**a. En no más de 15 renglones explique todo lo relacionado con el objeto VISTA.**

**b. En una carilla explique la funcionalidad de Seguridad de un RDBMS detallando someramente los objetos relacionados con la misma.**

**c. Crear una consulta que devuelva los siguientes cuatro atributos:**

* **Apellido, Nombre AS Cliente,**
* **Suma de todo lo comprado por el cliente AS totalCompra,**
* **Apellido, Nombre AS ClienteReferido,**
* **Suma de lo comprado por el referido\*0.05 AS totalComision**

**Consideraciones:**

* **En el caso que un no tenga OCs deberá mostrar 0 en el campo totalCompra**
* **En el caso que un Cliente no tenga Referidos deberá mostrar al mismo con NULL en las columnas ClienteReferido y totalComision.**
* **Para calcular la comisión del cliente se deberán sumar (cant\*precio) de todos los productos comprados por el ClienteReferido cuyo stock\_num sea 1,4,5,6,9. La comisión es del 5%.**
* **Se deberá ordenar la salida por el Apellido y Nombre del Cliente.**

**No se pueden utilizar tablas temporales, ni funciones de usuario.**

**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

**STORE PROCEDURE**

**d. Desarrollar un SP que maneje la inserción o modificación de un producto determinado.**

**Parámetros de entrada: stock\_num, manu\_code, unit\_price, unit\_code, description**

**Si existe el producto en la tabla products, actualizar los atributos que no pertenecen a la PK.**

**Si no existe el producto en la tabla products, insertar fila en la tabla, previamente validar lo siguiente:**

* **Existencia de manu\_code en la tabla manufact**
  + **Informando “Error por fabricante inexistente”.**
* **Existencia de stock\_num en la tabla product\_types**
  + **Si no existe, insertar registro en la tabla stock\_num.**
  + **Si existe, realizar un update del atributo description.**
* **Existencia del atributo unit\_code en la tabla units**
  + **Informando “Error por código de unidad inexistente”.**

**TRIGGER**

**e. Realizar un trigger de Insert sobre la siguiente Vista, insertando datos en las tablas correspondientes validando la existencia de sus respectivas PKs y que las mismas no sean nulas.**

*CREATE VIEW v\_Productos*

*(codCliente, nombre, apellido, codProvincia, fechaLlamado, usuarioId, codTipoLlamada, descrLlamada, descrTipoLlamada)*

*SELECT c.customer\_num, fname, lname, state, call\_dtime, user\_id, cc.call\_code, call\_descr, code\_descr*

*FROM customer c JOIN cust\_calls cc ON (c.customer\_num = cc.customer\_num)*

*JOIN call\_type ct ON (cc.call\_code = ct.call\_code)*

*WHERE ct.call\_code IN (‘B’, ‘D’, ‘I’, ‘L’, ‘O’)*

*AND state IN (SELECT code FROM state)*

*WITH CHECK OPTION*

**Tener en cuenta en el trigger la inserción de múltiples filas.**

**e.** CREATE or ALTER TRIGGER dbo.customerTr

   ON  dbo.v\_Productos

   INSTEAD OF INSERT AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

declare @codCliente       smallint

declare @nombre           varchar(15)

declare @apellido         varchar(15)

declare @codProvincia     char(2)

declare @fechaLlamado     datetime

declare @usuarioId        char(32)

declare @codTipoLlamada   char(1)

declare @descrLlamada     varchar(40)

declare @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

        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.

                   16, -- 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)

        end try

    begin catch

     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

GO