1 فهرس غير مجمع على email في

sales.customers

```
CREATE NONCLUSTERED INDEX idx_customers_email
ON sales.customers (email);
GO
```

2 فهرس مركب على category_id و brand_id

```
CREATE NONCLUSTERED INDEX idx_products_category_brand
ON production.products (category_id, brand_id);
GO
```

3 فهرس على order_date مع أعمدة مضمنة

```
CREATE NONCLUSTERED INDEX idx_orders_order_date
ON sales.orders (order_date)
INCLUDE (customer_id, store_id, order_status);
GO
```

الجداول المطلوبة للمشغلات

سجل نشاط العملاء

```
CREATE TABLE sales.customer_log (
    log_id INT IDENTITY(1,1) PRIMARY KEY,
    customer_id INT,
    action VARCHAR(50),
    log_date DATETIME DEFAULT GETDATE()
);
GO
```

```
-- سجل تغييرات الأسعار
CREATE TABLE production.price_history (
```

```
    history_id INT IDENTITY(1,1) PRIMARY KEY,
    product_id INT,
    old_price DECIMAL(10,2),
    new_price DECIMAL(10,2),
    change_date DATETIME DEFAULT GETDATE(),
    changed_by VARCHAR(100)
);
GO


-- سجل تدقيق الطلبات
CREATE TABLE sales.order_audit (
    audit_id INT IDENTITY(1,1) PRIMARY KEY,
    order_id INT,
    customer_id INT,
    store_id INT,
    staff_id INT,
    order_date DATE,
    audit_timestamp DATETIME DEFAULT GETDATE()
);
GO
```

4 مشغل إدراج عميل جديد (Welcome Log)

```
CREATE TRIGGER trg_InsertCustomerLog
ON sales.customers
AFTER INSERT
AS
BEGIN
    INSERT INTO sales.customer_log (customer_id, action)
    SELECT customer_id, 'New Customer Added'
```

```
    FROM inserted;
END;
GO
```

5 مشغل تسجيل تغيير السعر

```
CREATE TRIGGER trg_ProductPriceChange
ON production.products
AFTER UPDATE
AS
BEGIN
    INSERT INTO production.price_history
    (product_id, old_price, new_price, changed_by)
    SELECT
        d.product_id,
        d.list_price,
        i.list_price,
        SYSTEM_USER
    FROM deleted d
    JOIN inserted i
        ON d.product_id = i.product_id
    WHERE d.list_price <> i.list_price;
END;
GO
```

6 مشغل بدلاً من الحذف لمنع حذف فئة بها منتجات

```
CREATE TRIGGER trg_PreventCategoryDelete
ON production.categories
INSTEAD OF DELETE
AS
BEGIN
    IF EXISTS (
```

```
        SELECT 1
        FROM production.products p
        JOIN deleted d
            ON p.category_id = d.category_id
    )
    BEGIN
        RAISERROR ('لا يمكن حذف فئة تحتوي على منتجات', ١٦, ١);
        RETURN;
    END

    DELETE FROM production.categories
    WHERE category_id IN (SELECT category_id FROM deleted);
END;
GO
```

7 مشغل تقليل المخزون عند إضافة عنصر طلب

```
CREATE TRIGGER trg_UpdateStockAfterOrder
ON sales.order_items
AFTER INSERT
AS
BEGIN
    UPDATE s
    SET s.quantity = s.quantity - i.quantity
    FROM production.stocks s
    JOIN inserted i
        ON s.product_id = i.product_id
       AND s.store_id = i.store_id;
END;
GO
```

8 مشغل تسجيل الطلبات الجديدة

```
CREATE TRIGGER trg_OrderAudit
ON sales.orders
AFTER INSERT
AS
BEGIN
   INSERT INTO sales.order_audit
   (order_id, customer_id, store_id, staff_id, order_date)
   SELECT
      order_id,
      customer_id,
      store_id,
      staff_id,
      order_date
   FROM inserted;
END;
GO
```