THE COMPLETE DATABASE OF Heavy Metal

BY
CLAYTON ENRIGHT

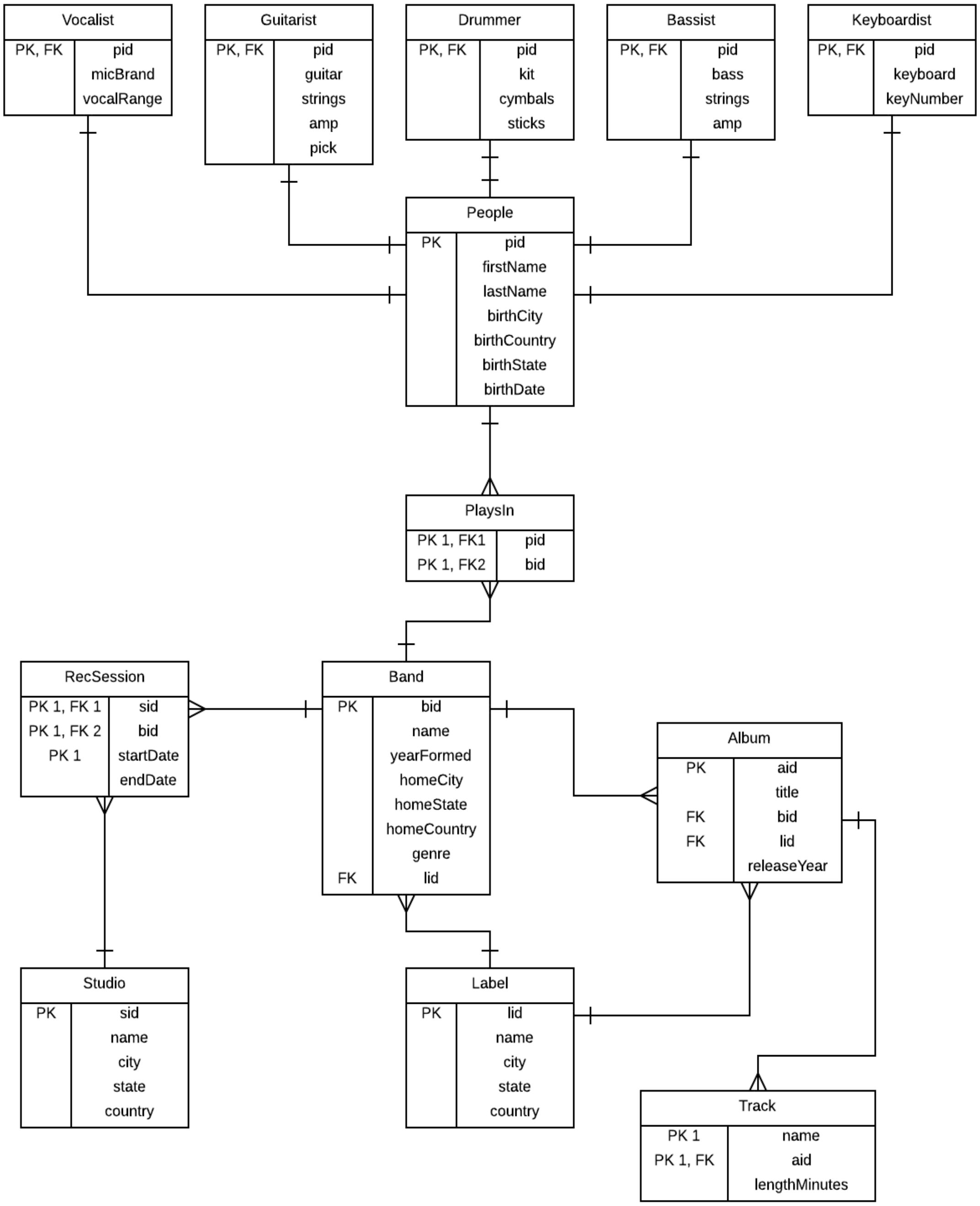
Table of Contents

Executive Summary:

The following database is designed to keep track of the ever-evolving genre of metal music. From the humble beginnings of heavy metal in England, the genre expanded into a wide array of subgenres full of unique bands each with their own take on the style. Users will not only be able to look up individual bands, but also individual artists and cross reference them to all of the projects or bands with whom they have worked. As another example, a person could retrieve a list of every recorded band that is a part of some particular subgenre, such as thrash metal.

