Movie Production Database

Istanbul Bilgi University

Faculty of Engineering and Natural Sciences

Department of Computer Engineering

CMPE 351

Database Systems

Dr. Elena Battini Sönmez

Project Report

Linda Abdullah 119200025

Firm And Its Sector

The database is about a Movie Production firm where its sectors are movie, production center, cast members, and crew members. Movie is the movie produced and is directed in a production center. A movie may also have multiple genres. Cast members act in a movie with a specific role in that movie, also, each crew member has his/her own role in a production center. A cast member can act in several movies and has an appearance duration, in terms of minutes, and role in a movie.

Database Design Explanation

The entities which I used in this database are MOVIE(s), PRODUCTION_CENTER(s), CAST_MEMBER(s) and CREW_MEMBER(s).

The relationships which were used are "ACTS_IN", "WORKS_IN" and "PRODUCED_IN". Implicit implementation was applied for the entities and their relations.

The relational schema looks like the following:

MOVIE (<u>MID</u>, PID,Mname, MDuration)

PRODUCTION_CENTER (<u>PID</u>, Pname, Plocation, PTelephoneNumber)

CAST_MEMBER (<u>CastID</u>, Castname, CastSurname)

CREW_MEMBER (<u>CrewMID</u>, PID, CrewName, CrewSurname, CrewRole)

GENRE (<u>MID</u>, <u>Genre</u>)

ACTS_IN (<u>CastID</u>, <u>MID</u>, Appearance, Role)

The MOVIE(s) entity has a table which includes MID, Mname, MDuration and PID attributes. MID(int) is the unique identifier of the entity MOVIE and, due to one-to-many relationship between MOVIE and PRODUCTION_CENTER, PID(int) is the foreign key that refers to PRODUCTION_CENTER's attribute, PID, because a movie will be produced in one specific production center. Mname(varchar(20)) is the name of the movie and MDuration(int) is the duration of the movie, in terms of minutes. Each movie has a Genre attribute, but it is a multi-value attribute which means that it has many values because a movie may or may not have many genres. So in the relational schema, it is considered as a table of its own.

The GENRE table is created to give the multi-value functionality to MOVIE entity's attribute, Genre. It is a table with attributes MID (int) and Genre (varchar(20)). Where Genre attribute's value is limited to only 'action', 'comedy', 'romance', 'mystery', 'crime'. MID and Genre are the compound unique identifiers or compound primary keys of the GENRE table where genre attribute is the genre of a specific movie. MID attribute in GENRE table is specified as a foreign key which refers to MOVIE's MID attribute. The compound key functionality helps show the multi-valued functionality. Genre can act like a many-to-many relationship where one movie may have many genres and one genre can be for many movies.

The PRODUCTION_CENTER entity has a table whose attributes are PID, Pname, Plocation and PTelephoneNumber. PID (int) is the unique identifier for a production center’s record, Pname(varchar(20)) is its name, Plocation(varchar(20)) is its location, and PTelephoneNumber(int) is the telephone number belonging to that production center.

The CAST_MEMBER entity has table whose attributes are CastID, CastName, CastSurname. The CastID(int) is the unique identifier or primary key of cast member’s record, Castname(varchar(20)) is the first name of the cast member and CastSurname(varchar(20)) is the surname of the cast member.

