Structure of Products table:

products

- ProductID INT(11)
- ProductName VARCHAR(40)
- SupplierID INT(11)
- CategoryID INT(11)
- QuantityPerUnit VARCHAR(20)
- UnitPrice DECIMAL(10,4)
- UnitsInStock SMALLINT(2)
- UnitsOnOrder SMALLINT(2)
- ReorderLevel SMALLINT(2)
- Discontinued BIT(1)

Indexes

PRIMARY
ProductName
FK_Products_Categories
FK_Products_Suppliers

Sample records of Products Table:

```
+-----------+------------------------------------+------------+------------+----------
| ProductID | ProductName                        | SupplierID | CategoryID | Quantity
+-----------+------------------------------------+------------+------------+----------
|         1 | Chai                               |          1 |          1 | 10 boxes
|         2 | Chang                              |          1 |          1 | 24 - 12
|         3 | Aniseed Syrup                      |          1 |          2 | 12 - 550
|         4 | Chef Anton's Cajun Seasoning       |          2 |          2 | 48 - 6 c
|         5 | Chef Anton's Gumbo Mix             |          2 |          2 | 36 boxes
|         6 | Grandma's Boysenberry Spread       |          3 |          2 | 12 - 8 c
|         7 | Uncle Bob's Organic Dried Pears    |          3 |          7 | 12 - 1 l
|         8 | Northwoods Cranberry Sauce         |          3 |          2 | 12 - 12
|         9 | Mishi Kobe Niku                    |          4 |          6 | 18 - 500
|        10 | Ikura                              |          4 |          8 | 12 - 200
|        11 | Queso Cabrales                     |          5 |          4 | 1 kg pkg
|        12 | Queso Manchego La Pastora          |          5 |          4 | 10 - 500
|        13 | Konbu                              |          6 |          8 | 2 kg box
|        14 | Tofu                               |          6 |          7 | 40 - 100
|        15 | Genen Shouyu                       |          6 |          2 | 24 - 250
....
+-----------+------------------------------------+------------+------------+----------
```

```
+------------+---------------+---------------+--------------+--------------+
| UnitPrice  | UnitsInStock  | UnitsOnOrder  | ReorderLevel | Discontinued |
+------------+---------------+---------------+--------------+--------------+
|    18.0000 |           39  |            0  |          10  |              |
|    19.0000 |           17  |           40  |          25  |              |
|    10.0000 |           13  |           70  |          25  |              |
|    22.0000 |           53  |            0  |           0  |              |
|    21.3500 |            0  |            0  |           0  | ©            |
|    25.0000 |          120  |            0  |          25  |              |
|    30.0000 |           15  |            0  |          10  |              |
|    40.0000 |            6  |            0  |           0  |              |
|    97.0000 |           29  |            0  |           0  | ©            |
|    31.0000 |           31  |            0  |           0  |              |
|    21.0000 |           22  |           30  |          30  |              |
|    38.0000 |           86  |            0  |           0  |              |
|     6.0000 |           24  |            0  |           5  |              |
|    23.2500 |           35  |            0  |           0  |              |
|    15.5000 |           39  |            0  |           5  |              |
```

2. Write a query to get Product name and quantity/unit.

```sql
SELECT ProductName, QuantityPerUnit
 FROM Products;
```

Write a query to get current Product list (Product ID and name).

```sql
SELECT ProductID, ProductName
FROM Products
WHERE Discontinued = "False"
ORDER BY ProductName;
```

3. Write a query to get most expense and least expensive Product list (name and unit price).

```sql
SELECT ProductName, UnitPrice
FROM Products
ORDER BY UnitPrice DESC;
```

4. Write a query to get Product list (id, name, unit price) where current products cost less than $20.

```sql
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE (((UnitPrice)<20) AND ((Discontinued)=False))
ORDER BY UnitPrice DESC;
```

5. Write a query to get Product list (name, unit price) where products cost between $15 and $25.

```sql
SELECT ProductName, UnitPrice
FROM Products
WHERE (((UnitPrice)>=15 And (UnitPrice)<=25)
AND ((Products.Discontinued)=False))
```

```sql
ORDER BY Products.UnitPrice DESC;
```

6. Write a query to get Product list (name, unit price) of above average price.

```sql
SELECT DISTINCT ProductName, UnitPrice
FROM Products
WHERE UnitPrice > (SELECT avg(UnitPrice) FROM Products)
ORDER BY UnitPrice;
```