The ER Diagram is presented first and shows the relationship between each of the entities tracked by the database. Following this are the individual tables with functional dependencies, sample reports, sample views, stored procedures, and finally security permissions. The presentation concludes with notes on the implantation of the database, possible future improvements, known short comings, and some final thoughts.

ER Diagram:

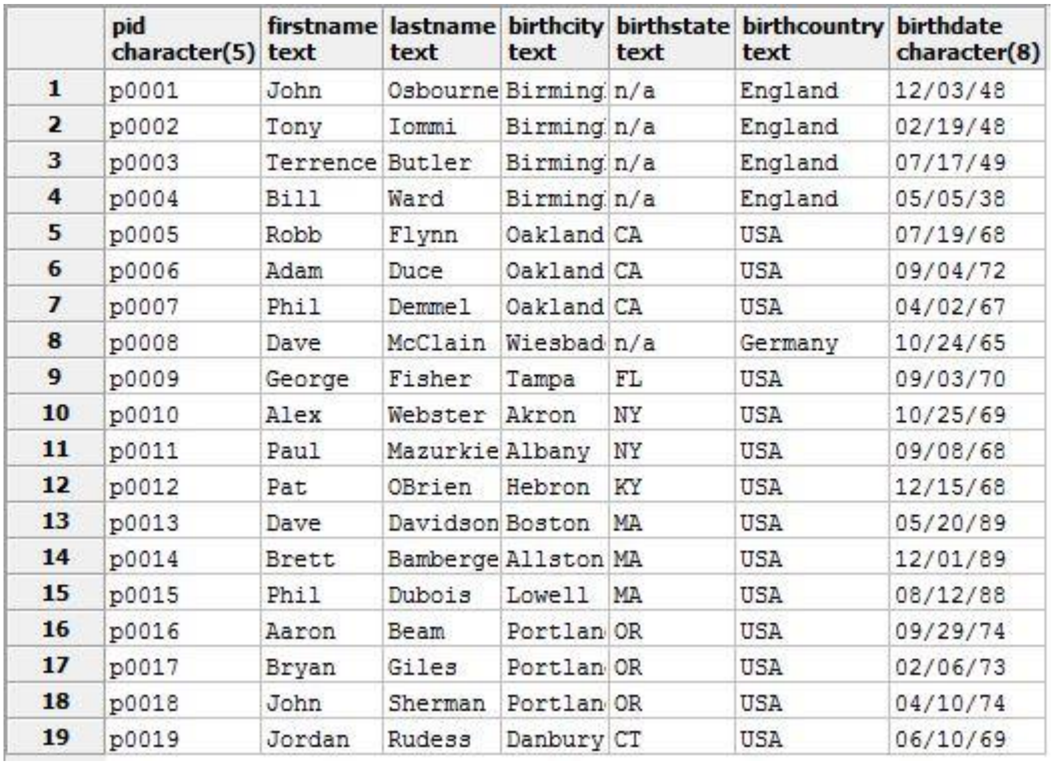

Vocalist
PK, FK
pid
micBrand
vocalRange
Guitarist
PK, FK
pid
guitar
strings
amp
pick
Drummer
PK, FK
pid
kit
cymbals
sticks
Bassist
PK, FK
pid
bass
strings
amp
Keyboardist
PK, FK
pid
keyboard
keyNumber
People
PK
pid
firstName
lastName
birthCity
birthCountry
birthState
birthDate
PlaysIn
PK 1, FK1
pid
PK 1, FK2
bid
RecSession
PK 1, FK 1
sid
PK 1, FK 2
bid
PK 1
startDate
endDate
Band
PK
bid
name
yearFormed
homeCity
homeState
homeCountry
genre
FK
lid
Album
PK
aid
title
FK
bid
FK
lid
releaseYear
Studio
PK
sid
name
city
state
country
Label
PK
lid
name
city
state
country
Track
PK 1
name
PK 1, FK
aid
lengthMinutes

People Table:

Contains a list of all the people involved in the bands.

```
CREATE TABLE people (
  pid          char(5) not null,
  firstName    text,
  lastName     text,
  birthCity    text,
  birthState   text,
  birthCountry text,
  birthDate    char(8),
 primary key(pid)
);
```

pid → firstName, lastName, birthCity, birthState, birthCountry, birthDate



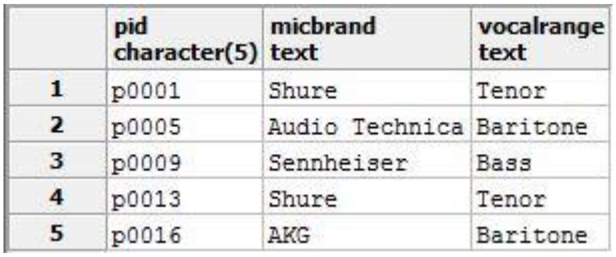
	pid character(5)	firstname text	lastname text	birthcity text	birthstate text	birthcountry text	birthdate character(8)
1	p0001	John	Osbourne	Birming	n/a	England	12/03/48
2	p0002	Tony	Iommi	Birming	n/a	England	02/19/48
3	p0003	Terrence	Butler	Birming	n/a	England	07/17/49
4	p0004	Bill	Ward	Birming	n/a	England	05/05/38
5	p0005	Robb	Flynn	Oakland	CA	USA	07/19/68
6	p0006	Adam	Duce	Oakland	CA	USA	09/04/72
7	p0007	Phil	Demmel	Oakland	CA	USA	04/02/67
8	p0008	Dave	McClain	Wiesbad	n/a	Germany	10/24/65
9	p0009	George	Fisher	Tampa	FL	USA	09/03/70
10	p0010	Alex	Webster	Akron	NY	USA	10/25/69
11	p0011	Paul	Mazurkie	Albany	NY	USA	09/08/68
12	p0012	Pat	OBrien	Hebron	KY	USA	12/15/68
13	p0013	Dave	Davidson	Boston	MA	USA	05/20/89
14	p0014	Brett	Bamberge	Allston	MA	USA	12/01/89
15	p0015	Phil	Dubois	Lowell	MA	USA	08/12/88
16	p0016	Aaron	Beam	Portlan	OR	USA	09/29/74
17	p0017	Bryan	Giles	Portlan	OR	USA	02/06/73
18	p0018	John	Sherman	Portlan	OR	USA	04/10/74
19	p0019	Jordan	Rudess	Danbury	CT	USA	06/10/69

Vocalist Table:

Contains all of the people who perform vocals for a band and their equipment choices.

```
CREATE TABLE vocalist (
  pid         char(5) not null references people(pid),
  micBrand    text,
  vocalRange  text,
 primary key(pid)
);
```

pid → micBrand, vocalRange



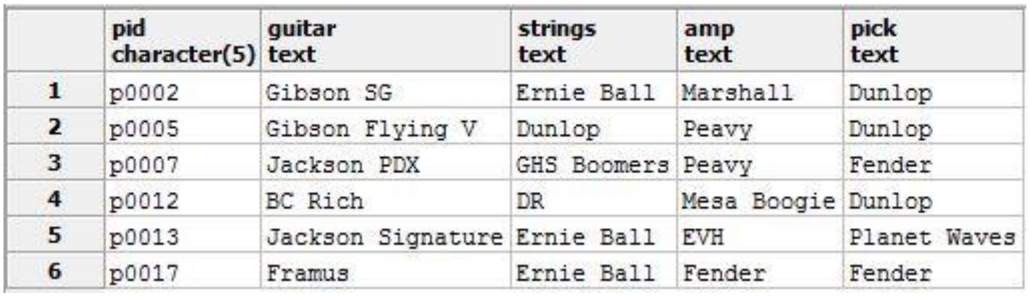
	pid character(5)	micbrand text	vocalrange text
1	p0001	Shure	Tenor
2	p0005	Audio Technica	Baritone
3	p0009	Sennheiser	Bass
4	p0013	Shure	Tenor
5	p0016	AKG	Baritone

Guitarist Table:

Contains all of the people who play guitar for a band and their instruments.

```
CREATE TABLE guitarist (
  pid         char(5) not null references people(pid),
  guitar      text,
  strings     text,
  amp         text,
  pick        text,
 primary key(pid)
);
```

pid → guitar, strings, amp, pick



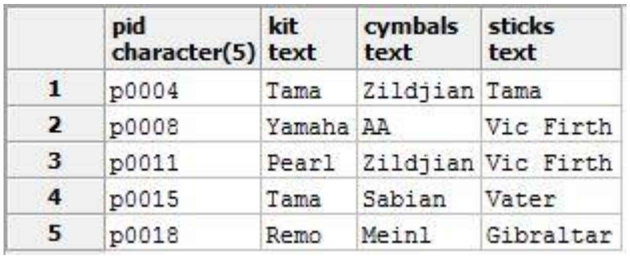
	pid character(5)	guitar text	strings text	amp text	pick text
1	p0002	Gibson SG	Ernie Ball	Marshall	Dunlop
2	p0005	Gibson Flying V	Dunlop	Peavy	Dunlop
3	p0007	Jackson PDX	GHS Boomers	Peavy	Fender
4	p0012	BC Rich	DR	Mesa Boogie	Dunlop
5	p0013	Jackson Signature	Ernie Ball	EVH	Planet Waves
6	p0017	Framus	Ernie Ball	Fender	Fender

Drummer Table:

Contains all of the people who play drums for a band and their kit choices.

```
CREATE TABLE drummer (
  pid         char(5) not null references people(pid),
  kit         text,
  cymbals     text,
  sticks      text,
 primary key(pid)
);
```

pid → kit, cymbals, sticks



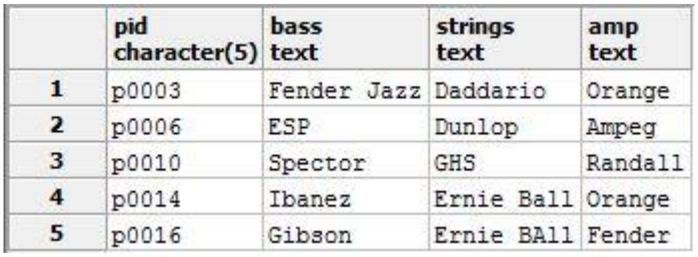
	pid character(5)	kit text	cymbals text	sticks text
1	p0004	Tama	Zildjian	Tama
2	p0008	Yamaha	AA	Vic Firth
3	p0011	Pearl	Zildjian	Vic Firth
4	p0015	Tama	Sabian	Vater
5	p0018	Remo	Meinl	Gibraltar

Bassist Table:

Contains all of the people who play bass for a band and their instruments.

```
CREATE TABLE bassist (
  pid         char(5) not null references people(pid),
  bass        text,
  strings     text,
  amp         text,
 primary key(pid)
);
```

pid → bass, strings, amp



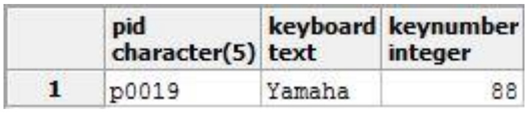
	pid character(5)	bass text	strings text	amp text
1	p0003	Fender Jazz	Daddario	Orange
2	p0006	ESP	Dunlop	Ampeg
3	p0010	Spector	GHS	Randall
4	p0014	Ibanez	Ernie Ball	Orange
5	p0016	Gibson	Ernie BAll	Fender

Keyboardist Table:

Contains all of the people who play the keyboard for a band and their equipment.

```
CREATE TABLE keyboardist (
  pid          char(5) not null references people(pid),
  keyboard     text,
  keyNumber    integer,
 primary key(pid)
);
```

pid → keyboard, keyNumber



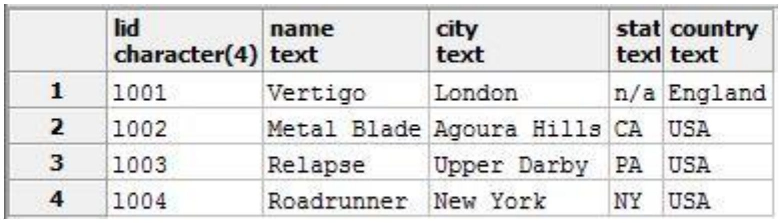
	pid	keyboard	keynumber
	character(5)	text	integer
1	p0019	Yamaha	88

