# NorthWind Database

**Tables and Their Structure**

**Categories**

* Stores information about product categories.
* Columns: CategoryID (Primary Key), CategoryName, Description.

**Customers**

* Stores information about customers.
* Columns: CustomerID (Primary Key), CustomerName, ContactName, Address, City, PostalCode, Country.

**Employees**

* Stores details about employees.
* Columns: EmployeeID (Primary Key), LastName, FirstName, BirthDate, Photo, Notes.

**Shippers**

* Stores information about shipping companies.
* Columns: ShipperID (Primary Key), ShipperName, Phone.

**Suppliers**

* Stores details about suppliers who provide products.
* Columns: SupplierID (Primary Key), SupplierName, ContactName, Address, City, PostalCode, Country, Phone.

**Products**

* Stores information about products offered by suppliers.
* Columns: ProductID (Primary Key), ProductName, SupplierID (Foreign Key), CategoryID (Foreign Key), Unit, Price.
* Foreign Keys:
  + SupplierID references Suppliers(SupplierID).
  + CategoryID references Categories(CategoryID).

**Orders**

* Stores information about customer orders.
* Columns: OrderID (Primary Key), CustomerID (Foreign Key), EmployeeID (Foreign Key), OrderDate, ShipperID (Foreign Key).
* Foreign Keys:
  + CustomerID references Customers(CustomerID).
  + EmployeeID references Employees(EmployeeID).
  + ShipperID references Shippers(ShipperID).

**OrderDetails**

* Stores detailed information about products in each order.
* Columns: OrderDetailID (Primary Key), OrderID (Foreign Key), ProductID (Foreign Key), Quantity.
* Foreign Keys:
  + OrderID references Orders(OrderID).
  + ProductID references Products(ProductID).

--------------------------------------------------------------------------------------------------------------------------------------**Relationships**

* The schema defines several key relationships:
  + **One-to-Many:** Between Categories and Products, Suppliers and Products, Customers and Orders, Employees and Orders.
  + **Many-to-Many:** Between Orders and Products through the OrderDetails table.

**Use Cases**

* This database structure supports a variety of use cases, such as tracking orders, managing customer information, maintaining product catalogs, and handling supplier relationships. It allows for complex queries involving joins between multiple tables to fetch relevant information.