**DECLARACION DE BLOQUE DE PL/SQL**

[DECLARE]

BEGIN

[DECLARE]

BEGIN

[EXCEPTION]

END;

[EXCEPTION]

END;

**DECLARACION DE VARIBLES**

* Tipos habituales: Number (x,y), date, VARCHAR2(X), boolean
* Menos habituales: Blob, clob,
* Referenciados a campos de tablas: employees.name%type
* Refenciados a tablas: emp employees%rowtype => emp.name
* Definidas como tipos definidas por el usuario o como arrays de tipos.

**ESTRUCTURAS DE CONTROL:**

* IF - THEN / END IF;
* IF - THEN / ELSE /END IF;
* IF - THEN / ELSEIF THEN / ELSE / END IF;

**BUCLES**

* LOOP / [EXIT] / [EXIT WHEN] / END LOOP;
* WHILE LOOP / END LOOP;
* FOR LOOP [REVERSE para el contador] / END LOOP;

**CIERRE DE TRANSACCIONES**=> COMMIT y ROLLBACK

**CURSORES**

FOR - El for abre el cursor, extrae los datos y se cierra de modo implícito; y además comprueba automáticamente si ya no quedan más filas.

FOR var\_tipo\_cursor IN CURSOR LOOP

…

END LOOP

| **CURSORES** | |
| --- | --- |
| 1. Declarar cursor   **CURSOR *nombre\_cursor* [ (parámetro1 [, parámetro2]…) ]*IS sentencia\_select ;***   1. Abrirlo  **OPEN** nombre\_cursor 2. Recoger los datos en Var PL  **FETCH** nombre\_cursor **INTO** varPL ó registroPL; 3. Cerrarlo  **CLOSE** nombre\_cursor; | ATRIBUTOS DE CURSORES:   * **%NOTFOUND** 🡪 Atributo booleano que devuelve TRUE cuando el último FETCH no ha recuperado ninguna fila. * **%ISOPEN** 🡪 Se utiliza para determinar si el cursor asociado está abierto o no. * **%ROWCOUNT** 🡪 Devuelve el número de filas extraídas por el cursor hasta el momento.   **Cursores Implícitos: (SQL%NOTFOUND)** |
| **PROCEDIMIENTOS Y FUNCIONES** | |
| ***CREATE [ OR Replace ] PROCEDURE*** *nombre\_proc* ***[(Argumento [ IN | OUT | IN OUT] tipo)] IS | AS***  **BEGIN**  **…**  **END *[nombre\_proc];***    **DROP PROCEDURE** *nombre\_dispa*; | ***CREATE [ OR Replace ] FUNCTION*** *nombre\_func* ***[(Argumento [ IN | OUT | IN OUT] tipo)] RETURN tipo\_retorno IS | AS***  **BEGIN**  **…**  **END *[nombre\_proc];***  **DROP FUNCTION** *nombre\_dispa*; |
| **PAQUETES** | |
| ***CREATE[OR REPLACE] PACKAGE nombre\_paq AS [ especificación procedim  “ funciones, var, tipos, excepciones, cursores ] END [ nombre\_paquete ];***  ***CREATE [OR REPLACE] PACKAGE BODY nombre\_paq AS***  ***END [ nombre\_paquete*** *];* | |
| **EXCEPCIONES** | |
| **Definidas por el usuario**  Se declaran en la sección declarativa. El ámbito de la excepción es igual al de cualquier variable declarada en la misma sección declarativa. Para lanzar o arrancar la excepción de usuario utilizamos ***RAISE nombre\_excep;***  ***Raise\_application\_error (nº error, mensaje de error );*** *Nº de error va desde -20000 a -20999* | **Excepciones Predefinidas**   * **NO\_DATA\_FOUND** * **TOO\_MANY\_ROOMS** * **VALUE\_ERROR** * **CURSOR\_ALREADY\_OPEN** …   **Funciones:**  **SQLCODE, SQLERRM** |
| **TRIGGERS (PL/SQL)** | |
| **CREATE [OR REPLACE] TRIGGER *nombre\_dispa* {BEFORE|AFTER} INSERT OR DELETE OR UPDATE [OF *columna*] ON *tabla* [FOR EACH ROW [WHEN condicion\_disparo]]**  **DECLARE**  **…**  **BEGIN**  **…**  **END;** | **DROP TRIGGER** *nombre\_dispa*; |
| **PSEUDOREGISTROS :OLD, :NEW**  A nivel de fila, accede a la fila que está siendo actualmente procesada. |
|  |
|  | |