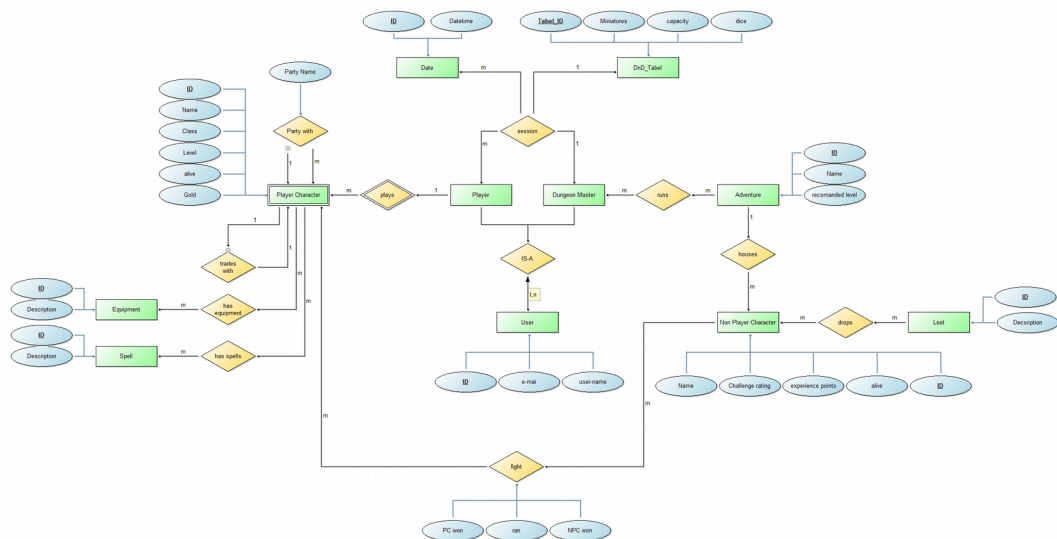


<Dungeon and Dragons Database - DDD>

Milestone 1:

In einem Geschäft stehen Usern mehrere Tische zur Verfügung auf denen sie DnD Sessions spielen können. Ein User ist eindeutig durch eine E-Mail und ein Passwort identifizierbar. Ein User kann entweder ein Player oder ein "Dungeon Master" (DM) oder beides sein. Ein Player spielt mehrere "Player Character" (PC) welche durch den Namen und dem Spieler identifiziert werden können. Ein DM hat mehrere Abenteuer in petto, welche aber auch von anderen DMs benutzt werden können. Ein Abenteuer hat einen eindeutigen Namen und ein empfohlenes PC level. Noch dazu beherbergt ein Abenteuer mehrere "Non Player Character" welche auch einen eindeutigen Namen, ein Challenge rating, Beute und Erfahrungspunkte haben. Eine Session findet auf einem Tisch statt zwischen mehreren Spielern und einem DM zu einem bestimmten Zeitpunkt. Auf einem Tisch sind bereits Miniaturfiguren und Würfel bereit gestellt. Des weiteren wird ein Tisch durch eine eindeutige Tischnummer gekennzeichnet und weist auch eine maximale Anzahl an Stühlen auf.



Milestone 2: Logical Design

DnD_Tabel(Table_ID, Miniatures, capacity, dice)
 PK = {Table_ID}

Player(player_id, e-mail, user-name)
 PK = {player_id}

DM(dungeon_master_id, e-mail, user-name)
PK = {dungeon_master_id}

Adventure(adventure_id, Name, recommend level)
PK = {adventure_id}

Non_Player_Character(non_player_character_id, Name, Challenge rating, experience points, alive, world)
PK = {non_player_character_id}
FK = {world}
Non_Player_Character.world <> Adventure.adventure_id

