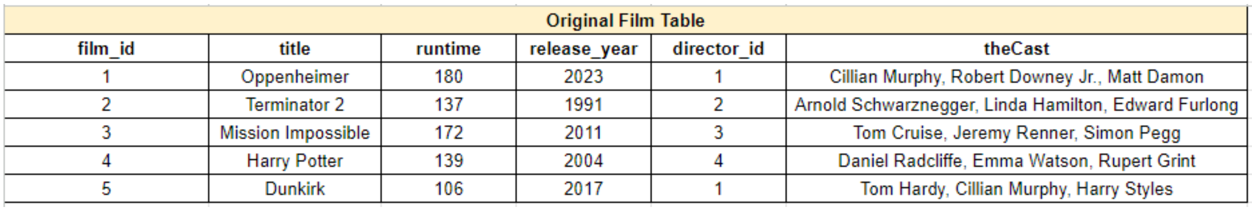
Section: 09
Course: CPS510 - Database Systems 1
Team Number: 17
Assignment Number: A7
TA Name: Reza Barati
Team Members: Anmol Panchal, Deep Patel, Aryan Patel

Firstly, let's acknowledge the baseline conditions required for a table to be in 3NF.

1. No repeating groups, and each column having atomic values ← (1NF)
2. All non-keyed attributes as functional dependencies on the primary key + 1NF ← (2NF)
3. All transitive dependencies must be removed + 2NF ← (3NF)

For the explanations and rubric 'Decomposition', please continue reading. The rubric 'Create a transitive dependency and compound primary key,' is contained in the middle and end.

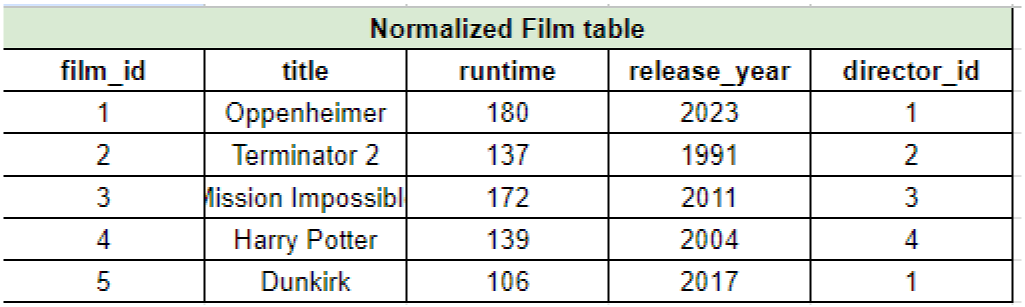
Original Film Table



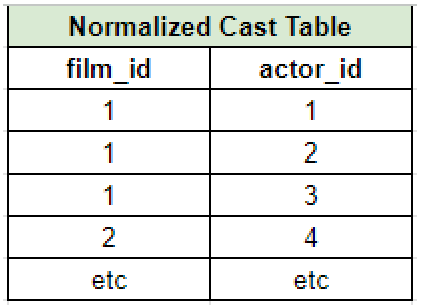
Original Film Table					
film_id	title	runtime	release_year	director_id	theCast
1	Oppenheimer	180	2023	1	Cillian Murphy, Robert Downey Jr., Matt Damon
2	Terminator 2	137	1991	2	Arnold Schwarznegger, Linda Hamilton, Edward Furlong
3	Mission Impossible	172	2011	3	Tom Cruise, Jeremy Renner, Simon Pegg
4	Harry Potter	139	2004	4	Daniel Radcliffe, Emma Watson, Rupert Grint
5	Dunkirk	106	2017	1	Tom Hardy, Cillian Murphy, Harry Styles

This table has dependents of title, runtime, release_year, and director_id and determinant of film_id, which is the primary key. According to clause 1 of the requirements aforementioned, each column should have atomic values (cannot be broken down further). Clearly, theCast column is in violation of this.

