**Investigación**

**Auto Estudio 4**

**Davor Cortés**

**Paula Guevara**

**A. Acciones referenciales**

1. ¿Para qué sirven las acciones referenciales?

Las acciones referenciales se utilizan para garantizar la integridad de los datos si las filas de la tabla maestra se van a eliminar (o van a ser actualizadas). Si todavía existen filas dependientes en tablas referendo, esas referencias tienen que primero validarse.

2. ¿Qué acciones soporta ORACLE? ¿Qué permite hacer cada una de ellas?

SET NULL: indica que si eliminamos un registro de la tabla referenciada cuyo valor existe en la tabla principal, dicho registro se elimina y los valores coincidentes en la tabla principal se conviertan a Null;.

CASCADE: indica que si eliminamos un registro de la tabla referenciada en una foreing key, los registros coincidentes en la tabla principal, también se eliminan; es decir, si eliminamos un registro al cual una clave foránea referencia, dicha eliminación se extiende a la otra tabla.

● NO ACTION: indica que si se intenta eliminar un registro de la tabla referenciada por una foreing key, Oracle no lo permita y muestre un mensaje de error. Se establece omitiendo la cláusula delete al establecer la restricción.

**B.PL/SQL**

1. ¿Qué es PL/SQL?

Es un lenguaje de procesamiento procedimental implementado por Oracle, el cual dispone de estructuras de programación similares a las de la mayoría de los lenguajes de programación y su objetivo es interactuar con la base de datos incluyendo nuevas características como:

* El manejo de variables.
* Estructuras modulares.
* Estructuras de control de flujo y toma de decisiones.
* Control de excepciones.

.

2. ¿Qué motores lo soportan?

Oracle

**C. Datos e instrucciones**

1. ¿Cuáles son los tipos de datos que ofrece PL/SQL?

Los Tipos básicos son:

NUMBER (numérico): Almacena todo tipo de números reales, sin límite de longitud, aunque esta puede limitarse.

CHAR (Carácter): Almacena hasta 32767 datos de tipo carácter, es decir letras.

VARCHAR (Carácter de longitud variable): Almacena datos de tipo carácter pero solo

utiliza la cantidad necesaria.

BOOLEAN (lógico): Se almacenan datos, de tipo booleano, es decir los datos de tipo True o False.

DATE (Fecha): Almacena datos de fechas que están contenidas entre el año de 4712 a.C. hasta el 4712 D.C.

2. ¿Cuál es la forma de definir constantes y variables?

DECLARE

[nombre\_variable] [tipo] [NOT NULL] :=/DEFAULT [expresion\_o\_valor];

BEGIN

[sentencia\_ejecutable];

END;

DECLARE

[nombre\_variable] CONSTANT [tipo] [NOT NULL] :=/DEFAULT [expresion\_o\_valor];

BEGIN

[sentencia\_ejecutable];

END;

3. ¿Cómo se define una variable con un tipo tomado de la base de datos?

DECLARE

[nombre\_variable] [tipo] [NOT NULL] :=/DEFAULT [expresion\_o\_valor];

BEGIN

[sentencia\_ejecutable];

END;

4. ¿Cuál es la forma en PL/SQL de los diferentes tipos de asignación? (Son

tres)

1. [nombre] [tipo\_dato] := [expresion\_valor]

2. [nombre] [tipo\_dato] DEFAULT [expresion\_valor]

3. [nombre] [tipo\_dato] NOT NULL [expresion\_valor]

**D.Cursores**

1. ¿Qué es un cursor implícito? ¿Para qué sirve?

Un cursor se forma por información devuelta de una instrucción SELECT, pero el cursor implícito se utiliza solo cuando la sentencia SELECT devuelve un solo registro, de manera que se pueda asignar a alguna tupla en específico si se quiere.

2.¿Qué es un cursor explícito? ¿Para qué sirve?

El cursor explícito a diferencia del anterior se utiliza cuando la sentencia SELECT puede devolver varios registros y si utilizar esta información para agregar a alguna tabla o crear vistas, por ejemplo.

3.¿Cuáles son las excepciones propias de uso de estos cursores?

Cursor Implícito:

○ NO\_DATA\_FOUND: Se produce cuando una sentencia SELECT intenta recuperar

datos pero ninguna fila satisface sus condiciones, es decir, cuando no hay datos.

○ TOO\_MANY\_ROWS: Dado que cada cursor implícito sólo es capaz de recuperar

una fila , esta excepción detecta la existencia de más de una fila.

Cursor Explicito:

○ CURSOR\_ALREADY\_OPEN: El programa intentó abrir un cursor que ya estaba

abierto.

○ INVALID\_CURSOR: El programa intentó efectuar una operación no válida sobre el

cursor.

**E.Modularidad**

1.¿Cuál es la estructura general de un bloque PL/SQL?

DECLARE \*Opcional\*

[Variables | cursores | excepciones definidas por el usuario]

BEGIN \*Obligatorio\*

(Indica que empieza el programa)

[Sentencias SQL | Sentencias de Control PL/SQL]

EXCEPTION \*Opcional \*

[Acciones a realizar cuando se produzcan errores]

END; \*Obligatorio\*

2.¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo,

procedimiento, función y disparador)

Bloque anónimo: permite crear y ejecutar código de procedimiento sin almacenar el código persistentemente como objetos de base de datos. El concepto de bloques anónimos es similar a los scripts shell UNIX, los cuales activan diversos comandos ingresados manualmente a ser agrupados y ejecutados como un solo paso. Como su nombre lo indica, los bloques anónimos no tienen nombre y por esta razón no pueden ser referenciados por otros objetos. Procedimiento: es un subprograma que ejecuta una acción específica, tiene un nombre, puede tener un conjunto de parámetros y un bloque de código. Tras crear el procedimiento, éste se compila y luego se almacena en la BD de forma compilada. Este procedimiento luego puede ser invocado desde cualquier bloque PL/SQL, de esta manera hace más práctico la construcción de los códigos.

Function: Una función es un bloque muy parecido al procedimiento, la mayor diferencia es que la función debe retornar un valor y el procedimiento puede ó no puede hacerlo, con este funcionamiento la función puede hacer uso posteriormente de resultados obtenidos en consultas.

Disparador: Un disparador es una estructura que se invoca automáticamente cuando un evento especificado ocurre. Algunos usos de los disparadores:

○ Generar valores de columnas automáticamente.

○ Registrar eventos

○ Prevenir transacciones inválidas

○ Restringir operaciones en una tabla después de cierta hora