

# Sistemas de Información

## Práctica 2

Curso 2016/17

3º curso

### El lenguaje SQL

#### Introducción

Como ya hemos aprendido, el lenguaje SQL nos permite tanto definir esquemas de bases de datos relacionales (actuando, por tanto, como un DDL – *Data Definition Language* –), como manipular los datos almacenados en dichas bases de datos (siendo, por tanto, también un DML – *Data Manipulation Language* –, o lenguaje de consulta).

Por otra parte, hablando ya de la **manipulación de datos** (DML), SQL tiene cuatro **sentencias básicas**, tres de ellas para la **modificación de datos** (**INSERT**, **UPDATE** y **DELETE**), cuya complejidad no es elevada, y una sentencia para la **consulta de datos** almacenados, que es, sin duda, la sentencia más compleja de SQL: **SELECT**.

*El lenguaje SQL es tanto un DDL como un DML. Por tanto, es el único lenguaje que debemos conocer para trabajar con bases de datos.*

Dentro de su vertiente DML, la sentencia más importante del lenguaje SQL es **CREATE TABLE**, que nos permite **crear cualquier tipo de relación que soporte el modelo relacional** (y, por tanto, **crear esquemas de bases de datos relacionales completas**, ya que, como ya sabemos, **una base de datos relacional, a nivel lógico está compuesta única y exclusivamente por relaciones**).

En esta práctica, trabajando aún con el intérprete interactivo *mysql* que nos proporciona nuestro entorno de laboratorio, avanzaremos en el conocimiento de la utilización y posibilidades de estas dos sentencias fundamentales de SQL: **CREATE TABLE** y **SELECT**.

#### Objetivos

...

Esta práctica tiene como objetivos fundamentales:

- Continuar aprendiendo los fundamentos de la creación de esquemas de bases de datos relacionales con SQL.
- Aprender a realizar consultas sencillas de una base de datos utilizando SQL.
- Comprender los recursos relacionales que nos permiten representar diferentes elementos definidos en los diagramas E-A a nivel conceptual.

#### Entorno

...

Si el alumno desea realizar las prácticas en su ordenador deberá:

1. Tener instalado el entorno MySQL.
2. Configurar dicho entorno siguiendo las indicaciones dadas en la Práctica 1

# El DDL de SQL: Definiendo esquemas de Bases de Datos

## Introducción

En esta primera parte de la práctica vamos a continuar aprendiendo las posibilidades de la sentencia `CREATE TABLE` de SQL (recuerda que nosotros la vamos a utilizar para crear relaciones, no tablas).

En concreto vamos a avanzar en el conocimiento de las restricciones adicionales sobre los datos que podemos definir (recuerda que el tipo de datos que definamos para cada atributo ya es una restricción sobre los datos).

## Creando una nueva base de datos

A lo largo de esta práctica vamos a utilizar una base de datos que almacenará información sobre películas, así como sobre los actores y directores de dichas películas. Por tanto, nuestro primer objetivo será crear dicha base de datos, cosa que vamos a ir haciendo incrementalmente para aprender los nuevos conceptos.

Nuestra primera versión básica de la base de datos va a estar formada por tres relaciones:

- Actores, que contendrá un identificador (`ID_actor`) y el nombre completo del actor.
- Directores, que contendrá un identificador (`ID_director`) y el nombre completo del director.
- Películas, que contendrá un identificador (`ID_pelicula`) y el título.

En estas relaciones utilizaremos una restricción que ya conocemos: `PRIMARY KEY`, y que nos permite poder definir las construcciones que utilizemos como relaciones y no como tablas.

Utilizaremos los identificadores definidos en cada relación como clave primaria.

Como podrás ver en la ayuda del comando `CREATE TABLE`, la restricción `PRIMARY KEY` se puede utilizar tanto a nivel de definición de columna (si afecta a una sola columna) como a nivel de definición de relación (si afecta a una o varias columnas). Utiliza ambas modalidades en la definición de las relaciones para familiarizarte con ellas.

