

## Planteamiento General

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

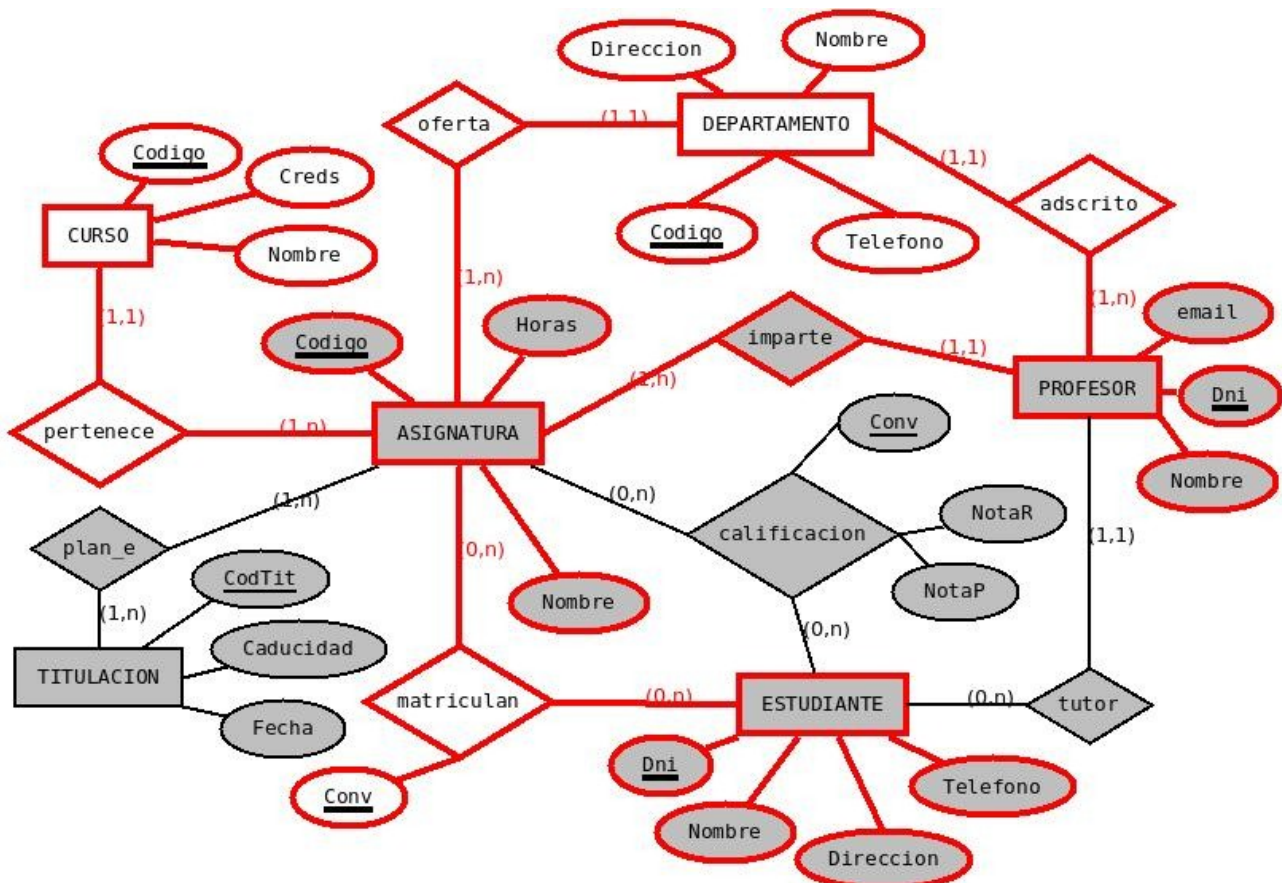
1. Implementación de una base de datos relacional a partir de una especificación conforme al Modelo Entidad Relación que emplea como notación de cardinalidad de las relaciones la de mínimas y máximas según Chen
2. Manipulación de una base de datos relacional mediante instrucciones de SQL
3. Explotación 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 conformados 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 lo que resulta imprescindible 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 denominada Parte I del proyecto incluye los dos apartados siguientes:

1. Escribir las instrucciones DDL para implementar la Base de Datos correspondiente a los **objetos con fondo gris** en 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. La solución debe usar únicamente las órdenes *CREATE TABLE* y *ALTER TABLE* de modo que la ejecución sea independiente del orden. En este apartado no se establece ninguna limitación relativa a la denominación de los objetos
2. Este apartado únicamente requiere entrega electrónica. El ejercicio consiste en escribir las instrucciones *CREATE TABLE* que requiere el paso a tablas de los objetos resaltados **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 se denominará exactamente con el mismo nombre del objeto (entidad o relación) que aparece en el DER
  - los tipos de los atributos de las tablas serán int o varchar

- 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 ni acentos, sólo expresiones SQL
- es fundamental que las órdenes en el string figuren en el orden correcto puesto que la creación de la base de datos 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, todo el apartado se considerará incorrecto aunque falle sólo una de las órdenes

## 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

Nombre completo (por ejemplo, “Laura Rodríguez”) de los clientes que viven en Londres y ordenados por el apellido del cliente

| <b>Cliente</b>  |
|-----------------|
| Laura Rodríguez |

#### Consulta 02

Número de clientes del videoclub por ciudad para aquellas ciudades que tengan más de 1 cliente:

| <b>Ciudad</b> | <b>Clientes</b> |
|---------------|-----------------|
| Nairobi       | 23              |
| Teruel        | 54              |

#### Consulta 03

Número de alquileres de películas (ordenadas por su título) que se alquilaron más de 30 veces:

| <b>Título</b>         | <b>Alquileres</b> |
|-----------------------|-------------------|
| El Recortazo          | 300               |
| Vampiros en la niebla | 234               |

### Consulta 04

Título y día del alquiler de las películas en la categoría comedia alquiladas entre el 01/06/2005 y el 31/08/2005. El informe resultante debe estar ordenado por título y fecha de alquiler

| Título                                      | Fecha      | Categoría |
|---|------------|-----------|
| Esperanza y Lucía, pánico de noche y de día | 2012-12-12 | Comedia   |
| ¿Por qué no te vas de caza?                 | 2012-21-21 | Comedia   |

### Consulta 05

Cientes que hayan alquilado más de 5 películas de acción y número de alquileres:

| Cliente                  | Alquileres |
|--------------------------|------------|
| Rigoberto R. Resultón    | 33         |
| Héctor Conde de Peñalver | 69         |

### Consulta 06

Obtener el máximo del número de veces que se ha alquilado una película de acción:

|               |
|---------------|
| <b>Maximo</b> |
| 89            |

### Alta de Cliente. Consulta 07

En este apartado se deberán implementar la operaciones pertinentes para completar el alta de un cliente **completamente nuevo** y con domicilio en una **dirección nueva** de la ciudad de Madrid. En aras de una mejor comprensión del ejercicio se detalla a continuación el escenario del problema:

El usuario de cada equipo del proyecto tiene los permisos en la base de datos *sakila* que corresponden a los dependientes del vídeo club. Así, por ejemplo, el operador no tiene posibilidad de borrar clientes. De acuerdo con esto, en caso de que se introdujeran datos erróneos al dar de alta el cliente, lo correcto es siempre actualizarlos y nunca crear otro cliente distinto con los datos corregidos. Por este motivo, en la evaluación se valorará que cada equipo sólo cree un cliente y, en todo caso, únicamente se considerará el último

Las órdenes necesarias para completar este apartado se entregarán como una sólo cadena en la que cada orden finaliza con el carácter ';'. El ejercicio requiere el uso de la función de MySQL *last\_insert\_id()*. Para terminar, es importante tener en cuenta que este cliente se utilizará posteriormente en la siguiente parte del proyecto

### 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

**customer\_id** – el código del cliente dado de alta en la Parte II del Proyecto

El programa debe 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 **ProyBBDD.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<sub>i</sub>** – 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<sub>i</sub>** – Nombre de la columna *i*-ésima de la *tabla* consultada

