

Planteamiento General

El desarrollo del presente proyecto consiste en la implementación y explotación de bases de datos relacionales. El ejercicio comprende las siguientes fases:

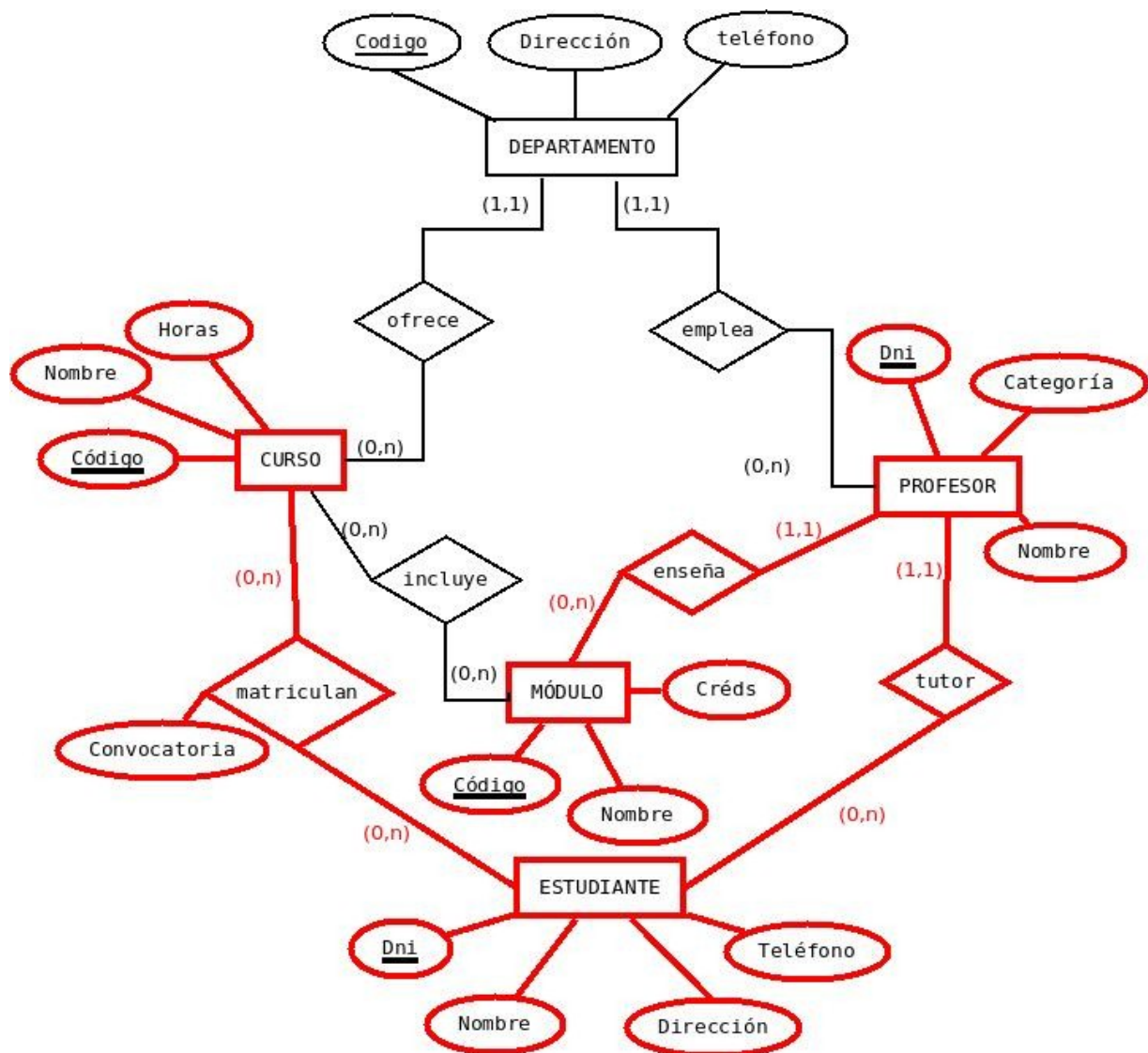
1. Implementación de una base de datos relacional a partir de una especificación conforme al Modelo Entidad Relación que emplea como notación de cardinalidad de las relaciones la de mínimas y máximas según Chen
2. Manipulación de una base de datos relacional mediante instrucciones de SQL
3. Explotación de una base de datos relacional desde un lenguaje de alto nivel. En este caso el lenguaje elegido es Java

