SQL y estudio del
SGBD ORACLE: gestión de transacciones.**

Curso 2021/2022



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Tabla de contenido

1. Introducción	3
2. Definición de datos en el lenguaje SQL del sistema ORACLE.	3
2.1. Definición de relación básica	3
2.2. Modificación de relaciones.....	5
2.3. Definición de vistas	5
2.4. Definición de privilegios	6
3. Manipulación de datos en el lenguaje SQL del sistema ORACLE	6
4. Instrucciones para consultar los objetos del esquema en ORACLE	6
Ejercicios:.....	7
5. Concepto de transacción.....	9
6. Gestión de transacciones en SQL	9
7. Gestión de transacciones en ORACLE.....	10
Ejercicios.....	10

1. Introducción

El lenguaje de definición de datos del SQL estándar se puede encontrar en el capítulo 4 del libro de texto. En este boletín se presenta sólo el lenguaje de definición del SQL de ORACLE, insistiendo en sus diferencias respecto al estándar. Asimismo, se presentan algunas particularidades del lenguaje de manipulación de ORACLE.

2. Definición de datos en el lenguaje SQL del sistema ORACLE.

En el sistema ORACLE no existe el concepto de esquema de base de datos tal como aparece en el lenguaje SQL estándar. Asociado a cada usuario definido en el sistema se crea una base de datos en la que se almacenarán todos los objetos (tablas, vistas, procedimientos,...) creados por él.

De todos los elementos que pueden incluirse en un esquema SQL estándar, en el sistema ORACLE pueden definirse los siguientes: relaciones básicas, vistas, y privilegios de acceso (no se pueden definir dominios). Además, pueden definirse algunos detalles relativos a la representación interna de los datos (índices, *tablespace*, *cluster*...); disparadores para modelar comportamiento activo; así como elementos de programación (funciones y procedimientos, paquetes de procedimientos,...).

2.1. Definición de relación básica

```
Definición_relación_básica ::= CREATE TABLE nom_relación
                               (elemento_relación1, elemento_relación2, ... )
```

```
elemento_relación ::= definición_atributo
                   | restricción_relación
```

```
definición_atributo ::= nom_atributo tipo_datos
                      [DEFAULT (expresión)]
                      [restricción_atributo1 restricción_atributo2 ...]
```

```
tipo_datos ::= CHAR (longitud)
            | VARCHAR2 (longitud)
            | NUMBER [(precisión[, escala])]
            | DATE
```

```
restricción_atributo ::= [CONSTRAINT nom_restricción]
                        {[NOT] NULL
                        | UNIQUE
                        | PRIMARY KEY
                        | REFERENCES nom_relación* [(nom_atributo*)]
                        [ON DELETE {CASCADE|SET NULL}]}
                        | CHECK (condición)}
                        [cuándo_comprobar]
```

```

restricción_relación ::=
[CONSTRAINT nom_restricción]
  { UNIQUE (nom_atributo1, nom_atributo2, ...)
  | PRIMARY KEY (nom_atributo1, nom_atributo2, ...)
  | FOREIGN KEY (nom_atributo1, nom_atributo2, ...)
    REFERENCES nom_relación [(nom_atributo1, nom_atributo2, ...)]
    [ON DELETE {CASCADE|SET NULL}]

  | CHECK (condición)}
[cuándo_comprobar]

cuándo_comprobar ::=
[NOT] DEFERRABLE [INITIALLY {IMMEDIATE | DEFERRED}]

```

Los tipos de datos disponibles son:

Numéricos:

- NUMBER [(precisión[, escala])] donde precisión es el número total de dígitos y escala el número de cifras decimales
- enteros: NUMBER (precisión)
- reales: NUMBER (precisión, escala)

Alfanuméricos:

- de longitud fija: CHAR (longitud)
- de longitud variable: VARCHAR (longitud)

Fechas:

- DATE

La expresión de la cláusula DEFAULT se construye a partir de constantes de los tipos de datos predefinidos, operadores y funciones del sistema siguiendo la sintaxis adecuada en cada caso. En la expresión no se puede hacer referencia a otros atributos de la relación. El tipo de la expresión debe coincidir con el tipo de datos del atributo en el que se incluye la cláusula.

En ORACLE sólo se contempla el tipo de integridad referencial *débil* y la posibilidad de incluir la directriz de restauración de la integridad referencial *borrado en cascada o a nulos*.

La condición de la cláusula CHECK es una expresión lógica que se define con la misma sintaxis que la condición de la cláusula WHERE de la sentencia SELECT con las siguientes limitaciones: sólo se puede hacer referencia a atributos de la relación sobre la que se define la restricción y no se pueden incluir subconsultas ni funciones agregadas. La relación satisface la restricción de integridad definida con la cláusula CHECK si para todas las tuplas la condición se evalúa a cierto o a indefinido (debido a la presencia de valores nulos).

