

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

Base de Datos - 1INF33
3da. Práctica (Tipo B)
(Segundo Semestre 2024)

Indicaciones generales:

- Duración: 110 minutos
- Pueden usar libros y apuntes de clase, pero no compartirlos.
- El archivo o archivos que contengan sus scripts no deberá ser comprimido.
- Los archivos indicados se subirán a PAIDEIA, en el espacio indicado por los jefes de laboratorio. **Se destinarán los últimos 10 minutos exclusivamente para subir los trabajos a PAIDEIA.**
- Guarde cada uno de sus archivos con el nombre que se le indica. Es importante seguir el estándar del nombre de archivo indicado.
- La presentación del trabajo influye en su calificación.
- Antes de comenzar el laboratorio, ejecute Oracle SQL Developer, cree una nueva conexión llamada **LAB3_CALIFICADA**, y ejecute el script contenido en los archivos:
 1. **1INF33_2024-2_Lab3_Drops.sql**, luego
 2. **1INF33_2024-2_Lab3_DDL.sql** y finalmente
 3. **1INF33_2024-2_Lab3_DML.sql**.

Puntaje: **20 puntos**

Caso propuesto: Producción de buses

Industrias Carrocerías Perú S.A. (INCAPESA) es una empresa dedicada al ensamblaje de ómnibus con motores diésel o motores a gas natural, entre su catálogo de productos destacan buses de transporte público, buses para transporte de personal, buses de transporte interprovincial y buses para el sector turismo. Asimismo, la empresa cuenta con 5 plantas de ensamblaje a nivel nacional que permite abastecer a distintas regiones del país.

Los clientes de INCAPESA son empresas de transporte urbanos, entidades públicas y empresas de transporte de personal, donde cada una de ellas debe tener obligatoriamente número RUC, razón social, dirección fiscal y teléfono de contacto. Además, los pedidos realizados por los clientes deben detallar la cantidad de buses solicitados, el tipo de bus, fecha estimada de entrega de lote y modo de pago.

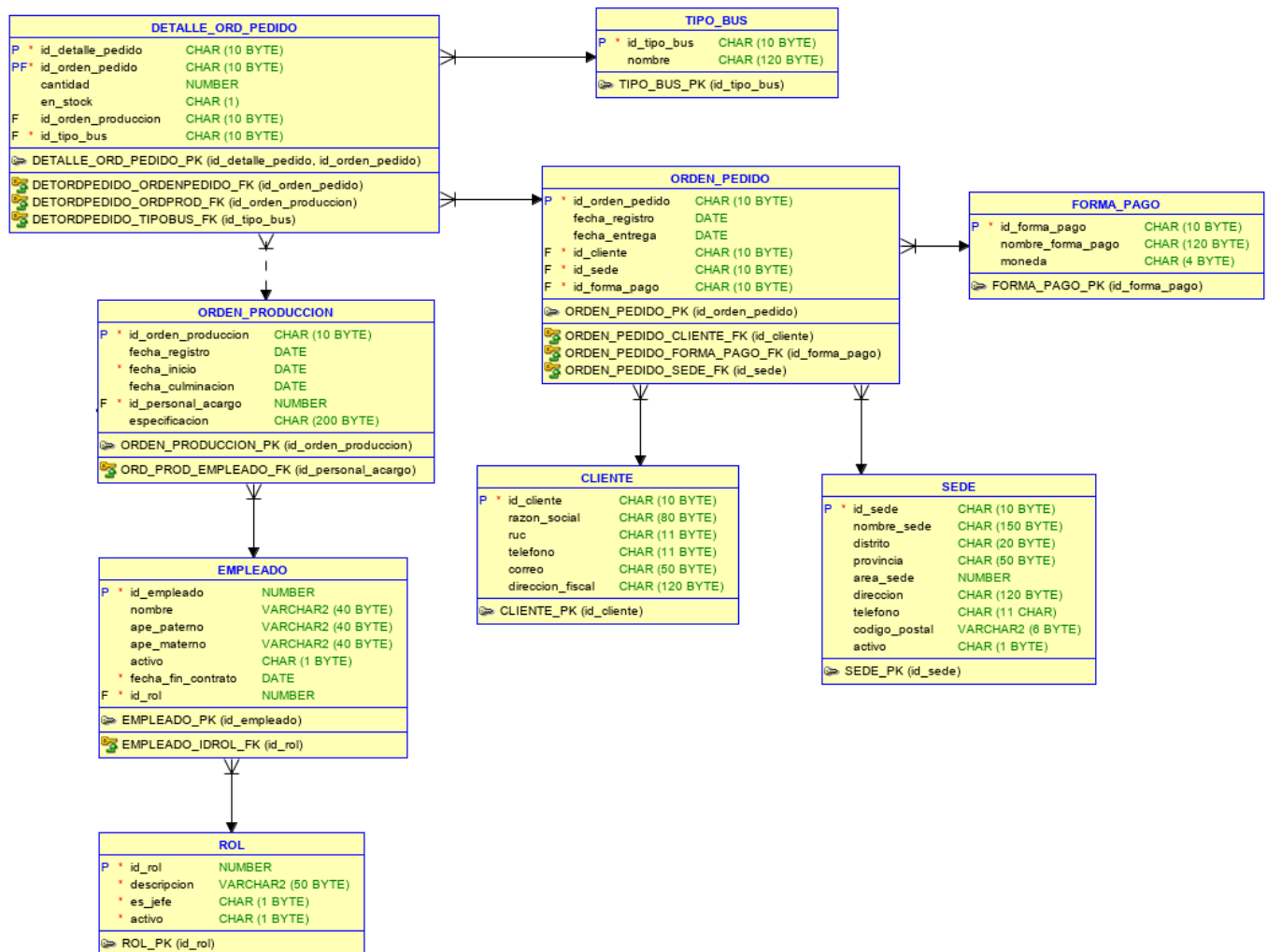
En INCAPESA, una orden de pedido es registrada por un sectorista, la cual es enviada a logística para evaluar si ya se cuenta con producto terminado, es decir buses ensamblados según el pedido del cliente o si es necesario registrar una orden de producción para solicitar los artículos para la fabricación de los buses; entre los principales artículos destacan los siguientes: tipo de motor, chasis, cantidad de asientos de pasajeros, carrocería, pintura, tanque de combustible que puede ser para Diesel o gas natural, faros de iluminación, entre otros.

