Unnormalised Data

`member_movies` table

member (KEY)	fav_genre	username	movies	movie_years
tomk@example.com	Horror	tommyk	The Shining	1980
johnd@example.com	Romance	themovieguy	Titanic	1997
janed@example.com	Comedy	iloveminions	Titanic, Minions	1997, 2015
maryw@example.com	Horror	horrorfan1990	The Shining, IT	1980, 2017

1st Normal Form

`movies` and and `movie_years` are mutli-value so we must split them out

`member` previously could uniquely identify a record but it now cannot. We now effectively have a composite key consisting of `member + movie` which together can identify a row.

`member_movies` table

member (KEY)	fav_genre	username	movie (KEY)	movie_year
tomk@example.com	Horror	tommyk	The Shining	1980
johnd@example.com	Romance	themovieguy	Titanic	1997
janed@example.com	Romance	iloveminions	Titanic	1997
janed@example.com	Comedy	iloveminions	Minions	2015
maryw@example.com	Horror	horrorfan1990	The Shining	1980
maryw@example.com	Horror	horrorfan1990	IT	2017

2nd Normal Form

We have parts of the table that only depend partially on the key.

`fav_genre` and `username` depend only on the `member` part of the key, so we must split them out to a new entity table to represent a site member.

We will also replace the email with a new numeric primary key `member_id`

`movie_year` depends only on the `movie` part of the key, so we must also split them out to a new entity table to represent a movie.

We will also replace the `movie` name with a new numeric primary key `movie_id`

`member_movies` Table | member_id | movie_id | |----------| | 1 | 1 | | 2 | 2 | | 3 | 2 | | 3 | 3 | | 4 | 1 | | 4 | 4 |

`movies` Table | movie_id | movie_name | movie_year | |----------|----------| | 1 | The Shining | 1980 | | 2 | Titanic | 1997 | | 3 | Minions | 2015 | | 4 | IT | 2017 |

`members` Table | member_id | email | fav_genre | username | |----------|----------|----------| | 1 | tomk@example.com | Horror | tommyk | | 2 | johnd@example.com | Romance | themovieguy | | 3 | janed@example.com | Comedy | iloveminions | | 4 | maryw@example.com | Horror | horrorfan1990 |

3rd Normal Form

There are no transitive dependencies, so strictly there is nothing to do! 😃

If we wanted to normalise further to reduce `redundacy` (duplication) we might choose to extract `genre` to its own table. This allows the name of a genre to be specified only once, and if it needs to be updated this can be done in one single place only.

Again in this case we will give the new `genre` table a numeric primary key

`member_movies` Table | member_id | movie_id | |----------| | 1 | 1 | | 2 | 2 | | 3 | 2 | | 3 | 3 | | 4 | 1 | | 4 | 4 |

`movies` Table | movie_id | movie_name | movie_year | |----------|----------| | 1 | The Shining | 1980 | | 2 | Titanic | 1997 | | 3 | Minions | 2015 | | 4 | IT | 2017 |

`members` Table | member_id | email | fav_genre_id | username | |----------|----------|----------| | 1 | tomk@example.com | 3 | tommyk | | 2 | johnd@example.com | 2 | themovieguy | | 3 | janed@example.com | 1 | iloveminions | | 4 | maryw@example.com | 3 | horrorfan1990 |

`genres` Table | genre_id | genre_name | |----------| | 1 | Comedy | | 2 | Romance | | 3 | Horror |