Label Table:

Contains all of the record labels that put out metal music.

```
CREATE TABLE label (
  lid          char(4) not null,
  name         text,
  city         text,
  state        text,
  country      text,
 primary key(lid)
);
```

lid → name, city, state, country



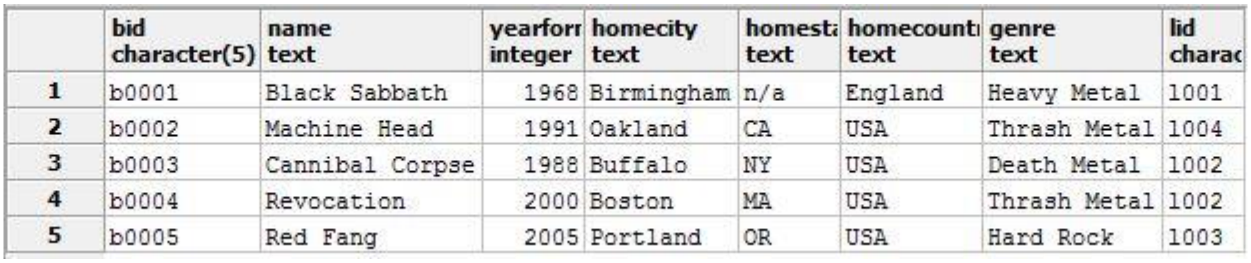
	lid	name	city	stat	country
	character(4)	text	text	text	text
1	1001	Vertigo	London	n/a	England
2	1002	Metal Blade	Agoura Hills	CA	USA
3	1003	Relapse	Upper Darby	PA	USA
4	1004	Roadrunner	New York	NY	USA

Band Table:

Contains the list of all the bands that are classified in some subgenre of metal.

```
CREATE TABLE band (
  bid         char(5) not null,
  name        text,
  yearFormed  integer,
  homeCity    text,
  homeState   text,
  homeCountry text,
  genre       text,
  lid         char(4) not null references label(lid),
 primary key(bid)
);
```

bid → name, yearFormed, homeCity, homeState, homeCountry, genre, lid



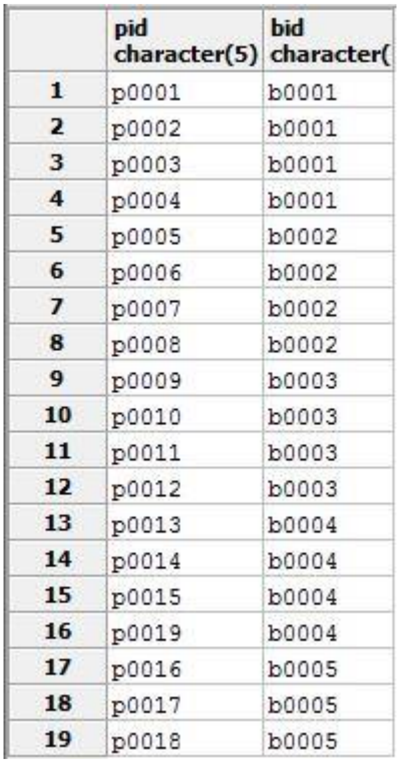
	bid character(5)	name text	yearforr integer	homecity text	homesta text	homecounti text	genre text	lid charac
1	b0001	Black Sabbath	1968	Birmingham	n/a	England	Heavy Metal	1001
2	b0002	Machine Head	1991	Oakland	CA	USA	Thrash Metal	1004
3	b0003	Cannibal Corpse	1988	Buffalo	NY	USA	Death Metal	1002
4	b0004	Revocation	2000	Boston	MA	USA	Thrash Metal	1002
5	b0005	Red Fang	2005	Portland	OR	USA	Hard Rock	1003

PlaysIn Table:

This table relates each person to a band, or multiple bands.

```
CREATE TABLE playsin (
  pid         char(5) not null references people(pid),
  bid         char(5) not null references band(bid),
 primary key(pid, bid)
);
```

pid, bid →



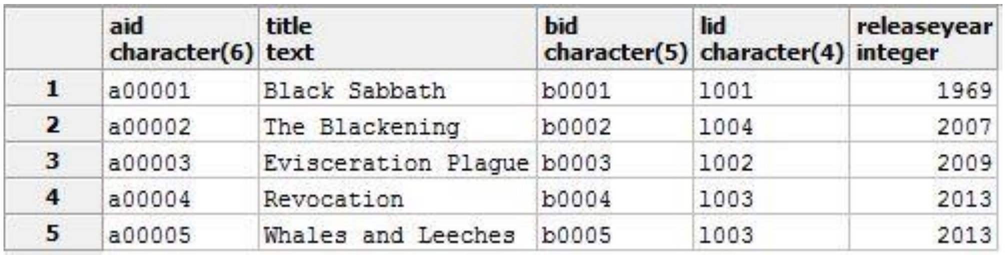
	pid character(5)	bid character(
1	p0001	b0001
2	p0002	b0001
3	p0003	b0001
4	p0004	b0001
5	p0005	b0002
6	p0006	b0002
7	p0007	b0002
8	p0008	b0002
9	p0009	b0003
10	p0010	b0003
11	p0011	b0003
12	p0012	b0003
13	p0013	b0004
14	p0014	b0004
15	p0015	b0004
16	p0019	b0004
17	p0016	b0005
18	p0017	b0005
19	p0018	b0005

Album Table:

Contains all of the albums released by any metal band and release details.

```
CREATE TABLE album (
  aid         char(6) not null,
  title       text,
  bid         char(5) not null references band(bid),
  lid         char(4) not null references label(lid),
  releaseYear integer,
 primary key(aid)
);
```

aid → title, bid, lid, releaseYear



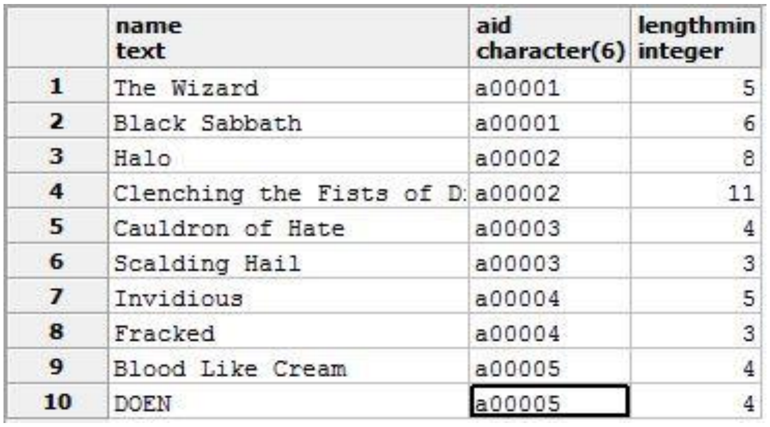
	aid character(6)	title text	bid character(5)	lid character(4)	releaseyear integer
1	a00001	Black Sabbath	b0001	1001	1969
2	a00002	The Blackening	b0002	1004	2007
3	a00003	Evisceration Plague	b0003	1002	2009
4	a00004	Revocation	b0004	1003	2013
5	a00005	Whales and Leeches	b0005	1003	2013

Track Table:

Contains all of the tracks appearing on the albums with song length in minutes.

```
CREATE TABLE track (
  name        text not null,
  aid         char(6) not null references album(aid),
  lengthMin   integer,
 primary key(name, aid)
);
```

name, aid → lengthMin



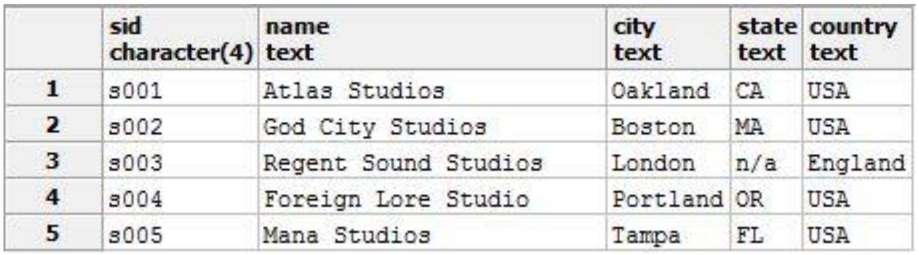
	name text	aid character(6)	lengthmin integer
1	The Wizard	a00001	5
2	Black Sabbath	a00001	6
3	Halo	a00002	8
4	Clenching the Fists of D	a00002	11
5	Cauldron of Hate	a00003	4
6	Scalding Hail	a00003	3
7	Invidious	a00004	5
8	Fracked	a00004	3
9	Blood Like Cream	a00005	4
10	DOEN	a00005	4