Los productos de almacén cuentan con un stock de seguridad por cada artículo, es decir nunca debe ser cero, por lo tanto, en caso la cantidad solicitada de artículos rompa el stock de seguridad debe lanzarse una alerta para solicitar la compra del artículo. Asimismo, por cada almacén se cuenta con un historial de la salida de cada artículo, la cual está asociada a una orden de producción de determinada sede.

El horario de producción es de 12 horas al día es decir existen dos turnos y cada uno es de 6 horas, por lo tanto, el almacén estará administrado por un responsable distinto en cada turno, por este motivo en la guía de salida de insumos será necesario precisar el turno de salida de cada insumo ya que el turno estará asociado a un responsable de almacén.

Por ello se debe diseñar una base de datos que permita a la empresa identificar cuál de las sedes es la que produce mayor cantidad de buses de manera que pueda generar una estrategia de abastecimiento oportuno en la fabricación de buses y el tiempo de producción sea menor.

Modelo del caso (solo se muestran las tablas que se necesitarán en este laboratorio)



Pregunta 1 (2 puntos)

Crear la función **“fn_total_buses_tipo”** que reciba el **ID de un tipo de bus** y retorne la **cantidad total de unidades pedidas** de ese tipo. Se debe validar y considerar lo siguiente:

1. Si el tipo de bus no existe se debe retornar 0.
2. La cantidad retornada nunca debe ser nula.

Funciones recomendadas

- NVL(): Para manejar nulos.

Salidas esperadas

```
-- Caso 1: Tipo de bus existente con pedidos
SELECT fn_total_buses_tipo(1) FROM DUAL;
```

FN_TOTAL_BUSES_TIPO(1)	
1	218

```
-- Caso 2: Tipo de bus existente sin pedidos
SELECT fn_total_buses_tipo(5) FROM DUAL;
```

FN_TOTAL_BUSES_TIPO(5)	
1	0

```
-- Caso 3: Tipo de bus que no existe
SELECT fn_total_buses_tipo(999) FROM DUAL;
```

FN_TOTAL_BUSES_TIPO(5)	
1	0

Pregunta 2 (4 puntos)

