Nicholas Lichtsinn

Project Parts 1 & 2

IST 659

**Yu Yu Hakusho Database**

**Summary:**

This database will house the characters and abilities shown in the manga/anime Yu Yu Hakusho. I will account for each character's name, power level, abilities, and their team affiliations. This concept could be expanded to include other manga/anime as well. I answered my data questions with sql code in the form of views, stored procedures, ‘select’ statements and in Access after I created the GUI. I chose to use Access because it was a great way to put the information together and show what I needed to and I hadn’t worked in Access before so I wanted to learn. I included snippets of my sql code to show my work and will attach the query file with my project submission if you need all of it.

**Stakeholders:**

The stakeholders for this database will be the users, it will be free to use for everyone.

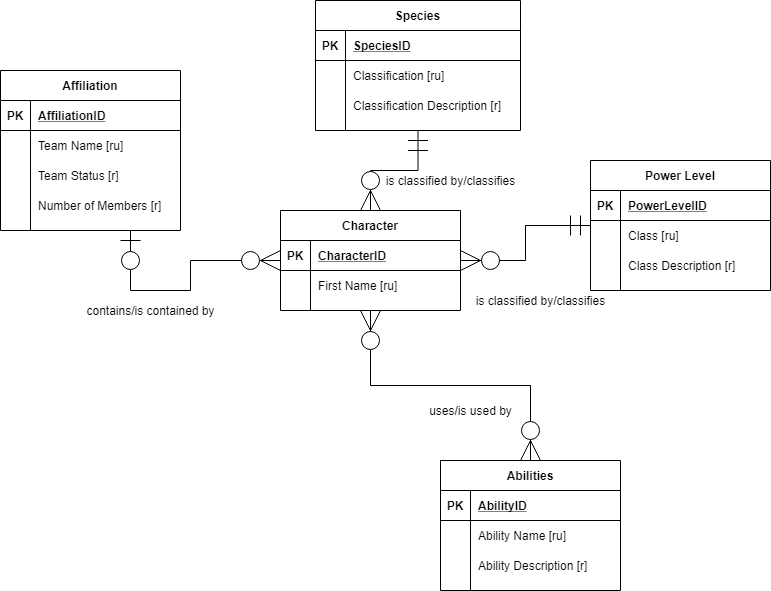
**Business Rules:**

* A character can have multiple affiliations.
* A character has one power level, each power level will have multiple characters.
* Each affiliation will have multiple characters.
* Each character can have multiple abilities and abilities can be shared by multiple characters.
* A character is classified by one species, each species will have multiple characters.
* Each character has one power level.

**Data Questions:**

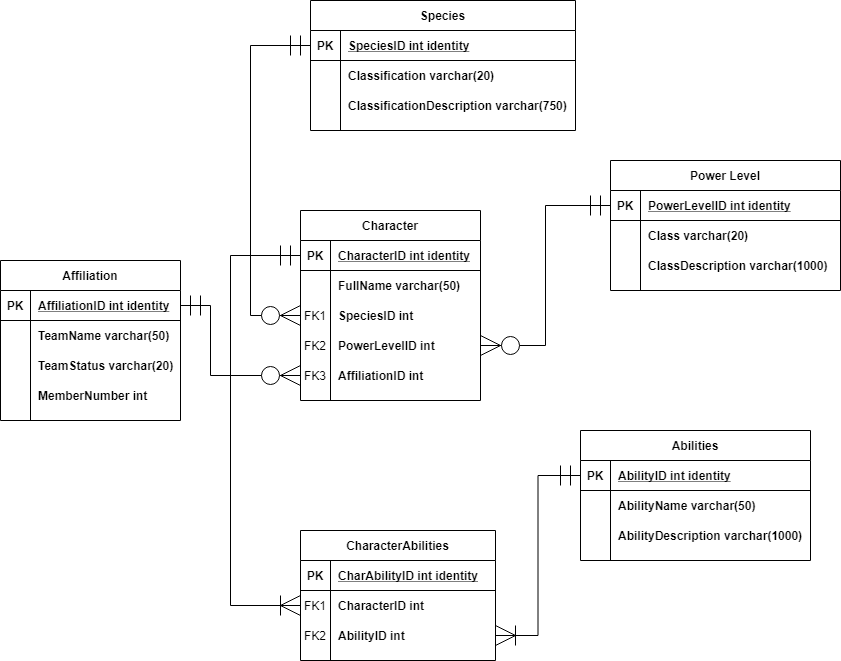
* What is the most common power level of the characters in the series?
* How many demons vs humans are there in the series?
* What are the teams that are still active and who is affiliated with them?
* Who are the strongest characters? Who are the weakest?
* If the humans and demons all fought, which group would win based on power level?

