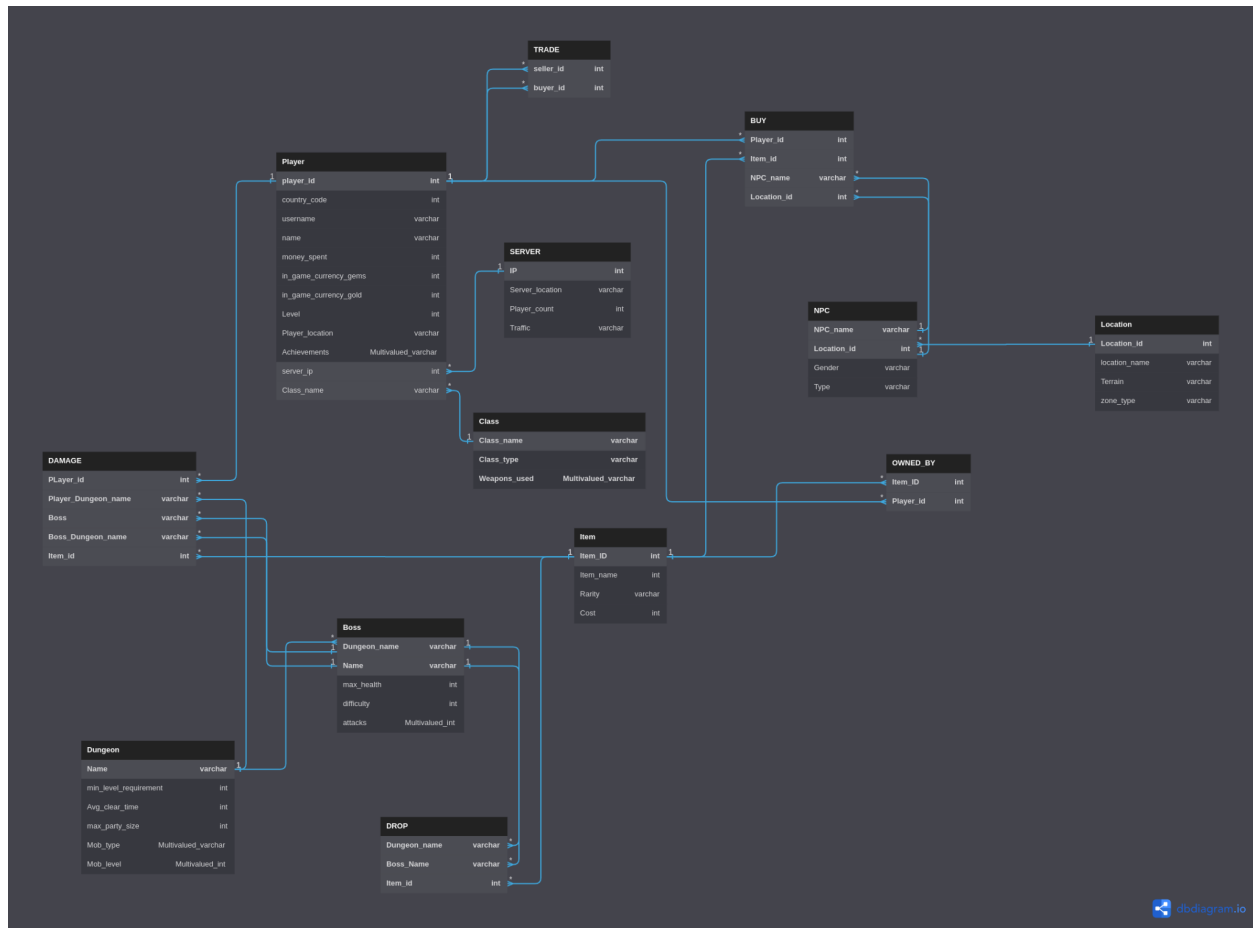


Relational Diagram without any Normalization :



