

Business Domain Study and Database Design Project

Software Platform Domain: Netflix

NETFLIX

Netflix, Inc.

Nattapol Phanmadee	6522781994
Chayathon Chaiyasak	6522771300
Kitssada Pavasutthipand	6522772530
Aekkarin Yimyaem	6522772498
Mutita Phansamdang	6522781143
Pavida Pipatanagovit	6722800073

CSS325: Database System
Asst. Prof. Dr. Preecha Tangworakitthaworn

Sirindhorn International Institute of Technology

Thammasat University

Table of Contents

Sirindhorn International Institute of Technology Thammasat University.....	0
Table of Contents.....	1
1 Netflix Overview.....	3
1.1 Business Processes.....	3
1.1.1 Netflix Tudum.....	3
1.1.3 Content Tagging Process.....	3
1.1.4 Netflix Shop.....	4
1.1.5 Netflix Games.....	4
1.2 ERD.....	5
2 Step-by-Step Relational Diagram Transformation.....	6
Step1: Strong Entity.....	6
Step2: Weak Entity.....	7
Step3: Binary 1:1 Relationship.....	7
Step4: Binary 1:M Relationship.....	7
Step5: Binary M:N Relationship.....	8
Step6: Multi-Valued Attribute.....	9
Step7: N-ary Relationship.....	9
Step8A: EERD.....	10
Final Relational Schema.....	11
3 Data Dictionary.....	12
4 Table Description & Implementation.....	16
4.1 Ads Table.....	16
4.2 Apparel Table.....	17
4.3 Apparel_color Table.....	17
4.4 Apparel_size Table.....	18
4.5 Article_tudum Table.....	19
4.6 Author Table.....	20
4.7 Author_write_article Table.....	21
4.8 Collectible Table.....	23
4.9 Content Table.....	24
4.10 Content_cast Table.....	26
4.11 Content_Director Table.....	27
4.12 Content_Writer Table.....	28
4.13 Coupon Table.....	29
4.14 Coupon_eligiblefor_item Table.....	30
4.15 Drinkware Table.....	31
4.16 Game Table.....	32
4.17 Game_Category Table.....	33

4.18 Game_Developer Table.....	34
4.19 Game_Gamemode Table.....	35
4.20 Game_Supported_Languages Table.....	37
4.21 Genre Table.....	38
4.22 Genre_Contain_Content Table.....	39
4.23 merchandise_item table.....	40
4.24 merchandise_order Table.....	41
4.25 merchandise_review Table.....	45
4.26 movie Table.....	48
4.27 order_contains_item Table.....	49
4.28 order_use_coupon Table.....	50
4.29 payment_method Table.....	51
4.30 plan Table.....	52
4.31 Profile Table.....	53
4.32 Subscription Table.....	55
4.33 tv_episode Table.....	57
4.34 tv_show Table.....	60
4.35 user_account Table.....	60
4.36 user_download_game Table.....	61
4.37 user_shown_ads Table.....	63
4.38 user_view_article Table.....	64
4.39 user_view_content Table.....	67

1 Netflix Overview

Netflix is a global streaming service that offers a vast library of TV shows, movies, documentaries, and original content. It allows users to watch their favorite content on various devices, including TVs, computers, smartphones, and tablets. Netflix has revolutionized the way people consume entertainment by providing on-demand access to a wide range of programming.

1.1 Business Processes

1.1.1 Netflix Tudum

Netflix Tudum is an online platform created by Netflix to celebrate its content and engage with its audience. The name "Tudum" is inspired by the sound that plays when you start watching something on Netflix. It serves as a way for the streaming giant to connect with fans by providing articles containing exclusive interviews, announcements, and behind-the-scenes access to its original movies, series, and stars.

1.1.2 User Subscription Process

This process encompasses all the steps involved in a user signing up for a subscription, managing their account, and maintaining their subscription over time. The goal is to create a seamless and user-friendly experience that encourages potential subscribers to sign up and helps retain them for the long term.

1.1.3 Content Tagging Process

Netflix employs content analysts, known colloquially as "taggers" - employees whose roles are to assign specific keywords to various content on the platform. [10] These keywords include the release year, director cast, theme, language, how much sex or profanity is in the

content and genre of said content. These keywords are then used as metadata to help promote content to the right audience.

1.1.4 Netflix Shop

Netflix Shop is an online store where you can find merchandise inspired by your favorite Netflix shows and movies. It offers a variety of products, including:

Apparel: T-shirts, hoodies, hats, and other clothing items featuring characters, logos, and quotes from popular Netflix series.

Collectibles: Figures, toys, and other collectible items based on your favorite shows.

Home Goods: Decorative items, such as mugs, blankets, and wall art, inspired by Netflix content.

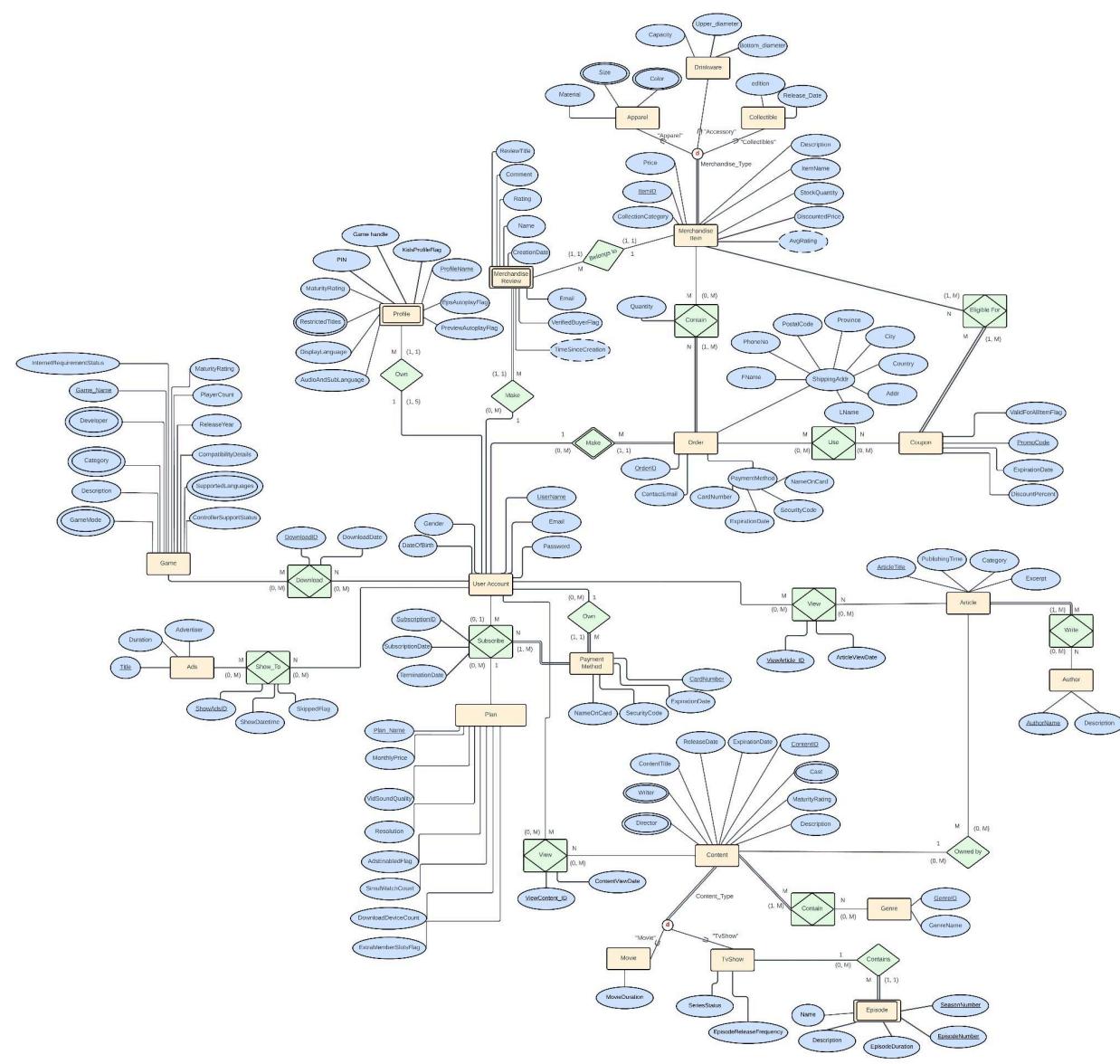
1.1.5 Netflix Games

Netflix Games is a gaming platform integrated within the Netflix app that allows subscribers to play a variety of mobile games on their smartphones and tablets. The games cover various genres, including action, puzzle, adventure, and strategy, and are designed to cater to different age groups and preferences. The games are free from ads, in-app purchases, and additional costs, making them accessible to all Netflix subscribers as part of their existing membership. Users must download the games before playing, making Netflix Games similar in functionality to game publishing platforms, such as Google Play.

1.1.6 Advertising Process

Netflix Advertising, introduced in 2022 with the "Basic with Ads" plan, is the company's venture into offering ad-supported streaming. This service allows users to subscribe at a lower cost while viewing ads during shows and movies.

1.2 ERD



2. Step-by-Step Relational Diagram Transformation

Step1: Strong Entity

Game

Game_Name	Description	InternetRequirementStatus	MaturityRating	PlayerCount	ReleaseYear	ControllerSupportStatus	CompatibilityDetails
-----------	-------------	---------------------------	----------------	-------------	-------------	-------------------------	----------------------

User Account

UserName	Email	Password	DateOfBirth	Gender
----------	-------	----------	-------------	--------

Plan

Plan_Name	MonthlyPrice	VidSoundQuality	Resolution	AdsEnabledFlag	SimulWatchCount	DownloadDeviceCount	ExtraMemberSlotsFlag
-----------	--------------	-----------------	------------	----------------	-----------------	---------------------	----------------------

Payment Method

CardNumber	ExpirationDate	NameOnCard	SecurityCode
------------	----------------	------------	--------------

Content

ContentTitle	ReleaseDate	ExpirationDate	ContentID	MaturityRating	Description	TotalDuration
--------------	-------------	----------------	-----------	----------------	-------------	---------------

Genre

GenreID	GenreName
---------	-----------

Article

ArticleTitle	PublishingTime	Category	Excerpt
--------------	----------------	----------	---------

Author

AuthorName	Description
------------	-------------

Order

OrderID	ContactEmail	CardNumber	ExpirationDate	SecurityCode	NameOnCard	FName	LName	PostalCode	Province	City	Country	Addr
---------	--------------	------------	----------------	--------------	------------	-------	-------	------------	----------	------	---------	------

Merchandise Item

CollectionCategory	ItemID	Price	Description	ItemName	StockQuantity	DiscountedPrice
--------------------	--------	-------	-------------	----------	---------------	-----------------

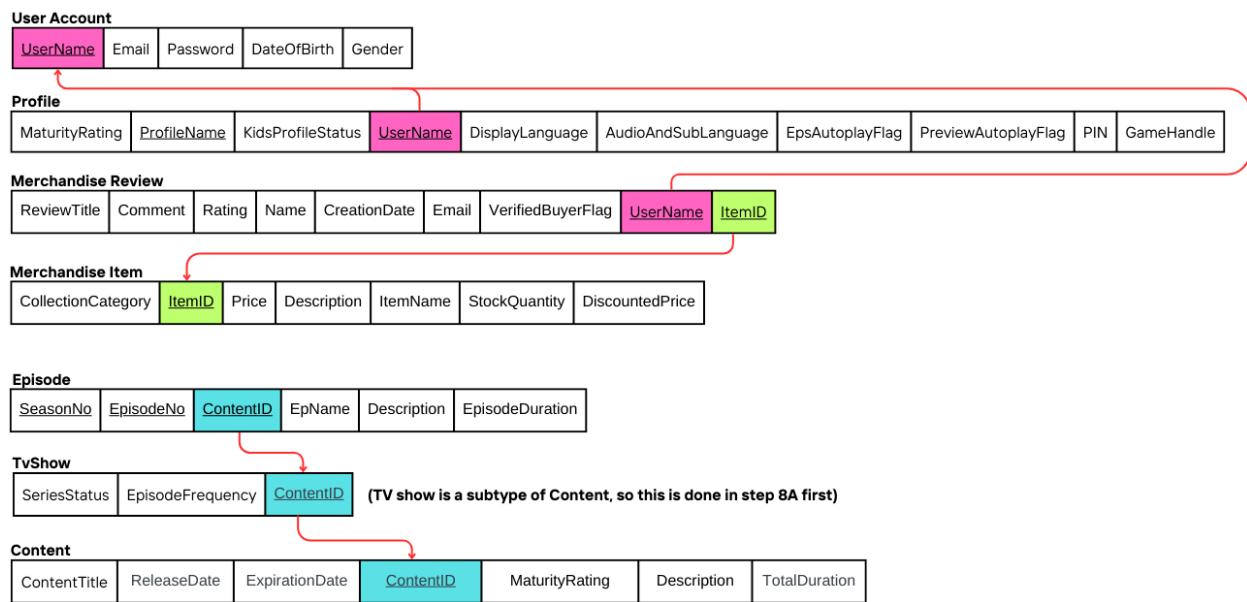
Ads

AdsTitle	Duration	Advertiser
----------	----------	------------

Coupon

PromoCode	ExpirationDate	DiscountPercent	IsValidForAllItem
-----------	----------------	-----------------	-------------------