**Conceptual Model:**

****

**Logical Model:**

* I will be implementing all Primary Keys as Surrogate Keys using the SQL Server identity property.



**Physical Database Design:**

--Dropping my tables

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'CharacterAbilities')

BEGIN

DROP TABLE CharacterAbilities

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'Characters')

BEGIN

DROP TABLE Characters

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'Abilities')

BEGIN

DROP TABLE Abilities

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'Affiliation')

BEGIN

DROP TABLE Affiliation

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'PowerLevel')

BEGIN

DROP TABLE PowerLevel

END

GO

IF EXISTS (SELECT \* FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'Species')

BEGIN

DROP TABLE Species

END

GO

--Creating my tables

--Creating Species table

CREATE TABLE Species (

-- Columns for the Species Table

SpeciesID int identity,

Classification varchar(20) not null,

ClassificationDescription varchar(750) not null

-- Constraints

CONSTRAINT PK\_Species PRIMARY KEY (SpeciesID),

CONSTRAINT U1\_Species UNIQUE (Classification)

)

--Creating PowerLevel table

CREATE TABLE PowerLevel (

-- Columns for the PowerLevel Table

PowerLevelID int identity,

Class varchar(20) not null,

ClassDescription VARCHAR(1000) not null,

-- Constraints

CONSTRAINT PK\_PowerLevel PRIMARY KEY (PowerLevelID),

CONSTRAINT U1\_PowerLevel UNIQUE (Class)

)

-- adding INSERT statements that show how to insert when a column value is not provided anne server is responsible for setting a column to a default.

ALTER TABLE PowerLevel ADD CONSTRAINT DF\_PowerLevel DEFAULT 'N/A' FOR ClassDescription -- setting default for class description to 'N/A'

-- Creating Affiliation table

CREATE TABLE Affiliation (

-- Columns for the Affiliation Table

AffiliationID int identity,

TeamName varchar(50) not null,

TeamStatus varchar(30) not null,

MemberNumber int not null

-- Constraints

CONSTRAINT PK\_Affiliation PRIMARY KEY (AffiliationID),

CONSTRAINT U1\_Affiliation UNIQUE (TeamName)

)

-- Creating Abilities table

CREATE TABLE Abilities (

-- Columns for the Abilities Table

AbilityID int identity,

AbilityName varchar(50) not null,

AbilityDescription varchar(1000) not null

-- Constraints

CONSTRAINT PK\_Abilites PRIMARY KEY (AbilityID),

CONSTRAINT U1\_Abilities UNIQUE (AbilityName)

)

-- Creating the Characters Table

CREATE TABLE Characters (

-- Columns for the Characters Table

CharacterID int identity,

FullName varchar(50) not null,

SpeciesID int not null,

PowerLevelID int not null,

AffiliationID int not null,

-- Constraints on the Characters Table

CONSTRAINT PK\_Characters PRIMARY KEY (CharacterID),

CONSTRAINT FK1\_Characters FOREIGN KEY (SpeciesID) REFERENCES Species(SpeciesID),

CONSTRAINT FK2\_Characters FOREIGN KEY (PowerLevelID) REFERENCES PowerLevel(PowerLevelID),

