**TRANSACCIONES Y SEGURIDAD**

1. **Transacciones**
2. ¿Cómo se define el comienzo y fin de una transacción en ORACLE?

Las sentencias de inicilización de transacción:

* Oracle es un sistema de base de datos puramente transaccional, de tal forma, que la instrucción BEGIN TRANSACTION no existe.
* Se inicia después del BEGIN.

Las sentencias de finalización de transacción son:

* COMMIT: La transacción termina correctamente, se vuelcan los datos al tablespace original y se vacía el RBS.
* ROLLBACK: Se rechaza la transacción y se vacía el RBS. Cualquier cambio realizado desde que se inició la transacción se deshace, quedando la base de datos en el mismo estado que antes de iniciarse la transacción.

1. ¿Cuáles son los diferentes tipos de aislamiento que soporta ORACLE? Para cada uno de ellos detalle, ¿cómo maneja los bloqueos? ¿qué problemas resuelve?

Oracle proporciona tres niveles de aislamiento:

* read-committed: Una consulta solo utiliza datos que fueron confirmados (Commited Data) antes del comienzo de la ejecución de la consulta.

Con este nivel de aislamiento se evita el fenómeno de la lectura sucia, porque los cambios no confirmados no son visibles para cualquier otra transacción, hasta que se confirme el cambio. Dentro de este nivel de aislamiento, cada SELECT utiliza su propia instantánea de los datos que se confirmó (commit) antes de la ejecución de la instrucción SELECT. Ahora, ya que cada SELECT tiene su propia instantánea, por lo que el mismo SELECT cuando se ejecuta varias veces durante la misma transacción podría regresar diferentes conjuntos de resultados. Este fenómeno se le llama lectura no repetible.

* serializable transactions: Cada consulta utiliza los datos que fueron confirmados antes del comienzo de la ejecución de la transacción, además de acceder también a los cambios realizados por sentencias INSERTE, DELETE, UPDATE que hayan sido ejecutadas dentro de esta transacción.

Con este nivel de aislamiento se evita el fenómeno de fantasma. Coloca un bloqueo de rango en el conjunto de datos, cuando las transacciones se ejecuta en este nivel de aislamiento se bloquean todos los registros y recursos que se tiene acceso, así bloquea todo cambio, impidiendo a otros usuarios actualizar o insertar filas en el conjunto de datos hasta que la transacción se ha completado. Este nivel de aislamiento es el más fuerte posible.

* read-only: Solo es visible la versión de los datos al momento del comienzo de la transacción y no se permiten sentencias INSERT, UPDATE, DELETE dentro de la transacción.

Bloqueos:

* X (Exclusivo): Consiste en bloquear toda la tabla, impidiendo la ejecución de sentencias DML sobre ella.
* S (Compartido): Se marca la tabla como “Sólo Lectura”. Varias transacciones pueden adquirir este bloqueo sobre una misma tabla.
* RS (Filas Compartidas): Este bloqueo se utiliza para indicar que se va a realizar en un futuro cercano alguna modificación a las filas de esta tabla.
* RX (Filas Exclusivas): Con este bloqueo se indica que ya se realizaron modificaciones INSERT, UPDATE o DELETE a las filas de esta tabla.
* SRX (Filas Compartidas Exclusivas): Solo una transacción puede obtener este bloqueo, ya que permite prohibirle a todas las demás la modificación de la fila, pero se permite la lectura.

1. **Seguridad**
2. El mecanismo de control de acceso discrecional, ¿cómo se define en ORACLE?

Explicite acciones, objetos y personas autorizadas.

El control de acceso discrecional (DAC) es un mecanismo de software para controlar el acceso de usuarios a archivos y directorios. DAC deja que la configuración de protecciones para archivos y directorios las realice el propietario según su criterio. Las dos formas de DAC son los bits de permisos UNIX y las listas de control de acceso (ACL).

Los bits de permisos permiten que el propietario establezca protección de lectura, escritura y ejecución por propietario, grupo y otros usuarios. En sistemas UNIX tradicionales, el superusuario o usuario robot puede sustituir la protección de DAC. Con Trusted Extensions, única los administradores y los usuarios autorizados tienen la capacidad de sustituir DAC. Las ACL proporcionan una granularidad de control de acceso más específica. Las ACL permiten que los propietarios establezcan permisos independientes para usuarios y grupos específicos.

1. **Vistas**
2. ¿Para qué sirve una vista?

* Realizar consultas complejas más facilmente, ya que permiten dividir la consulta en varias partes.
* Propocionar tablas con datos completos.
* Utilizar visiones especiales con los datos.
* Ser utilizadas como tablas que resumen todos los datos.
* Ser utilizadas como cursores de datos en los lenguajes procedimentales (como PL/SQL).

1. ¿Cuáles son los mecanismos para la creación y borrado de vistas en ORACLE?

Creación:

CREATE [OR REPLACE] [FORCE|NOFORCE]

VIEW nombre\_vista AS subconsulta

[WITH CHECK OPTION [CONSTRAINT nombre\_constraint]]

[WITH READ ONLY];

Borrado:

DROP VIEW nombre\_vista;

1. ¿Cuáles son las restricciones de las vistas en ORACLE?