La directriz *cuándo_comprobar* tiene el mismo significado que en el SQL estándar.

En la definición de una relación básica en ORACLE se pueden incluir otras cláusulas que no aparecen en la sintaxis del SQL estándar con las que se definen detalles de representación física de la relación.

Para borrar una relación se utiliza la sentencia DROP TABLE *nom_relación*.

2.2. Modificación de relaciones

```
modificación_relación ::= ALTER TABLE nom_relación
                        { ADD (elemento_relación1, elemento_relación2, ...)
                          | MODIFY (definición_atributo1, definición_atributo2, ...)
                          | DROP COLUMN (nom_atributo1 , nom_atributo2, ...)
                            [CASCADE]
                          | { DROP
                            | [VALIDATE | NOVALIDATE] ENABLE
                            | DISABLE } (restricción) }
```

```
restricción ::= { PRIMARY [CASCADE]
                 | UNIQUE (nom_atributo1, nom_atributo2, ...) [CASCADE]
                 | CONSTRAINT nom_restricción }
```

La opción ADD permite añadir nuevos atributos o restricciones a una relación.

La opción MODIFY permite modificar la definición de los atributos.

La opción ENABLE (resp. DISABLE) permite activar (resp. desactivar) una restricción de integridad. Con la opción VALIDATE (opción por defecto), el sistema se asegura que al activar una restricción los datos almacenados en la base de datos en el momento de la activación satisfacen la restricción.

La opción DROP permite borrar atributos o restricciones de integridad. La opción CASCADE borra en cascada cualquier restricción de integridad definida en el esquema que dependa del elemento que se acaba de borrar.

2.3. Definición de vistas

```
definición_vista ::= CREATE [OR REPLACE] VIEW nom_vista
                  [(nom_atributo1, nom_atributo2, ...)] AS sentencia_SELECT
                  [WITH CHECK OPTION]
```

Debido a que una vista puede definirse a través de cualquier sentencia SELECT, la traducción de una operación de actualización sobre la vista en actualizaciones sobre relaciones básicas no es siempre posible ya que pueden presentarse ambigüedades, por este motivo la actualización de vistas está sometida a ciertas restricciones que evitan esta posible ambigüedad.

Estas restricciones son en ORACLE:

- Una vista definida por una SELECT que contiene operadores conjuntistas (UNION, INTERSECT,...), el operador DISTINCT, funciones agregadas (SUM, AVG, ...) o la cláusula GROUP BY no es actualizable.
- Si la vista está definida sobre una única relación básica el sistema traducirá la actualización sobre la vista en una operación de actualización sobre la relación básica siempre que no se viole ninguna restricción de integridad definida sobre dicha relación.
- Si la vista está definida sobre una concatenación de relaciones, la actualización está sometida a las siguientes restricciones adicionales:
 - La actualización sólo puede modificar una de las relaciones básicas.
 - La actualización modificará la relación básica que cumpla la propiedad de *conservación de la clave*, es decir aquella relación tal que su clave primaria podría ser también clave de la vista si sus atributos fuesen seleccionados por la SELECT que define la vista.

- La actualización no se realizará si viola alguna de las restricciones definidas sobre la relación básica que se va a actualizar.

2.4. Definición de privilegios

El propietario del esquema de una base de datos es el propietario de todos los objetos definidos en él. Para que otro usuario pueda realizar operaciones sobre los objetos de la base de datos debe tener la autorización necesaria; estas autorizaciones deben concederlas el propietario del objeto por medio de la sentencia GRANT.

```
definición_operación_grant ::= GRANT privilegio_sistema1, privilegio_sistema2,...  
                                ON objeto  
                                TO {PUBLIC | usuario1, usuario2, ...}  
                                [WITH ADMIN OPTION]
```

- *usuario* es el identificador de un usuario.
- Con la cláusula PUBLIC se transmiten los privilegios a todos los usuarios.
- La cláusula WITH ADMIN OPTION concede permiso para ceder a terceros los privilegios obtenidos.

Los privilegios que se pueden otorgar son privilegios de administrador, es decir crear, borrar o modificar elementos de los esquemas: tablas, índices, vistas, etc.

3. Manipulación de datos en el lenguaje SQL del sistema ORACLE

Las sentencias de manipulación de datos del lenguaje SQL del sistema ORACLE son las mismas que las del lenguaje SQL estándar que ya han sido presentadas, con las siguientes excepciones relativas a la combinación de tablas con operadores conjuntistas.

- El operador de SQL **EXCEPT** se denomina en ORACLE **MINUS** (con la misma sintaxis).
- Salvo en **UNION ALL**, todos los operadores eliminan duplicados.