Consideraciones Particulares

1. El desarrollo del proyecto podría realizarse en cualquier gestor relacional. Sin embargo, se va a utilizar *MySQL* debido principalmente a que es el Sistema Gestor de Bases de Datos Relacionales (SGBDR) *Open Source* más popular en la actualidad. Así mismo, se va a utilizar la base de datos de demostración de *MySQL* denominada *Sakila*. Tanto en la [página web](#) como en el [ftp](#) de la asignatura está disponible información complementaria con los enlaces al software y a la documentación del gestor de base de datos *MySQL*
2. Los alumnos realizarán el proyecto en grupos formados por los profesores. Aunque este aspecto se detalla más adelante, la entrega electrónica de los resultados del proyecto requiere un usuario/password en el servidor *MySQL* de la asignatura. Por tanto, obtener acceso al servidor es un requerimiento previo imprescindible para la realización del proyecto
3. Las entregas electrónicas se consideran un ejercicio de manipulación de una base de datos relacional por lo que representan un elemento más de evaluación
4. Se recomienda desarrollar los ejercicios del proyecto contra una instancia propia de *MySQL* y, finalmente, realizar la ejecución definitiva contra la instancia de *MySQL* disponible en el servidor *telemaco.ls.fi.upm.es*. Llegados a este punto, conviene tener en cuenta que la base de datos del servidor simula un funcionamiento *on-line*, es decir, se están produciendo inserciones, borrados y actualizaciones que afectan a los resultados que se obtienen en los ejercicios. Por consiguiente, podría suceder que en instantes diferentes las consultas devuelvan resultados distintos y no por ello se deben considerar incorrectas. Más bien todo lo contrario, por ello se recomienda probar las consultas en el servidor antes de realizar cualquiera de las entregas

Paso a Tablas

Esta parte consiste en la implementación de una base de datos relacional que responda al modelo Entidad Relación (MER) siguiente:



La primera parte del primer ejercicio del proyecto consiste en escribir las instrucciones DDL para implementar la Base de Datos especificada por el DER adjunto. El resultado se entregará en papel impreso y vía ftp en un pdf donde se incluirán las instrucciones para implementar el **modelo completo**. La solución debe usar los órdenes CREATE TABLE y ALTER TABLE de modo que la ejecución sea independiente del orden. En este apartado tampoco se establece ninguna limitación relativa a la denominación de los objetos

La segunda parte de este apartado únicamente requiere entrega electrónica. El ejercicio consiste en escribir las instrucciones CREATE TABLE que requiere el paso a tablas de los objetos que en el subdiagrama anterior aparecen destacados **en color rojo**. A diferencia de la primera parte del

ejercicio, en las instrucciones SQL se permite únicamente la orden CREATE TABLE (no se admite ALTER TABLE) y se entregarán de forma electrónica atendiendo a los siguientes requisitos:

- cada tabla atenderá al siguiente esquema de denominación (ojo a las mayúsculas)
 - nombre de la tabla i-ésima: DDLi. Por ejemplo, la primera tabla será DDL1
 - atributo j de la tabla i: DDLiAj. Por ejemplo, el tercer atributo será DDL1A3
- el índice i refleja el orden de creación de las tablas necesario para satisfacer las dependencias que se deriven de las definiciones de integridad de las tablas
- los tipos de los atributos de las tablas serán int o varchar
- como se puede ver en la descripción de la entrega electrónica, todas las órdenes CREATE se entregarán a la vez, en un único string, donde cada orden finaliza con un carácter ';'.
- el string tampoco incluirá comentarios, sólo SQL
- es fundamental que las órdenes en el string figuren en el orden correcto puesto que, como es sabido, la ejecución de una orden CREATE TABLE fallará si no se satisfacen correctamente las restricciones de integridad. Dado que el objetivo de este apartado del proyecto es implementar completo el submodelo destacado en color rojo, el apartado completo se considerará incorrecto aunque falle sólo una de las órdenes que contiene

Parte II

Consultas DML

La segunda parte del proyecto consiste en escribir las instrucciones DML que resuelvan determinados accesos a una base de datos, en este caso, *Sakila*. En esta ocasión tanto la entrega en papel como la entrega electrónica plantean el mismo problema y ambas son obligatorias

En lo que sigue se muestran las consultas a resolver y el formato de salida que deben generar. Las palabras que aparecen resaltadas en color rojo son alias en las consultas