### Ejercicio 1:

Crea una nueva base de datos y un programa SQL<sup>1</sup> que contenga las sentencias necesarias para crear las relaciones descritas en este apartado e introducir algunos datos en cada una de ellas (de 5 a 10 filas por relación serán suficientes).

<sup>1</sup> Vamos a denominar “programa SQL” a un fichero que contenga un conjunto de sentencias SQL.

## Utilizando claves foráneas (asociaciones 1-N)

Un elemento fundamental a la hora de resolver asociaciones entre diferentes entidades de nuestros diseños son las claves foráneas.

El objetivo de este ejercicio es utilizar una clave foránea para asociar a cada película la información sobre su director (vamos a suponer que una película sólo puede tener un director).

### Ejercicio 2:

Investiga el comportamiento y utilidad de una clave foránea.

Modifica el programa del ejercicio anterior para incorporar la asociación entre directores y películas.

## Utilizando claves foráneas (asociaciones M-N)

En este ejercicio vamos a resolver la asociación entre películas y actores, de tal forma que un actor puede haber participado en varias películas y en una película pueden haber participado varios actores de la base de datos (asociación M-N)

Como podrás ver en la ayuda del comando `CREATE TABLE`, las claves foráneas se pueden especificar tanto a nivel de definición de columna (si afectan a una sola columna) como a

nivel de definición de relación (si afectan a una o varias columnas). En este ejercicio utiliza la modalidad que no hayas utilizado en el ejercicio anterior con el fin de que te familiarices con ambas.

### Ejercicio 3:

Utilizando claves foráneas, modifica el programa del ejercicio anterior para incorporar la asociación entre actores y películas.

## Utilizando restricciones de unicidad

La sentencia `CREATE TABLE` nos permite también definir restricciones de unicidad, lo que nos permite evitar que para un atributo o conjunto de atributos determinado se puedan repetir valores. Es decir, lo utilizamos para definir claves candidatas.

Como podrás ver en la ayuda del comando `CREATE TABLE`, las restricciones de unicidad se pueden especificar tanto a nivel de definición de columna (si afectan a una sola columna) como a nivel de definición de relación (si afectan a una o varias columnas). En este ejercicio deberás utilizar ambas modalidades.

### Ejercicio 4:

Modifica el programa del ejercicio anterior, añadiendo un nuevo atributo a cada una de las relaciones de películas, actores y directores que contenga el número de identificación de la película en IMDb<sup>2</sup> (número que es único)

---

<sup>2</sup> IMDb es una de las más conocidas bases de datos relacionadas con el cine y está disponible en internet (<http://www.imdb.es>) Para obtener el identificador de una película, actor o director simplemente debemos buscar la página correspondiente y quedarnos con el número final. Así, por ejemplo, si la url que corresponde a la película es <http://www.imdb.com/title/tt0057012/>, sólo deberemos guardar el valor 0057012 (y omitir el resto, incluso el prefijo *tt*)

## Permitiendo (o no) valores nulos

La sentencia de creación de relaciones también nos permite definir cuándo una determinada columna puede almacenar valores nulos o no.

### Ejercicio 5:

Modifica tu base de datos para que sólo puedan almacenar valores nulos los atributos que almacenan códigos *imdb*.

## Restricciones adicionales

La sentencia `CREATE TABLE` nos proporciona un elemento adicional para restringir los valores que puede tomar un determinado atributo. Se trata de la cláusula `CHECK`, que nos permite introducir restricciones adicionales que deben cumplir los datos que se almacenen en la relación correspondiente.

### Ejercicio 6:

Modifica tu base de datos añadiendo un nuevo atributo *edad* para los directores y actores.

Asóciate a este atributo una comprobación para asegurar que sólo admita valores entre 0 y 120.