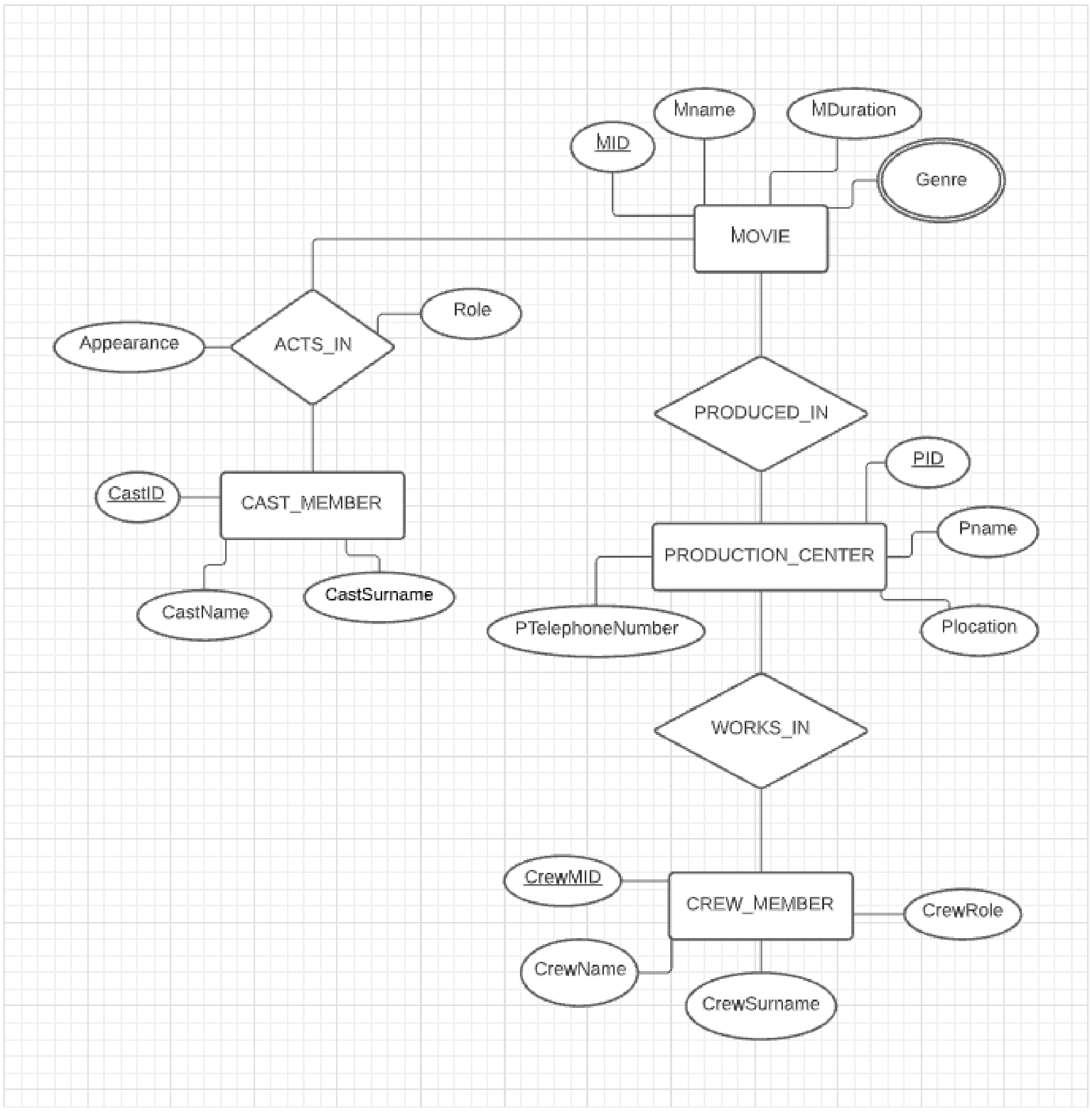
The CREW_MEMBER entity has table whose attributes are CrewMID, CrewName, CrewSurname, CrewRole and PID. CrewMID (int) is the unique identifier for a crew member’s record, CrewName(varchar(20)) and CrewSurname(varchar(20)) are the first and last names of the crew member, CrewRole(varchar(20)) is the role of the crew member in a production center where the role attribute’s value is limited to ’Director’, ’Assistant Director’, ’Camera Operator’, ’Set Dresser’, ’Make-up Artist’. Due to one-to-many relationship between crew member and a production center, PID (int) is the foreign key which refers to the production center’s PID attribute, because each crew member will have a specific role or job in only one specific production center.

