

Phase2 changes:

Inventory is now a subclass of Storage. Some descriptions are changed based on this adjustment.

Rename the relation between Account and Character is now Owns instead of Have.

Added Resilience attribute to armor.

Added separate tables for WeaponPerks and ArmorPerks to handle multi value attributes.

My program is an inventory management system for a looter shooter game.

The entities in this program are account, character, item, inventory, storage box. Item is a super class of Armor and Weapon. Armor is a super class for all the armor type. On the other hand, weapon is the super class of, primary, secondary, and heavy. These three entities are the parents of all the different weapons. Here only auto rifle, shotgun, and rocket launcher are shown for simplicity sake.

- Account contains the name of the account (Primary Key), the password and an email address.
- Character contains the character Name, which is the primary key.
- Storage only have a label as its key, however as a weak entity, it uses the Character Name from the character entity as a foreign key.
- The Item entity have an Item ID and Name as a composite key.

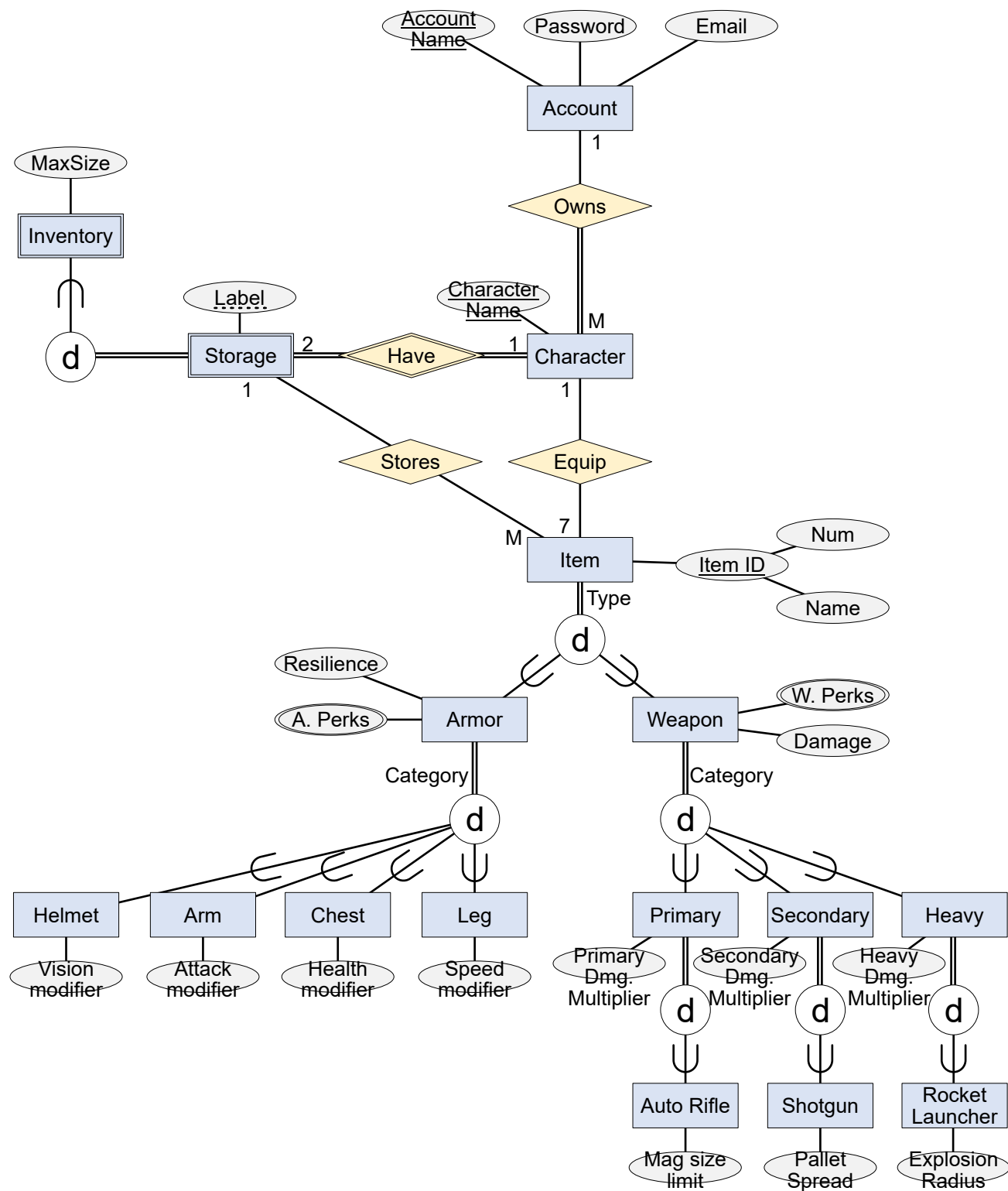
The relationships amongst the entities are, owns, equip, stores, and have.

- A character must have an account to exists, while the account can have multiple character.
- A character will always have a storage and an inventory. Both storage box and inventory can't exists without a character.
- Storage can store infinite amount of items.
- Inventory is a subclass of storage which can only store up to 63 items. To be specific, inventory can store 9 items from each specific item category (Helmet, arm, chest, leg, primary, secondary, heavy).
- Character can only equip one of each item category, 7 in total.

Primary key and Foreign key:

- For the account, Account name and email are both created by the user, and must be unique which will be checked for duplication during creation. For now Account Name is chose as the primary key.
- The character name is generated by the user and will also be checked for duplication. And this will be the primary key for Character.
- Since storage is reliant on the existence of a character entity, and are exclusive to a single character, we can identify it with the primary key of the character entity. Therefore introducing a foreign key to Storage.
- Item will have a composite key witch is combined by the name of the item which is unique if there is no duplicated loot. To make sure that we can distinguish duplicated loot a generated number, which is unique within the duplicated loot, is assigned to the item.

ER Diagram:



Relational Diagram:

