University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone 2

Group 26 **Oct. 17, 2023**

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Max Baker	41636465	s0w2b	bakermaxim877@gmail.com
Adwait Kulkarni	76264498	z7y0r	adwait.kul.2018@gmail.com
Angus Chow	10099935	aurus (e7q7w)	angus.c.y.chow@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

Description:

The domain of the project, Dungeons and Databases, lies within tabletop gaming and gaming utility. In this domain, the database primarily focuses on organizing and tracking specific details within and between DND campaigns. It will model aspects such as character definitions and campaign progression, and maintain the accessibility of this information for online gameplay.

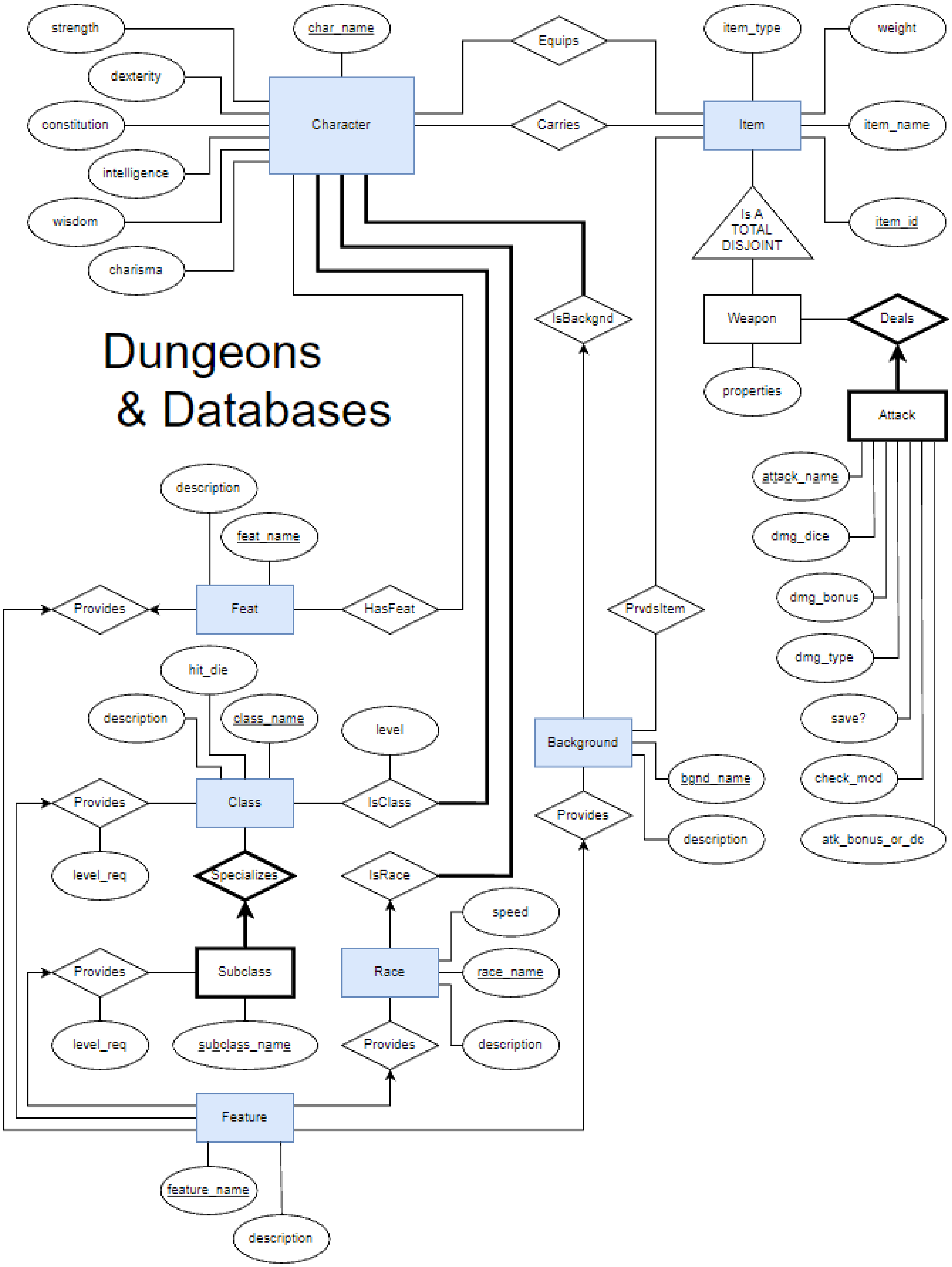
Functionality of the Database:

The database within Dungeons and Databases will offer a range of functionalities to enhance the tabletop gaming experience. Users can store complete character definitions including their race, class, background, and items, enabling them to maintain countless detailed character profiles. It also supports the updating of existing character definitions to reflect progression in a campaign. For practical use, the project will allow users to generate basic character sheets using a query to the database, making it easier to reference character information during gameplay. Another unique feature is the ability to generate characters randomly or based on predefined parameters, facilitating the creation of new characters for future campaigns.

Platform and Technology Stack:

Dungeons and Databases will be accessible through an online platform to ensure easy availability for long-distance gameplay. While specific technologies may vary, it could include web development tools like JSP or React for the front-end, and use back-end servlets and JDBC to handle database interactions and user authentication. The database itself will be built using a relational database management system (RDBMS) like MySQL or Oracle to efficiently store and retrieve data. Overall, the project aims to provide a seamless and user-friendly experience for tabletop gamers seeking to organize and enhance their unique character rosters.

ER Diagram:



We added description to Class for added detail and item_id to Item to make queries simpler

Red: Non-Foreign Key
Green: NOT NULL
Blue: Foreign Key
Underline: Primary Key

NOTE TO THE TA: The names of attributes in the relations have been shortened from the names of the original attributes in the ER diagram for better readability. (eg: Strength -> str, Dexterity -> dex, Constitution -> con etc.)

Character(char_name: VARCHAR(64),
str: TINYINT,
dex: TINYINT,
con: TINYINT,
int: TINYINT,
wis: TINYINT,
chr: TINYINT,
race_name: VARCHAR(64),
bgnd_name: VARCHAR(64))

EquippedInventory(char_name: VARCHAR(64),
item_name: VARCHAR(64))

CarriedInventory(char_name: VARCHAR(64),
item_name: VARCHAR(64))

CharacterClasses(char_name: VARCHAR(64),
class_name: VARCHAR(64),
subclass_name: VARCHAR(64))

Class(class_name: VARCHAR(64),
hit_die: VARCHAR(64),
class_description: TEXT)

Subclass(class_name: VARCHAR(64),
subclass_name: VARCHAR(64))

Race(race_name: VARCHAR(64),
speed: INT,
race_description: TEXT)

Background(bgnd_name: VARCHAR(64),
background_description: TEXT)

Feat(feat_name: VARCHAR(64),

```
          feat_description: TEXT,
          feature_name: VARCHAR(64))

Feature(            feature_name: VARCHAR(64),
                    feature_description: TEXT)
Item(     item_id: BIGINT,
          item_name: VARCHAR(64),
          item_type: ENUM(‘WEAPON’, ‘ETC’),
          item_weight: FLOAT,
          item_cost: FLOAT)

WeaponProperty(               item_id: VARCHAR(64),
                              weapon_property: VARCHAR(64))

WeaponAttack(                 item_id: VARCHAR(64),
                              attack_name: VARCHAR(64),
                              save?: BOOL,
                              check_ability: ENUM(‘STR’, ‘DEX’, ‘CON’, ‘INT’, ‘WIS’, ‘CHR’),
                              dmg_type: ENUM(     ‘PIERCING’,
                                                  ‘BLUDGEONING’,
                                                  ‘SLASHING’,
                                                  ‘FIRE’,
                                                  ‘COLD’,
                                                  ‘LIGHTNING’,
                                                  ‘THUNDER’,
                                                  ‘ACID’,
                                                  ‘POISON’,
                                                  ‘PSYCHIC’,
                                                  ‘RADIANT’,
                                                  ‘NECROTIC’),
                              dmg_bonus: INT,
                              dmg_dice: VARCHAR,
                              attack_bonus_or_dc: INT)

BackgroundItem(               bgnd_name: VARCHAR(64),
                              item_id: VARCHAR(64))

ClassFeature( class_name: VARCHAR(64),
              feature_name: VARCHAR(64),
              level_required: TINYINT)

SubclassFeature(              subclass_name: VARCHAR(64),
                              feature_name: VARCHAR(64),
                              level_required: TINYINT)
```

RaceFeature(race_name: VARCHAR(64),
feature_name: VARCHAR(64))

BackgroundFeature(bgnd_name: VARCHAR(64),
feature_name: VARCHAR(64))

Functional Dependencies:

NOTE TO THE TA: The names of attributes in the FDs have also been shortened from the names of the original attributes in the ER diagram for better readability. (eg: Strength -> str, Dexterity -> dex, Constitution -> con etc.)

Character:
char_name → str, dex, con, int, wis, chr, race_name, bgnd_name

EquippedInventory:

CarriedInventory:

CharacterClasses:
subclass_name → class_name

Item:
item_id → item_name, item_type, item_weight, item_cost

WeaponProperty:
item_id → weapon_property

WeaponAttack:
attack_name → item_id
item_id → save?, check_ability, dmg_type, dmg_bonus, dmg_dice, attack_bonus_or_dc

Class:
class_name → hit_die, class_description

Subclass:

Race:
race_name → speed, race_description

Feat:
feat_name → feat_description

Background:
bgnd_name → bgnd_description

BackgroundItem:
bgnd_name → item_id

Feature:
feature_name → feature_description

ClassFeature:
class_name, feature_name → level_req
feature_name → class_name, level_req

SubclassFeature:
subclass_name, feature_name → level_req
feature_name → subclass_name, level_req

RaceFeature:

BackgroundFeature:

Normalization of FDs:

Relation: Character(char_name: VARCHAR(64),
str: TINYINT,
dex: TINYINT,
con: TINYINT,
int: TINYINT,
wis: TINYINT,
chr: TINYINT,
race_name: VARCHAR(64),
bgnd_name: VARCHAR(64))

FD: char_name → str, dex, con, int, wis, chr, race_name, bgnd_name

Relation is in BCNF as well as in 3NF. We can check that by finding the keys using the table method.

Left	Middle	Right
char_name		str, dex, con, int, wis, chr, race_name, bgnd_name

char_name+ = {char_name, str, dex, con, int, wis, chr, race_name, bgnd_name}

char_name is a minimal key and it appears on the LHS, so the relation is in BCNF. Also, all attributes of the FD on the right are a part of the key, so the relation is in 3NF as well. So no normalization is required.

Relation: EquippedInventory(char_name: VARCHAR(64),
item_name: VARCHAR(64))

FD: char_name → item_id

Finding the minimal keys:

Left	Middle	Right
char_name		item_id

char_name+ = {char_name, item_id}

As we can see, char_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: CarriedInventory(char_name: VARCHAR(64),
item_name: VARCHAR(64))

FD: char_name → item_id

Finding the minimal keys:

Left	Middle	Right
char_name		item_id

char_name+ = {char_name, item_id}

As we can see, char_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: CharacterClasses(char_name: VARCHAR,
class_name: VARCHAR,
subclass_name: VARCHAR)

FD: char_name → class_name
subclass_name → class_name

Finding the minimal keys:

Left	Middle	Right
char_name, subclass_name		class_name

char_name, sublclass_name+ = {char_name, subclass_name, class_name}

Therefore, our minimal key is char_name, sublcass_name.

This relation is not in BCNF because the LHS is not a superkey for either of the FDs and hence it is not in 3NF.

Decomposing to BCNF:

FD char_name → class_name violates BCNF. So we decompose char_name → class_name on the relation:

Relation_1(char_name: VARCHAR,
subclass_name: VARCHAR)
Relation_2(char_name: VARCHAR,
class_name: VARCHAR)

Both are in BCNF.