The relations that were used:
There is a one-to-many relationship called "WORKS_IN" between "CREW_MEMBER" and "PRODUCTION_CENTER" entities. It’s one-to-many because a crew member works at one production center and a production center has several crew members. The relationship is implied implicitly in CREW_MEMBER entity, as explained in the entities.

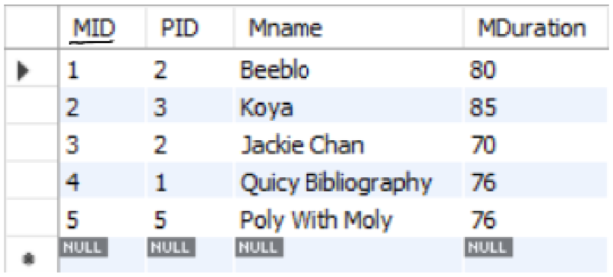
There is a many-to-many relationship called "ACTS_IN", between CAST_MEMBER and MOVIE where that relationship also stores the "Appearance" and "Role" attributes. The Appearance(varchar(20)) attribute shows the duration that a CAST_MEMBER acted in a movie and Role(varchar(20)) attribute specifies the role of a cast member in which movie. It is many-to-many because one cast member acts in several movies, also, a movie has several cast members. ACTS_IN has its own table which has attributes CastID, MID, Appearance, and Role because each cast member may have different appearance durations and different roles for which movie that he or she participates in.

There is a one-to-many relationship "PRODUCED_IN" between "PRODUCTION_CENTER" and "MOVIE". A production center may produce more than one movie and a movie is produced in only one production center. The relationship is implied implicitly in MOVIE entity, as explained in the entities.

ER Diagram and Snapshots

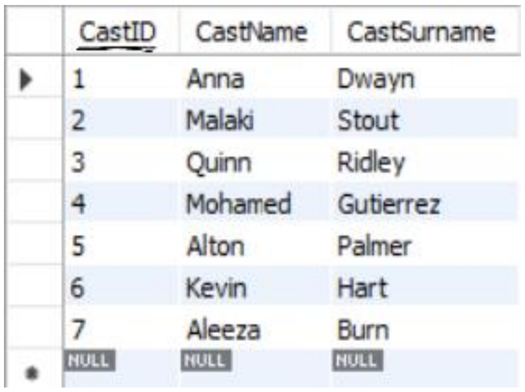


ER diagram



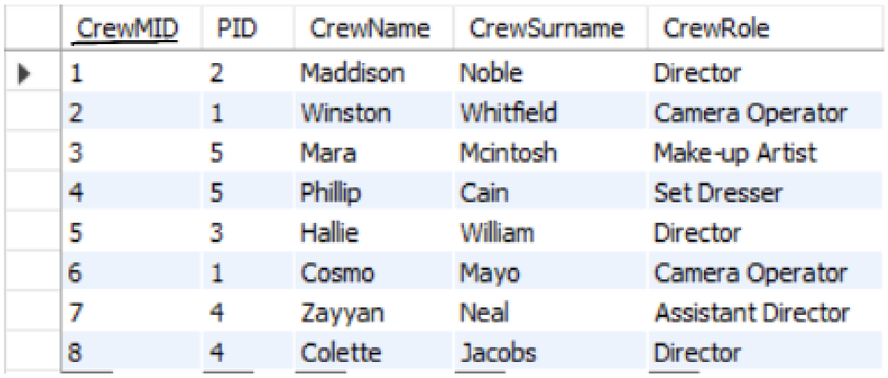
	MID	PID	Mname	MDuration
▶	1	2	Beeblo	80
	2	3	Koya	85
	3	2	Jackie Chan	70
	4	1	Quicy Bibliography	76
	5	5	Poly With Moly	76
*	NULL	NULL	NULL	NULL

