

## Song

```
string _tid, _MSD_artist_name, _MSD_title, _mXm_tid, _mXm_artist_name,  
_mXm_title;
```

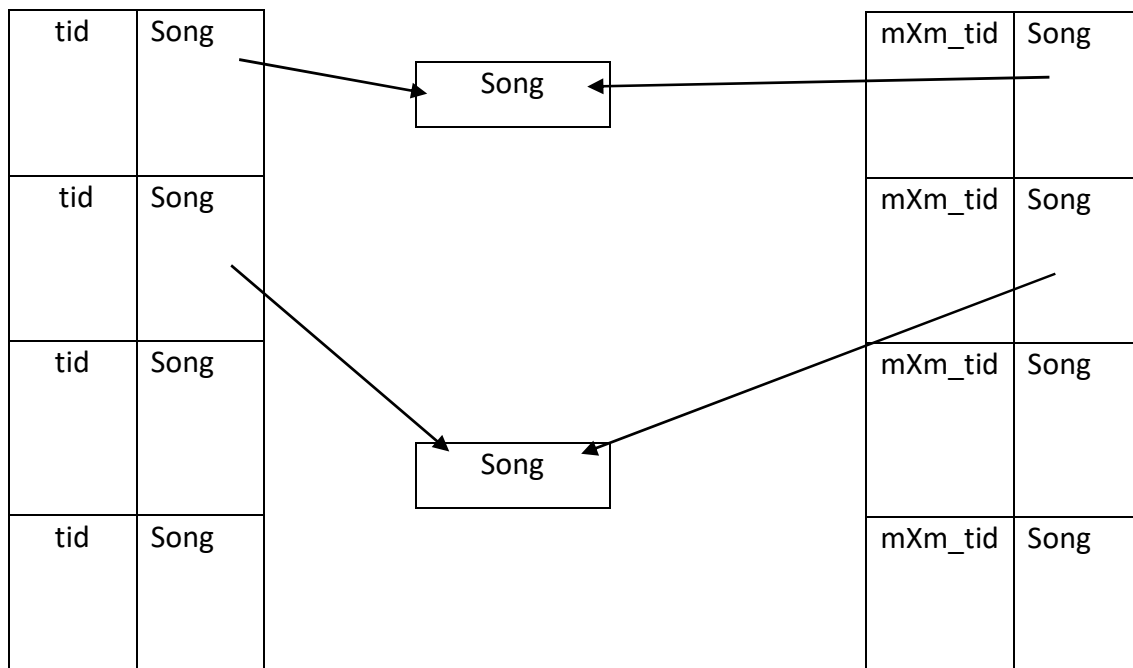
```
map<ushort, ushort> _mWords;
```

idParaula	idParaula	idParaula	idParaula	idParaula
contador	contador	contador	contador	contador

En aquest cas, s'utilitza un map degut a que en variis casos se'ns demana buscar una paraula pel seu identificador lo qual fa que el map sigui l'estructura més eficient. A més no admet repeticions.

## Songs

```
map<string, shared_ptr<Song>> _tidSong | map<string, shared_ptr<Song>> _mXmSong
```



Aquí es fa aquest dos map amb `shared_ptr` perquè cada cançó pot ser identificada per dos identificadors diferents, llavors el propòsit del map és semblant al del de la classe `Song`, i l'objectiu del `shared_ptr` és 'relacionar' una `Song` amb dos identificadors.

`vector<string> _lWords;`

Paraula	Paraula	Paraula	Paraula	Paraula	Paraula
---------	---------	---------	---------	---------	---------

En aquest cas, faig servir un vector ja que l'identificador de la paraula queda guardat com a la posició de la paraula en el vector. El fet de ser un vector és perquè és útil al agafar dades i accedir a aquestes.

En el main utilitzo alguna llista la qual serveix per guardar les dades extretes d'una funció i utilitzar aquestes dades que prèviament ja estaven ordenades.