Simpsons Character Database

Iglika Hadjiyska Database Management Spring 2016

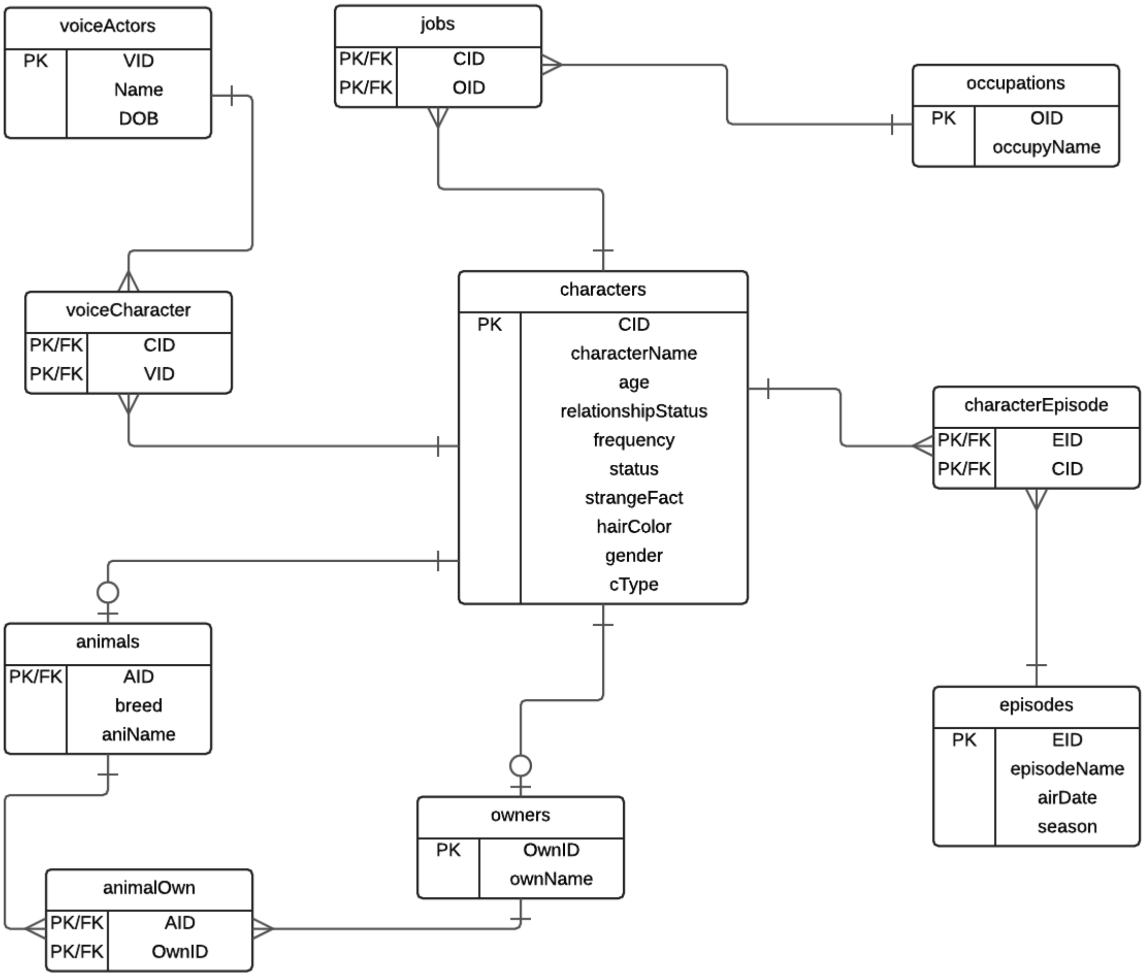
Table of Contents

Executive Summary

This document is meant to showcase the structure and functions of a database designed for The Simpsons television show. The show has close to 2,000 characters and counting, as well as an enormous numbers of episodes and voice actors. As a result, this database has been compiled to track these characters and information about them. Since there is so much information, the scope of this database has been reduced to the first nine seasons, with only canon episodes and characters. This is meant to keep this instance of the database short and manageable although the implemented database would be larger in scale. The expected users can be the viewing public but more specifically, it would be the show runners, as an attempt to provide consistency in characters, which has been lost in the recent seasons of the show.

The objective of this database is to provide quick access to characters in case of discontinuity within an episode, or to provide statistics about the characters themselves. As such, both the public and the writers would only be allowed certain security functions, as will be discussed later on.