## Diseñando nuestra base de datos

Aunque el proceso normal de diseño de una base de datos supone realizar primero el diseño a nivel conceptual (diagrama E-A) y su paso posterior a un diseño relacional, en un caso sencillo como es nuestro ejemplo vamos a hacerlo al contrario.

### Ejercicio 7:

Dibuja el diagrama E-A de tu base de datos de películas. Te será útil para las siguientes prácticas.

## El DML de SQL: Realizando consultas

### Introducción

En esta parte de la práctica vamos a empezar a ver las posibilidades de la sentencia `SELECT` para realizar **consultas** de los datos almacenados en nuestras bases de datos.

Ésta es la única sentencia definida con este propósito en el lenguaje SQL. Dado que el número de posibles consultas a una base de datos es inmenso, la complejidad de esta sentencia es, necesariamente, alta, ya que nos debe proporcionar elementos suficientes para soportar cualquiera de dichas consultas. En esta primera

aproximación a la sentencia `SELECT` vamos a plantear algunas consultas de poca complejidad para empezar a practicar y conocer sus posibilidades.

En primer lugar vamos a realizar algunas consultas sobre la base de datos que acabamos de crear y, a continuación, vamos a crear una nueva base de datos más compleja para realizar sobre ella algunas consultas también un poco más complejas.

### Consultando nuestra base de datos

Durante la primera parte de esta práctica hemos ido creando nuestra primera base de datos. Una vez creada, el siguiente paso es utilizarla. Para ello vamos a realizar sobre ella una serie de con-

sultas de dificultad creciente, que van a ir necesitando utilizar cada vez más características de la sentencia `SELECT`.

## Ejercicio 8:

Realiza las siguientes consultas sobre tu base de datos de películas, directores y actores:

1. Listar todos los actores.
2. Encontrar los nombres de todos los actores de la película "La Guerra de las Galaxias".
3. Encontrar todos los actores de más de 50 años.

4. Para todas las películas mostrar su identificador, título, imdb y nombre del director.
5. Mostrar cuántas películas ha dirigido cada director.
6. Mostrar todos los actores que no han actuado en ninguna película.
7. Para un determinado actor, muestra todos los directores que le han dirigido.

## Creando las bases de datos de ejemplo

Para soportar esta parte de la práctica vamos a utilizar una base de datos más compleja que la que hemos ido definiendo y consultando en los ejercicios anteriores.

En concreto vamos a utilizar una base de datos que modele una universidad (al menos, parte de ella). Esta base de datos sigue más o menos fielmente el esquema de la que estamos utilizando como ejemplo en clase de teoría. Por tanto, conviene que imprimas la diapositiva con el esquema conceptual de dicha base de datos y lo tengas a mano cuando trabajes sobre las consultas que vamos a diseñar en esta parte de la práctica.

En estos ejercicios vamos a utilizar dos bases de datos. Ambas seguirán el mismo esquema y la diferencia entre ellas será la cantidad de datos que contendrán. Así:

- La base de datos "pequeña" contendrá unos pocos datos en cada relación que podemos analizar a simple vista y nos permitirá validar nuestras consultas, ya que los resultados que se deben obtener para cada una de ellas se pueden calcular "a mano".
- La base de datos "grande" contiene bastantes más datos sobre una universidad un tanto "extraña", ya que para tener un número gran-

de de datos, las asociaciones se han generado aleatoriamente mediante un programa. Por tanto no esperes encontrar una lógica detrás de la asignación de materias a departamentos, de las materias que son prerequisites de otras, etc. Esta base de datos, que contiene información sobre 20 departamentos, 50 docentes, 2000 alumnos, 200 materias y 100 grupos, nos permitirá utilizar las consultas sobre una base de datos de tamaño más o menos "real".

## Ejercicio 9:

Crea dos nuevas bases de datos, que llamaremos *UnivPeque* y *UnivGrande*, utilizando el programa SQL "*DLL.sql*" que puedes encontrar en Faitic.

