**Autoestudio 05 - MBDA**

Jefer Alexis González Romero  
Angel Nicolas Cuervo Naranjo

**TRANSACCIONES**

**A.** **Transacciones**

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

- Comienzo de una transacción:

Una transacción comienza con la sentencia ‘BEGIN’ seguido de las operaciones que la componen.

- Fin de una transacción:

- Una transacción puede terminar bajo diferentes circunstancias. Una transacción finaliza cuando ocurre alguna de las siguientes acciones:

- Un usuario emite una declaración COMMIT o ROLLBACK sin una cláusula SAVEPOINT.

- Un usuario ejecuta un comando DDL como CREATE, DROP, RENAME o ALTER.

- Un proceso finaliza de manera anormal, lo que hace que la transacción se revierta.

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

- Read Committed Isolation Level (lectura comprometida)

Cada consulta ejecutada por una transacción solo ve los datos confirmados antes de que comenzara la consulta, no la transacción. Este nivel de aislamiento es adecuado para entornos de bases de datos en los que es probable que entren en conflicto pocas transacciones. Una consulta en una transacción de lectura confirmada evita leer los datos que se confirman mientras la consulta está en curso.

- Serializable Isolation Level (serializable)

Una transacción solo ve los cambios confirmados en el momento en que comenzó la transacción, no la consulta, y los cambios realizados por la propia transacción. Una transacción serializable opera en un entorno que hace que parezca que ningún otro usuario modifica los datos de la base de datos. Se asegura que cualquier fila leída por la transacción será la misma cuando se vuelva a leer. Se garantiza que cualquier consulta devolverá los mismos resultados durante la transacción, por lo que los cambios realizados por otras transacciones no son visibles para la consulta, independientemente de cuánto tiempo haya estado ejecutándose. Las transacciones serializables no experimentan lecturas sucias, lecturas difusas o lecturas fantasma.

- Read-Only Isolation Level (de solo lectura)

Las transacciones de solo lectura no permiten que se modifiquen los datos en la transacción a menos que el usuario sea SYS. Las transacciones de solo lectura son útiles para generar informes en los que el contenido debe ser coherente con respecto al momento en que comenzó la transacción.

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

Por defecto el tipo de aislamiento que ofrece Oracle es Read Committed (lectura comprometida)

**B. Seguridad**

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

El mecanismo de control de acceso discrecional (DAC) es un medio para controlar el acceso a la información a través de privilegios, que son permisos para realizar una operación dentro del sistema. El administrador otorga a los usuarios privilegios que determinan las operaciones (como lectura, escritura) que pueden realizar sobre los datos. Un usuario primero debe tener los privilegios DAC necesarios para acceder a los datos en un objeto (tabla).

Definición en ORACLE

* Roles

CREATE ROLE NOMBRE\_ROLE (BY CONTRASEÑA);

* Usuarios

CREATE USER NOMBRE IDENTIFIED BY CONTRASEÑA;

* Asignación de privilegios

GRANT privilegios

ON elemento

TO [ Usuario | Rol]

**C. Vistas**

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

Una vista, es una tabla virtual que no existe físicamente. Se crea mediante una consulta que une una o más tablas.

**Sintaxis de creación:**

CREATE [ OR REPLACE] [ FORCE | NOFORCE] VIEW nombre\_vista

AS consultaSELECT

[ WITH CHECK OPTION [ CONSTRAINT restricción]]

[WITH READ ONLY [ CONSTRAINT restricción]]

**Sintaxis de borrado:**

DROP VIEW nombre\_vista;

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

No pueden contener funciones

No pueden contener ninguna cláusula de agrupación

No siempre permiten operaciones DML.

**D. Modularidad Paquetes**

**1.** ¿Para qué sirve un paquete?

Un paquete es un objeto de esquema que agrupa tipos, variables, constantes, subprogramas, cursores y excepciones de PL/SQL relacionados lógicamente. Un paquete se compila y almacena en la base de datos, donde muchas aplicaciones pueden compartir su contenido.

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

**Sintaxis de creación:**

CREATE [OR REPLACE] [EDITIONABLE | NONEDITIONABLE]

PACKAGE nombe\_paquete;

**Sintaxis de modificación:**

ALTER PACKAGE [schema.] nombe\_paquete

{package\_compile\_clause | {EDITIONABLE | NONEDITIONABLE}};

**Sintaxis de borrado:**

DROP PACKAGE [BODY] [schema.] nombe\_paquete;

**E. SYS\_REFCURSOR**

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

Es un cursor de referencia débil predefinido que viene integrado con el software de base de datos de Oracle.

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

Este se define para retornar la información de una consulta y se asigna para el conjunto de resultados de una consulta.

CREATE OR REPLACE FUNCTION nombreFuncion

RETURN SYS\_REFCURSOR

AS

sys\_cursor SYS\_REFCURSOR;

BEGIN

OPEN sys\_cursor FOR

SELECT \*

FROM TABLA;

RETURN sys\_cursor;

END;

Se puede retornar de la siguiente manera:

VARIABLE elCursor REFCURSOR;

EXECUTE: elCursor := nombreFuncion;

PRINT: elCursor;

**BIBLIOGRAFIA**

Transacciones

<https://docs.oracle.com/database/121/CNCPT/transact.htm#CNCPT117>

Vistas

<https://jorgesanchez.net/manuales/sql/vistas-sql2016.html>

Paquetes

<https://docs.oracle.com/database/121/LNPLS/sqlstatements.htm#LNPLS99979>

Aislamiento

<https://docs.oracle.com/cd/E25054_01/server.1111/e25789/consist.htm#BABEAFAH>

Control de acceso discrecional

<https://docs.oracle.com/cd/B10500_01/network.920/a96578/intro.htm#:~:text=Discretionary%20Access%20Control,-Oracle9i%20provides%20discretionary&text=DAC%20is%20a%20means%20of,%2C%20UPDATE%2C%20and%20DELETE%20privileges>.

**B. Ofreciendo servicios**

**1.** Implemente los paquetes de componentes necesario para ofrecer las funciones básicas y consultas del ciclo actual del sistema (CRUD).

En el archivo musicians.sql

**2.** Proponga un caso de prueba exitoso por subprograma. (CRUDOK)

En el archivo musicians.sql

**3.** Proponga tres casos en los que el subprograma no se puede ejecutar. (CRUDNoOK)

En el archivo musicians.sql

**4.** Escriba las instrucciones necesarias para eliminar los paquetes. (CRUDX)

En el archivo musicians.sql