Normalized Film Table and Decomposition into Cast Table



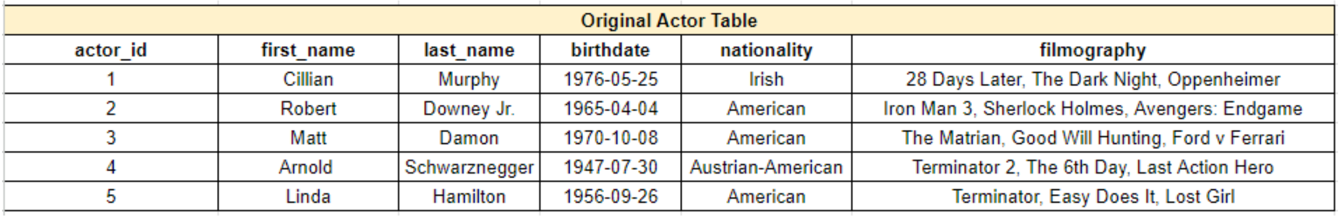
Normalized Film table				
film_id	title	runtime	release_year	director_id
1	Oppenheimer	180	2023	1
2	Terminator 2	137	1991	2
3	Mission Impossibl	172	2011	3
4	Harry Potter	139	2004	4
5	Dunkirk	106	2017	1



Normalized Cast Table	
film_id	actor_id
1	1
1	2
1	3
2	4
etc	etc

To meet the 1NF requirements, theCast column has been removed and associated with its own table, found below. Now, film_id → (title, runtime, release_year, director_id), (actor_id)
The table is in 2NF because all dependent attributes are non-keyed. 3NF can also be verified because there are no transitive dependencies.

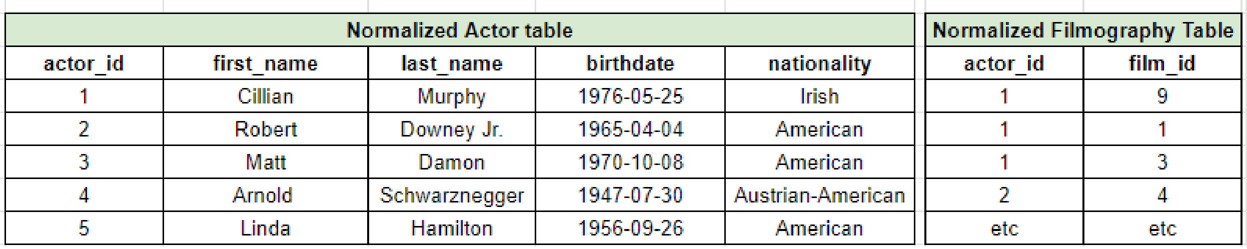
Original Actor Table



Original Actor Table					
actor_id	first_name	last_name	birthdate	nationality	filmography
1	Cillian	Murphy	1976-05-25	Irish	28 Days Later, The Dark Night, Oppenheimer
2	Robert	Downey Jr.	1965-04-04	American	Iron Man 3, Sherlock Holmes, Avengers: Endgame
3	Matt	Damon	1970-10-08	American	The Matrian, Good Will Hunting, Ford v Ferrari
4	Arnold	Schwarznegger	1947-07-30	Austrian-American	Terminator 2, The 6th Day, Last Action Hero
5	Linda	Hamilton	1956-09-26	American	Terminator, Easy Does It, Lost Girl

For 1NF normalization, we shall again take the filmography attribute and organize it into a dedicated table. Note how (first_name, last_name) → (birthday) can also ensure uniqueness.

Normalized Actor Table and **Decomposition into Actors_Filmography Table**

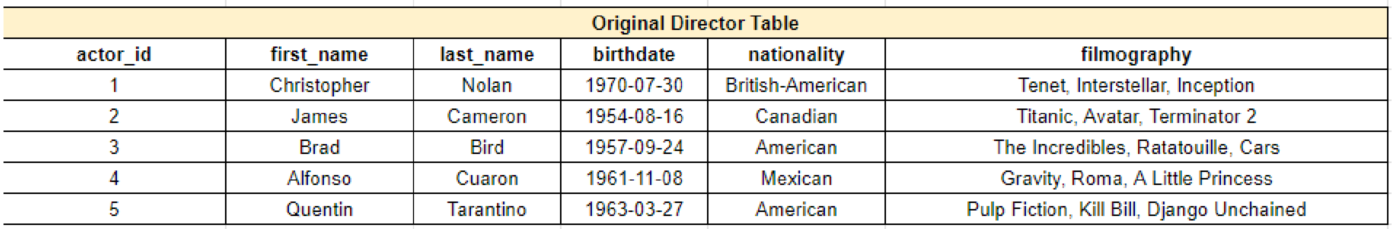


Normalized Actor table				
actor_id	first_name	last_name	birthdate	nationality
1	Cillian	Murphy	1976-05-25	Irish
2	Robert	Downey Jr.	1965-04-04	American
3	Matt	Damon	1970-10-08	American
4	Arnold	Schwarznegger	1947-07-30	Austrian-American
5	Linda	Hamilton	1956-09-26	American

Normalized Filmography Table	
actor_id	film_id
1	9
1	1
1	3
2	4
etc	etc

This 1NF normalization features a primary key of actor_id (determinant) and the attributes first_name, last_name, birthdate, and nationality (dependants) are all atomic attributes. Moreover, this is also 2NF because each dependent is a non-keyed attribute. Finally, this table can be verified as 3NF because there are no transitive dependencies. In other words, none of the non-key attributes are depending on other non-key attributes.

