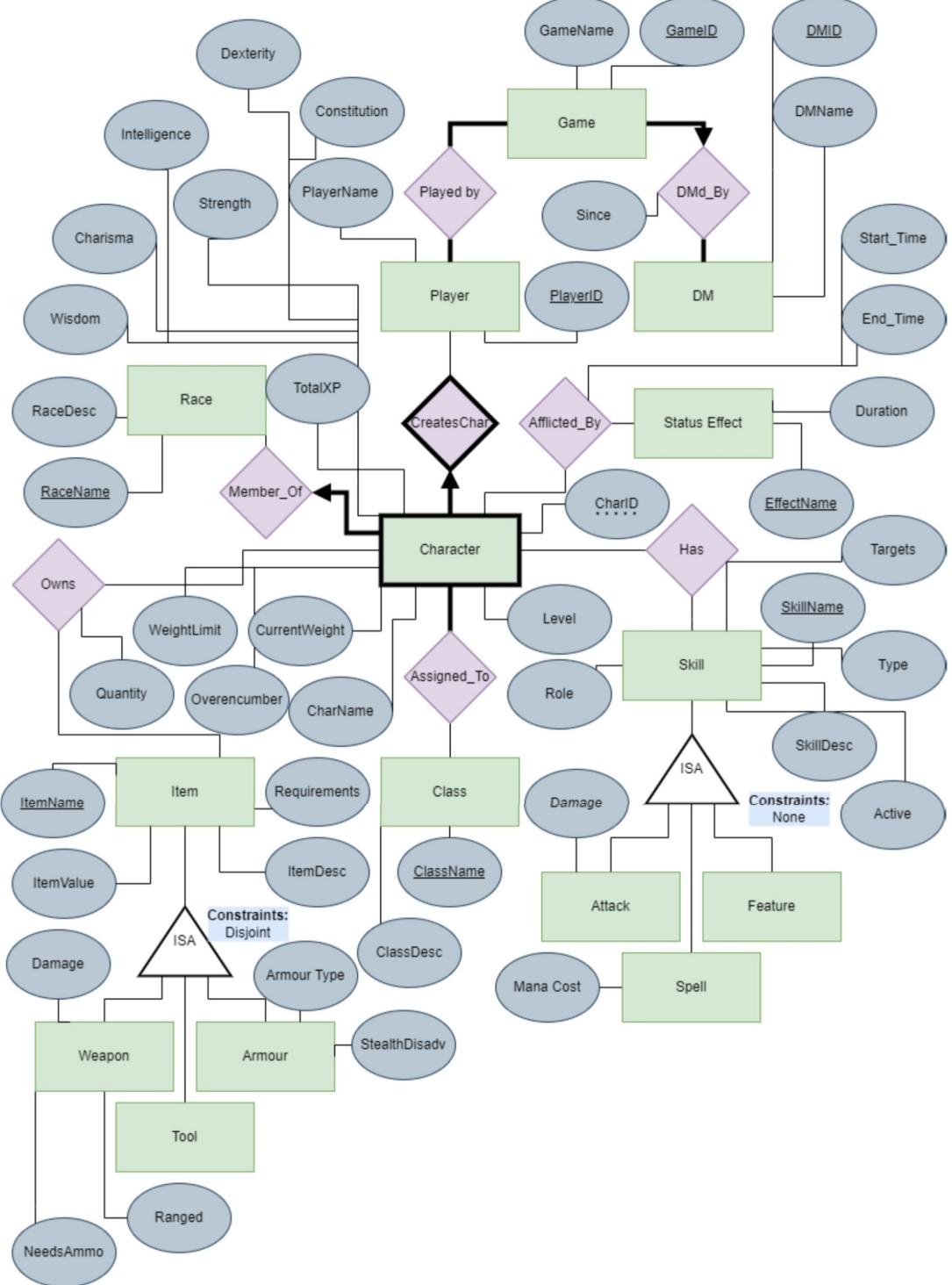


### **Project Summary (2-3 sentences)**

In the context of the 5th edition of Dungeons and Dragons, this database models a game with a DM, players, and player characters. It includes all player character skills, attributes, inventory, and other game-relevant aspects, such as status effects and character class and race. The database will allow players within a game to keep track of all of their characters and their individual inventories, abilities, and dynamic characteristics such as status-effects, and character level.

