

Section: 09

Course: CPS510 - Database Systems 1

Team Number: 17

Assignment Number: A6

TA Name: Reza Barati

Team Members: Anmol Panchal, Deep Patel, Aryan Patel

Functional Dependencies:

This assignment uses techniques in database normalization, in alignment with the rubric to provide functional dependencies (FDs), which we have highlighted in yellow.

Pre-Normalization using Functional Dependencies:

This section shows all of the FDs associated with the primary key for each table in our program. They are represented under each table name. The objective here is to reduce redundancy in data, and ensure that integrity is maintained with our data sets.

Film Table:

FILM_ID	TITLE	RUNTIME	RELEASE_YEAR	DIRECTOR_ID	THECAST
1	1 Oppenheimer	180	2023	1	Cillian Murphy, Robert Downey Jr., Matt Damon
2	2 Terminator 2: Judgement Day	137	1991	2	Arnold Schwarzenegger, Linda Hamilton, Edward Furlong
3	3 Mission: Impossible - Ghost Protocol	172	2011	3	Tom Cruise, Jeremy Renner, Simon Pegg
4	4 Harry Potter and The Prisoner of Azkaban	139	2004	4	Daniel Radcliffe, Rupert Grint, Emma Watson
5	5 Dunkirk	106	2017	1	Tom Hardy, Cillian Murphy, Harry Styles
6	6 Pulp Fiction	154	1994	5	John Travolta, Samuel L. Jackson, Uma Thurman
7	7 Jurassic Park	127	1993	6	Sam Neill, Laura Dern, Jeff Goldblum
8	8 The Matrix	136	1999	8	Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss
9	9 Forrest Gump	142	1994	9	Tom Hanks, Robin Wright, Gary Sinise

Film(film_id, title, runtime, release_year, director_id, theCast)

- film_id → title
- film_id → runtime
- film_id → release_year
- film_id → director_id
- film_id → theCast

FD: (film_id) → (title, runtime, release_year, director_id, theCast)

Explanation: The primary key of Film is film_id. The attributes shown above ensure that each film has a UNIQUE title, runtime, release_year, director, and cast information. Even if there is overlap between two films with the same title, the primary key of 'film_id' ensures that each film has a unique identifier.

Actor Table:

⚡	⚡	⚡	⚡	⚡	
ACTOR_ID	FIRST_NAME	LAST_NAME	BIRTHDATE	NATIONALITY	FILMOGRAPHY
1	1 Cillian	Murphy	76-05-25	Irish	28 Days Later, The Dark Knight, Oppenheimer
2	2 Robert	Downey Jr.	65-04-04	American	Iron Man 3, Sherlock Holmes, Avengers: Endgame
3	3 Matt	Damon	70-10-08	American	The Martian, Good Will, Hunting, Ford v Ferrari
4	4 Arnold	Schwarzenegger	47-07-30	Austrian-American	Terminator 2: Judgement Day, The 6th Day, Last Action Hero
5	5 Linda	Hamilton	56-09-26	American	Terminator: Dark Fate, Easy Does It, Lost Girl
6	6 Edward	Furlong	77-08-02	American	Terminator 2: Judgement Day, The Green Hornet, Detroit Rock City
7	7 Tom	Cruise	62-07-03	American	Top Gun: Maverick, Mission: Impossible - Dead Reckoning Part One, Jack Reacher
8	8 Jeremy	Renner	71-01-07	American	Captain America: Civil War, Kill the Messenger, Mission: Impossible - Rogue Nation
9	9 Simon	Pegg	70-02-14	British	The Ice Age Adventures of Buck Wild, Hot Fuzz, Mission: Impossible - Dead Reckoning Part One
10	10 Meryl	Streep	49-06-22	American	The Devil Wears Prada, Mamma Mia!, The Iron Lady
11	11 Leonardo	DiCaprio	74-11-11	American	Titanic, The Revenant, Inception
12	12 Angelina	Jolie	75-06-04	American	Tomb Raider, Maleficent, Girl, Interrupted

Actor(actor_id, first_name, last_name, birthdate, nationality, filmography)

- actor_id → first_name
- actor_id → last_name
- actor_id → birthdate
- actor_id → nationality
- actor_id → filmography

FD: (actor_id) → (first_name, last_name, birthdate, nationality, filmography)

Explanation: The primary key is actor_id. The representation above enables each actor to have a unique profile, such that the database maintains personal information integrity. Because the actor_id is the functional dependency here, it does not matter if we have two, three, or even four hundred people with the same name. Their relational data will maintain integrity due to ‘actor_id.’

Director Table:

⚡	⚡	⚡	⚡	⚡	
DIRECTOR_ID	FIRST_NAME	LAST_NAME	BIRTHDATE	NATIONALITY	FILMOGRAPHY
1	1 Christopher	Nolan	70-07-30	British-American	Tenet, Interstellar, Inception
2	2 James	Cameron	54-08-16	Canadian-American	Titanic, Avatar: The Way of Water, Terminator 2: Judgement Day
3	3 Brad	Bird	57-09-24	American	The Incredibles, Ratatouille, Cars
4	4 Alfonso	Cuaron	61-11-08	Mexican	Gravity, Roma, A Little Princess
5	5 Quentin	Tarantino	63-03-27	American	Pulp Fiction, Kill Bill, Django Unchained
6	6 Steven	Spielberg	46-12-18	American	Jurassic Park, E.T. the Extra-Terrestrial, Schindler's List
7	7 Lana	Wachowski	65-06-21	American	The Matrix, Cloud Atlas, Sense8
8	8 Robert	Zemeckis	51-05-14	American	Forrest Gump, Back to the Future, Cast Away
9	9 Frank	Darabont	59-01-28	American	The Shawshank Redemption, The Green Mile, The Mist

Director(director_id, first_name, last_name, birthdate, nationality, filmography)

- director_id → first_name
- director_id → last_name
- director_id → birthdate
- director_id → nationality
- director_id → filmography

FD: (director_id) → (first_name, last_name, birthdate, nationality, filmography)

Explanation: The primary key here is director_id, acting as a functional dependency to complete the same objective as the actor table. Even if there are overlaps between directors' names or birthdays, the functional dependency of director_id ensures we can uniquely and accurately identify directors in the system.

Producer Table:

PRODUCER_ID	FIRST_NAME	LAST_NAME	BIRTH...	NATIONALITY	FILMOGRAPHY
1	Emma	Thomas	71-12-09	British	Inception, Interstellar, Dunkirk
2	Edward	Furlong	77-08-02	American	Living , Heart of a Champion, Matt's Chance
3	Bryan	Burk	68-12-30	American	Mission: Impossible - Rogue Nation, Star Trek Into Darkness, The Cloverfield Paradox
4	Kathleen	Kennedy	53-06-05	American	Jurassic Park, E.T. the Extra-Terrestrial, Star Wars: The Force Awakens
5	Jerry	Bruckheimer	45-09-21	American	Pirates of the Caribbean, Top Gun, Armageddon

Producer(producer_id, first_name, last_name, birthdate, nationality, filmography)

- producer_id → first_name
- producer_id → last_name
- producer_id → birthdate
- producer_id → nationality
- producer_id → filmography

FD: (producer_id) → (first_name, last_name, birthdate, nationality, filmography)

Explanation: The functional dependency here is provided by producer_id, because it ensures we can uniquely identify each producer in the system. The logic for this representation is very similar to the explanations mentioned for Actor and Director.

TheUser Table:

USER_ID	FIRST_NAME	LAST_NAME	USERNAME	EMAIL	PASSWORD
1	Anmol	Panchal	IwasHuman	anmol.panchal@torontomu.ca	Admin1
2	Deep	Patel	Deep Studios	deep.c.patel@torontomu.ca	Admin2
3	Aryan	Patel	Rayna	aryan.patel@torontomu.ca	Admin3
4	John	Doe	JohnDoe123	john.doe@gmail.com	ilovemovies
5	Emma	Smithson	harrypotterfan450	emma.smithson@gmail.com	voldemortthedark
6	Daniel	Mosfet	007Lover	daniel.mosfet@007.com	007SecretAgent
7	System	Admin	AdminSir	sytem_admin@dbms.com	AdminMaster

TheUser(user_id, first_name, last_name, username, email, password)

- user_id → first_name
- user_id → last_name
- user_id → username
- user_id → email
- user_id → password

FD: (user_id) → (first_name, last_name, username, email, password) or
(username) → (user_id, first_name, last_name, email, password)

Explanation: The unique identifier of ‘user_id’ here is a functional dependency to help manage user login functionality. This is crucial to maintain integrity , for example, let’s say two people in our system have the same password, we do not want a user to obtain access to data that is not theirs. Alternatively, two different people may not have the same username, so this can also act as a functional dependency (unique) for the other attributes.

