

◀ Break Into Valhalla ▶

Database Schema

ER Model and Justification

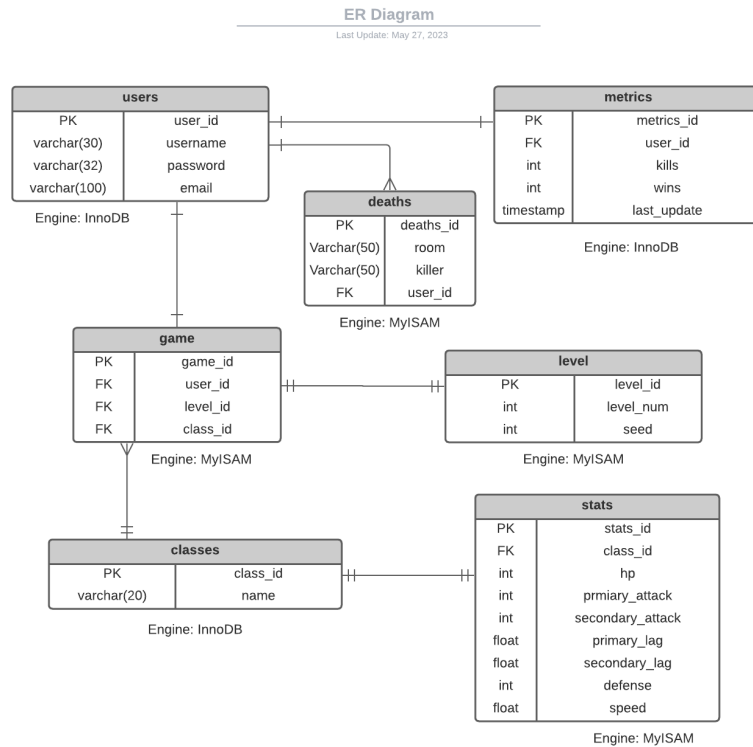
Version 1.0

May 27th, 2023

The Einherjar Team

Joaquín Badillo, Pablo Bolio, Shaul Zayat

Entity Relationship Diagram



Database Engines

There are 2 engines we can use in MySQL, InnoDB and MyISAM. The first one ensures data integrity, enables concurrency control and is ideal for managing the integrity of a schema where foreign keys may be present. MyISAM on the other hand has the main benefit of efficient reading speed. Therefore we decided to use MyISAM as the engine of the tables that are consulted and updated frequently or InnoDB otherwise.

Relationships

We used the following logic to determine the Relationships between Entities in the ER model:

- A user can have only one game and a game can only be related to one user, thus the relationship is 1:1.
- A user can have metrics and the metrics are related to a single user. This means that the relationship is 1:1.
- A user can have multiple deaths, the deaths belong to one user. Therefore users and deaths are related by a 1:N relationship.

- A game must exist if it has a level, a level must exist if it is related to a game. The relationship is 1:1.
- A game exists necessarily related to a class, a class can exist in multiple games. This relationship is therefore 1:N.
- A class must exist if it is related to stats, stats must exist if it belongs to a class. This means that classes and stats have a 1:1 relationship.

Normalization

For the schema to be in the 3rd normal form, we must verify that it is in the 1st normal form, then show that it is in the 2nd normal form and finally justify that there are no transitive functional dependencies.

1st Normal Form: To show that it is in the 1NF we can verify that all attributes are atomic (indivisible). This holds due to the fact that tables contain no tuples or composite attributes. For instance the player statistics and metrics have their own entity so that each one of them is an attribute.

2nd Normal Form: If the schema is in the 2NF then there are no partial functional dependencies. This holds due to the fact that entities (tables) were created so that all attributes depend on their primary keys. To illustrate this, the random seed that generated the dungeon belongs to a level and therefore the seed is an attribute of the level entity. Moreover this level belongs to a game and thus the level attribute is a foreign key in the game entity.

3rd Normal Form: As mentioned previously for the 3NF to hold the schema must not have no transitive functional dependencies. This means that there are no primary keys that depend on each other (which creates imposter tables). This holds due to the fact that the entities are independent. To reach this level of normalization we had to remove the number of deaths a player has from their metrics, due to the fact that each time they die there is a table that tracks the place of their death and the enemy that killed them.

Normalization was an important step to reduce redundancy in the schema and to improve the efficiency of the queries.