TEXT

```
FUNCTION get_order_ids
RETURN SYS_REFCURSOR
IS
   order_cursor SYS_REFCURSOR;
BEGIN
    OPEN order_cursor FOR
        SELECT ORDER_ID FROM ORDERS;
    RETURN order_cursor;
END get_order_ids;
FUNCTION get_payments
RETURN SYS_REFCURSOR
IS
   payments_cursor SYS_REFCURSOR;
BEGIN
    OPEN payments_cursor FOR
        SELECT PAYMENT_ID, AMOUNT, ORDER_ID, PAYMENT_DATE
         FROM PAYMENTS;
    RETURN payments_cursor;
END;
FUNCTION get_suppliers RETURN SYS_REFCURSOR AS
   supplier_cursor SYS_REFCURSOR;
BEGIN
    OPEN supplier_cursor FOR
    SELECT id, supplier_name, contact_number, email, address FROM suppliers;

    -- Print output for debugging in SQL Developer
    FOR rec IN (SELECT id, supplier_name, contact_number, email, address FROM suppliers) LOOP
        DBMS_OUTPUT.PUT_LINE('ID: ' || rec.id ||
                             ', Name: ' || rec.supplier_name ||
                             ', Contact: ' || rec.contact_number ||
                             ', Email: ' || rec.email ||
                             ', Address: ' || rec.address);
    END LOOP;

    RETURN supplier_cursor;
END get_suppliers;
FUNCTION is_contact_unique(p_contact VARCHAR2, p_customer_id NUMBER)
RETURN BOOLEAN IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
```

TEXT

```
    INTO v_count
    FROM customers
    WHERE contact = p_contact AND id != p_customer_id;

    IF v_count > 0 THEN
        RETURN FALSE;  -- Contact is not unique
    ELSE
        RETURN TRUE;  -- Contact is unique
    END IF;
END is_contact_unique;
FUNCTION is_email_unique(p_email VARCHAR2, p_customer_id NUMBER)
RETURN BOOLEAN IS
    v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM customers
    WHERE email = p_email AND id != p_customer_id;

    IF v_count > 0 THEN
        RETURN FALSE;  -- Email is not unique
    ELSE
        RETURN TRUE;  -- Email is unique
    END IF;
END is_email_unique;
FUNCTIONupdate_order_status(
    p_order_id IN NUMBER,
    p_action IN VARCHAR2
) RETURN VARCHAR2 IS
    v_status VARCHAR2(100);
BEGIN
    IF LOWER(p_action) = 'accept' THEN
        v_status := 'Your order has been accepted';
        UPDATE orders
        SET status = v_status
        WHERE order_id = p_order_id;

        RETURN 'Order accepted successfully';

    ELSIF LOWER(p_action) = 'cancel' THEN
        v_status := 'Your order has been cancelled';
```

TEXT

```
        -- Update order status
        UPDATE orders
        SET status = v_status
        WHERE order_id = p_order_id;

        -- Increase stock quantity for each item
        FOR item IN (
            SELECT product_id, quantity
            FROM order_items
            WHERE order_id = p_order_id
        ) LOOP
            UPDATE products
            SET stock_quantity = stock_quantity + item.quantity
            WHERE id = item.product_id;
        END LOOP;

        RETURN 'Order cancelled and stock updated';

    ELSE
        RETURN 'Invalid action';
    END IF;
END;
```