Entity Relationship Diagram



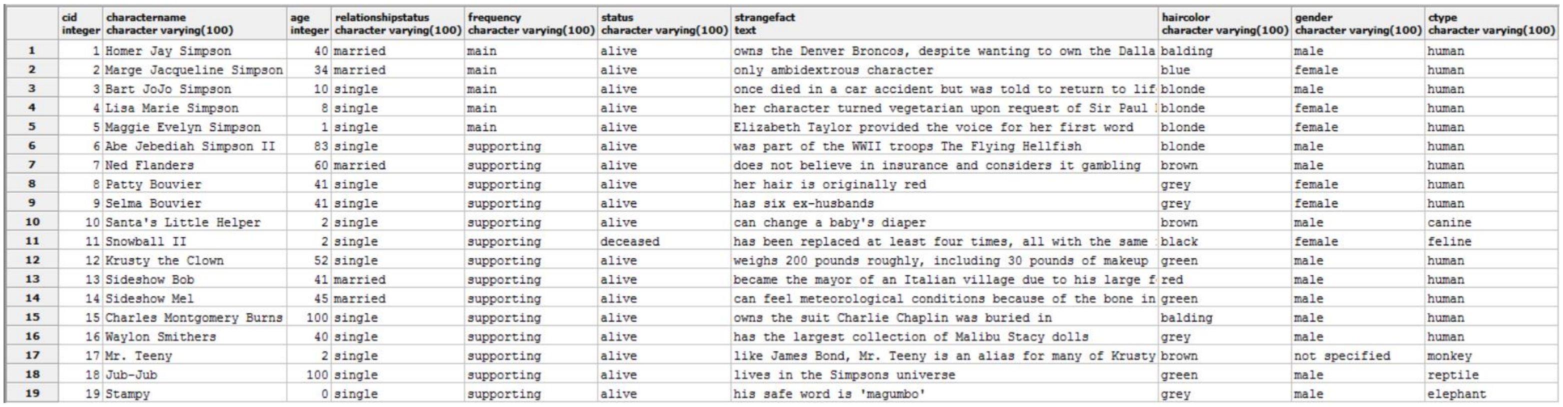
Tables

characters

This table lists the characters and their attributes

```
CREATE TABLE characters (
      cid integer NOT NULL,
      characterName varchar(100) NOT NULL DEFAULT 'not specified',
      age integer NOT NULL,
      relationshipStatus varchar(100) NOT NULL DEFAULT 'single',
      frequency varchar(100) DEFAULT 'not specified',
      status varchar(100) DEFAULT 'alive',
      strangeFact text DEFAULT 'lives in the Simpsons universe',
      hairColor varchar(100) DEFAULT 'not specified',
      gender varchar(100) DEFAULT 'not specified',
      cType varchar(100) DEFAULT 'not specified',
   primary key (cid)
);
```

```
(cid) → characterName, age, relationshipStatus, frequency, status,
strangeFact, hairColor, gender, cType
```

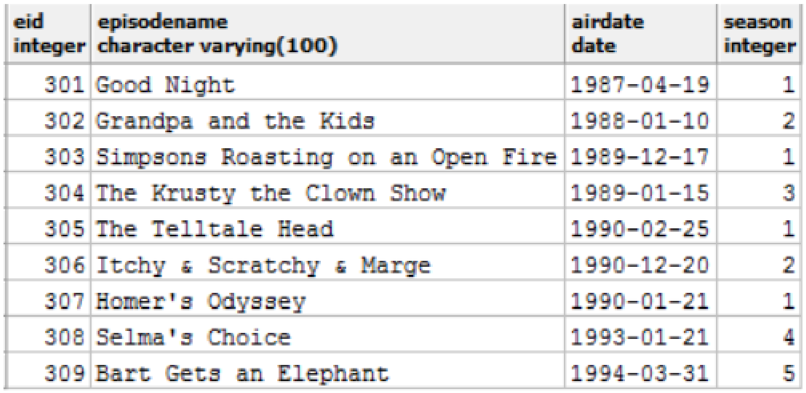


	cid	charactername	age	relationshipstatus	frequency	status	strangefact	haircolor	gender	ctype
	integer	character varying(100)	integer	character varying(100)	character varying(100)	character varying(100)	text	character varying(100)	character varying(100)	character varying(100)
1	1	Homer Jay Simpson	40	married	main	alive	owns the Denver Broncos, despite wanting to own the Dalla	balding	male	human
2	2	Marge Jacqueline Simpson	34	married	main	alive	only ambidextrous character	blue	female	human
3	3	Bart JoJo Simpson	10	single	main	alive	once died in a car accident but was told to return to lif	blonde	male	human
4	4	Lisa Marie Simpson	8	single	main	alive	her character turned vegetarian upon request of Sir Paul	blonde	female	human
5	5	Maggie Evelyn Simpson	1	single	main	alive	Elizabeth Taylor provided the voice for her first word	blonde	female	human
6	6	Abe Jebediah Simpson II	83	single	supporting	alive	was part of the WWII troops The Flying Hellfish	blonde	male	human
7	7	Ned Flanders	60	married	supporting	alive	does not believe in insurance and considers it gambling	brown	male	human
8	8	Patty Bouvier	41	single	supporting	alive	her hair is originally red	grey	female	human
9	9	Selma Bouvier	41	single	supporting	alive	has six ex-husbands	grey	female	human
10	10	Santa's Little Helper	2	single	supporting	alive	can change a baby's diaper	brown	male	canine
11	11	Snowball II	2	single	supporting	deceased	has been replaced at least four times, all with the same	black	female	feline
12	12	Krusty the Clown	52	single	supporting	alive	weighs 200 pounds roughly, including 30 pounds of makeup	green	male	human
13	13	Sideshow Bob	41	married	supporting	alive	became the mayor of an Italian village due to his large f	red	male	human
14	14	Sideshow Mel	45	married	supporting	alive	can feel meteorological conditions because of the bone in	green	male	human
15	15	Charles Montgomery Burns	100	single	supporting	alive	owns the suit Charlie Chaplin was buried in	balding	male	human
16	16	Waylon Smithers	40	single	supporting	alive	has the largest collection of Malibu Stacy dolls	grey	male	human
17	17	Mr. Teeny	2	single	supporting	alive	like James Bond, Mr. Teeny is an alias for many of Krusty	brown	not specified	monkey
18	18	Jub-Jub	100	single	supporting	alive	lives in the Simpsons universe	green	male	reptile
19	19	Stampy	0	single	supporting	alive	his safe word is 'magumbo'	grey	male	elephant

