

## Bases de datos - Sesión 6: SQL LDD y LDM

*Todas las operaciones requeridas durante la práctica han de realizarse mediante **sentencias sql SIN utilizar la interfaz gráfica del Workbench**, a menos que se indique lo contrario. Guarda todas las sentencias sql que ejecutes en un script .sql. Contesta a las preguntas que se formulan a lo largo de la práctica en el mismo script en forma de comentarios. Sube el fichero al aula virtual al final de la sesión.*

Trabajaremos con el ejercicio de la plataforma de streaming que se vio en el Tema 4, en el que se obtuvieron las siguientes relaciones:

Categoría (**idcat**, descripcion)

Precio (**codiprecio**, precio, descripcion)

Cliente (**idclie**, nombre, apellidos, email, movil, fechanac, sexo, activo)

Película (**idpeli**, titulo, duracion, anyo, director, interprete1, interprete2, **codiprecio**, **idcat**)

Visualiza (**idclie**, **idpeli**, **fechapago**, ffindisponibilidad, preciopagado)

con las siguientes consideraciones:

- Los identificadores de cliente y película se asignan mediante números consecutivos.
- El campo activo sirve para saber si el cliente sigue estando registrado o no.
- Cada película está clasificada en una única categoría temática y tiene asignado un precio.
- Todos los campos son obligatorios excepto 'descripcion' en la tabla *Precio*, 'activo', 'sexo' y 'movil' del *Cliente*, 'interprete1' e 'interprete2' de *Película* y 'ffindisponibilidad' de *Visualiza*.

Ejercicio 1. Conéctate a Workbench y crear una nueva base de datos mediante la sentencia: **CREATE SCHEMA `Plataforma`;**

Actualiza el menú SCHEMAS (a la izquierda) y comprueba que se ha creado la base de datos Plataforma. Haz doble clic sobre ella para que quede seleccionada y sea más fácil trabajar (así evitas tener que escribir el nombre de la base de datos en cada sentencia SQL).

Ejercicio 2. Crea las 5 tablas de la base de datos Plataforma con un nombre específico, indicando todos sus atributos y asignando un dominio a cada uno de ellos. Además, las tablas se deberán crear con las restricciones de clave primaria. (Ayuda: **CREATE TABLE**).

Ejercicio 3. Una vez creadas las tablas nos damos cuenta de que nos faltan algunos datos. Añadir los siguientes campos (Ayuda: **ALTER TABLE ... ADD COLUMN**):

- *modopago* y *paisnacimiento* en *Cliente*. El país de nacimiento será obligatorio.
- *pais* en la tabla *Película* (este campo debe aparecer entre los campos duración y anyo).
- *comentario* y *descuentoaplicado* en la tabla *Visualiza*. El *descuentoaplicado* será un valor en % y por defecto valdrá 0.

**Nota:** en una BD relacional el orden en el que están creados los atributos no influye, este es el motivo por el cual dicha especificación no es estándar (pero MySQL sí la incluye).

Ejercicio 4. Ahora hay que añadir restricciones a las tablas creadas. Añade las siguientes restricciones (Ayuda: **ALTER TABLE ... ADD CONSTRAINT ... CHECK / UNIQUE**):

- Añade a la tabla *Cliente* una restricción que impida que se puedan insertar teléfonos móviles incorrectos, entendiendo por correctos, los que empiezan por 6 y tienen una longitud de 9 dígitos y además han de ser únicos porque no podemos tener dos clientes con el mismo móvil.
- Añade a la tabla *Cliente* una restricción que evite que el email sea incorrecto (tiene que tener una @). Además ha de ser único.
- Los valores del campo sexo de la tabla *Cliente* sólo pueden ser M (masculino), F (femenino) u O (para indicar otro).
- Las películas tienen tres precios diferentes según el siguiente código: E, estreno, 5 €/visualización, N, no es un estreno, 3,5 €/visualización, y C, clásica, a 2 €/ visualización. Pon la restricción para que el código precio sólo pueda tomar uno de esos 3 valores.
- Además, nuestra plataforma de streaming tiene sus películas clasificadas en varias categorías temáticas, cada una de las cuales con un identificador numérico asociado y su descripción. Los valores de categoría que se pueden insertar sólo pueden ser los siguientes: Acción, Ciencia ficción, Comedia, Documental, Drama, Fantasía, Melodrama, Musical, Romance, Suspense, Terror, Animación y Aventuras.
- La fechapago tiene que ser anterior a la ffindisponibilidad.
- El preciopagado nunca podrá ser un valor negativo.

