

## Práctica 2: Instalación de Sistemas Gestores de Bases de Datos y Diseño de Bases de Datos con Características de Orientación a Objetos

### Bases de Datos 2

Grado en Ingeniería Informática  
Departamento de Informática e Ingeniería de Sistemas  
Universidad de Zaragoza

## 1.- Objetivos

En esta práctica se diseñará e implementará una base de datos relacional en la que será necesario adecuar algunas características de orientación a objetos reflejadas en el esquema conceptual (por ejemplo, jerarquías) para su representación utilizando el modelo relacional. Además, se proporcionarán también soluciones de acuerdo con el modelo objeto/relacional y el modelo orientado a objetos puro.

En mayor detalle, los objetivos de la práctica son:

1. Realizar el **diseño conceptual** de una base de datos adecuada para resolver un determinado problema, utilizando el **modelo E/R** (Entidad/Relación) extendido.
2. Realizar el correspondiente **diseño lógico** utilizando el modelo relacional (transformando el esquema conceptual a un esquema relacional) e **implementar dicho esquema** (transformándolo en **código SQL** apropiado) considerando un SGBD relacional.
3. Realizar el correspondiente **diseño lógico** utilizando el modelo objeto/relacional (transformando el esquema conceptual a un esquema objeto/relacional, es decir usando el estándar SQL:1999) e implementar dicho esquema, transformándolo en código SQL apropiado para el gestor de base de datos objetivo. Analizar y probar el soporte de las características del modelo objeto/relacional en los distintos gestores, teniendo en cuenta que puede haber diferencias entre diversas versiones del SGBD utilizado.
4. Introducir un número suficiente de datos de ejemplo (investigando y utilizando estrategias y herramientas que permitan generar datos).
5. Implementar y probar un esquema orientado a objetos utilizando db4o con Java (modelo orientado a objetos puro). Se pretende únicamente tener una toma de contacto con el modelo orientado a objetos puro, lo cual es interesante especialmente porque algunos productos comerciales más modernos (por ejemplo, *Caché de InterSystems* y *Actian NoSQL Database 10*) incluyen características de dicho modelo.
6. Desarrollar la capacidad de buscar documentación relevante sobre los gestores, experimentar con los mismos y resolver los problemas que puedan surgir de forma autónoma.

Si lo estimas oportuno, también puedes hacer algún comentario referente al diseño físico en la memoria de la práctica.

## 2.- Contexto del Problema

Se pretende desarrollar una base de datos para almacenar información básica de una entidad bancaria. Para cada cliente se desea disponer de la siguiente información: DNI, apellidos, nombre, edad, dirección, email (si lo tiene) y teléfono de contacto. Cada cliente podrá ser titular de varias cuentas bancarias y una cuenta bancaria podrá tener varios titulares.

Para cada cuenta se almacena la siguiente información: el número de cuenta, el IBAN, la fecha de creación, el saldo actual, la lista de titulares de la misma y el historial de operaciones realizadas con dicha cuenta. Existen dos tipos de cuentas: cuentas de ahorro y cuentas corrientes. Las cuentas corrientes son las cuentas bancarias tradicionales y están ligadas a una cierta oficina del banco. Por el contrario, las cuentas de ahorro son cuentas electrónicas (contratadas a través de Internet y, por tanto, no ligadas a una oficina física de la entidad bancaria). Las cuentas de ahorro que se van a manejar proporcionan un cierto tipo de interés por el dinero en ellas invertido, que se negocia directamente con cada cliente. Los intereses se devengan mensualmente, existiendo un procedimiento que se ejecuta cada noche para actualizar los saldos de las cuentas de ahorro en las que sea necesario acumular intereses.

TRIGGER NO?

Cada operación bancaria se identifica con un código numérico que indica el número de operación realizada con dicha cuenta. Las operaciones que se pretenden contemplar por el momento son de tres tipos: ingreso de dinero, retirada de dinero, y transferencia a otra cuenta de la misma entidad. Para cada operación se desea almacenar la información relevante para la misma: tipo de operación, fecha y hora de realización de la operación, cantidad involucrada (ingresada, retirada o transferida), cuenta desde la que se ordena la operación, sucursal en la que se realiza la operación en efectivo (en el caso de ingresos y retiradas de dinero), y cuenta receptora (en caso de que se trate de una transferencia). Además, de forma opcional, toda operación puede tener una descripción de texto libre asociada (por ejemplo, indicando brevemente el concepto de la operación).

De las oficinas se desea almacenar el código de oficina (4 dígitos), su dirección y su teléfono. Cuando se crea una cuenta (tanto si es una cuenta corriente abierta en una oficina física como si es una cuenta de ahorros abierta a través de Internet) el saldo inicial de la misma es de 0 euros. Las cuentas no tienen actualmente comisión de mantenimiento y no se contempla la posibilidad de saldos negativos.

## 3.- Aclaraciones

Es conveniente realizar las siguientes aclaraciones:

- Se ha planteado un escenario simplificado referente a las necesidades de información básicas de una entidad bancaria. Evidentemente, un escenario más complejo plantearía una problemática que sería difícil abordar en una única práctica de la asignatura. No obstante, es necesario acercarse lo máximo posible

a la realidad que describe el enunciado y justificar y aclarar cualquier hipótesis razonable que se haga acerca del problema planteado. Además, se aconseja consultarlas con el profesorado.