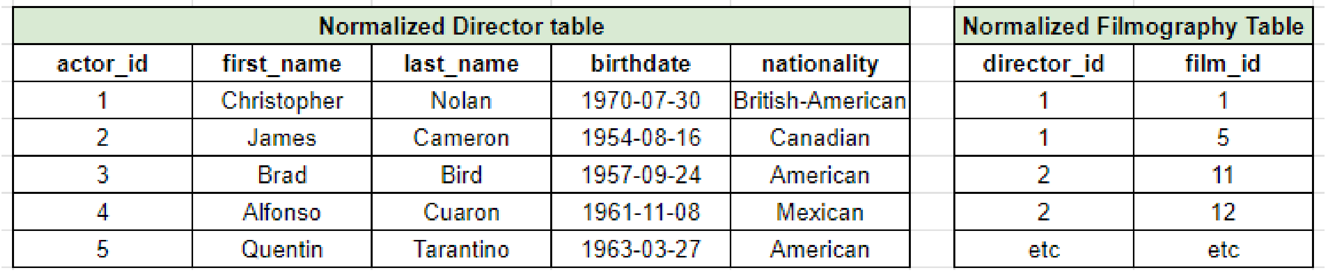
Original Director Table



Original Director Table					
actor_id	first_name	last_name	birthdate	nationality	filmography
1	Christopher	Nolan	1970-07-30	British-American	Tenet, Interstellar, Inception
2	James	Cameron	1954-08-16	Canadian	Titanic, Avatar, Terminator 2
3	Brad	Bird	1957-09-24	American	The Incredibles, Ratatouille, Cars
4	Alfonso	Cuaron	1961-11-08	Mexican	Gravity, Roma, A Little Princess
5	Quentin	Tarantino	1963-03-27	American	Pulp Fiction, Kill Bill, Django Unchained

For 1NF normalization, similar to Actor and Film tables, we can simplify the table to only have atomic attributes.

Normalized Director Table and **Decomposition into Directors_Filmography Table**

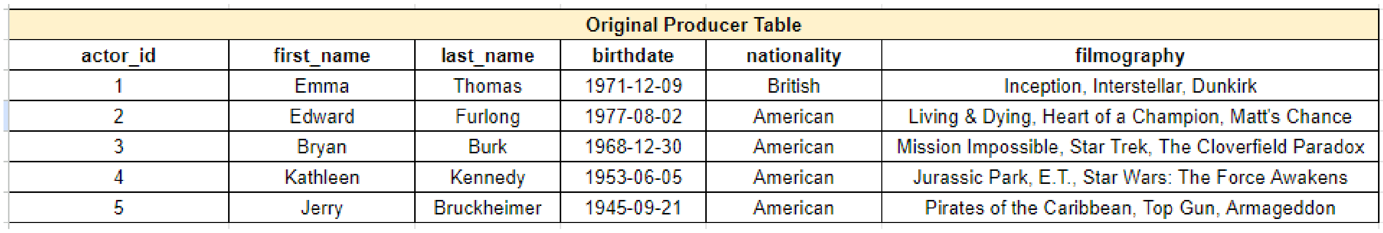


Normalized Director table				
actor_id	first_name	last_name	birthdate	nationality
1	Christopher	Nolan	1970-07-30	British-American
2	James	Cameron	1954-08-16	Canadian
3	Brad	Bird	1957-09-24	American
4	Alfonso	Cuaron	1961-11-08	Mexican
5	Quentin	Tarantino	1963-03-27	American

Normalized Filmography Table	
director_id	film_id
1	1
1	5
2	11
2	12
etc	etc

Now it is clear that the table is in 2NF and 3NF because all non-key attributes are functionally dependent. Also, there are no transitive dependencies, when non-key attributes are indirectly using the primary key to rely on other dependents.