Ejercicio 5. Añade ahora las restricciones de clave ajena necesarias. (Ayuda: **ALTER TABLE ... ADD CONSTRAINT ... FOREIGN KEY**).

Ejercicio 6: Una vez creadas las tablas con todas las restricciones necesarias, vamos a introducir información en ellas. (Ayuda: **INSERT INTO**).

- Introduce las categorías y los precios que se mencionaron en el Ejercicio 4.
- Introduce dos clientes: tú y tu compañero. Deja el campo *activo* vacío.
- Introduce 4 películas. Una de ellas ha de ser un documental y, por tanto, sin intérpretes.
- El último Cliente que se dio de alta en la plataforma usó hace tres días la promoción de bienvenida que permitía poder visualizar la primera película gratis y la segunda por tan solo 1 euro. La película que eligió como gratuita la empezó a ver el mismo día por lo que ya se le ha acabado el periodo de visualización, pero la otra película (que era la película más nueva que teníamos) ha comenzado a verla hoy, por lo que tiene hasta mañana como fecha de fin de disponibilidad.

Ejercicio 7. También es posible crear tablas utilizando la definición de otra tabla ya existente en la base de datos con la sentencia: **CREATE TABLE nombre\_tabla\_nueva AS SELECT \* FROM nombre\_tabla\_a\_copiar**;

Crea una nueva tabla de precios a partir de la antigua que se llame *Precio2025*.

- ¿Se copian también las restricciones? Si no, añade las restricciones oportunas.
- ¿Se copian los datos ya introducidos? Si es así, bórralos. (Ayuda: **DELETE FROM**).

**Nota:** Para que funcione el DELETE sin WHERE hay que desactivar el modo "Safe updates" en el menú Edit > Preferences > SQL Queries, salir de la conexión y volver a entrar.

Ejercicio 8. SQL también permite insertar tuplas en las distintas tablas creadas utilizando la información de otras tablas ya existentes en el sistema gracias a la sentencia: **INSERT INTO table2 SELECT \* FROM table1 WHERE condition;**

Utiliza dicha sentencia para copiar todos los precios, excepto el más barato, de la tabla *Precio* a la tabla *Precio2025*.

Ejercicio 9. Ahora queremos actualizar la siguiente información en la base de datos (Ayuda: **UPDATE**).

- En la tabla *Precio2025*, sube un 5% todos los precios.
- Introducir el comentario "*Todavía estás a tiempo...*" en la tabla *Visualiza* si hace más de 15 días que se pagó por una película y aún no se ha empezado a ver o si todavía pueden seguir disfrutando de la película porque la fecha de finalización es posterior a la fecha de hoy. Utiliza una única sentencia SQL y contesta a la siguiente pregunta: ¿cuántas tuplas han sido modificadas?
- Pasa a la situación de no activos a los clientes que no hayan realizado ningún pago por visualización desde el 15 de agosto de 2024. ¿Cuántas tuplas se han modificado?
- Introduce un nuevo precio, identificado con una O en "*codiprecio*", para poder marcar películas en oferta a 1 € y ponles a los documentales este nuevo código de precio.

Ejercicio 10. La forma de crear tablas vista en el ejercicio 7 también se utiliza para crear tablas con estadísticas estáticas (creadas en un momento concreto) de datos almacenados en tablas de una base de datos.

Crea una nueva tabla "*ResumenMayo25*" que muestre los datos de cada cliente junto con el número de visualizaciones por las que ha pagado, el número películas distintas que ha visualizado, el dinero que se ha gastado (usa el "*preciopagado*") y en cuántas visualizaciones se le han puesto alguna observación en "*comentario*".

