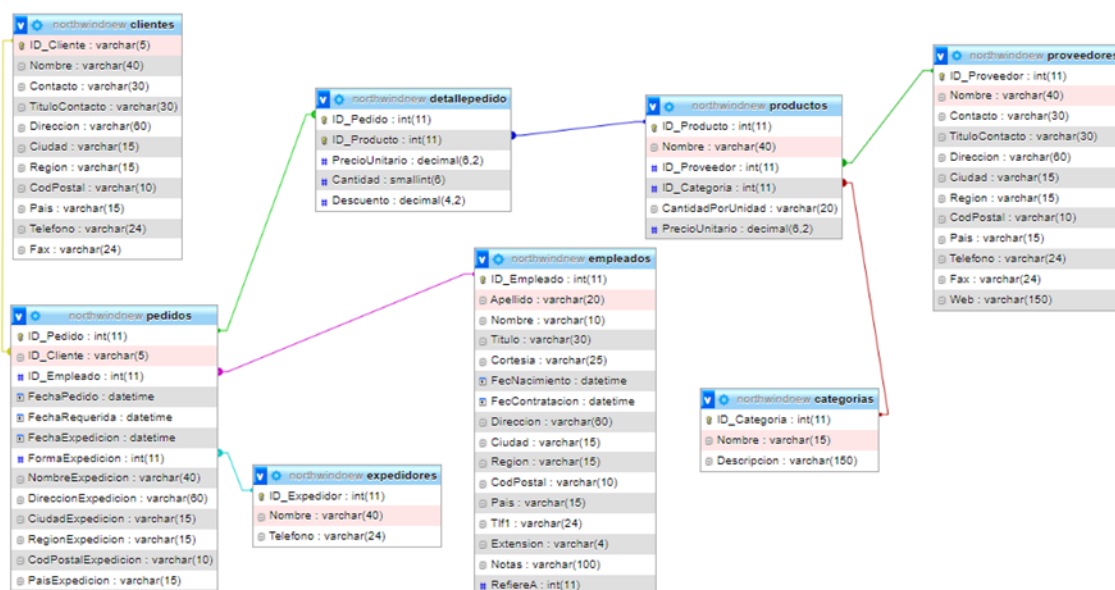


## CASO PRÁCTICO 1

### 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](#)

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,
    p.ID_Producto,
    p.Nombre,
    p.PrecioUnitario
FROM Productos p
JOIN Categorías c ON p.ID_Categoria = c.ID_Categoria;
```

2) Mostrar: código, nombre, ciudad y país de proveedor, código, nombre y precio del producto.

```
SELECT
    pr.ID_Proveedor,
    pr.Nombre AS Nombre_Proveedor,
    pr.Ciudad,
    pr.Pais,
    po.Nombre AS Nombre_Producto,
    po.PrecioUnitario
FROM Proveedores pr
JOIN Productos po ON pr.ID_Proveedor = po.ID_Proveedor;
```

3) Mostrar: núm. de la orden, fecha, persona de contacto del cliente y el nombre y apellidos del empleado

```
SELECT
    pe.ID_Pedido AS Numero_Orden,
    pe.FechaPedido,
    cl.Contacto AS Persona_Contacto,
    e.Nombre,
    e.Apellido
FROM Pedidos pe
JOIN Clientes cl ON pe.ID_Cliente = cl.ID_Cliente
JOIN Empleados e ON pe.ID_Empleado = e.ID_Empleado;
```

4) Visualizar el mínimo precio de los productos, mostrar el nombre de la categoría y el producto

```
SELECT
    pr.PrecioUnitario AS Precio_Minimo,
    pr.Nombre AS Nombre_Producto,
    ct.Nombre AS Nombre_Categoria
FROM Productos pr
JOIN Categorías ct ON pr.ID_Categoria = ct.ID_Categoria
ORDER BY pr.PrecioUnitario ASC
LIMIT 1;
```

5) Seleccionar las categorías que tengan más de 5 productos. Mostrar el nombre de la categoría y el número de productos

```
SELECT
    c.Nombre AS Nombre_Categoria,
    COUNT(p.ID_Producto) AS Numero_Productos
FROM Categorías c
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 AS Nombre_Empleado,
    COUNT(p.ID_Pedido) AS Numero_Ordenes
FROM Empleados e
LEFT JOIN Pedidos p ON e.ID_Empleado = p.ID_Empleado
GROUP BY e.Nombre;
```

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

```
SELECT
    p.Nombre AS Producto,
    p.PrecioUnitario
FROM Productos p
WHERE p.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 Nombre_Categoria,
    AVG(p.PrecioUnitario) AS Precio_Medio
FROM Productos p
JOIN Categorías 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 queremos 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 AS Nombre_Empleado,
    COALESCE(p.ID_Pedido, 0) AS ID_Pedido
FROM Empleados e
LEFT JOIN Pedidos p ON e.ID_Empleado = p.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 AS Nombre_Cliente,
    COALESCE(p.ID_Pedido, 'SIN PEDIDO') AS ID_Pedido
FROM Clientes c
LEFT JOIN Pedidos p ON c.ID_Cliente = p.ID_Cliente;
```

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.

```
BEGIN;
```

```
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