episodes

This tables lists the first episode a character appeared in

```
CREATE TABLE episodes (

      eid integer NOT NULL,

      episodeName varchar(100) NOT NULL DEFAULT 'not specified',

      airDate date,

      season int,

   primary key (eid)

);

(eid) → episodeName, airdate, season
```

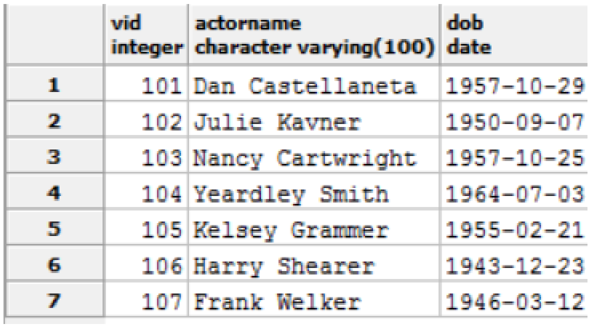


eid integer	episodename character varying(100)	airdate date	season integer
301	Good Night	1987-04-19	1
302	Grandpa and the Kids	1988-01-10	2
303	Simpsons Roasting on an Open Fire	1989-12-17	1
304	The Krusty the Clown Show	1989-01-15	3
305	The Telltale Head	1990-02-25	1
306	Itchy & Scratchy & Marge	1990-12-20	2
307	Homer's Odyssey	1990-01-21	1
308	Selma's Choice	1993-01-21	4
309	Bart Gets an Elephant	1994-03-31	5

voiceActors

This table showcases the voice actors of all the characters

```
CREATE TABLE voiceActors (

      vid integer NOT NULL,

      actorName varchar(100) NOT NULL DEFAULT 'not specified',

      dob date,

   primary key (vid)

);

(vid) → actorName, dob
```



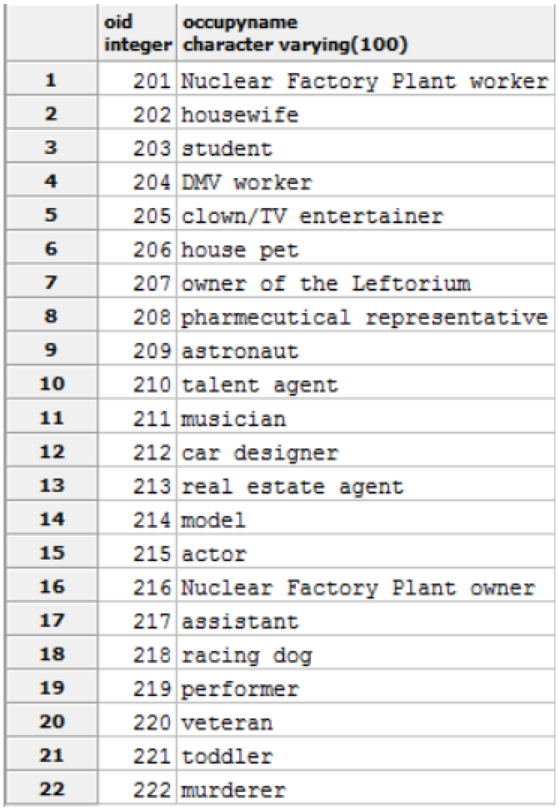
	vid integer	actorname character varying(100)	dob date
1	101	Dan Castellaneta	1957-10-29
2	102	Julie Kavner	1950-09-07
3	103	Nancy Cartwright	1957-10-25
4	104	Yeardley Smith	1964-07-03
5	105	Kelsey Grammer	1955-02-21
6	106	Harry Shearer	1943-12-23
7	107	Frank Welker	1946-03-12

occupations

This table lists all of the jobs that characters have ever had

```
CREATE TABLE occupations (

      oid integer NOT NULL,

      occupyName varchar(100) NOT NULL DEFAULT 'not specified',

   primary key (oid)

);
```