Relation: Item(item_id: BIGINT,
item_name: VARCHAR(64),
item_type: ENUM('WEAPON', 'ETC'),
item_weight: FLOAT,
item_cost: FLOAT)

FD: item_id → item_name, item_type, item_weight, item_cost

Finding the minimal keys:

Left	Middle	Right
item_id		item_name, item_type, item_weight, item_cost

item_id+ = {item_id, item_name, item_type, item_weight, item_cost}

As we can see, item_id is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: WeaponProperty(item_id: VARCHAR(64),
weapon_property: VARCHAR(64))

FD: item_id → weapon_property

Finding the minimal keys:

Left	Middle	Right
item_id		weapon_property

item_id+ = {item_id, weapon_property}

As we can see, item_id is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: WeaponAttack(item_id: VARCHAR(64),
attack_name: VARCHAR(64),
save?: BOOL,
check_ability: ENUM(‘STR’, ‘DEX’, ‘CON’, ‘INT’, ‘WIS’, ‘CHR’),
dmg_type: ENUM(‘PIERCING’,
‘BLUDGEONING’,
‘SLASHING’,
‘FIRE’,
‘COLD’,
‘LIGHTNING’,
‘THUNDER’,
‘ACID’,
‘POISON’,
‘PSYCHIC’,
‘RADIANT’,
‘NECROTIC’),
dmg_bonus: INT,
dmg_dice: VARCHAR,
attack_bonus_or_dc: INT)

FD: attack_name → item_id
item_id → save?, check_ability, dmg_type, dmg_bonus, dmg_dice, attack_bonus_or_dc

Finding the minimal keys:

Left	Middle	Right
attack_name	item_id	save?, check_ability, dmg_type, dmg_bonus,

		dmg_dice, attack_bonus_or_dc

attack_name+ = {attack_name, item_id, save?, check_ability, dmg_type, dmg_bonus, dmg_dice, attack_bonus_or_dc}

As we can see, attack_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation:		Class(	class_name: VARCHAR(64),
			hit_die: VARCHAR,
			class_description: TEXT)

FD: class_name → hit_die, class_description

Finding the minimal keys:

Left	Middle	Right
class_name		hit_die, class_description

class_name+ = {class_name, hit_die, class_description}

As we can see, class_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation:		Race(	race_name: VARCHAR(64),
			speed: INT,
			race_description: TEXT)

FD:	race_name → speed, race_description

Finding the minimal keys:

Left	Middle	Right
race_name		speed, race_description

race_name+ = {race_name, speed, race_description}

As we can see, race_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: Feat(feat_name: VARCHAR(64),
feat_description: TEXT)

FD: feat_name → feat_description

Finding the minimal keys:

Left	Middle	Right
feat_name		feat_description

feat_name+ = {feat_name, feat_description}

As we can see, feat_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: Background(bgnd_name: VARCHAR(64),
background_description: TEXT)

FD: bgnd_name → bgnd_description

Finding the minimal keys:

Left	Middle	Right
bgnd_name		bgnd_description

bgnd_name+ = {bgnd_name, bgnd_description}

As we can see, bgnd_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: BackgroundItem(bgnd_name: VARCHAR(64),
item_id: VARCHAR(64))

FD: bgnd_name → item_id

Finding the minimal keys:

Left	Middle	Right
bgnd_name		item_id

bgnd_name+ = {bgnd_name, item_id}

As we can see, bgnd_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: Feature(feature_name: VARCHAR(64),
feature_description: TEXT)

FD: feature_name → feature_description

Finding the minimal keys:

Left	Middle	Right
feature_name		feature_description

feature_name+ = {feature_name, feature_description}

As we can see, feature_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: ClassFeature(class_name: VARCHAR(64),
feature_name: VARCHAR(64),
level_required: TINYINT)

FD: class_name, feature_name → level_req
feature_name → class_name, level_req

Finding the minimal keys:

Left	Middle	Right
feature_name	class_name	level_req

feature_name+ = {feature_name, class_name, level_req}

As we can see, feature_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

Relation: SubclassFeature(subclass_name: VARCHAR(64),
feature_name: VARCHAR(64),
level_required: TINYINT)

FD: subclass_name, feature_name → level_req
feature_name → subclass_name, level_req

Finding the minimal keys:

Left	Middle	Right
feature_name	subclass_name	level_req

feature_name+ = {feature_name, subclass_name, level_req}

As we can see, feature_name is a minimal key and a superkey, so the relation is in BCNF and hence it is in 3NF as well.

