

## Instrucciones:

Deberás realizar el bloque o los bloques correspondientes a las evaluaciones no superadas. La prueba se deberá realizar usando OracleDB, SQL Developer y SQL Plus disponibles en la máquina virtual puesta a disposición del alumno.

De cada uno de los ejercicios, deberás entregar lo que se indique en el mismo. El documento y scripts entregados deberán estar bien ordenados, formados y debidamente comentados.

PUNTUACIÓN DE CADA APARTADO		VALORACIÓN MÁXIMA
<b>BLOQUE PRIMERA EVALUACIÓN</b>		
Ejercicio 1 (0.50 puntos)		0.50 puntos
Ejercicio 2 (3.50 puntos)	Apdo. A.	1.50 puntos
	Apdo. B.	1.00 puntos
	Apdo. C.	1.00 puntos
Ejercicio 3 (3.00 puntos)	Apdo. A.	1.50 puntos
	Apdo. B.	1.50 puntos
Ejercicio 4 (3.00 puntos)	Apdo. A.	1.75 puntos
	Apdo. B.	1.25 puntos
<b>BLOQUE SEGUNDA EVALUACIÓN</b>		
Ejercicio 5 (3.25 puntos)	Apdo. A.	2.25 puntos
	Apdo. B.	1.00 puntos
Ejercicio 6 (3.50 puntos)	Apdo. A.	1.75 puntos
	Apdo. B.	1.75 puntos
Ejercicio 7 (3.25 puntos)	Apdo. A.	1.00 puntos
	Apdo. B.	2.25 puntos

**BLOQUE PRIMERA EVALUACIÓN****Ejercicio 1:**

Crea un usuario de OracleDB con nombre c##ordi1 y la contraseña ordi1234. Este usuario deberá tener los permisos adecuados para conectar, crear tablas, borrar tablas, ejecutar consultas.... Crea una conexión llamada Ordi1 con este usuario desde Sql Developer a OracleBD. Adjunta a un documento de Word capturas del proceso seguido y del correcto funcionamiento de la conexión creada desde SQL Developer a OracleBD.

**Ejercicio 2:**

- A. Crea y guarda en un archivo llamado **ejercicio2A.sql** las sentencias PL/SQL para OracleDB que creen las siguientes tablas:

**Tabla cliente:**

CodCliente	Número entero	Autonumérico – Primary Key
NomCliente	Texto longitud 30	No nulo y debe tener contenido
cifCliente	Texto longitud 9	No nulo
fechaAlta	Fecha	Valor por defecto la fecha actual
fechaBaja	Fecha	Valor por defecto nulo

**Tabla articulo:**

CodArticulo	Número entero	Autonumérico – Primary Key
NomArticulo	Texto longitud 20	No puede ser nulo
PrecioArt	Número con dos decimales	El precio debe ser mayor o igual que cero

**Tabla pedido:**

CodPedido	Número entero	Autonumérico – Primary Key
fechaPedido	Fecha	Valor por defecto la fecha actual
codCliente	Número entero	No puede ser nulo

**Tabla pedidolinea:**

CodPedido	Número entero	No puede ser nulo
CodPedidoLinea	Número entero	Autonumérico
CodArticulo	Número entero	No puede ser nulo
Precio	Número con dos decimales	Valor por defecto 0
Cantidad	Número entero	Debe ser mayor que 0
Importe	Número con dos decimales	
Iva	Número con dos decimales	Valor por defecto 0.21
TotalLinea	Número con dos decimales	

CodPedido y CodPedidoLinea son la Primary Key de la tabla

Ejecuta las sentencias en Oracle Developer y adjunta una captura de las tablas creadas al documento de Word que creaste en el ejercicio 1.

- B. Crea y guarda en un archivo llamado ejercicio2B.sql las sentencias PL/SQL para OracleBD para modificar las tablas anteriores con las siguientes relaciones entre ellas:

La tabla Pedido y Cliente se relacionan mediante el campo CodCliente.

La tabla Artículo y PedidoLinea se relacionan mediante el campo CodArtículo.

La tabla Pedido y PedidoLinea se relacionan mediante el campo CodPedido.

- C. Obtener desde SQL Developer el esquema de las tablas y relaciones creadas. Adjunta captura del esquema en el documento de Word.

### Ejercicio 3:

- A. Dibuja el diagrama E/R del problema conceptual representado por el modelo físico implementado por las cuatro tablas con sus relaciones creadas en el ejercicio 2, añade el diagrama E/R al documento de Word.  
B. Normaliza el diagrama E/R hasta 3FN (incluye los pasos seguidos en el documento de Word)

### Ejercicio 4:

Crea un script llamado **ejercicio4.sql** y añade las sentencias SQL necesarias para obtener la siguiente información, teniendo en cuenta la estructura de tablas facilitada en el ejercicio 2 y las relaciones entre ellas:

- A. Consulta sql que devuelva el nombre del artículo con alias nombre, el precio del artículo con alias precio ordenado por el precio del artículo con los más caros primero.  
B. Consulta sql que devuelva el nombre del artículo con alias nombre, el precio del artículo con alias precio del artículo más caro que haya en la tabla. Debes usar una subconsulta.  
C. Consulta sql que devuelva el nombre del cliente con alias cliente, el total que ha comprado el cliente en todos sus pedidos con el alias totalcomprado de aquellos clientes cuyo nombre contenga RO o PE independientemente de que esté escrito en mayúsculas o minúsculas.  
D. Consulta sql que devuelva los códigos de pedido que no tienen líneas de pedido.  
E. Consulta sql que devuelva el nombre del cliente, el código de pedido y el total del pedido ordenado por el nombre del cliente y por el código de pedido.