## ER Diagram



### **ER Diagram Changes**

#### *1. Changed PlayedBy from a one-to-many relationship to a many-to-many relationship*

A Game can indeed be played by more than one Player. The original diagram restricted each Game to only one Player (addresses “Would a Game not be Playedby more than one Player? Error there?”).

#### *2. Added total Character participation to AssignedTo*

A Character can be AssignedTo multiple Classes, but a Character must have at least one Class.

#### *3. Added dotted line to CharID attribute of Character*

This was done to show that Character is a weak entity identified by the Player set. The primary key is made up of PlayerID, the owner set primary key, and CharID, which is now underlined to explicitly show the relationship between Player and Character (addresses “If Character is a weak entity, CharID should be underlined with a dashed or dotted line”).

### **Unincorporated Feedback**

#### *A Character has one Race and potentially multiple Classes?*

Yes. As the fifth edition of DND allows characters to choose multiple classes (ie. “multiclassing”) both at creation and throughout the course of a campaign, we wanted to provide support for this. Similarly, a character only has one race that’s chosen at creation, even if it is a combination of two other races (ie. “half-elf” vs. “elf” and “human”).

### **Schema**

Player(PlayerID:integer, PlayerName: char(100)) PlayerName NOT NULL

Player\_Creates\_Char(CharID:integer, PlayerID:integer, CharName: char(100), Level: integer,  
Dexterity:integer, Constitution:integer, Intelligence: integer, Strength: integer, Charisma: integer,  
Wisdom: integer, WeightLimit: integer, CurrentWeight: integer, Overencumber: boolean, TotalXP:integer)  
CharName, Level NOT NULL, PlayerID ON DELETE CASCADE

Member\_Of(CharID: integer, RaceName: char(100))

Class(ClassName: char(100), ClassDesc: char(255))

Assigned\_To(ClassName: char(100), CharID:integer, PlayerID:integer)

Game\_DMd\_By(GameID: integer, GameName:char(100), Since: date, DMID:integer) GameName NOT  
NULL

DM(DMID: integer, DMName: char(100)) DMName NOT NULL

PlayedBy(PlayerID: integer, GameID: integer)

Race(RaceName: char(100), RaceDesc: char(255)) RaceDesc UNIQUE, RaceDesc CANDIDATE KEY

StatusEffect(EffectName: *char(100)*, Duration: *real*)

Afflicted\_By(PlayerID: *integer*, CharID: *integer*, EffectID: *integer*, Start\_Time: *date*, End\_Time: *date*)

Item(ItemName: *char(100)*, Requirements: *char(255)*, ItemDesc: *char(255)*, ItemValue: *real*) ItemDesc  
UNIQUE, ItemDesc CANDIDATE KEY

Weapon(ItemName: *char(100)*, Damage: *integer*, NeedsAmmo: *boolean*, Ranged: *boolean*)

Armour(ItemName: *char(100)*, ArmourType: *char(100)*, StealthDisadv: *boolean*)

Tool(ItemName: *char(100)*)

Owns(ItemName: *char(100)*, PlayerID: *integer*, CharID: *integer*, Quantity: *integer*)

Skill(SkillName: *char(100)*, SkillDesc: *char(255)*, Type: *char(100)*, Targets: *char(255)*, Role: *char(100)*,  
Active: *boolean*) Active NOT NULL, SkillDesc UNIQUE, SkillDesc CANDIDATE KEY

Attack(SkillName: *char(100)*, Damage: *integer*)

Feature(SkillName: *char(100)*)

Spell(SkillName: *char(100)*, ManaCost: *integer*)

Has(SkillName: *char(100)*, CharID: *integer*, PlayerID: *integer*)

## **Functional Dependencies**

**RELATION:** Player

PlayerID → PlayerName

**RELATION:** Player\_Creates\_Char

CharID, PlayerID → CharName, Level, Dexterity, Constitution, Intelligence, Strength, Charisma,  
Wisdom, WeightLimit, CurrentWeight, Overencumber, TotalXP

TotalXP → Level (NON-PK/CK)

WeightLimit, CurrentWeight → Overencumber (NON-PK/CK)

**RELATION:** Member\_Of

CharID → RaceName

**RELATION:** Class

ClassName → ClassDesc

**RELATION:** Assigned\_To

N/A

**RELATION:** Game\_DMd\_By

GameID → GameName, Since, DMID

**RELATION:** DM

DMID → DMName

**RELATION:** PlayedBy

N/A

**RELATION:** Race

RaceName → RaceDesc

RaceDesc → RaceName

**RELATION:** StatusEffect

EffectName → Duration

**RELATION:** Afflicted\_By

PlayerID, CharID, EffectID → Start\_Time, End\_Time

**RELATION:** Item

ItemName → Requirements, ItemDesc, ItemValue

ItemDesc → Requirements, ItemName, ItemValue

**RELATION:** Weapon

ItemName → Damage, NeedsAmmo, Ranged

NeedsAmmo → Ranged (NON-PK/CK)

**RELATION:** Armour

ItemName → ArmourType, StealthDisadv

ArmourType → StealthDisadv (NON-PK/CK)

**RELATION:** Tool

N/A

**RELATION:** Owns

ItemName, PlayerID, CharID → Quantity

**RELATION:** Skill

SkillName → SkillDesc, Type, Targets, Active, Role

SkillDesc → SkillName, Type, Targets, Active, Role

**RELATION:** Attack

SkillName → Damage

**RELATION:** Feature

N/A

**RELATION:** Spell

SkillName → ManaCost

**RELATION:** Has

N/A

### **Normalization**

**RELATION:** Player. Player already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Player(PlayerID:integer, PlayerName: char(100)) PlayerName NOT NULL

**RELATION:** Player\_Creates\_Char. FDs “TotalXP → Level” and “WeightLimit, CurrentWeight → Overencumber” not in BCNF.

**SPLIT ON** “TotalXP → Level”:

PCC'(CharID, PlayerID, CharName, Dexterity, Constitution, Intelligence, Strength, Charisma, Wisdom, WeightLimit, CurrentWeight, Overencumber, TotalXP),  
PCC1(TotalXP, Level)

**SPLIT ON** “WeightLimit, CurrentWeight → Overencumber”:

PCC''(CharID, PlayerID, CharName, Dexterity, Constitution, Intelligence, Strength, Charisma, Wisdom, WeightLimit, CurrentWeight, TotalXP),  
PCC1(TotalXP, Level), PCC2(WeightLimit, CurrentWeight, Overencumber)

**FINAL TABLE:** Player\_Creates\_Char”(CharID:integer, PlayerID:integer, CharName: char(100), Dexterity:integer, Constitution:integer, Intelligence: integer, Strength: integer, Charisma: integer, Wisdom: integer, WeightLimit: integer, CurrentWeight: integer, TotalXP:integer) CharName, PlayerID ON DELETE CASCADE

**FINAL TABLE:** Player\_Creates\_Char1(Level: integer, TotalXP:integer)

**FINAL TABLE:** Player\_Creates\_Char2(WeightLimit: integer, CurrentWeight: integer, Overencumber:boolean)

**RELATION:** Member\_Of. Member\_Of already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Member\_Of(CharID:integer, RaceName: char(100))

**RELATION:** Class. Class already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Class(ClassName: char(100), ClassDesc: char(255))

**RELATION:** Assigned\_To. Assigned\_To already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Assigned\_To(ClassName: char(100), CharID:integer, PlayerID:integer)

**RELATION:** Game\_DMd\_By. Game\_DMd\_By already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Game\_DMd\_By(GameID: integer, GameName:char(100), Since: date, DMID:integer)

GameName NOT NULL

**RELATION:** DM. DM already in BCNF. Normalization unnecessary.

**FINAL TABLE:** DM(DMID: integer, DMName: char(100)) DMName NOT NULL

**RELATION:** PlayedBy. PlayedBy already in BCNF. Normalization unnecessary.

**FINAL TABLE:** PlayedBy(PlayerID: integer, GameID: integer)

**RELATION:** Race. Race already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Race(RaceName: char(100), RaceDesc: char(255)) RaceDesc UNIQUE, RaceDesc CANDIDATE KEY

**RELATION:** StatusEffect. StatusEffect already in BCNF. Normalization unnecessary.

**FINAL TABLE:** StatusEffect(EffectName: char(100), Duration: real)

**RELATION:** Afflicted\_By. Afflicted\_By already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Afflicted\_By(PlayerID: integer, CharID: integer, EffectID: integer, Start\_Time: date, End\_Time: date)

**RELATION:** Item. Item already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Item(ItemName: char(100), Requirements: char(255), ItemDesc: char(255), ItemValue: real) ItemDesc UNIQUE, ItemDesc CANDIDATE KEY

**RELATION:** Weapon. FD “NeedsAmmo → Ranged” not in BCNF.

**SPLIT ON** “NeedsAmmo → Ranged”:

W'(ItemName: char(100), Damage: integer, NeedsAmmo: boolean), W2(NeedsAmmo: boolean, Ranged: boolean)

**FINAL TABLE:** Weapon'(ItemName: char(100), Damage: integer, NeedsAmmo: boolean)

**FINAL TABLE:** Weapon1(NeedsAmmo: boolean, Ranged: boolean)

**RELATION:** Armour. FD “ArmourType → StealthDisadv” not in BCNF.

**SPLIT ON** “ArmourType → StealthDisadv”:

A'(ItemName: *char(100)*, ArmourType: *char(100)*), A2(ArmourType: *char(100)*, StealthDisadv: *boolean*)

**FINAL TABLE:** Armour'(ItemName: *char(100)*, ArmourType: *char(100)*)

**FINAL TABLE:** Armour1(ArmourType: *char(100)*, StealthDisadv: *boolean*)

**RELATION:** Tool. Tool already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Tool(ItemName: *char(100)*)

**RELATION:** Owns. Owns already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Owns(ItemName: *char(100)*, PlayerID: *integer*, CharID: *integer*, Quantity: *integer*)

**RELATION:** Skill. Skill already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Skill(SkillName: *char(100)*, SkillDesc: *char(255)*, Type: *char(100)*, Targets: *char(255)*, Role: *char(100)*, Active: *boolean*) Active NOT NULL, Description UNIQUE, Description CANDIDATE KEY

**RELATION:** Attack. Attack already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Attack(SkillName: *char(100)*, Damage: *integer*)

**RELATION:** Feature. Feature already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Feature(SkillName: *char(100)*)

**RELATION:** Spell. Spell already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Spell(SkillName: *char(100)*, ManaCost: *integer*)

**RELATION:** Has. Has already in BCNF. Normalization unnecessary.

**FINAL TABLE:** Has(SkillName: *char(100)*, CharID: *integer*, PlayerID: *integer*)

## SQL DDL

```
CREATE TABLE DM(
    DMID      INTEGER,
    DMName    CHAR(100)   NOT NULL,
    PRIMARY KEY (DMID)
);
```

```
CREATE TABLE Game_DMd_By(
    GameID INTEGER,
    GameName CHAR(100)   NOT NULL,
    Since    DATE,
    DMID     INTEGER,
```

```

PRIMARY KEY (GameID),
FOREIGN KEY (DMID) REFERENCES DM(DMID)
);

CREATE TABLE Player(
PlayerID      INTEGER,
PlayerName    CHAR(100)      NOT NULL,
PRIMARY KEY (PlayerID)
);

CREATE TABLE PlayedBy(
PlayerID      INTEGER,
GameID        INTEGER,
PRIMARY KEY (PlayerID, GameID),
FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID),
FOREIGN KEY (GameID) REFERENCES Game_DM_By(GameID)
    ON DELETE CASCADE
);

CREATE TABLE Race(
RaceName      CHAR(100),
RaceDesc      CHAR(255),
PRIMARY KEY (RaceName),
UNIQUE (RaceDesc)
);

CREATE TABLE XPtoLevel(
TotalXP       INTEGER,
Level         INTEGER NOT NULL,
PRIMARY KEY (TotalXP)
);

CREATE TABLE isEncumbered(
WeightLimit      INTEGER,
CurrentWeight   INTEGER      DEFAULT 0,
Overencumber     BOOLEAN,
PRIMARY KEY (WeightLimit, CurrentWeight)
);

CREATE TABLE Player_Creates_Char (
CharID        INTEGER,
PlayerID      INTEGER,