Consulta 01

Informe del total recaudado por el vídeo-club para todas las películas que alquila:

film_id	tittle	Total

Consulta 02

Recaudación por alquileres de la película 'ACADEMY DINOSAUR':

film_id	tittle	Total

Consulta 03

Cantidad recaudada en el videoclub por cada cliente :

customer_id	first_name	last_name	Total

Consulta 04

Obtener la máxima recaudación del videoclub para cualquier cliente:

GastoMax

Consulta 05

Informe de las películas que fueron devueltas con más de 9 días de demora:

title	return_date	rental_date

Consulta 06

Nombre completo (por ejemplo, “Laura Rodríguez”) de los clientes que hayan alquilado la película 'ACADEMY DINOSAUR' indicando la fecha de alquiler y la de entrega. El listado se mostrará ordenado por el apellido del cliente. El nombre completo de los clientes se obtiene utilizando la función [CONCAT_WS\(separator, str1, str2, ...\)](#) de MySQL:

Historial ordenado por copia y fecha de los alquileres del título “ACE GOLDFINGER”:

Cliente	return_date	rental_date

Consulta 07

Realizar las operaciones pertinentes contra la base de datos para completar el alta de un nuevo cliente del vídeo club.

Como en el caso del apartado correspondiente al DDL, es fundamental que las órdenes en el string de entrega figuren en el orden correcto puesto que la ejecución fallará si no se satisfacen correctamente las restricciones de integridad definidas sobre las tablas

Aplicación en lenguaje Java

La tercera y última entrega del proyecto consiste en la realización de una aplicación escrita en lenguaje Java que se conecte a un SGBD y que sea capaz de recuperar y almacenar datos en alguna de las bases de datos que alberga el gestor.

La aplicación constará de una sola clase de nombre **ProyectoBD** (OJO a las mayúsculas) con un método *main* que recibirá 4 parámetros a través de la línea de comandos. La forma de invocar al programa y la descripción de los parámetros se encuentra a continuación:

```
java ProyectoBD host usr passwd tabla
```

host – es el nombre o la dirección IP de la máquina donde reside el SGBD

usr – identificador del usuario en el SGBD indicado en el parámetro host

passwd – contraseña del usuario en el SGBD indicado en el parámetro host

tabla – nombre de una tabla de la base de datos que se utilizará para la consulta de sus metadatos. Esta tabla reside en una base de datos de nombre JavaBD.

El programa debe de realizar dos acciones:

1. Consultar los metadatos de la tabla que se pasa en el parámetro *tabla* ubicada en una base de datos denominada **JavaBD**. El resultado obtenido con esta acción se debe entregar **desde el propio** programa invocando al mismo procedimiento **Proy2012.entrega** ya utilizado en los apartados anteriores. En este caso la llamada al procedimiento entrega se realizará de la siguiente forma:

```
call entrega(ítem, resultado)
```

siendo :

ítem – la entrega de este resultado corresponde con el ítem número 9

resultado – la cadena de texto que se detalla a continuación:

```
NombreTabla;NCols;PK1,...,PKn;Atr1,TipoAtr1,LongAtr1;...;
Atrn,TipoAtrn,LongAtrn
```

NombreTabla – Debe contener el nombre de la *tabla* sobre la que se consultan los metadatos

NCols – Debe contener el número de columnas de la *tabla* consultada

PK_i – Atributo *i*-ésimo que forma la clave principal de la tabla. Si la clave principal está formada por más de un atributo hay que separarlos por comas

Atr_i – Nombre de la columna *i*-ésima de la *tabla* consultada

TipoAtr_i – Tipo de dato de la columna *i*-ésima de la *tabla* consultada

LongAtr_i – Longitud de la columna *i*-ésima de la *tabla* consultada. Por ejemplo si la columna 2 es de tipo VARCHAR(25), LongAtr₂ tiene que valer 25

Es muy importante notar que esta acción debe ser genérica, esto es, se debe poder invocar sobre cualquier tabla con la única condición de que resida en la base de datos **JavaBD**.

A modo de ejemplo, si el programa consultase los metadatos de una tabla **JavaBD.Ejemplo** cuya orden de creación es la que se muestra a continuación:

```
CREATE TABLE `Ejemplo` (
  `Name1` varchar(20) NOT NULL,
  `Name2` varchar(10) NOT NULL,
  `AutoPrune` tinyint(4) DEFAULT '0',
  `FileRetention` bigint(20) unsigned DEFAULT '0',
  `JobRetention` bigint(20) unsigned DEFAULT '0',
  PRIMARY KEY (`Name1`,`Name2`),
  UNIQUE KEY `Name` (`Name1`)
)
```

la cadena *resultado* a almacenar durante la entrega sería la siguiente:

```
"Ejemplo; 5; Name1, Name2; Name1, VARCHAR, 20; Name2, VARCHAR, 10;
AutoPrune, TINYINT, 3; FileRetention, BIGINT UNSIGNED, 20; JobRetention,
BIGINT UNSIGNED, 20"
```

2. Insertar en la tabla *customer* de la base de datos *sakila* un nuevo cliente teniendo en cuenta todas las restricciones (a nivel de columna y a nivel de integridad referencial) establecidas en dicha tabla. [Nótese que esta misma operación se realizó en la segunda parte del proyecto.](#) La diferencia es que ahora se realiza desde un programa.