MOVIE table



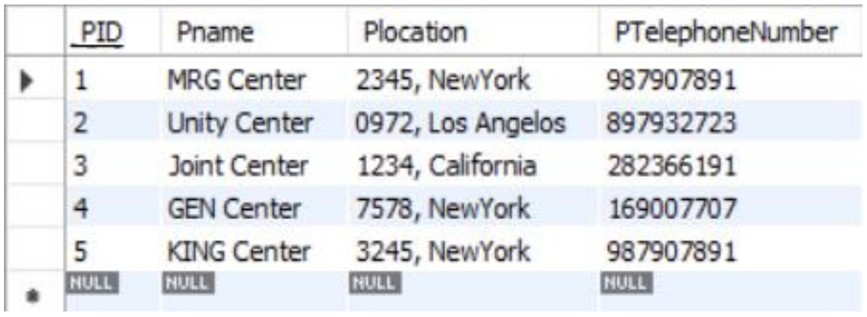
CastID	CastName	CastSurname
1	Anna	Dwayn
2	Malaki	Stout
3	Quinn	Ridley
4	Mohamed	Gutierrez
5	Alton	Palmer
6	Kevin	Hart
7	Aleeza	Burn
NULL	NULL	NULL

CAST_MEMBER table



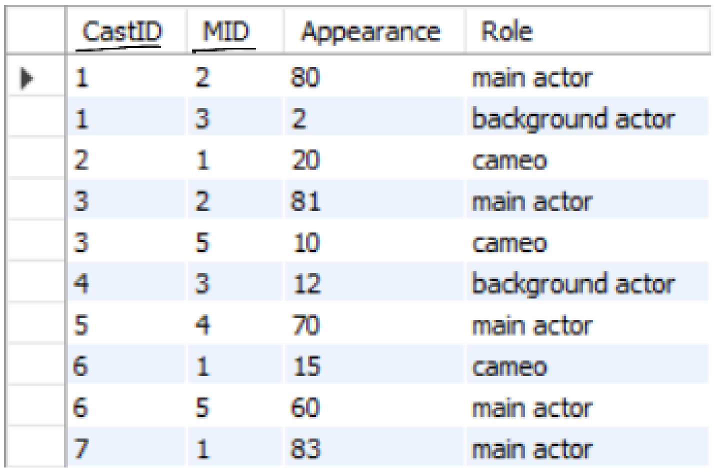
CrewMID	PID	CrewName	CrewSurname	CrewRole
1	2	Maddison	Noble	Director
2	1	Winston	Whitfield	Camera Operator
3	5	Mara	Mcintosh	Make-up Artist
4	5	Phillip	Cain	Set Dresser
5	3	Hallie	William	Director
6	1	Cosmo	Mayo	Camera Operator
7	4	Zayyan	Neal	Assistant Director
8	4	Colette	Jacobs	Director

CREW_MEMBER table



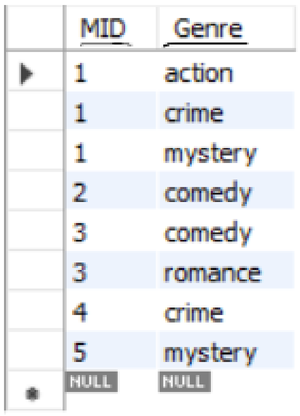
PID	Pname	Plocation	PTelephoneNumber
1	MRG Center	2345, NewYork	987907891
2	Unity Center	0972, Los Angelos	897932723
3	Joint Center	1234, California	282366191
4	GEN Center	7578, NewYork	169007707
5	KING Center	3245, NewYork	987907891
NULL	NULL	NULL	NULL

PRODUCTION_CENTER table



CastID	MID	Appearance	Role
1	2	80	main actor
1	3	2	background actor
2	1	20	cameo
3	2	81	main actor
3	5	10	cameo
4	3	12	background actor
5	4	70	main actor
6	1	15	cameo
6	5	60	main actor
7	1	83	main actor

ACTS_IN table



MID	Genre
1	action
1	crime
1	mystery
2	comedy
3	comedy
3	romance
4	crime
5	mystery
NULL	NULL

GENRE table

Explanation Of Keys and Cardinalities Of Each Table

MOVIE
Keys: MID is the primary key and PID is foreign key, referencing to PRODUCTION_CENTER table, due to its one-to-many implicit relationship with PRODUCTION_CENTER table. Having a PID for every movie can allow a production center to have many movies and a movie will be produced in one production center.