```

```

CharName           CHAR(100)      NOT NULL,
Dexterity          INTEGER,
Constitution       INTEGER,
Intelligence       INTEGER,
Strength           INTEGER,
Charisma           INTEGER,
Wisdom             INTEGER,
WeightLimit        INTEGER,
CurrentWeight      INTEGER      DEFAULT 0,
TotalXP            INTEGER      DEFAULT 0,
PRIMARY KEY (CharID, PlayerID),
FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID)
    ON DELETE CASCADE,
FOREIGN KEY (TotalXP) REFERENCES XptoLevel(TotalXP),
FOREIGN KEY (WeightLimit, CurrentWeight) REFERENCES isEncumbered(WeightLimit, CurrentWeight)
);

CREATE TABLE Member_Of(
CharID      INTEGER,
RaceName    CHAR(100)      NOT NULL,
PRIMARY KEY (CharID),
FOREIGN KEY (CharID) REFERENCES Player_Creates_Char(CharID)
    ON DELETE CASCADE
    ON UPDATE NO ACTION,
FOREIGN KEY (RaceName) REFERENCES Race(RaceName)
    ON DELETE CASCADE
    ON UPDATE NO ACTION
);

CREATE TABLE Class(
ClassName   CHAR(100),
ClassDesc   CHAR(255),
PRIMARY KEY (ClassName),
UNIQUE (ClassDesc)
);

CREATE TABLE Assigned_To(
ClassName   CHAR(100),
CharID      INTEGER,
PlayerID    INTEGER,
PRIMARY KEY (ClassName, CharID, PlayerID),
FOREIGN KEY (ClassName) REFERENCES Class(ClassName)

```

```

ON UPDATE CASCADE,
FOREIGN KEY (PlayerID, CharID) REFERENCES Player_Creates_Char(PlayerID, CharID)
    ON DELETE CASCADE
);

CREATE TABLE StatusEffect(
EffectName      CHAR(100),
Duration        REAL DEFAULT 0,
PRIMARY KEY (EffectName)
);

CREATE TABLE Afflicted_By(
PlayerID        INTEGER,
CharID          INTEGER,
EffectName      CHAR(100),
Start_Time      DATE,
End_Time        DATE,
PRIMARY KEY (PlayerID, CharID, EffectName),
FOREIGN KEY (PlayerID, CharID) REFERENCES Player_Creates_Char(PlayerID, CharID)
    ON DELETE CASCADE,
FOREIGN KEY (EffectName) REFERENCES StatusEffect(EffectName)
    ON DELETE CASCADE
);

CREATE TABLE Item(
ItemName        CHAR(100),
Requirements    CHAR(255),
ItemDesc        CHAR(255),
ItemValue       REAL,
PRIMARY KEY (ItemName),
UNIQUE (ItemDesc)
);

CREATE TABLE IsWeaponRanged(
NeedsAmmo      BOOLEAN,
Ranged BOOLEAN,
PRIMARY KEY (NeedsAmmo)
);

CREATE TABLE Weapon(
ItemName        CHAR(100),
Damage         INTEGER,

```

```

NeedsAmmo    BOOLEAN,
PRIMARY KEY (ItemName),
FOREIGN KEY (ItemName) REFERENCES Item(ItemName)
ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (NeedsAmmo) REFERENCES isWeaponRanged(NeedsAmmo)
);

CREATE TABLE IsArmourForStealth(
ArmourType        CHAR(100),
StealthDisadv    BOOLEAN,
PRIMARY KEY (ArmourType)
);

CREATE TABLE Armour(
ItemName          CHAR(100),
ArmourType        CHAR(100),
PRIMARY KEY (ItemName),
FOREIGN KEY (ItemName) REFERENCES Item(ItemName)
ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (ArmourType) REFERENCES isArmourForStealth(ArmourType)
);

CREATE TABLE Tool(
ItemName          CHAR(100),
PRIMARY KEY (ItemName),
FOREIGN KEY (ItemName) REFERENCES Item(ItemName)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Owns(
ItemName          CHAR(100),
PlayerID         INTEGER,
CharID           INTEGER,
Quantity         INTEGER,
PRIMARY KEY (ItemName, PlayerID, CharID),
FOREIGN KEY (ItemName) REFERENCES Item(ItemName)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
FOREIGN KEY (PlayerID, CharID) REFERENCES Player_Creates_Char(PlayerID, CharID)
);

```

```
        ON DELETE CASCADE
);

CREATE TABLE Skill(
SkillName      CHAR(100),
SkillDesc      CHAR(255),
Type          CHAR(100),
Targets        CHAR(255),
Role           CHAR(100),
Active         BOOLEAN      NOT NULL,
PRIMARY KEY (SkillName),
UNIQUE (SkillDesc)
);

CREATE TABLE Attack(
SkillName      CHAR(100),
Damage         INTEGER,
PRIMARY KEY (SkillName),
FOREIGN KEY (SkillName) REFERENCES Skill(SkillName)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Feature(
SkillName      CHAR(100),
PRIMARY KEY (SkillName),
FOREIGN KEY (SkillName) REFERENCES Skill(SkillName)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Spell(
SkillName      CHAR(100),
ManaCost       INTEGER,
PRIMARY KEY (SkillName),
FOREIGN KEY (SkillName) REFERENCES Skill(SkillName)
    ON UPDATE CASCADE
    ON DELETE CASCADE
);

CREATE TABLE Has(
SkillName      CHAR(100),
```

```
PlayerID      INTEGER,  
CharID       INTEGER,  
Quantity      INTEGER,  
PRIMARY KEY (SkillName, PlayerID, CharID),  
FOREIGN KEY (SkillName) REFERENCES Skill(SkillName)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE,  
FOREIGN KEY (PlayerID, CharID) REFERENCES Player_Creates_Char(PlayerID, CharID)  
    ON DELETE CASCADE  
);
```

