Melisa Uyar 200315037

Muhammet Berk Can 200315032

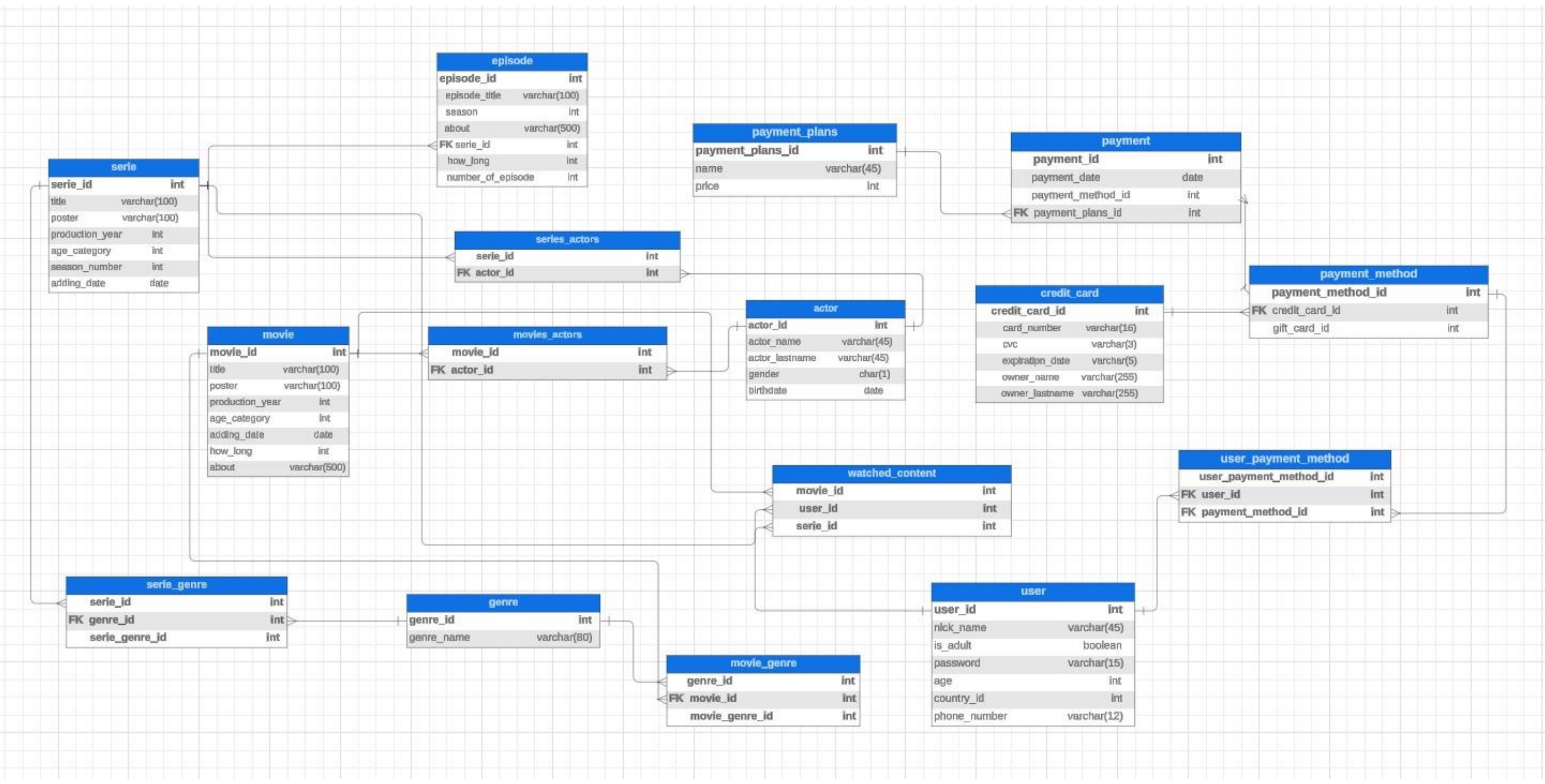
Oğuz Anıl Ateş 200315072

NETFLIX

The scenario involves the development of a streaming platform similar to Netflix. This platform allows users to watch movies and series, manage their watched content, and explore a diverse library of films and TV shows. Users can create accounts, customize their profiles, and choose from various payment methods, including credit cards and gift cards.

This Netflix-like application provides a seamless streaming experience, combining user-friendly profiles, diverse content, and secure payment options.

