Subject:

Year : Month : Day : ( ) page : ( )

Step 7: convert regular entities. composite attrs won't be written (only their components). Don't include multivalued attrs.

User:

user-id	user_Name	age	email	password

*subscribtion:

sub-id	start-date	end-date	1 monthplan	12 month plan	user id	paymut id
					FK	FK

*payment_history:

paymut-id	date	amout

*Commet:

commet-id	Commet	media-id	user_id
		FK	FK

*
user_rate:

rate-id	rate	user-id	media-id
		FK	FK

watch_later_list:

watchlist-id	media-id	user-id
	FK	FK

Subject:

Year : Month : Day : ( ) page : ( )

* Compay:

comp-id	comp-name	establish-year	comp-email	comp-phone

* Storage-loc:

loc-id	path	server

*media:

media-id	genre	Titele	rate	director	locid	comp-id
					Fk	Fk

*movie:

movie-media-id	production-year	run-Time

→ map sup/subtype Rel

*series:

Series-media-id	number of seasons	start-date	end-date	Number of episods

* mapping sup/subtype rel is in last step but i decided to do it here because it was necessary.

Step2: map multivalued attrs

*Actor:

media-id	actor-full-name
Fk	

Subject:

Year : Month : Day : ( ) page : ( )

*Producer:

media-id	producer-full-name
PK	

step3: map weak entities:

*episode:

step4: Binary 1-M : modify the table on the M side

1-M Rels: ↳ add the pk of 1side as FK to Mside

① user, sub

② user, user-rate

③ user, comment

④ media, comment

⑤ series, episode

⑥ company, media

⑦ media, user-rate

Subject:

Year : Month : Day : ( ) page : ( )

① sub table already has the user_id as fk so I won't modify that

② user_rate " " " " " " " "

③ comment " " " " " " " "

④, ⑦ : comment & user_rate already have media_id as fk.

⑤ episode already has series_media_id as fk

⑥ media already has the comp_id as fk.

step 5 : Binary M:N : create new table & the pk of M & N will be the fk of new table. Add the rel attr as attr to this new table. ↳ these together are pk for new table,

M:N Rels :

① watch_later_list , media

② user , media

+

① Is_Inside :

watchlist-id	media-id
fk	fk

↳ together => pk

Subject:

Year : Month : Day : ( ) page : ( )

② * watch :

user-id	media-id
Fk	Fk

step 6 : map binary one-one : modify the optional side
the attrs of the rel must be included in optional side.
the PK of mandatory side is included as fk in optional side :

1-1 Rels :

① user (optional), watch_later_list : Here due to specific circumstances I must modify both ⇒ but here it makes no sense so I continue on my earlier approach

* User :

user-id	user-name	age	email	password

* watchlaterlist :

watchlist-id	media-id	user-id
	Fk	Fk

The other 1-1 rels are media, storage_loc & sb, payment. history in which the fks are already defined correctly.

Subject:

Year : Month : Day : ( ) page : ( )

We're done here the final tables are defined by *

Subject:

Year : Month : Day : ( )

Normalization :

1NF:
- There are only single value Attrs
- Attrs dom won't change
- unique name for every cols
- The order in which data is stored doesn't matter
- pk is defined to uniquely define row in a Rel
- no duplicated rows or cols
- each col must have only one value for each row in table

So we now analyze these conditions on the tables that are signed with * ⟹ for simplicity I won't write the tables that are in 1NF & only write the ones which must convert to 1NF : I suppose that all the tables are in 1NF. There are some attrs that are suspicious to be multivalued or composite for which I'm going to indicate some assumptions to ensure that they're actually not multivalued or composite & won't violate 1NF.

The attrs that are about "date" exp: start_date , End_date, date,.... Since the dom & actually the datatype of these attrs are considered to be "Time" in DBMS so they're not composite and as a result they're atomic.

DateTime or Date

Subject:

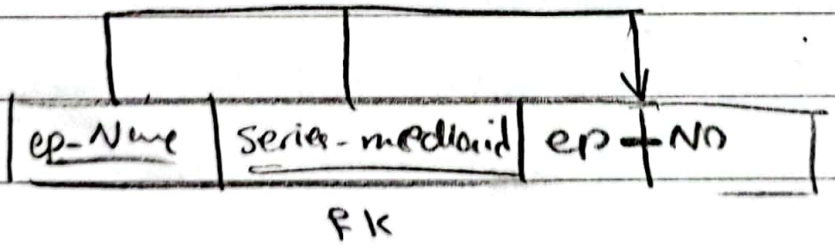
Year : Month : Day : ( ) page : ( )

In Actor & Producer tables the attrs Actor_Full_Name & Producer_Full_Name are still nd composite and they're atomic. The thing is that I necessarily don't need to seprate the 1st Name & last Names of Actors & producers so only their Full Name will do good in this case. And also in Company table company phone number is not multivalued since I assumed that each company only has 1 phone number.

2NF: must be in 1NF

- No partial functional dependencies.

of course in tables with only 1 pk there is no partial functional dependency we must only check those with more than 1 pk. In tables with 2 pks: episode:



it's in 2NF