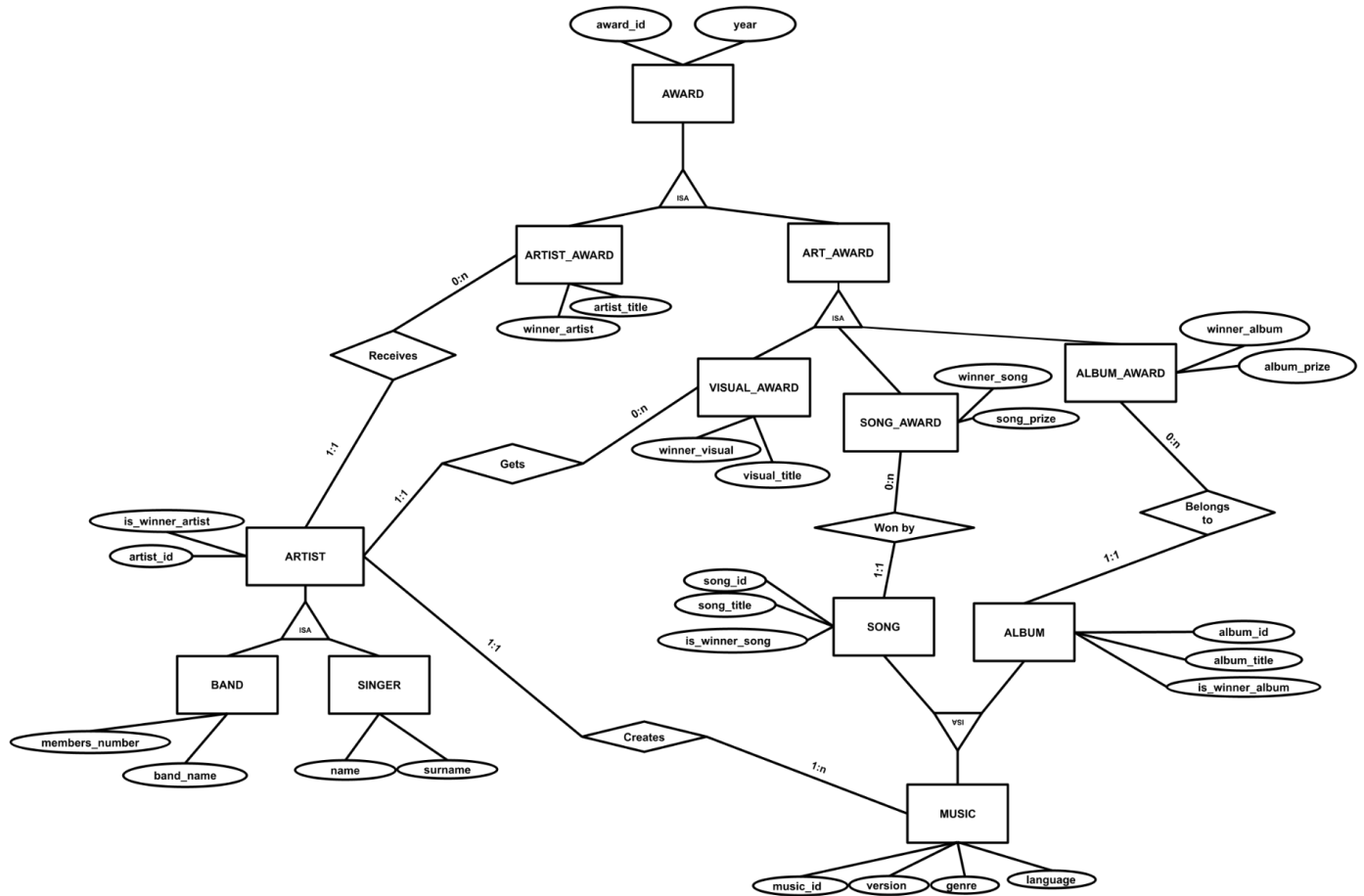


DBWS Homework 2 Group 11 - MAPPING APPROACH

We will describe the mapping approach of our slightly modified ER diagram shown below and in the *UpdatedER.pdf* file.

For all 3 sections of this document, we will make observations regarding possible alternatives, as well.

For the ISA hierarchies, superclasses that can be multiple subclasses are NOT allowed anywhere, meaning that all subclasses represent distinct entity sets that have no overlap with the other subclasses within the same superclass.



1) **AWARD ISA hierarchy and ART-AWARD ISA hierarchy:**

An AWARD entity can either be part of the ARTIST-AWARD subclass or ART-AWARD subclass. If part of the latter, it can either be classified as an ALBUM-AWARD, SONG-AWARD, or VISUAL-AWARD entity.

We considered that a char type value **award_id** attribute as a primary key (becoming also foreign key) for all entity sets within this hierarchy is more appropriate than an integer type value (even if this is an alternative). We decided this considering that there is a fixed number of "awards per year" and we would have to enter everything manually - therefore, for example, for an entity of the AWARD superclass, we can have *award_id* = 22SO12, where the first two digits represent the year (2022), SO represents the adherence to the SONG-AWARD subclass, and the other digits are to differentiate between elements of this subclass.

2) MUSIC ISA hierarchy:

A MUSIC entity can either be in the SONG or ALBUM subclass, but not in both.

Apart from the general attributes we chose for these sets of the ISA hierarchy, we have two boolean attributes **is_winner_song** and **is_winner_album**. They are TRUE for a certain entity when there exists a relationship WON_BY between that SONG entity object and some element of SONG-AWARD or, respectively, there is a relationship BELONGS_TO between that ALBUM entity and one of ALBUM-AWARD.

An alternative to this would have been creating a single relationship between MUSIC and ART-AWARD instead of the WON_BY and BELONGS_TO ones. Then, we could eliminate the boolean **is_winner_song** and **is_winner_album** and have a boolean attribute **is_winner_music** for the MUSIC superclass instead. However, when a user searches for a specific album or song / for a specific song award or album award, we would have to trace back to the superclass attributes. Even if the query might be easier this way, we consider that for us it is better to look at the more specific relationship sets between the subclasses.

3) ARTIST ISA hierarchy:

An artist can be either a SINGER or a BAND, but not both. The attributes of an ARTIST are stored in an automatically increasing integer value **artist_id**, which is unique, and a boolean value **is_winner_artist** which is TRUE when there exists one of the following:

- a relation RECEIVES between that ARTIST entity and an ARTIST-AWARD entity
- a relation GETS between that ARTIST entity and a VISUAL-AWARD entity
- a relation CREATES between that ARTIST entity and a MUSIC entity, either classified in the SONG subclass or ALBUM subclass, that has the boolean **is_winner_song** or **is_winner_album** as TRUE (here, an alternative would have been the one described for the AWARD ISA hierarchy - the one containing the **is_winner_music** attribute and a different relationship)

It is noteworthy to say that *"an artist can win no awards, one award, or many awards"* and that *"an award is won by exactly one artist"* - therefore the relationship sets defined above are 0:n on one end and 1:1 on the other end. The observation holds for the relationships between MUSIC and ARTIST, too, since *"an artist can create more songs or albums, but a musical piece is created by exactly one artist"*.