## Práctica realizada por:

U5P03: Creación y edición de tablas

**Cambios en esta versión: el apartado 16 se ha movido al 29.**

**Esta práctica no se entrega, no es evaluable pero es fundamental que la hagas con cuidado.**

**Se aconseja poner igualmente capturas para usarla luego para estudiar.**

# Primera parte: creación de tablas y edición de campos

Vamos a crear una base de datos sencilla con videojuegos y sus desarrolladores.

1- Crea mediante SQL una base de datos llamada “videojuegos”.

2- Indica a MySQL que a partir de ahora trabajaremos sobre esa base de datos. ¿Recuerdas cómo hacerlo mediante SQL?

3- Utilizando una sentencia SQL, crea una tabla “Videojuego” con los siguientes campos:

* Id: será de tipo INT, y será autoincrementado (será la clave primaria)
* Título: será de tipo texto (obligatorio y único)
* Nacionalidad: será de tipo texto
* Precio: será de tipo decimal, admitiendo hasta cinco dígitos de los que dos serán decimales
* Desarrollador: será de tipo texto, obligatorio, por defecto "Desconocido"
* Plataformas: será de tipo conjunto, incluyendo posibles valores como PC, PS4, PS5, Switch, XBox, etc.
* Multijugador: será de tipo booleano (consulta el PDF para ver cómo almacenar un valor lógico) (obligatorio, por defecto falso)

Para la creación debes utilizar atributos y restricciones de campo, sin incluir de momento ninguna restricción de tabla.

4- Elimina el campo Nacionalidad

5- Añade una nueva columna llamada "Año", que irá justo después de "Título", y que será de tipo YEAR.

6- Mediante **una sola** sentencia SQL, añade dos videojuegos a la tabla, pero incluyendo en la sentencia únicamente los campos Título, Año, Precio, Plataformas y Multijugador. Antes de hacerlo lee las siguientes indicaciones:

* Plataformas: aprovechamos para practicar el tipo de datos SET: en uno de los videojuegos pondrás varias plataformas, y en el otro ninguna (conjunto vacío).
* Para indicar si es multijugador, usa las palabras true y false: observa que si estuvieras rellenando directamente en el asistente visual de MySQL Workbench, tendrías que poner los valores que se almacenan realmente (1 y 0).

Cuando lo hayas hecho…

* Observa cómo al ser "Id" autoincrementado, MySQL se encarga de darle un valor.
* Comprueba que se usan los valores por defecto para Desarrollador y Multijugador

7- Mediante una sentencia SQL modifica el "Id" del segundo juego para que sea 35.

**Pista**: cuando no sepas hacer algo, intenta hacer el cambio con el entorno gráfico de MySQL Workbench sin llegar a aplicar los cambios: podrás copiar el código SQL para ayudarte. Si lo haces ten en cuenta que es un código generado de forma automática y suele ser más largo de lo necesario al incluir el nombre de la base de datos, abusar de comillas, etc. Las comillas en nombres de campos sólo son necesarias cuando hay espacios en blanco, y nosotros nunca los ponemos.

8- Introduce ahora un tercer videojuego, sin incluir su "Id". ¿Qué "Id" le pone automáticamente MySQL?

9- Intenta ejecutar dos o tres veces más la misma setencia en que añadías este último juego. Observa que fallará, debido al atributo UNIQUE del título (no puede haber dos iguales). Incluye captura del error para reconocerlo si te vuelve a aparecer.

10- Ahora introduce un cuarto juego, esta vez con título diferente. ¿Qué "Id" le ha puesto MySQL? Observa que los intentos de inserción, incluso fallidos, hacen que avance el contador de "Id". Por eso nunca es buena idea poner un tipo de datos muy pequeño (como TINYINT) en un campo autoincrementado.

*La idea del campo autoincrementado es que te puedas olvidar de los "Id" al insertar videojuegos, sin tener que pararte a pensar "¿cuál fue el último Id que utilicé?"*

11- Añade una restricción a la tabla (CHECK), llamada "chk\_positivo", que garantice que el precio será siempre positivo (ver página 45 de apuntes)

12- Intenta añadir un videojuego con precio 0 y observa el mensaje de error: debe aparecer el nombre de tu restricción, que ha sido violada.