(oid) → occupyName

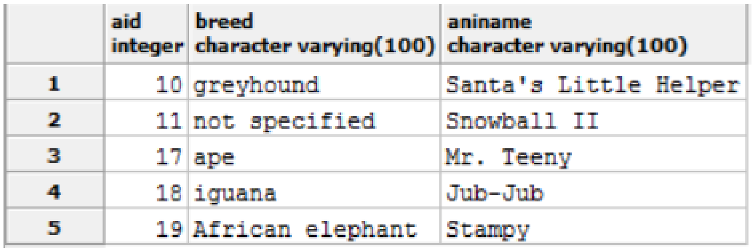


	oid integer	occupyname character varying(100)
1	201	Nuclear Factory Plant worker
2	202	housewife
3	203	student
4	204	DMV worker
5	205	clown/TV entertainer
6	206	house pet
7	207	owner of the Leftorium
8	208	pharmecutical representative
9	209	astronaut
10	210	talent agent
11	211	musician
12	212	car designer
13	213	real estate agent
14	214	model
15	215	actor
16	216	Nuclear Factory Plant owner
17	217	assistant
18	218	racing dog
19	219	performer
20	220	veteran
21	221	toddler
22	222	murderer

animals

This is an entity subtype of the table characters. Some characters are animals and have to be separated accounted for. The attribute ‘breed’ accounts for the specific type of animal, elaborating on the ‘cType’ attribute of the characters table.

```
CREATE TABLE animals (

     aid integer NOT NULL references characters(cid),

     breed varchar(100) DEFAULT 'not specified',

     aniName varchar(100) DEFAuLT 'not specified',

   primary key (aid)

);

(aid) → breed, aniName
```

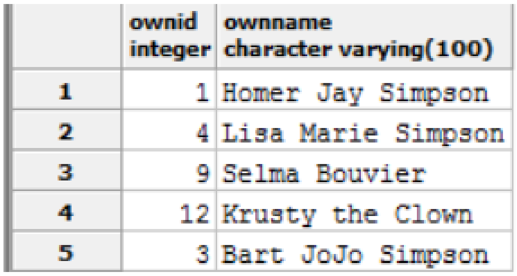


	aid integer	breed character varying(100)	aniname character varying(100)
1	10	greyhound	Santa's Little Helper
2	11	not specified	Snowball II
3	17	ape	Mr. Teeny
4	18	iguana	Jub-Jub
5	19	African elephant	Stampy

owners

This is also an entity subtype of characters. Some of the animals are pets and as a result, must have owners.

```
CREATE TABLE owners (

     ownid integer NOT NULL references characters(cid),

     ownName varchar(100) DEFAULT 'not specified',

   primary key (ownid)

);

(ownid) → ownName
```



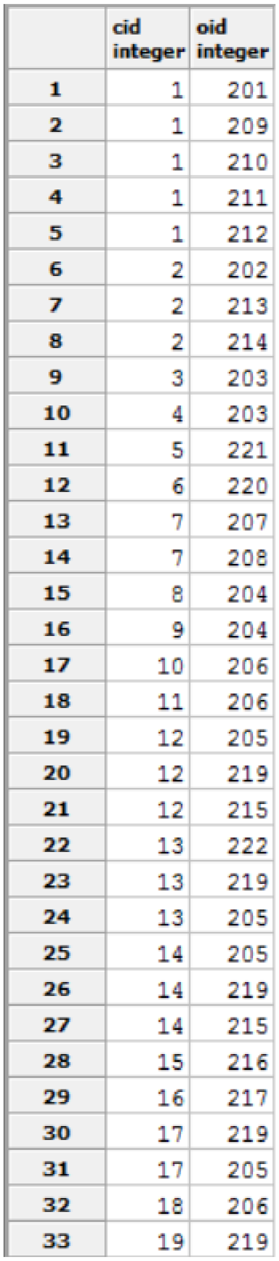
	ownid integer	ownname character varying(100)
1	1	Homer Jay Simpson
2	4	Lisa Marie Simpson
3	9	Selma Bouvier
4	12	Krusty the Clown
5	3	Bart JoJo Simpson

jobs

This is a linking table between characters and occupations, meant to link together the character names with their job names.

```
CREATE TABLE jobs (
      cid integer NOT NULL references characters(cid),
      oid integer NOT NULL references occupations(oid),
   primary key (cid, oid)
);
```

(cid,oid) →

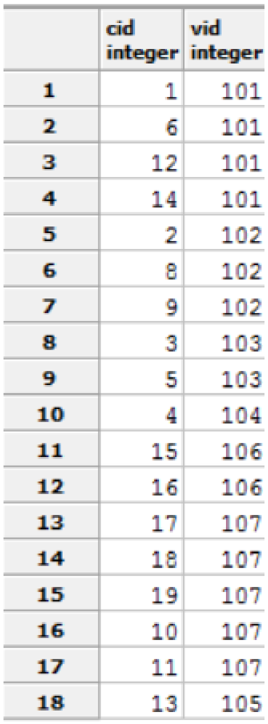


	cid integer	oid integer
1	1	201
2	1	209
3	1	210
4	1	211
5	1	212
6	2	202
7	2	213
8	2	214
9	3	203
10	4	203
11	5	221
12	6	220
13	7	207
14	7	208
15	8	204
16	9	204
17	10	206
18	11	206
19	12	205
20	12	219
21	12	215
22	13	222
23	13	219
24	13	205
25	14	205
26	14	219
27	14	215
28	15	216
29	16	217
30	17	219
31	17	205
32	18	206
33	19	219

voiceCharacter

This table is also a linking table between voiceActors and characters. It meant to link the two together, by their IDs.

```
CREATE TABLE voiceCharacter (

      cid integer NOT NULL references characters(cid),

      vid integer NOT NULL references voiceActors(vid),

   primary key (cid, vid)
);

(cid, vid) →
```

