

Bases de Datos

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Trabajo Práctico I: *ECSA - Entretenimiento Completo S.A*

23/05/2018

Integrante	LU	Correo electrónico
Enzo Samuel Cioppettini	405/15	tenstrings5050@gmail.com
Tomás Ariel Pastore	266/15	pastoretomas96@gmail.com
Patricio López Valiente	457/15	patriciolopezvaliente@gmail.com
Fernando Balboa	246/15	fbalboa95@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción	3
1.1. Asunciones	3
2. D.E.R	5
3. M.R	6
3.1. Restricciones	7
4. Testing	8
5. Conclusiones	9

1. Introducción

En este trabajo diseñaremos e implementaremos una Base de Datos relacional para “Entretenimiento Completo S.A.”. Esta empresa provee tarjetas de acceso personalizadas tanto a parques de diversiones como a diversos eventos. E.C.S.A., en búsqueda de brindar el mejor servicio a sus clientes, ofrece beneficios mediante descuentos a los titulares de las tarjetas, los cuales son utilizados para acceder a los entretenimientos antes mencionados a un costo menor que el usual. De esta forma, los clientes consumen en los diversos medios de entretenimiento y luego reciben mensualmente una factura correspondiente a dichos consumos, con los descuentos disponibles ya aplicados. Los descuentos dependen de la categoría asociada a la tarjeta del cliente, mientras que la categoría es determinada periódicamente en relación al consumo del mismo.

Se busca desarrollar un sistema que permita fácilmente obtener qué medios de entretenimiento están en su programa de beneficios y los datos asociados a estos, como por ejemplo el descuento ligado a un medio de entretenimiento según cada categoría. A su vez, es relevante archivar la información de sus clientes de manera efectiva, registrar sus pagos, consumos históricos y trayectoria en la vida del programa.

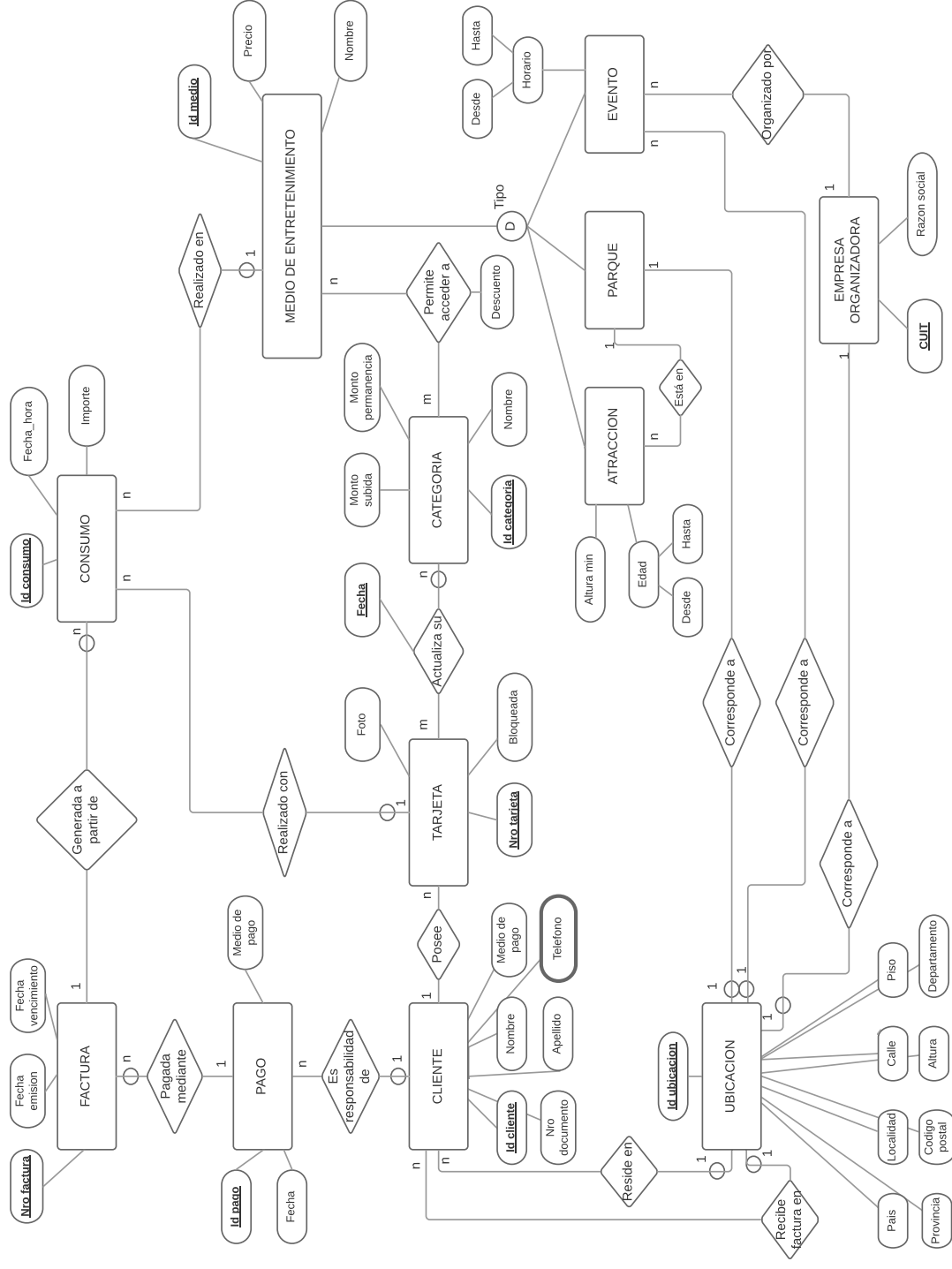
La base de datos desarrollada tiene por objetivo satisfacer de manera práctica las funcionalidades previamente dichas, y algunas más, como por ejemplo el cambio automático de categoría de los clientes evaluando sus consumos en un periodo. Para el desarrollo de la misma emplearemos un modelo de entidad relación, el cual se verá reflejado mediante el diagrama (D.E.R) presentado en la sección 2. El pasaje correspondiente de este al Modelo Relacional será detallado en la sección 3, acompañado por las restricciones de dominio que creemos pertinentes. Por último realizaremos la implementación del modelo en un motor de bases de datos relacional (MySQL), buscando elaborar conclusiones acerca de este tipo de soluciones y compartiendo algunos problemas que se pueden presentar en el trayecto.