* Se pueden realizar operaciones DML sobre vistas simples
* No se puede eliminar una fila si la vista contiene Funciones de grupo, una cláusula GROUP BY o el comando DISTINCT
* No es posible modificar los datos en la vista si contiene cualquiera de las condiciones anteriores, columnas definidas por expresiones o la pseudocolumna ROWNUM
* No se puede agregar datos si la vista contiene cualquiera de las condiciones anteriores o cualquier columna NOT NULL no incluida por la vista (tabla base)
* FORCE: Crea la vista sin importa que la tabla base exista o no
* WITH CHECK OPTION: Especifica que solamente las filas accesibles a la vista pueden sert insertadas o actualizadas
* CONSTRAINT: Nombre asignado a la restricción CHECK OPTION
* WITH READ ONLY: Asegura que ninguna operación DML pueda realizarse sobre esta vista

1. **Modularidad Paquetes**
2. ¿Para qué sirve un paquete?

Un paquete es un objeto PL/SQL que agrupa lógicamente otros objetos PL/SQL relacionados entre sí, encapsulándolos y convirtiéndolos en una unidad dentro de la base de datos.

Los paquetes están divididos en dos partes: especificación (obligatoria) y cuerpo (no obligatoria). La especificación o encabezado es la interfaz entre el paquete y las aplicaciones que lo utilizan y es allí donde se declaran los tipos, variables, constantes, excepciones, cursores, procedimientos y funciones que podrán ser invocados desde fuera del paquete.

1. ¿Cuáles son los mecanismos para la creación, invocación, modificación y borrado de paquetes en ORACLE?

Creación:

* Cabeza

create [or replace] package nombre\_paquete as  
       <declaraciones públicas>  
       <especificaciones de subprogramas>  
    end nombre\_paquete;

* Cuerpo

create [or replace] package body nombre\_paquete as  
       <declaraciones privadas>  
       <código de subprogramas>  
       [begin  
          <instrucciones iniciales>]  
    end nombre\_paquete;

Invocación:

Nombre\_paquete.función(x);

Nombre\_paquete.procedimiento(x);

Nombre\_paquete.**variable**;

Modificación:

ALTER PACKAGE emp\_mgmt

COMPILAR EL PAQUETE;

Borrado:

DROP PACKAGE emp\_mgmt;

1. **Cursores**
2. ¿Qué es un SYS\_REFCURSOR? ¿Para qué sirve?

Es un puntero a un área privada de SQL que almacena información sobre el procesamiento de una sentencia SELECT o lenguaje de manipulación de datos (DML) (INSERT, UPDATE, DELETE o MERGE).

SYS\_REFCURSOR se puede usar para pasar cursores desde y hacia un procedimiento almacenado.

1. ¿Cómo se define, se asigna y se retorna?

Definición:

CREATE OR REPLACE  
PROCEDURE GET\_EMP\_RS (V\_DEPTID IN EMPLOYEES.DEPARTMENT\_ID%TYPE,  
V\_EMPLOYEES OUT SYS\_REFCURSOR) AS  
BEGIN  
OPEN V\_EMPLOYEES FOR  
SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME  
FROM EMPLOYEES  
WHERE DEPARTMENT\_ID = V\_DEPTID  
ORDER BY EMPLOYEE\_ID;

END GET\_EMP\_RS;

Asignación:

CREATE DummyFunction(dummyValue AS NUMBER) RETURN SYS\_REFCURSOR

IS

RETURN\_DATA SYS\_REFCURSOR;

BEGIN

OPEN RETURN\_DATA

SELECT

TO\_CHAR(dummyValue) || 'A' AS ColumnA

,TO\_CHAR(dummyValue) || 'B' AS ColumnB

FROM

DUAL;

RETURN RETURN\_DATA;

END;

--sample query with sub-queries; does not work

SELECT

SELECT ColumnA FROM DummyFunction(1) FROM DUAL AS ColumnA

,SELECT ColumnB FROM DummyFunction(1) FROM DUAL AS ColumnB

FROM

DUAL;

Retorno:

CREATE OR REPLACE FUNCTION get\_employee\_details(p\_emp\_no IN EMP.EMPNO%TYPE)

RETURN SYS\_REFCURSOR

AS

o\_cursor SYS\_REFCURSOR;

BEGIN

OPEN o\_cursor FOR

SELECT EMPNO,

ENAME,

JOB,

MGR,

HIREDATE,

SAL,

COMM,

DEPTNO

FROM emp

WHERE EMPNO = p\_emp\_no;

RETURN o\_cursor;

-- exception part

END;

**BIBLIOGRAFÍA**

* <https://elbauldelprogramador.com/introduccion-plsql-transacciones/>
* <https://entendiendoaoracle.wordpress.com/2014/11/01/control-de-concurrencia-en-oracle/>
* <https://www.swapbytes.com/niveles-aislamiento-base-de-datos/>
* <https://docs.oracle.com/cd/E56339_01/html/E53985/ugintro-8.html>
* <http://www.tutorialesprogramacionya.com/oracleya/temarios/descripcion.php?cod=240&punto=&inicio>=
* <http://www.mundoracle.com/vistas.html?Pg=sql_plsql_9.htm>
* <https://elbauldelprogramador.com/plsql-paquetes-packages/>
* <https://desarrolloweb.com/articulos/paquetes-oracle.html>
* <https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_2005.htm>
* <https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/drop_package.htm#LNPLS99992>
* <https://stackoverflow.com/questions/2575653/oracle-pl-sql-selecting-from-sys-refcursor>