**MODELOS Y BASES DE DATOS**

**PL/ SQL Básico**

**2024-1**

**Guía autoestudio 4/6**

**NOMBRES: Ivan Santiago Forero Torres – Sebastián Albarracín Silva**

OBJETIVO

1. Conocer herramientas que facilitan el trabajo del desarrollador de una base de datos específicamente la herramienta SQL Developer.

2. Desarrollar competencias para definir e implementar restricciones de integridad con mecanismos declarativos y procedimentales.

TÓPICOS OBJETIVO 2

1. Acciones referenciales

2. Disparadores

3. Constantes y variables

4. Instrucciones básicas: asignación

5. Cursores: implícitos y explícitos

ENTREGA Publicar las respuestas en el espacio correspondiente en un archivo .zip , el nombre de este archivo debe ser la concatenación en orden alfabético de los primeros apellidos de cada uno de los miembros.

**INVESTIGACIÓN**

**A. Acciones referenciales**

1. ¿Para qué sirven las acciones referenciales?

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

Las acciones más comunes que Oracle soporta y lo que permite hacer cada una de ellas:

**Creación de tablas (CREATE TABLE)**

**Inserción de datos (INSERT)**

**Actualización de datos (UPDATE:**

**Eliminación de datos (DELETE)**

**Selección de datos (SELECT)**

**Creación de vistas (CREATE VIEW)**

**Creación de disparadores (CREATE TRIGGER)**

**B. PL/SQL**

1. ¿Qué es PL/SQL?

Es un lenguaje de programación utilizadopor la base de datos Oracle, combina las capacidades de SQL para manipular datos con las características de un lenguaje procedural, como variables, estructuras de control, procedimientos y funciones.

PL/SQL se utiliza principalmente para desarrollar aplicaciones de base de datos complejas, como procedimientos almacenados, funciones, disparadores y paquetes

2. ¿Qué motores lo soportan?

Ya que es un lenguaje especifico creado por Oracle solamente es soportado por los motores proporcionados por Oracle Corporationcomo Oracle Database

**C. Datos e instrucciones en PL/SQL**

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

Los tipos de datos que ofrece PL/SQL ofrece datos de tipos escalares, tipos de datos compuestos, tipos de datos coleccionables y Large Object.

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

Las constantes no cambian durante la ejecución, Las variables son contenedores de datos que pueden cambiar su valor durante la ejecución del programa,

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

* **<nombre\_variable>** es el nombre que le das a la variable que estás definiendo.
* **<nombre\_tabla>** es el nombre de la tabla de la cual deseas tomar el tipo de datos.
* **<nombre\_columna>** es el nombre de la columna en la tabla de la cual deseas tomar el tipo de datos.

4. ¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)

Los diferentes tipos de asignación que tiene PL/SQL son:

Asignacion simple (=)

<variable> := <expresión>;

Asignacion selectiva (Select into)

SELECT <columna1>, <columna2>, ... INTO <variable1>, <variable2>, ...

FROM <tabla>

WHERE <condición>;

Asignacion predeterminada (DEFAULT)

SELECT <columna1> INTO <variable> FROM <tabla> WHERE <condición>

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

<variable> := <valor\_predeterminado>;

**D. Cursores**

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

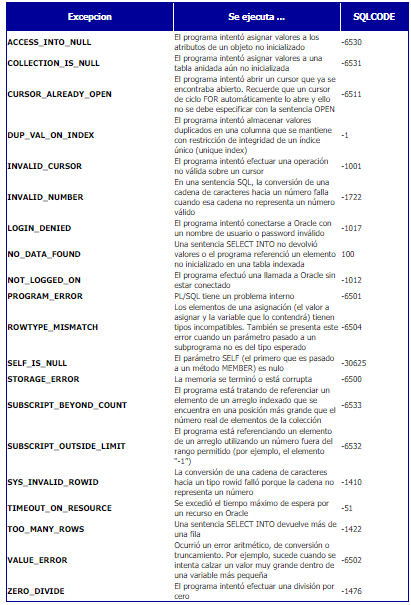
Un cursor es el nombre para un área memoria privada que contiene información procedente de la ejecución de una sentencia SELECT. Cada cursor tiene unos atributos que nos devuelven información útil sobre el estado del cursor en la ejecución de la sentencia SQL. Cuando un cursor está abierto y los datos referenciados por la consulta SELECT cambian, estos cambios no son recogidos por el cursor.