4. Instrucciones para consultar los objetos del esquema en ORACLE

- **Tablas que hay definidas:** SELECT * FROM {USER_TABLES| ALL_TABLES};
- **Restricciones que hay definidas:**
SELECT * FROM {USER_CONSTRAINTS | ALL_CONSTRAINTS};
- **Atributos de las tablas:** DESCRIBE tabla;
- **Ver errores de compilación de disparadores:** SHOW ERRORS nombre;

Ejercicios:

Se desea diseñar una base de datos para la gestión de una pequeña biblioteca de un departamento. Después de realizar el análisis del sistema, se han identificado los requerimientos que van a realizarse con más frecuencia; éstos son:

- Consultar los datos de un libro: código del libro, título, autores, temática y en caso de estar prestado, el socio que lo tiene actualmente en préstamo.
- Consultar la información sobre un socio: código del socio, nombre, dirección, teléfono y libros que actualmente tiene en préstamo, así como la fecha de préstamo.
- Consultar los préstamos históricos de un socio: código del libro, fecha del préstamo y fecha de la devolución.
- Dar de alta, dar de baja y modificar los datos de un socio.
- Gestionar los préstamos: prestar un libro a un socio y registrar la devolución de un libro.

Algunas restricciones de integridad que se han detectado son:

- El código del libro identifica unívocamente al libro.
- El código del socio identifica unívocamente al socio.
- El conjunto de temas utilizados para clasificar un libro es: *física, electricidad, mecánica y óptica*.
- Un libro solo se puede prestar una sola vez en un día
- La fecha de devolución de un libro debe ser posterior a la fecha de préstamo.

A partir de estos requerimientos se ha diseñado el siguiente esquema relacional.

```

SOCIO(scod:char(5), nombre:varchar2(60), direccion:varchar2(50),
      tel:varchar2(20))
      CP: {scod}

LIBRO(lcod:char(5), titulo:varchar(100),tematica:varchar(15))
      CP: {lcod}
      RI1: tematica es ('física','electricidad','mecánica','óptica')

AUTORES(lcod:char(5), autor:varchar2(40) )
      CP:{lcod, autor}
      CA:{lcod} -> libro(lcod)

PRESTAMO(scod:char(5), lcod:char(5), fecha_pre:date, fecha_dev:date)
      CP:{lcod, fecha_pre}
      VNN:{scod}
      CA:{scod} -> socio(scod)
      CA:{lcod} -> libro(lcod)
      RI2: fecha_dev is null or fecha_dev>fecha_pre

```

- Definir la base de datos en el sistema ORACLE utilizando el lenguaje SQL de definición. Todas las restricciones deben ser diferibles.
- Poblar la base de datos.

5. Concepto de transacción

Una transacción es una secuencia de instrucciones de manipulación de la base de datos que constituye una unidad lógica de ejecución. Una transacción debe satisfacer las propiedades de *atomicidad*, *consistencia*, *aislamiento* y *persistencia*. El mantenimiento de cada una de estas propiedades es responsabilidad de un módulo del SGBD, la atomicidad y la persistencia son responsabilidad del módulo de recuperación, el aislamiento es responsabilidad del módulo de control de la concurrencia y la consistencia es responsabilidad del módulo de comprobación de la integridad y del módulo de recuperación.

6. Gestión de transacciones en SQL

El lenguaje SQL ofrece las siguientes instrucciones para el manejo de transacciones:

- Inicio de una transacción: no existe ninguna instrucción para definir el inicio de una transacción.
- Fin de una transacción (con confirmación): COMMIT [WORK] (transacción confirmada parcialmente).
- Fin de una transacción (con anulación): ROLLBACK [WORK] (transacción anulada).

Es importante recordar que la finalización (confirmación parcial) de una transacción no significa su confirmación definitiva, ya que ésta puede estar condicionada a la comprobación posterior de restricciones de integridad. El SGBD tiene la capacidad de confirmar definitivamente o anular una transacción que el usuario ha finalizado con confirmación; y tiene la capacidad de interrumpir, anulándola, una transacción que todavía no había sido finalizada por el usuario. Para velar por el mantenimiento de las buenas propiedades de las transacciones el SGBD debe asegurar que los cambios realizados por una transacción confirmada (de forma definitiva) queden grabados permanentemente en la base de datos y que los cambios realizados por una transacción anulada o interrumpida que ya hayan sido grabados sean deshechos.

Una de las propiedades importantes de las transacciones es la consistencia, es decir la transacción debe conducir la base de datos a un estado consistente en el que se cumplan todas las restricciones de integridad, por ello es importante conocer la estrategia seguida por el sistema para realizar esta comprobación.