Cardinality: 5 is the cardinality of MOVIE table, because 5 tuples or entrances of MOVIE entity are added into its table to assign a production center for a movie.

CAST_MEMBER
Keys: CastID is the primary key because a cast member is identified by only its own primary key.

Cardinality: 7 is the cardinality of CAST_MEMBER table, because 7 tuples or entrances of cast members are added to the table.

CREW_MEMBER
Keys: CrewID primary key of the CREW_MEMBER table and PID is the foreign key referencing PRODUCTION_CENTER's attribute, PID, because there's a one-to-many implicit relationship between CREW_MEMBER and PRODUCTION_CENTER entities. Having a PID in CREW_MEMBER table shows that only one crew member is assigned to a production center, thus, a production center can have several crew members

Cardinality: 8 is the cardinality of CREW_MEMBER table, because 8 tuples or entrances of crew members are added to the table.

PRODUCTION_CENTER
Keys: PID is the primary key of PRODUCTION_CENTER entity because a production center is identified by only its own primary key.

Cardinality: 5 is the cardinality of PRODUCTION_CENTER table, because 5 tuples or entrances of production centers are added to the table.

ACTS_IN
Keys: CastID and MID is the superkey of ACTS_IN entity where both CastID and MID are compound primary keys of ACTS_IN table, because of the many-to-many relationship between CAST_MEMBER and MOVIE entities. This superkey helps to store the information of the roles, which are limited to "main actor", "background actor", "cameo" for every cast member in a movie, and appearance duration of each cast member for which movie.

Cardinality: 10 is the cardinality of ACTS_IN table, because each cast member is assigned to a movie. So, there are about 10 roles assigned to 7 cast members in many different movies.

GENRE
Keys: MID and Genre make the superkey of this table where MID and Genre are compound primary keys, because of the multi-vale attribute in MOVIE entity so a new table is created for GENRE to achieve the goal of a movie having more than one genre

Cardinality: 8 is the cardinality of GENRE table, because the table is populated according to one or more of the 5 movies may have more than one genre.

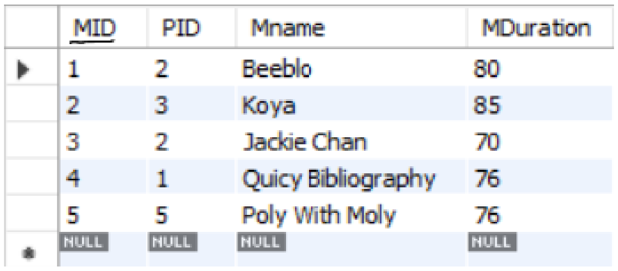
Converting Tables To 4NF

4NF MOVIE table

This table is in 1NF, because the attributes have atomic values of the same type, and each entrance is uniquely identified with MID. There are also no composite nor multi-value attributes.

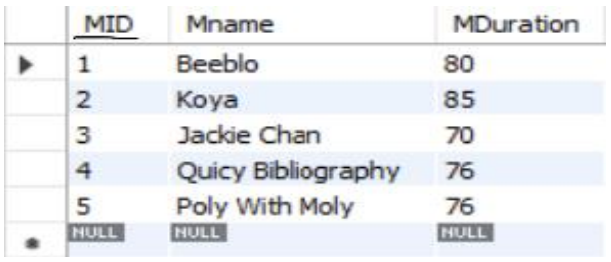
This is in 2NF and 3NF, because MID, a candidate key, uniquely identifies Mname, MDuration and PID. There are no partial dependencies.