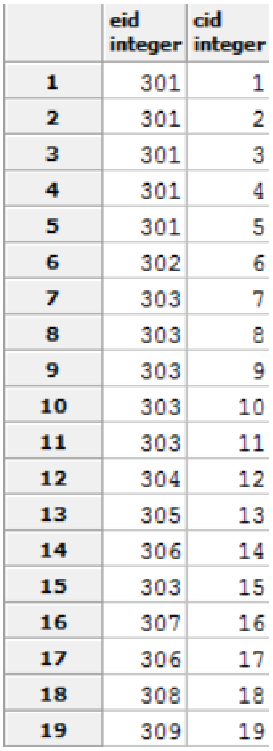


	cid integer	vid integer
1	1	101
2	6	101
3	12	101
4	14	101
5	2	102
6	8	102
7	9	102
8	3	103
9	5	103
10	4	104
11	15	106
12	16	106
13	17	107
14	18	107
15	19	107
16	10	107
17	11	107
18	13	105

characterEpisode

This table links characters and episodes, to show which characters were in which episdoes.

```
CREATE TABLE characterEpisode (

      eid integer NOT NULL references episodes(eid),

      cid integer NOT NULL references characters(cid),

   primary key (eid, cid)

);

(eid, cid) →
```



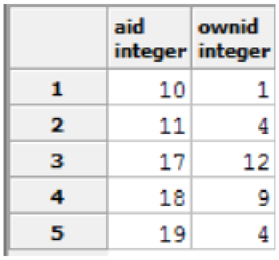
	eid integer	cid integer
1	301	1
2	301	2
3	301	3
4	301	4
5	301	5
6	302	6
7	303	7
8	303	8
9	303	9
10	303	10
11	303	11
12	304	12
13	305	13
14	306	14
15	303	15
16	307	16
17	306	17
18	308	18
19	309	19

animalOwn

This is a linking table between animals and owners, meant to link the two with their IDS.

```
CREATE TABLE animalOwn (

      aid integer NOT NULL references animals(aid),

      ownid integer NOT NULL references owners(ownid),

   primary key(aid, ownid)

);
```

(aid, ownid) →



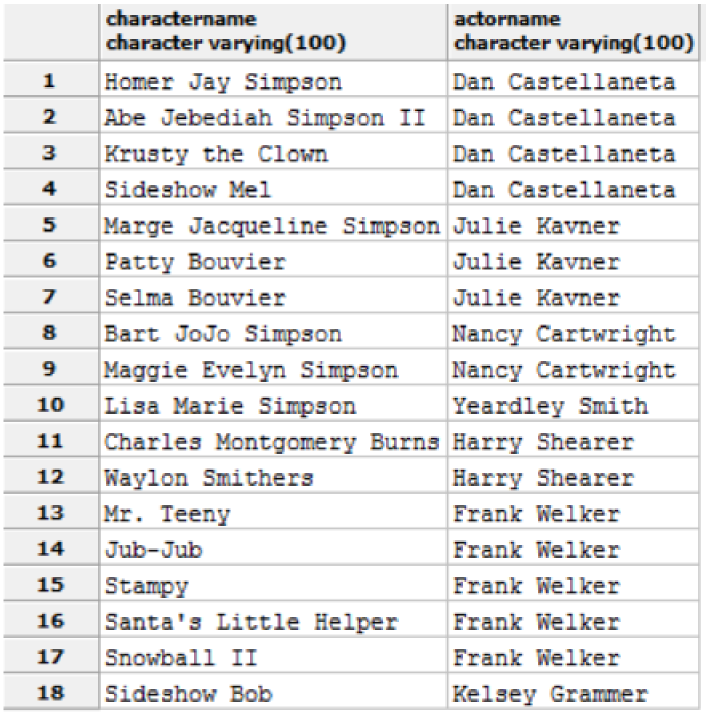
	aid integer	ownid integer
1	10	1
2	11	4
3	17	12
4	18	9
5	19	4

Views

whichVoice

This is a view meant to pair characters with their respective voice actors.

```
CREATE OR REPLACE VIEW whichVoice AS

SELECT c.characterName, va.actorName

FROM characters c

INNER JOIN voiceCharacter vc

ON vc.cid=c.cid

INNER JOIN voiceActors va

ON vc.vid = va.vid;
```