SQL DDL:

TA Note: the names of these columns have been shortened to the standard abbreviations of these stats in-game and int_stat in the case of int to eliminate syntax errors

```
CREATE TABLE Character (
   char_name VARCHAR(64) PRIMARY KEY,
   str TINYINT NOT NULL,
   dex TINYINT NOT NULL,
   con TINYINT NOT NULL,
   int_stat TINYINT NOT NULL,
   wis TINYINT NOT NULL,
   chr TINYINT NOT NULL,
   race_name VARCHAR(64) NOT NULL,
   bgnd_name VARCHAR(64) NOT NULL
);

INSERT INTO Character (char_name, str, dex, con, int_stat, wis, chr, race_name, bgnd_name)
VALUES
         ('Character1', 12, 14, 10, 8, 15, 16, 'Human', 'Noble'),
         ('Character2', 14, 10, 12, 16, 8, 10, 'Elf', 'Criminal'),
         ('Character3', 8, 16, 14, 10, 12, 14, 'Dwarf', 'Sage'),
         ('Character4', 10, 12, 14, 14, 16, 10, 'Half-Orc', 'Folk Hero'),
         ('Character5', 16, 8, 12, 12, 14, 10, 'Gnome', 'Acolyte');

CREATE TABLE Item (
   item_id BIGINT PRIMARY KEY,
   item_name VARCHAR(64) NOT NULL,
   item_type ENUM('WEAPON', 'ETC') NOT NULL,
   item_weight FLOAT,
   item_cost FLOAT
);

INSERT INTO Item (item_id, item_name, item_type, item_weight, item_cost)
VALUES
         (1, 'Sword', 'WEAPON', 5.0, 25.0),
         (2, 'Potion of Healing', 'ETC', 0.5, 50.0),
         (3, 'Bow', 'WEAPON', 3.0, 40.0),
         (4, 'Scroll of Fireball', 'ETC', 0.1, 100.0),
         (5, 'Axe', 'WEAPON', 4.0, 30.0);

CREATE TABLE EquippedInventory (
   char_name VARCHAR(64),
   item_name VARCHAR(64),
   PRIMARY KEY (char_name, item_name),
   FOREIGN KEY (char_name) REFERENCES Character(char_name),
   FOREIGN KEY (item_name) REFERENCES Item(item_name)
);

INSERT INTO EquippedInventory(char_name, item_name)
VALUES
        (‘Character1’, 'Sword'),
        (‘Character2’, Potion of Healing),
        (‘Character3’, Bow),
        (‘Character4’, Scroll of Fireball),
        (‘Character5’, Axe);


CREATE TABLE CarriedInventory (
   char_name VARCHAR(64),
   item_name VARCHAR(64),
   PRIMARY KEY (char_name, item_name),
   FOREIGN KEY (char_name) REFERENCES Character(char_name),
   FOREIGN KEY (item_name) REFERENCES Item(item_name)
);

INSERT INTO CarriedInventory(char_name, item_name)
VALUES
        (‘Character1’, 'Sword'),
        (‘Character2’, Potion of Healing),
        (‘Character3’, Bow),
        (‘Character4’, Scroll of Fireball),
        (‘Character5’, Axe);


CREATE TABLE CharacterClasses (
   char_name VARCHAR(64),
   class_name VARCHAR(64),
   subclass_name VARCHAR(64),
   PRIMARY KEY (char_name, subclass_name),
   FOREIGN KEY (char_name) REFERENCES Character(char_name),
   FOREIGN KEY (subclass_name) REFERENCES Subclass(subclass_name)
);

INSERT INTO CharacterClasses (char_name, class_name, subclass_name)
VALUES
        ('Character1', 'Warrior', 'Battle Master'),
        ('Character2', 'Ranger', 'Gloom Stalker Conclave'),
        ('Character3', 'Cleric', 'Life Domain'),
        ('Character4', 'Barbarian', 'Path of the Beast'),
        ('Character5', 'Wizard', 'School of Abjuration');
```