This is not in 4NF because PID attribute is independent of Mname and MDuration but dependent on MID. It has a transitive dependency due to PID. So, table is converted from 3NF MOVIE (MID, PID, Mname, MDuration) to 4NF as MOVIE (MID, Mname, MDuration). Since PID is uniquely identified by MID, it is separated from MOVIE entity and is added to PRODUCED_IN table along with MID in which MID is the primary key of the new table. The table illustrated in figure 1.1 is the original 3NF of MOVIE table and figure 1.2 is the 4NF of MOVIE table. Figure 1.3 illustrates the product of converting MOVE from 3NF to 4NF.



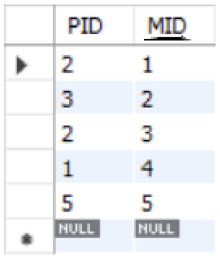
	MID	PID	Mname	MDuration
▶	1	2	Beeblo	80
	2	3	Koya	85
	3	2	Jackie Chan	70
	4	1	Quicy Bibliography	76
	5	5	Poly With Moly	76
*	NULL	NULL	NULL	NULL

Figure 1.1: MOVIE table



	MID	Mname	MDuration
▶	1	Beeblo	80
	2	Koya	85
	3	Jackie Chan	70
	4	Quicy Bibliography	76
	5	Poly With Moly	76
*	NULL	NULL	NULL

Figure 1.2: 4NF MOVIE table



	PID	MID
▶	2	1
	3	2
	2	3
	1	4
	5	5
*	NULL	NULL

Figure 1.3: 4NF PRODUCED_IN table

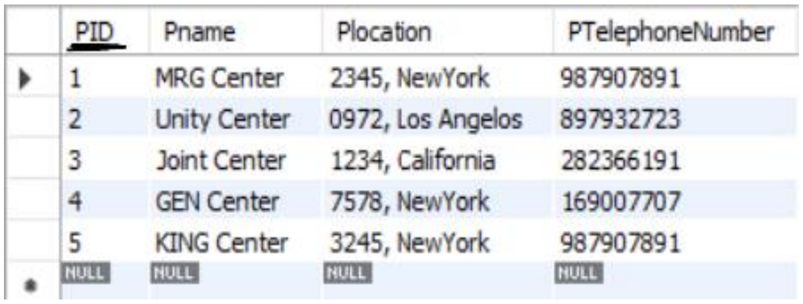
The results of transforming MOVIE entity into 4NF is illustrated in figures **1.1** to **1.3**

4NF PRODUCTION_CENTER table

PRODUCTION_CENTER table is in 1NF because all attribute's values are atomic, have same type and each entrance is uniquely identified by PID.

It is in 2NF and 3NF because all attributes depend on only one candidate key, PID. There are no partial dependencies.

It is also in 4NF because all attributes, Pname, Plocation, PTelephoneNumber, depend on PID. So, there is not multi-values dependency. The 4NF of PRODUCTION_CENTER table is illustrated in figure 2. There are no transitive dependencies.



PID	Pname	Plocation	PTelephoneNumber
1	MRG Center	2345, NewYork	987907891
2	Unity Center	0972, Los Angelos	897932723
3	Joint Center	1234, California	282366191
4	GEN Center	7578, NewYork	169007707
5	KING Center	3245, NewYork	987907891
NULL	NULL	NULL	NULL

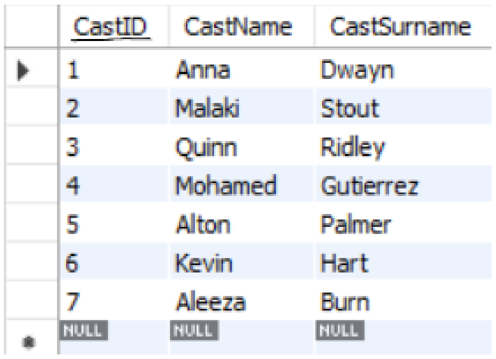
Figure 2: 4NF PRODUCTION_CENTER table

4NF CAST_MEMBER table

CAST_MEMEBER table is in 1NF because all attribute's values are atomic, have same type and each entrance is uniquely identified by CastID.