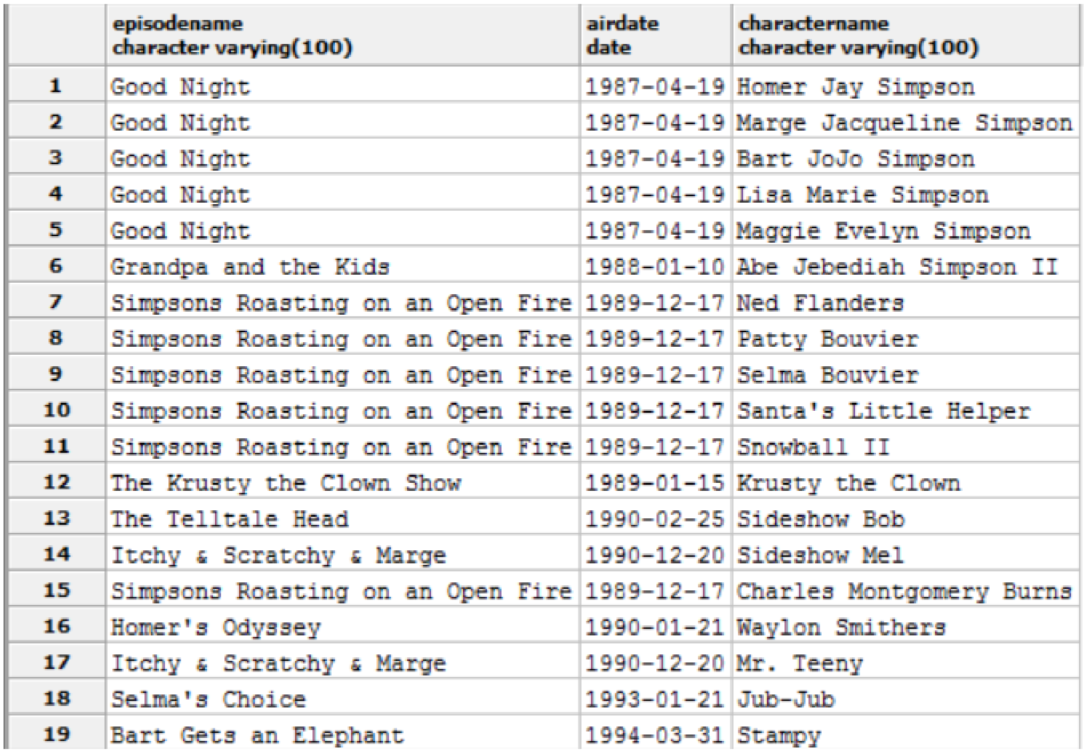
	charactername character varying(100)	actorname character varying(100)
1	Homer Jay Simpson	Dan Castellaneta
2	Abe Jebediah Simpson II	Dan Castellaneta
3	Krusty the Clown	Dan Castellaneta
4	Sideshow Mel	Dan Castellaneta
5	Marge Jacqueline Simpson	Julie Kavner
6	Patty Bouvier	Julie Kavner
7	Selma Bouvier	Julie Kavner
8	Bart JoJo Simpson	Nancy Cartwright
9	Maggie Evelyn Simpson	Nancy Cartwright
10	Lisa Marie Simpson	Yeardley Smith
11	Charles Montgomery Burns	Harry Shearer
12	Waylon Smithers	Harry Shearer
13	Mr. Teeny	Frank Welker
14	Jub-Jub	Frank Welker
15	Stampy	Frank Welker
16	Santa's Little Helper	Frank Welker
17	Snowball II	Frank Welker
18	Sideshow Bob	Kelsey Grammer

whichEpisode

This serves mostly the same purpose as the other view, but shows the characters with the episode that they first appeared in.

```
CREATE OR REPLACE VIEW whichEpisode AS
```

```
SELECT e.episodeName, e.airDate, c.characterName

FROM episodes e

INNER JOIN characterEpisode ce

ON e.eid = ce.eid

INNER JOIN characters c

ON ce.cid = c.cid;
```

