

From the above given tables perform the following queries:

PART-A

1. Create a cursor Product_Cursor to fetch all the rows from a products table.

```
DECLARE @Product_id INT,  
        @Product_Name VARCHAR(250),  
        @Price DECIMAL(10, 2);  
DECLARE Product_Cursor CURSOR  
FOR SELECT Product_id, Product_Name, Price FROM Products;  
  
OPEN Product_Cursor;  
  
FETCH NEXT FROM Product_Cursor INTO @Product_id, @Product_Name, @Price;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    PRINT 'Product ID: ' + CAST(@Product_id AS VARCHAR) + ', Product Name: ' +  
    @Product_Name + ', Price: ' + CAST(@Price AS VARCHAR);  
    FETCH NEXT FROM Product_Cursor  
    INTO @Product_id, @Product_Name, @Price;  
END;  
  
CLOSE Product_Cursor;  
  
DEALLOCATE Product_Cursor;
```

2. Create a cursor Product_Cursor_Fetch to fetch the records in form of ProductID_ProductName (Example: 1_Smartphone).

```
DECLARE @Product_id INT,  
        @Product_Name VARCHAR(250);  
DECLARE Product_Cursor_Fetch CURSOR  
FOR SELECT Product_id, Product_Name FROM Products;  
  
OPEN Product_Cursor_Fetch;  
  
FETCH NEXT FROM Product_Cursor_Fetch INTO @Product_id, @Product_Name;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    PRINT CAST(@Product_id AS VARCHAR) + '_' + @Product_Name;  
    FETCH NEXT FROM Product_Cursor_Fetch INTO @Product_id, @Product_Name;  
END;  
  
CLOSE Product_Cursor_Fetch;  
  
DEALLOCATE Product_Cursor_Fetch;
```

3. Create a cursor that displays the products price above 50000.

```
DECLARE @Product_id INT,  
        @Product_Name VARCHAR(250),  
        @Price DECIMAL(10, 2);  
  
DECLARE Product_Cursor_Above50K CURSOR  
FOR SELECT Product_id, Product_Name, Price  
FROM Products  
WHERE Price > 50000;  
  
OPEN Product_Cursor_Above50K;  
  
FETCH NEXT FROM Product_Cursor_Above50K INTO @Product_id, @Product_Name,  
@Price;  
  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    PRINT 'Product ID: ' + CAST(@Product_id AS VARCHAR) + ', Product Name: ' +  
    @Product_Name + ', Price: ' + CAST(@Price AS VARCHAR);  
  
    FETCH NEXT FROM Product_Cursor_Above50K  
    INTO @Product_id, @Product_Name, @Price;  
  
END;  
  
CLOSE Product_Cursor_Above50K;  
  
DEALLOCATE Product_Cursor_Above50K;
```

4. Create a cursor Product_CursorUpdate that retrieves all the data from the products table and increases the price by 10%.

```
DECLARE @product_id INT,  
        @price DECIMAL(10, 2);  
  
DECLARE product_cursor CURSOR  
FOR SELECT product_id, price FROM Products;  
  
OPEN product_cursor;  
  
FETCH NEXT FROM product_cursor INTO @product_id, @price;  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    SET @price = @price + (@price * 0.10); -- Increase price by 10%  
  
    UPDATE Products  
    SET price = @price  
    WHERE product_id=@product_id  
  
    FETCH NEXT FROM product_cursor INTO @product_id, @price;  
  
END;  
  
CLOSE product_cursor;
```

```
DEALLOCATE product_cursor;
```

PART-B

5. Create a cursor that finds product with maximum price.

```
DECLARE @Product_id INT, @Product_Name VARCHAR(250), @Price DECIMAL(10, 2);  
  
DECLARE Product_Cursor_MaxPrice CURSOR  
FOR  
    SELECT TOP 1 Product_id, Product_Name, Price  
    FROM Products  
    ORDER BY Price DESC;  
  
OPEN Product_Cursor_MaxPrice;  
  
FETCH NEXT FROM Product_Cursor_MaxPrice INTO @Product_id, @Product_Name, @Price;  
  
PRINT 'Product with Maximum Price: ' + CAST(@Product_id AS VARCHAR) + '-' +  
    @Product_Name + ', Price: ' + CAST(@Price AS VARCHAR);  
  
CLOSE Product_Cursor_MaxPrice;  
  
DEALLOCATE Product_Cursor_MaxPrice;
```