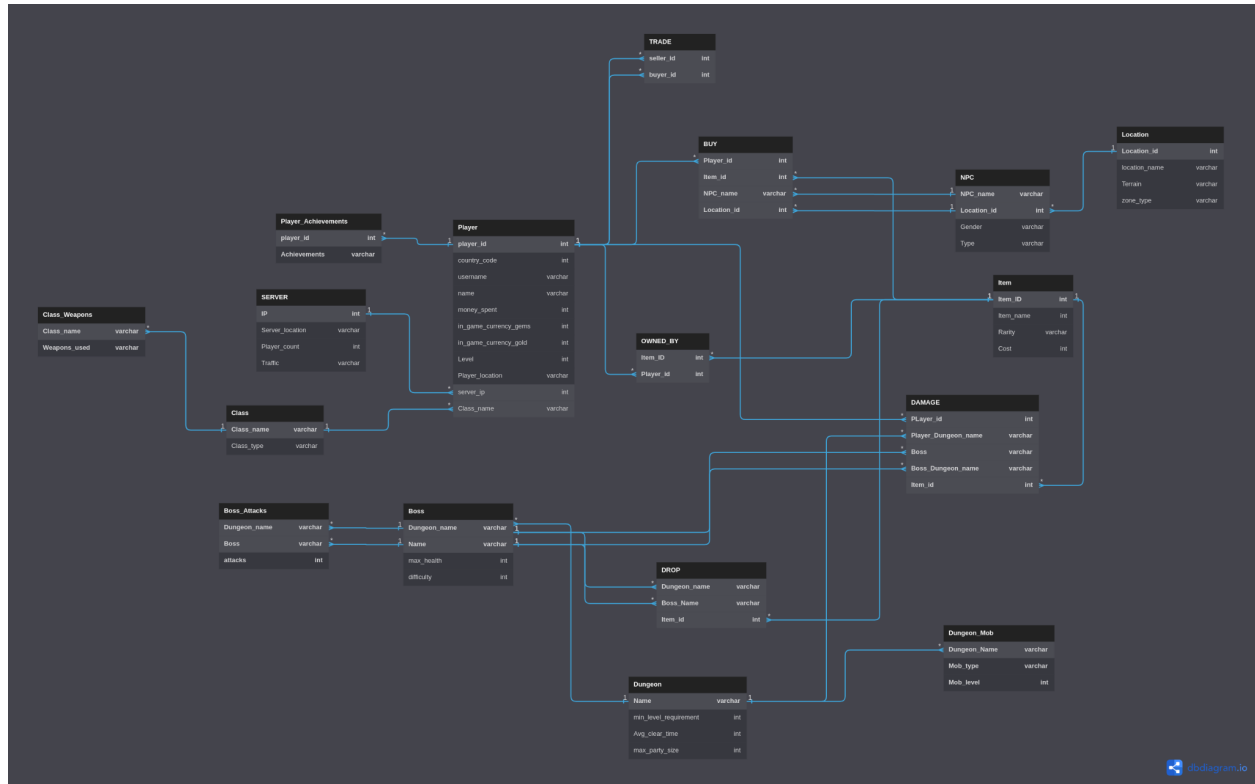
Link -

<https://dbdiagram.io/d/637092f5c9abfc61117236fd>

Explanation for ER to Relational-

1. For weak entities Boss and NPC, we have two attributes that are part of the primary key so any relationship that refers to it will have both of them as foreign keys. We put two individual arrows but essentially two attributes from relation refer to two attributes of Boss or NPC.
2. Composite attributes have been written individually like A(B, C) became A_B and A_C. This is because the relational diagram does not include composite attributes inherently and hence the tool doesn't provide a way to draw those either.
3. Multivalued attributes are represented by having "multivariable_" appended to its type.