13- Intenta añadir una nueva restricción que exija que el precio sea mayor que 50: observa que **no** te dejará hacerlo, ya que los datos existentes ya violan esa condición.

14- Elimina ahora la restricción "chk\_positivo" (ver página 46, recuerda ignorar el aviso de error).

*Nota importante: CHECK tiene soporte en MySQL desde la versión 8.0.16. Actualmente MySQL Workbench no permite visualizar estas restricciones y por tanto no puedes modificarlas o eliminarlas mediante el entorno gráfico.*

Entonces, ¿qué pasaría si añadiésemos una restricción CHECK sin nombre? Por ejemplo añade esta restricción:

ALTER TABLE videojuego ADD CHECK (Precio>0);

En este caso si queremos borrarla necesitaremos averiguar el nombre que le ha dado MySQL, y que MySQL Workbench no nos permite ver. Hay varias formas de ver este nombre, intenta probar las tres:

1. Intenta insertar un registro que viole la restricción: se te indicará el nombre de la restricción
2. Exporta la base de datos a un script: se verá en la sentencia CREATE TABLE el nombre que le ha dado
3. Utiliza la sentencia SQL "SHOW CREATE TABLE videojuego", que muestra la estructura de la tabla

15- Añade mediante SQL el campo generado o calculado PrecioConIVA, según los apuntes (página 22). Comprueba en tu tabla que ha aparecido el nuevo campo, y que su valor se calcula correctamente.

17-Mediante SQL vamos a añadir una restricción llamada “unico\_NombreAño” que impida que una pareja de campos Nombre y Año sea idéntica (ver página 45). Pon a prueba esta restricción insertando juegos que se llamen igual y tengan el mismo año.

* Curiosidad: a veces surge la duda de qué diferencia hay entre UNIQUE y PRIMARY KEY: la diferencia fundamental es que en este ejemplo tú puedes dejar el año en blanco (NULL), pero nunca podrías dejar una clave primaria en blanco.

### Trabajando con claves ajenas:

18- Crea una tabla “**desarrollador**” (si lo prefieres usa el entorno gráfico para hacerlo más rápido) en la que almacenaremos el nombre y la nacionalidad de un desarrollador de videojuegos. Como clave primaria utiliza otro código numérico autoincrementado.

19- Inserta un desarrollador de videojuegos en tu tabla

20- Intenta crear una restricción de clave ajena llamada “fk\_VideojuegoDesarrollador”, de forma que el campo “Desarrollador” de la tabla “videojuego” haga referencia al campo “Id” de la tabla “desarrollador”. ¿Entiendes el mensaje de error que obtienes? Los tipos de datos de ambos campos son incompatibles.

21- Modifica el campo “Desarrollador” de la tabla videojuego para que sea de tipo INT, y por tanto sea compatible con el tipo de datos del campo “Id” de la tabla “desarrollador”. ¿Entiendes el error que obtienes? Si ya tenías valores como “Ubisoft”, ahora no puedes decir que el campo es de tipo numérico.

**Importante:** La situación que acabas de experimentar es sólo un pequeño ejemplo de los trastornos que provoca modificar la estructura de una base de datos cuando ya está en marcha y tiene datos almacenados: de ahí la **gran importancia** de invertir en hacer un diseño cuidado.

Vamos a reparar la situación siguiendo los siguientes pasos:

a) En “desarrollador”: introduce en tu tabla valores para los desarrolladores de los juegos que tengas almacenados en tu tabla.

b) En “videojuego”: en el campo “Desarrollador” escribe el identificador del desarrollador, en lugar de su nombre (por ejemplo escribe “1” en lugar de “Ubisoft”. Esto se puede hacer porque en un campo de tipo texto puedes escribir números.

c) En “videojuego”: ahora modifica el tipo de datos del campo “Desarrollador” a INT: en este caso no habrá problemas ya que por ejemplo el texto “32” se convierte sin problemas al número 32.

d) En “videojuego”: por último ya puedes añadir la restricción de clave ajena sin problemas.