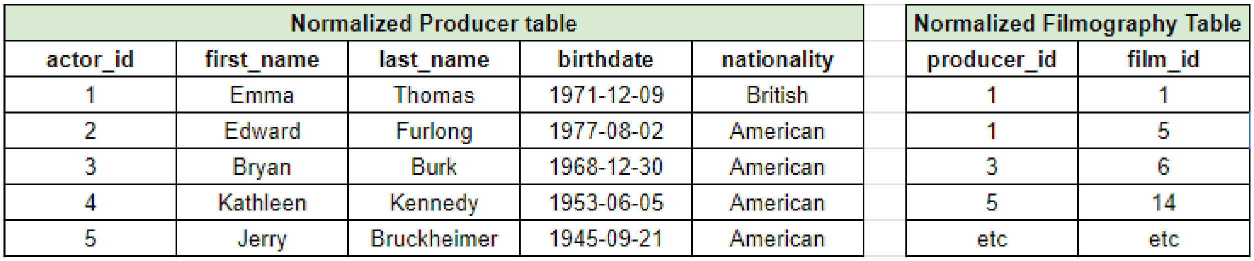
Original Producer Table



Original Producer Table					
actor_id	first_name	last_name	birthdate	nationality	filmography
1	Emma	Thomas	1971-12-09	British	Inception, Interstellar, Dunkirk
2	Edward	Furlong	1977-08-02	American	Living & Dying, Heart of a Champion, Matt's Chance
3	Bryan	Burk	1968-12-30	American	Mission Impossible, Star Trek, The Cloverfield Paradox
4	Kathleen	Kennedy	1953-06-05	American	Jurassic Park, E.T., Star Wars: The Force Awakens
5	Jerry	Bruckheimer	1945-09-21	American	Pirates of the Caribbean, Top Gun, Armageddon

Due to the highly similar format of this table to the Actor, and Director, the same methodologies of 1NF, and 2NF normalization are used.

Normalized Producer Table and Decomposition into Producers_Filmography Table

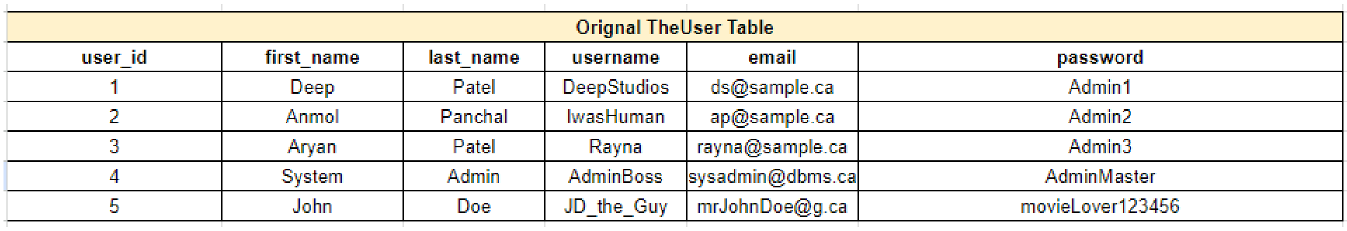


Normalized Producer table				
actor_id	first_name	last_name	birthdate	nationality
1	Emma	Thomas	1971-12-09	British
2	Edward	Furlong	1977-08-02	American
3	Bryan	Burk	1968-12-30	American
4	Kathleen	Kennedy	1953-06-05	American
5	Jerry	Bruckheimer	1945-09-21	American

Normalized Filmography Table	
producer_id	film_id
1	1
1	5
3	6
5	14
etc	etc

Here, the primary key of actor_id is the determinant for the dependent attributes of first_name, last_name, birthdate, and nationality. Due to the decomposition of the producer's filmographies, 3NF verification is complete.