### **INSERT Statements**

```
INSERT INTO DM  
(DMID, DMName)  
VALUES  
(10000, 'Jane Doe'),  
(10001, 'Marc Garneau'),  
(10002, 'Gord Downie'),  
(10003, 'Hidetaka Miyazaki'),  
(10004, 'Yuji Horii');
```

```
INSERT INTO Game_DMd_By  
(GameID, GameName, Since, DMID)  
VALUES  
(10000, 'The Five Friends', '2023-02-28', 10003),  
(10001, 'Gun City', '2019-06-10', 10004),  
(10002, 'Trebuchet Village', '2021-12-25', 10003),  
(10003, 'The Crazy Wizard', '2015-11-02', 10000),  
(10004, 'AwesomeLand', '1999-02-22', 10002);
```

```
INSERT INTO Player  
(PlayerID, PlayerName)  
VALUES  
(100000, 'Mitch Hedberg'),  
(100001, 'Lü Bu'),  
(100002, 'Rachel Smith'),  
(100003, 'Cao Cao'),  
(100004, 'Guan Yu'),  
(100005, 'Rebecca Washington'),  
(100006, 'Cecil McKinley');
```

```
INSERT INTO PlayedBy
(PlayerID, GameID)
VALUES
(100006, 10000),
(100004, 10000),
(100003, 10000),
(100000, 10000),
(100001, 10000),
(100000, 10001),
(100003, 10001),
(100004, 10001),
(100006, 10002),
(100005, 10002),
(100004, 10002),
(100003, 10003),
(100002, 10003),
(100001, 10003),
(100000, 10004),
(100005, 10004);
```

```
INSERT INTO Race
(RaceName, RaceDesc)
VALUES
('Dragonborn', 'Humanoid dragons.'),
('Dwarf', 'Bold & hardy.'),
('Elf', 'Magical graceful people.'),
('Gnome', 'Tiny & energetic.'),
('Half-Elf', 'Kind of an elf.'),
('Halfling', 'Little guys & gals.'),
('Half-Orc', 'Kind of an orc.'),
('Human', 'Homo sapiens.'),
('Tiefling', 'Devilish.');
```

```
INSERT INTO XPToLevel
(TotalXP, Level)
VALUES
(253665, 17),
(120, 1),
(40892, 8),
(115067, 12),
(125432, 13),
```

```
(6547, 5),  
(23010, 7),  
(67031, 10),  
(85441, 11),  
(2756, 4);
```

```
INSERT INTO isEncumbered  
(WeightLimit, CurrentWeight, Overencumber)  
VALUES  
(225, 104, False),  
(120, 157, True),  
(300, 178, False),  
(105, 44, False),  
(165, 39, False),  
(255, 17, False),  
(210, 92, False),  
(210, 119, False),  
(165, 210, True),  
(225, 201, False);
```

```
INSERT INTO Player_Creates_Char  
(CharID, PlayerID, CharName, Dexterity, Constitution, Intelligence, Strength, Charisma, Wisdom,  
WeightLimit, CurrentWeight, TotalXP)  
VALUES  
(100000, 100000, 'Bartibus Beetleby', 13, 11, 12, 7, 17, 9, 225, 104, 253665),  
(100001, 100000, 'The Smiling Giggler', 14, 12, 15, 9, 7, 13, 120, 157, 120),  
(100002, 100001, 'Crusher McSmashy', 10, 17, 6, 19, 7, 6, 300, 178, 40892),  
(100003, 100002, 'Kyra Darkblade', 16, 11, 15, 13, 9, 12, 105, 44, 115067),  
(100004, 100002, 'Ynnead Malekith', 19, 12, 16, 9, 11, 14, 165, 39, 125432),  
(100005, 100003, 'Zhao Ming', 15, 13, 14, 15, 9, 16, 255, 17, 6547),  
(100006, 100004, 'Sir Thaddeus von Totalerquatsch', 12, 14, 7, 14, 10, 9, 210, 92, 23010),  
(100007, 100005, 'Belac Sprinkles', 14, 7, 4, 5, 20, 8, 210, 119, 67031),  
(100008, 100006, 'Martha Aleguzzler', 9, 15, 17, 15, 8, 9, 165, 210, 85441),  
(100009, 100005, 'Patricia Petunia', 7, 18, 16, 15, 19, 4, 225, 201, 2756);
```

```
INSERT INTO Member_Of  
(CharID, RaceName)  
VALUES  
(100000, 'Halfling'),  
(100001, 'Human'),  