Studio Table:

Contains the studios that recorded each album.

```
CREATE TABLE studio (
  sid         char(4) not null,
  name        text,
  city        text,
  state       text,
  country     text,
 primary key(sid)
);
```

sid → name, city, state, country



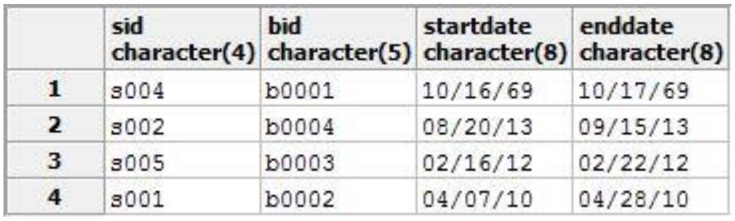
	sid character(4)	name text	city text	state text	country text
1	s001	Atlas Studios	Oakland	CA	USA
2	s002	God City Studios	Boston	MA	USA
3	s003	Regent Sound Studios	London	n/a	England
4	s004	Foreign Lore Studio	Portland	OR	USA
5	s005	Mana Studios	Tampa	FL	USA

Recording Session Table:

Details the start date, end date, and location of each recording session.

```
CREATE TABLE recsession (
  sid        char(4) not null references studio(sid),
  bid        char(5) not null references band(bid),
  startDate  char(8) not null,
  endDate    char(8),
 primary key (sid, bid, startDate)
);
```

sid, bid, startDate → endDate



	sid
	character(4)
		bid
		character(5)
			startdate
			character(8)
				enddate
				character(8)

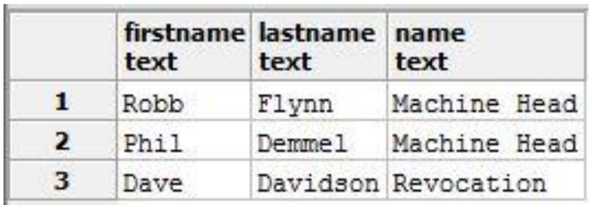
1	s004	b0001	10/16/69	10/17/69
2	s002	b0004	08/20/13	09/15/13
3	s005	b0003	02/16/12	02/22/12
4	s001	b0002	04/07/10	04/28/10

View Statements:

ThrashGuitarists View

Shows all of the guitarists who play in Thrash Metal bands.

```
CREATE VIEW ThrashGuitarists
AS
SELECT p.firstname, p.lastname, b.name
FROM band b, playsin i, people p, guitarist g
WHERE b.bid = i.bid
  AND p.pid = i.pid
  AND g.pid = p.pid
  AND b.genre = 'Thrash Metal'
```

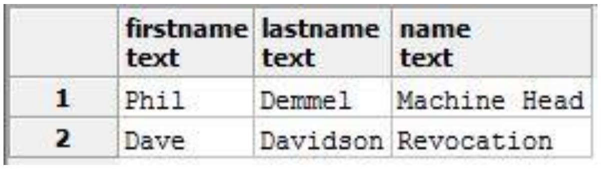


	firstname text	lastname text	name text
1	Robb	Flynn	Machine Head
2	Phil	Demmel	Machine Head
3	Dave	Davidson	Revocation

JacksonGuitarists View

Shows all of the guitarists who prefer Jackson guitars.

```
CREATE VIEW JacksonGuitarists
AS
SELECT p.firstname, p.lastname, b.name
FROM band b, playsin i, people p, guitarist g
WHERE b.bid = i.bid
  AND p.pid = i.pid
  AND g.pid = p.pid
  AND g.guitar LIKE 'Jackson%'
```



	firstname text	lastname text	name text
1	Phil	Demmel	Machine Head
2	Dave	Davidson	Revocation

