Sebastian Sojka

Project Proposal

RPG Database – Masks

For the final project, I plan to create a database for characters for Masks, a table-top RPG. In the game, you are playing a young team of super heroes protecting the city and bigger. Think Young Justice or Teen Titans. Player will control a character with an unique playbook (class) with special abilities. When you want to do something either confront a villain, protect a teammate, or solve a puzzle, you roll based on the labels and conditions. Labels are how your character view themselves: Danger, Freak, Savior, Superior, Mundane. Labels can be -2 to 3 Conditions are negatives to your rolls until the character does something to resolve or prove them wrong. The conditions are: Afraid, Angry, Guilty, Hopeless, and Hopeless. All five give -2 to certain rolls. Each playbook have unique moment of truth (ultimate to overcome impossible odds), team moves, potential (exp), advancement, and abilities (such as super speed, strength, martial arts, fireball), influence over people and people who have influence over you, and unique moves. I will try to add the unique moves to the database but each playbook have their own rules on this so it may be too big.

Entities:

* Characters – Different Characters of the game with unique playbook. potential, labels, relationship
* Conditions – List of conditions, and what they do
* Campaign – the campaign you are playing in, name
* Playbook - Name of playbook, starting labels

Relationships:

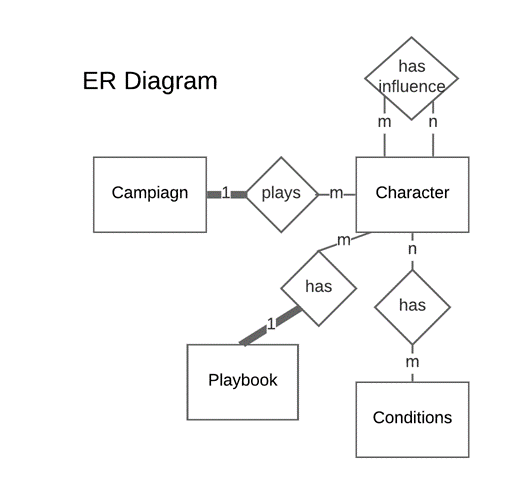
* Characters have conditions, many to many, characters can have multiple conditions and characters can have the same condition
* Characters have influence over other characters, many to many, characters can have influence over multiple characters and multiple characters can have influence over the same character.
* Character to playbook. 1 to many. Characters will have a single playbook while many can have the same playbook character must have a playbook
* Campaign to characters, 1 to many, a campaign may have many characters but a character may have only 1 campaign. character must be in a campaign

**Schema**

**A screenshot of a cell phone

Description generated with very high confidence**

**ERD**

****

**Initial Tables**

**The following sql tables creates and populates the playbook and condition tables.**

-- Table creation queries should immedatley follow the drop table queries, this is to facilitate testing on my end

DROP TABLE IF EXISTS `masksChar\_Con`;

DROP TABLE IF EXISTS `masksInfluence`;

DROP TABLE IF EXISTS `masksChar`;

DROP TABLE IF EXISTS `masksCon`;

DROP TABLE IF EXISTS `masksPlaybook`;

DROP TABLE IF EXISTS `masksCamp`;

-- Create a table for the playbook used by characters

-- PB\_id - an integer which is a foreign key reference to the playbook

-- name - a varchar with a maximum length of 255 characters, cannot be null, name of the plabook

-- Starting Labels - single int Character can have 5 labels to can range from -2 to +3 that effects the roll of the character. The labels are as following

-- -Danger

-- -Freak

-- -Savior

-- -Superior

-- -Mundane

CREATE TABLE `masksPlaybook` (

`PB\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

`Danger` int(1) NOT NULL DEFAULT '0',

`Freak` int(1) NOT NULL DEFAULT '0',

`Savior` int(1) NOT NULL DEFAULT '0',

`Superior` int(1) NOT NULL DEFAULT '0',

`Mundane` int(1) NOT NULL DEFAULT '0',

PRIMARY KEY (`PB\_ID`)

) ENGINE=InnoDB;

-- Create a table for the campaigns

-- PB\_id - an integer which is a foreign key reference to the campaign

-- name - a varchar with a maximum length of 255 characters, cannot be null, name of the campaign

CREATE TABLE `masksCamp` (

`Camp\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`Camp\_ID`)

) ENGINE=InnoDB;

-- Create a table for the characters of a game of Masks

-- Char\_ID - an auto incrementing integer which is the primary key

-- hero\_name - a varchar with a maximum length of 255 characters, cannot be null, the hero name of characte like Flash or Iron Man

-- real\_name - a varchar with a maximum length of 255 characters, the real name of the character link Barry Allen or Tony Stark

-- Labels - single int Character can have 5 labels to can range from -2 to +3 that effects the roll of the character. The labels are as following

-- -Danger

-- -Freak

-- -Savior

-- -Superior

-- -Mundane

-- campaign - a varchar with a maximum length of 255 characters, cannot be null, The name of the campaign the character is in.