TA Note: changed description to class_description to differentiate descriptions in other tables

```
CREATE TABLE Class (
  class_name VARCHAR(64) PRIMARY KEY,
  hit_die VARCHAR(64) NOT NULL,
  class_description TEXT
);

INSERT INTO Class (class_name, hit_die, class_description)
VALUES
        ('Warrior', ‘d3’, ‘Description for Warrior class’),
        ('Ranger', ‘d4’, ‘Description for Ranger class’),
        ('Cleric', ‘d5’, ‘Description for Cleric class’),
        ('Barbarian', ‘d1’, ‘Description for Barbarian class’),
        ('Wizard', ‘d2’, ‘Description for Wizard class’);


CREATE TABLE Subclass (
  class_name VARCHAR(64),
  subclass_name VARCHAR(64),
  PRIMARY KEY (class_name, subclass_name),
  FOREIGN KEY (class_name) REFERENCES Class(class_name)
);

INSERT INTO SubClass (class_name, subclass_name)
VALUES
        ('Warrior', 'Battle Master'),
        ('Ranger', 'Gloom Stalker Conclave'),
        ('Cleric', 'Life Domain'),
        ('Barbarian', ‘Path of the Beast’),
        ('Wizard', ‘School of Abjuration);
```

TA Note: changed description to race_description to differentiate descriptions in other tables

```
CREATE TABLE Race (
  race_name VARCHAR(64) PRIMARY KEY,
  speed INT NOT NULL,
  race_description TEXT
);
```

```
INSERT INTO RACE (race_name, speed, race_description)
VALUES
         ('Human', 12, ‘Description for Human’),
         (‘Elf’, 10, ‘Description for Elf’),
         (‘Dwarf’, 8, ‘Description for Dwarf’),
         (‘Half-Orc’, 15, ‘Description for Half-Orc’),
         (‘Gnome’, 17, ‘Description for Gnome’);

CREATE TABLE Background (
   bgnd_name VARCHAR(64) PRIMARY KEY,
   background_description TEXT
);

INSERT INTO Background (bgnd_name, background_description)
VALUES
   ('Noble', 'Description for Noble background'),
   ('Criminal', 'Description for Criminal background'),
   ('Sage', 'Description for Sage background'),
   ('Folk Hero', 'Description for Folk Hero background'),
   ('Acolyte', 'Description for Acolyte background');

CREATE TABLE Feat (
   feat_name VARCHAR(64) PRIMARY KEY,
   feat_description TEXT NOT NULL
);

INSERT INTO Feat(feat_name , feat_description)
VALUES
   (‘Feat1’, 'Description for Feat1'),
   (‘Feat2’, 'Description for Feat2'),
   (‘Feat3’, 'Description for Feat3'),
   ('Feat4’, 'Description for Feat4'),
   (‘Feat5’, 'Description for Feat5');


CREATE TABLE Feature (
   feature_name VARCHAR(64) PRIMARY KEY,
   feature_description TEXT NOT NULL
);

INSERT INTO Feature (feature_name, feature_description)
VALUES
   ('Feature1', 'Description for Feature 1'),
   ('Feature2', 'Description for Feature 2'),
   ('Feature3', 'Description for Feature 3'),
   ('Feature4', 'Description for Feature 4'),
   ('Feature5', 'Description for Feature 5');
```