NORMAS Y PROCEDIMIENTO DE ENTREGA

La entrega de cualquiera de las tres partes que constituyen el proyecto se compone de dos elementos complementarios: una memoria en papel y una entrega electrónica. La memoria impresa debe incluir el código SQL (DDL, DML) y del programa Java así como una breve documentación. A través del ftp se entregará un fichero denominado **2012XX.pdf** (el código de grupo) que corresponda con la memoria impresa. A partir de este punto la descripción afecta únicamente a la entrega electrónica de cada una de las partes del proyecto.

Cada equipo de trabajo tiene asignadas sendas cuentas (usuario y password) en el servidor ftp y en el gestor MySQL disponibles en la máquina *telemaco.ls.fi.upm.es*. Estas cuentas posibilitarán, respectivamente, la entrega de todos los documentos y resultados de las diferentes partes del proyecto así como acceder a la base de datos *sakila* contra la cual se deberán lanzar las consultas indicadas en el apartado siguiente. Además de *sakila* en el servidor MySQL existen tres bases de datos denominadas *Proy2012* y *BDcod_grupo*, dispuestas respectivamente como base de datos para las entregas y base de datos de trabajo de cada equipo y *JavaBD* para la tercera parte del proyecto. En particular, la entrega electrónica consiste en insertar en la tabla *Proy2012.EntGrado* el SQL de las consultas y el resultado que produce el programa en Java.

La entrega en MySQL se puede realizar repetidas veces, actualizándose las entregas previas, y consiste en invocar el procedimiento denominado “Entrega” según lo siguiente:

- **call Entrega (item, texto)**
 - item 1 (único) para todas las órdenes DDL (CREATE TABLE)
 - items de 2 a 7, ambos inclusive, para las consultas DML
 - item 8 para las órdenes que requiera la operación de alta de un cliente
 - item 9 para los resultados del programa Java
- **Ejemplos:**
 - **Entrega de la consulta Q3:** call Proy2012.Entrega (3, 'SQL de la consulta Q3');
 - **Entrega del resultado del programa Java:** call Proy2012.Entrega (9, 'Resultado Java');

Es muy importante comprobar si las entregas han sido registradas en la base de datos. Para ello, después de cada entrega, se recomienda hacer una consulta contra la vista denominada *Proy2012.vista_entregas_grado*.

Por último, pero no menos importante, como primera acción se recomienda a todos los equipos cambiar la clave de acceso de su usuario en el servidor. El cuadro siguiente muestra un ejemplo

```
santiago@Folio13:~$ mysql -h telemaco.ls.fi.upm.es -u 201200 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14173
Server version: 5.5.27 MySQL Community Server (GPL)
(...)
mysql> SET PASSWORD = PASSWORD('contraseña_nueva');
mysql> use BD201200;
Database changed
```

Es responsabilidad de cada equipo velar por la privacidad del acceso a sus cuentas pues, como se deduce de lo expuesto en el presente documento, es **imprescindible** acceder al gestor MySQL para poder realizar el proyecto. En particular, cada grupo obtendrá la contraseña para acceder al servidor

ftp y entregar el proyecto consultando convenientemente la tabla de entregas.

En lo que se refiere a la entrega electrónica de la tercera parte del proyecto, la que corresponde al desarrollo de un programa en lenguaje Java, procede entregar un único fichero conteniendo el código fuente necesario para la realización de las 2 acciones descritas en el enunciado. El nombre de dicho fichero será **ProyectoBD.java**. En este caso, la entrega del fichero con el código fuente se realizará mediante *ftp* a la máquina *telemaco.ls.fi.upm.es*. El nombre de usuario para acceder al ftp es el mismo que el utilizado para acceder a MySQL y la contraseña la que aparezca en la tabla de entregas.

Para que el proyecto se pueda considerar válido el programa debe compilar y los resultados de las acciones pedidas deben ser acorde a los formatos de entrega indicados en el enunciado