

Practica 3 - BASES DE DATOS 2

Hayk Kocharyan
757715@unizar.es

Juan José Tambo Tambo
755742@unizar.es

Pedro Tamargo Allué
758267@unizar.es

Jesús Villacampa Sagaste
755739@unizar.es

6 de mayo de 2020

Índice

1. Esfuerzos invertidos	1
2. Parte 1 - Esquema conceptual y lógico de la base de datos relacional diseñada en la práctica anterior	1
3. Parte 1 - Determinación del esquema lógico y conceptual de la base de datos Oracle a integrar	4
4. Parte 1 - Mejoras sugeridas para la base de datos a integrar	4
5. Parte 1 - Definición e implementación del esquema global en Oracle	4
6. Parte 2 - Enunciado de un problema de diseño de bases de datos	4
7. Parte 2- Esquema conceptual para el problema enunciado	5
8. Parte 2 - Esquema lógico 1 para el problema enunciado	5
9. Parte 2 - Esquema lógico 2 para el problema enunciado	5
10. Parte 2 - Esquema global y su implementación en PostgreSQL	5
11. Actualizaciones de datos sobre el esquema global	5

1. Esfuerzos invertidos

Los esfuerzos invertidos por cada integrante del equipo son:

- Hayk:
- Juanjo:
- Jesús:
- Pedro:

2. Parte 1 - Esquema conceptual y lógico de la base de datos relacional diseñada en la práctica anterior

Mostrar aquí el ER de la práctica anterior y contar su traducción a SQL diciendo las decisiones de diseño de forma DETALLADA.

Durante el desarrollo de la práctica 1 se diseñó una base de datos para un banco que quería gestionar cuentas con múltiples propietarios, diferentes tipos de cuentas (cuentas ahorro y cuentas corrientes), operaciones (transacciones entre cuentas, o movimientos de dinero en efectivo) y las sucursales de la entidad.

En la Figura 1 se puede observar el esquema entidad relación sobre el problema planteado. Se ha planteado las relaciones entre los distintos tipos de cuentas como una generalización, ya que todas comparten ciertos atributos como el número de cuenta, el *IBAN*, la fecha de apertura y el saldo restante. Esta generalización es disjunta ya que no se considera posible la capacidad de que una cuenta pertenezca a los dos tipos de entidades al mismo tiempo. También, se trata de una generalización obligatoria ya que no se considera el caso de que una cuenta no pertenezca a algunos de las entidades derivadas de *Cuenta*.

Con transacción ocurre lo mismo, una transferencia entre cuentas y las operaciones de retirada o ingreso de efectivo se pueden generalizar en una nueva entidad *Transacción* con los atributos comunes a ambas entidades, tales como: el número de la transacción, la fecha y hora de la misma, su importe y una descripción a modo de concepto. Se trata de una generalización disjunta ya que una transferencia no puede pertenecer a los dos subtipos a la vez. También, se trata de una generalización obligatoria ya que en el contexto del problema no tiene sentido que exista una transferencia que no pertenezca a *Operación (operaciones en efectivo)* o a *Transferencia (transferencia de saldo entre cuentas)*.

Se ha decidido que *Transacción* debía ser débil respecto a *cuenta* ya que depende en existencia e identificación de *cuenta*, por lo tanto la relación *Realizar* es una relación 1:N entre *Cuenta* y *Transacción*. No obstante, existe una transacción que no indica la debilidad de la entidad *Transacción* respecto a *Cuenta*, la relación *Recibir*, una relación 1:N que relaciona las entidades *Cuenta* y *Transferencia*, y cuyo significado es relacionar una transferencia con la cuenta beneficiaria.

Los clientes titulares de una cuenta se reflejan en la entidad *Cliente*, que almacena el DNI, el nombre, los apellidos la dirección y el email. Se ha decidido que el *DNI* sea la clave primaria ya que es único para los ciudadanos.

La relación de *Cliente* con *Cuenta* se encuentra en *Poseer*, una relación M:N que posibilita que una cuenta tenga más de un titular.

Para el almacenamiento de las sucursales de la entidad bancaria, se ha diseñado una entidad *Sucursal*, que almacena el código de la entidad, su dirección postal y el teléfono de la oficina. Se ha decidido que el código de la sucursal sea la clave primaria que identifique a las sucursales ya que dentro de la misma entidad bancaria no existen dos sucursales con el mismo código.

Se ha considerado que las transacciones se tienen que realizar en una sucursal, por lo tanto existe una relación entre *Transacción* y *Sucursal*. También, una cuenta corriente debe ser abierta en una determinada sucursal de la entidad.

En la traducción de las generalizaciones del esquema conceptual al esquema lógico (modelo relacional) se han escogido dos estrategias. En la generalización de *Cuenta* se ha mantenido la entidad, y las entidades derivadas de la misma solamente almacenan los atributos necesarios que no pertenecen a *Cuenta*.

