MANİSA CELAL BAYAR UNIVERSITY

DATABASE MANAGEMENT SYSTEMS PROJECT

COURSE PROJECT:

Youtube Demo

Furkan YAMAK-200316027

Furkan BULUT-210311561

Mine ALTUĞ -200316021

İçindekiler

1- Introduction

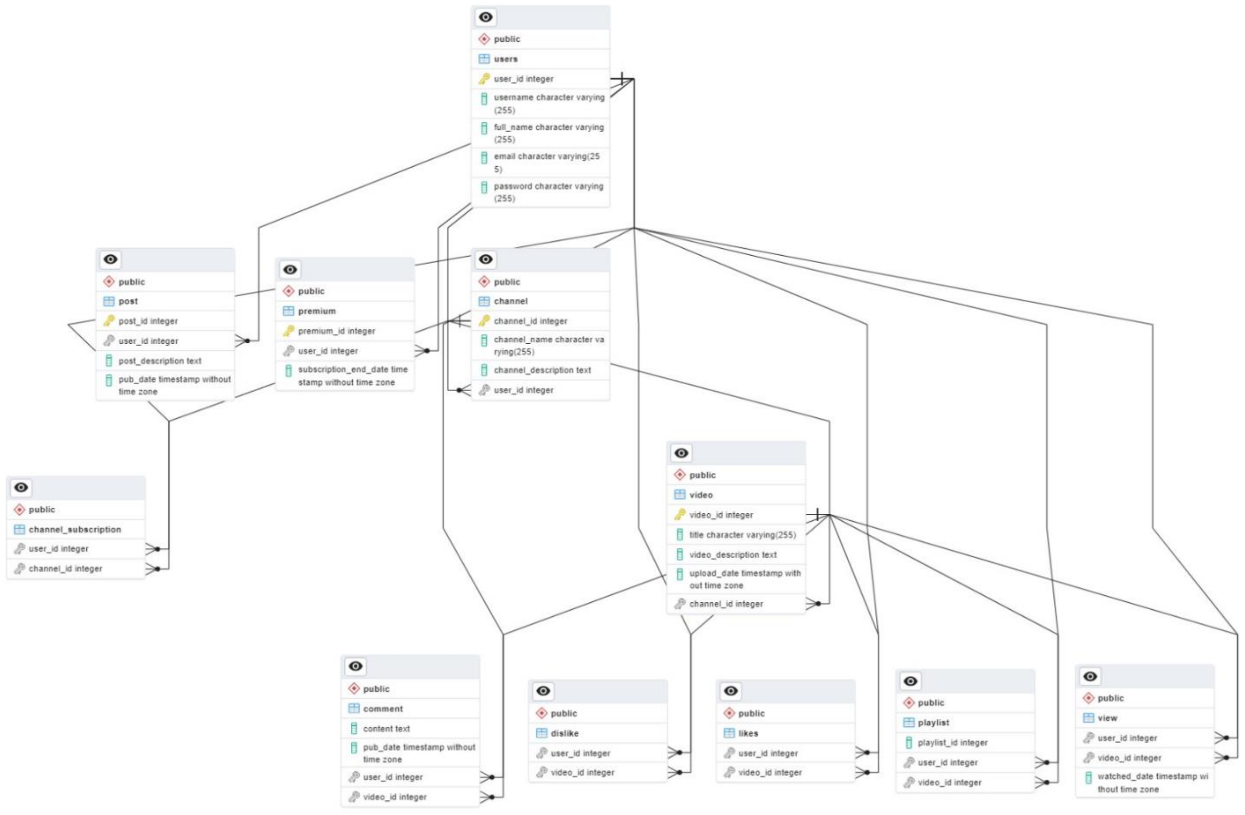
In this project, we were asked to choose one of the applications given to us and design a database. We chose the app called YouTube for ourselves.

This database helps create a video-sharing platform. Users can sign up, make channels, share, and view videos, interact with likes and comments, handle playlists, subscribe to channels, and use special features.

