

Juego de rol medieval

Fabrizio Fontanarrosa

Coderhouse - SQL

Noviembre 2025

Segunda Entrega

Problemática

Buscamos representar un mundo de fantasía perteneciente a un juego de rol medieval. Tendremos que almacenar información acerca de personajes, lugares del mundo, especies y profesiones, entre otras entidades y eventos del mundo.

Listado de tablas

Personajes

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_personaje	INT	X			X
nombre	VARCHAR(50)				
apellido	VARCHAR(50)			X	
titulo	VARCHAR(50)			X	
id_raza	INT		X		
id_clase	INT		X		
id_region	INT		X		
ataque	INT				
defensa	INT				
salud	INT				
velocidad	INT				

Razas

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_raza	INT	X			X
nombre_raza	VARCHAR(50)				
resistencia	VARCHAR(50)			X	



Clases

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_clase	INT	X			X
nombre_clase	VARCHAR(50)				
rol	VARCHAR(50)			X	
habilidad	VARCHAR(50)				

Regiones

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_region	INT	X			X
nombre_region	VARCHAR(50)				
clima	VARCHAR(50)				

Mascotas

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_mascota	INT	X			X
id_animal	INT		X		
id_personaje	INT		X		X
nombre_mascota	VARCHAR(50)				



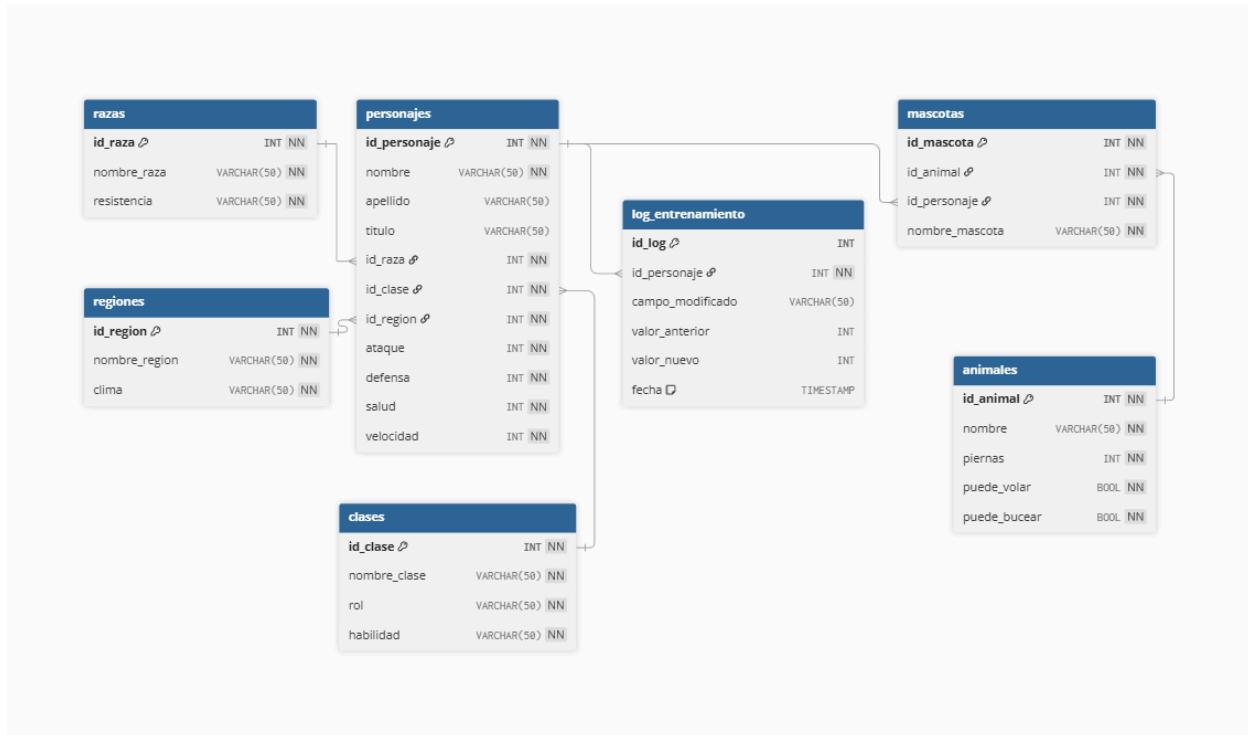
Animales

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_animal	INT	X			
nombre	VARCHAR(50)				
piernas	INT				
puede_volar	BOOL				
puede_bucear	BOOL				

log_entrenamiento

Campo	Tipo de dato	PK	FK	NULL	UNIQUE
id_log	INT	X			
id_personaje	INT		X		
campo_modificado	VARCHAR(50)				
valor_anterior	INT			X	
valor_nuevo	INT			X	
fecha	TIMESTAMP			X	

Diagrama de Entidad-Relación



Listado de vistas

detalle_personajes

Permite visualizar más fácilmente la información de los personajes, incluyendo su clase, región y raza. Muestra datos de las tablas *personajes*, *razas*, *regiones* y *clases*.