This table is in 2NF and 3NF because all attributes depend on only one candidate key, CastID. There are no partial dependencies

This table is also in 4NF, because all attributes, Castname, CastSurname depend on only one key, CastID. So, there is no multi-valued dependency. The 4NF of CAST_MEMBER table is illustrated in figure 3.



CastID	CastName	CastSurname
1	Anna	Dwayn
2	Malaki	Stout
3	Quinn	Ridley
4	Mohamed	Gutierrez
5	Alton	Palmer
6	Kevin	Hart
7	Aleeza	Burn
NULL	NULL	NULL

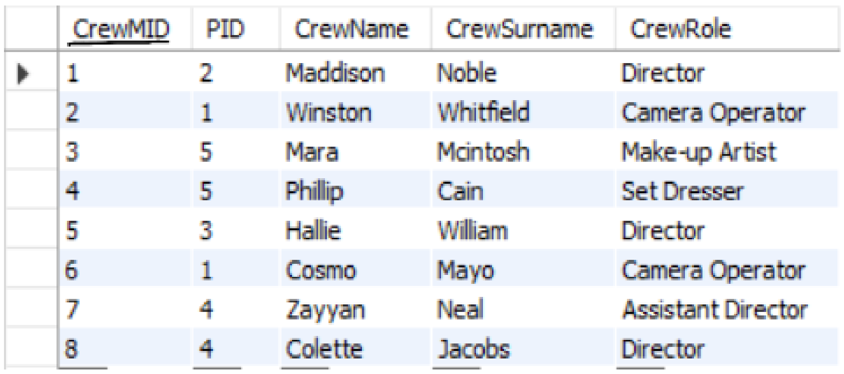
Figure 3: 4NF CAST_MEMBER table

4NF CREW_MEMBER table

CREW_MEMBER table is in 1NF because all attribute's values are atomic, have same type and each entrance is uniquely identified by CrewMID.

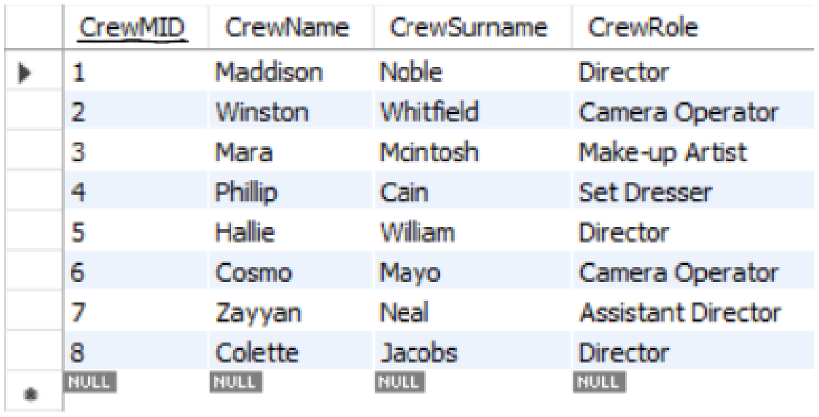
The table is in 2NF and 3NF because all attributes depend on only one candidate key, CrewMID.

It's not in 4NF, because there's a multi-value dependency or transitive dependency where PID is independent of CrewName, CrewSurname, and CrewRole yet is uniquely identified by CrewMID. So, table is converted from CREW_MEMBER (PID, CrewMID, CrewName, CrewSurname, CrewRole) to CREW_MEMBER(CrewMID, CrewName, CrewSurname, CrewRole). The results of converting the CREW_MEMBER table from 3NF into 4NF are illustrated in figures 4.1 and 4.2. Figure 4.1 is the original 3NF of the table and figure 4.2 is the 4NF of CREW_MEMBER table. Since PID is independent of CrewName, CrewSurname and CrewRole attributes, it is separated into its own table, WORKS_IN, with CrewMID, where PID is uniquely identified by CrewMID and CrewMID is the primary key of the WORKS_IN table.



