

DJs create playlists. A playlist is 'one radio show'.

Playlists are comprised of tracks.

playlist	Format	Default
id	PRIMARY KEY int NOT NULL	'0'
user_id	KEY int	NULL
air_name	varchar(30)	NULL
start_time	KEY datetime	NULL
end_time	datetime	NULL

playlist_track	Format	Default
playlist_id	KEY int NOT NULL	'0'
indx	int	NULL
is_current	char(1)	NULL
artist	varchar(60)	NULL
track_title	varchar(60)	NULL
album_title	varchar(60)	NULL
album_id	KEY int	NULL
album_label	varchar(60)	NULL
time_played	datetime	NULL

Albums also contain tracks.

album	Format	Default
id	PRIMARY KEY int NOT NULL	'0'
artist	FULLTEXT KEY varchar(60)	NULL
title	varchar(60)	NULL
category	KEY char(1)	NULL
media	char(1)	NULL
format	char(1)	NULL
dummy1	char(1)	NULL
is_collection	char(1)	NULL
location	KEY char(1)	NULL
dummy2	char(1)	NULL
add_date	date	NULL
mod_date	date	NULL
label_id	int	NULL
mp3	smallint	NULL
reviewer_djid	KEY int unsigned NOT NULL	'0'
date_assigned	KEY date NOT NULL	'0000-00-00'
date_reviewed	date NOT NULL	'0000-00-00'

track	Format	Default
album_id	KEY int NOT NULL	'0'
title	varchar(60)	NULL
indx	int	NULL
clean	char(1) NOT NULL	'0'

When an album is a compilation, we denote the artist.

coll_track	Format	Default
album_id	KEY int NOT NULL	'0'
title	varchar(60)	NULL
artist	FULLTEXT KEY varchar(60)	NULL
indx	int	NULL
clean	char(1) NOT NULL	'0'

KFJC Station Data is a read-only resource.
The architecture and format are not my option to change.
This source is used solely for question creation.

I think it will take one SQL Alchemy Class for each database because of the different types.
Python will serve as the go-between for question formation and gameplay.



Each user may answer several questions

users	Format	Default
id	PRIMARY KEY int NOT NULL, AUTO INCREMENT	'0'
email	varchar(320). A unique identifier for each user. We will never email you.	NULL
fname	varchar(30). To address player personally. Not collecting last name for privacy reasons.	NULL
password	varchar. See https://blog.codinghorror.com/youre-probably-storing-passwords-incorrectly/ about storing passwords as a hash.	NULL
salt	varchar. "The point of a salt (or nonce, if you prefer) is to make each password unique and long enough that brute force attacks are a waste of time. So, the user's password, instead of being stored as the hash of "myspace1", ends up being stored as the hash of 128 characters of random unicode string + "myspace1". You're now completely immune to rainbow table attack."	NULL

user_answers	Format	Default
id	PRIMARY KEY int NOT NULL, AUTO INCREMENT	'0'
user_id	int	NULL
question_id	int	NULL
answer_given	text	NULL
timestamp	datetime	'0000-00-00'

Each answer was in response to a question.

questions	Format	Default
id	PRIMARY KEY int NOT NULL, AUTO INCREMENT	'0'
question	text	NULL

Each question has one or more acceptable answers
(Such as "Name a track from this album")

correct_answers	Format	Default
id	PRIMARY KEY int NOT NULL, AUTO INCREMENT	'0'
question_id	int	NULL
acceptable_answers	text[]	NULL



Derived Quantities

Derived Quantity	Pseudocode
User Score on one question	PASS: user_answers.answer_given IN correct_answers.acceptable_answers FAIL: user_answers.answer_given NOT IN correct_answers.acceptable_answers SKIP: user_answers.answer_given IS NULL
User Stats	user_percent_correct = SUM(PASS) * 100 / (SUM(PASS) + SUM(FAIL)) Players may skip a question without penalty.
User Rank	Max to min of user_percent_correct.
Member Since	The smallest timestamp in user_answers is our record of the first time the user played.