TA Note: changed properties to weapon_property to make it clearer

```
CREATE TABLE WeaponProperty (
    item_id BIGINT,
    weapon_property VARCHAR(64) NOT NULL,
    PRIMARY KEY (item_id, weapon_property),
    FOREIGN KEY (item_id) REFERENCES Item(item_id)
);

INSERT INTO WeaponProperty (item_id, weapon_property)
VALUES
    (1, 'Two-Handed'),
    (2, 'Versatile'),
    (3, 'Ranged'),
    (4, 'Magical'),
    (5, 'Finesse');

CREATE TABLE WeaponAttack (
    item_id BIGINT,
    attack_name VARCHAR(64),
    save BOOL NOT NULL,
    check_ability ENUM('STR', 'DEX', 'CON', 'INT', 'WIS', 'CHR') NOT NULL,
    dmg_type ENUM(
        'PIERCING',
        'BLUDGEONING',
        'SLASHING',
        'FIRE',
        'COLD',
        'LIGHTNING',
        'THUNDER',
        'ACID',
        'POISON',
        'PSYCHIC',
        'RADIANT',
        'NECROTIC'
    ) NOT NULL,
    dmg_bonus INT,
    dmg_dice VARCHAR(64) NOT NULL,
    attack_bonus_or_dc INT NOT NULL,
    PRIMARY KEY (item_id, attack_name),
    FOREIGN KEY (item_id) REFERENCES Item(item_id)
);

INSERT INTO WeaponAttack (item_id, attack_name, save, check_ability, dmg_type,
dmg_bonus, dmg_dice, attack_bonus_or_dc)
VALUES
    (1, 'Slash', 0, 'STR', 'SLASHING', 2, '1d8', 4),
    (2, 'Shoot', 0, 'DEX', 'PIERCING', 2, '1d10', 5),
    (3, 'Stab', 0, 'DEX', 'PIERCING', 3, '1d6', 3),
    (4, 'Heal', 0, 'WIS', 'RADIANT', 0, '2d4', 0),
    (5, 'Fireball', 1, 'INT', 'FIRE', 8, '3d6', 15);

CREATE TABLE BackgroundItem (
    bgnd_name VARCHAR(64),
    item_id BIGINT,
    PRIMARY KEY (bgnd_name, item_id),
    FOREIGN KEY (bgnd_name) REFERENCES Background(bgnd_name),
    FOREIGN KEY (item_id) REFERENCES Item(item_id)
);

INSERT INTO BackgroundItem (bgnd_name, item_id)
VALUES
    ('Noble', 1),
    ('Criminal', 2),
    ('Sage', 3),
    ('Folk Hero', 4),
    ('Acolyte', 5);

CREATE TABLE ClassFeature (
    class_name VARCHAR(64),
    feature_name VARCHAR(64),
    level_required TINYINT NOT NULL,
    PRIMARY KEY (class_name, feature_name),
    FOREIGN KEY (class_name) REFERENCES Class(class_name),
    FOREIGN KEY (feature_name) REFERENCES Feature(feature_name)
);

INSERT INTO ClassFeature (class_name, feature_name, level_required)
VALUES
    ('Warrior', 'Feature1', 1),
    ('Ranger', 'Feature2', 2),
    ('Cleric', 'Feature3', 3),
    ('Barbarian', 'Feature4', 4),
    ('Wizard', 'Feature5', 5);
```

```
CREATE TABLE SubclassFeature (
    subclass_name VARCHAR(64),
    feature_name VARCHAR(64),
    level_required TINYINT NOT NULL,
    PRIMARY KEY (subclass_name, feature_name),
    FOREIGN KEY (subclass_name) REFERENCES Subclass(subclass_name),
    FOREIGN KEY (feature_name) REFERENCES Feature(feature_name)
);

INSERT INTO SubclassFeature (subclass_name, feature_name, level_required)
VALUES
    ('Battle Master', 'SubclassFeature1', 1),
    ('Gloom Stalker Conclave', 'SubclassFeature2', 2),
    ('Life Domain', 'SubclassFeature3', 3),
    ('Path of the Beast', 'SubclassFeature4', 4),
    ('School of Abjuration', 'SubclassFeature5', 5);

CREATE TABLE RaceFeature (
    race_name VARCHAR(64),
    feature_name VARCHAR(64),
    PRIMARY KEY (race_name, feature_name),
    FOREIGN KEY (race_name) REFERENCES Race(race_name)
);

INSERT INTO RaceFeature (race_name, feature_name)
VALUES
    ('Human', 'RaceFeature1'),
    ('Elf', 'RaceFeature2'),
    ('Dwarf', 'RaceFeature3'),
    ('Half-Orc', 'RaceFeature4'),
    ('Gnome', 'RaceFeature5');

CREATE TABLE BackgroundFeature (
    bgnd_name VARCHAR(64),
    feature_name VARCHAR(64),
    PRIMARY KEY (bgnd_name, feature_name),
    FOREIGN KEY (bgnd_name) REFERENCES Background(bgnd_name),
    FOREIGN KEY (feature_name) REFERENCES Feature(feature_name)
);

INSERT INTO BackgroundFeature (bgnd_name, feature_name)
VALUES
    ('Noble', 'Position of Privilege'),
    ('Criminal', 'Criminal Contact'),
    ('Sage', 'Researcher'),
    ('Folk Hero', 'Rustic Hospitality'),
    ('Acolyte', 'Shelter of the Faithful');
```

Tables:

Character

char_name	str	dex	con	int_stat	wis	chr	race_name	bgnd_name
Character1	12	14	10	8	15	16	Human	Noble
Character2	14	10	12	16	8	10	Elf	Criminal
Character3	8	16	14	10	12	14	Dwarf	Sage
Character4	10	12	14	14	16	10	Half-Orc	Folk Hero
Character5	16	8	12	12	14	10	Gnome	Acolyte