CONSTRAINT FK3\_Characters FOREIGN KEY (AffiliationID) REFERENCES Affiliation(AffiliationID),

CONSTRAINT U1\_Characters UNIQUE (FullName)

)

-- Creating the CharacterAbilities table

CREATE TABLE CharacterAbilities (

-- Columns for the CharacterAbilities Table

CharAbilityID int identity,

CharacterID int not null,

AbilityID int not null

-- Constraints

CONSTRAINT PK\_CharacterAbilities PRIMARY KEY (CharAbilityID),

CONSTRAINT FK1\_CharacterAbilities FOREIGN KEY (CharacterID) REFERENCES Characters(CharacterID),

CONSTRAINT FK2\_CharacterAbilities FOREIGN KEY (AbilityID) REFERENCES Abilities(AbilityID)

**Data Creation:**

-- inserting values into Species table with comma delimiter

INSERT INTO Species(Classification, ClassificationDescription) VALUES ('Human','The term psychic is used to define a human who possesses supernatural powers and has the ability to use Spirit Energy (or Demon Energy in some cases) as well as heightened spiritual awareness to see ghosts and demons, telepathy, shaping energy, and manipulating objects and/or matter. The known psychics in the English anime were either fighters like Genkai, who gained such abilities through severe martial arts training, people born with unnaturally high spiritual awareness like the Kuwabara siblings, or people who gained their powers as a result of the Sensui Seven''s Gatekeeper opening the Makai Barrier.')

, ('Demon', 'While the demons of Yu Yu Hakusho are not necessarily "demonic" one way or another, they amusingly all hail from Makai (Demon World) and eventually are ruled by a Mao (Demon King). There are several species of spirits/demons that inhabit the planes or worlds of Yu Yu Hakusho. Many of them appear very similar to humans and most have a humanoid features.')

, ('Example', 'Example for delete statement')

-- inserting values into Power Level table with individual lines

INSERT INTO PowerLevel(Class, ClassDescription) VALUES ('S', 'This class is remarkably harder to reach than other classes, however, though the powers of A class were already so high, they are nothing compared to this rank. S class warriors are viewed as deities of destruction, and A classes are fodder to S classes. S classes are incredibly more intelligent than A classes, barring a few exceptions, and are viewed as the nearly omniscient. Because of this, the strongest characters in the YuYu Hakusho franchise are put in this class')

INSERT INTO PowerLevel(Class, ClassDescription) VALUES ('A', 'The only humans who have reached this class went through an overwhelming trauma, and leveled into this class to deal with it. They were unaware of this jump in class. Demons with this power do not reside in the Human or Spirit World. They are only of Demon World, and are either born with this power, or have acquired it through immense training. The abilities are even more astounding than what can be displayed at B class power, however, this is the second smallest class, as just a few more months of intense training is what is necessary to reach the last class. Demons with this rank are typically far more intelligent than average humans. They are considered to be country level threats by Spirit World.')

-- inserting values into Affiliation table

INSERT INTO Affiliation(TeamName, TeamStatus, MemberNumber) VALUES ('Dr. Ichigaki Team', 'Disbanded', 6)

INSERT INTO Affiliation(TeamName, TeamStatus, MemberNumber) VALUES ('Meikai Gods', 'Active', 4)

INSERT INTO Affiliation(TeamName, TeamStatus, MemberNumber) VALUES ('Raizen''s Friends', 'Active', 9)

-- inserting values into Abilities table

INSERT INTO Abilities(AbilityName, AbilityDescription) VALUES ('Demon-Gun', 'Yusuke concentrates his Demon-Energy at the tip of the index finger, until it becomes unstable, then focuses on core energy and expands it and releases a mental trigger.')

INSERT INTO Abilities(AbilityName, AbilityDescription) VALUES ('Dimension Sword', 'Kuwabara channels all of his energy in the palm of the hand, and then increases the load so that a small peak of yellow/green energy starts to glow until it illuminates his entire hand. He manifests the energy into a sword, albeit different from his normal Spirit Sword.')

