



Campus Santa Fe

5/05/2023

Materia: Construcción de Software y Toma de Decisiones (Gpo 401)

4. Ejercicio de actualización de una base de datos relacional en MySQL y diagramas de UML.

Profesores:

Octavio Navarro Hinojosa

Gilberto Echeverría Furió

Esteban Castillo Juarez

Domingo Mora Perez | A01783317

Rafael Blanga Hanono | A01781442

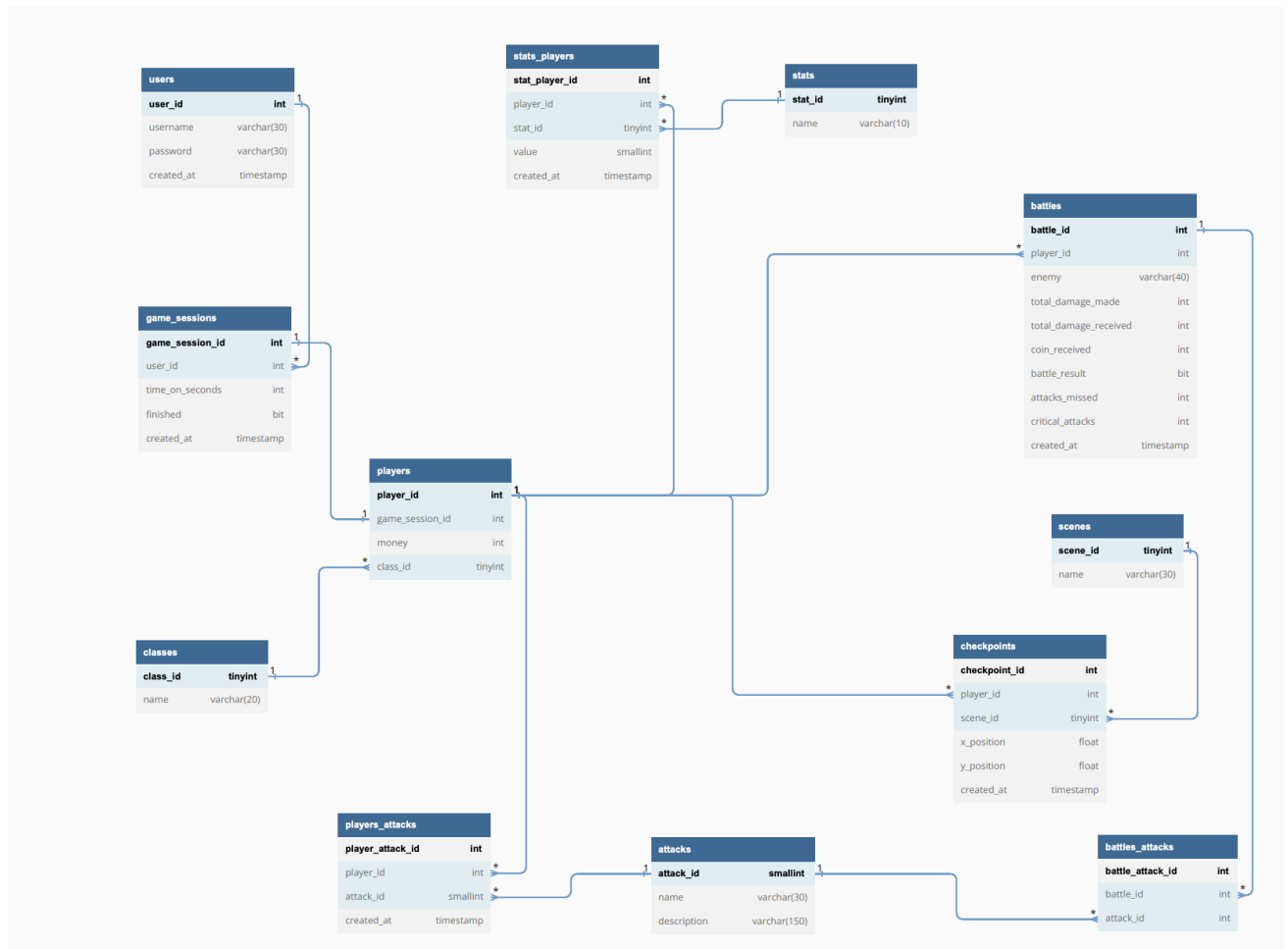
Ian Holender Mariaca

Francisco Martínez Gallardo Lascurain

Sergio Zuckermann A01024831

Santiago Tena

Diagrama Entidad Relación



Cardinalidad

- La tabla de usuarios tiene una cardinalidad de uno a muchos con la tabla game_sessions. Ya que un usuario puede tener muchos game sessions. (1) <--> (N)
- La tabla players tiene una cardinalidad de uno a uno con la tabla game_sessions, es decir, cada game_session tiene un jugador siempre. (1) <--> (1)
- La tabla players tiene una cardinalidad de muchos a uno con la tabla classes a través de la columna class_id. (N) <--> (1)
- La tabla checkpoints tiene una cardinalidad de uno a muchos con la tabla players, es decir, un jugador puede tener muchos checkpoints en los cuales respawnear. (1) <--> (N)
- La tabla checkpoints tiene una cardinalidad de muchos a uno con la tabla scenes, ya que varios checkpoints pueden estar en la misma escena. (N) <--> (1)
- La tabla stats y players tienen una relación de muchos a muchos en la cual un jugador puede tener muchos stats y muchos stats pueden ser usados por diversos jugadores. Es

por esta razón que para esta relación se utiliza una tabla intermedia (players_stats). (N) <--> (N)

- Las tablas players y attacks tiene una cardinalidad de muchos a muchos en el cual en una batalla se pueden usar muchos ataques y muchos ataques pueden ser usados varias batallas. Es decir que por esta razón que para esta relación se utiliza una tabla intermedia (players_attacks). (N) <--> (N)
- Las tablas battles y attacks tienen una cardinalidad de muchos a muchos es decir que en una batalla se pueden usar muchos ataques y muchos ataques pueden ser usados varias batallas. Es por esta razón que para esta relación se utiliza una tabla intermedia (battles_attacks). (N) <--> (N)
-

Restricciones de Integridad

- En la tabla "users", se definió la columna "username" como única para evitar duplicados en los nombres de usuario. No pueden ser nulos los valores de "username" y "password". También los valores no pueden ser nulos, por qué un usuario debe tener siempre un usuario y contraseña.
- En la tabla "game_sessions" se tiene una llave foránea a user_id.
