Back up of Database AdventureWikiDB

Create DATABASE adventurewikidb;

Use adventurewikidb;

SHOW Databases;

-- SET foreign\_key\_checks = 0; -- disables foreign key check and it is used to drop tables that arte referenced with a foreign key

-- SET foreign\_key\_checks = 1; -- reenables foreign key checks (helpful when wanting to reference tables again)

drop table if exists MonstersandSpecs;

-- Table set up for initial monster spawns

create table MonstersandSpecs

(

MonsterID int Not null,

MonsterType varchar(255),

Attack varchar (255),

Health int not null,

Atlocation varchar (255),

Primary key (MonsterID)

);

-- creating monsters for each floor of the dungeon

Insert Into MonstersandSpecs (MonsterId, MonsterType, Attack, Health, Atlocation)

values (1, "Goblin", "Swipe Claws", 10, "Town");

Insert Into MonstersandSpecs (MonsterId, MonsterType, Attack, Health, Atlocation)

values (2, "Chief Goblin", "Heavy Slash", 20, "Cave Entrance");

Insert Into MonstersandSpecs (MonsterId, MonsterType, Attack, Health, Atlocation)

values (3, "Orc", "Charge Club swing", 35, "Cave Secret Room");

Insert Into MonstersandSpecs (MonsterId, MonsterType, Attack, Health, Atlocation)

values (4, "Dragon", "Fire Breath", 50, "Deepest Room");

-- demo for deleting an enemy from the game

Insert Into MonstersandSpecs (MonsterId, MonsterType, Attack, Health, Atlocation)

values (5, "Lord of the Void", "Black Hole", 500, "Real Final Boss Room");

/\* Thought about adding an extra hard boss to the end of the game,

but had trouble implementing it. I decided to delete that row

\*/

delete from Monstersandspecs where MonsterID = 5;

select \* From MonstersandSpecs;

drop table if exists MonsterIndepthStats;

-- More indepth Table that describes Monsters and where they are located

create table MonsterIndepthStats

(

DetailsID varchar (255),

MonsterDescription varchar(255),

Attackdamage int not null,

defense int not null,

MonsterType varchar(255),

Primary key (DetailsID)

);

Insert Into MonsterIndepthStats (DetailsId, MonsterDescription, AttackDamage, Defense, MonsterType)

values ("1", "2 foot tall Green Goblin with sharp teeth, gangly figure, and 1 inch claws", 3, 0, "Goblin");

Insert Into MonsterIndepthStats (DetailsId, MonsterDescription, AttackDamage, Defense, MonsterType)

values ("2", "Dark Grey 5' tall Goblin with long sword", 6, 2, "Chief Goblin");

Insert Into MonsterIndepthStats (DetailsId, MonsterDescription, AttackDamage, Defense, MonsterType)

values ("3", "Dark Green Orc protruding tusks and a massive great sword", 12, 4, "Orc");

Insert Into MonsterIndepthStats (DetailsId, MonsterDescription, AttackDamage, Defense, MonsterType)

values ("4", "20 foot tall Onyx dragon four powerful legs, two set of wings with 30' wing span. Blood is pouring out of its mouth as it rips the last of the knights that entered the chamber before you", 25, 10, "Dragon");

drop table if exists Locations;

-- table creation for locations in game

create table Locations

(

LocationId int not null,

PlaceName varchar(255) Not null,

Description varchar (255),

MonsterHere varchar(255),

MonsterID int,

Primary key (LocationID),

Foreign key (MonsterID) references MonstersandSpecs(MonsterID)

);

Insert Into Locations (LocationId, PlaceName, Description , MonsterHere, MonsterID)

values (1, "Town", "The starting location for the adventure", "Goblin", 1);

Insert Into Locations (LocationId, PlaceName, Description , MonsterHere, MonsterID)

values (2, "Cave Entrance", "A dark dank cave located just south of the Town, you see the chief of the goblins in the distance", "Chief Goblin", 2);

Insert Into Locations (LocationId, PlaceName, Description , MonsterHere, MonsterID)

values (3, "Cave Secret Room", "Congrats you have found the Secret Room that was hidden behind the fake wall", "Orc", 3);

Insert Into Locations (LocationId, PlaceName, Description , MonsterHere, MonsterID)

values (4, "Deepest Room", "final boss room. Good Luck!", "Dragon", 4);

Insert Into Locations (LocationId, PlaceName, Description , MonsterHere, MonsterID)

values (5, "Champions Throne Room", "Winning Area: you have beaten the game Excellent work!", null, null);

show tables;

select \* From Locations;

Drop table if exists loot;

-- table for loot that each monster will hold and the location the loot is in

create table Loot

(

LootID int not null,

Lootname varchar(255),

attackvalue int,

monstername varchar(255),

PlaceName varchar(255),

locationID int,

Primary key (LootID),

Foreign key (LocationID) references Locations(LocationID)

);

Insert into Loot (LootId, Lootname, attackvalue, monstername, PlaceName, LocationID)

values (1, "long sword", 10, "Goblin", "Town", 1);

Insert into Loot (LootId, Lootname, attackvalue, monstername, PlaceName, locationID)

values (2, "great sword", 15, "Chief Goblin", "Cave Entrance", 2);

Insert into Loot (LootId, Lootname, attackvalue, monstername, PlaceName, LocationID)

values (3, "colossal great sword", 20, "Orc", "Cave Secret Entrance", 3);

Insert into Loot (LootId, Lootname, attackvalue, monstername, PlaceName, LocationID)

values (4, "Dragon Heart", null, "Dragon", "Deepest Room", 4);

show tables;

select \* from Loot;

select \* from MonstersandSpecs;

select \* from Locations;

select \* from loot;

SELECT \*FROM MonstersandSpecs WHERE health > 20;

Select \* from Loot where atlocation like "\_o%";

Select \* from MonstersandSpecs Cross Join MonsterIndepthStats;

Select locations.placename,

monsterhere

from

locations

Left Join loot on loot.PlaceName = locations.placename;

Update Monsterindepthstats set defense = 2 where detailsid = 1;

SELECT

\*

FROM

Monsterindepthstats;

Update Monsterindepthstats set defense = 0 where detailsid = 1;

SELECT

\*

FROM

Monsterindepthstats;

Alter Table Monsterindepthstats

add healthminusattack int;

select \* from monsterindepthstats;

alter table Monsterindepthstats

drop column healthminusattack;