E-R DİAGRAM:



Entities

user:

user_id (PK): User identification number

nick_name: User's nickname

is_adult: Boolean value indicating whether the user is an adult

password: User's password

age: User's age

phone_number: User's phone number

credit_card:

credit_card_id (PK): Credit card identification number

user_id (FK): User identification number associated with the credit card

Card_Number: Credit card number

cvc: Credit card security code

expiration_date: Expiry date of the credit card

owner_name: Name of the credit card owner

owner_lastname: Last name of the credit card owner

user_payment_method:

user_payment_method_id (PK): User payment method identification number

user_id (FK): User identification number

payment_method_id (FK): Identification number of the payment method the user can use

payment_method:

payment_method_id (PK): Payment method identification number

gift_card_id (FK): Identification number of the gift card

credit_card_id (FK): Identification number of the credit card

watched_content:

watched_id (PK): Identification number of the watched content user_id (FK): User identification number of the viewer

content_id (FK): Identification number of the watched content (Movie or Series)

genre:

genre_id (PK): Genre identification number

genre_name: Name of the genre (Drama, Comedy, Science Fiction, etc.)

movie:

movie_id (PK): Movie identification number

title: Movie title

poster: Movie poster

production_year: Year the movie was produced

age_category: Age category of the movie (e.g., 13+, 18+)

adding_date: Date when the movie was added to the platform

how_long: Duration of the movie

about: Information about the movie

series:

series_id (PK): Series identification number

title: Series title

poster: Series poster

production_year: Year the series was produced

age_category: Age category of the series (e.g., 13+, 18+)

season_number: Series season number

adding_date: Date when the series was added to the platform

episode_entity: Within the TV series (represented by the series entity), episodes are individually tracked using the episode entity.

Serie_genre:

serie_genre_id (PK): Series genre identification number

series_id (FK): Series identification number

genre_id (FK): Genre identification number

movie _genre:

movie_genre_id (PK): Movie genre identification number

movie_id (FK): Movie identification number

genre_id (FK): Genre identification number

series_actor:

series_actor_id (PK): Series actor identification number

series_id (FK): Series identification number

actor_id (FK): Actor identification number

movies_actor:

movies_actor_id (PK): Movie actor identification number

movie_id (FK): Movie identification number

actor_id (FK): Actor identification number

payment:

payment_method_id (PK): Payment method identification number

credit_card_id (FK): Identification number of the credit card

gift_card_id (FK): Identification number of the gift card

Functionality of system :

Users can create and customize their profiles with personal details.

They can manage their payment methods, including credit cards and gift cards.

Users can browse and stream a vast library of movies and series.

Movies and series are categorized into genres, enhancing content discovery.

Actors are associated with specific movies and TV series.

Users can make payments using various methods, and transactions are recorded.

10 Important Queries:

List the Movies and Series Watched by the User:

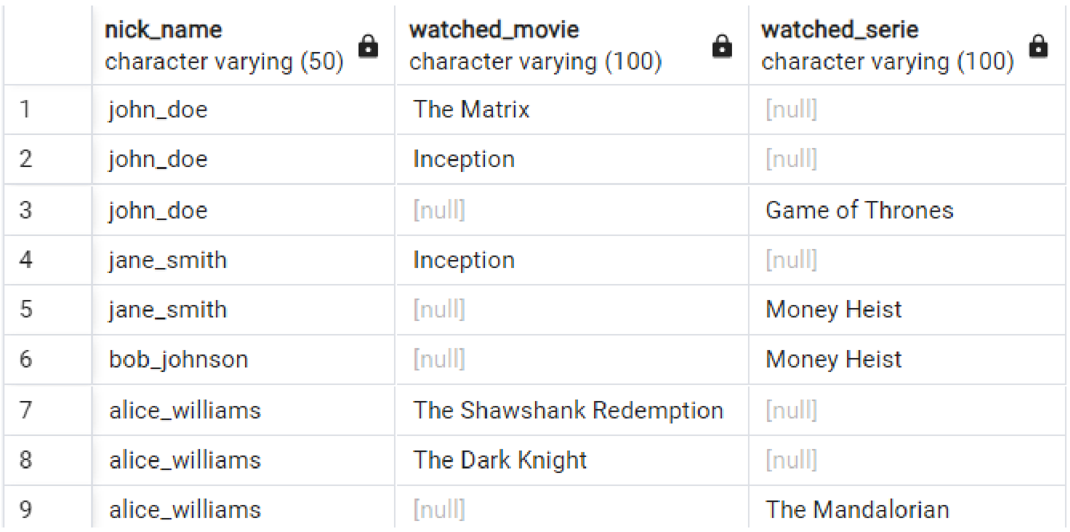
SELECT public.user.nick_name, movie.title AS watched_movie, serie.title AS watched_serie

