8. Write an SQL query that demonstrates the use of various types of joins (INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN) using the Products and Order Details tables.

1. CREATE TABLE Products (ProductID INT PRIMARY KEY, ProductName VARCHAR(255) NOT NULL, Category VARCHAR(100), Price DECIMAL(10,2), StockQuantity INT);

0. **Order Details table has the columns Order Detail ID, Order ID, Product ID, Quantity, and Unit Price.**

a. INNER JOIN

Returns only the matching rows between Products and OrderDetails i.e., products that have been ordered.

```
SELECT p.ProductID, p.ProductName, od.OrderID, od.Quantity, od.UnitPrice
FROM Products p
INNER JOIN OrderDetails od ON p.ProductID = od.ProductID;
```

b. LEFT JOIN

Returns all products, whether they have been ordered or not. If a product has no orders, OrderID, Quantity, and UnitPrice will be NULL.

```
SELECT p.ProductID, p.ProductName, od.OrderID, od.Quantity, od.UnitPrice
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID;
```

c. RIGHT JOIN

Returns all order details, including those that reference a non-existent product.

```
SELECT p.ProductID, p.ProductName, od.OrderID, od.Quantity, od.UnitPrice
FROM Products p
RIGHT JOIN OrderDetails od ON p.ProductID = od.ProductID;
```

d. FULL OUTER JOIN

Combines the results of LEFT JOIN and RIGHT JOIN, ensuring all products and orders are included.

```
SELECT p.ProductID, p.ProductName, od.OrderID, od.Quantity, od.UnitPrice
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
UNION ALL
SELECT p.ProductID, p.ProductName, od.OrderID, od.Quantity, od.UnitPrice
FROM Products p
RIGHT JOIN OrderDetails od ON p.ProductID = od.ProductID;
```