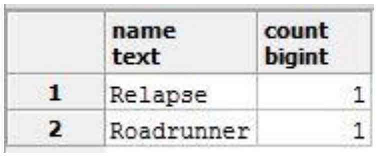
Sample Reports:

Top 2 Thrash Producing Labels

Shows the two labels that put out the most albums in the Thrash Metal genre.

```
select l.name, count(l.name)
from album a, label l, band b
where a.lid = l.lid
  and a.bid = b.bid
  and b.genre = 'Thrash Metal'
group by l.name
order by count(l.name) desc
limit 2
```

Example:



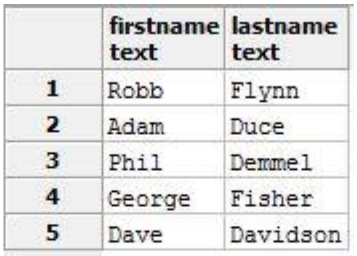
	name text	count bigint
1	Relapse	1
2	Roadrunner	1

Birth City, Rec City

Shows people who have recorded albums in the same city as they were born in.

```
select p.firstname, p.lastname
from people p, band b, playsin i, recsession r, studio s
where p.pid = i.pid
  and i.bid = b.bid
  and b.bid = r.bid
  and r.sid = s.sid
  and p.birthCity = s.city
```

Example:



	firstname text	lastname text
1	Robb	Flynn
2	Adam	Duce
3	Phil	Demmel
4	George	Fisher
5	Dave	Davidson

Stored Procedures

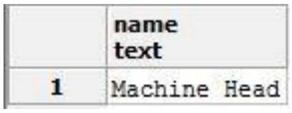
IsInBand()

Tells what bands a person belongs to.

```
create or replace function IsInBand(text, text, REFCURSOR)
returns refcursor as
$$
declare
   firstn     text       := $1;
   lastn      text       := $2;
   resultset  REFCURSOR  := $3;
begin
   open resultset for
      select b.name
      from   band b, people p, playsin i
       where  p.pid = i.pid
         and  b.bid = i.bid
         and  firstName = firstn
         and  lastName = lastn;
   return resultset;
end;
$$
language plpgsql;

select IsInBand('Robb', 'Flynn', 'results');
Fetch all from results;
```

Shows the band that Robb Flynn is a part of (currently only ‘Machine Head’):



	name text
1	Machine Head

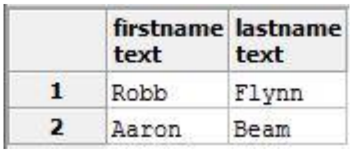
HasRange()

Tells the vocal range of a given singer.

```
create or replace function HasRange(text, REFCURSOR) returns
refcursor as
$$
declare
   range       text       := $1;
   resultset   REFCURSOR  := $2;
begin
   open resultset for
      select p.firstname, p.lastname
      from   people p, vocalist v
       where p.pid = v.pid
         and range = v.vocalRange;
   return resultset;
end;
$$
language plpgsql;

select HasRange('Baritone', 'results');
Fetch all from results;
```

Shows the singers whose voices are categorized as ‘Baritone’:



	firstname text	lastname text
1	Robb	Flynn
2	Aaron	Beam

Security:

Security privileges for this databases will be relatively straight-forward. There will need to be a role for admins to update and modify the data, and another role for public users to browse the data.

```
CREATE ROLE admin
CREATE ROLE publicuser


GRANT ALL PRIVILEGES
ON ALL TABLES IN SCHEMA PUBLIC
TO admin

GRANT SELECT
ON ALL TABLES IN SCHEMA PUBLIC
TO publicuser
```

Conclusion & Final Thoughts:

The implementation of this database for the most part was straight-forward. The biggest challenges involved relating bands to albums and tracks, and individual recording sessions to bands and studios. The only known problems with the current design are the relatively small amount of data and the over simplification of some concepts due to time constraints. These problems could certainly be corrected with future enhancements along with other improvements, such as the ability to track record sales and rank bands based on top selling albums. With some work, the database could even be expanded to include other rock genres as well.