

1. **Crear el package PACK_PER cuyas especificaciones tendrán lo siguiente:**
 - La definición de un **tipo registro** con los campos **MONTO_VENTAS** , **MONTO_COMISION**
 - La definición de un tipo **tabla asociativa** cuyos componentes son del tipo de registro creado,
 - La definición de una variable de tipo boolean denominada **V_CALCULA_BONIF** inicializado a TRUE.
 - La función **F_APELLIDO_NOMBRE** que recibe como parámetro la cédula del empleado y devuelve un varchar2 concatenando 'apellido, nombre'. Prever que la función pueda ser empleada en cualquier sentencia SELECT.
 - La función **F_CALCULAR_BONIFICACIÓN** que recibirá como parámetros el año y mes, calculará el monto de ventas y la bonificación correspondientes al año y mes introducidos por parámetro, de todos los empleados, las cargará en la tabla asociativa creada, indexando por cédula del empleado, y devolverá dicha tabla..
 - Sobrecargue la función **F_CALCULAR_BONIFICACIÓN** para que reciba solamente la cédula de un empleado, y devuelva la bonificación por venta correspondiente a ese empleado en el mes y año del sistema.
2. **Probar el paquete**
 - **Pruebe en un SELECT la función del paquete:** Seleccione cédula, nombre del empleado, nombre del jefe, utilizando la función de paquete F_APELLIDO_NOMBRE.
 - **Probando las funciones:** Cree un PL/SQL anónimo que haga lo siguiente:
 - a. Declare una variable del tipo tabla asociativa del paquete
 - b. Verifique el estado de la variable del paquete V_CALCULA_BONIF . Si es true invoque a la función F_CALCULAR_BONIFICACION enviándole el año y mes del sistema y cargue el resultado de la función en la variable recibida
 - c. Recorra la variable de tipo tabla asociativa e imprima el resultado obtenido.Recuerde que los tipos de datos, variables y subprogramas declarados en un paquete, pueden ser invocados desde un subprograma externo sólo con el prefijo del paquete.
3. **Cree el paquete PACK_ART que tiene los siguientes elementos:**
 - La función **F_CALCULAR_STOCK** que recibe el ID de una artículo y devuelve el stock actual
 - El procedimiento **P_INCREMENTAR_ARTICULO** que recibe como parámetro el ID del artículo y la cantidad, y resta del stockDesarrolle el paquete y cuando lo tenga depurado oculte el código interno del paquete
4. Se le ha encomendado desarrollar las funcionalidades de la siguiente pantalla diseñada para gestionar los sorteos anuales de premios.

Paquete: PCK_SORTEO

Salida:

CI Cliente:

Cantidad Cupones:

Premios:

Ganador:

CI	Nombre y Apellido	Premio Adjudicado
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Ejecute el script **EJER_14_SP_SORTEO.sql** que crea las tablas **B_CUPONES**, **B_CUPONES_GANADORES** y la secuencia **SEQ_ID_CUPON** que se encarga de generar la numeración de los cupones.

Cree el paquete **PCK_SORTEO** que contiene los procedimientos y funciones necesarios para la programación de los botones siguientes:

a.

Salida: **CANTIDAD DE CUPONES GENERADOS: 1250**

Este botón llama a la función **F_GENERAR_CUPONES** que recupera todas las ventas al contado efectuadas en el año 2011 y por las boletas con montos ≥ 100.000 gs genera cupones. Es decir inserta en la tabla **B_CUPONES** un registro por cada 100.000 gs.

Ejemplo: Si el importe total de la venta con ID 2 es de 500.000gs entonces se insertan 5 registros en la tabla mencionada.

Luego de haber hecho todas las inserciones retorna la cantidad de cupones generados.

b.

CI Cliente: **1207876 ROBERTO ABENTE GOMEZ**

Cantidad Cupones: **163**

Al pulsar este botón se invoca a la función **F_CONSULTAR_CUPONES**, la misma recibe como parámetro la cédula de un cliente y retorna la cantidad de cupones que se le han generado. En caso de que el cliente no existe emite un mensaje de error.

c.