Relational Diagram at 1NF :



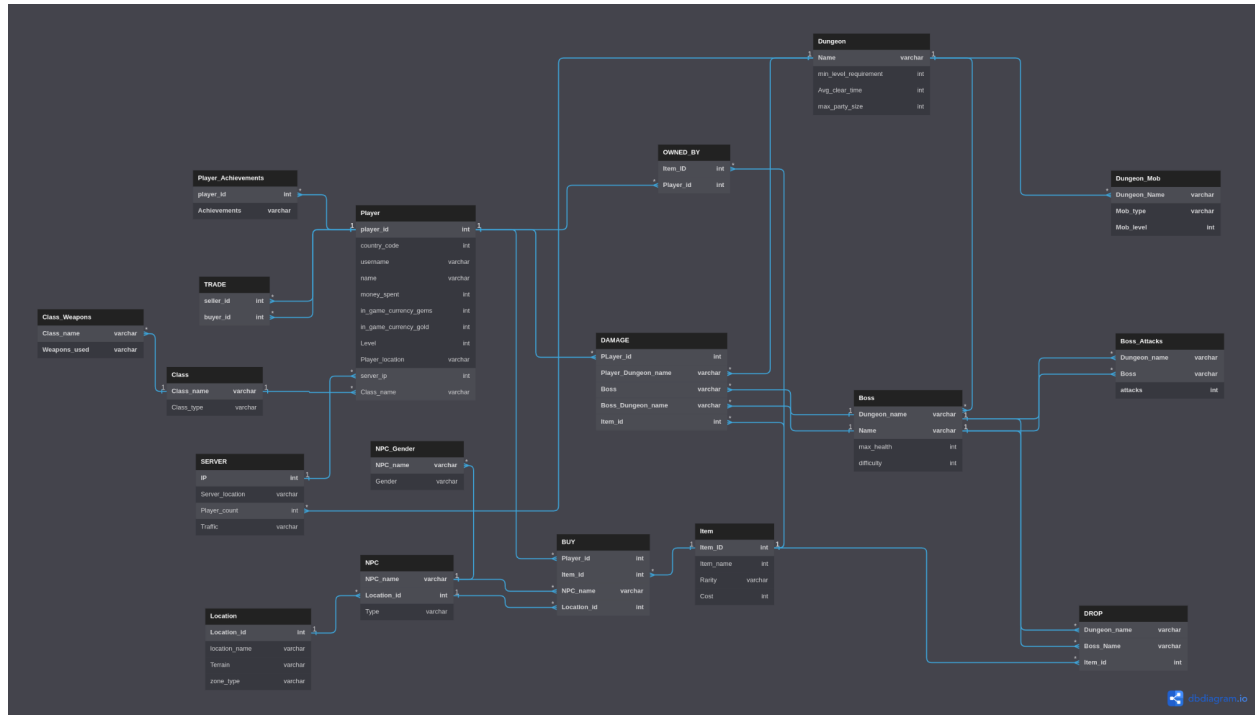
Link -

<https://dbdiagram.io/d/6370ae27c9abfc6117239c0>

Explanation for Relational to 1NF:

We made new tables for multivalued attributes: player_achievements, boss_attacks, class_weapons_used and dungeon_mobs as multivalued attributes are converted to tables when normalizing the relational model to 1NF.

Relational Diagram at 2NF and 3NF :



Link -

<https://dbdiagram.io/d/6370e347c9abfc6111723feb>

Explanation for 1NF to 2NF:

Relations with only one attribute as primary attribute satisfy the 2NF condition as we cannot remove any key.

Relations with all attributes as primary attributes satisfy the 2NF condition as there are no non-prime attributes.

The only relations left now are weak entity types Boss and NPC.

For Boss, none of their non prime attributes are functionally dependent on one prime attribute so they satisfy 2NF as well.

For NPC, the gender is functionally dependent on NPC_name, so we made a new table for it.

Explanation for 2NF to 3NF:

After examining the database thoroughly, we found that none of the relations have transitive functional dependency hence there was no change in going from 2NF to 3NF.