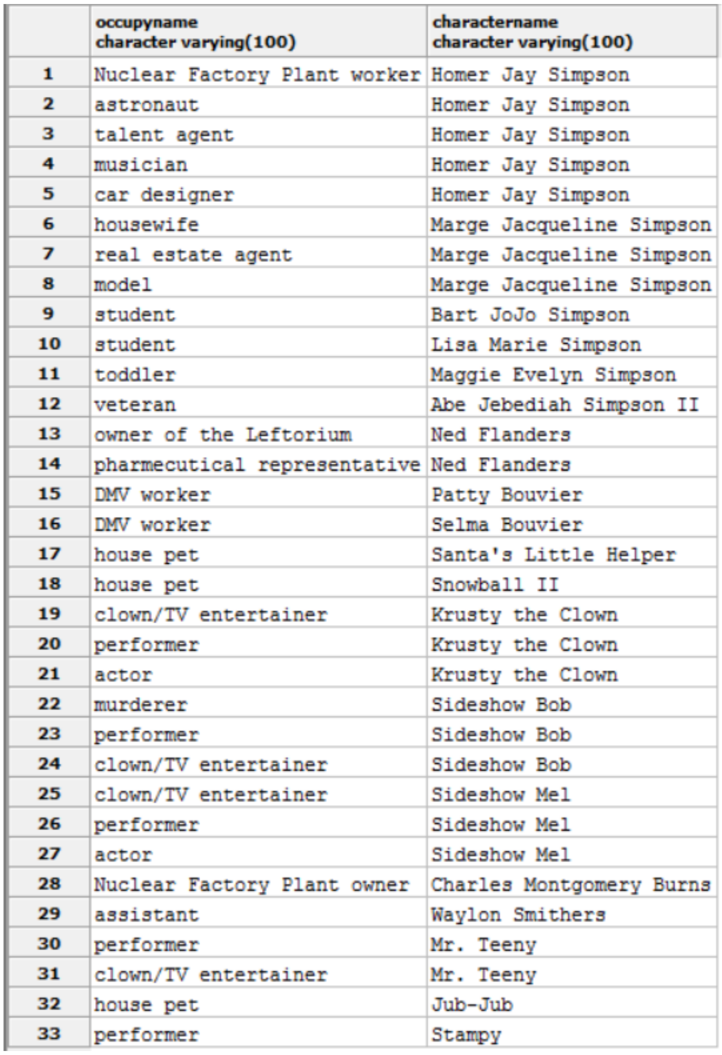


	episodename character varying(100)	airdate date	charactername character varying(100)
1	Good Night	1987-04-19	Homer Jay Simpson
2	Good Night	1987-04-19	Marge Jacqueline Simpson
3	Good Night	1987-04-19	Bart JoJo Simpson
4	Good Night	1987-04-19	Lisa Marie Simpson
5	Good Night	1987-04-19	Maggie Evelyn Simpson
6	Grandpa and the Kids	1988-01-10	Abe Jebediah Simpson II
7	Simpsons Roasting on an Open Fire	1989-12-17	Ned Flanders
8	Simpsons Roasting on an Open Fire	1989-12-17	Patty Bouvier
9	Simpsons Roasting on an Open Fire	1989-12-17	Selma Bouvier
10	Simpsons Roasting on an Open Fire	1989-12-17	Santa's Little Helper
11	Simpsons Roasting on an Open Fire	1989-12-17	Snowball II
12	The Krusty the Clown Show	1989-01-15	Krusty the Clown
13	The Telltale Head	1990-02-25	Sideshow Bob
14	Itchy & Scratchy & Marge	1990-12-20	Sideshow Mel
15	Simpsons Roasting on an Open Fire	1989-12-17	Charles Montgomery Burns
16	Homer's Odyssey	1990-01-21	Waylon Smithers
17	Itchy & Scratchy & Marge	1990-12-20	Mr. Teeny
18	Selma's Choice	1993-01-21	Jub-Jub
19	Bart Gets an Elephant	1994-03-31	Stampy

Reports

This report shows the jobs that each character has.

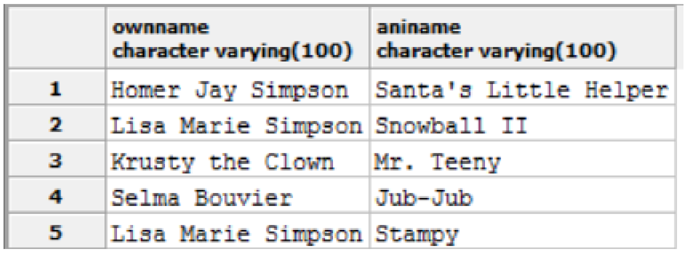
```
SELECT o.occupyName, c.characterName

FROM occupations o

INNER JOIN jobs j

ON o.oid = j.oid

INNER JOIN characters c

ON j.cid = c.cid;
```



	occupyname character varying(100)	charactername character varying(100)
1	Nuclear Factory Plant worker	Homer Jay Simpson
2	astronaut	Homer Jay Simpson
3	talent agent	Homer Jay Simpson
4	musician	Homer Jay Simpson
5	car designer	Homer Jay Simpson
6	housewife	Marge Jacqueline Simpson
7	real estate agent	Marge Jacqueline Simpson
8	model	Marge Jacqueline Simpson
9	student	Bart JoJo Simpson
10	student	Lisa Marie Simpson
11	toddler	Maggie Evelyn Simpson
12	veteran	Abe Jebediah Simpson II
13	owner of the Leftorium	Ned Flanders
14	pharmecutical representative	Ned Flanders
15	DMV worker	Patty Bouvier
16	DMV worker	Selma Bouvier
17	house pet	Santa's Little Helper
18	house pet	Snowball II
19	clown/TV entertainer	Krusty the Clown
20	performer	Krusty the Clown
21	actor	Krusty the Clown
22	murderer	Sideshow Bob
23	performer	Sideshow Bob
24	clown/TV entertainer	Sideshow Bob
25	clown/TV entertainer	Sideshow Mel
26	performer	Sideshow Mel
27	actor	Sideshow Mel
28	Nuclear Factory Plant owner	Charles Montgomery Burns
29	assistant	Waylon Smithers
30	performer	Mr. Teeny
31	clown/TV entertainer	Mr. Teeny
32	house pet	Jub-Jub
33	performer	Stampy

This report displays the character names of the owners with their respective pets.

```
SELECT o.ownName, a.aniName

FROM owners o

INNER JOIN animalOwn ao

ON o.ownid = ao.ownid

INNER JOIN animals a

ON ao.aid = a.aid;
```



	ownname character varying(100)	aniname character varying(100)
1	Homer Jay Simpson	Santa's Little Helper
2	Lisa Marie Simpson	Snowball II
3	Krusty the Clown	Mr. Teeny
4	Selma Bouvier	Jub-Jub
5	Lisa Marie Simpson	Stampy

