User-Defined Functions (UDF) Practice Questions

AdventureWorks2022 — Beginner to Advanced

A complete set of hands-on practice tasks to master scalar, inline table-valued, and multi-statement table-valued functions using the **AdventureWorks2022** database.

▧ SECTION 1 — Beginner Level (Scalar Functions)

1. Full Name Formatter

Create a scalar UDF that accepts FirstName, MiddleName, LastName and returns:

```
LastName, FirstName MiddleName
```

Apply it to `Person.Person`.

---

2. Calculate Product Discount

Create a scalar function:

```
dbo.CalculateDiscount(@ListPrice, @DiscountPercent)
```

Use it on `Production.Product` to compute discount at **10%**.

---

3. Product Weight Classification

Create a UDF that returns:

- "Light" (< 2)
- "Medium" (2–10)
- "Heavy" (> 10)
- "Unknown" (NULL)

Apply to `Production.Product`.

---

4. Convert Date to YYYYMM Format

Create `dbo.ToYYYYMM(@Date)` and use it on all dates in `Sales.SalesOrderHeader`.

---

5. High Value Customer Check

Create a UDF returning "Yes" if total purchases > 50,000 for a given CustomerID. Use on `Sales.Customer`.

▨ SECTION 2 — Intermediate Level (Inline Table-Valued Functions)

6. Products by Subcategory

Create an inline TVF:

```
dbo.GetProductsBySubCategory(@SubcategoryID)
```

Return ProductID, Name, ListPrice. Query it for SubcategoryID = 1.

---

7. Orders in Date Range

Create TVF:

```
dbo.GetOrdersByDateRange(@StartDate, @EndDate)
```

Return rows from `Sales.SalesOrderHeader`. Query for orders in 2020.

---

8. Employees in a Department

Create TVF:

```
dbo.GetEmployeesByDepartment(@DepartmentName)
```

Use: `EmployeeDepartmentHistory`, `Department`, `Person`. Return EmployeeID, FullName, Department, StartDate. Query for "Engineering".

---

9. Active Products (ListPrice > 0)

Create TVF returning ProductID, Name, Color, ListPrice. Join with `Production.ProductModel`.

---

10. Top N Most Expensive Products

Create TVF:

```
dbo.TopNProducts(@TopN)
```

Return top N products by ListPrice.

▨ SECTION 3 — Advanced Level (Multi-Statement TVFs)

11. Monthly Sales Summary (By Year)

Create mTVF:

```
dbo.MonthlySalesSummary(@Year)
```

Return MonthNumber, TotalSalesAmount, TotalOrders. Use data from `Sales.SalesOrderHeader`. Query for 2021.

---

12. Customer Order Statistics

Create function:

```
dbo.CustomerOrderStats(@CustomerID)
```

Return TotalOrders, TotalAmount, LastOrderDate. Query for CustomerID = 29847.

---

13. Low Stock Items

Low Stock = Quantity < 100. Use `Production.ProductInventory`. Return ProductID, TotalStock.

---

14. Inactive Employees (Last 3 Years)

Employee considered inactive if not present in `EmployeeDepartmentHistory` in last 3 years. Return EmployeeID, FullName, LastDepartment, LastEndDate.

---

15. Customers and Their First Purchase

Create function returning:

- CustomerID
- FirstOrderDate
- FirstOrderTotal

Use data from `Sales.SalesOrderHeader`.

▧ SECTION 4 — Expert Level (Mixed Scenarios & Performance)

16. Email Masker Function

Create scalar UDF masking an email such as:

```
john.smith@example.com → j***@example.com
```

Use on `Person.EmailAddress`.

---

17. Flexible Product Filters

Create inline TVF:

```
dbo.FilterProducts(@Color, @MinPrice, @MaxPrice)
```

Return all products matching filters. Call with:

- Color = 'Red'
- MinPrice = 100
- MaxPrice = 1000

---

18. Manager → Direct Reports Hierarchy

Create mTVF:

```
dbo.GetReportingHierarchy(@ManagerID)
```

Use `HumanResources.Employee` (self-join logic). Return Level, EmployeeID, FullName.

---

19. Rewrite Scalar UDF as Inline TVF (Performance Task)

Rewrite slow UDF:

```
dbo.GetProductProfit(ProductID)
```

as inline TVF:

```
dbo.GetProductProfit_iTVF(ProductID)
```

Compare performance using:

```
SET STATISTICS TIME, IO ON;
```

---

20. Employee Age Function

Create scalar function:

```
dbo.CalculateAge(@BirthDate)
```

Join it with `HumanResources.Employee` and `Person.Person` to list employees with age.

End of practice set

These tasks cover every dimension of UDF mastery — logic, SQL constructs, joins, performance, and real-world use cases within the AdventureWorks2022 schema.