Premios:	KIA RIO 0 KM	Sortear
Ganador:	KIA RIO 0 KM	
	PASAJE A NY	
	HELADERA TOKYO	

Al presionar el botón se llama a la función **F_SORTEAR** que recibe como parámetro el nombre del premio a sortear;

luego selecciona aleatoriamente (*) un número de cupón de la tabla B_CUPONES.

Si el cupón es de un cliente que aún no fue adjudicado con algún premio, inserta un registro en la tabla B_CUPONES_GANADORES, caso contrario vuelve a seleccionar otro número de cupón.

Por último, debe retornar el nombre y apellido del cliente ganador.

d.

CI	Nombre y Apellido	Premio Adjudicado
429987	JORGE AMADO PERREIRA	KIA RIO 0 KM
1207876	ROBERTO ABENTE GOMEZ	PASAJE A NY
2281987	CRISTHIAN ACHUCARRO	HELADERA TOKYO

Este botón invoca al procedimiento **P_OBTENER_GANADORES**, el mismo imprime los siguientes datos de los clientes ganadores:

CI o RUC, Nombre/Apellido Premio Adjudicado.

(*) *Investigue la función de BD: DBMS_RANDOM.VALUE para la selección aleatoria de números entre un valor mínimo y máximo.*

5. Dado el modelo de la clase, donde se expone la relación entre las tablas:

B_EMPLEADOS, B_POSICION_ACTUAL Y B_CATEGORIAS_SALARIALES.

Donde B_EMPLEADOS contiene el registro de empleados, B_POSICION_ACTUAL contiene el histórico de cargos del empleado. El cargo actual del empleado es aquel registro cuyo FECHA_FIN sea nulo, y B_CATEGORIAS_SALARIALES el salario de los cargos.

Se requiere desarrollar un paquete denominado **pkg_empleado** que permita implementar lo siguiente:

I. Los siguientes tipos de datos

- El tipo registro R_EMPLEADO [**apel_nomb** (nombre y apellido del empleado), **antiguedad_mes** (Antigüedad en meses en la empresa) , **direccion** , **localidad** , **categoria_actual** (nombre de categoria del empleado)]
- El tipo registro R_HISTORICO_CATEGORIA [**nombre_categoria** , **asignación** (Monto del salario), **fecha_ini** (inicio en el cargo) , **fecha_fin** (fin en cargo)]
- La tabla indexada TI_HISTORICO_CATEGORIA del tipo R_HISTORICO_CATEGORIA

II. Los siguientes procedimientos y funciones

- El procedimiento **PA_DETALLE_EMPLEADO**: debe recibir como parámetro de entrada: p_cedula, que es el número de cedula del empleado, y como parámetros de salida: un tipo r_empleado, Cod_Estado código de estado de tipo numérico, y en mensaje de tipo cadena.
El procedimiento debe implementar cuanto sigue:

- Recibe como parámetro el número de cedula.
- Verifica la existencia del registro en la tabla b_empleados.
- Si existe cargar la variable de tipo r_empleado, asignar los valores cero al código de estado y 'Existe' al mensaje.
- Si no existe el registro, asignar un valor -1 a código de estado y 'Registro no existe' al mensaje
- La función **FN_HISTO_CATEGORIAS** recibe como parámetro el número de cedula del empleado y retorna el tipo TI_HISTORICO_CATEGORIA
Los datos son obtenidos de la combinación de las tablas b_posicion_actual y b_categorias salariales. Los datos corresponden a las distintas categorías que tuvo el empleado en su permanencia en la empresa.
- El procedimiento **PA_IMPRIMIR:** debe recibir como parámetro de entrada el número de cédula del empleado y realizar lo siguiente:
 - Invocar al procedimiento pa_detalle_empleado.
 - En caso que el parámetro código de estado retorne cero imprimir en pantalla datos que retorna el parámetro r_empleado, luego invocar la función fn_histo_categorias e imprimir en pantalla los registros retornados en ti_historico_categoria.
 - En caso que el parámetro código de estado retorne distinto de cero imprimir en pantalla el parámetro mensaje.