

<b>EVALUACION</b>	Solución al examen	<b>GRUPO</b>	Todos	<b>FECHA</b>	22/Jul/2014
<b>MATERIA</b>	Bases de datos y Bases de datos 1				
<b>CARRERA</b>	AP – ATI - APW				
<b>CONDICIONES</b>	<b>- Puntos: 100</b> <b>- Duración: 3 horas</b> <b>- SIN material</b>				

### Ejercicio 1

Una empresa tiene un sitio en internet que se dedica a promover cupones de descuento a ser utilizados en productos que se venden en comercios en Uruguay. Cuando un "internauta" imprime un cupón, en éste se registra la fecha y la hora (timestamp). Además se imprime la información correspondiente al porcentaje de descuento o dinero que se ahorra al comprar un producto específico en un comercio determinado utilizando el cupón. Dicho descuento estará vigente hasta una determinada fecha que también debe constar en el mismo.

Un cupón puede habilitar el descuento al comprar una cantidad específica de unidades de un producto. (Por ejemplo 1 unidad de detergente Hurra de ½ lt tiene un 10% de descuento en la cadena de supermercados Tienda Inglesa, se obtiene en la misma cadena \$30 al comprar con cupón dos cajas de Cereales Nesquik). Eventualmente, si el comercio llega a un acuerdo con una determinada tarjeta de crédito, puede llegar a ofrecer un descuento extra si el producto se abona con la misma. (Por ejemplo, una empresa telefónica ofrece \$1.000 de descuento el celular Samsung Galaxy 7 hasta el 31 de Julio del 2014 y si se paga con Master Card se obtiene un 10% extra). Esto último también debe figurar en el cupón aunque el cliente no realice la compra con esa tarjeta ya que es parte de la promoción. Todos los descuentos por productos son en porcentaje o una cantidad determinada de pesos uruguayos mientras que los de tarjeta siempre son un porcentaje.

Una vez finalizada la promoción, se envía por correo electrónico a un contacto determinado de cada comercio la información de la cantidad de cupones impresos para cada producto ofrecido por el mismo. Cada cupón lleva impreso un código de barras que corresponde a una identificación numérica única, los 12 primeros dígitos identifican la promoción, los 4 siguientes un correlativo (de 0000 a 9999, hasta 10000 cupones impresos de una promoción)

A los efectos de facilitar las búsquedas, los productos se agrupan por categorías. Cada producto puede pertenecer a una o más categorías. Cada producto promocionado tiene un código único asignado por el sistema al darlo de alta. (Por ejemplo el código 1234567 corresponde a la Coca Cola 600cc en cualquier comercio que quiera promocionarla a través de cupones)