Item

item_id	item_name	item_type	item_weight	item_cost
1	Sword	WEAPON	5.0	25.0
2	Potion of Healing	ETC	0.5	50.0
3	Bow	WEAPON	3.0	40.0
4	Scroll of Fireball	ETC	0.1	100.0
5	Axe	WEAPON	4.0	30.0

EquippedInventory

char_name	item_name
Character1	Sword
Character2	Potion of Healing

Character3	Bow
Character4	Scroll of Fireball
Character5	Axe

CarriedInventory

char_name	**item_name**
Character1	Sword
Character2	Potion of Healing
Character3	Bow
Character4	Scroll of Fireball
Character5	Axe

CharacterClasses

char_name	**class_name**	**subclass_name**
Character1	Warrior	Battle Master
Character2	Ranger	Gloom Stalker Conclave
Character3	Cleric	Life Domain
Character4	Barbarian	Path of the Beast
Character5	Wizard	School of Abjuration

Class

class_name	**hit_die**	**class_description**
Warrior	d3	Description for Warrior class
Ranger	d4	Description for Ranger class

Cleric	d5	Description for Cleric class
Barbarian	d1	Description for Barbarian class
Wizard	d2	Description for Wizard class

Subclass

class_name	**subclass_name**
Warrior	Battle Master
Ranger	Gloom Stalker Conclave
Cleric	Life Domain
Barbarian	Path of the Beast
Wizard	School of Abjuration

Race

race_name	**speed**	**race_description**
Human	12	Description for Human
Elf	10	Description for Elf
Dwarf	8	Description for Dwarf
Half-Orc	15	Description for Half-Orc
Gnome	17	Description for Gnome

Background

bgnd_name	**background_description**
Noble	Description for Noble background
Criminal	Description for Criminal background

Sage	Description for Sage background
Folk Hero	Description for Folk Hero background
Acolyte	Description for Acolyte background

Feat

feat_name	**feat_description**
Feat1	Description for Feat1
Feat2	Description for Feat2
Feat3	Description for Feat3
Feat4	Description for Feat4
Feat5	Description for Feat5

Feature

feature_name	**feature_description**
Feature1	Description for Feature 1
Feature2	Description for Feature 2
Feature3	Description for Feature 3
Feature4	Description for Feature 4
Feature5	Description for Feature 5

WeaponProperty

item_id	**weapon_property**
1	Two-Handed

2	Versatile
3	Ranged
4	Magical
5	Finesse

WeaponAttack

item_id	**attack_name**	**save**	**check_ability**	**dmg_type**	**dmg_bonus**	**dmg_dice**	**attack_bonus_or_dc**
1	Slash	0	STR	SLASHING	2	1d8	4
2	Shoot	0	DEX	PIERCING	2	1d10	5
3	Stab	0	DEX	PIERCING	3	1d6	3
4	Heal	0	WIS	RADIANT	0	2d4	0
5	Fireball	1	INT	FIRE	8	3d6	15

BackgroundItem

bgnd_name	**item_id**
Noble	1
Criminal	2
Sage	3
Folk Hero	4
Acolyte	5

ClassFeature

class_name	**feature_name**	**level_required**
Warrior	Feature1	1

Ranger	Feature2	2
Cleric	Feature3	3
Barbarian	Feature4	4
Wizard	Feature5	5

SubclassFeature

subclass_name	**feature_name**	**level_required**
Battle Master	SubclassFeature1	1
Gloom Stalker Conclave	SubclassFeature2	2
Life Domain	SubclassFeature3	3
Path of the Beast	SubclassFeature4	4
School of Abjuration	SubclassFeature5	5

RaceFeature

race_name	**feature_name**
Human	RaceFeature1
Elf	RaceFeature2
Dwarf	RaceFeature3
Half-Orc	RaceFeature4
Gnome	RaceFeature5

BackgroundFeature

bgnd_name	**feature_name**
Noble	Position of Privilege
Criminal	Criminal Contact

Sage	Researcher
Folk Hero	Rustic Hospitality
Acolyte	Shelter of the Faithful