-- PBid - an integer which is a foreign key reference to the playbook table

CREATE TABLE `masksChar` (

`Char\_ID` int(11) NOT NULL AUTO\_INCREMENT,

`hero\_name` varchar(255) NOT NULL,

`real\_name` varchar(255),

`Danger` int(1) NOT NULL DEFAULT '0',

`Freak` int(1) NOT NULL DEFAULT '0',

`Savior` int(1) NOT NULL DEFAULT '0',

`Superior` int(1) NOT NULL DEFAULT '0',

`Mundane` int(1) NOT NULL DEFAULT '0',

`potential` int(1) NOT NULL DEFAULT '0',

`PB\_ID` int(11) NOT NULL,

`Camp\_ID` int(11) NOT NULL ,

KEY `PB\_ID` (`PB\_ID`),

PRIMARY KEY (`Char\_ID`),

CONSTRAINT `playbook` FOREIGN KEY (`PB\_ID`) REFERENCES `masksPlaybook` (`PB\_ID`),

CONSTRAINT `camps` FOREIGN KEY (`Camp\_ID`) REFERENCES `masksCamp` (`Camp\_ID`) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create a table to work with conditions and what roll the effect

-- Con\_id - an auto incrementing integer which is the primary key

-- name - a varchar with a maximum length of 255 characters, not null, name of the condition

-- Rolls - a varchar with a maximum length of 255 characters, cannot be null, the rolls that are effect by the condition

CREATE TABLE `masksCon` (

`Con\_id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

`rolls` TEXT NOT NULL,

PRIMARY KEY (`Con\_id`)

) ENGINE=InnoDB;

-- Create a table to show how has influence over who

-- Char\_id - an integer which is a foreign key reference to character who has influence

-- Influence\_id - an integer which is a foreign key reference to character is influenced by a character

CREATE TABLE `masksInfluence` (

`Char\_id` int(11) NOT NULL DEFAULT '0',

`Influence\_id` int(11) NOT NULL DEFAULT '0',

PRIMARY KEY (`Char\_id`,`Influence\_id`),

KEY `Influence\_id` (`Influence\_id`),

CONSTRAINT `char\_inf` FOREIGN KEY (`Char\_id`) REFERENCES `masksChar` (`Char\_id`) ON DELETE CASCADE,

CONSTRAINT `influence\_char` FOREIGN KEY (`Char\_id`) REFERENCES `masksChar` (`Char\_id`) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create a table to work with character and conditions the character has

-- Char\_id - an integer which is a foreign key reference to character who has a condition

-- Con\_id - an integer which is a foreign key reference to the condition

CREATE TABLE `masksChar\_Con` (

`Char\_id` int(11) NOT NULL DEFAULT '0',

`Con\_id` int(11) NOT NULL DEFAULT '0',

KEY `Con\_id` (`Con\_id`),

PRIMARY KEY (`Char\_id`,`Con\_id`),

CONSTRAINT `con\_char` FOREIGN KEY (`Char\_id`) REFERENCES `masksChar` (`Char\_id`) ON DELETE CASCADE,

CONSTRAINT `con\_con` FOREIGN KEY (`Con\_id`) REFERENCES `masksCon` (`Con\_id`)

) ENGINE=InnoDB;

INSERT INTO `masksCon` (`name`, `rolls`)

VALUES ('Afraid', 'directly engage a threat');

INSERT INTO `masksCon` (`name`, `rolls`)

VALUES ('Angry', 'comfort or support or pierce the mask');

INSERT INTO `masksCon` (`name`, `rolls`)

VALUES ('Guilty', 'provoke someone or assess the situation');

INSERT INTO `masksCon` (`name`, `rolls`)

VALUES ('Hopeless', 'unleash your powers');

INSERT INTO `masksCon` (`name`, `rolls`)

VALUES ('Insecure', 'defend someone or reject others’ influence');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Beacon', '-1', '-1', '2', '0', '2');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Bull', '2', '1', '-1', '1', '-1');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Delinquent', '0', '0', '-1', '2', '1');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Doomed', '1', '1', '1', '-1', '0');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Janus', '0', '-1', '0', '0', '3');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Legacy', '-1', '0', '2', '0', '1');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Nova', '1', '2', '0', '0', '-1');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Outsider', '-1', '1', '0', '2', '0');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Protege', '-1', '0', '1', '2', '0');

INSERT INTO `masksPlaybook` (`name`, `Danger`, `Freak`, `Savior`, `Superior`, `Mundane`)