(100002, 'Half-Orc'),  
(100003, 'Tiefling'),
```

```
(100004, 'Elf'),  
(100005, 'Dragonborn'),  
(100006, 'Human'),  
(100007, 'Gnome'),  
(100008, 'Dwarf'),  
(100009, 'Halfling');
```

```
INSERT INTO Class  
(ClassName, ClassDesc)  
VALUES  
(‘Barbarian’, ‘A raging warrior.’),  
(‘Bard’, ‘An inspiring magician.’),  
(‘Cleric’, ‘A priestly champion.’),  
(‘Druid’, ‘A priest of nature.’),  
(‘Fighter’, ‘A master of martial combat.’),  
(‘Monk’, ‘A master of martial arts.’),  
(‘Paladin’, ‘A holy warrior.’),  
(‘Ranger’, ‘A warrior of the wilderness.’),  
(‘Rogue’, ‘A stealthy scoundrel.’),  
(‘Sorcerer’, ‘A gifted spellcaster.’),  
(‘Warlock’, ‘A magic wielder with a supernatural bargain.’),  
(‘Wizard’, ‘A scholarly magic user.’);
```

```
INSERT INTO Assigned_To  
(ClassName, CharID, PlayerID)  
VALUES  
(‘Bard’, 100000, 100000),  
(‘Bard’, 100001, 100000),  
(‘Barbarian’, 100002, 100001),  
(‘Warlock’, 100003, 100002),  
(‘Ranger’, 100004, 100002),  
(‘Paladin’, 100005, 100003),  
(‘Warrior’, 100006, 100004),  
(‘Sorcerer’, 100007, 100005),  
(‘Fighter’, 100008, 100006),  
(‘Monk’, 100009, 100005);
```

```
INSERT INTO StatusEffect  
(EffectName, Duration)  
VALUES  
(‘Blinded’, 200),
```

```
('Charmed', 50),  
('Deafened', 75),  
('Frightened', 90),  
('Invisible', 75);
```

```
INSERT INTO Afflicted_By  
(PlayerID, CharID, EffectName, Start_Time, End_Time)  
VALUES  
(100003, 100002, 'Invisible', '2023-02-26', '2023-02-28'),  
(100000, 100000, 'Blinded', '2023-02-14', '2023-02-15'),  
(100000, 100000, 'Frightened', '2023-02-14', '2023-02-15'),  
(100008, 100006, 'Frightened', '2023-01-14', '2023-02-25'),  
(100001, 100000, 'Deafened', '2023-01-29', '2023-02-06');
```

```
INSERT INTO Item  
(ItemName, Requirements, ItemDesc, ItemValue)  
VALUES  
('Battleaxe', null, 'A big axe for killing things.', 10),  
('Warhammer', 'Strength 12+', 'A big hammer that can bludgeon.', 15),  
('Longbow', 'Dexterity 8+', 'A bow that can shoot things.', 50),  
('Dull Dagger', null, 'A dull dagger that seems to once hold value...', 0.5),  
('Spear', 'Dexterity 3+', 'A basic spear.', 5),  
('Ring Mail', 'Strength 10+', 'Leather armor with heavy rings sewn into it.', 30),  
('Leather', null, 'Armour made out of leather boiled in oil.', 10),  
('Chain Mail', 'Strength 13+', 'Armour made of metal rings chained together', 40),  
('Studded Leather', null, 'Armour made out of leather with studs.', 15),  
('Basic Shield', null, 'A standard shield.', 10),  
('Gold Piece', null, 'A single piece of gold. Used to buy things.', 1),  
('Jewelers Tools', null, 'A jewelry artisans toolkit.', 25),  
('Disguise Kit', 'Dexterity 3+', 'A kits to disguise yourself with', 25),  
('Flute', null, 'A neatly crafted wooden flute.', 2),  
('Basic Alchemy Set', 'Intelligence 5+', 'Standard kit for alchemy', 50),  
('Painters Tools', null, 'A painters supplies.', 10);
```

```
INSERT INTO IsWeaponRanged
```

```
(NeedsAmmo, Ranged)
```

```
VALUES
```

```
(True, True),  
(False, False);
```