- Se implementará la solución sobre algunos de los gestores de bases de datos instalados y configurados en la práctica 1. En concreto, se deberá implementar la solución sobre PostgreSQL, Oracle y DB2, además de desarrollar un prototipo de persistencia utilizando db4o.
- Es importante extraer conclusiones: resultados obtenidos, indagaciones realizadas, diferencias observadas entre los distintos Sistemas Gestores de Bases de Datos (SGBD) en cuanto a sus posibilidades para implementar modelos objeto-relacionales y orientados a objetos, dificultades encontradas, valoración de los aprendizajes o mejoras de habilidades conseguidas, opinión personal, etc. En la medida de lo posible, y dentro de lo razonable, la memoria de prácticas contendrá el suficiente nivel de detalle como para poder reproducir el trabajo realizado.

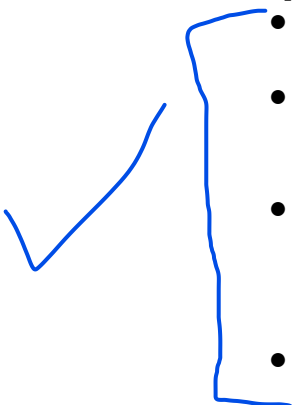
## 4.- Entrega de la Práctica

Se debe entregar un fichero zip denominado p2-<nip>.zip (donde <nip> representa el NIP del coordinador/responsable del grupo) con el siguiente contenido:

- Fichero autores.txt: contendrá el nombre y apellido de los autores de la práctica y sus NIP.
- Directorio *fuentes*: contendrá todo el código desarrollado para la práctica:
  - Ficheros con las implementaciones de bases de datos relacionales:
    - Fichero bd-bancaria-oracle.sql: código SQL para la creación de la BD en Oracle.
  - Ficheros con las implementaciones de bases de datos objeto/relacionales para diversos gestores:
    - Fichero bd-bancaria-or-oracle.sql: código SQL para la creación de la BD en Oracle.
    - Fichero bd-bancaria-or-postgresql.sql: código SQL para la creación de la BD en PostgreSQL.
    - Fichero bd-bancaria-or-db2.sql: código SQL para la creación de la BD en DB2.
  - Fichero bd-bancaria-pruebas.sql: código SQL con las pruebas realizadas para las implementaciones con el modelo relacional (inserción de datos, consultas de ejemplo, etc.). Si es necesario, se incluirá un fichero de este tipo para cada uno de los SGBD utilizados.
  - Fichero bd-bancaria-pruebas-or.sql: código SQL con las pruebas realizadas con el modelo objeto/relacional (inserción de datos, consultas de ejemplo, etc.). Si es necesario, se incluirá un fichero de este tipo para cada uno de los SGBD utilizados.
  - Fichero proyecto-db4o.zip con el código fuente de las clases utilizadas para la implementación y pruebas en db4o, así como con los scripts de compilación y ejecución necesarios.

Importante: cada uno de los ficheros .sql indicados deberá poder ejecutarse sin problemas como fichero de script con independencia del estado inicial de la base de datos destino, de forma que en el script hay que asegurarse de limpiar adecuadamente el espacio de tablas de la base de datos para evitar conflictos con objetos del mismo nombre que pudieran existir previamente (tablas, restricciones, etc.): como ejemplo de prueba, debe ser posible ejecutar dos veces seguidas el script y no producirse errores.

- Fichero memoria-p2.pdf, conteniendo la memoria de la práctica, de acuerdo con las instrucciones generales de elaboración de prácticas. La memoria de esta práctica deberá contener al menos las siguientes secciones:

- 
- **“Esfuerzos invertidos”**: horas invertidas por cada integrante del grupo en las distintas tareas.
  - **“Diseño Conceptual de la Base de Datos”**. Para el diseño conceptual se utilizará el modelo E/R. En el esquema conceptual figurará al menos una jerarquía.
  - **“Diseño Lógico de una Base de Datos Relacional”**. Debe incluir un análisis y justificación adecuada de la forma en la que se han transformado las jerarquías (especialización/generalización) del esquema conceptual a un esquema relacional.
  - **“Implementación con el Modelo Relacional”**: código SQL para la creación de la base de datos relacional.
  - **“Diseño Lógico de una Base de Datos Objeto/Relacional”**. Se proporcionará un esquema objeto/relacional apropiado (siguiendo, por tanto, el estándar SQL:1999), analizando y justificando las decisiones tomadas.
  - **“Implementación con el Modelo Objeto/Relacional”**: código SQL para la creación de la base de datos en los distintos gestores utilizando el modelo objeto/relacional adaptado a las funcionalidades ofrecidas por dichos gestores, con los comentarios y explicaciones que resulten relevantes.
  - **“Generación de Datos y Pruebas”**: código SQL para insertar datos de prueba y consultas de ejemplo para probar la base de datos. En este apartado se deben analizar estrategias y herramientas para la generación de datos de ejemplo.
  - **“Implementación con db4o”**: código Java junto con los comentarios y explicaciones que resulten relevantes. Se deben incluir scripts de compilación y ejecución.
  - **“Comparación de los SGBD”**: descripción acerca de las diferencias observadas entre los distintos gestores, en cuanto al soporte del modelo relacional, el modelo objeto/relacional, y otros aspectos relevantes en el contexto de la práctica.

#### Instrucciones de envío:

El fichero anterior se someterá a través de *Moodle* (se habilitará una opción a tal efecto) utilizando la cuenta del coordinador/responsable del grupo.

Hay que asegurarse de que la práctica funciona correctamente y que será fácilmente reproducible (incluir ficheros de scripts para Windows y Linux). También es importante someter código limpio (donde se ha evitado introducir mensajes de depuración y comentarios que no proporcionan información relevante).

Fecha límite de entrega:

La fecha límite de entrega será el día anterior al de realización de la práctica 3.