22- En la regla de creación de clave ajena no hemos especificado las reglas de comportamiento ON DELETE y ON UPDATE. Como no es posible modificar restricciones, tendremos que eliminar la restricción actual (gracias a su nombre) y crear una nueva. En este caso hazlo a través del entorno gráfico (pestaña “Foreign Keys”, desplegable ON UPDATE: CASCADE). Comprueba cómo MySQL borra la restricción y la crea de nuevo. Incluye aquí el código SQL como referencia.

23- Experimenta con la regla ON UPDATE: si modificas el código de un desarrollador, debería actualizarse automáticamente en los videojuegos de ese desarrollador.

24- Experimenta con la regla ON DELETE: como lo hemos dejado en “RESTRICT”, no deberías poder eliminar desarrolladores mientras tengas videojuegos suyos en la base de datos. Ten en cuenta que la opción SET NULL no se puede usar en este caso porque el campo “Desarrollador” en “videojuego” es obligatorio (nunca podría valer NULL).

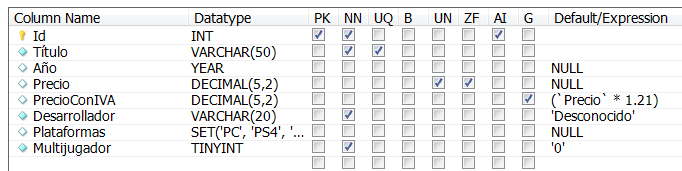
*Sugerencia: si quisiéramos igualmente almacenar videojuegos de desarrollador desconocido con este diseño de bases de datos, una solución sería incluir “Desconocido” en la tabla de desarrolladores, con su propio identificador, y sin nacionalidad asociada.*

### Trabajando con MySQL Workbench:

En este apartado vamos a comentar la forma de visualizar (y en ocasiones modificar) la información en MySQL Workbench.

25- Observa ahora el panel de edición de tu tabla “videojuego” en MySQL Workbench: fíjate en la columna Default/Expression:

* En general aquí se pone el valor por defecto del campo (“Madrid”, “Desconocido”, 0…).
* Cuando está marcada la casilla G, significa que es un campo generado. En este caso lo que aparece aquí es la fórmula de cálculo (podrías editarla desde aquí).



26- Ahora ve a la pestaña “Foreign keys”, como ya has hecho otras veces, y observa que allí está tu clave ajena: aquí podrías editar las reglas de comportamiento y también podrías eliminar la clave ajena o crear otras nuevas.

27- Observa que las restricciones CHECK no están por ninguna parte, al ser una funcionalidad nueva (no en SQL, pero sí en MySQL que no las ha implementado hasta hace muy poco).

28- Exporta tu base de datos a un script SQL y analiza la sentencia CREATE TABLE que incluye para tu tabla. Encontrarás unas líneas en las que se declara un índice o clave (KEY) para cada restricción UNIQUE y para cada clave ajena. En MySQL Workbench accede a la pestaña **“Indexes”** y encontrarás esas claves. ¿Qué es un **índice**? Se trata de una estructura que permite acceder rápidamente a los campos, optimizando la eficiencia del motor de bases de datos al realizar búsquedas. MySQL crea un índice por defecto para cualquier restricción de clave ajena o UNIQUE, porque son restricciones lentas de comprobar: imagina que hay millones de filas y tienes que comprobar si hay dos iguales, o si el código de desarrollador está presente en la otra tabla).

29-Al crear la tabla habíamos indicado que el campo Nombre es único. Era una restricción sin nombre y por tanto no podemos eliminarla con DROP UNIQUE Para que deje de serlo, accede a la pestaña “Indexes” y elimina “Título-UNIQUE”: observa que la sentencia SQL que se ejecuta es DROP INDEX.

30- Una vez has salvaguardado tu base de datos, prueba a eliminar alguna restricción mediante una sentencia SQL (por ejemplo borra la restricción de unicidad de las parejas nombre y año).