-- inserting values into Characters tables

INSERT INTO Characters(FullName, SpeciesID, PowerLevelID, AffiliationID) VALUES ('Yusuke Urameshi', 1, 1, 7)

INSERT INTO Characters(FullName, SpeciesID, PowerLevelID, AffiliationID) VALUES ('Kazuma Kuwabara', 1, 1, 10)

INSERT INTO Characters(FullName, SpeciesID, PowerLevelID, AffiliationID) VALUES ('Kurama', 2, 1, 10)

INSERT INTO Characters(FullName, SpeciesID, PowerLevelID, AffiliationID) VALUES ('Hiei', 2, 1, 10)

INSERT INTO Characters(FullName, SpeciesID, PowerLevelID, AffiliationID) VALUES ('Genkai', 1, 3, 10)

GO

-- dropping stored procedure

IF OBJECT\_ID ('InsertStoredProcedure', 'P') IS not null

DROP PROCEDURE InsertStoredProcedure;

GO

-- insert from a stored procedure

CREATE PROCEDURE InsertStoredProcedure

AS

BEGIN

INSERT INTO [Characters]([FullName], [SpeciesID], [PowerLevelID], [AffiliationID])

VALUES ('Yakumo', 2, 1, 2), ('Raigo', 2, 2, 2), ('Majari', 2, 2, 2), ('Kaiki', 2, 2, 2)

END

GO

-- inserting values into CharacterAbilities table

INSERT INTO CharacterAbilities(CharacterID, AbilityID) VALUES (1, 12)

**Data Manipulation:**

-- example of an update statement

UPDATE Characters SET AffiliationID = 10 WHERE CharacterID= 1

SELECT \* FROM Characters

-- example of a delete statement

DELETE FROM Species WHERE Classification = 'Example'

SELECT \* FROM Species

-- executing insert stored procedure

EXEC dbo.InsertStoredProcedure

GO

SELECT \* FROM Characters

GO

**Answering Data Questions:**

**-- Question 1. What is the most common power level of the characters in the series?**

-- dropping and creating a view to show characters and their power level class

IF OBJECT\_ID ('CharacterPowerLevel') IS not null

DROP VIEW CharacterPowerLevel;

GO

CREATE VIEW CharacterPowerLevel AS

SELECT FullName, PowerLevel.PowerLevelID, Class, ClassDescription

FROM Characters, PowerLevel

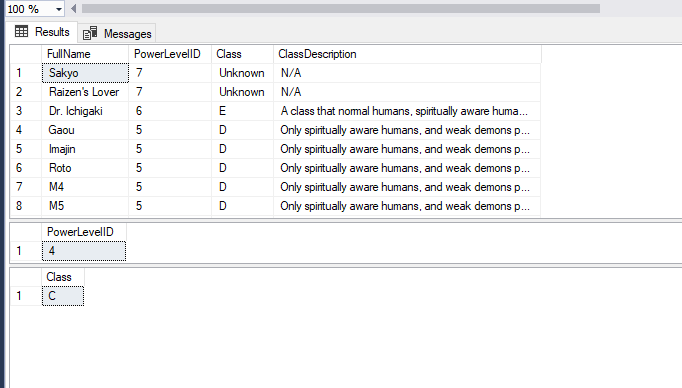
WHERE Characters.PowerLevelID = PowerLevel.PowerLevelID

GO

SELECT \* FROM CharacterPowerLevel ORDER BY PowerLevelID DESC -- looking at the view with the higher power levels on top

SELECT TOP 1 PowerLevelID FROM Characters GROUP BY [PowerLevelID] ORDER BY COUNT(\*) DESC -- counting the powerlevelid's to get the most common power level

SELECT Class FROM PowerLevel WHERE PowerLevelID = 4 -- getting the class for the associated power level id

-- The most common power level is Class 'C'

**-- Question 2. How many Demons vs Humans are there in the show**

IF OBJECT\_ID ('DemonsVsHumans') IS not null

DROP VIEW DemonsVsHumans;