First, we created the tables we need. Thus, we normalized our database as much as we could. First, we express the table and relations of the design you developed on an E-R diagram and create our tables on PostgreSQL in accordance with this table. Then, we made 10 SQL queries, taking into account the relationships in the tables, and then added them to the document. Likewise, the E-R diagram is available in the following sections.

1.1 E-R Diagram

The relationships of the tables are indicated in the E-R diagram.



1.2 Primary Key-Foreign Key Table

In the table below, primary key, foreign key and unique values in the database are given.

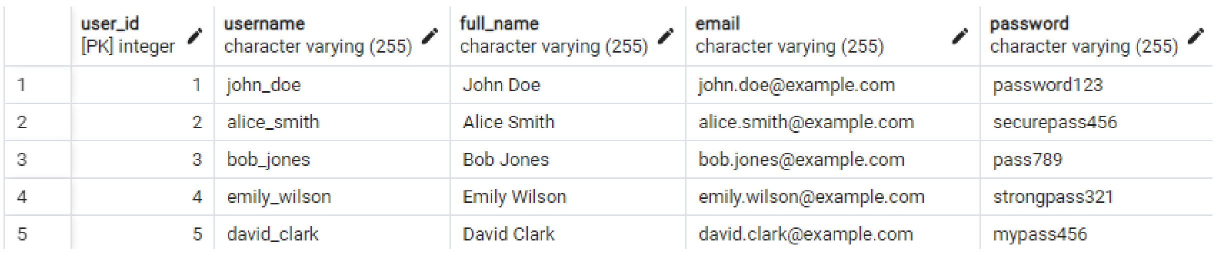
	Primary Key	**Foreign Key**
users	user_id	
video	video_id	channel_id
playlist	channel_id	user_id , video_id
post	post_id	user_id
premium	premium_id	user_id
likes	like_id	user_id , video_id
dislike		user_id , video_id
view		user_id , video_id
comment		user_id ,video_id
channel	channel_id	user_id
channel_subscription		channel_id, user_id

2- Tables and Values

2.1- Create users Table

We created channel table use user_id for primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE users (
    user_id int NOT NULL PRIMARY KEY,
    username varchar(255) NOT NULL,
    full_name varchar(255) NOT NULL,
    email varchar(255) NOT NULL,
    password varchar(255) NOT NULL
);
```

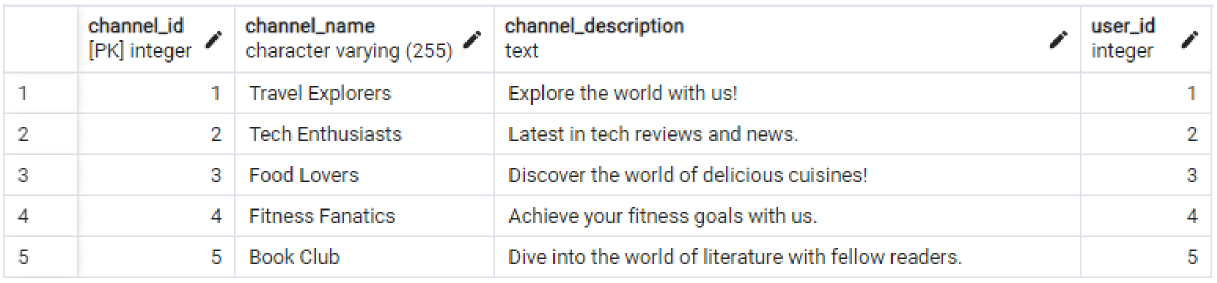


	user_id [PK] integer	username character varying (255)	full_name character varying (255)	email character varying (255)	password character varying (255)
1	1	john_doe	John Doe	john.doe@example.com	password123
2	2	alice_smith	Alice Smith	alice.smith@example.com	securepass456
3	3	bob_jones	Bob Jones	bob.jones@example.com	pass789
4	4	emily_wilson	Emily Wilson	emily.wilson@example.com	strongpass321
5	5	david_clark	David Clark	david.clark@example.com	mypass456

2.2- Create channel Table

We created channel table use channel_id for primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE channel (
    channel_id int NOT NULL PRIMARY KEY,
    channel_name varchar(255) NOT NULL,
    channel_description text NOT NULL,
    user_id int NOT NULL UNIQUE REFERENCES users(user_id)
);
```