Este tipo de cursores se utiliza para operaciones **SELECT INTO**. Se usan cuando la consulta devuelve un único registro.

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

Son los cursores que son declarados y controlados por el programador. Se utilizan cuando la consulta devuelve un conjunto de registros. Ocasionalmente también se utilizan en consultas que devuelven un único registro por razones de eficiencia. Son más rápidos.

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

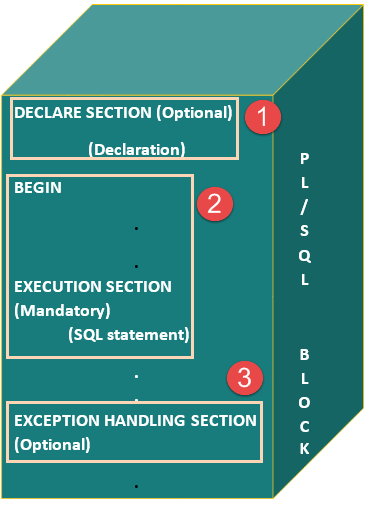


**E. Modularidad**

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

Los bloques PL/SQL tienen una estructura predefinida en la que se agrupará el código. A continuación, se muestran diferentes secciones de bloques PL/SQL.

1. Sección de declaración
2. Sección de ejecución
3. Sección de manejo de excepciones



2. ¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo, procedimiento, función y disparador

Los bloques anónimos son bloques PL/SQL que no tienen ningún nombre asignado. Deben crearse y usarse en la misma sesión porque no se almacenarán en el servidor como objetos de base de datos.

Como no necesitan almacenarse en la base de datos, no necesitan pasos de compilación. Se escriben y ejecutan directamente, y la compilación y ejecución ocurren en un solo proceso.

Los bloques con nombre tienen un nombre específico y único para ellos. Se almacenan como objetos de base de datos en el servidor. Dado que están disponibles como objetos de base de datos, se puede hacer referencia a ellos o utilizarlos siempre que estén presentes en el servidor. El proceso de compilación de bloques con nombre ocurre por separado mientras se crean como objetos de base de datos.

CREATE TABLE BOOKING(

BOOKING\_ID INT NOT NULL,

BOOKING\_DATE DATE NULL,

ROOM\_NO INT NULL,

GUEST\_ID INT NOT NULL,

OCCUPANTS INT NOT NULL,

ROOM\_TYPE\_REQUESTED VARCHAR(6) NULL,

NIGHTS INT NOT NULL,

ARRIVAL\_TIME VARCHAR(6) NULL

);

CREATE TABLE EXTRA(

EXTRA\_ID INT NOT NULL,

BOOKING\_ID INT NULL,

DETAIL VARCHAR(50) NOT NULL,

AMOUNT INT NOT NULL,

DISCOUNT INT NOT NULL

);

CREATE TABLE DETAIL (

EXTRA\_ID INT NOT NULL,

DETAIL VARCHAR(50) NOT NULL

);

-- ATRIBUTOS --

ALTER TABLE EXTRA

ADD CONSTRAINT CK\_ID\_EXTRA CHECK ((EXTRA\_ID > 0) AND EXTRA\_ID LIKE '9%');

ALTER TABLE EXTRA

ADD CONSTRAINT CK\_AMOUNT\_EXTRA CHECK (AMOUNT > 0);

ALTER TABLE EXTRA

ADD CONSTRAINT CK\_DISCOUNT\_EXTRA CHECK (DISCOUNT > 0);

-- AGREGAR OTRA COLUMNA AL DETAIL--

ALTER TABLE DETAIL ADD DETAIL\_ID INT NOT NULL;

-- LLAVES PK--

ALTER TABLE EXTRA ADD CONSTRAINT PK\_EXTRA

PRIMARY KEY (EXTRA\_ID);

ALTER TABLE BOOKING ADD CONSTRAINT PK\_BOOKING

PRIMARY KEY (BOOKING\_ID);

ALTER TABLE DETAIL ADD CONSTRAINT PK\_DETAIL

PRIMARY KEY (DETAIL\_ID);

-- LLAVES FK--

ALTER TABLE EXTRA ADD CONSTRAINT FK\_EXTRA

FOREIGN KEY (BOOKING\_ID) REFERENCES BOOKING(BOOKING\_ID);

ALTER TABLE DETAIL ADD CONSTRAINT FK\_DETAIL

FOREIGN KEY (EXTRA\_ID) REFERENCES EXTRA(EXTRA\_ID);

--