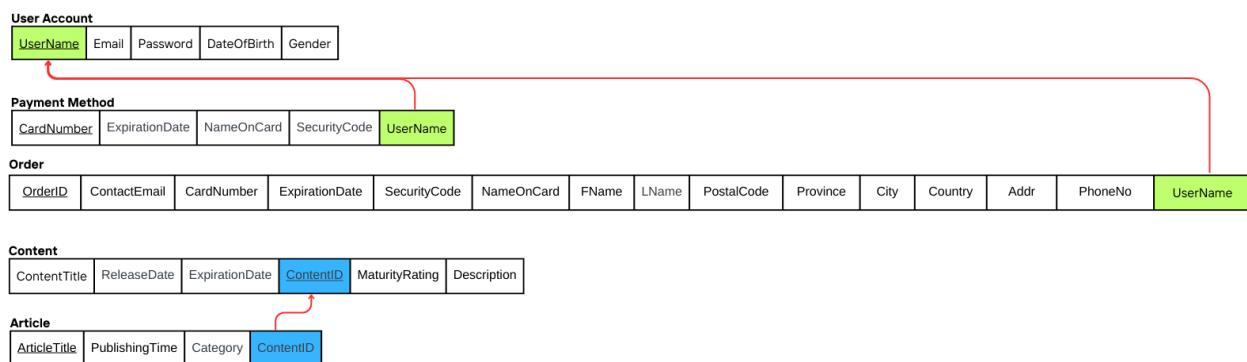
Step2: Weak Entity



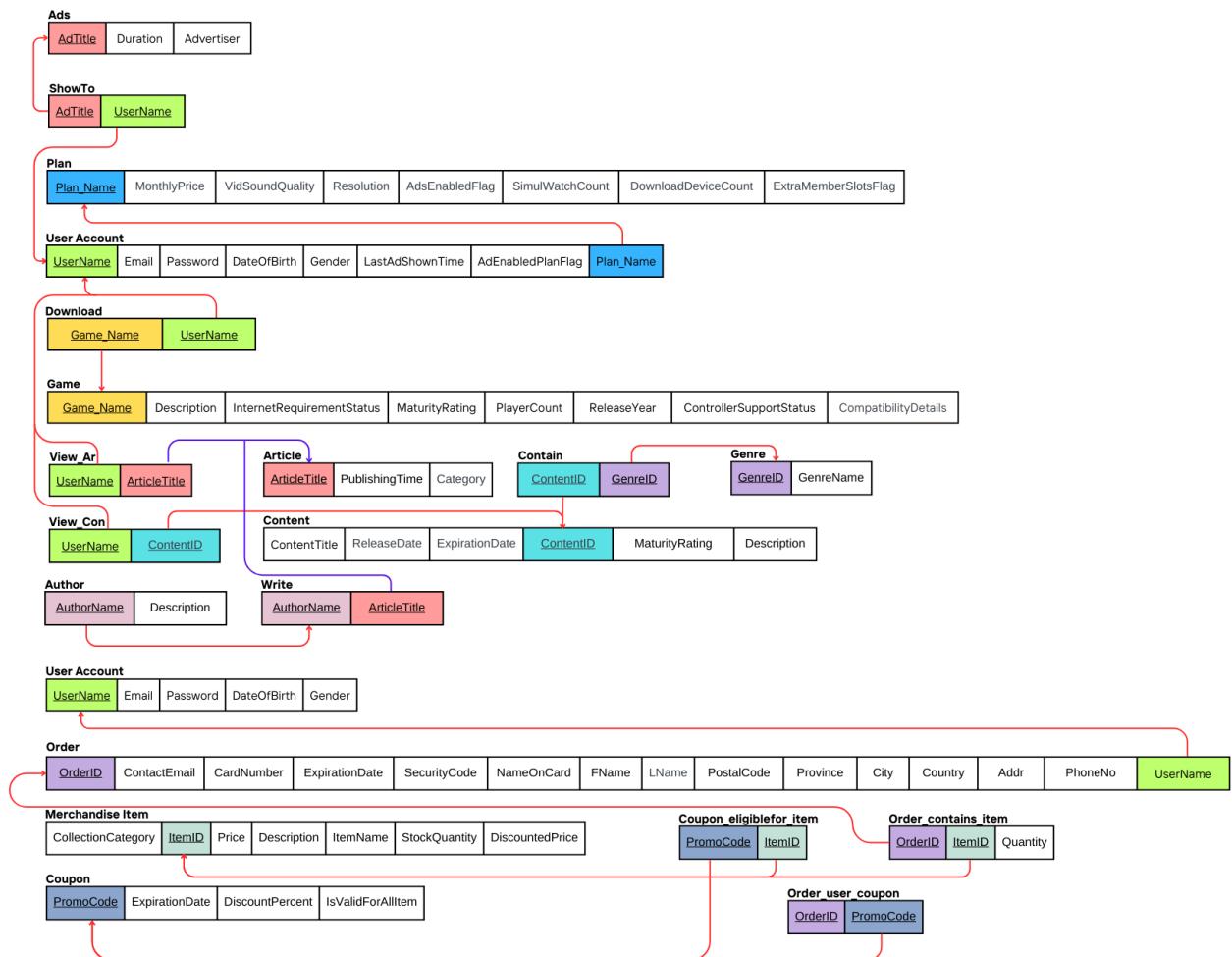
Step3: Binary 1:1 Relationship

There exists no 1:1 relationship in this schema

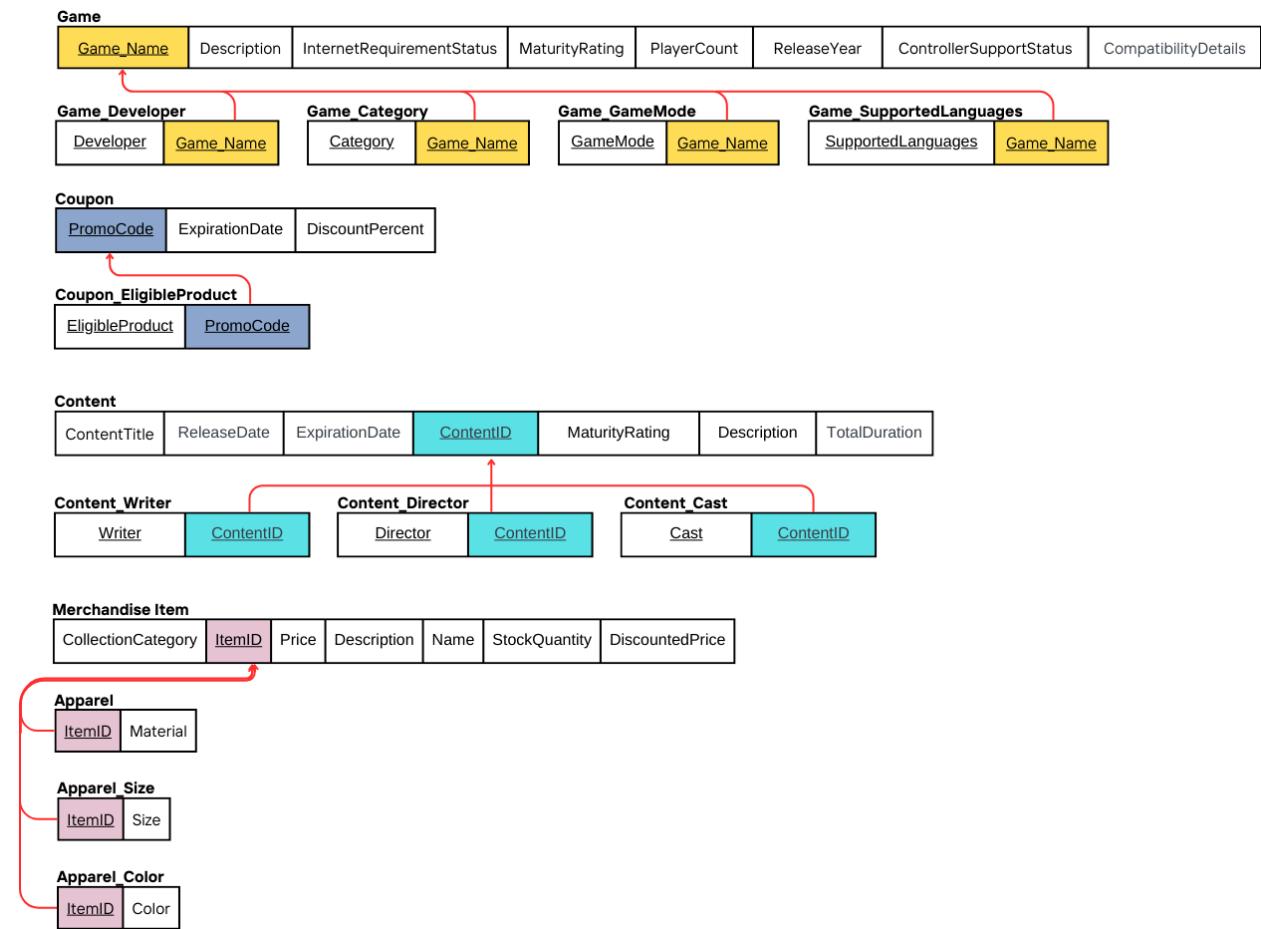
Step4: Binary 1:M Relationship



Step5: Binary M:N Relationship



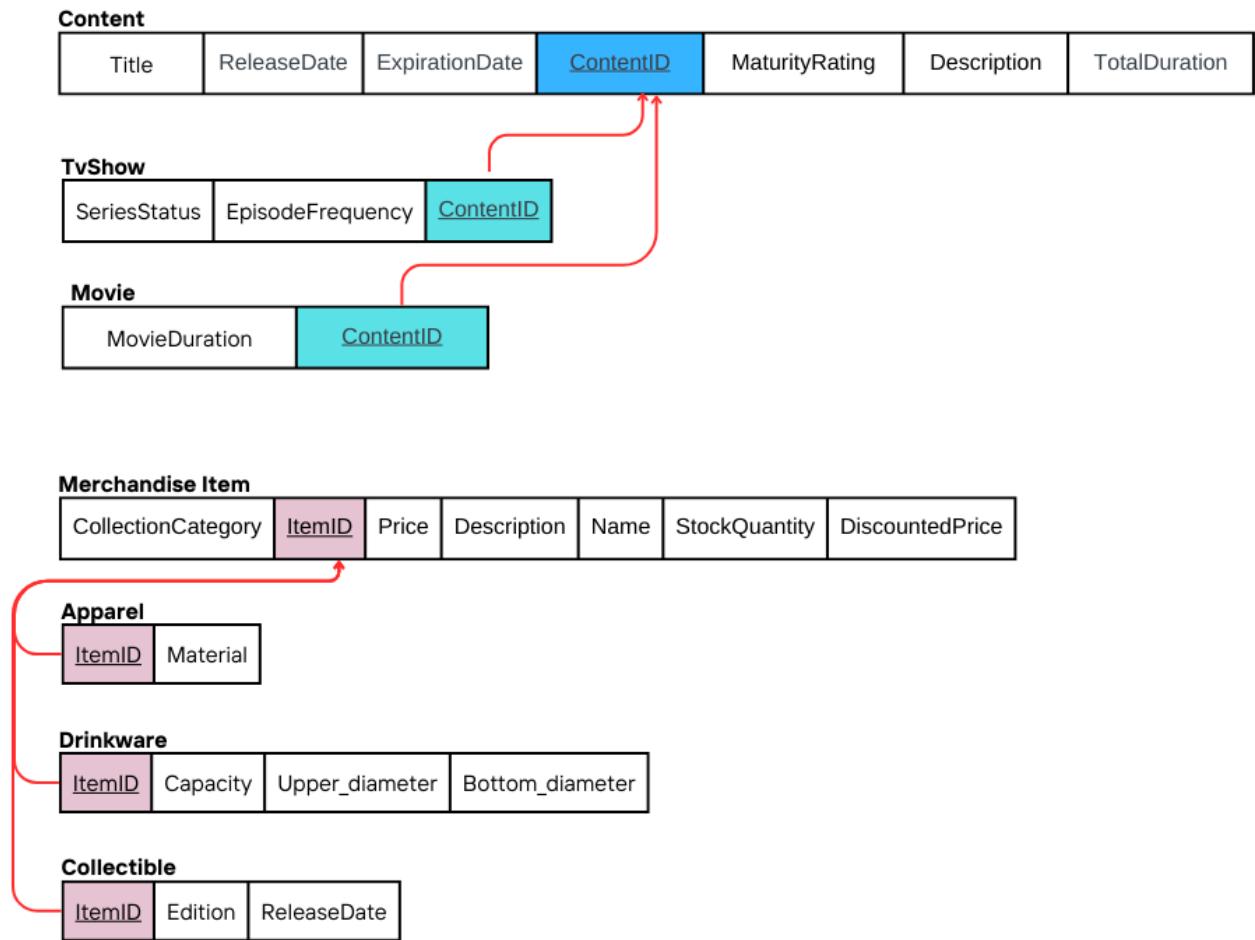
Step6:Multi-Valued Attribute



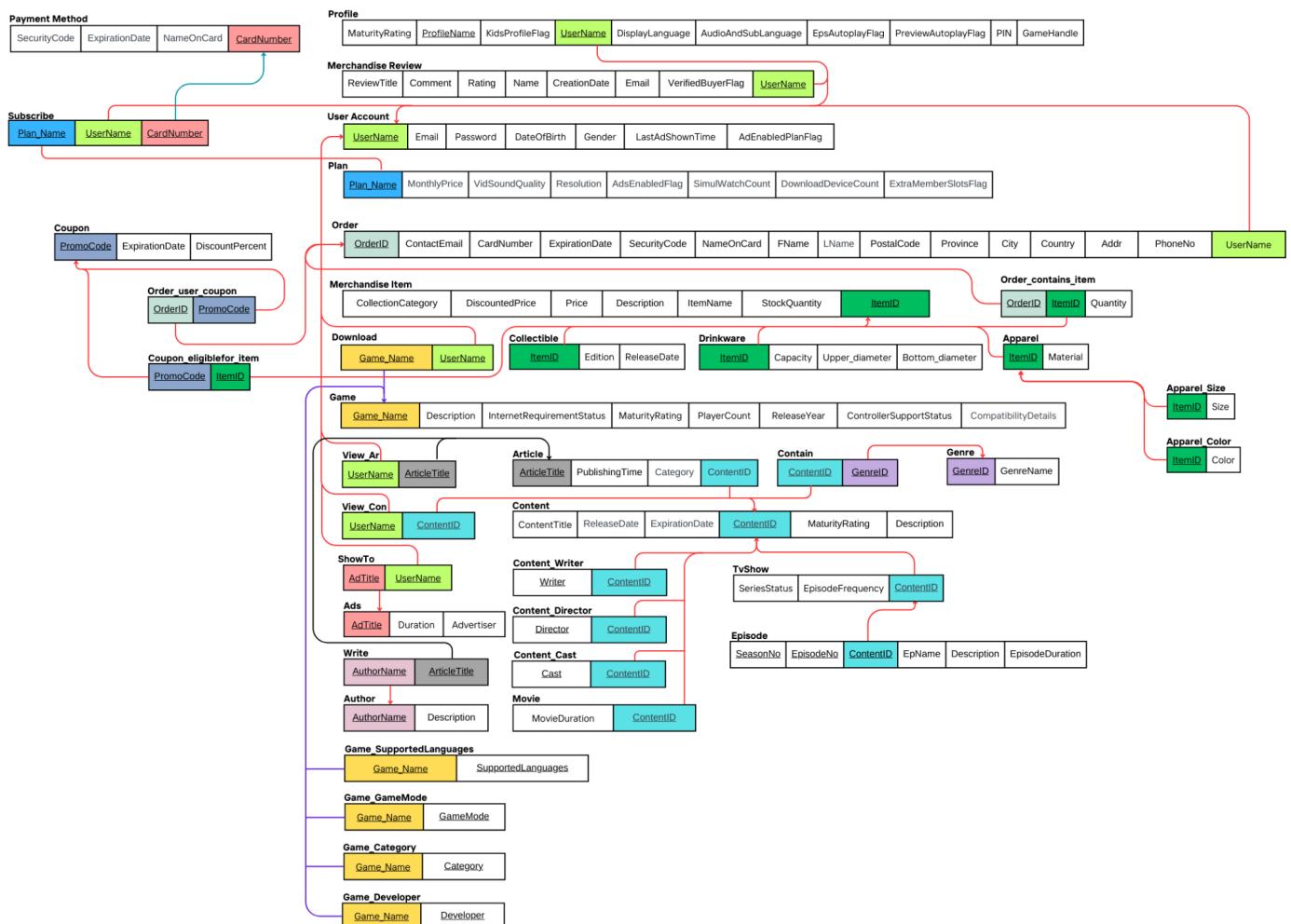
Step7: N-ary Relationship

There exists no N-ary relationship in this schema

Step8A: Specialization



Final Relational Schema



3 Data Dictionary

Table Name	Attribute Name	Contents	Type	Format	Nullable	Range	Key	FK Referenced Table
ads	ads_title	The title of an ad.	varchar(20)	xxxxxxxx			PK	
	duration	The ad's duration, measured in seconds.	int(11)	x		0 to 60		
	advertiser	The advertiser of this ad.	varchar(20)	xxxxxxx				
apparel	item_id	Item ID.	int(11)	x			PK,FK	item_id [merchandise_item]
	material	Description of the material the item is made of.	varchar(50)	xxxxxxx				
apparel_color	item_id	Item ID	int(11)	x			PK,FK	item_id [merchandise_item]
	color	Available color of the item.	varchar(10)	xxxxxxxx			PK	
apparel_size	item_id	Item ID	int(11)	x			PK,FK	item_id [apparel]
	size	Available size of the apparel.	varchar(4)	xxxx			PK	
article_tudum	article_title	An article's title	varchar(50)	xxxxxxxx			PK	
	publishing_time	The datetime the article is published on.	datetime	YYYY-MM-DD hh:mm:ss				
	category	The category of an article.	varchar(10)	xxxxxxxx				
	excerpt	Brief excerpt of the article.	varchar(300)	xxxxxxxx				
author	content_id	Content ID	int(11)	x			FK	content_id [content]
	author_name	An author's name.	varchar(50)	xxxxxxxx			PK	
	description	The description of an author.	varchar(300)	xxxxxxxx				
author_write_article	author_name	Author names	varchar(50)	xxxxxxxx			PK,FK	author_name [author]
	article_title	Article title	varchar(100)	xxxxxxxx			PK,FK	article_title [article_tudum]
collectible	item_id	Item ID.	int(11)	x			PK,FK	item_id [merchandise_item]
	edition	Edition number of the collection.	int(2)	x				
	release_date	Release date of the collection.	date	YYYY-MM-DD				
content	content_id	Content ID	int(11)	x			PK	
	content_title	Content title	varchar(70)	xxxxxxxx				
	release_date	Release date	date	YYYY-MM-DD				
	expiration_date	date	date	YYYY-MM-DD	Y			
	maturity_rating	Maturity rating	varchar(5)	xxxxxx				
	description	Description	text	xxxxxxxx				
content_cast	content_id	Content ID	int(11)	x			PK,FK	content_id [content]
	cast	Cast name	varchar(50)	xxxxxxxx			PK	
content_director	content_id	Content ID	int(11)	x			PK,FK	content_id [content]
	director	Director name	varchar(50)	xxxxxxxx			PK	
content_writer	content_id	Content ID	int(11)	x			PK,FK	content_id [content]
	writer	Writer name	varchar(50)	xxxxxxxx			PK	
coupon	promo_code	Promotion code.	varchar(10)	xxxxxxxx			PK	
	expiration_date	Coupon expiration date.	date	YYYY-MM-DD				
	discount_percent	Discount percent offered by coupon.	int(3)	x				
	is_valid_for_all_items	Mark if coupon is valid for all items.	tinyint(1)	x				order_id [merchandise_order]
coupon_eligiblefor_item	promo_code	Coupon's promo code	varchar(10)	xxxxxxxx			PK,FK	promo_code[coupon]
	item_id	Merchandise item ID.	int(20)	x			PK,FK	item_id [merchandise_item]
drinkware	item_id	Item ID	int(11)	x			PK,FK	item_id [merchandise_item]
	capacity	Capacity of drinkware, measured in ml.	int(5)	x				
	upper_diameter	Upper diameter of drinkware, measured in inches.	int(2)	x				
game	bottom_diameter	Bottom diameter of drinkware, measured in inches.	int(2)	x				
	game_name	A game's name	varchar(20)	xxxxxxxx			PK	
	description	Game's description	text	x				
	internet_requirement_status	Indicates if internet is required to play.	tinyint(1)	x				
	maturity_rating	Maturity rating	varchar(4)	xxxx				
	player_count	Min/max amount of players required to play.	varchar(5)	xxxxx				
	release_year	Release year	year(4)	YYYY				
controller_support_status	controller_support_status	Indicates if controller is supported.	tinyint(1)	x				
	compatibility_details	Device specs needed to play.	text	xxxxxxxx				

game_category	game_name	Game name	varchar(50)	xxxxxxxx			PK,FK	game_name [game]
	category	Game Category	varchar(20)	xxxxxxxx			PK	
game_developer	game_name	Game Name	varchar(50)	xxxxxxxx			PK,FK	game_name [game]
	developer	The game's developer	varchar(20)	xxxxxxxx			PK	
game_gamemode	game_name	Game name	varchar(50)	xxxxxxxx			PK,FK	game_name [game]
	game_mode	Different game modes of the game.	varchar(20)	xxxxxxxx			PK	
game_supported_languages	game_name	Game name	varchar(50)	xxxxxxxx			PK,FK	game_name [game]
	supported_lang	Supported languages of the game.	varchar(20)	xxxxxxxx			PK	
genre	genre_id	Genre ID	int(11)	x			PK	
	genre_name	Genre Name	varchar(30)	xxxxxxxx				
genre_contain_content	genre_id	Genre ID	int(11)	x			PK,FK	genre_id [genre]
	content_id	Content ID	int(11)	x			PK,FK	content_id [content]
merchandise_item	item_id	Item ID	int(11)	x			PK	
	collection_category	Collection category the item belongs to.	varchar(50)	xxxxxxxx				
	price	Price	decimal(10,2)	xxxxxxxx				
	description	Description	varchar(300)	x				
	item_name	Item Name	varchar(50)	xxxxxxxx		Y		
	discounted_price	Discounted price.	decimal(10,2)	xxxxxxxx.xx				
	stock_quantity	Stock quantity	int(11)	x				
merchandise_order	order_id	Order's ID	int(11)	x			PK	
	username	User name	varchar(20)	xxxxxxxx			FK	username [user_account]
	contact_email	Contact Email	varchar(254)	xxxxxxxx				
	card_number	Credit card number	varchar(19)	xxxxxxxx				
	expiration_date	Expiration date on credit card	char(5)	xxxxx				
	security_code	Security code on credit card	char(3)	xxx				
	name_on_card	Name on credit card	varchar(26)	xxxxxxxx				
	fname	First Name	int(50)	x				
	lname	Last Name	varchar(50)	xxxxxxxx				
	postal_code	Postal Code	varchar(10)	xxxxxxxx				
	province	Province	varchar(20)	xxxxxxxx				
	city	City	varchar(20)	xxxxxxxx				
	country	Country	varchar(20)	xxxxxxxx				
	addr	Extra address information	varchar(200)	xxxxxxxx		Y		
	order_date	Creation date of the order	date	YYYY-MM-DD				
	status	Order completion status	varchar(10)	xxxxxxxx	Shipping, Cancelled, Completed			
merchandise_review	review_id	Review ID	int(11)	xxxxxxxx			PK	
	username	User name	varchar(20)	xxxxxxxx			PK,FK	username [user_account]
	item_id	Item ID	int(11)	x			PK,FK	item_id [merchandise_item]
	review_title	Review Title	varchar(50)	xxxxxxxx				
	comment	Textual content of the review	varchar(200)	xxxxxxxx				
	rating	Rating given	int(1)	x				
	name	Displayed name on the review	varchar(20)	xxxxxxxx				
	creation_date	Creation date	date	YYYY-MM-DD				
	email	Email	varchar(254)	xxxxxxxx				
	verified_buyer_flag	Indicates if user is a verified buyer.	tinyint(1)	x				
movie	content_id	Content ID	int(11)	x			PK,FK	content_id [content]
	movieduration	Movie duration	time	HH:MM:SS				
order_contains_item	order_id	Order's ID	int(11)	x			PK,FK	order_id [merchandise_order]
	item_id	The ID of the merchandise item	int(11)	x			PK,FK	item_id [merchandise_item]
	quantity	The quantity of the item in the order	int(11)	x				
order_use_coupon	order_id	Order's ID	int(11)	x			PK,FK	order_id [merchandise_order]
	promo_code	The coupon code used in the order	varchar(10)	xxxxxxxx			PK,FK	promo_code [coupon]

payment_method	card_number	Credit Card Number	varchar(19)	xxxxxxxx		PK	
	username	Username of the user owning this payment payment method	varchar(20)	xxxxxxxx		FK	username [user_account]
	expiration_date	Expiration date of credit card	char(5)	xxxxx			
	security_code	Security code of credit card.	char(3)	xxx			
	name_on_card	Name on credit card.	varchar(26)	xxxxxxxx			
plan	plan_name	Plan name	char(10)	xxxxx		PK	
	monthly_price	Monthly price	decimal(4,2)	xxxx.xx			
	vid_sound_quality	Video and sound quality level	varchar(15)	xxxx			
	resolution	Content resolution	varchar(10)	xxxxxx			
	ads_enabled_flag	Mark if plan is ads enabled	tinyint(1)	x			
	simul_watch_count	Count how many devices are allowed to watch on this account	int(1)	x			
	download_device_count	Count how many devices are allowed to download content using this account	int(1)	x			
	extra_member_slots	Count how many other user accounts you can invite to become extra members	int(1)	x			
profile	profile_name	Profile name	varchar(10)	xxxxxxxx		PK	
	username	Username	varchar(20)	xxxxxxxx		PK,FK	username [user_account]
	maturity_rating	Maturity rating for profile content	varchar(3)	xxxxxx			
	kids_profile_status	Flag for kids-specific profile	tinyint(1)	x			
	display_language	Display language for user interface	varchar(3)	xxx			
	audio_sub_language	Preferred audio and subtitle language	varchar(10)	xxxxxxxx			
	eps_autoplay_flag	Flag for episode autoplay in series	tinyint(1)	x			
	preview_autoplay_flag	Flag for autoplay of content previews	tinyint(1)	x	Y		
	pin	PIN for profile access	int(6)	x	Y		
	game_handle	Game handle for Netflix Games	varchar(20)	xxxxxxxx			
subscription	subscription_id	Subscription ID	int(11)	x		PK	
	plan_name	Plan name	char(10)	xxxxxxxx		FK	plan_name [plan]
	username	User name	varchar(20)	xxxxxxxx		FK	username [user_account]
	card_number	Card number	varchar(19)	xxxxxxxx		FK	card_number [payment_method]
	subscription_date	Subscription date	date	YYYY-MM-DD			
	termination_date	Termination date. (Null if subscription is still active)	date	YYYY-MM-DD	Y		
tv_episode	content_id	Content ID of the episode	int(11)	x		PK,FK	content_id [tv_show]
	season_no	The number of season the episode belongs to	int(11)	x		PK	
	ep_no	The number of the episode	int(11)	x		PK	
	ep_name	An episode's name	varchar(100)	xxxxxxxx			
	description	The episode's description	varchar(300)	xxxxxxxx			
	episode_duration	An episode's duration in seconds	int(11)	x			
tv_show	content_id	Content ID representing the TV show	int(11)	x		PK,FK	content_id [content]
	series_status	The current status of the series	varchar(20)	xxxxxxxx			
	episode_release_frequency	Frequency of episode releases	varchar(20)	xxxxxxxx			
user_account	username	Username of the account holder	varchar(20)	xxxxxxxx		PK	
	email	Email associated with the account	varchar(254)	xxxxxxxx			
	password	Hashed password	int(64)	x			
	date_of_birth	Date of birth of the account holder	date	YYYY-MM-DD			
	gender	Gender of the user	varchar(1)	X	Y M,F		
user_download_game	download_id	Download ID	int(11)	x		PK	
	game_name	Game name	varchar(50)	xxxxxxxx		FK	game_name [game]
	username	User name	varchar(20)	xxxxxxxx		FK	username [user_account]
	download_date	The date the game is downloaded by the user	date	YYYY-MM-DD			
user_shown_ads	showads_id	A unique identifier for each ad shown.	int(11)	xxxxxxxx		PK	
	ads_title	The title of the advertisement shown to the user.	varchar(20)	xxxxxxxx		FK	ads_title [ads]
	user_name	The name of the user who was shown the ad.	varchar(20)	xxxxxxxx		FK	username [user_account]
	show_datetime	The date and time when the ad was displayed to the user.	datetime	YYYY-MM-DD			
	is_skipped	Indicates if the ad was skipped by the user	tinyint(1)	x	1,0		

user_view_article	viewarticle_id	View article ID	int(11)	x			PK	
	username	User name	varchar(20)	xxxxxx			FK	username [user_account]
	article_title	Article title	varchar(100)	xxxxxx			FK	article_title [article_tudum]
	article_view_date	The date the article is viewed	date	YYYY-MM-DD				
user_view_content	viewcontent_id	View Content ID	int(11)	x			PK,FK	username [user_account]
	username	User name	varchar(20)	xxxxxx			FK	
	content_id	Content ID	int(11)	x			FK	content_id [content]
	content_view_date	The date the content is viewed	date	YYYY-MM-DD				

4 Table Description & Implementation

4.1 Ads Table

The ads table holds information about the advertisement clips that are shown to Netflix users with an ad-enabled plan subscription.

4.1.1 Ads Table DDL script

```
CREATE TABLE ads (
    ads_title varchar(20) NOT NULL COMMENT 'The title of an ad.',
    duration int(11) NOT NULL COMMENT 'The ad"s duration, measured in seconds.',
    advertiser varchar(20) NOT NULL COMMENT 'The advertiser of this ad.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE ads
ADD PRIMARY KEY (ads_title);
```

4.1.2 Ads table example data

```
INSERT INTO ads (ads_title, duration, advertiser) VALUES
('Electric Cars', 30, 'EcoMotors'), ('Gadget Promotion', 5,
'TechWorld'), ('Healthy Snacks', 10, 'Foodies'),
('Holiday Travel', 60, 'GoGlobalTrips'), ('New Movie Release', 30,
'FilmStudio'),
('Streaming Deals', 60, 'StreamPlus'), ('Summer Collection', 45,
'FashionHub'),
('Super Sale', 10, 'ShopNow'), ('Tech Innovations', 45, 'InnovaTech'),
('Workout Gear', 5, 'FitStore');
```

4.2 Apparel Table

The apparel table is a specialization of merchandise items table. It holds information about the clothing type of items offered on the Netflix Shop.

4.2.1 Apparel table DDL script

```
CREATE TABLE apparel (
    item_id int(11) NOT NULL COMMENT 'Item ID.',
```

```

    material varchar(50) NOT NULL COMMENT 'Description of the material
the item is made of.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE apparel
ADD PRIMARY KEY (item_id);
ADD KEY fk_apparel_itemid (item_id);
ADD CONSTRAINT fk_apparel_itemid FOREIGN KEY (item_id)
REFERENCES merchandise_item (item_id);

```

4.2.2 Apparel table example data

```

INSERT INTO apparel (item_id, material) VALUES
(1, '100% Cotton'), (2, 'Polyester/Cotton Blend'), (3, 'Organic Cotton'), (4,
'Polyester');

```

4.3 Apparel_color Table

The apparel_color table is a table representing the apparel items' colors as a multivalued attribute.

4.3.1 Apparel_color Table DDL script

```

CREATE TABLE apparel_color (
item_id int(11) NOT NULL COMMENT 'Item ID',
color varchar(10) NOT NULL COMMENT 'Available color of the item.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE apparel_color
ADD PRIMARY KEY (item_id,color);
ADD CONSTRAINT fk_color_itemid FOREIGN KEY (item_id)
REFERENCES merchandise_item (item_id);

```

4.3.2 Apparel_color table example data

```

INSERT INTO apparel_color (item_id, color) VALUES
(1, 'Black'), (1, 'Red'), (2, 'Black'), (2, 'Gray'), (3, 'Blue'), (3, 'White'), (4,
'Green'), (4, 'Yellow');

```

4.4 Apparel_size Table

The apparel_size table is a table representing the apparel items' available sizes as a multivalued attribute.

4.4.1 Apparel_size Table DDL script

```
CREATE TABLE apparel_size (
    item_id int(11) NOT NULL COMMENT 'Item ID',
    size varchar(4) NOT NULL COMMENT 'Available size of the apparel.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE apparel_size
    ADD PRIMARY KEY (item_id,size);
    ADD CONSTRAINT fk_size_itemid FOREIGN KEY (item_id)
REFERENCES apparel (item_id);
```

4.4.2 Apparel_size table example data

```
INSERT INTO apparel_size (item_id, size) VALUES
(1, 'L'),(1, 'M'),(1, 'S'),(2, 'L'),(2, 'M'),(2, 'XL'),(3, 'L'),(3, 'M'),
(3, 'S'),(4, 'L'),(4, 'XL');
```

4.5 Article_tudum Table

The article_tudum table holds information about the written articles featured on Netflix Tudum, Netflix's online magazine service.