GO

CREATE VIEW DemonsVSHumans AS

SELECT TOP 1 (SELECT TOP 1 COUNT(SpeciesID) FROM Characters WHERE SpeciesID = 1) AS 'Human Count', (SELECT TOP 1 COUNT(SpeciesID) FROM Characters WHERE SpeciesID = 2) AS 'Demon Count'

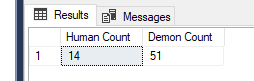
FROM Species

INNER JOIN Characters ON Characters.SpeciesID = Species.SpeciesID

GO

SELECT \* FROM DemonsVsHumans -- There are 14 Humans and 51 Demons

GO

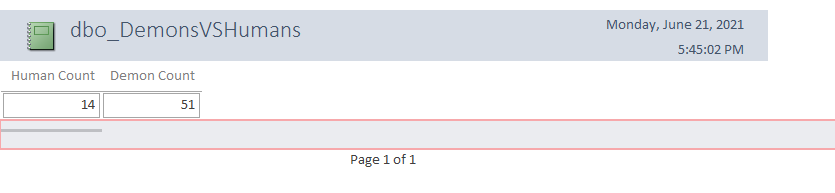


In Access:

Form



Report



**-- Question 3. What is the number of teams/affiliations that are still active?**

IF OBJECT\_ID ('ActiveAffiliations') IS not null -- dropping procedure

DROP PROCEDURE ActiveAffiliations;

GO

CREATE PROCEDURE ActiveAffiliations AS -- creating procedure to show current active teams since this can be prone to change

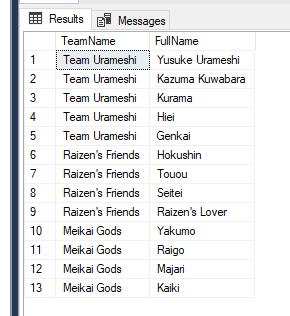
SELECT TeamName, FullName

FROM Affiliation

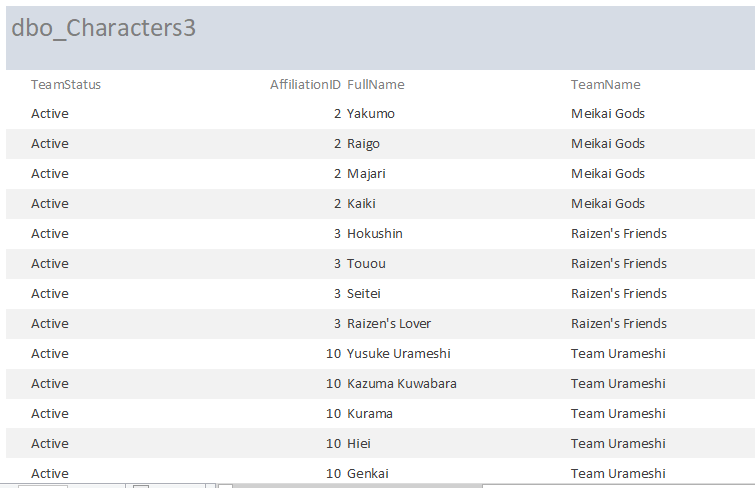
INNER JOIN Characters ON Characters.AffiliationID = Affiliation.AffiliationID

WHERE TeamStatus = 'Active'

GO

EXEC dbo.ActiveAffiliations

In Access: Report



**-- Question 4. Who are the strongest characters? Who are the weakest?**

IF OBJECT\_ID ('StrongestCharacters') IS not null -- dropping procedure

DROP PROCEDURE StrongestCharacters;

GO

CREATE PROCEDURE StrongestCharacters AS -- creating procedure to show strongest characters since this can be prone to change

SELECT FullName, Class

FROM Characters

INNER JOIN PowerLevel ON Characters.PowerLevelID = PowerLevel.PowerLevelID

WHERE Characters.PowerLevelID = 1

GO

EXEC dbo.StrongestCharacters -- shows list of the current characters that are classed in the strongest power level class

