**MODELOS Y BASES DE DATOS**

**PL/ SQL Básico**

**2019-02**

**Guía autoestudio 6/6 [:)]**

**Integrantes:**

* Fabian Mauricio Ramírez Pinto
* Iván Camilo Rincón Saavedra

**TRANSACCIONES Y SEGURIDAD**

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

* **Comienzo**

Para comenzar una transacción en ORACLE se procede a utilizar la siguiente sintaxis:

**CREATE OR REPLACE**

**PROCEDURE TRANSACCION [**Descripción]

**IS**

**BEGIN**

* **Finalización**

Para finalizar una transacción en ORACLE, se procede a utilizar la siguiente sintaxis:

**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?**

* Existen 2 tipos de aislamiento que soporta ORACLE, a continuación, su explicación más a fondo:

**Serializable:** Esta se encarga de especificar que todas las transacciones ocurran de un mismo modo aislado, el cual es una transacción hacía de otra.

**Lectura Comprometida:** solo permite la lectura de datos que se encuentren comprometidos.

1. **.¿Cuál es el tipo de aislamiento por defecto en ORACLE?**

ORACLE proporciona 3 niveles de aislamiento, pero solo uno por defecto:

**Read-commited:** Nivel de aislamiento por defecto, cada consulta de la transacción ve solo los que se encuentran confirmados, antes de que la consulta comenzara.

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

Los mecanismos de control discrecional funcionan con permisos, de la siguiente forma:

* **Dar permisos:** Se dan permisos de la siguiente forma:

GRANT privilege

ON element

TO [user | role | PUBLIC]

[WITH GRANT OPTION]

* **Quitar permisos:** Se quitan permisos de la siguiente forma:

REVOKE privilege

ON element

FROM [user | role | PUBLIC]

[RESTRICT | CASCADE]

* **Acciones:** Con respecto a las acciones, existen 4 tipos de acciones:

INSERT [(columnas)]

DELETE

UPDATE [(columnas)]

SELECT [(columnas)]

1. **Vistas** 
   1. **¿Cuáles son los mecanismos para la creación y borrado de vistas en ORACLE?**
      * Para la creación de vistas en Oracle se utiliza la siguiente sentencia:

**CREATE VIEW** nombre vista **AS (**consulta SQL**);**

* + - Para el borrado de vistas en ORACLE se utiliza la siguiente sentencia:

**DROP VIEW** nombre vista**;**

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

Las restricciones que existen en las vistas en ORACLE se identifican por 4 caracteres, las cuales se identificaran a continuación:

* + **FORCE:** Esta función se encarga de crear la vista, así la tabla base no exista.
  + **WITH CHECK POINT:** Esta función se encarga de que a la hora de que se inserten las filas o se actualicen, solo se especifiquen las filas accesibles.
  + **WITH READ ONLY:** Esta función se encarga de que no se realiza ninguna operación sobre la vista, si no que solamente sea solo para lectura.
  + **CONSTRAINT:** Es el nombre que se le asigna a los checks.

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

Un paquete es una estructura que agrupa objetos de PL/SQL compilados(procedures, funciones, variables, tipos ...) en la base de datos. Esto nos permite agrupar la funcionalidad de los procesos en programas.  
   
Lo primero que debemos tener en cuenta es que los paquetes están formados por dos partes: la **especificación** y el **cuerpo**. La especificación de un paquete y su cuerpo se crean por separado.

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

Los mecanismos utilizados para la creación, invocación, modificación y borrado son los siguientes, cada uno de estos se usan en ORACLE:

* + - **Creación:** Se utiliza la siguiente sintaxis:

**CREATE OR REPLACE PACKAGE** (NOMBRE)

* + - **Invocación:** Se utiliza la siguiente sintaxis:

**EXECUTE Packete.Funcion** (ARGUMENTOS)

* + - **Modificación:** Se utiliza la siguiente sintaxis:

**ALTER PACKAGE** (NOMBRE)

* + - **Eliminación:** Se utiliza la siguiente sintaxis:

**DROP PACKAGE** Paquete:

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

Es una variable que es definida de tipo cursor, que se encargara de ser un tipo cursor, que apuntara o hará la referencia al resultado que genere un cursor.

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

A continuación, se define la sintaxis para definir, para asignar y para retornar:

* + **Definir:**

TYPE var\_refcursor IS REF CURSOR;

* + **Asignar:**

create or replace PACKAGE BODY ejemplo\_package IS

FUNCTION retorna\_datos RETURN var\_refcursor IS var\_cursorpackage var\_refcursor;

* + **Retornar:**

FUNCTION retorna\_datos RETURN var\_refcursor;

END ejemplo\_package;

RETURN var\_cursorpackage;

END;

END ejemplo\_package;

**Referencias:**

1. <http://concurrenciaenoracle.blogspot.com/2014/10/primero-que-nada-que-es-laconcurrencia.html>
2. <http://usodeoracle.blogspot.com/p/que-es-un.html>
3. <https://www.lawebdelprogramador.com/foros/Oracle/982754-manejo-de-sys_refcursor.html>
4. <https://docs.oracle.com/cd/E12839_01/bi.1111/b32122/orbr_refcur002.htm#RSBDR2044>