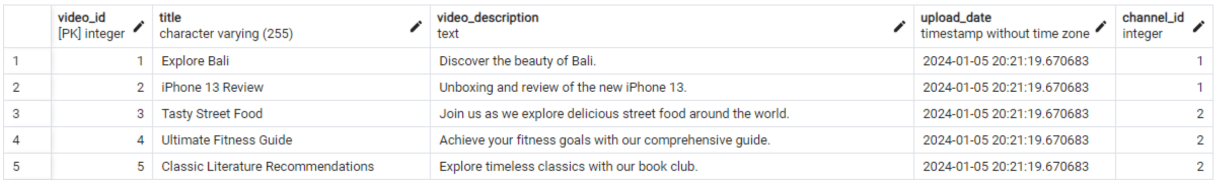


	channel_id [PK] integer	channel_name character varying (255)	channel_description text	user_id integer
1	1	Travel Explorers	Explore the world with us!	1
2	2	Tech Enthusiasts	Latest in tech reviews and news.	2
3	3	Food Lovers	Discover the world of delicious cuisines!	3
4	4	Fitness Fanatics	Achieve your fitness goals with us.	4
5	5	Book Club	Dive into the world of literature with fellow readers.	5

2.3- Create video Table

We created channel table use video_id for primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE video (
    video_id int NOT NULL PRIMARY KEY,
    title varchar(255) NOT NULL,
    video_description text NOT NULL,
    upload_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    channel_id int NOT NULL REFERENCES channel(channel_id)
);
```

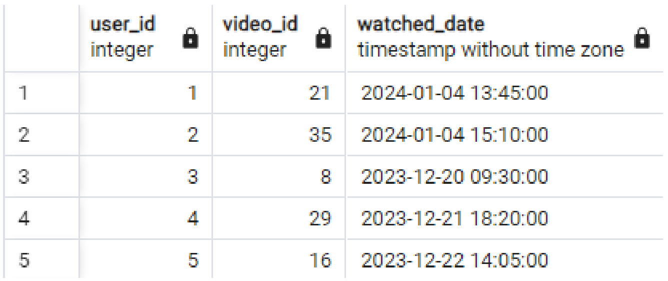


	video_id [PK] integer	title character varying (255)	video_description text	upload_date timestamp without time zone	channel_id integer
1	1	Explore Bali	Discover the beauty of Bali.	2024-01-05 20:21:19.670683	1
2	2	iPhone 13 Review	Unboxing and review of the new iPhone 13.	2024-01-05 20:21:19.670683	1
3	3	Tasty Street Food	Join us as we explore delicious street food around the world.	2024-01-05 20:21:19.670683	2
4	4	Ultimate Fitness Guide	Achieve your fitness goals with our comprehensive guide.	2024-01-05 20:21:19.670683	2
5	5	Classic Literature Recommendations	Explore timeless classics with our book club.	2024-01-05 20:21:19.670683	2

2.4- Create view Table

We created channel table and didn't use any primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE view (
    user_id int NOT NULL REFERENCES users(user_id),
    video_id int NOT NULL REFERENCES video(video_id),
    watched_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```

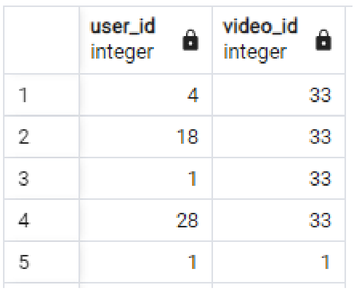


	user_id integer	video_id integer	watched_date timestamp without time zone
1	1	21	2024-01-04 13:45:00
2	2	35	2024-01-04 15:10:00
3	3	8	2023-12-20 09:30:00
4	4	29	2023-12-21 18:20:00
5	5	16	2023-12-22 14:05:00

2.5- Create likes Table

We created channel table and didn't use any primary key. Also, we added a constraint. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE likes (
    user_id int NOT NULL REFERENCES users(user_id),
    video_id int NOT NULL REFERENCES video(video_id),
    CONSTRAINT unique_like_user_video UNIQUE (user_id, video_id)
);
```