**TipoAtr<sub>i</sub>** – Tipo de dato de la columna *i*-ésima de la *tabla* consultada

**LongAtr<sub>i</sub>** – Longitud de la columna *i*-ésima de la *tabla* consultada. Por ejemplo si la columna 2 es de tipo VARCHAR(25), LongAtr<sub>2</sub> 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. Implementar las operaciones necesarias para:

- a) registrar el alquiler de una copia de una película por el cliente creado en el último apartado de la parte anterior del proyecto. Como entonces, el registro deberá ser único. Los datos relativos a la copia y la película son los siguientes:
  - Título de la película: 'Krakatoa, Este de Java'
  - Copia de la película: la que tiene el mismo Id que el código de grupo
- b) registrar la devolución del alquiler previo
- c) registrar el pago del alquiler. El registro debe ser único

Hay que tener en cuenta que la película (copia) solicitada podría figurar en la base de datos como no disponible (por algún error o inconsistencia) y el programa debe contemplar este supuesto. Por lo tanto, para registrar este control el programa java debe invocar adicionalmente al procedimiento *ProyBBDD.RentStatus(valor)* con *valor* igual a 0 si detecta que la copia está ocupada (no se puede realizar el alquiler) o con valor igual a 1 si la copia estaba libre y ha sido posible realizar el alquiler

Para poder realizar esta parte del proyecto hay que hacer uso de las siguientes funciones o procedimientos:

- Definidos en MySQL: *last\_insert\_id()*
- Definidos en Sakila:
  - *inventory\_in\_stock(inventory\_id)*
  - *get\_customer\_balance(customer\_id, date)*

## DETALLE DEL PROCEDIMIENTO DE ENTREGA

Las entregas del proyecto incluyen 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 cualquier explicación relativa al desarrollo del proyecto que se considere relevante. A través del ftp se entregará un fichero denominado MemoriaXX.pdf (XX, el código de grupo) que corresponda con la memoria impresa

A continuación se describe 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 la entrega de todos los documentos y resultados de las diferentes partes del proyecto. En particular, el usuario MySQL permite acceder a la base de datos del vídeo-club con el perfil de un dependiente. Esta base de datos se denomina *sakila* y contra ella se deberán lanzar las consultas indicadas en el Parte II del proyecto. Además de *sakila*, en el servidor MySQL existen otras tres bases de datos denominadas *ProyBBDD* (base de datos para la entrega), *BDcod\_grupo* (base de datos de trabajo de cada equipo) y *JavaBD* base de datos para la tercera parte del proyecto. En particular, la entrega electrónica consiste en insertar en la tabla *ProyBBDD.EntGrado* el SQL de las consultas de las Partes I y II y los resultados que produce el programa Java de la Parte III

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 ProyBBDD.Entrega (4, 'SQL de la consulta Q3');
  - **Entrega del resultado del programa Java:** call ProyBBDD.Entrega (9, 'Resultado Java');

Después de cada entrega es muy importante comprobar si ha sido correctamente registrada en la base de datos. Para ello es suficiente emitir la consulta siguiente:

```
select * from ProyBBDD.vista_entregas_grado;
```

Por último, pero lo más 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 tercera parte del proyecto, la que corresponde al desarrollo de un programa en lenguaje Java, requiere entregar vía **ftp** a la máquina **telemaco.ls.fi.upm.es** 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**.

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**

## TABLA RESUMEN DE RESULTADOS PARA LA ENTREGA

| Apartado |      | Resultados  | Formatos                      |
|----------|------|---|-------------------------------|
| I        | 1    | DDL submodelo gris y documentación en MemoriaXX.pdf | En papel y vía ftp            |
|          | 2    | DDL submodelo rojo y documentación en MemoriaXX.pdf | Entrega electrónica y vía ftp |
| II       | SQL  | DML y documentación en MemoriaXX.pdf                | Entrega electrónica y vía ftp |
| III      | Java | Fichero fuente ProyectoBD.java                      | Entrega electrónica y vía ftp |