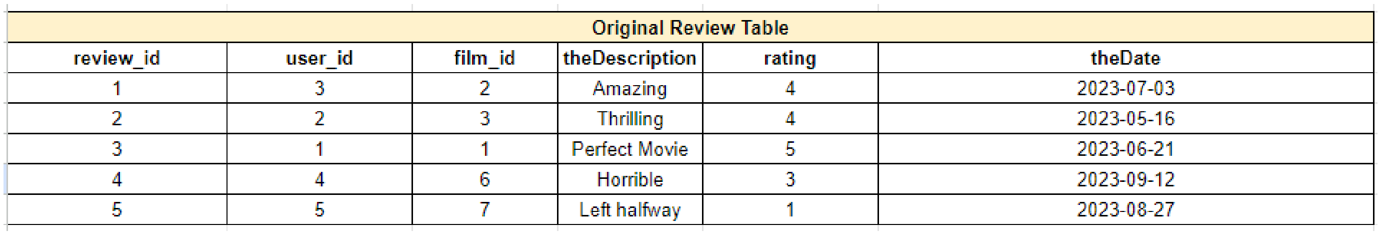
Original TheUser Table



Orignal TheUser Table					
user_id	first_name	last_name	username	email	password
1	Deep	Patel	DeepStudios	ds@sample.ca	Admin1
2	Anmol	Panchal	IwasHuman	ap@sample.ca	Admin2
3	Aryan	Patel	Rayna	rayna@sample.ca	Admin3
4	System	Admin	AdminBoss	sysadmin@dbms.ca	AdminMaster
5	John	Doe	JD_the_Guy	mrJohnDoe@g.ca	movieLover123456

Notice that the determinant is user_id, and the dependents are first_name, last_name, username, email, and password. Every single dependent here is already an atomic value, thus 1NF. Also, there are no foreign keys in this table. Considering all non-key dependents are functional dependencies on the primary key, the table is 2NF verified. Finally, there are no transitive dependencies. Thus the table is already in 3NF.

Original Review Table



Original Review Table

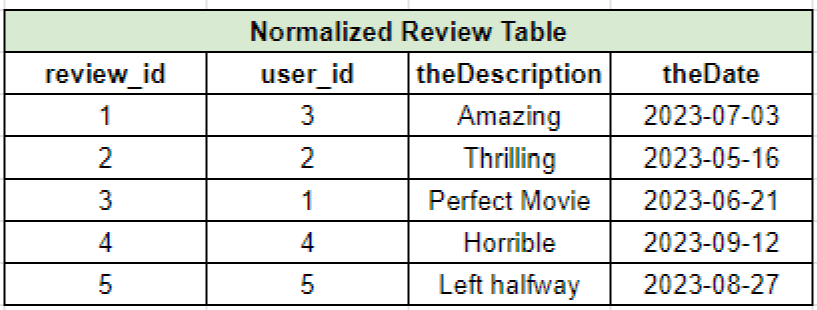
review_id	user_id	film_id	theDescription	rating	theDate
1	3	2	Amazing	4	2023-07-03
2	2	3	Thrilling	4	2023-05-16
3	1	1	Perfect Movie	5	2023-06-21
4	4	6	Horrible	3	2023-09-12
5	5	7	Left halfway	1	2023-08-27

This table requires the most work. Immediately, there is a primary key of review_id (determinant), but also two foreign keys (user_id and film_id). Due to the all non-key dependants (theDescription, rating, theDate) being functionally dependent on the primary_key and the table containing all atomic values. The table is verified 1NF and 2NF. For 3NF, consider the below procedure.

We know that the rating depends on the review_id, but it also is indirectly connected to the film_id too. To have a rating at all, we need to have a film in mind, and a record log.

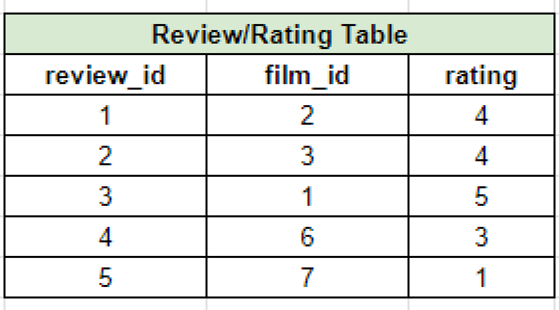
To remove this transitive dependency, split the table.

Normalized Review Table and **Decomposition into Reviews_Ratings_Table**



Normalized Review Table

review_id	user_id	theDescription	theDate
1	3	Amazing	2023-07-03
2	2	Thrilling	2023-05-16
3	1	Perfect Movie	2023-06-21
4	4	Horrible	2023-09-12
5	5	Left halfway	2023-08-27



