PARTE 1 - (c) QUERY COMPLEJO

```
SELECT c1.lname+','+c1.fname AS CLIENTE,
       COALESCE(SUM(i1.total_price*i1.quantity),0) AS totalCompra,
       ref.lname + ', ' + ref.fname AS Referido,
       ref.totalCompraReferido * 0.05 AS totalComision
FROM customer c1 LEFT JOIN orders o1 ON (c1.customer_num=o1.customer_num)
                 LEFT JOIN items i1 ON (o1.order_num=i1.order_num)
                 LEFT JOIN (SELECT c2.customer_num,
                                   c2.lname,
                                   c2.fname,
                                   SUM(i2.total_price * i2.quantity) AS
totalCompraReferido,
                                   c2.customer_num_referedBy
                            FROM customer c2 JOIN orders o2 ON (c2.customer_num =
o2.customer_num)
                                             JOIN items i2 ON
(i2.order_num=o2.order_num)
                            WHERE c2.customer_num_referedBy IS NOT NULL
                              AND i2.stock_num IN (1,4,5,6,9)
                            GROUP BY c2.customer_num,
                                     c2.lname,
                                     c2.fname,
                                     c2.customer_num_referedBy
                       ) AS Ref ON (Ref.customer_num_referedBy=c1.customer_num)
GROUP BY c1.customer_num, c1.lname, c1.fname,
         Ref.customer_num, Ref.lname, Ref.fname,
         Ref.totalCompraReferido
ORDER BY 1
```

PARTE 2 - (e) TRIGGER

```
CREATE TRIGGER dbo.customerTr

ON dbo.v_Productos

INSTEAD OF INSERT AS

BEGIN

DECLARE @codCliente       SMALLINT,
        @nombre           VARCHAR(15),
        @apellido         VARCHAR(15),
        @codProvincia     CHAR(2),
        @fechaLlamado     DATETIME,
        @usuarioId        CHAR(32),
        @codTipoLlamada   CHAR(1),
        @descrLlamada     VARCHAR(40),
        @descrTipoLlamada VARCHAR(30)

DECLARE contactos_cur CURSOR FOR
        SELECT * FROM inserted

OPEN contactos_cur

FETCH NEXT FROM contactos_cur
    INTO @codCliente, @nombre, @apellido, @codProvincia, @fechaLlamado, @usuarioId,

@codTipoLlamada, @descrLlamada, @descrTipoLlamada;

WHILE @@FETCH_STATUS = 0 BEGIN

    BEGIN TRY

        BEGIN TRANSACTION

        IF @codCliente IS NULL OR @codProvincia IS NULL OR @codTipoLlamada IS NULL OR

@fechaLlamado IS NULL BEGIN

            RAISERROR ('El código de cliente, la provincia, el tipo de llamada y la

fecha de llamado son OBLIGATORIAS.', -- Message text.  6, -- Severity.  1 -- State.);

      END

        IF @codTipollamada NOT IN ('B','D','I','L','O') BEGIN

            RAISERROR ('Codigo de Llamada inválido.', 16, 1)

        END

        IF NOT EXISTS (SELECT 1 FROM dbo.state where code = @codProvincia) BEGIN

            RAISERROR ('Codigo de Provincia inexistente.', 16, 1);

        END
        IF NOT EXISTS (SELECT 1 FROM dbo.customer c WHERE c.customer_num =
@codCliente) BEGIN
            INSERT INTO CUSTOMER (customer_num, fname, lname, state)
                VALUES (@codCliente, @nombre, @apellido, @codProvincia)
        END
        IF NOT EXISTS (SELECT 1 FROM call_type ct WHERE ct.call_code =
@codTipoLlamada) BEGIN
            INSERT INTO call_type (call_code, code_descr)
                VALUES (@codTipoLlamada, @descrTipoLlamada);
        END
        INSERT INTO cust_calls (customer_num, call_dtime, user_id, call_code,
call_descr)
            VALUES (@codCliente, @fechaLlamado, @usuarioId,  @codTipoLlamada,
@descrLlamada)
          COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
          ROLLBACK TRANSACTION
          PRINT 'hubo un error' -- esto es para debug
    END CATCH
    FETCH NEXT FROM contactos_cur
        INTO @codCliente, @nombre, @apellido, @codProvincia, @fechaLlamado,
@usuarioId, @codTipoLlamada, @descrLlamada, @descrTipoLlamada;
END
CLOSE contactos_cur
DEALLOCATE contactos_cur
END
```