Crear la función **“fn_eficiencia_entrega_sede”** que calcule el porcentaje de órdenes entregadas a tiempo de una sede. Se debe validar o considerar lo siguiente:

1. La sede debe existir. Caso contrario retornar 0.
2. Si la sede no tiene ordenes, entonces, retornar 0.
3. El porcentaje debe estar entre 0 y 100 sin decimales
4. Solo considerar órdenes con fecha de entrega no nula

Funciones recomendadas

- ROUND(): Para redondear el porcentaje
- NVL(): Para manejar divisiones por cero

Salidas esperadas

<pre>-- Caso 1: Sede con órdenes SELECT fn_eficiencia_entrega_sede(1) FROM DUAL;</pre>	
Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0.007 segundos	
FN_EFICIENCIA_ENTREGA_SEDE(1)	
1	80

<pre>-- Caso 2: Sede sin órdenes SELECT fn_eficiencia_entrega_sede(7) FROM DUAL;</pre>	
Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0.007 segundos	
FN_EFICIENCIA_ENTREGA_SEDE(7)	
1	0

<pre>-- Caso 3: Sede que no existe SELECT fn_eficiencia_entrega_sede(999) FROM DUAL;</pre>	
Resultado de la Consulta x	
SQL Todas las Filas Recuperadas: 1 en 0.007 segundos	
FN_EFICIENCIA_ENTREGA_SEDE(999)	
1	0

Pregunta 3 (3 puntos)

Desarrollar un **procedimiento “sp_registrar_cliente”** que registre un nuevo cliente con todos sus datos. El ID del cliente debe ser pasado como parámetro. Se deba validar o considerar lo siguiente:

1. ID de cliente no debe existir
2. La razón social no puede ser nula o vacía.
3. El RUC no puede ser nulo o vacía.
4. Teléfono no puede ser nulo o vacía.
5. El correo no puede ser nulo o vacía.
6. Se debe imprimir en consola los mensajes de registro exitoso y de validación.
7. El formato de las salidas debe coincidir con lo mostrado en **salidas esperadas**.
8. Debe ejecutar SET SERVEROUTPUT ON; para poder imprimir en pantalla.

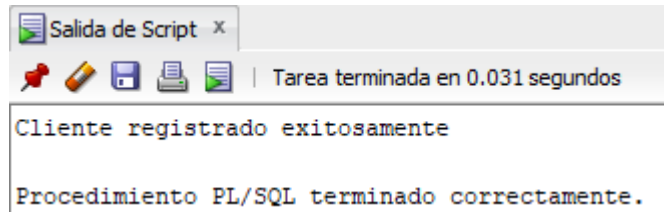
Funciones recomendadas

- NVL(): Para campos opcionales
- LENGTH(): Para validar campos
- DBMS_OUTPUT.PUT_LINE(): Para imprimir en pantalla

Salidas esperadas

1. Caso 1: Cliente válido con todos los datos

```
EXEC sp_registrar_cliente(9, 'Transportes Express', '20505050505', '998877665',  
'contact@express.com', 'Av. Principal 123');
```



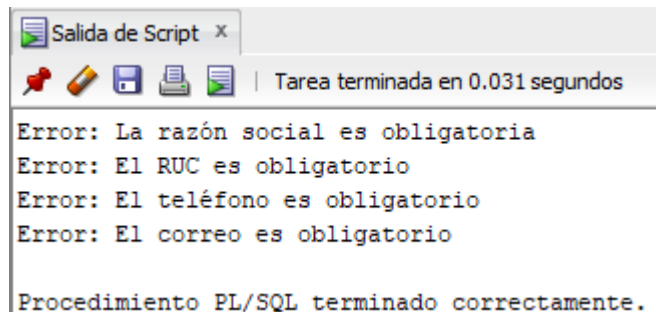
Salida de Script x

Tarea terminada en 0.031 segundos

```
Cliente registrado exitosamente  
  
Procedimiento PL/SQL terminado correctamente.
```

