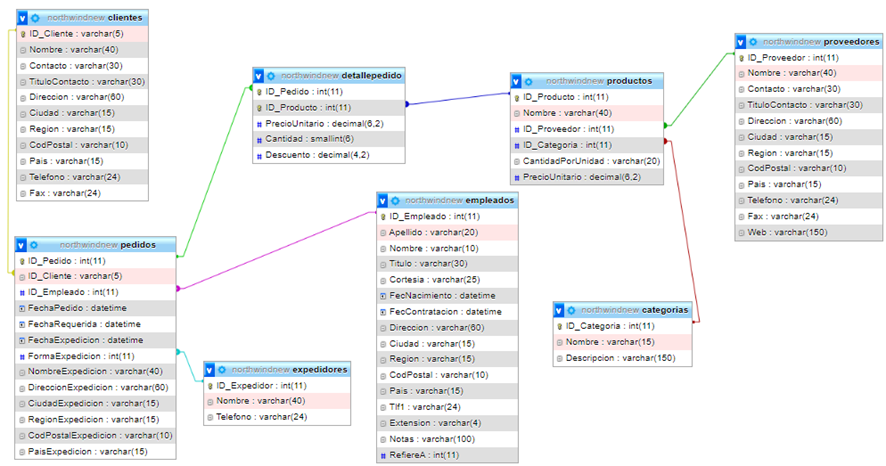
# CASO PRÁCTICO 1

**CONSULTAS AVANZADAS Y TRANSACCIONES**

## Contexto

En tu empresa, realizas múltiples consultas en bases de datos internas, y muchas de ellas son consultas avanzadas. Esta semana se ha incorporado a tu departamento un desarrollador nuevo, y te han pedido que lo ayudes. Para ello, y dada la siguiente estructura de base de datos:



Para crear las tablas e insertar los registros para la tarea, puedes ejecutar el script adjunto: [Recurso caso](https://cesurformacion0.sharepoint.com/:u:/s/RepoRecursosContenidos/EfqIzkhoPw1PlqGekhi4w00Bu_2kQs-e-ySD56dV3luClw?e=BsMooS)

Te piden que resuelvas los siguientes ejercicios:

1) Mostrar: código de la categoría, el nombre de la categoría, cod.Producto, nombre del producto y precio.

SELECT c.ID\_Categoria, c.Nombre AS NombreCategoria, p.ID\_Producto, p.Nombre AS NombreProducto, p.PrecioUnitario

FROM Categorias c

INNER JOIN Productos p ON c.ID\_Categoria = p.ID\_Categoria;  
  
2) Mostrar: código, nombre, ciudad y país de proveedor, código, nombre y precio del producto.SELECT pr.ID\_Proveedor, pr.Nombre, pr.Ciudad, pr.Pais, p.ID\_Producto, p.Nombre AS NombreProducto, p.PrecioUnitario

FROM Proveedores pr

INNER JOIN Productos p ON pr.ID\_Proveedor = p.ID\_Proveedor;

3) Mostrar: núm. de la orden, fecha, persona de contacto del cliente y el nombre y apellidos del empleado  
SELECT o.ID\_Pedido, o.FechaPedido, c.Contacto, e.Nombre, e.Apellido

FROM Pedidos o

INNER JOIN Clientes c ON o.ID\_Cliente = c.ID\_Cliente

INNER JOIN Empleados e ON o.ID\_Empleado = e.ID\_Empleado;

4) Visualizar el mínimo precio de los productos, mostrar el nombre de la categoría y el productoSELECT MIN(p.PrecioUnitario) AS MinimoPrecio, c.Nombre AS NombreCategoria, p.Nombre AS NombreProducto

FROM Productos p

INNER JOIN Categorias c ON p.ID\_Categoria = c.ID\_Categoria

GROUP BY c.Nombre, p.Nombre;

5) Seleccionar las categorías que tengan más de 5 productos. Mostrar el nombre de la categoría y el número de productos  
  
SELECT c.Nombre, COUNT(p.ID\_Producto) AS NumeroProductos

FROM Categorias c

INNER JOIN Productos p ON c.ID\_Categoria = p.ID\_Categoria

GROUP BY c.Nombre

HAVING COUNT(p.ID\_Producto) > 5;

6) Mostrar el nombre de todos los empleados y el número de órdenes que han hecho.