Carga los datos correspondientes en cada una de ellas utilizando los ficheros "*UnivInsertPeque.sql*" y "*UnivInsertGrande.sql*" que también encontrarás en Faitic.

Analiza el fichero *DLL.sql* y comprueba que entiendes perfectamente todas las sentencias y cláusulas que se utilizan en él.

## Consultando las bases de datos de ejemplo

Los siguientes ejercicios definen una serie de consultas de complejidad creciente sobre las bases de datos que acabamos de definir.

Crea cada una de ellas en un fichero SQL y eje-

cútalas sobre ambas bases de datos. Puesto que son consultas, no modificarán ni el esquema de las mismas ni los datos, por lo que puedes ejecutarlas tantas veces como consideres necesario.

## Ejercicio 10:

Realizar las siguientes consultas:

1. Listar todos los docentes.
2. Encontrar el nombre de todos los docentes del departamento de Ingeniería Telemática.
3. Listar el nombre de todos los docentes del departamento de Ingeniería Telemática con salario superior a 70000 euros.
4. Para todos los docentes listar su ID, nombre, salario, departamento, edificio de su departamento y presupuesto de su departamento.
5. Encontrar los nombres de las materias del departamento de Ingeniería Telemática que tengan 3 créditos.
6. Encontrar todos los ID y nombres de materias que está cursando un alumno con ID determinado (por ejemplo 12345)
7. Listar el nombre de todos los docentes y alumnos, ordenador alfabéticamente.
8. Repetir la consulta anterior, pero mostrando el número total de créditos que está cursando el alumno (no mostrar el valor almacenado en el atributo tot\_creditos de la relación alumno\_3ciclo. Utilizar agregación sobre las materias cursadas por el alumno).
9. Repetir la consulta anterior pero mostrando el número total de créditos que cursa cada uno

de los alumnos de la base de datos, junto con el ID del alumno (no mostrar el nombre del alumno ni datos sobre los alumnos que no cursen ninguna materia)

10. Buscar los nombres de todos los alumnos que hayan cursado en algún momento alguna asignatura impartida por el departamento de Ingeniería Telemática (no deberían aparecer nombres duplicados)
11. Mostrar los IDs de todos los docentes que nunca hayan impartido una materia (ni está previsto que la impartan)
12. Repetir la consulta anterior mostrando los nombres de los docente, no sus IDs.
13. Encontrar los grupos con mayor y menor número de alumnos matriculados en nuestra universidad (sin tener en cuenta los grupos que no tienen ningún alumno).
14. Encontrar, para cada materia, todos los grupos que tengan el máximo número de alumnos (junto con dicho número) utilizando una consulta anidada.
15. Repetir la consulta anterior mostrando los resultados ordenados de mayor a menor número de alumnos matriculados.
16. Mostrar los diez grupos (y su materia correspondiente) con más alumnos matriculados de nuestra universidad.

## Resumen

Los principales resultados esperados de esta práctica son:

- Ser capaces de crear relaciones con cualquier tipo de restricción de integridad en un SGBD relacional que utilice SQL.
- Ser capaces de realizar consultas de complejidad baja-media.
- Practicar la conversión de asociaciones entre entidades a estructuras relacionales.

Como trabajo adicional del alumno, se proponen las siguientes líneas:

- Pensar en posibles consultas sobre las bases de datos que hemos utilizado e intentar implementarlas. Cuanto más se practique la definición de sentencias para realizar consultas, mejor.
- Definir relaciones en las que se utilicen tipos de datos diferentes a los utilizados hasta ahora (hemos utilizado solamente datos numéricos y cadenas de caracteres, pero los SGBD soportan muchos más)
- Diseñar un diagrama E-A para una base de datos sobre un tema de interés para el alumno e intentar transformarlo en construcciones relacionales e implementarlo sobre MySQL.