1.1. Asunciones

1. Los clientes siempre son creados con una tarjeta ya asignada porque siempre deben tener una tarjeta activa.
2. Por simplicidad nuestras categorías serán "BRONCE", "PLATA" y "ORO".
3. Si un cliente no consume en todo un mes no se emitirá factura para dicho mes.
4. Un consumo no tiene factura si y sólo si es del mes actual. A fin de mes se crea la factura con todos los respectivos consumos del cliente.
5. Los consumos que consideramos para los efectos en movimientos de categoría son los que ya se encuentran pagos.
6. Cada pago puede pagar más de una factura, pero siempre en su totalidad. En otras palabras, un pago no puede pagar parcialmente una factura.
7. El cliente puede cambiar su medio de pago, pero para cada pago se guarda el medio de pago que tenía en la fecha de pago.

8. Cada cliente sólo puede realizar un pago por día. Esto es una consecuencia de no guardar la hora de los pagos y sólo guardar la fecha. De todas maneras nos parece razonable teniendo en cuenta que un pago puede anular muchas facturas a la vez.
9. Si un cliente consume pero con descuentos de %100, debemos registrar los consumos para contabilizar dichas visitas. La factura asociada tendrá eventualmente 0 pesos de importe, sin embargo en este caso sí emitiremos una factura de costo 0 para mantener el invariante de representación entre consumos y facturas (item 4). La factura automáticamente figurará paga ya que no hay nada que pagar. Notar que es un caso diferente al del item 3 ya que hay que registrar visitas.
10. El procedimiento almacenado que evalúa las categorías fue pensado para evaluar todos los clientes a la vez y asumimos que se corre de manera automática 1 vez al día para mantener la base actualizada. Si un día se olvida de correrlo se debería correr llamándolo con fecha del día olvidado y luego con la actual. Se podría pensar en un procedimiento que corriera por pago para evaluar sólo a un cliente, pero no es requerido y además presentaría algunos problemas por el hecho de no guardar la hora de los pagos y cambios de categoría.
11. Asumimos que hay una categoría con montos de subida y permanencia 0 0 que es la inicial. En nuestro caso BRONCE.
12. Si el monto de subida entre dos categorías es mayor entonces el monto de permanencia también lo es.
13. Los montos de subida de categorías son todos distintos entre sí y los de permanencia también.