## **BLOQUE SEGUNDA EVALUACIÓN**

**Ejercicio 5:** Crea un archivo **ejercicio5.sql** y agrega las sentencias sql necesarias para dar respuesta a lo solicitado en los siguientes apartados:

- A. Crea la sentencia/sentencias sql necesarias para insertar la siguiente información en las tablas Artículo y Cliente.

#### *Datos tabla Artículo*

codArtículo	nomArtículo	precioart
1	Refresco	1.00
2	Bollo	2.00
3	Pipas	0.50
4	Chicle	0.05
5	RefrescoCola	1.25
6	Bollo Nata	1.95
7	Pipas saladas	0.60
8	Chicle Menta	0.05

#### *Datos tabla Cliente*

CodCliente	NomCliente	CifCliente	fechaAlta	fechaBaja
1	Pepe Perez	A11111111	2022-05-29	NULL
2	Juana Ratos	A22222222	2022-01-02	NULL
3	Kike Rodríguez	C33333333	2022-03-31	NULL

En el cliente código 1 la fecha debe ser la fecha del día en que se produzca la grabación del cliente tomada automáticamente con la función adecuada.

Datos tabla pedidos y sus líneas son:

Primer pedido

fechapedido	nomcliente	nomArticulo	cantidad
2022-05-29	Pepe Perez	Refresco	2
2022-05-29	Pepe Perez	Bollo	4

Segundo pedido

fechapedido	nomcliente	nomArticulo	cantidad
2022-03-04	Juana Ratos	Pipas	3
2022-03-04	Juana Ratos	Refresco	5
2022-03-04	Juana Ratos	Bollo	1

- B. Crea la sentencia/sentencias de actualización necesarias para actualizar el campo *importe* según  $\text{cantidad} \times \text{precio}$  y el campo *totallinea* según  $\text{importe} + \text{importe} \times \text{iva}$ .

**Ejercicio 6:** Crea un archivo **ejercicio6.sql** y agrega las sentencias para dar respuesta a lo solicitado

- A. Crea un procedimiento PL/SQL que permita comprobar ciertos aspectos de los pedidos. El procedimiento debe recibir como parámetro un código de pedido.
- Debe comprobar primero si el pedido tiene líneas. Si el pedido no tiene líneas debe lanzar un error indicando 'El pedido número \_ no tiene líneas'.
  - Si el pedido tiene líneas debe recorrer las líneas del pedido y comprobar para cada línea si el importe es distinto a  $\text{cantidad} \times \text{precio}$  de esa línea. En caso de que sea distinto, debe actualizar el campo *importe* en esa línea al valor correcto ( $\text{cantidad} \times \text{precio}$ ) y mostrar un mensaje que indique que se ha actualizado la línea número \_ del pedido número \_.
  - Escribe las sentencias para llamar al procedimiento con el código de pedido 2.
- B. Crea una función PL/SQL llamada *dameNumPedidos* que reciba como parámetro un código de cliente y devuelva el número de pedidos que tiene ese cliente con líneas.
- Crea un disparador para que cada vez que se inserte o actualice una línea de pedido, se actualicen correctamente el campo *precio* del artículo según el precio que el artículo de la línea tenga en la tabla *artículo*, el *importe* ( $\text{cantidad} \times \text{precio}$ ) y el campo *totallinea* según la información facilitada en el ejercicio 5B.

**Ejercicio 7:** Crea un archivo **ejercicio7.sql** y agrega las sentencias para dar respuesta a lo solicitado:

- A. Crea un usando PL/SQL de OracleDB los siguientes objetos con los atributos y métodos indicados

Persona(Supertipo): dni (texto de longitud 9), nombre (texto de longitud 15), apellidos (texto de longitud 30)

Profesor(Subtipo de persona): especialidad (texto de longitud 30)

Alumno(Subtipo de persona): ciclo (texto de longitud 30), curso (texto de longitud 10)

ListaAlumnos: colección de objetos de tipo alumno

Modulo: nombremod (texto longitud 25), profe (de tipo Profesor), alumnos (de tipo ListaAlumnos)

**B.** Agrega al archivo ejercicio7.sql las sentencias para:

- Añade al tipo de objeto Modulo un constructor personalizado que reciba como parámetros el nombre del módulo, el nombre y apellidos del profesor, el dni del profesor y la especialidad del profesor. Este constructor inicializará todos los atributos necesarios del tipo de objeto Modulo.
- Añade al tipo de objeto Modulo un método llamado anadeAlumno que reciba como parámetros la información necesaria para crear un alumno y se encargue de añadir un alumno con los datos recibidos a la lista de alumnos.
- Crea una tabla para almacenar información de tipos de objeto Modulo e inserta en ella un módulo con los siguientes datos:

Nombre: Bases De Datos

Profesor: Marisa Pelayo Fernández 11111111B Informática

Alumnos:

Pepe Pérez Ruiz 22222222C DAM Primero

Juana Ronda Guti 33333333D DAM Primero

Muestra en la salida DBMS el número de alumnos del módulo de Base de Datos.