- En la tabla "players", se ha definido la columna "game_session_id" a la tabla game_session y una llave foránea la columna "class_id" de la tabla classes.
- En la tabla "checkpoints", se ha definido la columna "player_id" como clave foránea a la tabla players, También tiene la columna scene_id que hace referencia a la tabla scenes.
- En la tabla "battles", se ha definido la columna "player_id" como clave foránea de la tabla players.
- En la tabla "battles_attacks", la columna battle_id hace referencia como llave foránea a la tabla battles. Al igual que la columna attack_id hace referencia a la tabla attacks.
- En la tabla "players_attacks", la columna player_id hace referencia como llave foránea a la tabla players. Al igual que la columna attack_id hace referencia a la tabla attacks.
- En la tabla "stats_players", la columna player_id hace referencia como llave foránea a la tabla players. Al igual que la columna stat_id hace referencia a la tabla stats.
- En la tabla "stats" la columna name tiene que ser única, ya que no pueden existir dos estadísticas con el mismo nombre.
- En la tabla "attacks" la columna name tiene que ser única, ya que no pueden existir dos estadísticas con el mismo nombre.
- En la tabla "scenes" la columna name tiene que ser única, ya que no pueden existir dos estadísticas con el mismo nombre.
- En la tabla "classes" la columna name tiene que ser única, ya que no pueden existir dos estadísticas con el mismo nombre.

Nota: En todas las tablas se agregó el id como autoincremental, esto para que todas tengan un registro único y sean lo más óptimos posibles. También todas las llaves foráneas tienen un update cascade para que se eliminen en caso de perder la llave primaria a la referenciaban.

Forma Normal

El diagrama se encuentra en 3era forma normal en la gran mayoría de tablas, ya que en todas las tablas no existen relaciones transitivas. Esto es porque todas las tablas están unidas utilizando llaves foráneas para poder alcanzar esto. La única tabla que no está en tercera forma normal es la tabla de batallas en la cual, el atributo enemigos debería estar en su propia tabla. En el caso de las tablas intermedias se les agregó ids para que los atributos de las tablas fueran dependientes de estos y estuvieran en 3era forma normal. Me gustaría agregar que se podría considerar que por los timestamps el diagrama no cumpliría con la primera forma normal. Aunque las fechas se pueden considerar como un atributo especial en el que esto no importa.