4.5.1 Article_tudum Table DDL script

```
CREATE TABLE article_tudum (
    article_title varchar(100) NOT NULL COMMENT 'An article's title',
    publishing_time datetime NOT NULL COMMENT 'The datetime the article is
published on.',
    category varchar(20) NOT NULL COMMENT 'The category of an article.',
    excerpt varchar(300) NOT NULL COMMENT 'Brief excerpt of the article.',
    content_id int(11) NOT NULL COMMENT 'Content ID'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE apparel_size
    ADD PRIMARY KEY (item_id,size);
    ADD CONSTRAINT fk_size_itemid FOREIGN KEY (item_id) REFERENCES
apparel (item_id);
```

4.5.2 Article_tudum Table example data

```
INSERT INTO article_tudum (article_title, publishing_time, category, excerpt, content_id) VALUES  
('Sadie Sink on Max's Trip to the Upside Down in 'Stranger Things' Season 4', '2022-06-29 11:30:00', 'meet the cast', 'Sadie Sink discusses Max\'s journey in Season 4, including her painful experiences and the significance of \"Running Up That Hill.\'", 5), ('The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two Volumes', '2022-04-06 10:00:00', 'deep dive', 'The Duffer Brothers discuss the decision to split Season 4 into two volumes, teasing a grand finale as long as a movie.', 5), ('The Queen\'s Gambit Cast, News, Videos and more', '2020-10-23 10:30:00', 'news', 'An overview of \"The Queen\'s Gambit,\" including cast information, news, and videos.', 3), ('Who Is Eddie? Steve and Dustin\'s 'Stranger Things' Friendship', '2022-06-22 09:00:00', 'meet the cast', 'Joe Keery, Gaten Matarazzo, and new star Joe Quinn preview the fourth season and their beloved bromance.', 5), ('Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far', '2022-07-01 13:00:00', 'news', 'Information about the new animated series set in the Stranger Things universe, in partnership with Shawn Levy\'s 21 Laps and Eric Robles.', 5), ('Stranger Things' Cast Red Carpet Premiere Photos', '2022-05-17 17:30:00', 'meet the cast', 'Photos from the red carpet premieres in Paris, Berlin, and Madrid, featuring stars like Noah Schnapp, Natalia Dyer, Charlie Heaton, and Priah Ferguson.', 5), ('Stranger Things' Season 4 Easter Eggs', '2022-06-04 12:45:00', 'explainer', 'A breakdown of Easter eggs in Season 4, including references to Kobe Bryant and Tom Cruise\'s \"Risky Business\" haircut.', 5), ('Stranger Things' Season 4 Episode Length Revealed', '2022-05-16 14:00:00', 'news', 'Details about the episode lengths for Season 4, including 7 episodes in Volume 1 and 2 episodes in Volume 2, totaling approximately 13 hours.', 5), ('Stranger Things' Season 4 First Look Photos', '2022-05-18 16:00:00', 'deep dive', 'First look photos from Season 4, showcasing the darkest and boldest chapter yet.', 5), ('Stranger Things' Season 4 Release Date Announced', '2022-02-17 08:30:00', 'news', 'The Duffer Brothers announce the release date for Season 4 and confirm that the series will conclude with Season 5.', 5);
```

4.6 Author Table

The author table represents the author of articles on Netflix Tudum.

4.6.1 Author Table DDL script

```
CREATE TABLE author (  
    author_name varchar(50) NOT NULL COMMENT 'An author\'s name.',
```

```
description varchar(300) NOT NULL COMMENT 'The description of an author.'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
ALTER TABLE author  
ADD PRIMARY KEY (author_name);
```

4.6.2 Author table example data

```
INSERT INTO author (author_name, description) VALUES  
('Chris Taylor', 'A journalist who covers film, TV, and celebrity culture for multiple platforms.'),  
('Daniel Lee', 'An author known for writing about the evolution of TV series and their cultural impact.'),  
('David Martinez', 'A film critic who writes extensively about new releases and their impact on the industry.'),  
('Emily Johnson', 'A writer with a passion for in-depth analyses of popular TV series and films.'),  
('James Moore', 'A seasoned writer with expertise in Netflix original content and the world of streaming.'),  
('Jane Smith', 'A pop culture critic and journalist, specializing in Netflix original content.'),  
('Jessica Clark', 'A journalist who specializes in casting news and trends in TV and film.'),  
('John Doe', 'A seasoned entertainment writer with a focus on TV shows and movies.'),  
('Laura Harris', 'A pop culture enthusiast and journalist with a focus on Netflix originals.'),  
('Maria Garcia', 'An entertainment journalist with a focus on interviews and content analysis.'),  
('Michael Brown', 'A long-time contributor to online entertainment publications, covering TV and film.'),  
('Olivia Davis', 'A writer with a focus on behind-the-scenes details and production insights.'),  
('Robert White', 'A TV and film writer known for his reviews and deep-dive content on Netflix shows.'),  
('Sarah Williams', 'An entertainment writer known for her insightful reviews and interviews with actors.'),  
('Sophia Anderson', 'An entertainment writer who dives deep into the stories behind popular media.');
```

4.7 Author_write_article Table

The author_write_article table represents a M:N “write” relationship between articles and the authors.

4.7.1 Author_write_article Table DDL script

```
CREATE TABLE author_write_article (
author_name varchar(50) NOT NULL COMMENT 'Author names',
article_title varchar(100) NOT NULL COMMENT 'Article title'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE author_write_article
ADD PRIMARY KEY (author_name,article_title),
ADD KEY fk_write_articletitle (article_title);
ADD CONSTRAINT fk_write_articletitle FOREIGN KEY (article_title)
REFERENCES article_tudum (article_title),
ADD CONSTRAINT fk_write_authorname FOREIGN KEY (author_name)
REFERENCES author (author_name);
```

4.7.2 Author_write_article table example data

```
INSERT INTO author_write_article (author_name, article_title) VALUES
('John Doe', 'Sadie Sink on Max's Trip to the Upside Down in \'Stranger Things\' Season 4'),
('Maria Garcia', 'Sadie Sink on Max's Trip to the Upside Down in \'Stranger Things\' Season 4'),
('Jane Smith', 'The Duffer Brothers on Why \'Stranger Things\' Season 4 Is Two Volumes'),
('John Doe', 'The Duffer Brothers on Why \'Stranger Things\' Season 4 Is Two Volumes'),
('Chris Taylor', 'The Queen\'s Gambit Cast, News, Videos and more'),
('Sarah Williams', 'The Queen\'s Gambit Cast, News, Videos and more'),
('Emily Johnson', 'Who Is Eddie? Steve and Dustin\'s \'Stranger Things\' Friendship'),
('Michael Brown', 'Who Is Eddie? Steve and Dustin\'s \'Stranger Things\' Friendship'),
('James Moore', "Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far"),
('Jessica Clark', "Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far"),
('Daniel Lee', "Stranger Things' Cast Red Carpet Premiere Photos"),
('David Martinez', "Stranger Things' Cast Red Carpet Premiere Photos"),
('Olivia Davis', "Stranger Things' Cast Red Carpet Premiere Photos"),
```

('Sophia Anderson', "Stranger Things' Cast Red Carpet Premiere Photos"),
 ('John Doe', "Stranger Things' Season 4 Easter Eggs'),
 ('Laura Harris', "Stranger Things' Season 4 Easter Eggs'),
 ('Maria Garcia', "Stranger Things' Season 4 Easter Eggs'),
 ('Robert White', "Stranger Things' Season 4 Easter Eggs'),
 ('Chris Taylor', "Stranger Things' Season 4 Episode Length Revealed'),
 ('Laura Harris', "Stranger Things' Season 4 Episode Length Revealed'),
 ('Robert White', "Stranger Things' Season 4 Episode Length Revealed'),
 ('Sarah Williams', "Stranger Things' Season 4 Episode Length Revealed'),
 ('Daniel Lee', "Stranger Things' Season 4 First Look Photos'),
 ('David Martinez', "Stranger Things' Season 4 First Look Photos'),
 ('Olivia Davis', "Stranger Things' Season 4 First Look Photos'),
 ('Sophia Anderson', "Stranger Things' Season 4 First Look Photos'),
 ('Emily Johnson', "Stranger Things' Season 4 Release Date Announced'),
 ('James Moore', "Stranger Things' Season 4 Release Date Announced'),
 ('Jessica Clark', "Stranger Things' Season 4 Release Date Announced'),
 ('Michael Brown', "Stranger Things' Season 4 Release Date Announced");

4.8 Collectible Table

The collectible is a specialization of the merchandise item table. It represents collectible items, which are items made primarily for displaying purposes, that are offered on Netflix Shop.

4.8.1 Collectible Table DDL script

```

CREATE TABLE collectible (
  item_id int(11) NOT NULL COMMENT 'Item ID.',
  edition int(2) NOT NULL COMMENT 'Edition number of the collection.',
  release_date date NOT NULL COMMENT 'Release date of the collection.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE collectible
  ADD PRIMARY KEY (item_id);
  ADD CONSTRAINT fk_collectible_itemid FOREIGN KEY (item_id)
  REFERENCES merchandise_item (item_id);
  
```

4.8.2 Collectible table example data

```

INSERT INTO collectible (item_id, edition, release_date) VALUES
(9, 1, '2023-04-15'),(10, 2, '2023-06-01'),(11, 1, '2023-09-20'),(12, 3,
'2023-11-10');
  
```

4.9 Content Table

The content table represents the information of shows offered on Netflix. It can be specialized into two types: 'movie', and 'tv_show'.

4.9.1 Content Table DDL script

```
CREATE TABLE content (
    content_id int(11) NOT NULL COMMENT 'Content ID',
    content_title varchar(70) NOT NULL COMMENT 'Content title',
    release_date date NOT NULL COMMENT 'Release date',
    expiration_date date DEFAULT NULL COMMENT 'Expiration date',
    maturity_rating varchar(5) NOT NULL COMMENT 'Maturity rating',
    description text NOT NULL COMMENT 'Description'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE content
    ADD PRIMARY KEY (content_id);
```

4.9.2 Content table example data

```
INSERT INTO content (content_id, content_title, release_date, expiration_date,
maturity_rating, description) VALUES
```

```
(1, 'A Killer Paradox', '2024-01-15', NULL, 'TV-MA', 'A 2024 thriller where a retail worker inadvertently kills a serial killer, leading to unexpected consequences.'),
```

```
(2, 'The Lincoln Lawyer', '2022-05-13', NULL, 'TV-14', 'A legal drama following a defense attorney who runs his practice from a Lincoln Town Car, navigating complex cases and personal challenges.'),
```

```
(3, 'The Queen's Gambit', '2020-10-23', NULL, 'TV-MA', 'A period drama about a young chess prodigy's rise in the male-dominated world of competitive chess during the Cold War era.'),
```

```
(4, 'One Piece', '2023-08-31', NULL, 'TV-14', 'A live-action adaptation of the popular manga and anime, following Monkey D. Luffy and his crew in search of the legendary One Piece treasure.'),
```

```
(5, 'Stranger Things', '2016-07-15', NULL, 'TV-14', 'A sci-fi horror series set in the 1980s, where a group of kids uncover supernatural mysteries in their small town.'),
```

```
(6, 'The Crown', '2016-11-04', NULL, 'TV-MA', 'A historical drama chronicling the reign of Queen Elizabeth II, exploring political and personal events that shaped the second half of the 20th century.'),
```

```
(7, 'Bridgerton', '2020-12-25', '2025-12-25', 'TV-MA', 'A romantic drama set in Regency-era London, focusing on the Bridgerton family and their entanglements in high society.'),
```

(8, 'Money Heist', '2017-05-02', NULL, 'TV-MA', 'A Spanish heist crime drama where a group of robbers, led by \'The Professor,\' execute meticulously planned heists.'),

(9, 'The Witcher', '2019-12-20', NULL, 'TV-MA', 'A fantasy series following Geralt of Rivia, a monster hunter navigating a turbulent world filled with magic and political intrigue.'),

(10, 'Squid Game', '2021-09-17', NULL, 'TV-MA', 'A South Korean survival drama where contestants in dire financial situations compete in deadly children\'s games for a massive cash prize.'),

(11, 'The Killer', '2024-10-30', NULL, 'TV-MA', 'A 2024 thriller directed by David Fincher, following an assassin who begins to question his own sanity.'),

(12, 'The Harder They Fall', '2021-10-22', NULL, 'R', 'A 2021 Western featuring an all-Black cast, where an outlaw reunites his gang to seek revenge.'),

(13, 'Red Notice', '2021-11-05', NULL, 'PG-13', 'A 2021 action-comedy starring Dwayne Johnson, Gal Gadot, and Ryan Reynolds, involving an Interpol agent tracking the world\'s most wanted art thief.'),

(14, 'Don\'t Look Up', '2021-12-24', NULL, 'R', 'A 2021 satirical science fiction film where two astronomers attempt to warn humanity about an approaching comet that will destroy Earth.'),

(15, 'The Power of the Dog', '2021-11-17', NULL, 'R', 'A 2021 Western drama exploring themes of love, grief, and resentment between two brothers on a Montana ranch.'),

(16, 'Bird Box', '2018-12-14', '2024-12-14', 'TV-MA', 'A post-apocalyptic thriller where survivors must navigate a world where seeing mysterious creatures leads to death.'),

(17, 'Marriage Story', '2019-11-06', NULL, 'R', 'A 2019 drama about a couple undergoing a painful divorce, examining their emotions and the toll it takes on their family.'),

(18, 'The Old Guard', '2020-07-10', NULL, 'R', 'A 2020 action movie about immortal warriors who fight to protect humanity, starring Charlize Theron.'),

(19, 'The Irishman', '2019-11-27', NULL, 'R', 'A 2019 epic crime film directed by Martin Scorsese, focusing on the life of a mob hitman and his involvement with organized crime.'),

(20, 'Enola Holmes', '2020-09-23', NULL, 'PG-13', 'A 2020 mystery-adventure film about Sherlock Holmes\'s younger sister, Enola, who uncovers a conspiracy while searching for her missing mother.');

4.10 Content_cast Table

The content_cast table represents the cast members of each show as a multivalued attribute.

4.10.1 Content_cast Table DDL script

```
CREATE TABLE content_cast (
    content_id int(11) NOT NULL COMMENT 'Content ID',
    cast varchar(50) NOT NULL COMMENT 'Cast name'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE content_cast
    ADD PRIMARY KEY (content_id,cast);
    ADD CONSTRAINT fk_cast_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id);
```

4.10.2 Content_cast table example data

```
INSERT INTO content_cast (content_id, cast) VALUES
(1, 'Michael Fassbender'),(1, 'Tilda Swinton'),(2, 'Idris Elba'),
(2, 'Jonathan Majors'),(3, 'Dwayne Johnson'),(3, 'Gal Gadot'),
(3, 'Ryan Reynolds'),(4, 'Jennifer Lawrence'),(4, 'Leonardo DiCaprio'),
(4, 'Meryl Streep'),(5, 'Benedict Cumberbatch'),(5, 'Jesse Plemons'),
(5, 'Kirsten Dunst'),(6, 'John Malkovich'),(6, 'Sandra Bullock'),
(6, 'Trevante Rhodes'),(7, 'Adam Driver'),(7, 'Laura Dern'),
(7, 'Scarlett Johansson'),(8, 'Charlize Theron'),(8, 'Chiwetel Ejiofor'),
(8, 'KiKi Layne'),(9, 'Al Pacino'),(9, 'Joe Pesci'),(9, 'Robert De Niro'),
(10, 'Henry Cavill'),(10, 'Millie Bobby Brown'),(10, 'Sam Claflin'),
(11, 'Michael Fassbender'),(11, 'Tilda Swinton'),(12, 'Idris Elba'),
(12, 'Jonathan Majors'),(13, 'Dwayne Johnson'),(13, 'Gal Gadot'),
(13, 'Ryan Reynolds'),(14, 'Jennifer Lawrence'),(14, 'Leonardo DiCaprio'),
(14, 'Meryl Streep'),(15, 'Benedict Cumberbatch'),(15, 'Jesse Plemons'),
(15, 'Kirsten Dunst'),(16, 'John Malkovich'),(16, 'Sandra Bullock'),
(16, 'Trevante Rhodes'),(17, 'Adam Driver'),(17, 'Laura Dern'),
(17, 'Scarlett Johansson'),(18, 'Charlize Theron'),(18, 'Chiwetel Ejiofor'),
(18, 'KiKi Layne'),(19, 'Al Pacino'),(19, 'Joe Pesci'),(19, 'Robert De Niro'),
(20, 'Henry Cavill'),(20, 'Millie Bobby Brown'),(20, 'Sam Claflin');
```

4.11 Content_Director Table

The content director table represents the directors on each show as a multivalued attribute.

4.11.1 Content_Director DDL Script

```
CREATE TABLE content_director (
```

```

    content_id int(11) NOT NULL COMMENT 'Content ID',
    director varchar(50) NOT NULL COMMENT 'Director name'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE content_director
ADD PRIMARY KEY (content_id,director);
ALTER TABLE content_director
ADD CONSTRAINT fk_director_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id);

```

4.11.2 Content_Director table example data

```

INSERT INTO content_director (content_id, director) VALUES
(1, 'Cary Joji Fukunaga'),(1, 'David Fincher'),(2, 'Jeymes Samuel'),(3, 'Rawson
Marshall Thurber'),(4, 'Adam McKay'),
(5, 'Jane Campion'),(6, 'Susanne Bier'),(7, 'Noah Baumbach'),(8, 'Gina
Prince-Bythewood'),(9, 'Martin Scorsese'),
(10, 'Harry Bradbeer'),(11, 'David Fincher'),(12, 'Jeymes Samuel'),(13, 'Rawson
Marshall Thurber'),
(14, 'Adam McKay'),(15, 'Jane Campion'),(16, 'Susanne Bier'),(17, 'Noah
Baumbach'),(18, 'Gina Prince-Bythewood'),
(19, 'Martin Scorsese'),(20, 'Harry Bradbeer');

```

4.12 Content_Writer Table

The content writer table represents the writers of each show as a multivalued attribute.

4.12.1 Content_Writer DDL Script

```

CREATE TABLE content_writer (
content_id int(11) NOT NULL COMMENT 'Content ID',
writer varchar(50) NOT NULL COMMENT 'Writer name'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE content_writer
ADD PRIMARY KEY (content_id,writer);
ALTER TABLE content_writer
ADD CONSTRAINT fk_writer_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id);

```

4.12.2 Content_Writer table example data

```
INSERT INTO content_writer (content_id, writer) VALUES  
(1, 'Andrew Kevin Walker'), (1, 'David Fincher'), (2, 'Jeymes Samuel'), (3, 'Rawson Marshall Thurber'), (4, 'Adam McKay'), (4, 'David Sirota'), (5, 'Jane Campion'), (6, 'Susanne Bier'), (7, 'Noah Baumbach'), (8, 'Gina Prince-Bythewood'),  
(8, 'Greg Rucka'), (9, 'Jay Cocks'), (9, 'Steven Zaillian'), (10, 'Jack Thorne'), (11, 'David Fincher'), (12, 'Jeymes Samuel'),  
(13, 'Rawson Marshall Thurber'), (14, 'Adam McKay'), (14, 'David Sirota'), (15, 'Jane Campion'), (16, 'Susanne Bier'),  
(17, 'Noah Baumbach'), (18, 'Gina Prince-Bythewood'), (18, 'Greg Rucka'),  
(19, 'Jay Cocks'), (19, 'Steven Zaillian'), (20, 'Jack Thorne');
```

4.13 Coupon Table

The coupon table represents the coupon used

4.13.1 Coupon DDL Script

```
CREATE TABLE coupon (  
    promo_code varchar(10) NOT NULL COMMENT 'Promotion code.',  
    expiration_date date NOT NULL COMMENT 'Coupon expiration date.',  
    discount_percent int(3) NOT NULL COMMENT 'Discount percent offered by  
    coupon.',  
    is_valid_for_all_items tinyint(1) NOT NULL COMMENT 'Mark if coupon is valid  
    for all items.'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
ALTER TABLE coupon  
ADD PRIMARY KEY (promo_code);
```

4.13.2 Coupon table example data

```
INSERT INTO coupon (promo_code, expiration_date, discount_percent,  
is_valid_for_all_items) VALUES  
('BLACKFRI', '2024-11-29', 25, 1), ('BRIDGE15', '2024-11-30', 15, 0), ('FALLDEAL',  
'2024-11-30', 10, 1),  
('NEWYEAR5', '2024-01-01', 5, 1), ('SPRING25', '2024-03-31', 25, 1), ('SQUID20',  
'2024-10-31', 20, 0),  
('STRANGE10', '2024-12-31', 10, 0), ('SUMMER10', '2024-06-30', 10,  
1), ('WITCHER5', '2024-12-31', 5, 0),  
('XMAS2024', '2024-12-25', 30, 1);
```

4.14 Coupon_eligiblefor_item Table

The coupon_eligiblefor_item holds information on which merchandise item is eligible for which coupon on Netflix Shop.

4.14.1 coupon_eligiblefor_item DDL Script

```
CREATE TABLE coupon_eligiblefor_item (
    promo_code varchar(10) NOT NULL COMMENT 'Coupon's promo code',
    item_id int(20) NOT NULL COMMENT 'Merchandise item ID.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE coupon_eligiblefor_item
    ADD PRIMARY KEY (promo_code,item_id),
    ADD KEY fk_validfor_itemid (item_id);
    ADD CONSTRAINT fk_validfor_itemid FOREIGN KEY (item_id) REFERENCES
merchandise_item (item_id),
    ADD CONSTRAINT fk_validfor_promocode FOREIGN KEY (promo_code)
REFERENCES coupon (promo_code);
```

4.14.2 coupon_eligiblefor_item table example data

```
INSERT INTO coupon_eligiblefor_item (promo_code, item_id) VALUES
('STRANGE10', 1),('WITCHER5', 2),('BRIDGE15', 3),('SQUID20', 4),
('STRANGE10', 5),('WITCHER5', 6),('BRIDGE15', 7),('SQUID20', 8),
('STRANGE10', 9),('WITCHER5', 10),('BRIDGE15', 11),('SQUID20', 12);
```

4.15 Drinkware Table

The drinkware table is a specialization of the merchandise item. It represents the drinkware items offered on Netflix Shop, such as water bottles and cups.

4.15.1 Drinkware DDL Script

```
CREATE TABLE drinkware (
    item_id int(11) NOT NULL,
    capacity int(5) NOT NULL COMMENT 'Capacity of drinkware, measured in ml.',
    upper_diameter int(2) NOT NULL COMMENT 'Upper diameter of drinkware,
measured in inches.',
    bottom_diameter int(2) NOT NULL COMMENT 'Bottom diameter of drinkware,
measured in inches.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE drinkware
```

```
ADD PRIMARY KEY (item_id);
ALTER TABLE drinkware
  ADD CONSTRAINT fk_drinkware_itemid FOREIGN KEY (item_id)
  REFERENCES merchandise_item (item_id);
```

4.15.2 DrinkWare table example data

```
INSERT INTO drinkware (item_id, capacity, upper_diameter, bottom_diameter)
VALUES
(5, 350, 3, 3),(6, 450, 3, 2),(7, 500, 3, 3),(8, 750, 3, 2);
```

4.16 Game Table

The game table represents games offered on the Netflix Games program.