Party(Party_Name)
PK = {Party_Name}

Player_Character(player_character_id, Name, Character_of, Class, Level, Party_Name, alive, trade_partner, Gold, race)
PK = {player_character_id}
FK = {Character_of, Party_name, trade_partner}
Player_Character.Character_of <> Player.player_id
Player_Character.Party_Name <> Party_Name.Party_Name
Player_Character.trade_partner <> Player_Character.player_character_id

Equipment(equipment_id, Description)
PK = {equipment_id}

has_equipment(equipment, owned_by)
PK = {equipment, owned_by}
FK = {equipment, owned_by}
has_equipment.equipment <> loot.loot_id
has_equipment.owned_by <> Player_character.player_character_id

Spell(spell_id, Description)
PK = {spell_id}

has_spells(spell, casted_by)
PK = {spell, casted_by}
FK = {spell, casted_by}
has_spells.spell <> spell.spell_id
has_spells.casted_by <> Player_character.player_character_id

Loot(loot_id, Description)
PK = {loot_id}

Drop_table(Loot, drops_from)
PK = {Loot, drops_from}
FK = {Loot, drops_from}
Drops_table.loot <> Loot.loot_id

Drops_table.drops_from<>Non_Player_Character.non_player_character_id

Runs(DM, Adventure)

PK = {DM, Adventure}

FK = {DM, Adventure}

Runs.DM<>DM.dungeon_master_id

Runs.Adventure<>Adventure.adventure_id

Date(game_date_id, game_datetime)

PK = {game_date_id, game_datetime}

Session(Tabel_ID, Player, DM, Date)

PK = {Tabel, ID, Player, DM, Date, Time}

FK = {Tabel, ID, Player, DM, Date, Time}

Session.Tabel_ID<>Tabel.Tabel_ID

Session.Player<>Player.player_id

Session.DM<>DM.dungeon_master_id

Session.Date<>Date.game_date_id

Party(Group_Name, Member)

PK = {Member, Group_Name}

FK = {Member}

Party.Member<>Player_charater.Player_character_id

Fight(PC, NPC, PC_won)

PK = {PC, NPC}

FK = {PC, NPC}

Fight.PC<>Player_character.Player_character_id

Fight.NPC<>Non_Player_Character.Non_Player_character_id

Milestone 4: Implementation

Beschreibung vom Projekt: Es handelt sich um eine Datenbank für Dungeons and Dragon – ein Tabetop Pen and Paper Fantasie spiel welches mit Stift, Papier und Würfeln gespielt wird. Einsatz von kleinen Spielfiguren ist nicht unüblich. Die Idee: Falls man eine Veranstaltung abhalten wollen würde und eine Übersicht über alle Spieltische haben will.

Java

Python erstellt NUR FÜR TABLES MIT BESCHREIBENDEN ATTRIBUTEN einen String mit dem insert command, welche Java dann ausliefert. Für Tables mit Fremdschlüssel funktioniert die Datengenerierung nur über Java ohne csv Dateien, wie auch verlangt wurde.

PHP

Es gibt 5 Pages.

Index:

präsentiert meine Views.

richest_player_character → gibt mir für jeden Player den reichsten Character aus (Player haben Player_character/PC)

party_gold_summary → gibt mir das Gold jede Gruppe aus (PC's können in Gruppen sein)

end_game_party → gibt mir alle Gruppen aus welchen einen Durchschnittslevel von 15 haben.

SearchPlayerResults: Gelangt man über das Form in index. Hier wird eine suche nach Playern gemacht. Es werden alle Player ausgegeben welche den Inhalt des Forms haben. Hier können auch Updates und Deletes von Playern gemacht werden.

SearchDungeonMasterResults: Funktioniert wie searchPlayerResults.

Über die Navbar gelangt man zu den Inserts. Hier kann man Dungeonmaster einfügen und auch ein Adventure erstellen. Das gleiche passiert auch bei Player aber man kann leider „noch“ keine Characters für Players einfügen.

Referenzen:

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

https://www.w3schools.com/php/php_superglobals_server.asp