FROM public.user

LEFT JOIN watched_content ON public.user.user_id = watched_content.user_id

LEFT JOIN movie ON watched_content.movie_id = movie.movie_id

LEFT JOIN serie ON watched_content.serie_id = serie.serie_id;



	nick_name character varying (50)	watched_movie character varying (100)	watched_serie character varying (100)
1	john_doe	The Matrix	[null]
2	john_doe	Inception	[null]
3	john_doe	[null]	Game of Thrones
4	jane_smith	Inception	[null]
5	jane_smith	[null]	Money Heist
6	bob_johnson	[null]	Money Heist
7	alice_williams	The Shawshank Redemption	[null]
8	alice_williams	The Dark Knight	[null]
9	alice_williams	[null]	The Mandalorian

2) List the User's Credit Card Information and Used Payment Methods:

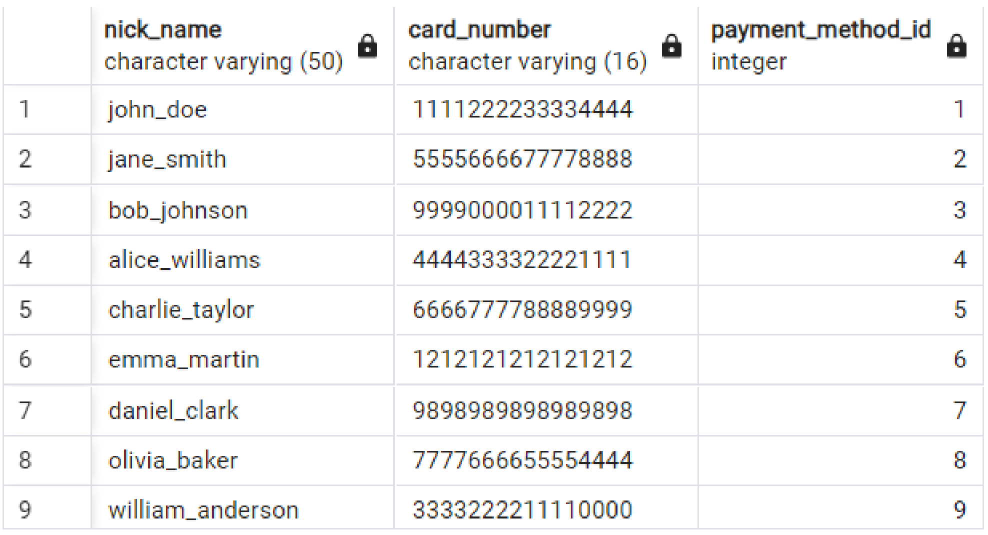
SELECT public.user.nick_name, credit_card.card_number, payment_method.payment_method_id

FROM public.user

JOIN user_payment_method ON public.user.user_id = user_payment_method.user_id

JOIN payment_method ON user_payment_method.payment_method_id = payment_method.payment_method_id

JOIN credit_card ON payment_method.credit_card_id = credit_card.credit_card_id;

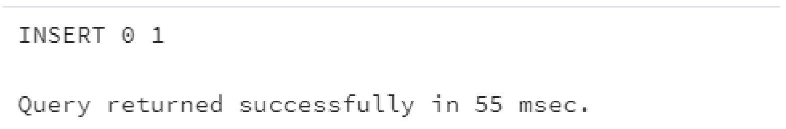


	nick_name character varying (50)	card_number character varying (16)	payment_method_id integer
1	john_doe	1111222233334444	1
2	jane_smith	5555666677778888	2
3	bob_johnson	9999000011112222	3
4	alice_williams	4444333322221111	4
5	charlie_taylor	6666777788889999	5
6	emma_martin	1212121212121212	6
7	daniel_clark	9898989898989898	7
8	olivia_baker	7777666655554444	8
9	william_anderson	3333222211110000	9

3)Data Entry for the User:

INSERT INTO "user" (user_id, nick_name, is_adult, password, age, country_id, phone_number)

VALUES (1, 'john_doe', true, 'password123', 30, 1, '+1234567890');



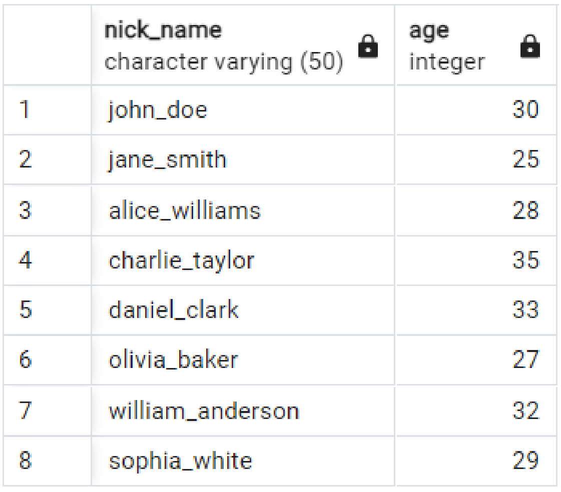
```
INSERT 0 1

Query returned successfully in 55 msec.
```

4) List Users Within a Certain Age Range:

SELECT nick_name, age

FROM public.user

WHERE age BETWEEN 25 AND 35;



	nick_name character varying (50)	age integer
1	john_doe	30
2	jane_smith	25
3	alice_williams	28
4	charlie_taylor	35
5	daniel_clark	33
6	olivia_baker	27
7	william_anderson	32
8	sophia_white	29

5) List Payments Within a Specific Date Range:

SELECT public.user.nick_name, payment.payment_date, payment_plans.name, payment_plans.price

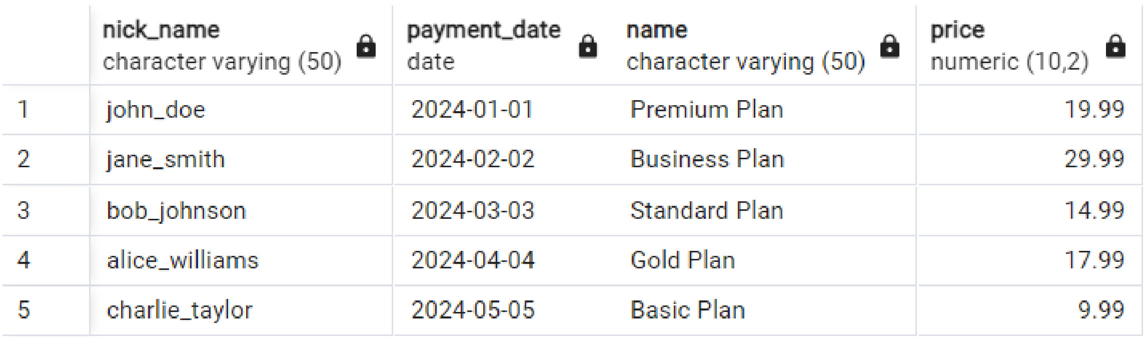
FROM public.user

JOIN payment ON public.user.user_id = payment.payment_method_id

JOIN payment_plans ON payment.payment_plans_id = payment_plans.payment_plans_id

WHERE payment.payment_date BETWEEN '2024-01-01' AND '2024-06-01'

ORDER BY payment.payment_date;



	nick_name character varying (50)	payment_date date	name character varying (50)	price numeric (10,2)
1	john_doe	2024-01-01	Premium Plan	19.99
2	jane_smith	2024-02-02	Business Plan	29.99
3	bob_johnson	2024-03-03	Standard Plan	14.99
4	alice_williams	2024-04-04	Gold Plan	17.99
5	charlie_taylor	2024-05-05	Basic Plan	9.99

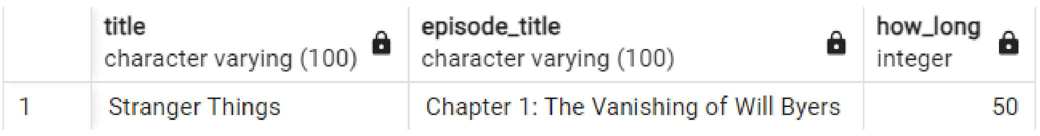
6) List Episodes of a Series and Calculate Durations:

SELECT serie.title, episode.episode_title, episode.how_long

FROM serie

JOIN episode ON serie.serie_id = episode.serieid

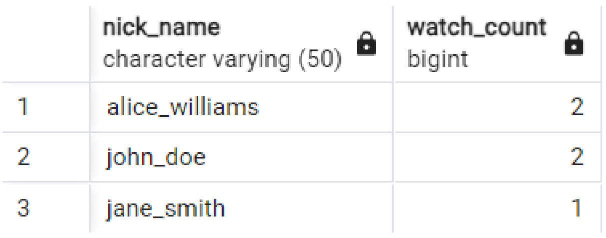
WHERE serie.title = 'Stranger Things';



	title character varying (100)	episode_title character varying (100)	how_long integer
1	Stranger Things	Chapter 1: The Vanishing of Will Byers	50