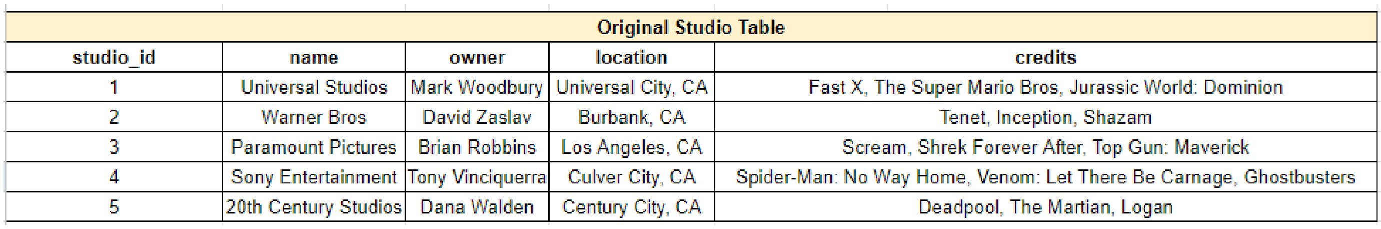
Review/Rating Table

review_id	film_id	rating
1	2	4
2	3	4
3	1	5
4	6	3
5	7	1

Previously we had a transitive dependency of review_id → rating → film_id.
Now, the normalized review table is 1NF because all values are atomic. It is also 2NF because all of the non-key attributes are functionally dependent on the primary key. Also it is 3NF, because the transitive dependency is removed. There is an entirely new table now, which handles the relational between review_id, film_id, and the rating, keeping the record log and film connected while being 3NF.

Normalization continued on the next page.

Original Studio Table

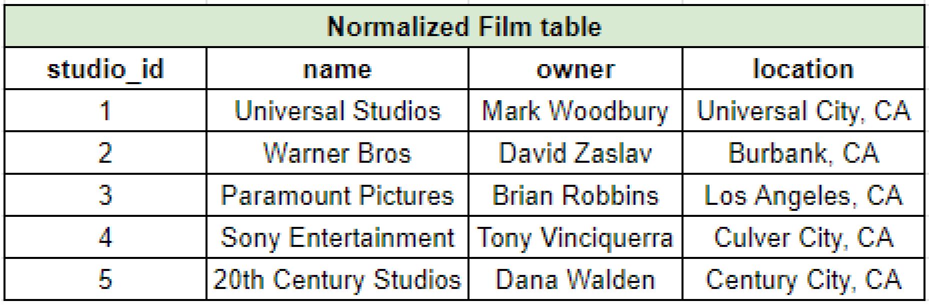


Original Studio Table

studio_id	name	owner	location	credits
1	Universal Studios	Mark Woodbury	Universal City, CA	Fast X, The Super Mario Bros, Jurassic World: Dominion
2	Warner Bros	David Zaslav	Burbank, CA	Tenet, Inception, Shazam
3	Paramount Pictures	Brian Robbins	Los Angeles, CA	Scream, Shrek Forever After, Top Gun: Maverick
4	Sony Entertainment	Tony Vinciquerra	Culver City, CA	Spider-Man: No Way Home, Venom: Let There Be Carnage, Ghostbusters
5	20th Century Studios	Dana Walden	Century City, CA	Deadpool, The Martian, Logan

The primary key is studio_id (determinant), with all of the attributes being non-keyed attributes. The list includes name, owner, location, and credits (determinants). This table needs to be decomposed to fix the credits attribute and ensure all values are atomic.

Normalized Review Table

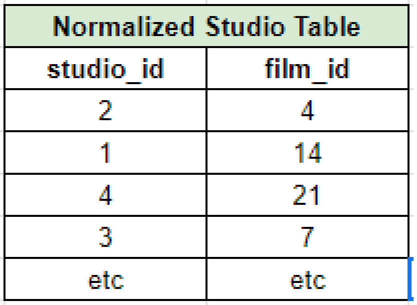


Normalized Film table

studio_id	name	owner	location
1	Universal Studios	Mark Woodbury	Universal City, CA
2	Warner Bros	David Zaslav	Burbank, CA
3	Paramount Pictures	Brian Robbins	Los Angeles, CA
4	Sony Entertainment	Tony Vinciquerra	Culver City, CA
5	20th Century Studios	Dana Walden	Century City, CA

This table is also 2NF, because all of the non-keyed attributes are functionally dependent on the primary key.

