

# Database interface functions

\* type = [album or artist or track]

- get\_album\_favorites(user) -> returns multiple "album\_id"
- get\_artist\_favorites(user) -> returns multiple "artist\_id"
- get\_track\_favorites(user) -> returns multiple "track\_id"
- add\_favorite(User, item\_id, type) -> returns nothing
- get\_album\_reccomendations(user) -> return multiple "album\_id" with "recommended = false"
- get\_artist\_reccomendations(user) -> return multiple "artist\_id" with "recommended = false"
- get\_track\_reccomendations(user) -> return multiple "track\_id" with "recommended = false"
- add\_reccomendation(user, type, name, image, Optional description, Optional release\_date) -> returns nothing
- get\_name(item\_id, type) -> returns name
- get\_image(item\_id, type) -> returns image
- get\_description(item\_id, type) -> returns description
- get\_release\_date(item\_id, type) -> returns release\_date
- set\_recommended(item\_id, type) -> returns nothing

## Song Surf entity relationship diagram

Note: This database does not store the input searches of a user. It stores what's been recommended, what has been favorited, and what has been disliked

Note: As you may know, an album can have many artists, or a track can have many artists. Since last.fm returns only one artist (the main artist) for an album or a track, this diagram keeps the relationship between an album/track and artist as one to many. To solve the multiple artists issue, the track title has (feat. artistX, artistY)

