

CONCEPTOS BÁSICOS:

- Los **cursores explícitos** se usan en consultas que pueden devolver más de una fila (o ninguna).
 - Los formatos de las principales operaciones que se realizan con un cursor explícito son:

Declarar: `CURSOR <nombrecursor> IS SELECT <sentencia select>;`

Abrir: `OPEN <nombrecursor>;`

Cerrar: `CLOSE <nombrecursor>;`

Recuperar fila: `FETCH <nombrecursor> INTO {<variable>|<listavARIABLES>;}`

 - Los atributos del cursor sirven para conocer información de su estado. Los principales son: %FOUND, %NOTFOUND, %ISOPEN, %ROWCOUNT
 - El **cursor FOR LOOP** incluye de manera implícita las acciones de abrir, declarar variable para recuperar los datos, recorrer y recuperar las filas y cerrar el cursor. Su formato genérico es:

`FOR <nombrevareg> IN <nombrecursor>
LOOP <instrucciones>; END LOOP;`
 - En instrucciones con **cursores implícitos** podemos preguntar por los atributos del cursor aunque estará cerrado.
- Para actualizar una tabla en PL/SQL podemos optar por usar en la **cláusula WHERE**:
 - CURRENT OF para indicar que la fila a actualizar es la actual en cursores declarados FOR UPDATE.
 - El ROWID previamente guardado, o cualquier expresión que determine la/s fila/s a actualizar.
- Cuando se levanta una **excepción** el control pasa a la sección EXCEPTION donde buscará un manejador y dará por finalizada la ejecución del bloque actual. Si no encuentra tratamiento, se propagará hasta encontrarlo o retornará al entorno que lanzó la aplicación dando la ejecución de la aplicación por fallida. Una vez tratada la excepción el control retorna a la línea siguiente a la que llamó al programa que trató la excepción.
 - WHEN OTHERS se usa para tratar cualquier excepción que no aparezca en las cláusulas WHEN anteriores.
 - Las excepciones definidas por el usuario deben ser declaradas y levantadas explícitamente según los formatos estudiados.
- Una **transacción** es un conjunto de operaciones dependientes unas de otras. Para que la transacción se complete han de realizarse todas las operaciones que la componen; en el caso de que alguna falle se dará por fallida toda la transacción.
 - La transacción comienza con la primera orden SQL de la sesión o con la primera orden SQL posterior a la finalización de la transacción anterior y finaliza cuando se ejecuta un comando de control de transacciones (COMMIT o ROLLBACK), una orden de definición de datos (DDL) o cuando finaliza la sesión.
 - Cuando un subprograma almacenado falla y no se controla la excepción que produjo el fallo, Oracle automáticamente ejecuta ROLLBACK sobre todo lo realizado por el subprograma, salvo que en el subprograma hubiese algún COMMIT, en cuyo caso lo confirmado no sería deshecho.
 - Para garantizar la consistencia de los datos recuperados entre distintas consultas usaremos SET TRANSACTION READ ONLY antes de ejecutar las instrucciones SELECT; y una vez ejecutadas deberemos incluir COMMIT para liberar la transacción de sólo lectura.
 - Para crear una transacción autónoma crearemos un bloque o programa que realice las acciones previstas e incluiremos en la sección declarativa la directiva o pragma AUTONOMOUS_TRANSACTION. Debemos confirmar o rechazar la transacción antes de salir del ámbito del programa.

EJERCICIOS:

1 Desarrolla un procedimiento que visualice el apellido y la fecha de alta de todos los empleados ordenados por apellido.

2 Codifica un procedimiento que muestre el nombre de cada departamento y el número de empleados que tiene.

3 Escribe un programa que visualice el apellido y el salario de los cinco empleados que tienen el salario más alto.

4 Codifica un programa que visualice los dos empleados que ganan menos de cada oficina.

5 Desarrolla un procedimiento que permita insertar nuevos departamentos según las siguientes especificaciones:

- Se pasará al procedimiento el nombre del departamento y la localidad.
- El procedimiento insertará la fila nueva asignando como número de departamento la decena siguiente al número mayor de la tabla.
- Se incluirá la gestión de posibles errores.

6 Codifica un procedimiento que reciba como parámetros un número de departamento, un importe y un porcentaje; y que suba el salario a todos los empleados del departamento indicado en la llamada. La subida será el porcentaje o el importe que se indica en la llamada (el que sea más beneficioso para el empleado en cada caso).

7 Escribe un procedimiento que suba el sueldo de todos los empleados que ganen menos que el salario medio de su oficina. La subida será del 50 por 100 de la dife-

9 Crea la tabla T_liquidacion con las columnas apellido, departamento, oficina, salario, trienios, comp_responsabilidad, comisión y total; y modifica la aplicación anterior para que, en lugar de realizar el listado directamente en pantalla, guarde los datos en la tabla. Se controlarán todas las posibles incidencias que puedan ocurrir durante el proceso.

10 Escribe un programa para introducir nuevos pedidos según las siguientes especificaciones:

- Recibirá como parámetros PEDIDO_NO, PRODUCTO_NO, CLIENTE_NO, UNIDADES y la FECHA_PEDIDO (opcio-

renencia entre el salario del empleado y la media de su oficina. Se deberá hacer que la transacción no se quede a medias, y se gestionarán los posibles errores.

8 Diseña una aplicación que simule un listado de liquidación de los empleados según las siguientes especificaciones:

- El listado tendrá el siguiente formato para cada empleado:

```
*****
Liquidación del empleado      : (1)
Dpto                          : (2)
Oficina                       : (3)
Salario                       : (4)
Trienios                      : (5)
Comp. responsabilidad        : (6)
Comisión                      : (7)
*****
Total                         : (8)
*****
```

Donde:

- 1, 2, 3 y 4 corresponden a apellido, departamento, oficina y salario del empleado.
- 5 es el importe en concepto de trienios. Un trienio son tres años completos, desde la fecha de alta hasta la de emisión, y supone 50 €.
- 6 es el complemento por responsabilidad. Será de 100 € por cada empleado que se encuentre directamente a cargo del empleado en cuestión.
- 7 es la comisión. Los valores nulos serán sustituidos por ceros.
- 8 es la suma de todos los conceptos anteriores.

El listado irá ordenado por Apellido.

nal, por defecto la del sistema). Verificará todos estos datos así como las unidades disponibles del producto y el límite de crédito del cliente y fallará enviado un mensaje de error en caso de que alguno sea erróneo.

- Insertará el pedido y actualizará la columna DEBE de clientes incrementándola el valor del pedido (UNIDADES * PRECIO_ACTUAL). También actualizará las unidades disponibles del producto e incrementará la comisión para el empleado correspondiente al cliente en un 5% del valor total del pedido. Todas estas operaciones se realizarán como una única transacción.