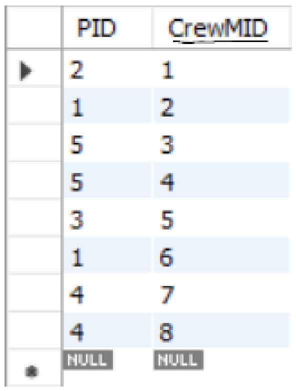
CrewMID	PID	CrewName	CrewSurname	CrewRole
1	2	Maddison	Noble	Director
2	1	Winston	Whitfield	Camera Operator
3	5	Mara	Mcintosh	Make-up Artist
4	5	Phillip	Cain	Set Dresser
5	3	Hallie	William	Director
6	1	Cosmo	Mayo	Camera Operator
7	4	Zayyan	Neal	Assistant Director
8	4	Colette	Jacobs	Director

Figure 4.1: CREW_MEMBER table



CrewMID	CrewName	CrewSurname	CrewRole
1	Maddison	Noble	Director
2	Winston	Whitfield	Camera Operator
3	Mara	Mcintosh	Make-up Artist
4	Phillip	Cain	Set Dresser
5	Hallie	William	Director
6	Cosmo	Mayo	Camera Operator
7	Zayyan	Neal	Assistant Director
8	Colette	Jacobs	Director
NULL	NULL	NULL	NULL

Figure 4.2: 4NF CREW_MEMBER table



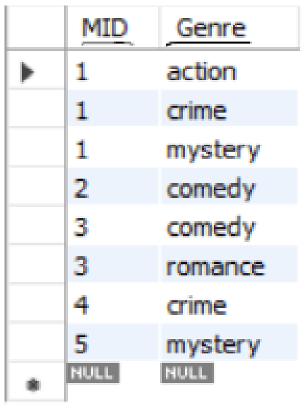
PID	CrewMID
2	1
1	2
5	3
5	4
3	5
1	6
4	7
4	8
NULL	NULL

Figure 4.3: 4NF WORKS_IN table

The results of transforming CREW_MEMBER entity into 4NF in figures 4.1 to 4.3.

4NF GENRE table

The GENRE table is already in 1NF because it has no composite nor multi-value attributes. It is in 2NF and 3Nf because both attributes depend on the other. It is already in 4NF because there is no multi-value dependency.

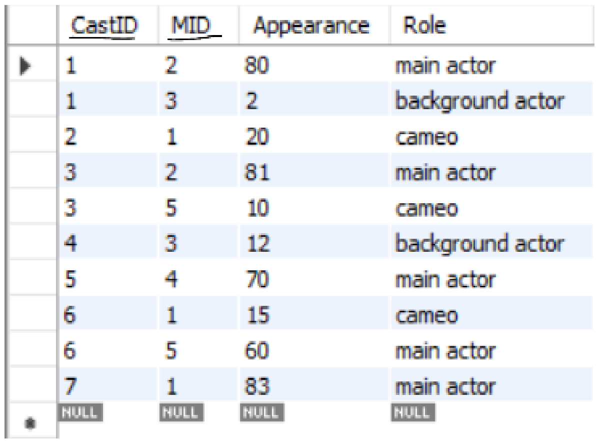


MID	Genre
1	action
1	crime
1	mystery
2	comedy
3	comedy
3	romance
4	crime
5	mystery
NULL	NULL

Figure 5: 4NF GENRE table

4NF ACTS_IN table

The ACTS_IN table is already 1NF because its attributes' values are atomic, of the same type and each entrance is uniquely identified by CastID and MID. It is in 2NF and 3NF because all entrances depend on MID and CastID attributes. It is also 4NF because both MID and CastID is a superkey so both attributes depend on each other while the attributes, Appearance and Role depend on the superkey.



CastID	MID	Appearance	Role
1	2	80	main actor
1	3	2	background actor
2	1	20	cameo
3	2	81	main actor
3	5	10	cameo
4	3	12	background actor
5	4	70	main actor
6	1	15	cameo
6	5	60	main actor
7	1	83	main actor
NULL	NULL	NULL	NULL

Figure 6: 4NF ACTS_IN table