**Autoestudio 06**

**TRANSACCIONES Y SEGURIDAD**

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

**Para el comienzo y fin de una transacción en ORACLE se define como:**

CREATE OR REPLACE

PROCEDURE Name(atributos)

IS

BEGIN

--Código

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE\_APPLICATION\_ERROR(-20999,SQLERRM);

END;

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

Los diferentes tipos de aislamiento que soporta ORACLE son:

READ COMMITTED: Permite que se ejecuten las transacciones solo si las anteriores han dado un COMMIT, evita las lecturas “sucias”.

REPETABLE READ (Si a la transacción se le coloca SERIALIZABLE): Por ser SERIALIZABLE en ORACLE su detalle es el mismo que SERIALIZABLE.

SERIALIZABLE: Las transacciones se ejecutan en secuencia, como si estuvieran aisladas una de las otras, evitando problemas como las lecturas “sucias”, evita que no se lea una tupla que haya cambiado, y evita lecturas “fantasma”.

1. **Seguridad**
2. **El mecanismo de control de acceso discrecional, ¿Cómo se define en ORACLE? Explicite acciones, objetos y personas autorizadas.**

Existen dos mecanismos de control de acceso discrecional, GRANT y REVOKE, que permiten dar y quitar privilegios respectivamente, en ORACLE se definen de la siguiente manera.

Los privilegios que se puede dar y quitar con GRANT y REVOKE son:

* Borrar de una tabla especifica.
* Insertar datos a una tabla especifica.
* Crear llaves foráneas.
* Seleccionar de una tabla o vista.
* Crear un disparador en una tabla.
* Actualizar una tabla.
* Correr un procedimiento o función especifico.

Sintaxis para tablas (GRANT):

GRANT tipo-de-privilegios

ON [TABLE] {nombre-tabla | nombre-vista}

TO usuarios

Sintaxis para rutinas (GRANT):

GRANT EXECUTE

ON {FUNCTION | PROCEDURE} routine-designator

TO usuarios

Sintaxis para tablas(REVOKE):

REVOKE tipo-de-privilegios

ON [TABLE] {nombre-tabla | nombre-vista}

FROM usuarios

Sintaxis para rutinas(REVOKE):

REVOKE EXECUTE

ON {FUNCTION | PROCEDURE} route-designator

FROM usuarios

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

Una vista es una consulta que se presenta como una tabla (virtual) a partir de un conjunto de tablas en base de datos relacional. Sirve para “esconder” datos de los usuarios, tener consultas listas, son más fácil de entender.

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

Para crear una vista en ORACLE se emplea la siguiente sintaxis:

CREATE OR REPLACE nombre-vista AS (Consulta SQL)

Para eliminar una vista en ORACLE se emplea la siguiente sintaxis:

DROP VIEW nombre-vista

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

Las restricciones son:

1. Si una vista es definida por una consulta que contenga SET o DISTINCT, un GROUP BY, entonces las filas no se pueden insertar, actualizar ni borrar de la base de datos usando la vista.
2. Si una vista es definida con WITH CHECK OPTION, una fila no puede ser insertada o actualizada, usando la vista, si la vista no puede consultar la fila de la base de datos.
3. Si una columna NOT NULL que no tiene una clausula DEFAULT es omitida de la vista, entonces una fila no puede ingresar a través de la vista en la base de datos.
4. Si una vista es creada usando una expresión, entonces las filas no se pueden insertar ni actualizar usando la vista.
5. **Modularidad Paquetes**
6. **¿Para qué sirve un paquete?**

Sirve para colocar un grupo de tipos, variables, constantes, subprogramas, cursores y excepciones. Un paquete es compilado y guardado en la base de datos, donde muchas aplicaciones pueden compartir su contenido.

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

Creación:

CREATE OR REPLACE PACKAGE (nombre)

Invocación:

DECLARE

Nombre-variable TYPE

BEGIN

Nombre-variable := nombre\_package.nombre\_funcion(parámetros)

END;

Modificación:

ALTER PACKAGE (nombre)

Borrado:

DROP PACKAGE (nombre)

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

Es un puntero a el resultado de una consulta, permite devolver conjunto de registros de procedimientos y funciones.

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

Para definir un REFCURSOR, primero es necesario crear un paquete que lo defina, luego se crea una consulta que use al REFCURSOR.

PACKAGE nombre IS

TYPE nombre2 IS RECORD (Atributos)

TYPE nombre3 is REFCURSOR RETURN nombre2

Para asignar y retornar:

CREATE OR REPLACE PACKAGE PC\_Nombre IS

FUNCTION nombre RETURN SYS\_REFCURSOR;

BEGIN   
OPEN var\_cursorpackage FOR

SELECT atributos

FROM tabla

RETURN cursor  
 END

END PC\_Nombre;

**REFERENCIAS:**

* Notas de clase (Campus virtual)
* <https://docs.oracle.com/database/121/ADFNS/adfns_sqlproc.htm#ADFNS99932>
* <https://docs.oracle.com/javadb/10.8.3.0/ref/crefsqlj39374.html>
* <https://www.essentialsql.com/what-is-a-relational-database-view/>
* <https://es.wikipedia.org/wiki/Vista_(base_de_datos)>
* <https://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_8004.htm#SQLRF01504>
* <https://docs.oracle.com/cd/B28359_01/server.111/b28310/views001.htm#ADMIN11781>
* <https://docs.oracle.com/database/121/LNPLS/packages.htm#GUID-8D02540E-C697-4498-9261-848F6D4E5CB5>
* <https://docs.oracle.com/database/121/LNPLS/create_package.htm#LNPLS01371>
* <https://www.techonthenet.com/oracle/questions/exec_function.php>
* <https://oracle-base.com/articles/misc/using-ref-cursors-to-return-recordsets>
* <https://docs.oracle.com/cd/E12839_01/bi.1111/b32122/orbr_refcur002.htm#RSBDR2044>
* <https://www.lawebdelprogramador.com/foros/Oracle/982754-manejo-de-sys_refcursor.html>