7) List the Top 3 Users Who Watched the Most Films and Their Watch Counts:

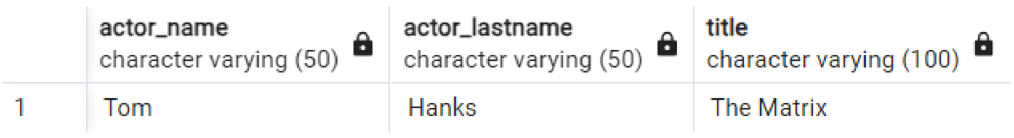
```
SELECT public.user.nick_name, COUNT(watched_content.movie_id) AS watch_count
FROM public.user
LEFT JOIN watched_content ON public.user.user_id = watched_content.user_id
GROUP BY public.user.nick_name
ORDER BY watch_count DESC
LIMIT 3;
```



	nick_name character varying (50)	watch_count bigint
1	alice_williams	2
2	john_doe	2
3	jane_smith	1

8) List Films Played by a Specific Actor:

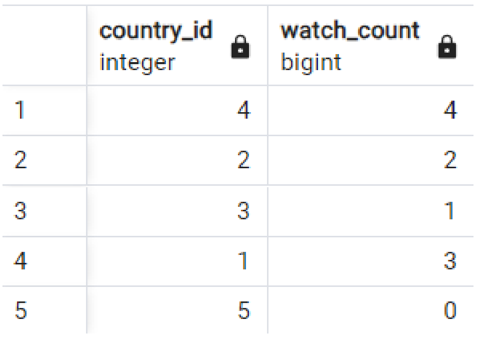
```
SELECT actor.actor_name, actor.actor_lastname, movie.title
FROM actor
JOIN movies_actors ON actor.actor_id = movies_actors.actor_id
JOIN movie ON movies_actors.movie_id = movie.movie_id
WHERE actor.actor_name = 'Tom' AND actor.actor_lastname = 'Hanks';
```



	actor_name character varying (50)	actor_lastname character varying (50)	title character varying (100)
1	Tom	Hanks	The Matrix

9) List Users Living in a Specific Country and Their Total Watch Counts:

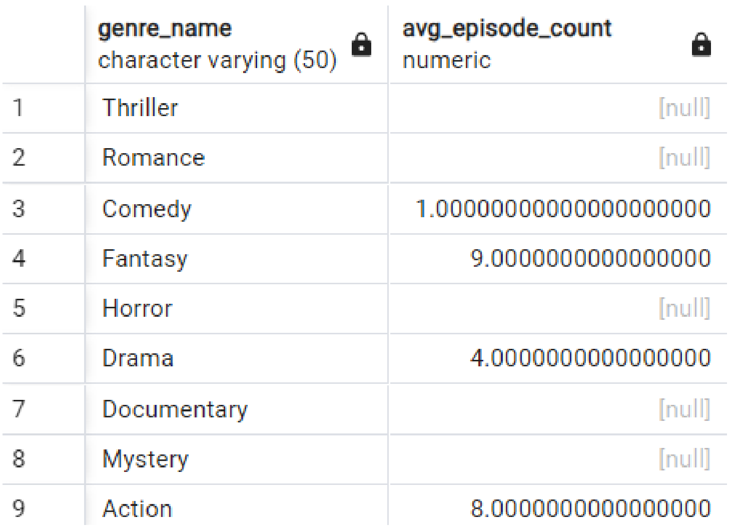
```
SELECT public.user.country_id, COUNT(watched_content.user_id) AS watch_count

FROM public.user

LEFT JOIN watched_content ON public.user.user_id = watched_content.user_id

GROUP BY public.user.country_id;
```



	country_id integer	watch_count bigint
1	4	4
2	2	2
3	3	1
4	1	3
5	5	0

10) List the Average Episode Counts of Series in a Specific Genre:

```
SELECT genre.genre_name, AVG(episode.number_of_episode) AS avg_episode_count

FROM genre

LEFT JOIN serie_genre ON genre.genre_id = serie_genre.genre_id

LEFT JOIN episode ON serie_genre.serie_id = episode.serieid

GROUP BY genre.genre_name;
```



	genre_name character varying (50)	avg_episode_count numeric
1	Thriller	[null]
2	Romance	[null]
3	Comedy	1.00000000000000000000
4	Fantasy	9.0000000000000000
5	Horror	[null]
6	Drama	4.0000000000000000
7	Documentary	[null]
8	Mystery	[null]
9	Action	8.0000000000000000