4.16.1 Game DDL Script

```
CREATE TABLE game (
  game_name varchar(50) NOT NULL COMMENT 'A game"s name',
  description text NOT NULL COMMENT 'Game"s description',
  internet_requirement_status tinyint(1) NOT NULL COMMENT 'Is internet
required to play or not?',
  maturity_rating varchar(4) NOT NULL COMMENT 'Maturity rating',
  player_count varchar(5) NOT NULL COMMENT 'Min/max amount of player
required to play the game.',
  release_year year(4) NOT NULL COMMENT 'Release year',
  controller_support_status tinyint(1) NOT NULL COMMENT 'Is controller
supported or not?',
  compatibility_details text NOT NULL COMMENT 'Short description of the device
specs needed to play this game.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4.16.2 Game table example data

```
INSERT INTO game (game_name, description, internet_requirement_status,
maturity_rating, player_count, release_year, controller_support_status,
compatibility_details) VALUES
('Asphalt Xtreme', 'Off-road racing game with various vehicle classes and
multiplayer.', 1, 'E10+', '1-8', 2021, 1, 'Needs stable internet and devices with
good graphics support.'),
('Exploding Kittens', 'Card game adaptation with humorous strategy elements.', 0,
'E', '2-5', 2021, 0, 'Requires Android 5.0 or iOS 11 and up.'),
```

('Heads Up!', 'A fun party game where players guess words based on clues.', 1, 'E', '2+', 2018, 0, 'Requires iOS or Android with camera access.'),
 ('Hextech Mayhem', 'Rhythm-based runner set in the League of Legends universe.', 1, 'T', '1', 2021, 1, 'Compatible with most modern devices with touchscreen.'), ('Into the Breach', 'Turn-based strategy where players control mechs to defend cities from aliens.', 0, 'E10+', '1', 2018, 1, 'Compatible with mid-range devices, iOS and Android.'),

('Kentucky Route Zero', 'A magical realist adventure game exploring a mysterious highway.', 0, 'T', '1', 2019, 1, 'Requires Android 6.0 or iOS 12 and up.'),
 ('Moonlighter', 'Action RPG with dungeon crawling and shop management.', 0, 'T', '1', 2018, 1, 'Runs on most Android 6.0+ and iOS 12+ devices.'),
 ('Oxenfree', 'Supernatural thriller adventure game involving dialogue choices.', 0, 'T', '1', 2016, 1, 'Requires Android 5.0 or iOS 11 and up.'),
 ('Shatter Remastered', 'A modernized version of a brick-breaking game with power-ups.', 0, 'E', '1', 2022, 1, 'Compatible with iOS and Android devices from 2018 onward.'),
 ('Stranger Things: 1984', 'Pixel-art action game set in the Stranger Things universe.', 0, 'T', '1', 2020, 1, 'Playable on low-end Android and iOS devices.');

4.17 Game_Category Table

The game category table represents how games on Netflix may belong to multiple categories, thus being represented by a multivalued attribute table.

4.17.1 Game_Category Table DDL Script

```
CREATE TABLE game_category (
    game_name varchar(50) NOT NULL COMMENT 'Game name',
    category varchar(20) NOT NULL COMMENT 'Game Category'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE game_category
    ADD PRIMARY KEY (category,game_name),
    ADD KEY fk_mtval_gamecategory (game_name);
```

4.17.2 Game_Category Table Example Data

```
INSERT INTO game_category (game_name, category) VALUES
('Asphalt Xtreme', 'Multiplayer'),('Asphalt Xtreme', 'Racing'),
('Exploding Kittens', 'Card Game'),('Exploding Kittens', 'Casual'),
```

```

('Heads Up!', 'Party Game'), ('Heads Up!', 'Trivia'),
('Hextech Mayhem', 'Action'), ('Hextech Mayhem', 'Rhythm'),
('Into the Breach', 'Strategy'), ('Into the Breach', 'Turn-Based'),
('Kentucky Route Zero', 'Adventure'), ('Kentucky Route Zero', 'Narrative'),
('Moonlighter', 'Action RPG'), ('Moonlighter', 'Dungeon Crawler'),
('Oxenfree', 'Adventure'), ('Oxenfree', 'Thriller'),
('Shatter Remastered', 'Arcade'), ('Shatter Remastered', 'Puzzle'),
('Stranger Things: 1984', 'Action'), ('Stranger Things: 1984', 'Retro');

```

4.18 Game_Developer Table

The game developer table represents the developers on a game as a multivalued attribute.

4.18.1 Game_Developer Table DDL Script

```

CREATE TABLE game_developer (
    game_name varchar(50) NOT NULL COMMENT 'Game name',
    developer varchar(50) NOT NULL COMMENT 'The game''s developer'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE game_developer
    ADD PRIMARY KEY (developer,game_name),
    ADD KEY fk_mtval_gamedev (game_name);
ALTER TABLE game_developer
    ADD CONSTRAINT fk_mtval_gamedev FOREIGN KEY (game_name)
REFERENCES game (game_name);

```

4.18.2 Game_Developer Table Example Data

```

INSERT INTO game_developer (game_name, developer) VALUES
('Asphalt Xtreme', 'Gameloft'), ('Exploding Kittens', 'Exploding Kittens, Inc.'),
('Heads Up!', 'Warner Bros. Games'), ('Hextech Mayhem', 'Choice Provisions'),
('Into the Breach', 'Subset Games'), ('Kentucky Route Zero', 'Cardboard Computer'),
('Moonlighter', 'Digital Sun'), ('Oxenfree', 'Night School Studio'),
('Shatter Remastered', 'PikPok'), ('Stranger Things: 1984', 'BonusXP');

```

4.19 Game_Gamemode Table

The game_gamemode table represents the different game modes for games on Netflix as a multivalued attribute table.

4.19.1 Game_Gamemode Table DDL Script

```
CREATE TABLE game_gamemode (
    game_name varchar(50) NOT NULL COMMENT 'Game name',
    game_mode varchar(20) NOT NULL COMMENT 'The different game modes of
the game.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE game_gamemode
    ADD PRIMARY KEY (game_name,game_mode);
ALTER TABLE game_gamemode
    ADD CONSTRAINT fk_mtval_gamemode FOREIGN KEY (game_name)
REFERENCES game (game_name);
```

4.19.2 Game_Gamemode Table Example Data

```
INSERT INTO game_gamemode (game_name, game_mode) VALUES
('Asphalt Xtreme', 'Multiplayer'),('Asphalt Xtreme', 'Single Player'),
('Exploding Kittens', 'Multiplayer'),('Exploding Kittens', 'Single Player'),
('Heads Up!', 'Multiplayer'),('Hextech Mayhem', 'Single Player'),
('Into the Breach', 'Single Player'),('Kentucky Route Zero', 'Single Player'),
('Moonlighter', 'Single Player'),('Oxenfree', 'Single Player'),
('Shatter Remastered', 'Single Player'),('Stranger Things: 1984', 'Single Player');
```

4.20 Game_Supported_Languages Table

The game_supported_languages table represents the multiple languages a game may support as a multivalued attribute table.

4.20.1 Game_Supported_Languages Table DDL Script

```
CREATE TABLE game_supported_languages (
    game_name varchar(50) NOT NULL COMMENT 'Game name',
    supported_lang varchar(20) NOT NULL COMMENT 'What languages the
game supports.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE game_supported_languages
    ADD PRIMARY KEY (game_name,supported_lang);
ALTER TABLE game_supported_languages
    ADD CONSTRAINT fk_suplang_gamenname FOREIGN KEY (game_name)
REFERENCES game (game_name);
```

4.20.2 Game_Supported_Languages Table Example Data

```

INSERT INTO game_supported_languages (game_name, supported_lang)
VALUES
('Asphalt Xtreme', 'English'), ('Asphalt Xtreme', 'French'), ('Asphalt Xtreme', 'Italian'),
('Exploding Kittens', 'English'), ('Exploding Kittens', 'French'), ('Exploding Kittens', 'Spanish'),
('Heads Up!', 'English'), ('Heads Up!', 'German'), ('Heads Up!', 'Spanish'),
('Hextech Mayhem', 'English'), ('Hextech Mayhem', 'German'), ('Hextech Mayhem', 'Portuguese'),
('Into the Breach', 'English'), ('Into the Breach', 'French'), ('Into the Breach', 'German'),
('Into the Breach', 'Spanish'), ('Kentucky Route Zero', 'English'), ('Kentucky Route Zero', 'French'),
('Kentucky Route Zero', 'Spanish'), ('Moonlighter', 'English'), ('Moonlighter', 'French'),
('Moonlighter', 'German'), ('Oxenfree', 'English'), ('Oxenfree', 'French'),
('Oxenfree', 'German'), ('Shatter Remastered', 'English'), ('Shatter Remastered', 'Spanish'),
('Stranger Things: 1984', 'English'), ('Stranger Things: 1984', 'Spanish');

```

4.21 Genre Table

The genre table represents the different genres a show on netflix may belong to. It includes a unique genre ID, and a genre name.

4.21.1 Genre DDL Script

```

CREATE TABLE genre (
    genre_id int(11) NOT NULL COMMENT 'Genre ID',
    genre_name varchar(30) NOT NULL COMMENT 'Genre Name'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE genre
ADD PRIMARY KEY (genre_id);

```

4.21.2 Genre table example data

```

INSERT INTO genre (genre_id, genre_name) VALUES
(1, 'Action'), (2, 'Adventure'), (3, 'Comedy'), (4, 'Crime'),
(5, 'Drama'), (6, 'Fantasy'), (7, 'Historical'), (8, 'Horror'),

```

```
(9, 'Mystery'),(10, 'Romance'),(11, 'Science Fiction'),  
(12, 'Thriller'),(13, 'Western');
```

4.22 Genre_CContain_Content Table

The genreContainContent table represents the M:N “contain” relationship between the genres and shows on Netflix.

4.22.1 Genre_CContain_Content DDL Script

```
CREATE TABLE genreContainContent (  
    genre_id int(11) NOT NULL COMMENT 'Genre ID',  
    content_id int(11) NOT NULL COMMENT 'Content ID'  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
ALTER TABLE genreContainContent  
    ADD PRIMARY KEY (genre_id,content_id),  
    ADD KEY fkContainContentId (content_id);  
  
ALTER TABLE genreContainContent  
    ADD CONSTRAINT fkContainContentId FOREIGN KEY (content_id)  
REFERENCES content (content_id),  
    ADD CONSTRAINT fkContainGenreId FOREIGN KEY (genre_id)  
REFERENCES genre (genre_id);
```

4.22.2 Genre_CContain_Content Example Data

```
INSERT INTO genreContainContent (genre_id, content_id) VALUES  
(4, 1),(12, 1),(5, 2),(13, 2),(1, 3),(2, 3),(3, 3),(3, 4),(5, 4),(5, 5),(10, 5),(5, 6),(12,  
6),(3, 7),(5, 7),(1, 8),(2, 8),(4, 9),  
(5, 9),(2, 10),(9, 10),(5, 11),(6, 11),(5, 12),(6, 12),(2, 13),(5, 13),(8, 13),(4, 14),(5,  
14),(12, 14),(4, 15),(5, 15),  
(5, 16),(10, 16),(5, 17),(9, 17),(12, 17),(5, 18),(7, 18),(1, 19),(2, 19),(5, 19),(4,  
20),(5, 20),(12, 20);
```

4.23 merchandise_item table

The merchandiseItem table represents all items offered on the Netflix Shop. It can be specialized into three different types: ‘apparel’, ‘drinkware’, and ‘collectible’.

4.23.1 merchandise_item table DDL Script

```

CREATE TABLE merchandise_item (
    item_id int(11) NOT NULL COMMENT 'Item ID',
    collection_category varchar(50) NOT NULL COMMENT 'The collection category
this item belongs to.',
    price decimal(10,2) NOT NULL COMMENT 'Price',
    description varchar(300) NOT NULL COMMENT 'Description',
    item_name varchar(50) NOT NULL COMMENT 'Item Name',
    discounted_price decimal(10,2) DEFAULT NULL COMMENT 'Discounted
price.',
    stock_quantity int(11) NOT NULL COMMENT 'Number of items remaining in
stock.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

ALTER TABLE merchandise_item
ADD PRIMARY KEY (item_id);

```

4.23.2 merchandise_item Example Data

```

INSERT INTO merchandise_item (item_id, collection_category, price,
description, item_name, discounted_price, stock_quantity) VALUES
(1, 'Stranger Things Collection', '29.99', 'Stranger Things Logo T-Shirt', 'Stranger
Things T-Shirt', NULL, 100),
(2, 'The Witcher Collection', '39.99', 'The Witcher Pullover Hoodie', 'The Witcher
Hoodie', NULL, 50),
(3, 'Bridgerton Collection', '25.99', 'Bridgerton Lace Tee', 'Bridgerton T-Shirt',
NULL, 75),
(4, 'Squid Game Collection', '49.99', 'Squid Game Player Tracksuit', 'Squid Game
Tracksuit', '44.99', 30),
(5, 'Stranger Things Collection', '15.99', 'Stranger Things themed mug with
high-quality ceramic material.', 'Stranger Things Mug', '12.99', 50),
(6, 'Bridgerton Collection', '19.99', 'Bridgerton crystal wine glass set, ideal for any
occasion.', 'Bridgerton Wine Glass', NULL, 30),
(7, 'The Witcher Collection', '12.99', 'The Witcher inspired stainless steel tumbler
for hot or cold drinks.', 'The Witcher Tumbler', NULL, 40),
(8, 'Squid Game Collection', '25.99', 'Squid Game glass bottle with iconic symbols
from the show.', 'Squid Game Glass Bottle', '22.99', 20),
(9, 'Stranger Things Collection', '14.99', 'Limited edition Stranger Things Funko
Pop featuring Eleven.', 'Stranger Things Funko Pop', NULL, 100),
(10, 'The Witcher Collection', '9.99', 'Detailed keychain inspired by The Witcher
series, featuring Geralt's medallion.', 'The Witcher Keychain', NULL, 200),

```

(11, 'Squid Game Collection', '19.99', 'Exclusive Squid Game figure with collectible display stand.', 'Squid Game Figure', NULL, 150),

(12, 'Bridgerton Collection', '24.99', 'Elegant Bridgerton mini replica of the Queen\'s crown for display.', 'Bridgerton Crown Replica', NULL, 75);

4.24 merchandise_order Table

The merchandise order table represents an order made by a user on the Netflix Shop.

4.24.1 merchandise_order DDL Script

```
CREATE TABLE merchandise_order (
    order_id int(11) NOT NULL COMMENT 'Order"s ID',
    username varchar(20) NOT NULL COMMENT 'User name',
    contact_email varchar(254) NOT NULL COMMENT 'Contact Email',
    card_number varchar(19) NOT NULL COMMENT 'Card Number on a credit
card.',
    expiration_date char(5) NOT NULL COMMENT 'Expiration Date on credit card.',
    security_code char(3) NOT NULL COMMENT 'Security code on a credit card.',
    name_on_card varchar(26) NOT NULL COMMENT 'Name on the credit card.',
    fname varchar(50) NOT NULL COMMENT 'First Name',
    lname varchar(50) NOT NULL COMMENT 'Last Name',
    postal_code varchar(10) NOT NULL COMMENT 'Postal Code',
    province varchar(20) NOT NULL COMMENT 'Province',
    city varchar(20) NOT NULL COMMENT 'City',
    country varchar(20) NOT NULL COMMENT 'Country',
    addr varchar(200) DEFAULT NULL COMMENT 'Extra address information',
    order_date date NOT NULL COMMENT 'Creation date of the order.',
    status varchar(10) NOT NULL COMMENT 'Order completion status. (Shipping,
Cancelled, Completed)'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE merchandise_order
    ADD PRIMARY KEY (order_id),
    ADD KEY fk_order_username (username);
ALTER TABLE merchandise_order
    ADD CONSTRAINT fk_order_username FOREIGN KEY (username)
    REFERENCES user_account (username);
```

4.24.2 merchandise_order example data

```

INSERT INTO merchandise_order (order_id, username, contact_email,
card_number, expiration_date, security_code, name_on_card, fname, lname,
postal_code, province, city, country, addr, order_date, status) VALUES
(1, 'alice_b', 'alice.b@example.com', '4532 1234 5678 9012', '02/34', '321', 'Alice
Brown', 'Alice', 'Brown', '12345', 'Ontario', 'Toronto', 'Canada', '901 Chestnut St.',
'2024-10-01', 'completed'),
(2, 'bob_the_builder', 'bob.builder@example.com', '4716 2345 6789 0123',
'01/35', '654', 'Bob Builder', 'Bob', 'Builder', '67890', 'Quebec', 'Montreal',
'Canada', '902 Cedar Ave.', '2024-10-03', 'shipped'),
(3, 'charlie123', 'charlie123@example.com', '4916 3456 7890 1234', '12/36', '987',
'Charlie Lee', 'Charlie', 'Lee', '13579', 'British Columbia', 'Vancouver', 'Canada',
'903 Maple Rd.', '2024-10-05', 'cancelled'),
(4, 'david_k', 'david.k@example.com', '4532 4567 8901 2345', '11/37', '210',
'David King', 'David', 'King', '24680', 'Alberta', 'Calgary', 'Canada', '904 Pine
Cres.', '2024-10-07', 'completed'),
(5, 'emily.r', 'emily.r@example.com', '4716 5678 9012 3456', '10/38', '543', 'Emily
Rose', 'Emily', 'Rose', '11223', 'Manitoba', 'Winnipeg', 'Canada', '905 Spruce Ln.',
'2024-10-09', 'shipped'),
(6, 'frank88', 'frank88@example.com', '4916 6789 0123 4567', '09/39', '876',
'Frank Gold', 'Frank', 'Gold', '44556', 'Nova Scotia', 'Halifax', 'Canada', '906 Birch
Blvd.', '2024-10-11', 'completed'),
(7, 'gina_l', 'gina.l@example.com', '4532 7890 1234 5678', '08/40', '109', 'Gina
Lake', 'Gina', 'Lake', '88990', 'Ontario', 'Ottawa', 'Canada', '907 Oak Pl.',
'2024-10-13', 'cancelled'),
(8, 'harry_p', 'harry.p@example.com', '4716 8901 2345 6789', '07/41', '432',
'Harry Potter', 'Harry', 'Potter', '22334', 'Newfoundland', 'St. John's', 'Canada',
'908 Aspen Ct.', '2024-10-15', 'shipped'),
(9, 'jane_smith', 'jane.smith@example.com', '4916 9012 3456 7890', '06/42',
'765', 'Jane Smith', 'Jane', 'Smith', '55678', 'New Brunswick', 'Moncton', 'Canada',
'909 Redwood Dr.', '2024-10-17', 'completed'),
(10, 'john_doe', 'john.doe@example.com', '4532 0123 4567 8901', '05/43', '098',
'John Doe', 'John', 'Doe', '78901', 'Yukon', 'Whitehorse', 'Canada', '910 Alder
Way', '2024-10-19', 'completed'),
(11, 'alice_b', 'alice.b@example.com', '4532 1234 5678 9012', '02/34', '321', 'Alice
Brown', 'Alice', 'Brown', '12345', 'Ontario', 'Toronto', 'Canada', '901 Chestnut St.',
'2024-11-01', 'shipped'),
(12, 'bob_the_builder', 'bob.builder@example.com', '4716 2345 6789 0123',
'01/35', '654', 'Bob Builder', 'Bob', 'Builder', '67890', 'Quebec', 'Montreal',
'Canada', '902 Cedar Ave.', '2024-11-03', 'completed'),

```

(13, 'charlie123', 'charlie123@example.com', '4916 3456 7890 1234', '12/36', '987', 'Charlie Lee', 'Charlie', 'Lee', '13579', 'British Columbia', 'Vancouver', 'Canada', '903 Maple Rd.', '2024-11-05', 'shipped'),
(14, 'david_k', 'david.k@example.com', '4532 4567 8901 2345', '11/37', '210', 'David King', 'David', 'King', '24680', 'Alberta', 'Calgary', 'Canada', '904 Pine Cres.', '2024-11-07', 'cancelled'),
(15, 'emily.r', 'emily.r@example.com', '4716 5678 9012 3456', '10/38', '543', 'Emily Rose', 'Emily', 'Rose', '11223', 'Manitoba', 'Winnipeg', 'Canada', '905 Spruce Ln.', '2024-11-09', 'completed'),
(16, 'frank88', 'frank88@example.com', '4916 6789 0123 4567', '09/39', '876', 'Frank Gold', 'Frank', 'Gold', '44556', 'Nova Scotia', 'Halifax', 'Canada', '906 Birch Blvd.', '2024-11-11', 'shipped'),
(17, 'gina_l', 'gina.l@example.com', '4532 7890 1234 5678', '08/40', '109', 'Gina Lake', 'Gina', 'Lake', '88990', 'Ontario', 'Ottawa', 'Canada', '907 Oak Pl.', '2024-11-13', 'completed'),
(18, 'harry_p', 'harry.p@example.com', '4716 8901 2345 6789', '07/41', '432', 'Harry Potter', 'Harry', 'Potter', '22334', 'Newfoundland', 'St. John\l's', 'Canada', '908 Aspen Ct.', '2024-11-15', 'completed'),
(19, 'jane_smith', 'jane.smith@example.com', '4916 9012 3456 7890', '06/42', '765', 'Jane Smith', 'Jane', 'Smith', '55678', 'New Brunswick', 'Moncton', 'Canada', '909 Redwood Dr.', '2024-11-17', 'cancelled'),
(20, 'john_doe', 'john.doe@example.com', '4532 0123 4567 8901', '05/43', '098', 'John Doe', 'John', 'Doe', '78901', 'Yukon', 'Whitehorse', 'Canada', '910 Alder Way', '2024-11-19', 'shipped'),
(21, 'alice_b', 'alice.b@example.com', '4532 1234 5678 9012', '02/34', '321', 'Alice Brown', 'Alice', 'Brown', '12345', 'Ontario', 'Toronto', 'Canada', '901 Chestnut St.', '2024-10-03', 'shipped'),
(22, 'bob_the_builder', 'bob.builder@example.com', '4716 2345 6789 0123', '01/35', '654', 'Bob Builder', 'Bob', 'Builder', '67890', 'Quebec', 'Montreal', 'Canada', '902 Cedar Ave.', '2024-10-06', 'completed'),
(23, 'charlie123', 'charlie123@example.com', '4916 3456 7890 1234', '12/36', '987', 'Charlie Lee', 'Charlie', 'Lee', '13579', 'British Columbia', 'Vancouver', 'Canada', '903 Maple Rd.', '2024-10-09', 'cancelled'),
(24, 'david_k', 'david.k@example.com', '4532 4567 8901 2345', '11/37', '210', 'David King', 'David', 'King', '24680', 'Alberta', 'Calgary', 'Canada', '904 Pine Cres.', '2024-10-12', 'completed'),
(25, 'emily.r', 'emily.r@example.com', '4716 5678 9012 3456', '10/38', '543', 'Emily Rose', 'Emily', 'Rose', '11223', 'Manitoba', 'Winnipeg', 'Canada', '905 Spruce Ln.', '2024-10-15', 'shipped'),

```

(26, 'frank88', 'frank88@example.com', '4916 6789 0123 4567', '09/39', '876',
'Frank Gold', 'Frank', 'Gold', '44556', 'Nova Scotia', 'Halifax', 'Canada', '906 Birch
Blvd.', '2024-10-18', 'completed'),
(27, 'gina_l', 'gina.l@example.com', '4532 7890 1234 5678', '08/40', '109', 'Gina
Lake', 'Gina', 'Lake', '88990', 'Ontario', 'Ottawa', 'Canada', '907 Oak Pl.',
'2024-10-21', 'cancelled'),
(28, 'harry_p', 'harry.p@example.com', '4716 8901 2345 6789', '07/41', '432',
'Harry Potter', 'Harry', 'Potter', '22334', 'Newfoundland', 'St. John\l's', 'Canada',
'908 Aspen Ct.', '2024-10-24', 'shipped'),
(29, 'jane_smith', 'jane.smith@example.com', '4916 9012 3456 7890', '06/42',
'765', 'Jane Smith', 'Jane', 'Smith', '55678', 'New Brunswick', 'Moncton', 'Canada',
'909 Redwood Dr.', '2024-10-27', 'completed'),
(30, 'john_doe', 'john.doe@example.com', '4532 0123 4567 8901', '05/43', '098',
'John Doe', 'John', 'Doe', '78901', 'Yukon', 'Whitehorse', 'Canada', '910 Alder
Way', '2024-10-30', 'completed');

```

4.25 merchandise_review Table

The merchandise_review table represents the reviews made by users, for items on Netflix Shop.

4.24.1 merchandise_review DDL Script

```

CREATE TABLE merchandise_review (
    review_id int(11) NOT NULL COMMENT 'Review ID',
    username varchar(20) NOT NULL COMMENT 'User name',
    item_id int(11) NOT NULL COMMENT 'Item ID',
    review_title varchar(50) NOT NULL COMMENT 'Review Title',
    comment varchar(200) NOT NULL COMMENT 'Textual content of the review.',
    rating int(1) NOT NULL COMMENT 'Rating given.',
    name varchar(20) NOT NULL COMMENT 'Name displayed on the review.',
    creation_date date NOT NULL COMMENT 'Creation date.',
    email varchar(254) NOT NULL COMMENT 'Email',
    verified_buyer_flag tinyint(1) NOT NULL COMMENT 'Mark if user is a verified
buyer of this item.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE merchandise_review
ADD PRIMARY KEY (review_id,username,item_id),
ADD KEY fk_review_username (username),
ADD KEY fk_review_itemid (item_id)
ALTER TABLE merchandise_review

```