La existencia del concepto de transacción y su carácter de unidad lógica de ejecución, plantea dos posibles estrategias para la comprobación de la integridad. Una restricción se puede comprobar en modo inmediato (IMMEDIATE) después de la ejecución de cada instrucción SQL relevante para la restricción, o en modo diferido (DEFERRED) después de la finalización de cualquier transacción que incluya una instrucción SQL relevante para la restricción. El modo de una restricción se puede cambiar localmente para una transacción si la restricción se ha definido como diferible (DEFERRABLE); una restricción es diferible si tiene la propiedad de poder diferirse su comprobación al final de la transacción. El modo de comprobación de una restricción se define en el esquema de la base de datos con la cláusula:

```
[[[NOT] DEFERRABLE] [INITIALLY {IMMEDIATE | DEFERRED}]]
```

Y la instrucción SQL que permite cambiar, localmente en una transacción, el modo de una restricción definida como diferible, es:

```
SET CONSTRAINT { restricción1, ..., restricciónn | ALL } {IMMEDIATE | DEFERRED}
```

7. Gestión de transacciones en ORACLE

En el uso interactivo del sistema ORACLE una transacción se inicia con la primera instrucción SQL¹ ejecutada por el usuario desde que finalizó la última transacción o desde el inicio de la sesión y termina cuando el usuario la finaliza explícitamente con la sentencia COMMIT [WORK] (transacción confirmada parcialmente) o la anula explícitamente con la sentencia ROLLBACK [WORK] (transacción anulada) o bien cuando el sistema la finaliza implícitamente debido al cierre de la sesión (transacción confirmada parcialmente) o la anula implícitamente debido a la ocurrencia de un error (transacción anulada). Es decir, en ORACLE la operación que inicia una transacción es implícita y la operación que termina una transacción puede ser explícita o implícita.

Respecto a la estrategia de comprobación de las restricciones en ORACLE, se sigue la propuesta del SQL estándar con el siguiente comportamiento respecto a la violación de una restricción: “si durante la ejecución de una transacción se viola una restricción con modo inmediato el sistema deshace tan solo el efecto de la operación SQL que ha causado la violación de la restricción (*statement rollback*) y la transacción puede continuar; si al finalizar una transacción se viola una restricción con modo diferido el sistema anula la transacción y deshace su efecto global (*transaction rollback*)”.

Ejercicios

Considere la base de datos que ha diseñado en la práctica anterior y cuyo esquema se encuentra en el anexo para la gestión de una biblioteca y realice los siguientes ejercicios.

- En el sistema ORACLE no existe la directriz actualización en cascada para la restauración de la integridad referencial. ¿Cómo podría modificar el código de un socio que ha tenido préstamos, sin violar la integridad referencial de la relación de préstamos históricos? (necesidad del concepto de transacción).
- Diseñe una transacción que realice inserciones de tuplas sobre la relación Socios según el siguiente esquema de transacción:

```
COMMIT;  
INSERT INTO socio VALUES ('s1', ..., ..., ...);  
INSERT INTO socio VALUES ('s2', ..., ..., ...);  
INSERT INTO socio VALUES ('s1', ..., ..., ...);  
INSERT INTO socio VALUES ('s3', ..., ..., ...);  
COMMIT;
```

(donde s1, s2 y s3 son códigos de socio que no haya usado hasta el momento de ejecutar la transacción y los puntos suspensivos representan valores cualesquiera para los atributos *nombre*, *dirección* y *teléfono*).

Esta transacción viola la restricción de clave primaria para la relación *Socio*.

Ejecute dos veces el anterior esquema de transacción, una vez con la restricción de clave primaria para la relación socio en modo inmediato; borre las tuplas insertadas y vuelva a realizarlo con la restricción en modo diferido. ¿Qué diferencias observa en ambos casos? (atomicidad y consistencia).

¹ Se supone que sólo se utilizan sentencias de DML.

- c) Inicie desde su PC dos sesiones distintas sobre la misma base de datos y en cada sesión realice las siguientes transacciones (los t_i indican el orden en que se deben realizar las operaciones):

Sesión 1	Sesión 2
t_0 “consultar el número total de socios”	
	t_1 “consultar el número total de socios”
t_2 “insertar un nuevo socio”	
t_3 “consultar el número total de socios”	
	t_4 “consultar el número total de socios”
t_5 “confirmar la transacción”	
	t_6 “consultar el número total de socios”
	t_7 “insertar un nuevo socio”
	t_8 “consultar el número total de socios”
	t_9 “anular la transacción”
t_{10} “consultar el número total de socios”	
	t_{11} “consultar el número total de socios”

¿Cómo se interpretan los resultados de las operaciones de consulta t_3 , t_4 , t_6 y t_8 ? ¿y el resultado de las consultas finales (t_{10} y t_{11})? (aislamiento y persistencia).

¿Con las observaciones realizadas puedes afirmar que el sistema mantiene la propiedad de persistencia de una transacción?