2. D.E.R



3. M.R

Realizando el pasaje a Modelo Relacional a partir del diagrama de entidad-relación presentado en la sección anterior, se obtiene:

- **FACTURA** (Nro factura, Fecha emision, Fecha vencimiento, Id_pago)
PK = {Nro factura}
FK = {Id pago}
- **PAGO** (Id pago, Medio de pago, Fecha, Id_cliente)
PK = {Id pago}
FK = {Id cliente}
- **CONSUMO** (Id consumo, Fecha_hora, Importe, Nro_factura, Nro_tarjeta, Id_medio)
PK = {Id consumo}
FK = {Nro factura, Nro tarjeta, Id medio}
- **CLIENTE** (Id cliente, Nro documento, Nombre, Apellido, Medio de pago, Id_ubicacion_residencia, Id_ubicacion_facturacion)
PK = {Id cliente}
FK = {Id ubicacion residencia, Id ubicacion facturacion}
- **TELEFONO** (Id cliente, Telefono)
PK = {(Id cliente, Telefono)}
FK = {Id cliente}
- **TARJETA** (Nro tarjeta, Bloqueada, Foto, Id_cliente)
PK = {Nro tarjeta}
FK = {Id cliente}
- **ACTUALIZA SU** (Fecha, Nro tarjeta, Id categoria)
PK = {(Fecha, Nro tarjeta, Id categoria)}
FK = {Nro tarjeta, Id categoria}
- **CATEGORIA** (Id categoria, Nombre, Monto subida, Monto permanencia)
PK = {Id categoria}
FK = {}
- **PERMITE ACCEDER A** (Id categoria, Id medio, Descuento)
PK = {(Id categoria, Id medio)}
FK = {Id categoria, Id medio}

- **MEDIO DE ENTRETENIMIENTO** (Id medio, Nombre, Precio, Tipo)
 PK = {Id medio}
 FK = {}

- **ATRACCION** (Id medio, Altura min, Edad Desde, Edad Hasta, Id medio parque)
 PK = {Id medio}
 FK = {Id medio, Id medio parque}

- **PARQUE** (Id medio, Id ubicacion)
 PK = {Id medio}
 FK = {Id medio, Id ubicacion}

- **EVENTO** (Id medio, Horario Desde, Horario Hasta, Id ubicacion, CUIT Organizadora)
 PK = {Id medio}
 FK = {Id medio, Id ubicacion, CUIT Organizadora}

- **EMPRESA ORGANIZADORA** (CUIT, Razon social, Id ubicacion)
 PK = {CUIT}
 FK = {Id ubicacion}

- **UBICACION** (Id ubicacion, Pais, Provincia, Localidad, Codigo postal, Calle, Altura, Piso, Departamento)
 PK = {Id ubicacion}
 FK = {}

3.1. Restricciones

1. La fecha de emisión de una factura es siempre menor o igual a su fecha de vencimiento.
2. Existe a lo sumo una factura por mes para cada cliente.
3. La fecha de un pago es siempre mayor o igual a todas las fechas de emisión de las facturas asociadas a él.
4. El medio de pago asociado a un pago es el medio de pago asociado con el cliente en el momento en el cual se realizó el pago.
5. Un pago sólo puede pagar las facturas más antiguas impagas. Por ejemplo, no es posible pagar una factura de marzo sin haber pagado las de enero y febrero (en caso de existir).
6. Existe exactamente una tarjeta activa (no bloqueada) por cliente en todo momento.
7. Los montos de subida y permanencia de las categorías son enteros no negativos.
8. El descuento provisto por una categoría debe ser un entero entre 0 y 100 (representa porcentaje).