```
ADD CONSTRAINT fk_review_itemid FOREIGN KEY (item_id) REFERENCES
merchandise_item (item_id),
```

```
ADD CONSTRAINT fk_review_username FOREIGN KEY (username)
REFERENCES user_account (username);
```

4.24.2 merchandise_review example data

```
INSERT INTO merchandise_review (review_id, username, item_id, review_title,
comment, rating, name, creation_date, email, verified_buyer_flag) VALUES
(1, 'john_doe', 1, 'Perfect Fit!', 'The Stranger Things T-Shirt fits great and feels
comfortable. The design is awesome!', 5, 'John D.', '2024-10-12',
'john.doe@example.com', 1),
(2, 'jane_smith', 2, 'Cozy and Warm', 'The Witcher Hoodie is perfect for chilly
days. Love the design!', 5, 'Jane S.', '2024-10-13', 'jane.smith@example.com', 1),
(3, 'alice_b', 3, 'Not as Expected', 'The Bridgerton T-Shirt didn\'t meet my
expectations. The lace feels cheap.', 2, 'Alice B.', '2024-10-14',
'alice.b@example.com', 0),
(4, 'bob_the_builder', 4, 'Great Value', 'Got the Squid Game Tracksuit on sale. It
fits well and looks exactly like in the show!', 4, 'Bob B.', '2024-10-15',
'bob.builder@example.com', 1),
(5, 'charlie123', 5, 'Nice Mug', 'Stranger Things Mug is nice, but I wish it were a
bit bigger.', 4, 'Charlie', '2024-10-16', 'charlie123@example.com', 1),
(6, 'david_k', 6, 'Elegant Glasses', 'The Bridgerton Wine Glasses are stunning.
Perfect for entertaining!', 5, 'David K.', '2024-10-17', 'david.k@example.com', 1),
(7, 'emily.r', 7, 'Works Well', 'The Witcher Tumbler is great for hot coffee. Keeps it
warm for hours.', 5, 'Emily R.', '2024-10-18', 'emily.r@example.com', 0),
(8, 'frank88', 8, 'Stylish Bottle', 'The Squid Game Glass Bottle looks amazing, but
it's a bit fragile.', 3, 'Frank88', '2024-10-19', 'frank88@example.com', 1),
(9, 'gina_l', 9, 'Love It!', 'The Stranger Things Funko Pop of Eleven is so cute! A
must-have for fans.', 5, 'Gina L.', '2024-10-20', 'gina.l@example.com', 1),
(10, 'harry_p', 10, 'Not Worth It', 'The Witcher Keychain is a bit overpriced for
what it is. Small but nice design.', 3, 'Harry P.', '2024-10-21',
'harry.p@example.com', 0),
(11, 'john_doe', 11, 'Amazing Detail', 'The Squid Game Figure looks great!
Detailed and worth collecting.', 5, 'John D.', '2024-10-22',
'john.doe@example.com', 1),
(12, 'jane_smith', 12, 'Beautiful Display Piece', 'The Bridgerton Crown Replica is
exquisite. Looks amazing on my shelf.', 5, 'Jane S.', '2024-10-23',
>jane.smith@example.com', 1),
(13, 'alice_b', 1, 'Nice Tee', 'The Stranger Things T-Shirt is cool, but the fabric
could be softer.', 4, 'Alice B.', '2024-10-24', 'alice.b@example.com', 1),
```

(14, 'bob_the_builder', 4, 'Very Comfortable', 'The Squid Game Tracksuit is super comfortable. The fit is true to size.', 5, 'Bob B.', '2024-10-25', 'bob.builder@example.com', 1),
(15, 'charlie123', 5, 'Good Quality', 'Stranger Things Mug is well made, perfect for a hot drink.', 5, 'Charlie', '2024-10-26', 'charlie123@example.com', 1),
(16, 'david_K', 8, 'Unique Design', 'Love the Squid Game Glass Bottle. The design is unique and stands out.', 4, 'David K.', '2024-10-27', 'david.k@example.com', 1),
(17, 'emily.r', 9, 'Collector's Item', 'Stranger Things Funko Pop is amazing for collectors. Highly recommended.', 5, 'Emily R.', '2024-10-28', 'emily.r@example.com', 1),
(18, 'frank88', 7, 'Functional and Cool', 'The Witcher Tumbler works well and looks great. Perfect for fans.', 5, 'Frank88', '2024-10-29', 'frank88@example.com', 0),
(19, 'gina_I', 2, 'Warm and Cozy', 'The Witcher Hoodie is very comfortable and warm. I wear it all the time.', 5, 'Gina L.', '2024-10-30', 'gina.l@example.com', 1),
(20, 'harry_p', 10, 'Disappointing', 'The Witcher Keychain is smaller than expected and feels cheap.', 2, 'Harry P.', '2024-10-31', 'harry.p@example.com', 0),
(21, 'john_doe', 3, 'Just OK', 'The Bridgerton T-Shirt is just alright. Expected a bit more quality.', 3, 'John D.', '2024-11-01', 'john.doe@example.com', 1),
(22, 'jane_smith', 12, 'Stunning', 'The Bridgerton Crown Replica is beautiful. Looks very regal on my shelf.', 5, 'Jane S.', '2024-11-02', 'jane.smith@example.com', 1),
(23, 'alice_b', 6, 'Elegant Set', 'The Bridgerton Wine Glasses look fantastic. Very elegant design.', 5, 'Alice B.', '2024-11-03', 'alice.b@example.com', 0),
(24, 'bob_the_builder', 8, 'Great Design', 'Squid Game Glass Bottle is stylish but a bit fragile. Handle with care.', 4, 'Bob B.', '2024-11-04', 'bob.builder@example.com', 1),
(25, 'charlie123', 9, 'Awesome Figure', 'The Stranger Things Funko Pop is a great addition to my collection!', 5, 'Charlie', '2024-11-05', 'charlie123@example.com', 1),
(26, 'david_K', 11, 'Worth It!', 'The Squid Game Figure is amazing. Great quality and detail.', 5, 'David K.', '2024-11-06', 'david.k@example.com', 1),
(27, 'emily.r', 4, 'Soft Fabric', 'The Squid Game Tracksuit is super soft and comfy. I love it!', 5, 'Emily R.', '2024-11-07', 'emily.r@example.com', 1),
(28, 'frank88', 1, 'Not Bad', 'Stranger Things T-Shirt is nice but runs a bit small. Order a size up!', 3, 'Frank88', '2024-11-08', 'frank88@example.com', 0),
(29, 'gina_I', 7, 'Sturdy Tumbler', 'The Witcher Tumbler is very sturdy and keeps drinks hot for hours.', 5, 'Gina L.', '2024-11-09', 'gina.l@example.com', 1),

```
(30, 'harry_p', 5, 'Cool Mug!', 'The Stranger Things Mug looks great and is very durable.', 4, 'Harry P.', '2024-11-10', 'harry.p@example.com', 1);
```

4.26 movie Table

The movie table represents the movie type of show on Netflix. It is a specialization of the content table.

4.25.1 movie DDL Script

```
CREATE TABLE movie (
    content_id int(11) NOT NULL COMMENT 'Content ID',
    movieduration time NOT NULL COMMENT 'Movie duration'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE movie
    ADD PRIMARY KEY (content_id);
ALTER TABLE movie
    ADD CONSTRAINT fk_movie_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id);
```

4.25.2 movie example data

```
INSERT INTO movie (content_id, movieduration) VALUES
(11, '02:01:00'),(12, '02:19:00'),(13, '01:58:00'),(14, '02:18:00'),(15,
'02:06:00'),(16, '02:04:00'),(17, '02:16:00'),
(18, '02:05:00'),(19, '03:29:00'),(20, '02:03:00');
```

4.27 order_contains_item Table

The order_contains_item table represents the M:N “contain” relationship between orders and merchandise items on Netflix Shop.

4.27.1 order_contains_item DDL Script

```
CREATE TABLE order_contains_item (
    order_id int(11) NOT NULL COMMENT 'The ID of the order',
    item_id int(11) NOT NULL COMMENT 'The ID of the merchandise item',
    quantity int(11) NOT NULL COMMENT 'The quantity of the item in the order'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE order_contains_item
    ADD PRIMARY KEY (order_id,item_id),
    ADD KEY item_id (item_id);
ALTER TABLE order_contains_item
```

```

ADD CONSTRAINT order_contains_item_ibfk_2 FOREIGN KEY (item_id)
REFERENCES merchandise_item (item_id),
ADD CONSTRAINT order_contains_item_ibfk_3 FOREIGN KEY (order_id)
REFERENCES merchandise_order (order_id);

```

4.27.2 order_contains_item example data

```

INSERT INTO order_contains_item (order_id, item_id, quantity) VALUES
(1, 1, 2),(1, 3, 1),(2, 2, 3),(2, 4, 1),(3, 5, 2),(3, 6, 1),(4, 7, 3),(5, 8, 2),(5, 9, 1),(6,
10, 4),(7, 11, 3),(7, 12, 1),(8, 1, 2),
(9, 2, 1),(9, 3, 2),(10, 4, 3),(10, 5, 1),(11, 6, 4),(11, 7, 1),(12, 8, 2),(13, 9, 3),(13,
10, 1),(14, 11, 2),(15, 12, 4),(16, 1, 3),
(16, 5, 1),(17, 2, 4),(17, 6, 2),(18, 7, 1),(18, 8, 3),(19, 9, 2),(19, 10, 4),(20, 11,
3),(21, 12, 2),(22, 1, 1),(22, 3, 3),
(23, 4, 2),(23, 5, 4),(24, 6, 3),(25, 7, 2),(25, 8, 1),(26, 9, 4),(27, 10, 2),(28, 11,
1),(29, 12, 3),(30, 1, 4);

```

4.28 order_use_coupon Table

The order_use_coupon table represents the M:N “use” relationship between coupons and orders.

4.28.1 order_use_coupon DDL Script

```

CREATE TABLE order_use_coupon (
    order_id int(11) NOT NULL COMMENT 'The ID of the order',
    promo_code varchar(10) NOT NULL COMMENT 'The coupon code used in the
order'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE order_use_coupon
    ADD PRIMARY KEY (order_id,promo_code),
    ADD UNIQUE KEY unique_order_coupon (order_id,promo_code),
    ADD KEY promo_code (promo_code);
ALTER TABLE order_use_coupon
    ADD CONSTRAINT order_use_coupon_ibfk_1 FOREIGN KEY (order_id)
REFERENCES merchandise_order (order_id),
    ADD CONSTRAINT order_use_coupon_ibfk_2 FOREIGN KEY (promo_code)
REFERENCES coupon (promo_code);

```

4.28.2 order_use_coupon example data

```

INSERT INTO order_use_coupon (order_id, promo_code) VALUES
(6, 'BLACKFRI'),(15, 'BLACKFRI'),(1, 'BRIDGE15'),(9, 'BRIDGE15'),(11,
'BRIDGE15'),(5, 'FALLDEAL'),

```

```

(12, 'FALLDEAL'),(14, 'NEWYEAR5'),(22, 'NEWYEAR5'),(23, 'NEWYEAR5'),(30,
'SPRING25'),(2, 'SQUID20'),
\,(12, 'SQUID20'),(1, 'STRANGE10'),(13, 'STRANGE10'),(16, 'STRANGE10'),(4,
'SUMMER10'),(9, 'SUMMER10'),
(17, 'SUMMER10'),(19, 'SUMMER10'),(25, 'SUMMER10'),(26, 'SUMMER10'),(2,
'WITCHER5'),(6, 'WITCHER5'),
(9, 'WITCHER5'),(11, 'WITCHER5'),(13, 'WITCHER5'),(2, 'XMAS2024'),(10,
'XMAS2024'),(20, 'XMAS2024');

```

4.29 payment_method Table

The payment_method table represents a payment method a user account can create, in order to subscribe to one of Netflix plans.

4.29.1 payment_method DDL Script

```

CREATE TABLE payment_method (
    card_number varchar(19) NOT NULL COMMENT 'Credit card number',
    username varchar(20) NOT NULL COMMENT 'The username of the user owning this
payment payment method.',
    expiration_date char(5) NOT NULL COMMENT 'Expiration date of credit card',
    security_code char(3) NOT NULL COMMENT 'Security code of credit card.',
    name_on_card varchar(26) NOT NULL COMMENT 'Name on credit card.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE payment_method
    ADD PRIMARY KEY (card_number),
    ADD KEY fk_paymentmethod_username (username);
ALTER TABLE payment_method
    ADD CONSTRAINT fk_paymentmethod_username FOREIGN KEY (username)
REFERENCES user_account (username);

```

4.29.2 payment_method example data

```

INSERT INTO payment_method (card_number, username, expiration_date,
security_code, name_on_card) VALUES
('3714 4963 0399 505', 'emily.r', '09/23', '345', 'Emily Roberts'),
('3787 3449 3676 001', 'emily.r', '08/25', '234', 'Emily Roberts'),
('4111 1111 1111 1111', 'alice_b', '12/25', '123', 'Alice Brown'),
('4111 1111 1111 1117', 'jane_smith', '01/25', '678', 'Jane Smith'),
('4111 1111 1111 1120', 'david_k', '09/26', '012', 'David Knight'),
('4111 1111 1111 1130', 'charlie123', '03/27', '678', 'Charlie Brown'),
('4500 0000 0000 0011', 'bob_the_builder', '11/26', '345', 'Bob Builder'),
('5105 1051 0510 5100', 'harry_p', '03/25', '345', 'Harry Park'),
('5222 2222 2222 2224', 'david_k', '12/24', '012', 'David Knight'),

```

```
('5500 0000 0000 0004', 'bob_the_builder', '06/24', '456', 'Bob Builder'),
('5500 0000 0000 0005', 'frank88', '04/24', '567', 'Frank Green'),
('6011 0000 0000 0009', 'charlie123', '10/23', '789', 'Charlie Brown'),
('6011 0000 0000 0010', 'gina_I', '02/27', '890', 'Gina Lee'),
('6011 0000 0000 0022', 'alice_b', '05/23', '234', 'Alice Brown'),
('6200 0000 0000 0003', 'john_doe', '07/24', '901', 'John Doe');
```

4.30 plan Table

The plan table represents the different subscription plans offered by Netflix.

4.30.1 plan DDL Script

```
CREATE TABLE plan (
    plan_name char(10) NOT NULL COMMENT 'Plan name',
    monthly_price decimal(4,2) NOT NULL COMMENT 'Monthly price',
    vid_sound_quality varchar(15) NOT NULL COMMENT 'Video and sound
quality',
    resolution varchar(10) NOT NULL COMMENT 'Content resolution',
    ads_enabled_flag tinyint(1) NOT NULL COMMENT 'Mark if plan is ads
enabled.',
    simul_watch_count int(1) NOT NULL COMMENT 'Count how many devices are
allowed to watch on this account.',
    download_device_count int(1) NOT NULL COMMENT 'Count how many
devices are allowed to download content using this account.',
    extra_member_slots int(1) NOT NULL COMMENT 'Count how many other user
accounts you can invite to become extra members.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE plan
ADD PRIMARY KEY (plan_name);
```

4.30.2 plan example data

```
INSERT INTO plan (plan_name, monthly_price, vid_sound_quality, resolution,
ads_enabled_flag, simul_watch_count, download_device_count,
extra_member_slots) VALUES
('BASIC', '9.99', 'Stereo', 'SD', 0, 1, 1, 0),
('BASICADS', '6.99', 'Stereo', 'SD', 1, 1, 1, 0),
('MOBILE', '3.99', 'Stereo', 'SD', 0, 1, 1, 0),
('PREMIUM', '19.99', 'Dolby Atmos', 'UHD/4K', 0, 4, 6, 2),
('STANDARD', '15.99', 'Dolby Audio', 'HD', 0, 2, 2, 1);
```

4.31 Profile Table

The profile table represents a profile that belongs to a user account. Each user account has a minimum of 1 profile, and can have up to 5 profiles.

4.31.1 Profile Table DDL Script

```
CREATE TABLE profile (
    profile_name varchar(20) NOT NULL COMMENT 'Profile name',
    username varchar(20) NOT NULL COMMENT 'User name',
    maturity_rating varchar(3) NOT NULL COMMENT 'Maturity rating',
    kids_profile_status tinyint(1) NOT NULL COMMENT 'Mark if the profile is a
    "kids profile".',
    display_language varchar(3) NOT NULL COMMENT 'UI display language',
    audio_sub_language varchar(10) NOT NULL COMMENT 'Audio and subtitles
    language',
    eps_autoplay_flag tinyint(1) NOT NULL COMMENT 'Mark if next episode in a
    series should play automatically after the last one ends.',
    preview_autoplay_flag tinyint(1) NOT NULL COMMENT 'Mark if a preview for a
    content should play automatically when hovering.',
    pin int(6) DEFAULT NULL COMMENT 'PIN to secure profile access.',
    game_handle varchar(20) DEFAULT NULL COMMENT 'Game handle for user
    profile playing Netflix Games.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE profile
    ADD PRIMARY KEY (profile_name,username),
    ADD KEY fk_profile_username (username);
ALTER TABLE profile
    ADD CONSTRAINT fk_profile_username FOREIGN KEY (username)
    REFERENCES user_account (username);
```

4.31.2 Profile table example data

```
INSERT INTO profile (profile_name, username, maturity_rating,
kids_profile_status, display_language, audio_sub_language, eps_autoplay_flag,
preview_autoplay_flag, pin, game_handle) VALUES
('AdventureZone', 'bob_the_builder', '18+', 0, 'ENG', 'ENG', 1, 1, 654321,
'bob149'),
('BeastMode', 'frank88', '18+', 0, 'ENG', 'ENG', 1, 1, 223344, 'frank257'),
('CaptainElite', 'harry_p', '18+', 0, 'ENG', 'ENG', 0, 1, 778899, 'harry259'),
('CosmicQueen', 'charlie123', '18+', 0, 'ENG', 'FRE', 0, 1, 112233, 'charlie223'),
```

```

('LilBuddy', 'bob_the_builder', 'G', 1, 'ENG', 'GER', 1, 0, NULL, NULL),
('LilDreamer', 'emily.r', 'G', 1, 'ENG', 'POR', 1, 0, NULL, NULL),
('Maverick', 'emily.r', '18+', 0, 'ENG', 'ENG', 0, 1, 998877, 'emily321'),
('MiniExplorer', 'charlie123', 'G', 1, 'ENG', 'ENG', 1, 0, NULL, NULL),
('RogueMaster', 'gina_l', '18+', 0, 'ENG', 'ENG', 1, 1, 445566, 'gina512'),
('SpaceCadet', 'david_k', 'G', 1, 'ENG', 'ITA', 1, 0, NULL, NULL),
('SuperHeroX', 'alice_b', '18+', 0, 'ENG', 'ENG', 1, 1, 123456, 'alice257'),
('TechGuru', 'david_k', '18+', 0, 'ENG', 'ENG', 1, 1, 334455, 'david789'),
('TinyAdventurer', 'frank88', 'G', 1, 'ENG', 'SPA', 1, 0, NULL, NULL),
('TinyTots', 'alice_b', 'G', 1, 'ENG', 'SPA', 1, 0, NULL, NULL),
('WonderKid', 'gina_l', 'G', 1, 'ENG', 'ENG', 1, 0, NULL, NULL);

```

4.32 Subscription Table

The subscription table represents the ternary relationship “subscribe” between user_account, “plan”, and “payment_method”.

4.32.1 Subscription DDL Script

```

CREATE TABLE subscription (
    subscription_id int(11) NOT NULL COMMENT 'Subscription ID.',
    plan_name char(10) NOT NULL COMMENT 'Plan name',
    username varchar(20) NOT NULL COMMENT 'User name',
    card_number varchar(19) NOT NULL COMMENT 'Card number',
    subscription_date date NOT NULL COMMENT 'Subscription date',
    termination_date date DEFAULT NULL COMMENT 'Termination date. (Null if
subscription is still active)'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE subscription
    ADD PRIMARY KEY (subscription_id),
    ADD KEY fk_subscription_cardnumber (card_number),
    ADD KEY fk_subscription_username (username),
    ADD KEY fk_subscription_planname (plan_name);
ALTER TABLE subscription
    ADD CONSTRAINT fk_subscription_cardnumber FOREIGN KEY
(card_number) REFERENCES payment_method (card_number),
    ADD CONSTRAINT fk_subscription_planname FOREIGN KEY (plan_name)
REFERENCES plan (plan_name),
    ADD CONSTRAINT fk_subscription_username FOREIGN KEY (username)
REFERENCES user_account (username);

```

4.32.2 Subscription table example data

```
INSERT INTO subscription (subscription_id, plan_name, username, card_number, subscription_date, termination_date) VALUES  
(1, 'BASIC', 'alice_b', '4111 1111 1111 1111', '2024-01-01', '2024-06-01'),(2, 'BASICADS', 'alice_b', '6011 0000 0000 0022', '2024-06-02', NULL),(3, 'MOBILE', 'bob_the_builder', '4500 0000 0000 0011', '2023-03-01', '2024-05-01'),(4, 'PREMIUM', 'bob_the_builder', '5500 0000 0000 0004', '2024-05-02', NULL),(5, 'BASIC', 'charlie123', '4111 1111 1111 1130', '2024-02-01', '2024-04-01'),(6, 'BASICADS', 'charlie123', '6011 0000 0000 0009', '2024-04-02', NULL),(7, 'MOBILE', 'david_k', '4111 1111 1111 1120', '2023-11-01', '2024-02-01'),(8, 'PREMIUM', 'david_k', '5222 2222 2222 2224', '2024-02-02', NULL),(9, 'BASIC', 'emily.r', '3714 4963 0399 505', '2023-10-01', '2024-03-01'),(10, 'STANDARD', 'emily.r', '3787 3449 3676 001', '2024-03-02', NULL),(11, 'BASIC', 'frank88', '5500 0000 0000 0005', '2023-09-01', '2024-02-01'),(12, 'MOBILE', 'frank88', '6011 0000 0000 0010', '2024-02-02', NULL),(13, 'STANDARD', 'gina_l', '6011 0000 0000 0010', '2023-12-01', '2024-05-01'),(14, 'PREMIUM', 'gina_l', '5500 0000 0000 0005', '2024-05-02', NULL),(15, 'BASIC', 'harry_p', '5105 1051 0510 5100', '2023-08-01', '2024-02-01'),(16, 'STANDARD', 'harry_p', '4111 1111 1111 1117', '2024-02-02', NULL),(17, 'PREMIUM', 'jane_smith', '6200 0000 0000 0003', '2023-07-01', '2024-01-01'),(18, 'BASICADS', 'jane_smith', '4111 1111 1111 1117', '2024-01-02', NULL),(19, 'STANDARD', 'john_doe', '6011 0000 0000 0022', '2024-02-01', '2024-04-01'),(20, 'PREMIUM', 'john_doe', '4111 1111 1111 1111', '2024-04-02', NULL),(21, 'BASIC', 'alice_b', '6011 0000 0000 0022', '2024-06-01', '2024-08-01'),(22, 'PREMIUM', 'alice_b', '4500 0000 0000 0011', '2024-08-02', NULL),(23, 'MOBILE', 'bob_the_builder', '5500 0000 0000 0004', '2024-07-01', '2024-09-01'),(24, 'BASICADS', 'bob_the_builder', '4500 0000 0000 0011', '2024-09-02', NULL),(25, 'STANDARD', 'charlie123', '6011 0000 0000 0009', '2024-05-01', '2024-07-01'),(26, 'PREMIUM', 'charlie123', '5222 2222 2222 2224', '2024-07-02', NULL),(27, 'BASIC', 'david_k', '4111 1111 1111 1120', '2024-04-01', '2024-06-01'),(28, 'PREMIUM', 'david_k', '5222 2222 2222 2224', '2024-06-02', NULL),(29, 'STANDARD', 'emily.r', '3714 4963 0399 505', '2024-01-01', '2024-03-01'),(30, 'BASICADS', 'emily.r', '3787 3449 3676 001', '2024-03-02', NULL);
```

4.33 tv_episode Table

The tv_episode table represents an episode contained within a TV show on Netflix.

4.33.1 tv_episode DDL Script