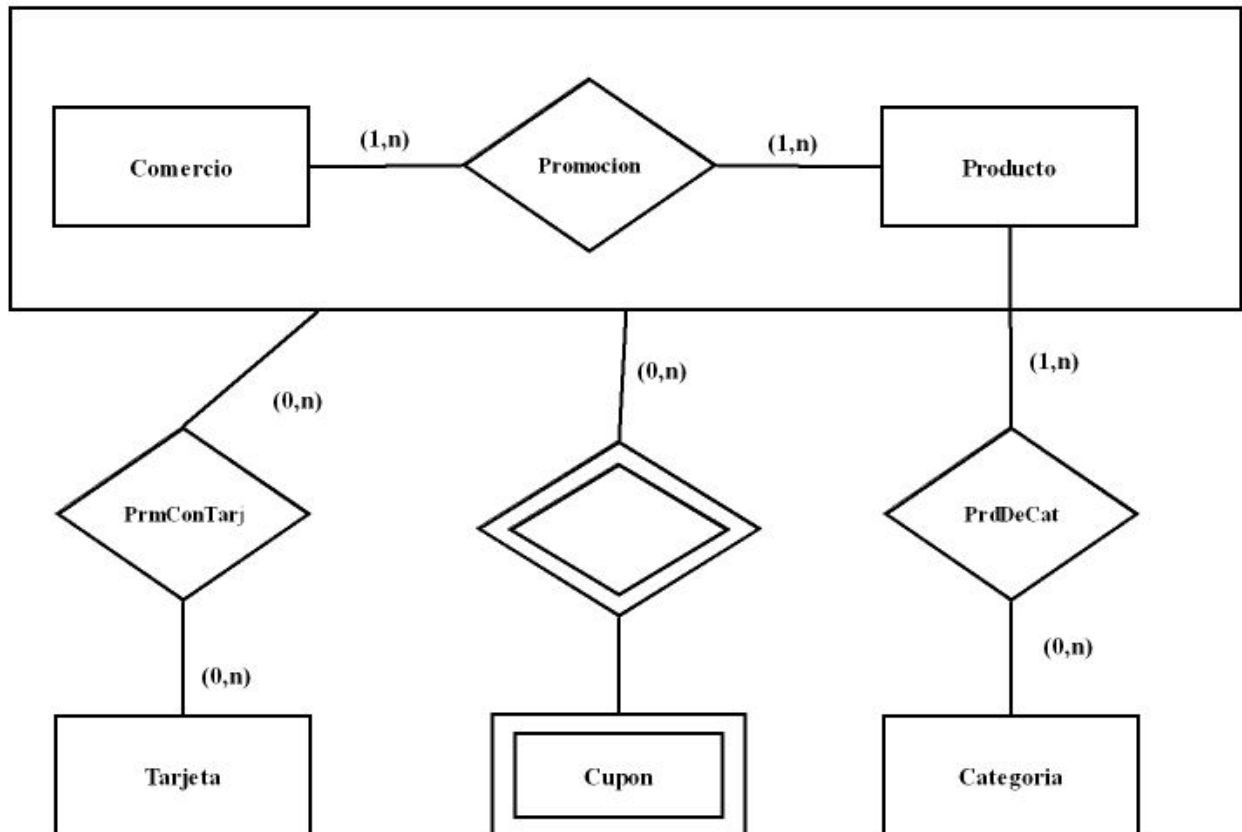
Las promociones que se registran son las vigentes. Una vez finalizada, se envían los datos al comercio y a la tarjeta, si corresponde y se borran. Tampoco puede haber dos promociones simultáneas para un mismo producto en el mismo comercio. De cada comercio que promueve sus productos se conoce además su RUT que es único, su razón social, su dirección fiscal, su nombre fantasía y el nombre, teléfono y correo de la persona que debemos contactar en la misma.

- Se pide modelo entidad-relación correspondiente donde se debe especificar claramente los atributos de cada entidad y relación y en caso de que no sean obvios, que representan. También se debe especificar la cardinalidad y participación de las entidades en las relaciones y en caso de utilizar este tipo de estructuras, las agregaciones y/o categorizaciones/generalizaciones, entidades débiles y evitar toda situación que pudiera permitir el almacenamiento de datos inconsistentes que no se pueda deducir del modelo utilizando restricciones de integridad no estructurales. **(Máximo: 30 puntos)**
- Convertir el MER anterior en su correspondiente esquema relacional llevándolo al menos hasta 3ª forma normal y explicitando sus correspondientes restricciones de integridad (estructurales y no estructurales). **(Máximo 20 puntos)**
- Codificar en el Lenguaje de definición de datos de SQL la sentencia necesaria para la tabla que contenga los datos identificatorios de cada cupón explicitando su clave primaria, unicidades, foráneas (aunque no se creen las tablas correspondientes) y restricciones de atributo si corresponde. **(Máximo 10 puntos)**

**(Máximo del Ejercicio 1 es 60 puntos)**

**Solución:**

**Modelo Entidad-Relación:**



**Atributos de ENTIDADES del MER:**

**Comercio:** RUT, RazonSocial, DireccionFiscal, NombreFantasia, NombreContacto, TelefonoContacto, EmailContacto

**Producto:** IdProducto, DescripcionProducto (En descripción está todo el texto que describe exactamente el producto)

**Tarjeta :** IdTarjeta, NombreTarjeta,

**Categoría:** IdCategoría, NombreCategoría,

**Cupon:** NroCorrelativoCupon, FechaHoraImpresion

**Atributos de ENTIDADES del MER:**

**Promocion:** IdPromocion, Cantidad, Descuento, PorcentajeOPesos, FechaDesde, FechaHasta

**PrmConTarj:** PorcentajeDescuento

**PrdDeCat :** sin atributos descriptivos

**Restricciones de Integridad (no estructurales):**

R1) En Promoción, el IdPromoción no se puede repetir (explícita en la letra)