SELECT e.Nombre, e.Apellido, COUNT(o.ID\_Pedido) AS NumeroOrdenes

FROM Empleados e

LEFT JOIN Pedidos o ON e.ID\_Empleado = o.ID\_Empleado

GROUP BY e.Nombre, e.Apellido;

7) Mostrar el nombre y el precio del producto sólo cuando dicho precio sea igual o superior al precio medio de los productos.

SELECT Nombre, PrecioUnitario

FROM Productos

WHERE PrecioUnitario >= (SELECT AVG(PrecioUnitario) FROM Productos);

8) Mostrar el precio medio de los productos agrupados por categorías solo cuando dicho precio medio sea mayor que el precio medio de los productos de la categoría 4. Hazlo usando la cláusula HAVING

SELECT c.Nombre AS NombreCategoria, AVG(p.PrecioUnitario) AS PrecioUnitarioMedio

FROM Productos p

INNER JOIN Categorias c ON p.ID\_Categoria = c.ID\_Categoria

GROUP BY c.Nombre

HAVING AVG(p.PrecioUnitario) > (SELECT AVG(PrecioUnitario) FROM Productos WHERE ID\_Categoria = 4);

9) Obtener el Id del empleado, el nombre y el campo ID\_Pedido para los empleados cuyo id de empleado sea mayor que 5. Ten en cuenta que en el caso de que algún empleado no haya registrado ningún pedido también querremos que aparezca. Para estos casos, haz que en el campo ORDER\_ID aparezca un 0 en lugar de NULL.  
  
SELECT e.ID\_Empleado, e.Nombre, IFNULL(o.ID\_Pedido, 0) AS ORDER\_ID

FROM Empleados e

LEFT JOIN Pedidos o ON e.ID\_Empleado = o.ID\_Empleado

WHERE e.ID\_Empleado > 5;

10) Haz lo mismo que en el apartado anterior, pero para los clientes que no han realizado ningún pedido. Haz que en lugar de mostrar el campo ORDER\_ID muestre el valor “SIN PEDIDO”

SELECT c.ID\_Cliente, c.Nombre, IFNULL(o.ID\_Pedido, 'SIN PEDIDO') AS ORDER\_ID

FROM Clientes c

LEFT JOIN Pedidos o ON c.ID\_Cliente = o.ID\_Cliente

WHERE o.ID\_Pedido IS NULL;

11) Crea una transacción en la que, sobre la tabla empleados, incrementes en 100 el salario del empleado con ID = 4 y reduzcas en 100 el salario del empleado con ID = 8. Por último, deshaz los cambios para finalizar la transacción.  
  
START TRANSACTION;

UPDATE Empleados SET Salario = Salario + 100 WHERE ID\_Empleado = 4;

UPDATE Empleados SET Salario = Salario - 100 WHERE ID\_Empleado = 8;  
ROLLBACK

## Cuestiones a resolver

Resuelve los ejercicios planteados, teniendo en cuenta estos aspectos:

1. Utilizarás phpMyAdmin de XAMPP o MySQL Workbench para trabajar con SQL y probar tus sentencias.
2. Una vez probada, copiarás las sentencias que quieras entregar a este documento Word, en los recuadros correspondientes a cada apartado.
3. Cambia el color de la fuente a azul fuerte para poder distinguir el código SQL mejor dentro del Word.
4. Pasa el documento a PDF renombrándolo de la siguiente forma:
   * DAM-BD-UD0501-Apellidos, nombre.pdf
   * DAM-BD-UD0501-Pérez Valera, David.pdf
   * DAW-BD-UD0501-Apellidos, nombre.pdf
   * DAW-BD-UD0501- Pérez Valera, David.pdf

## Recursos

Se deberá consultar el contenido de la unidad, internet, libros, revistas y utilizar medios informáticos para la presentación del caso práctico (Word, Power-Point…)

## Objetivos

Llevar a cabo consultas a más de una tabla y aplicar los diferentes tipos de composiciones. Realizar consultas con subconsultas e implementar transacciones.