VALUES ('The Transformed', '1', '3', '0', '-1', '-1');

-- Test

Insert INTO `masksCamp` (`name`)

values('MCU');

Insert INTO `masksCamp` (`name`)

values('DC');

INSERT INTO `masksChar` (`hero\_name` ,`real\_name` ,`Danger` ,`Freak` ,`Savior` ,`Superior` ,

`Mundane` ,`PB\_ID` ,`Camp\_ID`)

VALUES ('Iron Man', 'Tony Stark' , '0', '0', '0', '0', '0', '2', '1');

INSERT INTO `masksChar` (`hero\_name` ,`Danger` ,`Freak` ,`Savior` ,`Superior` , `Mundane` ,`PB\_ID` ,`Camp\_ID`)

VALUES ('Thor', '0', '0', '0', '0', '0', '5', '1');

INSERT INTO `masksChar` (`hero\_name` ,`real\_name` ,`Danger` ,`Freak` ,`Savior` ,`Superior` ,`Mundane` ,`PB\_ID` ,`Camp\_ID`)

VALUES ('Arrow', 'Oliver Queen' , '0', '0', '0', '0', '0', '3', '2');

INSERT INTO `masksChar` (`hero\_name` ,`real\_name` ,`Danger` ,`Freak` ,`Savior` ,`Superior` ,`Mundane` ,`PB\_ID` ,`Camp\_ID`)

VALUES ('Starfire', ' Koriandr' , '0', '0', '0', '0', '0', '6', '2');

INSERT INTO `masksInfluence` (`Char\_ID`, `Influence\_id`)

VALUES ('1', '2');

INSERT INTO `masksChar\_Con` (`Char\_ID`, `Con\_id`)

VALUES ('1', '2');

**Data Manipulation Queries**

**Queries that your website uses to let your users interact with data**

SELECT Danger, Freak, Savior, Superior, Mundane FROM masksPlaybook WHERE PB\_ID = [character ID]

SELECT PB\_ID, name FROM masksPlaybook

SELECT Con\_id, name, rolls FROM masksCon

SELECT masksChar.Char\_ID, hero\_name, masksPlaybook.name AS playbook, masksCamp.name as camp FROM masksChar INNER JOIN masksPlaybook ON masksChar.PB\_ID = masksPlaybook.PB\_ID INNER JOIN masksCamp on masksCamp.Camp\_ID = masksChar.Camp\_ID

Gets characters that are not the character choosen

SELECT Char\_ID, hero\_name FROM masksChar WHERE Char\_ID != [character ID]

SELECT masksChar.Char\_ID, hero\_name, real\_name, PB\_ID, masksChar.Danger, masksChar.Freak, masksChar.Savior, masksChar.Superior, masksChar.Mundane, Potential FROM masksChar WHERE Char\_ID = [character ID]

SELECT masksChar.Char\_ID, hero\_name, real\_name, PB\_ID, masksChar.Danger, masksChar.Freak, masksChar.Savior, masksChar.Superior, masksChar.Mundane, Potential, Camp\_ID FROM masksChar WHERE Char\_ID = ?

Deletes all the conditions inflected on the character

DELETE FROM masksChar\_Con WHERE Char\_id = [character ID]

SELECT Con\_id FROM masksChar\_Con WHERE masksChar\_Con.Char\_ID = [character ID]

SELECT masksChar.hero\_name FROM masksInfluence INNER JOIN masksChar ON masksChar.Char\_ID = masksInfluence.Char\_ID WHERE Influence\_id = [character ID]

INSERT INTO `masksChar\_Con` (Char\_id, Con\_id) VALUES ([character ID], [condition ID])

Deletes all the influence the character has over other players

DELETE FROM masksInfluence WHERE Char\_id = [character ID]

INSERT INTO masksInfluence (Char\_id, Influence\_id) VALUES (([character ID], [another chracter ID])

INSERT INTO masksChar (hero\_name, real\_name, Danger, Freak, Savior, Superior, Mundane, PB\_ID) VALUES ([hero\_name],[real name], starting Danger from playbook , starting Freak from playbook , starting Savior from playbook , starting Superior from playbook , starting Mundane from playbook , [playbook id])

UPDATE masksChar SET hero\_name=[hero name], real\_name= [real name], PB\_ID= [playbook id], Danger = [danger], Freak = [freak], Savior = [savior], Superior = [superior], Mundane = [mundane], Potential = [pontential], Camp\_ID = [camp id] WHERE Char\_ID=[character\_id]

DELETE FROM masksChar WHERE Char\_ID = [character id]

SELECT Camp\_ID, name FROM masksCamp

INSERT INTO masksCamp (name) VALUES ([name of campaign])

DELETE FROM masksCamp WHERE Camp\_ID = [camp id]