6. Create a cursor to insert details of Products table into the NewProducts table if the product is "Laptop".

```
DECLARE @product_id int, @product_name nvarchar(100), @Price decimal(10,2) ;  
DECLARE product_cursor_insert CURSOR  
FOR SELECT product_id, product_name, price FROM products;  
  
OPEN product_cursor_insert;  
  
FETCH NEXT FROM product_cursor_insert  
INTO @product_id, @product_name, @price;  
  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    IF @product_name = 'Laptop'  
        INSERT INTO new_products  
        VALUES(@product_id, @product_name, @price)  
  
    FETCH NEXT FROM product_cursor_insert  
    INTO @product_id, @product_name, @price;  
  
END;  
  
CLOSE product_cursor_insert;  
  
DEALLOCATE product_cursor_insert;
```

PART-C

7. Create a cursor that increase price of products by 5000 if price is below 40000.

```
DECLARE @Product_id INT, @Price DECIMAL(10, 2);

DECLARE Product_Cursor_IncreasePrice CURSOR
FOR SELECT Product_id, Price
FROM Products
WHERE Price < 40000;

OPEN Product_Cursor_IncreasePrice;

FETCH NEXT FROM Product_Cursor_IncreasePrice INTO @Product_id, @Price;

WHILE @@FETCH_STATUS = 0
BEGIN
    UPDATE Products
    SET Price = @Price + 5000
    WHERE Product_id = @Product_id;

    FETCH NEXT FROM Product_Cursor_IncreasePrice INTO @Product_id, @Price;
END;

CLOSE Product_Cursor_IncreasePrice;
DEALLOCATE Product_Cursor_IncreasePrice;
```

8. Create a cursor that displays products with prices below the average price.

```
DECLARE @Product_id INT, @Product_Name VARCHAR(250), @Price DECIMAL(10, 2);
DECLARE @AvgPrice DECIMAL(10, 2);

SELECT @AvgPrice = AVG(Price) FROM Products;

DECLARE Product_Cursor_BelowAvgPrice CURSOR
FOR SELECT Product_id, Product_Name, Price
FROM Products
WHERE Price < @AvgPrice;

OPEN Product_Cursor_BelowAvgPrice;

FETCH NEXT FROM Product_Cursor_BelowAvgPrice
INTO @Product_id, @Product_Name, @Price;

WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT 'Product ID: ' + CAST(@Product_id AS VARCHAR) + ', Product Name: ' +
    @Product_Name + ', Price: ' + CAST(@Price AS VARCHAR);

    FETCH NEXT FROM Product_Cursor_BelowAvgPrice
    INTO @Product_id, @Product_Name, @Price;
END;
```

```
CLOSE Product_Cursor_BelowAvgPrice;
```

```
DEALLOCATE Product_Cursor_BelowAvgPrice;
```

9. Create a cursor Product_CursorDelete that deletes all the data from the Products table.

```
DECLARE @Product_id INT;
```

```
DECLARE Product_CursorDelete CURSOR
```

```
FOR SELECT Product_id
```

```
FROM Products;
```

```
OPEN Product_CursorDelete;
```

```
FETCH NEXT FROM Product_CursorDelete INTO @Product_id;
```

```
WHILE @@FETCH_STATUS = 0
```

```
BEGIN
```

```
DELETE FROM Products WHERE Product_id = @Product_id;
```

```
FETCH NEXT FROM Product_CursorDelete INTO @Product_id;
```

```
END;
```

```
CLOSE Product_CursorDelete;
```

```
DEALLOCATE Product_CursorDelete;
```

Darshan
UNIVERSITY
योग: कर्मसु कौशलम्