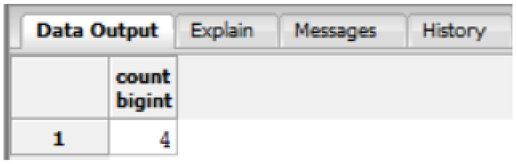
Stored Procedures

frequencyVoice

This stored procedure takes in the name of a voice actor, then outputs the number of characters they voice for the show.

```
CREATE OR REPLACE FUNCTION frequencyVoice(varchar(100), REFCURSOR)

RETURNS refcursor as $$

DECLARE

      actorNameInput varchar(100) := $1;

      resultset REFCURSOR := $2;

BEGIN

   open resultset for

      SELECT count(actorNameInput)

      FROM whichVoice

      WHERE actorNameInput = actorName;

   return resultset;

end;

$$

language plpgsql;


SELECT frequentVoice('Dan Castellaneta', 'results');

FETCH ALL FROM results;
```

The return value for this is four characters, based on the sample data inputted.

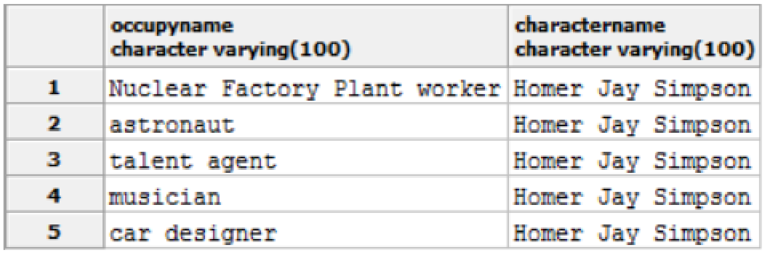


	count bigint
1	4

jobschar

This stored procedure displays all of the jobs that an inputted character has had.

```
CREATE OR REPLACE FUNCTION jobschar(varchar(100), REFCURSOR)

RETURNS refcursor as $$

DECLARE

      characterInput varchar(100) := $1;

      resultset REFCURSOR := $2;

BEGIN

  open resultset for

      SELECT o.occupyName, c.characterName

      FROM occupations o

      INNER JOIN jobs j

      ON o.oid = j.oid

      INNER JOIN characters c

      ON j.cid = c.cid

           where characterInput = c.characterName;

  return resultset;

end;

$$

language plpgsql;

SELECT jobschar('Homer Jay Simpson', 'results');

FETCH ALL FROM results;
```



	occupyname character varying(100)	charactername character varying(100)
1	Nuclear Factory Plant worker	Homer Jay Simpson
2	astronaut	Homer Jay Simpson
3	talent agent	Homer Jay Simpson
4	musician	Homer Jay Simpson
5	car designer	Homer Jay Simpson

addjobs()

This stored procedure updates the jobs linking table if a new character or occupation is added.

```
CREATE OR REPLACE FUNCTION addjobs()

RETURNS TRIGGER AS $body$

BEGIN

      INSERT INTO jobs (cid, oid)

            values(new.cid, new.oid);

END

$body$

language plpgsql;
```

Triggers

addOccupation

This trigger updates on the occupation table, by adding any data inserted into the table into the jobs table.

```
CREATE TRIGGER addOccupation

AFTER INSERT ON occupations

EXECUTE PROCEDURE addjobs();
```

addCharacter

This trigger updates on the character table alternatively. There must be two different triggers because there are two separate updates on two separate tables.

```
CREATE TRIGGER addCharacter

AFTER INSERT ON characters

EXECUTE PROCEDURE addjobs();
```

Security

The only users of this database would be the administrators and the public (user). The user role includes the writers of the show and other viewers who wish to access information. The admin naturally are granted all features because they must upkeep the database. The users are only given a SELECT permission so they can view but not alter any information within the database.

```
CREATE ROLE admin;

GRANT ALL ON ALL TABLES

IN SCHEMA PUBLIC

TO admin;


CREATE ROLE user

GRANT SELECT

ON ALL TABLES IN SCHEMA PUBLIC

TO user;
```

Notes, Problems and Enhancements

The implementation of the database was fairly easy, considering it was tracking on characters and a few other things associated with them, like voice actors and episodes. I chose to include less data to make the database more manageable, otherwise the database would be too big to display here, considering how many characters are in the first nine seasons. The problems I ran into were mostly trying to display certain kinds of information through views and queries, such as the table that displays owner names and animal names. I also encountered problems with the stored procedures and triggers, mostly with monitoring inserts on certain tables. If I could enhance the database more, I would attempt to add all of the data needed for the characters. I would hesitate from including seasons beyond season 10, for continuity reasons; most characters began to be altered beyond what would make sense for their character. There might also be redundant and/or contradicting data. Furthermore, it would be interesting to create a database of the entire town of Springfield, including stores, streets, and important town members. However, that database would be far beyond the scope allowed for this document.