---

**Modelo Relacional en 3ª forma normal:**

**Comercio** (RUT, RazonSocial, DireccionFiscal, NombreFantasia, NombreContacto, TelefonoContacto, EmailContacto)

**Producto** ( IdProducto, DescripcionProducto)

**Tarjeta** (IdTarjeta<sub>1</sub>, NombreTarjeta<sub>2</sub>, ) NombreTarjeta se declara como unicidad

**Categoría** ( IdCategoría<sub>1</sub>, NombreCategoría<sub>2</sub>, ) NombreCategoría se declara como unicidad

**Cupon** (IdPromocion, NroCorrelativoCupon, FechaHoraImpresion) *IdPromocion es FK de Promocion*

**Promocion** (IdPromocion, IdProducto, RUT, Cantidad, Descuento, PorcentajeOPesos, FechaDesde, FechaHasta )

*IdProducto es FK de Producto, RUT es FK de Comercio*

**PrmConTarj** (IdPromocion, PorcentajeDescuento)

**PrdDeCat** (IdCategoría, idProducto) *IdProducto es FK de Producto, IdCategoría es FK de Categoría*

**Creación de tabla con datos identificatorios del Cupón**

```
CREATE TABLE Cupon (  
  IdPromocion DECIMAL (12) NOT NULL,  
  NroCorrelativoCupon DECIMAL(20) NOT NULL,  
  FechaHoraImpresion DATETIME2 NOT NULL,  
  CONSTRAINT FK_Cupon_Promocion FOREIGN KEY (IdPromocion) REFERENCES Promocion (IdPromocion),  
  CONSTRAINT pk_Cupon PRIMARY KEY (IdPromocion, NroCorrelativoCupon)  
);
```

## Ejercicio 2

Dadas las siguientes relaciones, que forman parte del modelo de una empresa de televisión para abonados, escribir las consultas en SQL para responder las consultas enumeradas a continuación.

**Canal** (NroCnl, NomCnl )

**Programa** (CodProg<sub>1</sub>, NomProg<sub>2</sub>, CodTipo, Durac )

**ProgEmiEnCnl** (CodProg, NroCnl, FchEmi, HoraEmi ) Programa emitido en Canal

**TipoDeProg** (CodTipo, DescrTipo ) Tipo de Programa (serie, película, musical, evento deportivo, etc.)

Se debe tener en cuenta que:

- En todos los casos, se debe evitar datos repetidos si no aportan información útil para la consulta.
- Se puede asumir concordancia en todas las claves foráneas pero no la participación total.

1. Obtener una lista de los nombres de canales que hayan emitido algún (uno o más) programas deportivos (tipo “DEP”) y/o algún (uno o más) informativos (tipo “INF”).

```
SELECT  NomCnl
FROM    Canal
WHERE   NroCnl IN (SELECT DISTINCT(E.NroCnl)
                  FROM    Programa P, ProgEmiEnCnl E
                  WHERE   P.CodProg=E.CodProg
                  AND     (CodTipo='INF' OR CodTipo='DEP'))
```

2. Obtener una lista de códigos de programa que se hayan emitido en un único canal.

```
SELECT  DISTINCT CodProg
FROM    ProgEmiEnCnl U
WHERE   NOT EXISTS (SELECT *
                   FROM    ProgEmiEnCnl D
                   WHERE   U.CodProg=D.CodProg
                   AND     U.NroCnl<>D.NroCnl)
```

3. Obtener la lista de nombres de canal que nunca hayan emitido programas deportivos (código de tipo “DEP”).

```
(SELECT  NomCnl
FROM    Canal C)
EXCEPT
(SELECT DISTINCT(E.NroCnl)
FROM    Programa P, ProgEmiEnCnl E
WHERE   P.CodProg=E.CodProg
AND     CodTipo='DEP')
```

4. Obtener nombres de programa y cantidad de veces que se emitieron siempre que se hayan emitido en más de tres canales.

```
SELECT  NomProg, COUNT(DISTINCT(NroCnl))
FROM    Programa P, ProgEmiEnCnl E
WHERE   P.CodProg=E.CodProg
GROUP BY NomProg
HAVING  COUNT(DISTINCT(NroCnl))>3
```

Nota:(COUNT(\*) hubiera sido lo mismo por la clave primaria)

- 
5. Siendo Durac la duración (en minutos) de un programa, obtener los nombres de los programas de mínima duración emitidos por algún canal en lo que va del año.

```
SELECT NomProg
Programa P, ProgEmiEnCnl E
WHERE P.CodProg=E.CodProg
AND YEAR(FchEmi)=2014
AND Duraci = (SELECT MIN(Durac)
              FROM Programa P, ProgEmiEnCnl E
              WHERE P.CodProg=E.CodProg
              AND YEAR(FchEmi)=2014)
```

(Cada consulta vale 8 puntos. Total del ejercicio 2 = 40 puntos).