2. Caso 2: Cliente con campos nulos

```
EXEC sp_registrar_cliente(10, NULL, NULL, NULL, NULL, NULL);
```



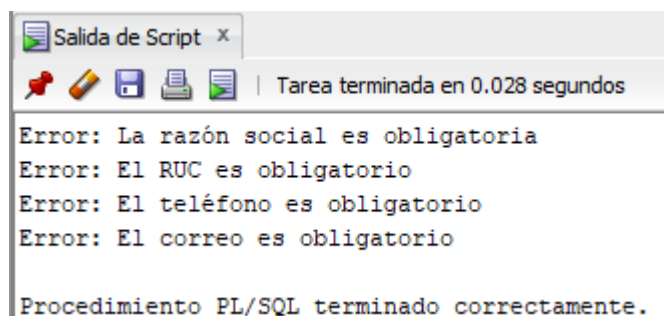
Salida de Script x

Tarea terminada en 0.031 segundos

```
Error: La razón social es obligatoria  
Error: El RUC es obligatorio  
Error: El teléfono es obligatorio  
Error: El correo es obligatorio  
  
Procedimiento PL/SQL terminado correctamente.
```

3. Caso 3: Cliente con campos vacíos

```
EXEC sp_registrar_cliente(11, '', '', '', '', '');
```



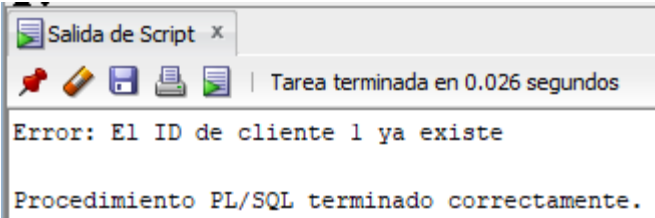
Salida de Script x

Tarea terminada en 0.028 segundos

```
Error: La razón social es obligatoria  
Error: El RUC es obligatorio  
Error: El teléfono es obligatorio  
Error: El correo es obligatorio  
  
Procedimiento PL/SQL terminado correctamente.
```

4. Caso 4: ID existente

```
EXEC sp_registrar_cliente(1, 'Nueva Empresa', '20606060606', '999999999',  
'info@nueva.com', 'Calle Nueva 789');
```



```
Salida de Script x
Tarea terminada en 0.026 segundos

Error: El ID de cliente 1 ya existe

Procedimiento PL/SQL terminado correctamente.
```

Pregunta 4 (3 puntos)

Implementar un **procedimiento "sp_cliente_mas_pedidos"** que imprima en consola el **id_cliente** del cliente que haya realizado más pedidos en un año específico pasado como parámetro. Se debe validar o considerar lo siguiente:

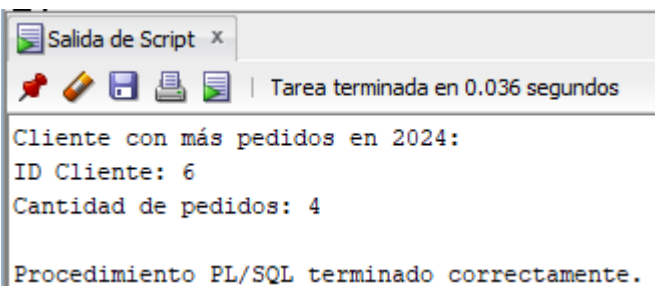
1. El año debe ser válido (no años futuros)
2. Validar que existan pedidos en ese año. Sino existen debe mostrar un mensaje.
3. En caso de empate devolver solo uno de ellos.
4. El formato de las salidas debe coincidir con lo mostrado en **salidas esperadas**.
5. Debe ejecutar SET SERVEROUTPUT ON; para poder imprimir en pantalla.

Funciones recomendadas

- TO_NUMBER(): Para convertir texto a número.
- TO_CHAR(): Para extraer el año de una fecha
- SYSDATE(): Para obtener la fecha actual
- DBMS_OUTPUT.PUT_LINE(): Para imprimir en pantalla

Salidas esperadas

1. Caso 1: Año con pedidos
EXEC sp_cliente_mas_pedidos(2024);

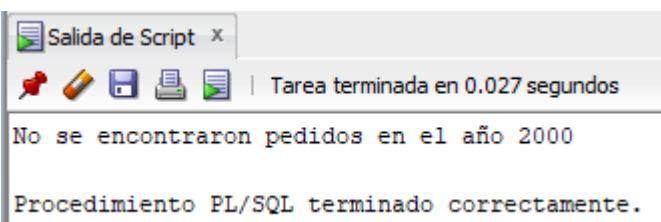


```
Salida de Script x
Tarea terminada en 0.036 segundos

Cliente con más pedidos en 2024:
ID Cliente: 6
Cantidad de pedidos: 4

Procedimiento PL/SQL terminado correctamente.
```