Por otra parte, en la generalización de *Transacción* se ha optado por eliminar la entidad generalizada, obteniendo así dos entidades independientes y duplicando algunos de sus atributos. **Leyendo esto me doy cuenta que nos hacía falta un *Trigger*.** En esta misma generalización se ha traducido la relación 1:N entre *Cuenta* y *Transacción* como un atributo (el número de cuenta) en las tablas derivadas de *Transacción*, siendo una referencia a la cuenta

que ha realizado la transacción.

¿CÓMO LES PONEMOS EL ESQUEMA LÓGICO? Un diagrama?

Qué hacemos con esta figura? La llevamos a otro sitio?

Diagrama ER BANCO

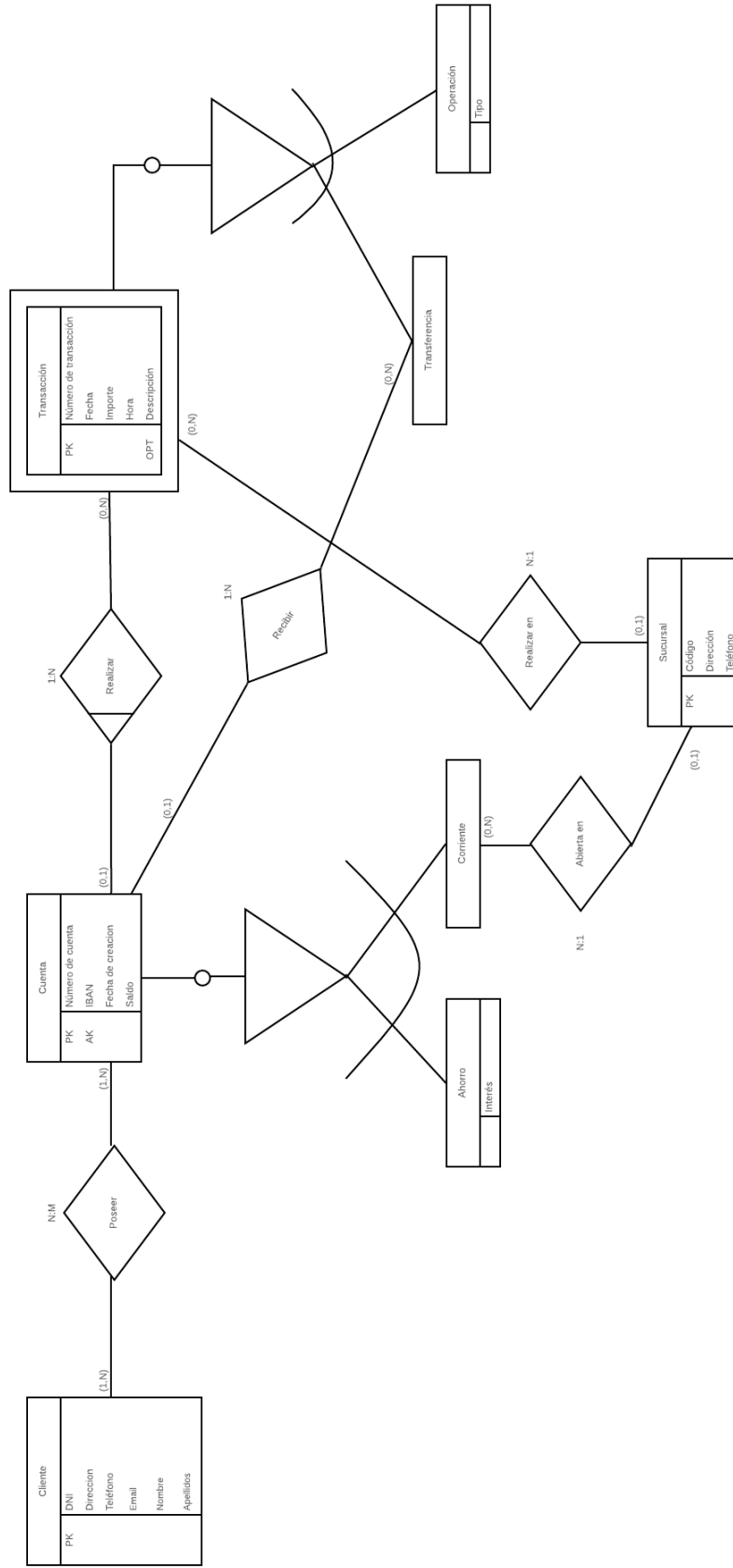


Figura 1: Esquema ER de la base de datos diseñada en la práctica 1

3. Parte 1 - Determinación del esquema lógico y conceptual de la base de datos Oracle a integrar

Esquema ER de la base de datos a integrar y PONER LAS CONSULTAS CON LAS QUE HEMOS SACADO LAS COSAS.