```
CREATE TABLE tv_episode (
    content_id int(11) NOT NULL COMMENT 'Content ID of the episode.',
    season_no int(11) NOT NULL COMMENT 'The number of season the episode
belongs to.',
    ep_no int(11) NOT NULL COMMENT 'The number of the episode.',
    ep_name varchar(100) NOT NULL COMMENT 'An episode"s name',
    description varchar(300) NOT NULL COMMENT 'The episode"s description',
    episode_duration int(11) NOT NULL COMMENT 'An episode"s duration in
seconds.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE tv_episode
    ADD PRIMARY KEY (content_id,season_no,ep_no);
ALTER TABLE tv_episode
    ADD CONSTRAINT tv_episode_ibfk_1 FOREIGN KEY (content_id)
REFERENCES tv_show (content_id);
```

4.33.2 tv_episode table example data

```
INSERT INTO tv_episode (content_id, season_no, ep_no, ep_name, description,
episode_duration) VALUES
(1, 1, 1, 'Episode 1: The Awakening', 'The protagonist begins a life-changing
investigation into a paradox.', 2730),
(1, 1, 2, 'Episode 2: Paradox Revealed', 'The investigation takes a dramatic turn
as new information surfaces.', 2760),
(1, 1, 3, 'Episode 3: The Final Answer', 'All the mysteries come to a head in this
thrilling conclusion.', 2830),
(2, 1, 1, 'Episode 1: First Case', 'The lawyer takes on his first case that seems
impossible to win.', 2910),
(2, 1, 2, 'Episode 2: Legal Troubles', 'The lawyer faces difficulties as his client's
past comes back to haunt them.', 2940),
(2, 1, 3, 'Episode 3: The Courtroom Battle', 'The lawyer takes the case to court,
where things get intense.', 3000),
(3, 1, 1, 'Episode 1: The Beginning', 'Beth Harmon discovers her passion for
chess at an early age.', 3000),
(3, 1, 2, 'Episode 2: The Challenge', 'Beth faces her first big competition, a
challenging tournament.', 3120),
```

- (3, 1, 3, 'Episode 3: The Rival', 'Beth faces off against her toughest opponent yet in the chess world.', 3090),
- (4, 1, 1, 'Episode 1: The Pirate King\'s Will', 'Luffy begins his journey to become the next Pirate King.', 2700),
- (4, 1, 2, 'Episode 2: The Straw Hat Crew', 'Luffy gathers his first companions for the adventure ahead.', 2790),
- (4, 1, 3, 'Episode 3: The First Battle', 'The crew faces their first major battle with rival pirates.', 2820),
- (4, 2, 1, 'Episode 1: The Grand Line', 'The crew sets sail for the Grand Line in search of treasure.', 2820),
- (4, 2, 2, 'Episode 2: The Lost Island', 'Luffy and the crew discover an island filled with mysteries.', 2880),
- (4, 2, 3, 'Episode 3: The Conqueror', 'A confrontation with a pirate king shakes the crew to their core.', 2850),
- (5, 1, 1, 'Episode 1: The Vanishing', 'A young boy mysteriously disappears in Hawkins, Indiana.', 2740),
- (5, 1, 2, 'Episode 2: The Upside Down', 'The group of friends discover a strange world beyond our own.', 2880),
- (5, 1, 3, 'Episode 3: The Monster', 'The kids face off against a terrifying creature from the Upside Down.', 3030),
- (5, 2, 1, 'Episode 1: The Return', 'The gang reunites after some time apart, but danger lurks once again.', 2800),
- (5, 2, 2, 'Episode 2: The Terrors of Hawkins', 'More secrets of Hawkins are revealed as the group digs deeper into the mysteries.', 2940),
- (5, 2, 3, 'Episode 3: The Final Showdown', 'The group confronts their fears in a climactic battle.', 3000),
- (6, 1, 1, 'Episode 1: The New Queen', 'Elizabeth II ascends to the throne after her father's sudden death.', 3120),
- (6, 1, 2, 'Episode 2: The Burden of Rule', 'Queen Elizabeth struggles with the weight of responsibility.', 3180),
- (6, 1, 3, 'Episode 3: The First Crisis', 'A political crisis forces the new Queen to make difficult decisions.', 3240),
- (6, 2, 1, 'Episode 1: The Suez Crisis', 'The Queen faces her first international crisis involving the Suez Canal.', 3270),
- (6, 2, 2, 'Episode 2: The King\'s Dilemma', 'The Queen\'s husband, Prince Philip, faces his own struggles.', 3330),
- (6, 2, 3, 'Episode 3: The Family Dilemma', 'The Queen must navigate familial challenges as well as political ones.', 3380),
- (7, 1, 1, 'Episode 1: The Duke and the Duchess', 'The love story between Daphne and Simon begins.', 2760),

- (7, 1, 2, 'Episode 2: The Ball', 'A grand ball brings the high society of London together.', 2850),
- (7, 1, 3, 'Episode 3: The Secret', 'Daphne discovers a secret about Simon's past that threatens their relationship.', 2900),
- (7, 2, 1, 'Episode 1: The Proposal', 'A shocking proposal changes the course of Daphne and Simon's future.', 2890),
- (7, 2, 2, 'Episode 2: The Return', 'Daphne and Simon deal with the fallout from their decisions.', 2980),
- (7, 2, 3, 'Episode 3: The Resolution', 'The couple faces their final challenges, culminating in a heartwarming conclusion.', 3030),
- (8, 1, 1, 'Episode 1: The Plan', 'The professor unveils his elaborate plan to rob the Royal Mint of Spain.', 2700),
- (8, 1, 2, 'Episode 2: The Robbery', 'The heist begins, and everything quickly escalates.', 2850),
- (8, 1, 3, 'Episode 3: The First Obstacle', 'The robbers face their first major challenge during the heist.', 2800),
- (8, 2, 1, 'Episode 1: The Escape', 'The crew begins planning their escape after the robbery goes sideways.', 2930),
- (8, 2, 2, 'Episode 2: The Betrayal', 'One of the crew members betrays the group, putting everything at risk.', 2950),
- (8, 2, 3, 'Episode 3: The Tension', 'Tension builds as the heist reaches its breaking point.', 3000),
- (9, 1, 1, 'Episode 1: The Last Wish', 'Geralt of Rivia embarks on his first adventure involving a dangerous wish.', 2700),
- (9, 1, 2, 'Episode 2: The Hunt', 'Geralt hunts down a deadly creature that has been terrorizing a village.', 2780),
- (9, 1, 3, 'Episode 3: The Witcher's Code', 'Geralt faces a moral dilemma regarding the Witcher's code.', 2900),
- (9, 2, 1, 'Episode 1: The Warlock', 'Geralt encounters a powerful warlock who has secrets of his own.', 2820),
- (9, 2, 2, 'Episode 2: The Sorceress', 'Geralt forms an unlikely alliance with a sorceress to defeat a greater evil.', 2890),
- (9, 2, 3, 'Episode 3: The Betrayal', 'Geralt must make a choice that could alter his fate forever.', 2980),
- (10, 1, 1, 'Episode 1: The Game Begins', 'Participants enter the deadly game with their lives at stake.', 2670),
- (10, 1, 2, 'Episode 2: The First Challenge', 'The first challenge begins, and the participants realize the stakes.', 2750),
- (10, 1, 3, 'Episode 3: The Betrayal', 'A shocking betrayal occurs during the second round of the game.', 2850),

- (10, 2, 1, 'Episode 1: The Survivor', 'The few remaining players must face the final game of survival.', 2900),
- (10, 2, 2, 'Episode 2: The Winner\'s Dilemma', 'The winner must deal with the consequences of their actions.', 2980),
- (10, 2, 3, 'Episode 3: The Price of Victory', 'The final episode reveals the shocking price of victory in the Squid Game.', 3000);

4.34 tv_show Table

The tv_show table represents a tv_show type of show on Netflix. It is a specialization of the content table.

4.34.1 tv_show DDL Script

```
CREATE TABLE tv_show (
    content_id int(11) NOT NULL COMMENT 'Content ID',
    series_status varchar(20) NOT NULL COMMENT 'The current status of the
series (e.g., Ongoing, Completed, Canceled)',
    episode_release_frequency varchar(20) DEFAULT NULL COMMENT
'Frequency of episode releases (e.g., Weekly, All at Once)'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE tv_show
    ADD PRIMARY KEY (content_id);
ALTER TABLE tv_show
    ADD CONSTRAINT fk_tvshow_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id);
```

4.34.2 tv_show table example data

```
INSERT INTO tv_show (content_id, series_status, episode_release_frequency)
VALUES
(1, 'Ongoing', 'Weekly'),(2, 'Ongoing', 'All at Once'),(3, 'Completed', 'All at
Once'),(4, 'Ongoing', 'All at Once'),(5, 'Ongoing', 'All at Once'),(6, 'Ongoing', 'All at
Once'),(7, 'Ongoing', 'All at Once'),(8, 'Completed', 'All at Once'),(9, 'Ongoing', 'All
at Once'),(10, 'Ongoing', 'All at Once');
```

4.35 user_account Table

The user_account table represents the user accounts registered on netflix.

4.35.1 user_account DDL Script

```
CREATE TABLE user_account (
    username varchar(20) NOT NULL COMMENT 'User name',
    email varchar(254) NOT NULL COMMENT 'User Email',
    password int(64) NOT NULL COMMENT 'User password',
    date_of_birth date DEFAULT NULL COMMENT 'User date of birth',
    gender varchar(1) DEFAULT NULL COMMENT 'User gender'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE user_account
    ADD PRIMARY KEY (username);
```

4.35.2 user_account table example data

```
INSERT INTO user_account (username, email, password, date_of_birth, gender)
VALUES
('alice_b', 'alice.b@example.com', 34567890, NULL, NULL),('bob_the_builder',
'bob.builder@example.com', 45678901, '2000-05-12', 'M'),('charlie123',
'charlie123@example.com', 56789012, NULL, NULL),('david_k',
'david.k@example.com', 67890123, NULL, NULL),('emily.r', 'emily.r@example.com',
78901234, '1995-12-25', 'F'),('frank88', 'frank88@example.com', 89012345, NULL,
NULL),('gina_l', 'gina.l@example.com', 90123456, '1992-03-08', 'F'),('harry_p',
'harry.p@example.com', 12345679, NULL, NULL),('jane_smith',
'jane.smith@example.com', 23456789, '1985-07-21', 'F'),('john_doe',
'john.doe@example.com', 12345678, '1990-01-15', 'M');
```

4.36 user_download_game Table

The user_download_game table represents the M:N relationship “download” between users and games offered on the Netflix Game program.

4.36.1 user_download_game DDL Script

```
CREATE TABLE user_download_game (
    download_id int(11) NOT NULL COMMENT 'Download ID',
    game_name varchar(50) NOT NULL COMMENT 'Game name',
    username varchar(20) NOT NULL COMMENT 'User name',
    download_date date NOT NULL COMMENT 'The date the game is downloaded
by the user.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE user_download_game
```

```

ADD PRIMARY KEY (download_id),
ADD KEY username (username),
ADD KEY game_name (game_name);
ALTER TABLE user_download_game
    MODIFY download_id int(11) NOT NULL AUTO_INCREMENT COMMENT
'Download ID', AUTO_INCREMENT=32;
ALTER TABLE user_download_game
    ADD CONSTRAINT user_download_game_ibfk_1 FOREIGN KEY (username)
REFERENCES user_account (username),
    ADD CONSTRAINT user_download_game_ibfk_2 FOREIGN KEY
(game_name) REFERENCES game (game_name);

```

4.36.2 user_download_game table example data

```

INSERT INTO user_download_game (download_id, game_name, username,
download_date) VALUES
(1, 'Asphalt Xtreme', 'alice_b', '2024-01-10'),(2, 'Shatter Remastered', 'alice_b',
'2024-11-28'),(3, 'Stranger Things: 1984', 'alice_b', '2024-12-30'),(4, 'Asphalt
Xtreme', 'bob_the_builder', '2024-11-11'),(5, 'Exploding Kittens', 'bob_the_builder',
'2024-02-12'),(6, 'Stranger Things: 1984', 'bob_the_builder', '2024-11-30'),(7,
'Asphalt Xtreme', 'charlie123', '2024-11-05'),(8, 'Exploding Kittens', 'charlie123',
'2024-12-01'),(9, 'Heads Up!', 'charlie123', '2024-03-05'),(10, 'Exploding Kittens',
'david_k', '2024-11-10'),(11, 'Heads Up!', 'david_k', '2024-12-05'),(12, 'Hextech
Mayhem', 'david_k', '2024-04-01'),(13, 'Heads Up!', 'emily.r', '2024-11-12'),(14,
'Hextech Mayhem', 'emily.r', '2024-12-12'),(15, 'Into the Breach', 'emily.r',
'2024-05-15'),(16, 'Hextech Mayhem', 'frank88', '2024-11-15'),(17, 'Into the
Breach', 'frank88', '2024-12-15'),(18, 'Kentucky Route Zero', 'frank88',
'2024-06-20'),(19, 'Into the Breach', 'gina_l', '2024-11-18'),(20, 'Kentucky Route
Zero', 'gina_l', '2024-12-20'),(21, 'Moonlighter', 'gina_l', '2024-07-17'),(22,
'Kentucky Route Zero', 'harry_p', '2024-11-20'),(23, 'Moonlighter', 'harry_p',
'2024-12-22'),(24, 'Oxenfree', 'harry_p', '2024-08-22'),(25, 'Heads Up!',
'jane_smith', '2024-12-03'),(26, 'Moonlighter', 'jane_smith', '2024-11-22'),(27,
'Oxenfree', 'jane_smith', '2024-12-25'),(28, 'Shatter Remastered', 'jane_smith',
'2024-09-09'),(29, 'Oxenfree', 'john_doe', '2024-11-25'),(30, 'Shatter Remastered',
'john_doe', '2024-12-28'),(31, 'Stranger Things: 1984', 'john_doe', '2024-10-30');

```

4.37 user_shown_ads Table

The user_shown_ads table logs when a user with an ad-enabled subscription plan is shown ads.

4.37.1 user_shown_ads DDL Script

```
CREATE TABLE user_shown_ads (
    showads_id int(11) NOT NULL COMMENT 'Showing Ads ID',
    ads_title varchar(20) NOT NULL COMMENT 'The title of an ad.',
    user_name varchar(20) NOT NULL COMMENT 'User name',
    show_datetime datetime NOT NULL COMMENT 'The datetime when this ad
was shown to the user.',
    is_skipped tinyint(1) NOT NULL COMMENT 'Mark if the ad was skipped.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE user_shown_ads
    ADD PRIMARY KEY (showads_id),
    ADD KEY user_name (user_name),
    ADD KEY ads_title (ads_title);

ALTER TABLE user_shown_ads
    ADD CONSTRAINT user_shown_ads_ibfk_1 FOREIGN KEY (user_name)
REFERENCES user_account (username),
    ADD CONSTRAINT user_shown_ads_ibfk_2 FOREIGN KEY (ads_title)
REFERENCES ads (ads_title);
```

4.37.2 user_shown_ads table example data

```
INSERT INTO user_shown_ads (showads_id, ads_title, user_name,
show_datetime, is_skipped) VALUES
(1, 'Electric Cars', 'alice_b', '2024-11-01 08:30:00', 1),(2, 'Gadget Promotion',
'alice_b', '2024-11-01 09:00:00', 1),(3, 'Healthy Snacks', 'alice_b', '2024-11-01
09:30:00', 1),(4, 'Holiday Travel', 'alice_b', '2024-11-01 10:00:00', 0),(5, 'New
Movie Release', 'alice_b', '2024-11-01 10:30:00', 1),(6, 'Streaming Deals',
'alice_b', '2024-11-01 11:00:00', 1),(7, 'Electric Cars', 'charlie123', '2024-11-02
08:30:00', 1),(8, 'Gadget Promotion', 'charlie123', '2024-11-02 09:00:00', 0),(9,
'Healthy Snacks', 'charlie123', '2024-11-02 09:30:00', 1),(10, 'Holiday Travel',
'charlie123', '2024-11-02 10:00:00', 1),(11, 'New Movie Release', 'charlie123',
'2024-11-02 10:30:00', 1),(12, 'Streaming Deals', 'charlie123', '2024-11-02
11:00:00', 1),(13, 'Electric Cars', 'jane_smith', '2024-11-03 08:30:00', 1),(14,
'Gadget Promotion', 'jane_smith', '2024-11-03 09:00:00', 1),(15, 'Healthy Snacks',
'jane_smith', '2024-11-03 09:30:00', 0),(16, 'Holiday Travel', 'jane_smith',
'2024-11-03 10:00:00', 1),(17, 'New Movie Release', 'jane_smith', '2024-11-03
10:30:00', 1),(18, 'Streaming Deals', 'jane_smith', '2024-11-03 11:00:00', 1),(19,
'Electric Cars', 'bob_the_builder', '2024-11-04 08:30:00', 1),(20, 'Gadget
Promotion', 'bob_the_builder', '2024-11-04 09:00:00', 1),(21, 'Healthy Snacks',
```

```
'bob_the_builder', '2024-11-04 09:30:00', 1),(22, 'Holiday Travel',
'bob_the_builder', '2024-11-04 10:00:00', 0),(23, 'New Movie Release',
'bob_the_builder', '2024-11-04 10:30:00', 1),(24, 'Streaming Deals',
'bob_the_builder', '2024-11-04 11:00:00', 1),(25, 'Electric Cars', 'emily.r',
'2024-11-05 08:30:00', 1),(26, 'Gadget Promotion', 'emily.r', '2024-11-05
09:00:00', 1),(27, 'Healthy Snacks', 'emily.r', '2024-11-05 09:30:00', 1),(28,
'Holiday Travel', 'emily.r', '2024-11-05 10:00:00', 0),(29, 'New Movie Release',
'emily.r', '2024-11-05 10:30:00', 1),(30, 'Streaming Deals', 'emily.r', '2024-11-05
11:00:00', 1);
```

4.38 user_view_article Table

The user_view_article table logs whenever a user views an article on Netflix Tudum.

4.38.1 user_view_article DDL Script

```
CREATE TABLE user_view_article (
    viewarticle_id int(11) NOT NULL COMMENT 'View article ID',
    username varchar(20) NOT NULL COMMENT 'User name',
    article_title varchar(100) NOT NULL COMMENT 'Article title',
    article_view_date date NOT NULL COMMENT 'The date the article is viewed.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE user_view_article
    ADD PRIMARY KEY (viewarticle_id),
    ADD KEY fk_viewarticle_username (username),
    ADD KEY fk_viewarticle_articletitle (article_title);
ALTER TABLE user_view_article
    MODIFY viewarticle_id int(11) NOT NULL AUTO_INCREMENT COMMENT
    'View article ID', AUTO_INCREMENT=36;
ALTER TABLE user_view_article
    ADD CONSTRAINT fk_viewarticle_articletitle FOREIGN KEY (article_title)
REFERENCES article_tudum (article_title),
    ADD CONSTRAINT fk_viewarticle_username FOREIGN KEY (username)
REFERENCES user_account (username);
```

4.38.2 user_view_article table example data

```
INSERT INTO user_view_article (viewarticle_id, username, article_title,
article_view_date) VALUES
(1, 'bob_the_builder', 'Sadie Sink on Max's Trip to the Upside Down in 'Stranger
Things' Season 4', '2024-02-11'),
```

- (2, 'bob_the_builder', 'Sadie Sink on Max's Trip to the Upside Down in 'Stranger Things' Season 4', '2024-11-15'),
- (3, 'bob_the_builder', 'Sadie Sink on Max's Trip to the Upside Down in 'Stranger Things' Season 4', '2024-12-05'),
- (4, 'charlie123', 'The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two Volumes', '2024-03-05'),
- (5, 'charlie123', 'The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two Volumes', '2024-11-17'),
- (6, 'charlie123', 'The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two Volumes', '2024-12-07'),
- (7, 'alice_b', 'The Queen's Gambit Cast, News, Videos and more', '2024-01-10'),
- (8, 'alice_b', 'The Queen's Gambit Cast, News, Videos and more', '2024-11-11'),
- (9, 'alice_b', 'The Queen's Gambit Cast, News, Videos and more', '2024-12-03'),
- (10, 'david_k', 'Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship', '2024-04-01'),
- (11, 'david_k', 'Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship', '2024-11-18'),
- (12, 'david_k', 'Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship', '2024-12-10'),
- (13, 'emily.r', "Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far", '2024-05-15'),
- (14, 'emily.r', "Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far", '2024-11-20'),
- (15, 'emily.r', "Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far", '2024-12-12'),
- (16, 'frank88', "Stranger Things' Cast Red Carpet Premiere Photos", '2024-06-20'),
- (17, 'frank88', "Stranger Things' Cast Red Carpet Premiere Photos", '2024-11-22'),
- (18, 'frank88', "Stranger Things' Cast Red Carpet Premiere Photos", '2024-12-15'),
- (19, 'gina_l', "Stranger Things' Season 4 Easter Eggs", '2024-07-17'),
- (20, 'gina_l', "Stranger Things' Season 4 Easter Eggs", '2024-11-25'),
- (21, 'gina_l', "Stranger Things' Season 4 Easter Eggs", '2024-12-17'),
- (22, 'harry_p', "Stranger Things' Season 4 Episode Length Revealed", '2024-08-22'),
- (23, 'harry_p', "Stranger Things' Season 4 Episode Length Revealed", '2024-11-26'),
- (24, 'harry_p', "Stranger Things' Season 4 Episode Length Revealed", '2024-12-18'),

```

(25, 'jane_smith', "Stranger Things' Season 4 First Look Photos", '2024-09-09'),
(26, 'bob_the_builder', 'Sadie Sink on Max's Trip to the Upside Down in 'Stranger
Things' Season 4', '2024-10-05'),
(27, 'charlie123', 'The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two
Volumes', '2024-10-10'),
(28, 'alice_b', 'The Queen\l's Gambit Cast, News, Videos and more',
'2024-10-15'),
(29, 'david_k', 'Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship',
'2024-10-20'),
(30, 'frank88', "Stranger Things' Cast Red Carpet Premiere Photos",
'2024-10-25'),
(31, 'bob_the_builder', 'Sadie Sink on Max's Trip to the Upside Down in 'Stranger
Things' Season 4', '2024-10-05'),
(32, 'charlie123', 'The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two
Volumes', '2024-10-10'),
(33, 'alice_b', 'The Queen\l's Gambit Cast, News, Videos and more',
'2024-10-15'),
(34, 'david_k', 'Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship',
'2024-10-20'),
(35, 'frank88', "Stranger Things' Cast Red Carpet Premiere Photos",
'2024-10-25');

```

4.39 user_view_content Table

The user_view_content table logs whenever a user decides to view a show on Netflix.

4.39.1 user_view_content DDL Script

```

CREATE TABLE user_view_content (
    viewcontent_id int(11) NOT NULL COMMENT 'View Content ID',
    username varchar(20) NOT NULL COMMENT 'User name',
    content_id int(11) NOT NULL COMMENT 'Content ID',
    content_view_date date NOT NULL COMMENT 'The date the content is
viewed.'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE user_view_content
    ADD PRIMARY KEY (viewcontent_id),
    ADD KEY fk_viewcontent_username (username),
    ADD KEY fk_viewcontent_contentid (content_id);
ALTER TABLE user_view_content

```

```

MODIFY viewcontent_id int(11) NOT NULL AUTO_INCREMENT COMMENT
'View Content ID', AUTO_INCREMENT=31;
ALTER TABLE user_view_content
    ADD CONSTRAINT fk_viewcontent_contentid FOREIGN KEY (content_id)
REFERENCES content (content_id),
    ADD CONSTRAINT fk_viewcontent_username FOREIGN KEY (username)
REFERENCES user_account (username);

```

4.39.2 user_view_content table example data