2. Caso 2: Año sin pedidos
EXEC sp_cliente_mas_pedidos(2000);



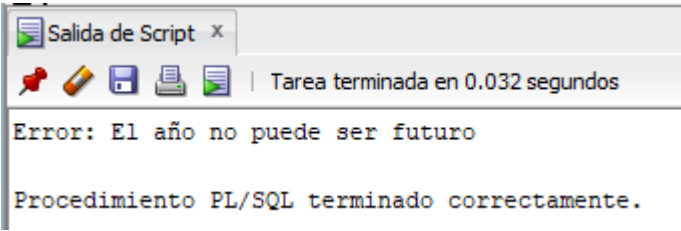
```
Salida de Script x
Tarea terminada en 0.027 segundos

No se encontraron pedidos en el año 2000

Procedimiento PL/SQL terminado correctamente.
```

3. Caso 3: Año futuro

```
EXEC sp_cliente_mas_pedidos(2025);
```



Pregunta 5 (4 puntos)

Implementar un **procedimiento "sp_total_buses_pedidos"** que muestre el total de buses producidos por una sede en un mes específico, y de un tipo de bus específico y mostrando la cantidad. Se debe validar y considerar lo siguiente:

1. La sede debe existir.
2. El tipo de bus debe existir.
3. En caso no haya datos se debe imprimir 0 unidades como total de pedidos.
4. El formato de las salidas debe coincidir con lo mostrado en **salidas esperadas**.
5. Debe ejecutar SET SERVEROUTPUT ON; para poder imprimir en pantalla.

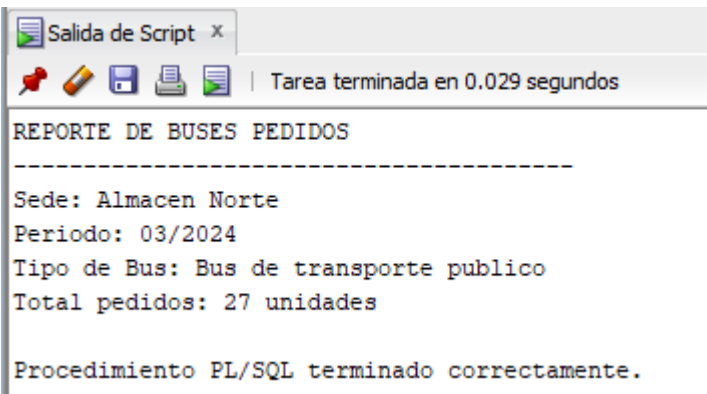
Funciones recomendadas

- TO_CHAR(): Para extraer el año de una fecha
- TRIM(): Para limpiar los espacios vacíos de los nombres.
- SYSDATE(): Para obtener la fecha actual
- NVL(): Para manejar nulos en sumas
- DBMS_OUTPUT.PUT_LINE(): Para imprimir en pantalla

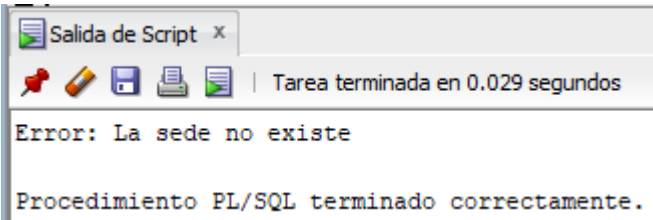
Salidas esperadas

1. Caso 1: Datos válidos existentes

```
EXEC sp_total_buses_pedidos(5, '03/2024', 1);
```



2. Caso 2: Sede que no existe
EXEC sp_total_buses_pedidos(999, '03/2024', 1);



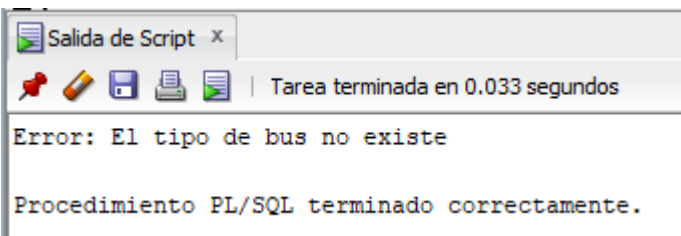
Salida de Script x

Tarea terminada en 0.029 segundos

Error: La sede no existe

Procedimiento PL/SQL terminado correctamente.