Ejercicio 11. Queremos eliminar ciertas tuplas de una tabla. (Ayuda: **DELETE FROM**).

- Borra los Clientes inactivos. ¿Qué error aparece y por qué?
- Borra las visualizaciones realizadas de la película más antigua.

Ejercicio 12. Repasa los criterios para la creación de índices descritos en el libro de Elmasri y Navathe, apartado 16.1.1: "Factores que influyen en el diseño físicos de una base de datos", de la A a la E (páginas 528 y 529), donde "Estructuras de acceso" o "Rutas de acceso" es equivalente a "Índices".

[http://www.ingebook.com/ib/NPcd/IB\\_Escritorio\\_Visualizar?cod\\_primaria=1000193&libro=2886](http://www.ingebook.com/ib/NPcd/IB_Escritorio_Visualizar?cod_primaria=1000193&libro=2886)

En nuestra base de datos de la plataforma de streaming, explica qué consultas crees que serán las más frecuentes, qué tablas se actualizarán más, si hay unas consultas más "urgentes" que otras...

En base a eso, crea los índices que creas necesarios (Ayuda: **CREATE INDEX**).

Ejercicio 13. Para observar cómo afecta la creación de índices al rendimiento, vamos a introducir más datos en la tabla *Película* y a hacer una pequeña prueba:

- Ejecuta el script “*películas.sql*” que hay en el aula virtual para insertar más películas en la base de datos.
- Si has creado un índice para el campo “director” de la tabla “*Película*”, bórralo (Ayuda: **DROP INDEX**).
- Busca aquellas películas cuyo director sea 'Director 1'. ¿Cuánto tarda en ejecutarse la consulta?
- Crea ahora el índice para este campo.
- Vuelve a ejecutar la misma consulta. ¿Cuánto ha tardado ahora?

Ejercicio 14. Como se vio en teoría, SQL cuenta con dos mecanismos diferentes implicados en el mantenimiento de la seguridad: el sistema de gestión de vistas y el subsistema de autorización que permite asignar a los usuarios permisos de acceso a determinadas tablas y vistas de la base de datos. Una vista es una tabla virtual en la base de datos cuyos contenidos están definidos por una consulta. Para el usuario de la base de datos, la vista aparece igual que una tabla real pero, a diferencia de ésta, no existe en la base de datos como conjunto almacenado de valores.

Nuestra plataforma de streaming quiere que sus clientes puedan acceder a su base de datos para consultar únicamente la información de las películas.

- Define una vista llamada “*VistaPelículasClientes*” en la base de datos *Plataforma* que muestre los campos *titulo*, *director*, *interprete1*, *interprete2*, *anyo*, *pais*, *duracion* y la *categoria* a la que pertenece (con el nombre completo – descripción – de la categoría, no con el *idcat*). (Ayuda: **CREATE VIEW**).

Comprueba que se ha creado correctamente en el menú de la izquierda (SCHEMAS > Plataforma > Views).

- Crea un nuevo usuario “cotilla” que únicamente tenga permiso de lectura sobre la vista “*VistaPelículasClientes*”. No podrá ver ni modificar nada más de la base de datos. Ayuda:
  - **CREATE USER '[nombre usuario]@'localhost' IDENTIFIED BY '[contraseña]';**
  - **GRANT [permiso] ON [nombre bases de datos].[nombre tabla] TO '[nombre usuario]@'localhost';**

Verificar que el usuario se ha creado correctamente:

- En el menú de la izquierda, accede a MANAGEMENT > Users and Privileges.

Ahí debe aparecer el nuevo usuario “cotilla”.

- Crea una nueva conexión con el “Username” y el “Password” del usuario “cotilla”.

Accede y comprueba que su base de datos “*Plataforma*” no tiene tablas pero sí la vista “*VistaPelículasClientes*”.

Intenta insertar una nueva tupla en ella. ¿Puedes?

Enséñale esta parte al profesor durante la práctica.