```

INSERT INTO user_view_content (viewcontent_id, username, content_id,
content_view_date) VALUES
(1, 'alice_b', 1, '2024-01-10'),(2, 'alice_b', 2, '2024-03-01'),(3, 'bob_the_builder', 2,
'2024-01-12'),(4, 'charlie123', 3, '2024-01-15'),(5, 'bob_the_builder', 4,
'2024-03-03'),(6, 'david_k', 4, '2024-01-18'),(7, 'emily.r', 5, '2024-01-20'),(8,
'charlie123', 6, '2024-03-05'),(9, 'frank88', 6, '2024-01-22'),(10, 'gina_l', 7,
'2024-01-25'),(11, 'david_k', 8, '2024-03-08'),(12, 'harry_p', 8, '2024-01-27'),(13,
'jane_smith', 9, '2024-01-30'),(14, 'emily.r', 10, '2024-03-10'),(15, 'john_doe', 10,
'2024-02-01'),(16, 'alice_b', 11, '2024-02-05'),(17, 'bob_the_builder', 12,
'2024-02-07'),(18, 'frank88', 12, '2024-03-12'),(19, 'charlie123', 13,
'2024-02-09'),(20, 'david_k', 14, '2024-02-11'),(21, 'gina_l', 14, '2024-03-15'),(22,
'emily.r', 15, '2024-02-13'),(23, 'frank88', 16, '2024-02-16'),(24, 'harry_p', 16,
'2024-03-17'),(25, 'gina_l', 17, '2024-02-18'),(26, 'harry_p', 18, '2024-02-20'),(27,
'jane_smith', 18, '2024-03-19'),(28, 'jane_smith', 19, '2024-02-22'),(29, 'john_doe',
20, '2024-02-24'),(30, 'john_doe', 20, '2024-03-21');

```

5 SQL Queries

5.1 Basic SQL Queries

5.1.1 Merchandise with Discounts

Report	Merchandise with Discounts																
Objective	Retrieve all merchandise items that currently have a discounted price, along with the percentage discount.																
Query																	
SELECT item_name, price, discounted_price, ROUND(((price - discounted_price) / price) * 100, 2) AS discount_percentage FROM merchandise_item WHERE discounted_price < price;																	
Query Result																	
<table><thead><tr><th>item_name Item Name</th><th>price Price</th><th>discounted_price Discounted price.</th><th>discount_percentage</th></tr></thead><tbody><tr><td>Squid Game Tracksuit</td><td>49.99</td><td>44.99</td><td>10.00</td></tr><tr><td>Stranger Things Mug</td><td>15.99</td><td>12.99</td><td>18.76</td></tr><tr><td>Squid Game Glass Bottle</td><td>25.99</td><td>22.99</td><td>11.54</td></tr></tbody></table>		item_name Item Name	price Price	discounted_price Discounted price.	discount_percentage	Squid Game Tracksuit	49.99	44.99	10.00	Stranger Things Mug	15.99	12.99	18.76	Squid Game Glass Bottle	25.99	22.99	11.54
item_name Item Name	price Price	discounted_price Discounted price.	discount_percentage														
Squid Game Tracksuit	49.99	44.99	10.00														
Stranger Things Mug	15.99	12.99	18.76														
Squid Game Glass Bottle	25.99	22.99	11.54														

5.1.2 Profiles with Kids' Restrictions

Report	Profiles with Kids' Restrictions
Objective	Find all user profiles designated as kids' profiles, including the associated maturity rating.
Query	
SELECT profile_name, username,	

```

maturity_rating
FROM
profile
WHERE
kids_profile_status = 1;

```

Query Result

profile_name Profile name	username User name	maturity_rating Maturity rating
LilBuddy	bob_the_builder	G
LilDreamer	emily.r	G
MiniExplorer	charlie123	G
SpaceCadet	david_k	G
TinyAdventurer	frank88	G
TinyTots	alice_b	G
WonderKid	gina_l	G

5.1.3 Number of Active Subscriptions

Report	Number of Active Subscriptions		
Objective	Count the total number of active subscriptions in the database.		
Query			
<pre> SELECT COUNT(subscription_id) AS active_subscriptions FROM subscription WHERE termination_date IS NULL; </pre>			
Query Result			
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>active_subscriptions</td> </tr> <tr> <td>15</td> </tr> </table>		active_subscriptions	15
active_subscriptions			
15			

5.1.4 Maximum Downloads for a Single Game

Report	Maximum Downloads for a Single Game
---------------	-------------------------------------

Objective	Find the highest number of downloads recorded for any game.												
Query													
<pre>SELECT usa.user_name, COUNT(usa.showads_id) AS total_ads_shown FROM user_shown_ads usa GROUP BY usa.user_name ORDER BY total_ads_shown DESC;</pre>													
Query Result													
<table border="1"> <thead> <tr> <th>user_name</th> <th>total_ads_shown</th> </tr> </thead> <tbody> <tr> <td>charlie123</td> <td>6</td> </tr> <tr> <td>emily.r</td> <td>6</td> </tr> <tr> <td>alice_b</td> <td>6</td> </tr> <tr> <td>jane_smith</td> <td>6</td> </tr> <tr> <td>bob_the_builder</td> <td>6</td> </tr> </tbody> </table>		user_name	total_ads_shown	charlie123	6	emily.r	6	alice_b	6	jane_smith	6	bob_the_builder	6
user_name	total_ads_shown												
charlie123	6												
emily.r	6												
alice_b	6												
jane_smith	6												
bob_the_builder	6												

5.1.5 Coupon Usage Analysis

Report	Coupon Usage Analysis
Objective	Determine the number of times each coupon code has been used
Query	
<pre>SELECT promo_code, COUNT(*) AS usage_count FROM order_use_coupon GROUP BY promo_code ORDER BY usage_count DESC;</pre>	
Query Result	

promo_code	usage_count
SUMMER10	6
WITCHER5	5
STRANGE10	3
BRIDGE15	3
NEWYEAR5	3
XMAS2024	3
FALLDEAL	2
SQUID20	2
BLACKFRI	2
SPRING25	1

5.1.6 Daily Ad Skipping Rate

Report	Daily Ad Skipping Rate												
Objective	Calculate the percentage of ads skipped on each date												
Query													
<pre> SELECT DATE(show_datetime) AS ad_date, COUNT(CASE WHEN is_skipped = 1 THEN 1 END) * 100.0 / COUNT(*) AS skip_rate FROM user_shown_ads GROUP BY ad_date ORDER BY ad_date ASC; </pre>													
Query Result													
<table border="1"> <thead> <tr> <th>ad_date</th> <th>skip_rate</th> </tr> </thead> <tbody> <tr> <td>2024-11-01</td> <td>83.33333</td> </tr> <tr> <td>2024-11-02</td> <td>83.33333</td> </tr> <tr> <td>2024-11-03</td> <td>83.33333</td> </tr> <tr> <td>2024-11-04</td> <td>83.33333</td> </tr> <tr> <td>2024-11-05</td> <td>83.33333</td> </tr> </tbody> </table>		ad_date	skip_rate	2024-11-01	83.33333	2024-11-02	83.33333	2024-11-03	83.33333	2024-11-04	83.33333	2024-11-05	83.33333
ad_date	skip_rate												
2024-11-01	83.33333												
2024-11-02	83.33333												
2024-11-03	83.33333												
2024-11-04	83.33333												
2024-11-05	83.33333												

5.1.7 User Activity on Articles

Report	User Activity on Articles
Objective	Trac the total number of articles viewed by each user.
Query	
SELECT username, COUNT(*) AS total_articles_viewed FROM user_view_article GROUP BY username ORDER BY total_articles_viewed DESC LIMIT 10;	

username	total_articles_viewed
david_k	5
alice_b	5
bob_the_builder	5
frank88	5
charlie123	5
harry_p	3
emily.r	3
gina_l	3
jane_smith	1

5.1.8 Merchandise Stock Summary

Report	Merchandise Stock Summary
Objective	Finn the total and average stock quantities for all merchandise items.
Query	

```

SELECT
    SUM(stock_quantity) AS total_stock,
    AVG(stock_quantity) AS average_stock
FROM
    merchandise_item;

```

Query Result

total_stock	average_stock
920	76.6667

5.1.9 Average Ad Duration

Report	Average Ad Duration		
Objective	Calculate the average duration of all advertisements		
Query			
SELECT AVG(duration) AS average_duration FROM ads;			
Query Result			
<table> <thead> <tr> <th>average_duration</th> </tr> </thead> <tbody> <tr> <td>30.0000</td> </tr> </tbody> </table>		average_duration	30.0000
average_duration			
30.0000			

5.1.10 Top Performing Advertisements

Report	Top Performing Advertisements
Objective	Identify the most frequently shown ads.
Query	
SELECT ads_title, COUNT(*) AS times_shown FROM user_shown_ads GROUP BY ads_title ORDER BY times_shown DESC LIMIT 5;	

Query Result

ads_title	times_shown
New Movie Release	5
Gadget Promotion	5
Streaming Deals	5
Healthy Snacks	5
Holiday Travel	5

5.1.11 Content View Count

Report	Content View Count																						
Objective	Determine the most-watched content over time																						
Query																							
SELECT content_id, COUNT(*) AS view_count FROM user_view_content GROUP BY content_id ORDER BY view_count DESC LIMIT 10;																							
Query Result																							
<table><thead><tr><th>content_id</th><th>view_count</th></tr></thead><tbody><tr><td>4</td><td>2</td></tr><tr><td>20</td><td>2</td></tr><tr><td>6</td><td>2</td></tr><tr><td>8</td><td>2</td></tr><tr><td>10</td><td>2</td></tr><tr><td>12</td><td>2</td></tr><tr><td>14</td><td>2</td></tr><tr><td>16</td><td>2</td></tr><tr><td>2</td><td>2</td></tr><tr><td>18</td><td>2</td></tr></tbody></table>		content_id	view_count	4	2	20	2	6	2	8	2	10	2	12	2	14	2	16	2	2	2	18	2
content_id	view_count																						
4	2																						
20	2																						
6	2																						
8	2																						
10	2																						
12	2																						
14	2																						
16	2																						
2	2																						
18	2																						

5.1.12 Plan popularity

Report	Plan Popularity												
Objective	Determine the most subscribed plans.												
Query													
SELECT plan_name, COUNT(*) AS subscriber_count FROM subscription GROUP BY plan_name ORDER BY subscriber_count DESC;													
Query Result													
<table><thead><tr><th>plan_name</th><th>subscriber_count</th></tr></thead><tbody><tr><td>PREMIUM</td><td>8</td></tr><tr><td>BASIC</td><td>7</td></tr><tr><td>STANDARD</td><td>6</td></tr><tr><td>BASICADS</td><td>5</td></tr><tr><td>MOBILE</td><td>4</td></tr></tbody></table>		plan_name	subscriber_count	PREMIUM	8	BASIC	7	STANDARD	6	BASICADS	5	MOBILE	4
plan_name	subscriber_count												
PREMIUM	8												
BASIC	7												
STANDARD	6												
BASICADS	5												
MOBILE	4												

5.1.13 Count Kids Profiles

Report	Count Kids Profiles		
Objective	Find the total number of profiles marked as kids' profiles.		
Query			
SELECT COUNT(*) AS total_kids_profiles FROM profile WHERE kids_profile_status = 1;			
Query Result			
<table><thead><tr><th>total_kids_profiles</th></tr></thead><tbody><tr><td>7</td></tr></tbody></table>		total_kids_profiles	7
total_kids_profiles			
7			

5.1.14 Distribution of Apparel Colors

Report	Distribution of Apparel Colors																
Objective	Understand the variety and distribution of colors available for items.																
Query																	
SELECT color, COUNT(*) AS count FROM apparel_color GROUP BY color ORDER BY count DESC;																	
Query Result																	
<table><thead><tr><th>color</th><th>count</th></tr></thead><tbody><tr><td>Black</td><td>2</td></tr><tr><td>Red</td><td>1</td></tr><tr><td>Gray</td><td>1</td></tr><tr><td>Blue</td><td>1</td></tr><tr><td>White</td><td>1</td></tr><tr><td>Green</td><td>1</td></tr><tr><td>Yellow</td><td>1</td></tr></tbody></table>		color	count	Black	2	Red	1	Gray	1	Blue	1	White	1	Green	1	Yellow	1
color	count																
Black	2																
Red	1																
Gray	1																
Blue	1																
White	1																
Green	1																
Yellow	1																

5.1.15 Profiles with PIN Enabled

Report	Profiles with PIN Enabled		
Objective	Count how many profiles have a PIN set.		
Query			
SELECT COUNT(*) AS profiles_with_pin FROM profile WHERE pin IS NOT NULL;			
Query Result			
<table><thead><tr><th>profiles_with_pin</th><th>8</th></tr></thead></table>		profiles_with_pin	8
profiles_with_pin	8		

5.1.16 Sizes Availability Insights

Report	Sizes Availability Insights												
Objective	Determine the availability of different sizes to track inventory trends.												
Query													
SELECT size, COUNT(*) AS count FROM apparel_size GROUP BY size ORDER BY count DESC;													
Query Result													
<p>A screenshot of a data visualization interface. At the top, there is a header with two columns: 'size' and 'count'. Below the header, the text 'Available size of the apparel.' is displayed. The data is presented in a table with four rows. The first row has a gray background and contains the column headers. The subsequent three rows have white backgrounds and list the sizes L, M, S, and XL along with their respective counts 4, 3, 2, and 2. The counts are aligned with the right edge of the 'count' column.</p> <table><thead><tr><th>size</th><th>count</th></tr></thead><tbody><tr><td colspan="2">Available size of the apparel.</td></tr><tr><td>L</td><td>4</td></tr><tr><td>M</td><td>3</td></tr><tr><td>S</td><td>2</td></tr><tr><td>XL</td><td>2</td></tr></tbody></table>		size	count	Available size of the apparel.		L	4	M	3	S	2	XL	2
size	count												
Available size of the apparel.													
L	4												
M	3												
S	2												
XL	2												

5.1.17 Top Advertisers by Total Ad Duration

Report	Top Advertisers by Total Ad Duration
Objective	Determine the most subscribed plans.
Query	
SELECT advertiser, SUM(duration) AS total_ad_duration FROM ads GROUP BY advertiser ORDER BY total_ad_duration DESC;	
Query Result	

Advertiser		The advertiser of this ad.	total_ad_duration
	Advertiser Name		Duration
<input type="checkbox"/>	Edit Copy Delete StreamPlus	StreamPlus	60
<input type="checkbox"/>	Edit Copy Delete GoGlobalTrips	GoGlobalTrips	60
<input type="checkbox"/>	Edit Copy Delete InnovaTech	InnovaTech	45
<input type="checkbox"/>	Edit Copy Delete FashionHub	FashionHub	45
<input type="checkbox"/>	Edit Copy Delete EcoMotors	EcoMotors	30
<input type="checkbox"/>	Edit Copy Delete FilmStudio	FilmStudio	30
<input type="checkbox"/>	Edit Copy Delete Foodies	Foodies	10
<input type="checkbox"/>	Edit Copy Delete ShopNow	ShopNow	10
<input type="checkbox"/>	Edit Copy Delete FitStore	FitStore	5
<input type="checkbox"/>	Edit Copy Delete TechWorld	TechWorld	5

5.1.18 Most Popular Content Maturity Ratings

Report	Most Popular Content Maturity Ratings
Objective	Determine the count of content entries grouped by their maturity ratings to understand audience preferences
Query	
<pre>SELECT maturity_rating, COUNT(*) AS content_count FROM content GROUP BY maturity_rating ORDER BY content_count DESC;</pre>	
Query Result	
maturity_rating	content_count
Maturity rating	1
TV-MA	9
R	6
TV-14	3
PG-13	2

5.1.19 Highest-Rated Merchandise

Report	Highest-Rated Merchandise																										
Objective	Find the average rating for each merchandise item and rank them.																										
Query																											
SELECT item_id, AVG(rating) AS average_rating FROM merchandise_review GROUP BY item_id ORDER BY average_rating DESC;																											
Query Result																											
<table border="1"> <thead> <tr> <th>item_id Item ID</th><th>average_rating</th></tr> </thead> <tbody> <tr><td>6</td><td>5.0000</td></tr> <tr><td>7</td><td>5.0000</td></tr> <tr><td>9</td><td>5.0000</td></tr> <tr><td>11</td><td>5.0000</td></tr> <tr><td>12</td><td>5.0000</td></tr> <tr><td>2</td><td>5.0000</td></tr> <tr><td>4</td><td>4.6667</td></tr> <tr><td>5</td><td>4.3333</td></tr> <tr><td>1</td><td>4.0000</td></tr> <tr><td>8</td><td>3.6667</td></tr> <tr><td>10</td><td>2.5000</td></tr> <tr><td>3</td><td>2.5000</td></tr> </tbody> </table>		item_id Item ID	average_rating	6	5.0000	7	5.0000	9	5.0000	11	5.0000	12	5.0000	2	5.0000	4	4.6667	5	4.3333	1	4.0000	8	3.6667	10	2.5000	3	2.5000
item_id Item ID	average_rating																										
6	5.0000																										
7	5.0000																										
9	5.0000																										
11	5.0000																										
12	5.0000																										
2	5.0000																										
4	4.6667																										
5	4.3333																										
1	4.0000																										
8	3.6667																										
10	2.5000																										
3	2.5000																										

5.1.20 User Article Engagement

Report	User Article Engagement
Objective	Count how many times each article has been viewed.
Query	
SELECT article_title, COUNT(*) AS view_count FROM user_view_article GROUP BY article_title ORDER BY view_count DESC;	
Query Result	

article_title	view_count
Article title	
The Duffer Brothers on Why 'Stranger Things' Season 4 Is Two Volumes	5
The Queen's Gambit Cast, News, Videos and more	5
Who Is Eddie? Steve and Dustin's 'Stranger Things' Friendship	5
Sadie Sink on Max's Trip to the Upside Down in 'Stranger Things' Season 4	5
'Stranger Things' Cast Red Carpet Premiere Photos	5
'Stranger Things' Season 4 Easter Eggs	3
'Stranger Things' Season 4 Episode Length Revealed	3
'Stranger Things' Animated Series Announced: What The Duffer Brothers Revealed So Far	3
'Stranger Things' Season 4 First Look Photos	1

5.1.21 Total Stock by Collection Category

Report	Total Stock by Collection Category												
Objective	Calculate the total stock available for each collection category to identify inventory levels.												
Query													
<pre> SELECT collection_category, SUM(stock_quantity) AS total_stock FROM merchandise_item GROUP BY collection_category ORDER BY total_stock DESC; </pre>													
Query Result													
<table border="1"> <thead> <tr> <th>collection_category</th> <th>total_stock</th> </tr> </thead> <tbody> <tr> <td>The collection category this item belongs to.</td> <td></td> </tr> <tr> <td>The Witcher Collection</td> <td>290</td> </tr> <tr> <td>Stranger Things Collection</td> <td>250</td> </tr> <tr> <td>Squid Game Collection</td> <td>200</td> </tr> <tr> <td>Bridgerton Collection</td> <td>180</td> </tr> </tbody> </table>		collection_category	total_stock	The collection category this item belongs to.		The Witcher Collection	290	Stranger Things Collection	250	Squid Game Collection	200	Bridgerton Collection	180
collection_category	total_stock												
The collection category this item belongs to.													
The Witcher Collection	290												
Stranger Things Collection	250												
Squid Game Collection	200												
Bridgerton Collection	180												

5.1.22 The Average Price of All Items in a Collection

Report	The Average Price of All Items in a Collection										
Objective	Find the average item price in each collection category.										
Query											
<pre>SELECT collection_category, AVG(price) AS avg_price FROM merchandise_item GROUP BY collection_category ORDER BY avg_price DESC;</pre>											
Query Result											
<p>A screenshot of a database query result. The columns are labeled 'collection_category' and 'avg_price'. The data shows four rows: Squid Game Collection (31.990000), Bridgerton Collection (23.656667), The Witcher Collection (20.990000), and Stranger Things Collection (20.323333). The 'avg_price' column has a dropdown arrow icon.</p> <table><thead><tr><th>collection_category</th><th>avg_price</th></tr></thead><tbody><tr><td>Squid Game Collection</td><td>31.990000</td></tr><tr><td>Bridgerton Collection</td><td>23.656667</td></tr><tr><td>The Witcher Collection</td><td>20.990000</td></tr><tr><td>Stranger Things Collection</td><td>20.323333</td></tr></tbody></table>		collection_category	avg_price	Squid Game Collection	31.990000	Bridgerton Collection	23.656667	The Witcher Collection	20.990000	Stranger Things Collection	20.323333
collection_category	avg_price										
Squid Game Collection	31.990000										
Bridgerton Collection	23.656667										
The Witcher Collection	20.990000										
Stranger Things Collection	20.323333										

5.1.23 Count the Number of Shows by Maturity Rating

Report	Count the Number of Shows by Maturity Rating
Objective	This query provides an overview of the count of shows grouped by their maturity ratings.
Query	
<pre>SELECT maturity_rating, COUNT(*) AS show_count FROM content GROUP BY maturity_rating ORDER BY show_count DESC;</pre>	
Query Result	

maturity_rating	show_count
Maturity rating	1
TV-MA	9
R	6
TV-14	3
PG-13	2

5.1.24 Number of articles by category

Report	Number of articles by category												
Objective	Count the number of articles by category.												
Query													
SELECT category, COUNT(*) AS article_count FROM article_tudum GROUP BY category ORDER BY article_count DESC;													
Query Result													
<table border="1"> <thead> <tr> <th>category</th> <th>article_count</th> </tr> </thead> <tbody> <tr> <td>The category of an article.</td> <td>1</td> </tr> <tr> <td>news</td> <td>4</td> </tr> <tr> <td>meet the cast</td> <td>3</td> </tr> <tr> <td>deep dive</td> <td>2</td> </tr> <tr> <td>explainer</td> <td>1</td> </tr> </tbody> </table>		category	article_count	The category of an article.	1	news	4	meet the cast	3	deep dive	2	explainer	1
category	article_count												
The category of an article.	1												
news	4												
meet the cast	3												
deep dive	2												
explainer	1												

5.2 Advanced SQL Queries

5.2.1 Top-Rated Merchandise Items

Report	Top-Rated Merchandise Items																				
Objective	Find the top 5 merchandise items with the highest average customer rating.																				
Query	<pre>SELECT mi.item_name, ROUND(AVG(mr.rating), 2) AS average_rating, COUNT(mr.review_id) AS total_reviews FROM merchandise_item mi JOIN merchandise_review mr ON mi.item_id = mr.item_id GROUP BY mi.item_name ORDER BY average_rating DESC, total_reviews DESC LIMIT 5;</pre>																				
Query Result																					
<table><thead><tr><th>item_name</th><th>average_rating</th><th>total_reviews</th></tr></thead><tbody><tr><td>The Witcher Tumbler</td><td>5.00</td><td>3</td></tr><tr><td>Stranger Things Funko Pop</td><td>5.00</td><td>3</td></tr><tr><td>Bridgerton Crown Replica</td><td>5.00</td><td>2</td></tr><tr><td>Squid Game Figure</td><td>5.00</td><td>2</td></tr><tr><td>The Witcher Hoodie</td><td>5.00</td><td>2</td></tr></tbody></table>				item_name	average_rating	total_reviews	The Witcher Tumbler	5.00	3	Stranger Things Funko Pop	5.00	3	Bridgerton Crown Replica	5.00	2	Squid Game Figure	5.00	2	The Witcher Hoodie	5.00	2
item_name	average_rating	total_reviews																			
The Witcher Tumbler	5.00	3																			
Stranger Things Funko Pop	5.00	3																			
Bridgerton Crown Replica	5.00	2																			
Squid Game Figure	5.00	2																			
The Witcher Hoodie	5.00	2																			

5.2.2 Game Download Count

Report	Game Download Count	
Objective	Identify how many times each game has been downloaded.	
Query	<pre>SELECT g.game_name, COUNT(udg.download_id) AS download_count FROM game g JOIN user_download_game udg ON g.game_name = udg.game_name GROUP BY g.game_name ORDER BY download_count DESC;</pre>	
Query Result		

game_name	download_count
Heads Up!	4
Stranger Things: 1984	3
Moonlighter	3
Hextech Mayhem	3
Asphalt Xtreme	3
Shatter Remastered	3
Kentucky Route Zero	3
Oxenfree	3
Into the Breach	3
Exploding Kittens	3

5.2.3 Top-Selling Merchandise by Revenue

Report	Top-Selling Merchandise by Revenue
Objective	Identify the top 5 merchandise items that generated the most revenue, including their total sales revenue.
Query	
<pre> SELECT mi.item_name, SUM(o.quantity * mi.price) AS total_revenue FROM merchandise_item mi JOIN order_contains_item o ON mi.item_id = o.item_id GROUP BY mi.item_name ORDER BY total_revenue DESC LIMIT 5; </pre>	
Query Result	

item_name	total_revenue
Stranger Things T-Shirt	359.88
The Witcher Hoodie	319.92
Squid Game Tracksuit	299.94
Bridgerton Crown Replica	249.90
Squid Game Glass Bottle	207.92

5.2.4 Recently Released TV Shows

Report	Most Popular Games by Downloads									
Objective	Retrieve the titles of TV shows released in the last 2 years and their series status.									
Query										
<pre> SELECT c.content_title AS tv_show_title, ts.series_status, c.release_date FROM content c JOIN tv_show ts ON c.content_id = ts.content_id WHERE c.release_date >= DATE_ADD(CURRENT_DATE(), INTERVAL -2 YEAR); </pre>										
Query Result										
<table border="1"> <thead> <tr> <th>tv_show_title</th> <th>series_status</th> <th>release_date</th> </tr> </thead> <tbody> <tr> <td>A Killer Paradox</td> <td>Ongoing</td> <td>2024-01-15</td> </tr> <tr> <td>One Piece</td> <td>Ongoing</td> <td>2023-08-31</td> </tr> </tbody> </table>		tv_show_title	series_status	release_date	A Killer Paradox	Ongoing	2024-01-15	One Piece	Ongoing	2023-08-31
tv_show_title	series_status	release_date								
A Killer Paradox	Ongoing	2024-01-15								
One Piece	Ongoing	2023-08-31								

5.2.5 Retrieve Apparel Colors and Materials

Report	Retrieve Apparel Colors and Materials
Objective	To display all apparel items along with their respective colors and materials, providing a complete overview of available products.
Query	

