HW - VITS Queries:

1. Show me distinct city that all supplier supplies :

```
SELECT DISTINCT City
FROM Supplier;
```

2. Show me products with price less than 15:

```
SELECT *
FROM Products
WHERE Price < 15;
```

3. **Show me suppliers in Osaka, Paris, or Berlin:**

```
SELECT *
FROM Suppliers
WHERE City IN ('Osaka', 'Paris', 'Berlin');
```

4. Show me products sorted by SupplierID and then Price descending:

```
SELECT *
FROM Products
ORDER BY SupplierID, Price DESC
```

5. Insert a product in Product table(insert):

```
INSERT INTO Product (ProductName, SupplierID, CategoryID, QuantityPerUnit, Price)
VALUES ('VITS Product', 1, 2, '10 boxes', 12.50);
```

6. Show me customers that live in countries where country names has the text ‘lan’ in them(use like/wildcard in statement)

```
SELECT *
FROM Customers
WHERE Country LIKE '%lan%';
```

7. Show me productname, price with max price(use max function):

```
SELECT ProductName, Price
FROM Product
WHERE Price = (SELECT MAX(Price) FROM Products);
```

8. Show me all products with column name changed `Pname` instead of `ProductName` (use alias) :

```
SELECT ProductName AS Pname, Price, CategoryID, SupplierID
FROM Products;
```

9. Show count of suppliers by country:

```
SELECT Country, COUNT(*)
FROM Suppliers
GROUP BY Country;
```

10. Show count of products in every order (`OrderDetails` table):

```
SELECT OrderID, COUNT(*) AS ProductCount
FROM OrderDetails
GROUP BY OrderID;
```

11. Report to show: OrderID, OrderDate, ShipperName (Inner Join):

```
SELECT o.OrderID, o.OrderDate, s.ShipperName
FROM Orders o
INNER JOIN Shippers s ON o.ShipperID = s.ShipperID;
```

12. Report: OrderID, OrderDate, ProductName, CategoryName, Price (all orders where product price is less then 15) (inner join) :

```
SELECT o.OrderID, o.OrderDate, p.ProductName, c.CategoryName, p.Price
FROM Orders o
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Products p ON od.ProductID = p.ProductID
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
WHERE p.Price < 15;
```

13. Report: show me Products that have not been ordered yet `ProductName`, `OrderID` (NULL) - (left join):

```
SELECT p.ProductName, od.OrderID
FROM Products p
LEFT JOIN OrderDetails od ON p.ProductID = od.ProductID
WHERE od.OrderID IS NULL;
```

14. Report: show me the Top 10 most ordered products (join / group by) - sum of qty :

```
SELECT p.ProductName, SUM(od.Quantity) AS TotalQuantity
FROM OrderDetails od
INNER JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalQuantity DESC
LIMIT 10;
```

15. Report:show me the Top 10 most revenue-generating products (join / group by) - (`Revenue = Quantity * Price`):

```
SELECT p.ProductName, SUM(od.Quantity * od.Price) AS TotalRevenue
FROM OrderDetails od
INNER JOIN Products p ON od.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalRevenue DESC
LIMIT 10;
```

16. Report: show me the Top 10 customers by revenue(join / group by):

```
SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity * od.Price) AS TotalRevenue
FROM Orders o
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY c.CustomerID, c.CustomerName
ORDER BY TotalRevenue DESC
LIMIT 10;
```