3. Caso 3: Tipo de bus que no existe
EXEC sp_total_buses_pedidos(1, '03/2024', 999);



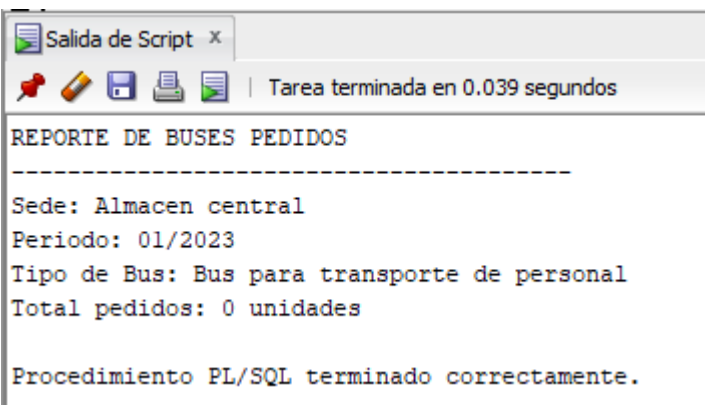
Salida de Script x

Tarea terminada en 0.033 segundos

Error: El tipo de bus no existe

Procedimiento PL/SQL terminado correctamente.

4. Caso 4: Periodo sin datos
EXEC sp_total_buses_pedidos(1, '01/2023', 2);



Salida de Script x

Tarea terminada en 0.039 segundos

REPORTE DE BUSES PEDIDOS

Sede: Almacen central
Periodo: 01/2023
Tipo de Bus: Bus para transporte de personal
Total pedidos: 0 unidades

Procedimiento PL/SQL terminado correctamente.

Pregunta 6 (4 puntos)

Crear un **procedimiento "sp_resumen_cliente"** que muestre el resumen de un cliente específico. Se debe validar y considerar lo siguiente:

1. El cliente debe existir.
2. En caso el cliente no tenga pedidos se debe mostrar un mensaje.
3. El formato de las salidas debe coincidir con lo mostrado en **salidas esperadas**.
4. Debe ejecutar SET SERVEROUTPUT ON; para poder imprimir en pantalla.

Funciones recomendadas

- TO_CHAR(): Para formato de fechas y números
- NVL(): Para manejar nulos
- SUM(): Para totales
- MAX(): Para última fecha
- DBMS_OUTPUT.PUT_LINE(): Para imprimir en pantalla

Salidas esperadas

1. Caso 1: Cliente con varios pedidos
EXEC sp_resumen_cliente(4);

```
Salida de Script x
Tarea terminada en 0.029 segundos

RESUMEN COMERCIAL DE CLIENTE
-----
Cliente: Universidad del Centro
Total buses pedidos: 112
Último pedido: 13/09/2024

Procedimiento PL/SQL terminado correctamente.
```

2. Caso 2: Cliente sin pedidos
EXEC sp_resumen_cliente(8);

```
Salida de Script x
Tarea terminada en 0.026 segundos

RESUMEN COMERCIAL DE CLIENTE
-----
Cliente: Inversiones Group Lima
Total buses pedidos: 0
Último pedido: Sin pedidos registrados

Procedimiento PL/SQL terminado correctamente.
```

3. Caso 3: Cliente que no existe
EXEC sp_resumen_cliente(999);

```
Salida de Script x
Tarea terminada en 0.03 segundos

Error: Cliente con ID 999 no existe

Procedimiento PL/SQL terminado correctamente.
```

Guarde un único archivo con el siguiente formato: **L3_codigoalumno.sql** (Ejemplo: **L3_20241234.sql**)

Dentro del archivo indicar en un comentario qué número de pregunta es, por ejemplo:

```
-- Pregunta 1  
(solución de la pregunta 1)  
  
-- Pregunta 2  
(solución de la pregunta 2)  
  
-- Pregunta 3  
(solución de la pregunta 3)
```

25 de noviembre de 2024

Elaborado por DBA - DM