TRANSACCIONES

**A. Transacciones**

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

En Oracle, las transacciones no requieren un comando explícito como BEGIN TRANSACTION o BEGIN para comenzar. La transacción empieza implícitamente tan pronto como se ejecuta una sentencia que modifique datos, como un INSERT, UPDATE, DELETE

El comando COMMIT confirma y hace permanentes todos los cambios realizados durante la transacción en la base de datos. Una vez que se ejecuta COMMIT, los cambios son visibles para otras sesiones y no se pueden deshacer.

* COMMIT;

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 el 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.

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

Existen 4 niveles donde el primero Read Uncommitted no es soportado en Oracle ya que no permite lecturas sucias.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nivel de aislamiento | Descripción | Problemas que resuelve | Cómo maneja los bloqueos | |  | | --- | | **Posibles problemas** |  |  | | --- | |  | |
| |  | | --- | | **Read Committed**  **NIVEL PREDETERMINADO** |  |  | | --- | |  | | Solo se leen datos confirmados por otras transacciones. Es el nivel por defecto. | Evita lecturas sucias, ya que solo se leen datos confirmados. | Bloqueo a nivel de fila: Mientras una transacción lee una fila, esa fila se bloquea para otras transacciones que intenten modificarla. | **Lecturas no repetibles**: los datos leídos pueden cambiar entre lecturas dentro de la misma transacción. |
| |  | | --- | | **Repeatable Read** | | Las filas leídas durante la transacción no pueden ser modificadas por otras. | Resuelve las lecturas no repetibles: asegura que los datos leídos no cambien durante la transacción. | Bloqueo de filas: Bloquea las filas leídas, pero las consultas que agregan o eliminan filas pueden no ser bloqueadas. Oracle maneja esto con versiones de datos (MVCC). | Lecturas fantasma (phantom reads): nuevas filas pueden ser insertadas o eliminadas por otras transacciones, afectando los resultados de una consulta. |
| Serializable | Las transacciones se ejecutan como si fueran serializadas, sin intersección. | Resuelve todas las anomalías de lectura (sucias, no repetibles y fantasma). | Bloqueo de filas y tablas: Oracle aplica bloqueos estrictos en todas las filas que interactúan en la transacción, lo que evita que otras transacciones modifiquen o lean los datos. | Bajo rendimiento debido a bloqueos más estrictos, y reducción de concurrencia. |

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

El tipo de aislamiento por defecto en Oracle es READ COMMITTED.

**B. Vistas**

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

CREAR:

**CREATE VIEW nombre\_vista AS**

**SELECT columnas**

**FROM tabla**

**[WHERE condiciones];**

BORRAR :

**DROP VIEW nombre\_vista;**

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

No se puede crear una vista que referencie tablas inexistentes. No se puede crear una vista si existe un objeto con ese nombre. No se pueden crear triggers en vistas: Los triggers solo se pueden asociar a tablas, no a vistas.

**C. Modularidad Paquetes**

1. ¿Para qué sirve un paquete?

Los paquetes agrupan procedimientos, funciones, variables, tipos de datos, cursores, excepciones y otros elementos relacionados de forma modular. Los paquetes permiten organizar y encapsular la lógica de negocio y las operaciones relacionadas en un solo objeto de la base de datos.

Los paquetes permiten organizar la lógica del negocio en unidades más manejables. Al agrupar funciones y procedimientos relacionados, se facilita la gestión y el mantenimiento del código.

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

Los Paquetes están divididos en 2 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.

CREACION y MODIFICACION:

CREATE OR REPLACE PACKAGE nombre\_paquete AS

PROCEDURE nombre\_procedimiento(parámetros);

FUNCTION nombre\_función(parámetros) RETURN tipo;

EXCEPTION nombre\_excepción;

END nombre\_paquete;

CREATE OR REPLACE PACKAGE BODY nombre\_paquete AS

PROCEDURE nombre\_procedimiento(parámetros) IS

BEGIN

END nombre\_procedimiento;

FUNCTION nombre\_función(parámetros) RETURN tipo IS

BEGIN

END nombre\_función;

END nombre\_paquete;

Eliminar:

Para borrar tanto la especificación como el cuerpo del paquete, se usa el comando:

DROP PACKAGE nombre\_paquete;

O por separado:

DROP PACKAGE BODY nombre\_paquete;

DROP PACKAGE SPECIFICATION nombre\_paquete;

Invocacion:

* Los procedimientos se invocan con nombre\_paquete.nombre\_procedimiento (parámetros).
* Las funciones se invocan con resultado := nombre\_paquete.nombre\_función (parámetros).

**D. SYS\_REFCURSOR**

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

Un **SYS\_REFCURSOR** es un tipo de cursor en Oracle que se utiliza para manejar conjuntos de resultados de manera flexible. A diferencia de los cursores explícitos, un **SYS\_REFCURSOR** es un cursor genérico que puede contener cualquier tipo de consulta SQL y no requiere definir previamente el tipo de datos o las columnas que devolverá. Es útil para pasar resultados dinámicos entre procedimientos o funciones y para devolver conjuntos de resultados a aplicaciones externas.

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

Se define como:

DECLARE

my\_cursor SYS\_REFCURSOR;

BEGIN

END;

Y se asigna con la clave OPEN FOR:

OPEN my\_cursor FOR

SELECT column1, column2 FROM my\_table WHERE condition;

Y se retorna:

* Desde una **función**: RETURN my\_cursor;
* Desde un **procedimiento**: Usar el parámetro OUT para pasar el cursor a la llamada externa.

**Bibliografía**

1. El Baúl del Programador. (s.f.). *Introducción a PL/SQL - Transacciones*. Recuperado de <https://elbauldelprogramador.com/introduccion-plsql-transacciones/>
2. Oracle. (s.f.). *PL/SQL Procedural Language and SQL* (Documentación oficial de Oracle). Recuperado de <https://docs.oracle.com/cd/E58626_01/E69681/html/E69652/z40016581004135293.html>
3. Tutoriales de Programación Ya. (s.f.). *Descripción de los temarios de Oracle*. Recuperado de <https://www.tutorialesprogramacionya.com/oracleya/temarios/descripcion.php?cod=240&punto=&inicio=>
4. Tutoriales de Programación Ya. (s.f.). *Descripción de las vistas en Oracle*. Recuperado de https://www.tutorialesprogramacionya.com/oracleya/temarios/descripcion.php?cod=240&punto=&inicio=#:~=Al%20crear%20una%20vista%2C%20Oracle,un%20objeto%20con%20ese%20nombre.
5. El Baúl del Programador. (s.f.). *PL/SQL Paquetes (Packages)*. Recuperado de <https://elbauldelprogramador.com/plsql-paquetes-packages/>
6. MagicPLSQL. (2016). *Ref Cursor de Oracle*. Recuperado de <https://magicplsql.blogspot.com/2016/12/ref-cursor-de-oracle.html>