9. Todos los consumos de un cliente realizados en un mismo mes, con cualquiera de sus tarjetas, se encuentran en una única factura correspondiente a dicho mes.
10. La categoría de la tarjeta asociada a cada consumo debe o debía poder permitir el acceso al medio de entretenimiento en la fecha de dicho consumo.
11. El importe que figura un consumo equivale al precio del medio de entretenimiento con el descuento correspondiente ya aplicado.
12. La tarjeta que produce el consumo debía estar activa en la fecha y hora del mismo.
13. El horario *desde* de los eventos es menor al horario *hasta*.
14. La altura mínima esta comprendida entre 0 y 300 (representan centímetros).
15. Las edades de las atracciones son enteros no negativos, y *desde* es menor o igual a *hasta*.

4. Testing

Los datos utilizados para probar las queries y el procedimiento almacenado, fueron generados manualmente. Para esto, se tuvieron en cuenta todas las restricciones enumeradas en la sección anterior para que los datos sean consistentes. A su vez, para facilitar la implementación y lectura de los datos, se crearon distintos archivos que se podrán encontrar en `scripts/popdbFiles`, los cuales separan los datos según su semántica. Por ejemplo: ubicaciones, clientes y tarjetas, categorías y cambios de categoría, descuentos por categoría y otro archivo para consumos, facturas y pagos; en la carpeta `scripts` hay un archivo que se encarga de juntar de manera ordenada todas estas partes para popular la base de datos, con nombre `gatherPopulation`. Lo pueden correr en bash con `./gatherPopulation.sh` dando permisos de ejecución o bien pueden correr `python gatherPopulation.py` que hace lo mismo, el output es un script con el nombre **`populatedb.sql`**. De esta manera, una vez creado el archivo **`populatedb.sql`**, podemos crear el esquema de la base de datos (tablas vacías) corriendo el script **`createdb.sql`** y luego popular la base con `populatedb`. Las queries se encuentran en `scripts/queries.sql` y el procedimiento almacenado en `scripts/update_categories.sql`. Las asunciones para solventar las ambigüedades que encontramos en las consignas para las queries, son detalladas bajo un comentario junto a la solución de las mismas en `queries.sql`.

5. Conclusiones

- El M.E.R es de gran utilidad para dar un rápido panorama del problema que se está tratando modelar. Expresando la información de manera resumida pero a la vez concisa, permite entender la semántica y las relaciones entre las diversas partes de manera fluida, quitando en su mayoría las ambigüedades presentadas por el lenguaje natural.
- La regla para realizar el pasaje al modelo relacional a partir de un D.E.R, es también una herramienta eficaz para la resolución de este tipo de problemas, puesto a que se realiza de manera simple y ofrece un esquema normalizado como resultado del procedimiento.
- La elaboración previa de un diseño eficiente del modelo, es un punto clave para proveer oportunidades a futuras mejoras del sistema a desarrollar. Sin sobrespecificar para evitar agregar complejidad no deseada, recomendamos resolver el problema de la forma más general posible, para permitir más libertades en el futuro a la hora de realizar cambios.
- La descripción del problema por el interesado en el desarrollo debe ser lo más específica posible. A su vez, debemos considerar qué datos espera poder obtener del sistema desarrollado (qué consultas va a realizar), y con qué frecuencia estimada serán consultados. Dichos puntos pueden resultar decisivos y fundamentales en el diseño del modelo.
- Si bien el hecho de no tener redundancia en el esquema de la base de datos es una práctica recomendable para simplificar la implementación y la consistencia de la información, es posible que se considere sacrificar en parte esta consigna en busca de ganar performance. Nuevamente las consultas a realizar, jugaran un rol importante en este tipo de decisiones.
- En un sistema un poco más complejo, creemos recomendable elaborar herramientas para construir datos de prueba, ya sea scripts, funciones u otras soluciones que permitan al programador abstraerse de las restricciones necesarias para la base de datos y ahorrar un tiempo considerable en la fase de test. La generación a mano de los datos de prueba fue tediosa y requirió más tiempo del que esperábamos.
- En un caso de negocio basado en transacciones, es recomendable guardar la hora y no sólo la fecha. Esto permite mas libertad y menos problemas a la hora de la implementación. En nuestro caso se vio reflejado en la necesidad de asumir o restringir ciertas libertades a los clientes, como por ejemplo la posibilidad de hacer múltiples pagos en un mismo día.