## Resultados de aprendizaje y criterios de evaluación

Consulta la información almacenada en una base de datos empleando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

* Se han identificado las herramientas y sentencias para realizar consultas.
* Se han realizado consultas simples sobre una tabla.
* Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas.
* Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas.
* Se han realizado consultas resumen.
* Se han realizado consultas con subconsultas.
* Se han realizado consultas que implican múltiples selecciones

Modifica la información almacenada en la base de datos utilizando asistentes, herramientas gráficas y el lenguaje de manipulación de datos.

* Se han diseñado guiones de sentencias para llevar a cabo tareas complejas.
* Se ha reconocido el funcionamiento de las transacciones.
* Se han anulado parcial o totalmente los cambios producidos por una transacción.
* Se han identificado los efectos de las distintas políticas de bloqueo de registros.
* Se han adoptado medidas para mantener la integridad y consistencia de la información.

# RUBRICA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CRITERIOS | Excelente | Notable | Satisfactorio | Insuficiente |
| Calidad de las consultas SQL  40% | Proporciona consultas SQL precisas y optimizadas para todos los ejercicios | Proporciona consultas SQL correctas para la mayoría de los ejercicios, pero algunas pueden no estar optimizadas | Proporciona consultas SQL correctas para algunos ejercicios, pero otras pueden ser incorrectas o faltar | No proporciona consultas SQL correctas para la mayoría de los ejercicios |
|  | 4.0 puntos | 3.0 puntos | 2.0 puntos | 0 puntos |
| Comprensión de las transacciones SQL  20% | Implementa correctamente las transacciones SQL, demostrando una comprensión sólida de su funcionamiento | Demuestra una comprensión adecuada de las transacciones SQL, con algunos errores menores | Implementa parcialmente las transacciones SQL pero muestra signos de comprensión limitada | No implementa correctamente las transacciones SQL ni demuestra comprensión de su funcionamiento |
|  | 2.0 puntos | 1.5 puntos | 1.0 punto | 0 puntos |
| Uso de la cláusula HAVING y subconsultas  20% | Utiliza correctamente la cláusula HAVING y las subconsultas en las consultas SQL | Utiliza principalmente la cláusula HAVING y las subconsultas correctamente, pero hay algunos errores | Utiliza la cláusula HAVING y las subconsultas en algunas consultas, pero hay un uso inadecuado o ausente en otras | No utiliza correctamente la cláusula HAVING ni las subconsultas en las consultas SQL |
|  | 2.0 puntos | 1.5 puntos | 1.0 punto | 0 puntos |
| Presentación, redacción y ortografía  10% | Presenta un discurso ordenado y comprensible, profundizando en todos los conceptos. Además, no se aprecian errores de gramática, ortografía o puntuación. | Presenta un discurso ordenado y comprensible, pero aparecen algunos errores de gramática, ortografía o puntuación. | Presenta un discurso comprensible, aunque algo desordenado y sin profundizar en conceptos e ideas y/o se observan varios errores ortográficos, de gramática o de puntuación. | Presenta un discurso desordenado que dificulta la comprensión de los conceptos e ideas que se exponen y/o aparecen numerosos errores gramaticales, de ortografía o puntuación. |
|  | 1 punto | 0.75 puntos | 0.5 puntos | 0 puntos |
| Uso de recursos adicionales y creatividad  10% | Responde con gran originalidad, expresando ideas creativas e ingeniosas. Utiliza numerosas fuentes de información relevantes, fiables y actualizadas. Aporta imágenes, gráficos y recursos que clarifican la respuesta. | Muestra cierta originalidad en su respuesta. Utiliza diversas fuentes de información, aunque no todas son relevantes. Aporta alguna imagen, gráfico o recurso adicional. | Utiliza alguna fuente de información. Hace uso de cita de ideas de autores, pero no se aportan ideas y puntos de vista originales propios. | No hace uso de fuentes fiables ni añade recursos o se utilizan ideas de otros autores sin citar. |
|  | 1 punto | 0.75 puntos | 0.5 puntos | 0 puntos |