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_personaje	INT	personajes		X
nombre	VARCHAR(50)	personajes		
apellido	VARCHAR(50)	personajes	X	
nombre_raza	VARCHAR(50)	razas		
nombre_region	VARCHAR(50)	regiones		
titulo	VARCHAR(50)	personajes	X	
nombre_clase	VARCHAR(50)	clases		
rol	VARCHAR(50)	clases		

regiones_magos

Sirve para ver un listado de las regiones de donde provienen los personajes que son magos en la base de datos. Muestra datos de la tabla *regiones*, trabaja con registros de las tablas *personajes* y *clases*.

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_region	INT	regiones		X
nombre_region	VARCHAR(50)	regiones		

tiene_dragon

Permite ver una lista de nombres y apellidos de los personajes que tienen un dragón como mascota. Muestra campos de *personajes*, trabaja también con *mascotas* y *animales*.

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_personaje	INT	personajes		X
nombre	VARCHAR(50)	personajes		
apellido	VARCHAR(50)	personajes	X	

mascotas_voladoras

Sirve para ver un listado de las mascotas que pueden volar, y sus respectivos dueños. Muestra datos de las tablas *personajes* y *mascotas*, trabaja también con *animales*.

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_mascota	INT	mascotas		X
nombre_mascota	VARCHAR(50)	mascotas		
nombre (AS nombre_dueño)	VARCHAR(50)	personajes	X	
apellido (AS apellido_dueño)	VARCHAR(50)	personajes		

daños_con_montura

Permite ver una lista de los personajes cuyo rol es de Daño y tienen una mascota, trabaja con las tablas *personajes*, *mascotas* y *animales*

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_personaje	INT	personajes		X
nombre	VARCHAR(50)	personajes		
apellido	VARCHAR(50)	personajes	X	



nombre_clase	VARCHAR(50)	clases		
nombre_mascota	VARCHAR(50)	mascotas		
nombre (AS animal)	VARCHAR(50)	animales		

clasificacion_personajes

Muestra una lista de los personajes junto a su *puntuación* (promedio de las cuatro estadísticas), y ordena los registros en función de ese campo, de manera descendente. Trabaja con las tablas *personajes* y *clases*, y utiliza la función *promedio_estadisticas(p_id_personaje INT)*.

Campo	Tipo de dato	Tabla	NULL	UNIQUE
id_personaje	INT	personajes		X
nombre	VARCHAR(50)	personajes		
apellido	VARCHAR(50)	personajes	X	
nombre_clase	VARCHAR(50)	clases		
promedio_estadisticas(id_personaje) (AS puntuacion)	VARCHAR(50)	N/A (función)		

Listado de funciones

promedio_estadisticas(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y devuelve el promedio de sus cuatro estadísticas: Ataque, Defensa, Salud y Velocidad.

cantidad_clase(p_id_clase INT)

Recibe la primary key de un registro de la tabla *clases* y devuelve la cantidad de personajes que hay de esa clase.

Listado de stored procedures

entrenar_ataque(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística de ataque en 5.

entrenar_defensa(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística de defensa en 5.

entrenar_salud(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística de salud en 5.

entrenar_velocidad(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística de velocidad en 5.

entrenar_fortaleza(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística más alta en 5.

entrenar_debilidad(p_id_personaje INT)

Recibe la primary key de un registro de la tabla *personajes* y aumenta su estadística más baja en 5.

Listado de triggers

log_cambio_estadistica

Cada vez que se modifica una estadística de un personaje, se logea la modificación en la tabla *log_entrenamiento*.

estadistica_max

Cada vez que se modifica una estadística de un personaje, se verifica que esta no haya superado el número 100, puesto que es el máximo que se puede obtener por estadística.

Procedimiento de importación de datos

En la carpeta *data_examples* se encuentran seis archivos .csv enumerados. Para importar correctamente los datos, hay que copiar esos archivos al siguiente directorio:

C:\ProgramData\MySQL\MySQL Server 8.0\Uploads

Puede ser que la dirección varíe si tiene una versión distinta de MySQL Server, pero siempre termina en *Uploads*.

Una vez hecho esto, abrimos MySQL Workbench con normalidad. Luego de ejecutar los scripts *tablas.sql* y *script.sql*, ejecutamos *importacion_datos.sql*, y el estado de ejecución de las solicitudes debería mostrarse en verde en la parte inferior de la interfaz de Workbench, indicando que los datos se importaron correctamente.