GO

IF OBJECT\_ID ('WeakestCharacters') IS not null -- dropping procedure

DROP PROCEDURE WeakestCharacters;

GO

CREATE PROCEDURE WeakestCharacters AS -- creating procedure to show strongest characters since this can be prone to change

SELECT FullName, PowerLevel.Class

FROM Characters

INNER JOIN PowerLevel ON Characters.PowerLevelID = PowerLevel.PowerLevelID

WHERE Characters.PowerLevelID >= 5

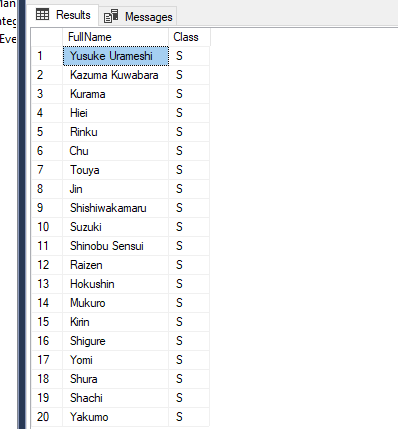
GROUP BY PowerLevel.PowerLevelID, PowerLevel.Class, FullName

ORDER BY PowerLevel.PowerLevelID ASC

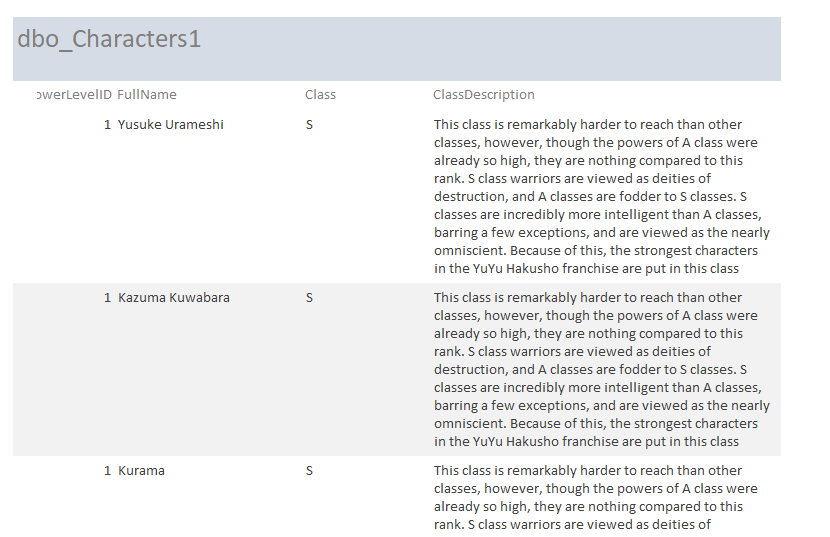
GO

EXEC dbo.WeakestCharacters -- shows list of the current characters that are classed in the weakest or unknown power levels ordered by power level

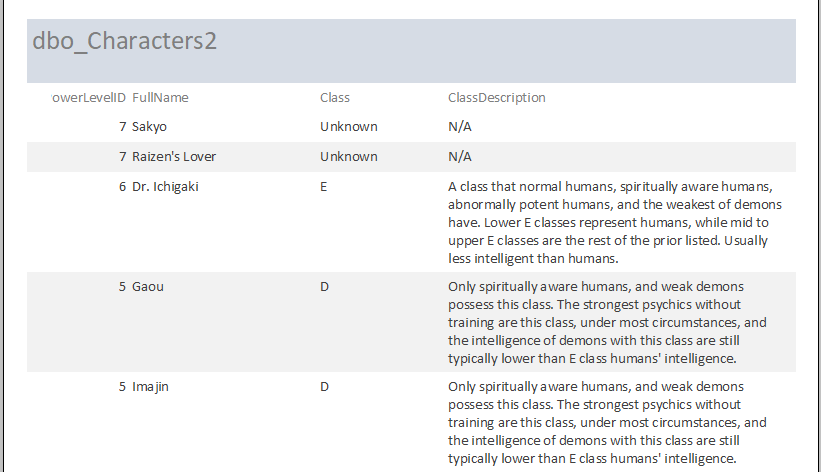
GO



In Access: Strongest Report

****

Weakest Report



**-- Question 5. If the humans and demons all fought, which group would win based on power level?**

IF OBJECT\_ID ('SpeciesPowerLevels') IS not null -- dropping procedure

DROP PROCEDURE SpeciesPowerLevels;