```

SELECT a.item_id, a.material, ac.color
FROM apparel a
JOIN apparel_color ac ON a.item_id = ac.item_id;

```

Query Result

item_id	material	color
1	100% Cotton	Black
1	100% Cotton	Red
2	Polyester/Cotton Blend	Black
2	Polyester/Cotton Blend	Gray
3	Organic Cotton	Blue
3	Organic Cotton	White
4	Polyester	Green
4	Polyester	Yellow

5.2.6 Find all apparel items available in "Black" color.

Report	Find all apparel items available in "Black" color.
Objective	To identify all apparel items that are available in the "Black" color, useful for targeted inventory queries or promotional campaigns.

Query

```

SELECT a.item_id, a.material
FROM apparel a
JOIN apparel_color ac ON a.item_id = ac.item_id
WHERE ac.color = 'Black';

```

Query Result

item_id	material
1	100% Cotton
2	Polyester/Cotton Blend

5.2.7 List apparel items and sizes where the material is 'Organic Cotton'.

Report	List apparel items and sizes where the material is 'Organic Cotton'.
Objective	To list all apparel items made from "Organic Cotton" along with their available sizes, useful for eco-conscious customer segments.
Query	

```
SELECT a.item_id, a.material, s.size  
FROM apparel a  
JOIN apparel_size s ON a.item_id = s.item_id  
WHERE a.material = 'Organic Cotton';
```

Query Result

item_id	material	size
3	Organic Cotton	L
3	Organic Cotton	M
3	Organic Cotton	S

5.2.8 List All Apparel Items with Color and Size

Report	List All Apparel Items with Color and Size
Objective	Identify ads that relate to articles (via content IDs or similar relationships).
Query	
SELECT ap.item_id, ap.material, ac.color,asz.size FROM apparel ap JOIN apparel_color ac ON ap.item_id = ac.item_id JOIN apparel_size asz ON ap.item_id = asz.item_id;	
Query Result	

item_id	material	color	size
1	100% Cotton	Black	L
1	100% Cotton	Black	M
1	100% Cotton	Black	S
1	100% Cotton	Red	L
1	100% Cotton	Red	M
1	100% Cotton	Red	S
2	Polyester/Cotton Blend	Black	L
2	Polyester/Cotton Blend	Black	M
2	Polyester/Cotton Blend	Black	XL
2	Polyester/Cotton Blend	Gray	L
2	Polyester/Cotton Blend	Gray	M
2	Polyester/Cotton Blend	Gray	XL
3	Organic Cotton	Blue	L
3	Organic Cotton	Blue	M
3	Organic Cotton	Blue	S
3	Organic Cotton	White	L
3	Organic Cotton	White	M
3	Organic Cotton	White	S
4	Polyester	Green	L
4	Polyester	Green	XL
4	Polyester	Yellow	L
4	Polyester	Yellow	XL

5.2.9 Most popular TV show by views.

Report	Most popular TV show by views.
Objective	Identify the TV show with the most views in descending order.
Query	
<pre>SELECT c.content_title, COUNT(v.viewcontent_id) AS view_count FROM tv_show t JOIN user_view_content v ON t.content_id = v.content_id JOIN content c ON t.content_id = c.content_id GROUP BY t.content_id ORDER BY view_count DESC;</pre>	
Query Result	

content_title	view_count
The Crown	2
One Piece	2
The Lincoln Lawyer	2
Squid Game	2
Money Heist	2
Bridgerton	1
Stranger Things	1
The Queen's Gambit	1
A Killer Paradox	1
The Witcher	1

5.2.10 Users Who Viewed Content Across Multiple Dates

Report	Users Who Viewed Content Across Multiple Dates																						
Objective	List users who viewed content on more than one distinct date.																						
Query																							
<pre>SELECT v.username, COUNT(DISTINCT v.content_view_date) AS view_dates FROM user_view_content v GROUP BY v.username HAVING view_dates > 1;</pre>																							
Query Result																							
<table border="1"> <thead> <tr> <th>username</th> <th>view_dates</th> </tr> </thead> <tbody> <tr> <td>alice_b</td> <td>3</td> </tr> <tr> <td>bob_the_builder</td> <td>3</td> </tr> <tr> <td>charlie123</td> <td>3</td> </tr> <tr> <td>david_k</td> <td>3</td> </tr> <tr> <td>emily.r</td> <td>3</td> </tr> <tr> <td>frank88</td> <td>3</td> </tr> <tr> <td>gina_l</td> <td>3</td> </tr> <tr> <td>harry_p</td> <td>3</td> </tr> <tr> <td>jane_smith</td> <td>3</td> </tr> <tr> <td>john_doe</td> <td>3</td> </tr> </tbody> </table>		username	view_dates	alice_b	3	bob_the_builder	3	charlie123	3	david_k	3	emily.r	3	frank88	3	gina_l	3	harry_p	3	jane_smith	3	john_doe	3
username	view_dates																						
alice_b	3																						
bob_the_builder	3																						
charlie123	3																						
david_k	3																						
emily.r	3																						
frank88	3																						
gina_l	3																						
harry_p	3																						
jane_smith	3																						
john_doe	3																						

5.2.11 Users Who Interacted with Content or Games

Report	Users Who Interacted with Content or Games
Objective	Identify all users who either viewed content or downloaded games.
Query	

```

SELECT username FROM user_view_content
UNION
SELECT username FROM user_download_game;

```

Query Result

username
alice_b
bob_the_builder
charlie123
david_k
emily.r
frank88
gina_l
harry_p
jane_smith
john_doe

5.2.12 Gender Percentage of Viewers On Each Content

Report	Gender Percentage of Viewers On Each Content
Objective	Display the gender percentage of viewers on each content. Note that only users subscribed to ad-enabled plans will have their gender data available.
Query	
	<pre> SELECT c.content_title, ROUND(SUM(CASE WHEN ua.gender = 'M' THEN 1 ELSE 0 END) * 100.0 / COUNT(ua.username), 2) AS male_percentage, ROUND(SUM(CASE WHEN ua.gender = 'F' THEN 1 ELSE 0 END) * 100.0 / COUNT(ua.username), 2) AS female_percentage FROM user_view_content uvc JOIN user_account ua ON uvc.username = ua.username JOIN content c ON uvc.content_id = c.content_id WHERE ua.gender IS NOT NULL GROUP BY c.content_title ORDER BY c.content_title; </pre>
Query Result	

content_title	male_percentage	female_percentage
Bridgerton	0.00	100.00
Don't Look Up	0.00	100.00
Enola Holmes	100.00	0.00
Marriage Story	0.00	100.00
One Piece	100.00	0.00
Squid Game	50.00	50.00
Stranger Things	0.00	100.00
The Harder They Fall	100.00	0.00
The Irishman	0.00	100.00
The Lincoln Lawyer	100.00	0.00
The Old Guard	0.00	100.00
The Power of the Dog	0.00	100.00
The Witcher	0.00	100.00

5.2.13 Find Advertisers Whose Ads Were Shown to Users

Report	Find Advertisers Whose Ads Were Shown to Users
Objective	Determine which advertisers reached users through their ad campaigns.
Query	<pre>SELECT DISTINCT a.advertiser, u.user_name FROM ads a JOIN user_shown_ads u ON a.ads_title = u.ads_title;</pre>
Query Result	

advertiser	user_name
EcoMotors	alice_b
EcoMotors	charlie123
EcoMotors	jane_smith
EcoMotors	bob_the_builder
EcoMotors	emily.r
TechWorld	alice_b
TechWorld	charlie123
TechWorld	jane_smith
TechWorld	bob_the_builder
TechWorld	emily.r
Foodies	alice_b
Foodies	charlie123
Foodies	jane_smith
Foodies	bob_the_builder
Foodies	emily.r
GoGlobalTrips	alice_b
GoGlobalTrips	charlie123
GoGlobalTrips	jane_smith
GoGlobalTrips	bob_the_builder
GoGlobalTrips	emily.r
FilmStudio	alice_b
FilmStudio	charlie123
FilmStudio	jane_smith
FilmStudio	bob_the_builder
FilmStudio	emily.r

5.2.14 Find Drinkware Items Ordered and Their Sizes

Report	Find Drinkware Items Ordered and Their Sizes
Objective	Extract detailed information on ordered drinkware items for inventory or sales analysis.
Query	
<pre>SELECT mi.item_name, d.capacity, d.upper_diameter, d.bottom_diameter FROM merchandise_item mi JOIN drinkware d ON mi.item_id = d.item_id JOIN order_contains_item oci ON mi.item_id = oci.item_id;</pre>	
Query Result	

item_name	capacity	upper_diameter	bottom_diameter
Stranger Things Mug	350	3	3
Stranger Things Mug	350	3	3
Stranger Things Mug	350	3	3
Stranger Things Mug	350	3	3
Bridgerton Wine Glass	450	3	2
Bridgerton Wine Glass	450	3	2
Bridgerton Wine Glass	450	3	2
Bridgerton Wine Glass	450	3	2
The Witcher Tumbler	500	3	3
The Witcher Tumbler	500	3	3
The Witcher Tumbler	500	3	3
Squid Game Glass Bottle	750	3	2
Squid Game Glass Bottle	750	3	2
Squid Game Glass Bottle	750	3	2
Squid Game Glass Bottle	750	3	2

5.2.15 List All Directors for Content in the 'Action' Genre

Report	List All Directors for Content in the 'Action' Genre								
Objective	Identify directors associated with a particular genre for genre-specific promotions or trends analysis.								
Query									
SELECT cd.director, g.genre_name FROM content_director cd JOIN genre_contain_content gc ON cd.content_id = gc.content_id JOIN genre g ON gc.genre_id = g.genre_id WHERE g.genre_name = 'Action';									
Query Result									
<table border="1"> <thead> <tr> <th>director</th><th>genre_name</th></tr> </thead> <tbody> <tr> <td>Rawson Marshall Thurber</td><td>Action</td></tr> <tr> <td>Gina Prince-Bythewood</td><td>Action</td></tr> <tr> <td>Martin Scorsese</td><td>Action</td></tr> </tbody> </table>		director	genre_name	Rawson Marshall Thurber	Action	Gina Prince-Bythewood	Action	Martin Scorsese	Action
director	genre_name								
Rawson Marshall Thurber	Action								
Gina Prince-Bythewood	Action								
Martin Scorsese	Action								

5.2.16 Display Users Who Subscribed to Specific Plans

Report	Display Users Who Subscribed to Specific Plans																																																																														
Objective	Link user accounts to their subscription details to understand plan distribution and revenue streams.																																																																														
Query																																																																															
<pre>SELECT u.username, s.plan_name, p.monthly_price FROM user_account u JOIN subscription s ON u.username = s.username JOIN plan p ON s.plan_name = p.plan_name;</pre>																																																																															
Query Result																																																																															
<table border="1"> <thead> <tr> <th>username</th> <th>plan_name</th> <th>monthly_price</th> </tr> </thead> <tbody> <tr><td>alice_b</td><td>BASIC</td><td>9.99</td></tr> <tr><td>charlie123</td><td>BASIC</td><td>9.99</td></tr> <tr><td>emily.r</td><td>BASIC</td><td>9.99</td></tr> <tr><td>frank88</td><td>BASIC</td><td>9.99</td></tr> <tr><td>harry_p</td><td>BASIC</td><td>9.99</td></tr> <tr><td>alice_b</td><td>BASIC</td><td>9.99</td></tr> <tr><td>david_k</td><td>BASIC</td><td>9.99</td></tr> <tr><td>alice_b</td><td>BASICADS</td><td>6.99</td></tr> <tr><td>charlie123</td><td>BASICADS</td><td>6.99</td></tr> <tr><td>jane_smith</td><td>BASICADS</td><td>6.99</td></tr> <tr><td>bob_the_builder</td><td>BASICADS</td><td>6.99</td></tr> <tr><td>emily.r</td><td>BASICADS</td><td>6.99</td></tr> <tr><td>bob_the_builder</td><td>MOBILE</td><td>3.99</td></tr> <tr><td>david_k</td><td>MOBILE</td><td>3.99</td></tr> <tr><td>frank88</td><td>MOBILE</td><td>3.99</td></tr> <tr><td>bob_the_builder</td><td>MOBILE</td><td>3.99</td></tr> <tr><td>bob_the_builder</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>david_k</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>gina_l</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>jane_smith</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>john_doe</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>alice_b</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>charlie123</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>david_k</td><td>PREMIUM</td><td>19.99</td></tr> <tr><td>emily.r</td><td>STANDARD</td><td>15.99</td></tr> </tbody> </table>		username	plan_name	monthly_price	alice_b	BASIC	9.99	charlie123	BASIC	9.99	emily.r	BASIC	9.99	frank88	BASIC	9.99	harry_p	BASIC	9.99	alice_b	BASIC	9.99	david_k	BASIC	9.99	alice_b	BASICADS	6.99	charlie123	BASICADS	6.99	jane_smith	BASICADS	6.99	bob_the_builder	BASICADS	6.99	emily.r	BASICADS	6.99	bob_the_builder	MOBILE	3.99	david_k	MOBILE	3.99	frank88	MOBILE	3.99	bob_the_builder	MOBILE	3.99	bob_the_builder	PREMIUM	19.99	david_k	PREMIUM	19.99	gina_l	PREMIUM	19.99	jane_smith	PREMIUM	19.99	john_doe	PREMIUM	19.99	alice_b	PREMIUM	19.99	charlie123	PREMIUM	19.99	david_k	PREMIUM	19.99	emily.r	STANDARD	15.99
username	plan_name	monthly_price																																																																													
alice_b	BASIC	9.99																																																																													
charlie123	BASIC	9.99																																																																													
emily.r	BASIC	9.99																																																																													
frank88	BASIC	9.99																																																																													
harry_p	BASIC	9.99																																																																													
alice_b	BASIC	9.99																																																																													
david_k	BASIC	9.99																																																																													
alice_b	BASICADS	6.99																																																																													
charlie123	BASICADS	6.99																																																																													
jane_smith	BASICADS	6.99																																																																													
bob_the_builder	BASICADS	6.99																																																																													
emily.r	BASICADS	6.99																																																																													
bob_the_builder	MOBILE	3.99																																																																													
david_k	MOBILE	3.99																																																																													
frank88	MOBILE	3.99																																																																													
bob_the_builder	MOBILE	3.99																																																																													
bob_the_builder	PREMIUM	19.99																																																																													
david_k	PREMIUM	19.99																																																																													
gina_l	PREMIUM	19.99																																																																													
jane_smith	PREMIUM	19.99																																																																													
john_doe	PREMIUM	19.99																																																																													
alice_b	PREMIUM	19.99																																																																													
charlie123	PREMIUM	19.99																																																																													
david_k	PREMIUM	19.99																																																																													
emily.r	STANDARD	15.99																																																																													

5.2.17 Advertisers with Most Ads Shown

Report	Advertisers with Most Ads Shown
Objective	Identify advertisers whose ads were shown most frequently to users.
Query	

```

SELECT a.advertiser, COUNT(usa.ads_title) AS ads_shown_count
FROM ads a
JOIN user_shown_ads usa ON a.ads_title = usa.ads_title
GROUP BY a.advertiser
ORDER BY ads_shown_count DESC;

```

Query Result

advertiser	ads_shown_count
FilmStudio	5
TechWorld	5
StreamPlus	5
Foodies	5
GoGlobalTrips	5
EcoMotors	5

5.2.18 Top 5 Rated Merchandise Items

Report	Top 5 Rated Merchandise Items
Objective	Find the top 5 highest-rated merchandise items based on user reviews.
Query	
<pre> SELECT mi.collection_category, SUM(mi.price) AS total_revenue FROM merchandise_item mi JOIN merchandise_order mo ON mi.item_id = mo.order_id WHERE mo.status = 'completed' GROUP BY mi.collection_category ORDER BY total_revenue DESC; </pre>	
Query Result	

collection_category	total_revenue	1
Squid Game Collection	49.99	
Stranger Things Collection	44.98	
Bridgerton Collection	44.98	
The Witcher Collection	9.99	

5.2.19 Revenue from Discounted Merchandise

Report	Revenue from Discounted Merchandise
Objective	Calculate the total revenue generated from merchandise sold at discounted prices.
Query	
SELECT SUM(oi.quantity * mi.discounted_price) AS discounted_revenue FROM order_contains_item oi JOIN merchandise_item mi ON oi.item_id = mi.item_id WHERE mi.discounted_price IS NOT NULL;	
Query Result	
discounted_revenue 557.78	

5.2.20 Total View Count of Each Genre

Report	Total View Count of Each Genre
Objective	List the genres and the total view count under each one.
Query	
SELECT gc.genre_name, COUNT(uvc.content_id) AS view_count FROM user_view_content uvc JOIN genre_contain_content gcc ON uvc.content_id = gcc.content_id JOIN genre gc ON gcc.genre_id = gc.genre_id GROUP BY gc.genre_name ORDER BY view_count DESC;	
Query Result	

genre_name	view_count	1
Drama	24	
Thriller	8	
Crime	7	
Adventure	7	
Comedy	4	
Action	4	
Mystery	3	
Romance	3	
Fantasy	3	
Historical	2	
Western	2	
Horror	1	

5.2.21 Active Subscriptions by Plan

Report	Active Subscriptions by Plan
Objective	Count the number of active subscriptions for each plan and the total revenue from those subscriptions.
Query	
<pre> SELECT p.plan_name, COUNT(s.subscription_id) AS active_subscriptions, SUM(p.monthly_price) AS total_monthly_revenue FROM plan p JOIN subscription s ON p.plan_name = s.plan_name WHERE s.termination_date IS NULL OR s.termination_date > CURRENT_DATE() GROUP BY p.plan_name ORDER BY </pre>	

active_subscriptions DESC;																				
Query Result																				
<table border="1"> <thead> <tr> <th>plan_name</th> <th>active_subscriptions</th> <th>▼ 1</th> <th>total_monthly_revenue</th> </tr> </thead> <tbody> <tr> <td>PREMIUM</td> <td>7</td> <td></td> <td>139.93</td> </tr> <tr> <td>BASICADS</td> <td>5</td> <td></td> <td>34.95</td> </tr> <tr> <td>STANDARD</td> <td>2</td> <td></td> <td>31.98</td> </tr> <tr> <td>MOBILE</td> <td>1</td> <td></td> <td>3.99</td> </tr> </tbody> </table>	plan_name	active_subscriptions	▼ 1	total_monthly_revenue	PREMIUM	7		139.93	BASICADS	5		34.95	STANDARD	2		31.98	MOBILE	1		3.99
plan_name	active_subscriptions	▼ 1	total_monthly_revenue																	
PREMIUM	7		139.93																	
BASICADS	5		34.95																	
STANDARD	2		31.98																	
MOBILE	1		3.99																	

5.2.22 Find the most-watched maturity rating category.

Report	Find the most-watched maturity rating category.				
Objective	To determine the most-watched maturity rating category based on user viewing data				
Query					
<pre>SELECT c.maturity_rating, COUNT(uvc.viewcontent_id) AS total_views FROM content AS c JOIN user_view_content AS uvc ON c.content_id = uvc.content_id GROUP BY c.maturity_rating ORDER BY total_views DESC LIMIT 1;</pre>					
Query Result					
<table border="1"> <thead> <tr> <th>maturity_rating</th> <th>total_views</th> </tr> </thead> <tbody> <tr> <td>TV-MA</td> <td>13</td> </tr> </tbody> </table>		maturity_rating	total_views	TV-MA	13
maturity_rating	total_views				
TV-MA	13				

5.2.23 Find the total number of views for each piece of content.

Report	Find the total number of views for each piece of content.
Objective	To identify the total number of views for each piece of content, providing insights into content performance and popularity.
Query	
<pre>SELECT c.content_title, COUNT(uvc.viewcontent_id) AS total_views FROM content AS c LEFT JOIN user_view_content AS uvc ON c.content_id = uvc.content_id GROUP BY c.content_title ORDER BY total_views DESC;</pre>	

Query Result

content_title	total_views
One Piece	2
The Harder They Fall	2
Enola Holmes	2
The Lincoln Lawyer	2
Squid Game	2
The Old Guard	2
Money Heist	2
Bird Box	2
The Crown	2
Don't Look Up	2
Stranger Things	1
Red Notice	1
The Queen's Gambit	1
The Killer	1
The Irishman	1
A Killer Paradox	1
The Witcher	1
Marriage Story	1
Bridgerton	1
The Power of the Dog	1

5.2.24 The number of cast members on each content title.

Report	The number of cast members on each content title.
Objective	To determine the number of cast members associated with each content title.
Query	
<pre>SELECT c.content_title, COUNT(uvc.viewcontent_id) AS total_views FROM content AS c LEFT JOIN user_view_content AS uvc ON c.content_id = uvc.content_id GROUP BY c.content_title ORDER BY total_views DESC;</pre>	
Query Result	

content_title	total_cast
Marriage Story	3
The Crown	3
Bird Box	3
Stranger Things	3
The Power of the Dog	3
One Piece	3
Don't Look Up	3
The Queen's Gambit	3
Red Notice	3
Squid Game	3
Enola Holmes	3
The Witcher	3
The Irishman	3
Money Heist	3
The Old Guard	3
Bridgerton	3
The Lincoln Lawyer	2
The Harder They Fall	2
A Killer Paradox	2
The Killer	2