Igual estaría bien comentar de donde hemos sacado la información.

4. Parte 1 - Mejoras sugeridas para la base de datos a integrar

Durante el interrogatorio al catálogo de datos de la base de datos de *Banquete* se han encontrado diversas anomalías y diferencias respecto a nuestra base de datos.

En la base de datos a integrar una cuenta solo puede disponer de un titular. Además en *Titular* (se corresponde con *Cliente* de *Banquito*), no dispone de un atributo destinado para la dirección, si no que cuenta con una clave ajena a una entidad *Dirección*, que no está relacionada con su entidad *CodPostal*. Esta entidad *CodPostal* almacena: calle, ciudad y código postal, por lo tanto a partir de una dirección si que se puede obtener su código postal pero al no estar relacionado habría que realizar consultas extras y a la par de que la información se encuentra duplicada en dos tablas de la base de datos (calle y ciudad).

La entidad *Sucursal* no dispone de un atributo como clave ajena de *Dirección*, en este caso solamente se dispone de una restricción de consistencia sobre su valor que comprueba que este atributo no sea *NULL*. Esto puede duplicar la información existente en *Dirección* y en este atributo *Dir*.

En la entidad *CodEntidades* encontramos el problema de que no está relacionada con ninguna otra. En este caso no duplica información y en ella se encuentran los códigos de las distintas entidades bancarias conocidas por el sistema de cuentas de *Banquete*.

La entidad *OpTransferencia* tiene un atributo destinado a albergar el número de cuenta corriente de la cuenta destino, no obstante, al igual que con *Sucursal*, este atributo no es una clave ajena de una *Cuenta*. Esto puede ser debido a que *Banquete* (transferencia entre *Banquete* y *Banco Generosidad*) ofrece la posibilidad de realizar transferencias entre cuentas que no tienen porque pertenecer a su entidad. Este aspecto no se tuvo en cuenta a la hora de diseñar la base de datos de *Banquito*.

Un problema similar existe en la entidad *OpEfectivo*, ya que tiene un atributo destinado a identificar la *Sucursal* en la que se ha realizado una operación con dinero en efectivo, pero este campo no se trata de una restricción de integridad referencial (clave ajena) si no que se trata de una restricción de consistencia que se impide que este campo sea *NULL*.

Sobre la generalización de *Cuenta* en *CuentaCorriente* y *CuentaAhorro* se ha descubierto que en la base de datos de *Banquete* no se trata de una generalización disjunta si no que existen cuentas que pueden pertenecer a ambos tipos al mismo tiempo.

Hablar aquí de los problemas que hemos encontrado y lo que supone que esté mal.

5. Parte 1 - Definición e implementación del esquema global en Oracle

Se me ocurre que podríamos explicar lo de que no ponemos como una entidad aparte lo de *Dirección* exponiendo que no queríamos “romper” el mal trabajo que hicieron con *Sucursal* y que queríamos unificarlo. Poner aquí el esquema ER y decir las decisiones de diseño (y el por qué).

6. Parte 2 - Enunciado de un problema de diseño de bases de datos

Se desea diseñar una base de datos para gestionar la información inherente a un hospital. En el hospital trabajan médicos y enfermeros. Los médicos tienen asignados una serie de pacientes. Los enfermeros tienen asignada una planta.

La información sobre los trabajadores del hospital deberá contener información sobre su DNI, nombre, apellidos, fecha de nacimiento, dirección, teléfono de contacto. Los médicos deberán almacenar su especialidad y su número de colegiado. Los enfermeros almacenarán el servicio al que están asignados (planta, hematología, diálisis...).

Nuestro hospital está dividido en plantas, cada planta alberga pacientes ingresados y almacena la información relativa al uso destinado para esa planta.

Los pacientes almacenan su DNI, nombre, apellidos, fecha de nacimiento, dirección y teléfono. Disponemos de un historial clínico que almacena la información sobre los diagnósticos realizados a los pacientes.

Poner aquí ese enunciado que hemos redactado

7. Parte 2- Esquema conceptual para el problema enunciado

Poner aquí los dos esquemas ER y explicar que se ha tomado esta decisión para dificultar la integración.

8. Parte 2 - Esquema lógico 1 para el problema enunciado

Pues explicar las decisiones de diseño de este (CHUS Y RATAMBO).

9. Parte 2 - Esquema lógico 2 para el problema enunciado

Pues explicar las decisiones de diseño de este (PEDRO Y HAYK).

10. Parte 2 - Esquema global y su implementación en PostgreSQL

Puse aquí tenemos la incertidumbre de si querrán que sean vistas o una base de datos completa, pero de todas formas habrá que explicar las decisiones de diseño.

11. Actualizaciones de datos sobre el esquema global

Viabilidad? Pues depende de dónde quieras guardarlo. Pero esto habrá que hablar.