Review Table:

REVIEW_ID	USER_ID	FILM_ID	THEDESCRIPTION	RATING	THEDATE
1	1	3	2 This movie was action packed with my favorite action hero Arnold!	4	23-07-03
2	2	2	3 The iconic run of Tom Cruise made this movie epic to watch.	4	23-05-16
3	3	1	1 Amazing cinematics from Nolan once again. Pristine performance!.	5	23-06-21
4	4	4	6 An unforgettable cinematic experience! Tarantino's masterpiece.	5	23-09-12
5	5	5	7 A thrilling adventure with dinosaurs! Spielberg's classic.	4	23-08-27

Review(review_id, user_id, film_id, theDescription, rating, theDate)

- review_id → user_id
- review_id → film_id
- review_id → theDescription
- review_id → rating
- review_id → theDate

FD: (review_id) -> (user_id, film_id, theDescription, rating, theDate)

Explanation: Each review is identified uniquely with a review_id. The objective here is to maintain data integrity for the ratings given to movies, and the functional dependency helps to ensure that in each review there is information about the user, film, and of course how the user liked the film.

Studio Table:

STUDIO_ID	NAME	OWNER	LOCATION	CREDITS
1	1 Universal Studios	Mark Woodbury	Universal City, California	Fast X, The Super Mario Bros, Jurassic World: Dominion
2	2 Warner Bros	David Zaslav	Burbank, California	Tenet, Inception, Shazam
3	3 Paramount Pictures	Brian Robbins	Los Angeles, California	Scream, Shrek Forever After, Top Gun: Maverick
4	4 Sony Pictures Entertainment	Tony Vinciguerra	Culver City, California	Spider-Man: No Way Home, Venom: Let There Be Carnage, Ghostbusters: Afterlife
5	5 20th Century Studios	Dana Walden	Century City, California	Deadpool, The Martian, Logan
6	6 New Line Cinema	Ann Sarnoff	Burbank, California	The Lord of the Rings: The Fellowship of the Ring, A Nightmare on Elm Street, The Conjuring

Studio(studio_id, name, owner, location, credits)

- studio_id → name
- studio_id → owner
- studio_id → location
- studio_id → credits

FD: (studio_id) -> (name, owner, location, credits)

Explanation: The functional dependency here is the studio_id which contains the data for studio name, owner, and location. This is done to ensure the database contains accurate information

about what studios are actively making movies, and to avoid situations such as wrongly crediting a particular studio for a film, when the studio claimed in the database doesn't even exist, etc.

Awards Table:

	AWARD_ID	THENAME	PRESENTER	YEAR_OF_WIN	RECEIVER_ID	WINNER_TYPE
1	1	Best Film	Academy Awards	2022	1	Film
2	2	Best Actor	Oscars Ceremony	2023	2	Person
3	3	Best Director	Academy Awards	2014	3	Person
4	4	Best Actress	Academy Awards	2018	2	Person
5	5	Best Original Score	Grammy Awards	2022	1	Film

Awards(award_id, theName, presenter, year_of_win, receiver_id, winner_type)

- award_id → theName
- award_id → presenter
- award_id → year_of_win
- award_id → receiver_id
- award_id → winner_type

FD: (award_id) → (theName, presenter, year_of_win, receiver_id, winner_type)

Explanation: The primary key of the Awards table is award_id. It acts as a functional dependency to keep a unique track of which awards are won. Whether it is a film or person, the representation set keeps track of the award presenter, year of winning and who won it.

Receiver Table:

	RECEIVER_ID	THENAME	THEDATE
1	1	Harry Potter and The Philosopher's Stone	18-03-13
2	2	Steve Hamilton	19-05-18
3	3	Dwayne Johnson	12-01-11
4	4	Interstellar	23-02-17
5	5	Inception	10-07-08
6	6	Jodie Foster	13-03-02

Receiver(receiver_id, theName, theDate)

- receiver_id → theName
- receiver_id → theDate

FD: (receiver_id) → (theName, theDate)

Explanation: The 'receiver_id' acts as the functional dependency of the receiver table, because it helps to accurately keep track of the award receivers. This can also reduce redundancy with the Awards table to ensure that Receiver table entries have won type of award, and Award table entries are in fact receivers.