\*Only two possible tuples.

```
INSERT INTO Weapon
(ItemName, Damage, NeedsAmmo)
VALUES
('Battleaxe', 10, False),
('Warhammer', 14, False),
('Longbow', 8, True);
('Dull Dagger', 1, False),
('Spear', 5, False);
```

```
INSERT INTO IsArmourForStealth
(ArmourType, StealthDisadv)
VALUES
('Heavy Armour', True),
('Light Armour', False),
('Heavy Shield', True),
('Light Shield', False);
*only four possible tuples
```

```
INSERT INTO Armour
(ItemName, ArmourType)
VALUES
('Ring Mail', 'Heavy Armour'),
('Leather', 'Light Armour'),
('Chain Mail', 'Heavy Armour'),
('Studded Leather', 'Light Armour'),
('Basic Shield', 'Light Shield');
```

```
INSERT INTO Tool
(ItemName)
VALUES
('Jewelers Tools'),
('Disguise Kit'),
('Flute'),
('Basic Alchemy Set'),
('Painters Tools');
```

```
INSERT INTO Owns
(ItemName, PlayerID, CharID, Quantity)
('Warhammer', 100002, 100001, 1),
('Warhammer', 100005, 100003, 1),
('Gold Piece', 100000, 100000, 142),
('Gold Piece', 100001, 100000, 6),
```

```
('Gold Piece', 100002, 100001, 12),
('Gold Piece', 100003, 100002, 533),
('Gold Piece', 100004, 100002, 543),
('Gold Piece', 100005, 100003, 96),
('Gold Piece', 100006, 100004, 457),
('Gold Piece', 100007, 100005, 1363),
('Gold Piece', 100008, 100006, 86),
('Gold Piece', 100009, 100005, 2),
('Longbow', 100003, 100002, 1),
('Ring Mail', 100006, 100004, 1),
('Leather', 100005, 100003, 1),
('Jewelers Tools', 100001, 100000, 1);
```

```
INSERT INTO Skill
(SkillName, SkillDesc, Type, Targets, Role, Active)
VALUES
('Black Tentacles', 'Creepy black tentacles...', 'Spell', '90ft', True),
('Confusion', 'Makes people confused!!', 'Spell', '10ft Sphere', True),
('Heal', 'Heal chosen character', 'Spell', 'Single character', True),
('Light', 'Create a ball of light to illuminate your path', 'Spell', '10ft', True),
('Zone of Truth', 'Make people tell the truth within zone', 'Spell', '5ft Sphere', True),
('Slashing Attack', 'Slices someone with a sharp blade.', 'Attack', '5ft', True),
('Bludgeoning Attack', 'Hits someone with a dull, hard object.', 'Attack', '5ft', True),
('Kick', 'Kick them.', 'Attack', '3ft', True),
('Thrust', 'Lunge forward to pierce enemy', 'Attack', '15ft', True),
('Punch', 'Hit someone with fists.', 'Attack', '2ft', True),
('Disguised', 'This character is disguised.', 'Feature', null, False);
```

```
INSERT INTO Attack
(SkillName, Damage)
VALUES
('Slashing Attack', 8),
('Bludgeoning Attack', 5),
('Kick', 2),
('Thrust', 4),
('Punch', 1);
```

```
INSERT INTO Feature
(SkillName)
VALUES
('Disguised');
```

```
INSERT INTO Spell
(SkillName, ManaCost)
VALUES
('Black Tentacles', 15),
('Confusion', 10),
('Heal', 5),
('Light', 2),
('Zone of Truth', 10);
```

```
INSERT INTO Has
(SkillName, PlayerID, CharID, Quantity)
VALUES
('Black Tentacles', 100001, 100000, 1),
('Black Tentacles', 100003, 100002, 1),
('Confusion', 100005, 100003, 1),
('Disguised', 100001, 100000, 1),
('Bludgeoning Attack', 100008, 100006, 3);
```