Decomposition into Studio_Credits_Table

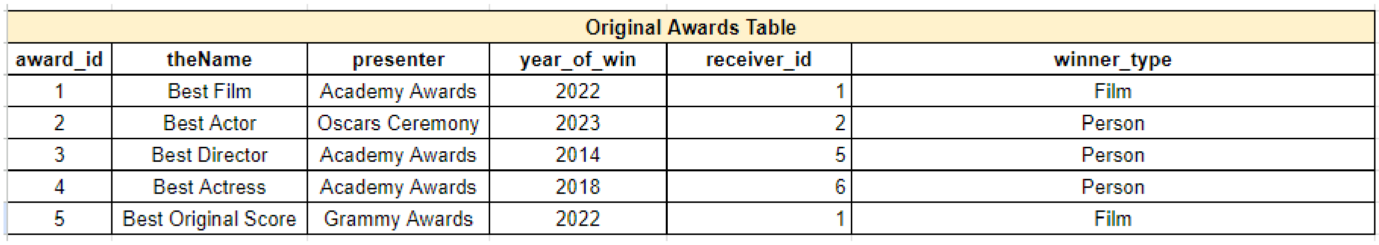


Normalized Studio Table

studio_id	film_id
2	4
1	14
4	21
3	7
etc	etc

The film credits for each studio can be made more efficient to the 3NF format by ensuring all transitive dependencies are removed. The table on the left has completed this objective in accordance with the normalized Review table.

Original Awards Table



Original Awards Table

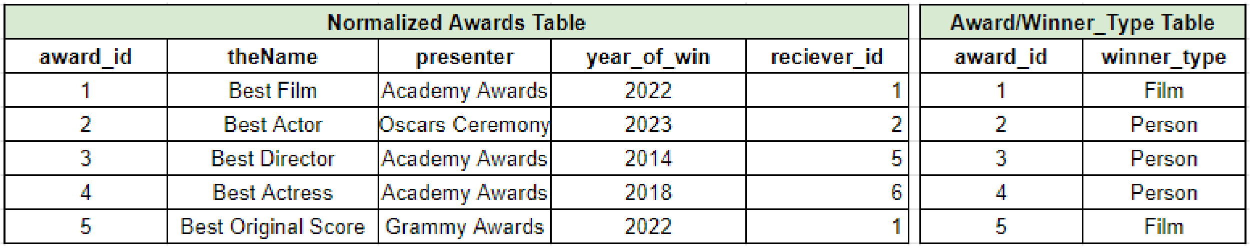
award_id	theName	presenter	year_of_win	receiver_id	winner_type
1	Best Film	Academy Awards	2022	1	Film
2	Best Actor	Oscars Ceremony	2023	2	Person
3	Best Director	Academy Awards	2014	5	Person
4	Best Actress	Academy Awards	2018	6	Person
5	Best Original Score	Grammy Awards	2022	1	Film

This table clearly violates 3NF. Let's fix it.

Firstly, note that the primary key here is award_id and it acts as a determinant for theName, presenter, year_of_win, and winner_type, which are the dependent attributes. For 2NF, the non-key attributes must be functionally dependent, which they are, thus 2NF verified.

The reason the table violates 3NF is because a transitive dependency exists, which is award_id → winner_type → receiever_id

Normalized Awards Table and **Decomposition into Winner_Type_Table**

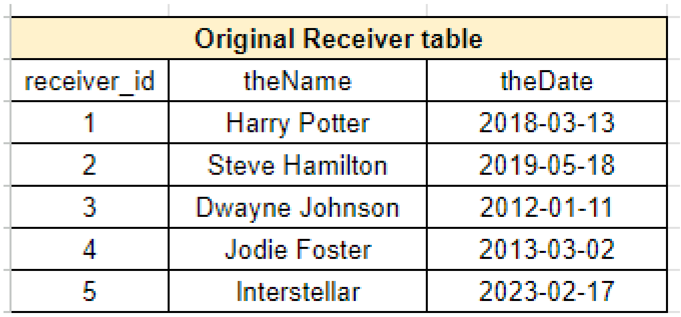


Normalized Awards Table				
award_id	theName	presenter	year_of_win	reciever_id
1	Best Film	Academy Awards	2022	1
2	Best Actor	Oscars Ceremony	2023	2
3	Best Director	Academy Awards	2014	5
4	Best Actress	Academy Awards	2018	6
5	Best Original Score	Grammy Awards	2022	1

Award/Winner_Type Table	
award_id	winner_type
1	Film
2	Person
3	Person
4	Person
5	Film

Now it is clear that the normalized awards table only contains attributes directly related to the primary key. The decomposition creates a new table which has winner_type as a function dependent on the award_id. This effectively removed the transitive dependency, thus 3NF verified.