GO

CREATE PROCEDURE SpeciesPowerLevels AS -- creating procedure to show species power level

SELECT Species.Classification, SUM(Characters.PowerLevelID) AS 'Species PowerLevel'

FROM Characters

JOIN Species ON Characters.SpeciesID = Species.SpeciesID

GROUP BY Characters.SpeciesID, Species.Classification

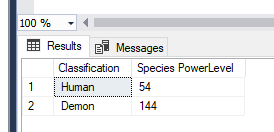
ORDER BY 'Species PowerLevel' ASC

GO

EXEC dbo.SpeciesPowerLevels -- shows the sum of the power level ID's giving us the total power level of each group

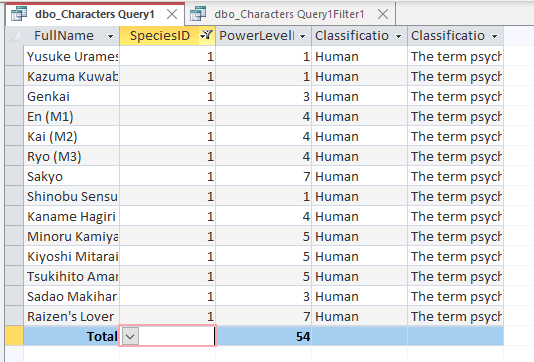
GO

-- Humans are much stronger because Power Levels are inverse to the PowerLevelID's

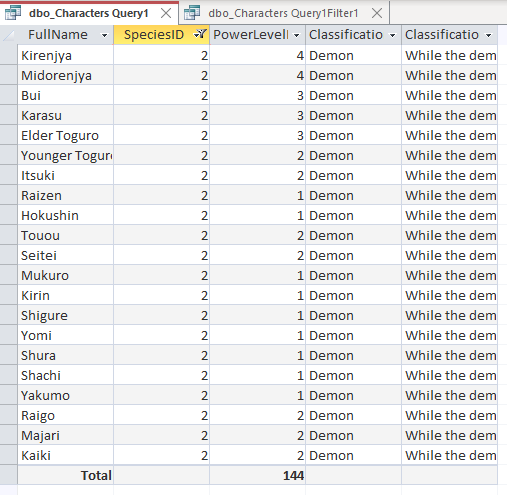


In Access:

Species Power Level Humans



Species Power Level Demons



**Reflection:**

This took me a lot longer than I expected but I really enjoyed learning the ins and outs of the process along the way, learning what will and will not work for my sql code and having to troubleshoot to get everything right was a very enjoyable process. As I am sure you will notice I started with a different idea for the first half of the class and changed to something that I was more interested in and could use in the future. I did not go into this course expecting to enjoy sql all that much, but I somehow had a blast creating this database and learning more about databases, sql and access.

The next time I do this I will make sure to pick a topic that I am passionate about from the beginning but also a topic that allows for more flexibility and challenges that I did not get to approach this time around. Getting the conceptual and logical models right is imperative and can really set you up for success for the rest of the project.

This class has given me a lot of insight into how databases are set up, how databases and DMS’s work together with applications and users and I have a much better understanding of how to work between all these tools and why there are limitations on certain things depending on how it was setup and works with the other tools.

Lastly, I would just like to thank you again for teaching this class the format you provided is very conducive to learning this material that can be confusing. It is also nice to learn from someone who is obviously passionate about the topic and about teaching.