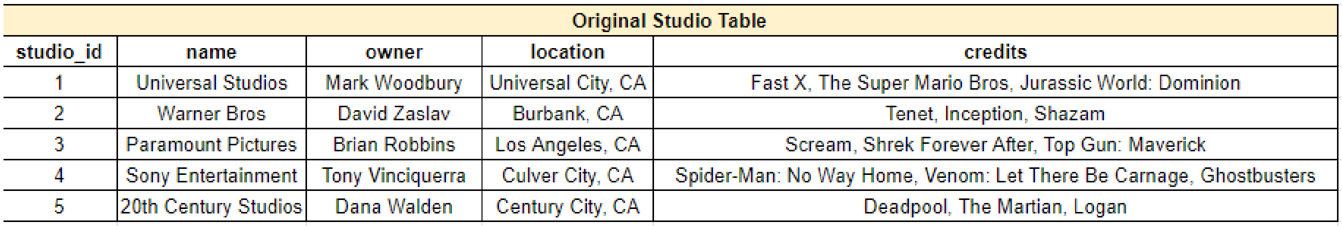
Original Receiver Table



Original Receiver table		
receiver_id	theName	theDate
1	Harry Potter	2018-03-13
2	Steve Hamilton	2019-05-18
3	Dwayne Johnson	2012-01-11
4	Jodie Foster	2013-03-02
5	Interstellar	2023-02-17

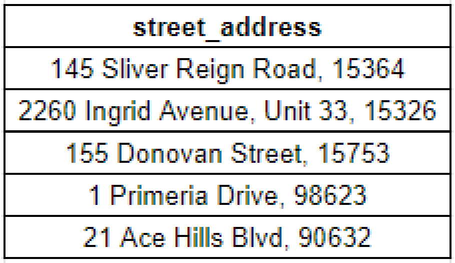
Finally, the Receiver table was already in 1NF and 2NF. This is due to all non-key attributes (in this case theName and theDate) being functional dependencies of the primary key (in this case receiver_id). Moreover, the table is also in 3NF, because no transitive dependencies are present.

In conclusion, we can consider compound primary keys as a group of existing attributes that help to uniquely identify a data set. In this case, the Actor table uses the primary key of actor_id, but also the notation (first_name, last_name) → (birthdate, nationality). Constraints can also be used to group multiple primary keys together under as common, and this would act like a compound key. For example, consider the Studio table, on the next page.



Original Studio Table				
studio_id	name	owner	location	credits
1	Universal Studios	Mark Woodbury	Universal City, CA	Fast X, The Super Mario Bros, Jurassic World: Dominion
2	Warner Bros	David Zaslav	Burbank, CA	Tenet, Inception, Shazam
3	Paramount Pictures	Brian Robbins	Los Angeles, CA	Scream, Shrek Forever After, Top Gun: Maverick
4	Sony Entertainment	Tony Vinciquerra	Culver City, CA	Spider-Man: No Way Home, Venom: Let There Be Carnage, Ghostbusters
5	20th Century Studios	Dana Walden	Century City, CA	Deadpool, The Martian, Logan

Here, the studio_id is the primary key, but introducing a more specific location (address components) can act as a compound primary key. No two studios can coexist at the same street address!

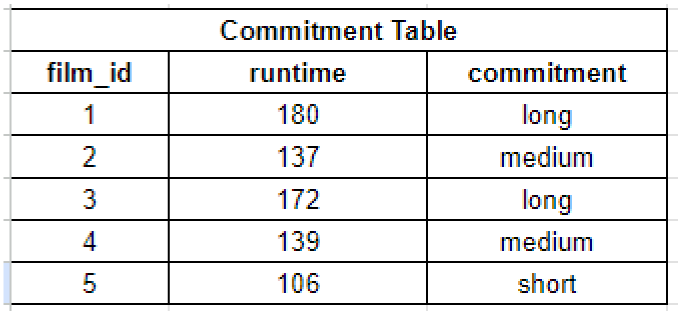


street_address
145 Sliver Reign Road, 15364
2260 Ingrid Avenue, Unit 33, 15326
155 Donovan Street, 15753
1 Primeria Drive, 98623
21 Ace Hills Blvd, 90632

(studio_id, street_address) → (name, owner, credits).

The LHS here is effectively a compound key, because it uses certain data constraints to ensure multiple unique identifiers are grouped to act as a compound key.

In regards to the transitive functional dependency, examine the Film table. Let's introduce an attribute called commitment, which wants to track how long users have to commit to sit down and watch the film. Its possible values are short, medium, and long. In this case the commitment attribute is dependent on the runtime, while the runtime is dependent on the film_id. This is a transitive dependency of form film_id → runtime → commitment.



Commitment Table		
film_id	runtime	commitment
1	180	long
2	137	medium
3	172	long
4	139	medium
5	106	short

Thus a transitive FD is created.

This concludes the A7 report, focused on 3NF normalization. The next assignment continues to improve the database by performing BCNF normalization.