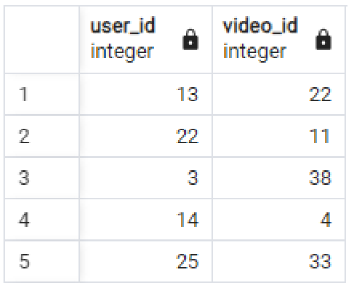


	user_id integer	video_id integer
1	4	33
2	18	33
3	1	33
4	28	33
5	1	1

2.6- Create dislike Table

We created channel table and didn't use any primary key. Also, we added a constraint. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE dislike (
    user_id int NOT NULL REFERENCES users(user_id),
    video_id int NOT NULL REFERENCES video(video_id),
    CONSTRAINT unique_dislike_user_video UNIQUE (user_id, video_id)
);
```

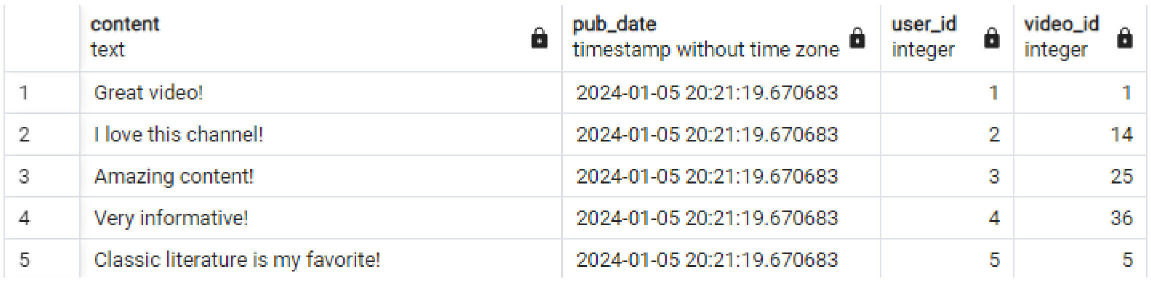


	user_id integer	video_id integer
1	13	22
2	22	11
3	3	38
4	14	4
5	25	33

2.7- Create comment Table

We created channel table and didn't use any primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE comment (
    content text NOT NULL,
    pub_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    user_id int NOT NULL REFERENCES users(user_id),
    video_id int NOT NULL REFERENCES video(video_id)
);
```

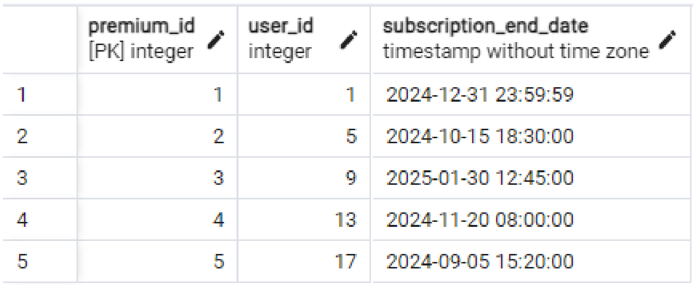


	content text	pub_date timestamp without time zone	user_id integer	video_id integer
1	Great video!	2024-01-05 20:21:19.670683	1	1
2	I love this channel!	2024-01-05 20:21:19.670683	2	14
3	Amazing content!	2024-01-05 20:21:19.670683	3	25
4	Very informative!	2024-01-05 20:21:19.670683	4	36
5	Classic literature is my favorite!	2024-01-05 20:21:19.670683	5	5

2.8- Create premium Table

We created channel table use premium_id for primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE premium (
    premium_id serial NOT NULL PRIMARY KEY,
    user_id int NOT NULL UNIQUE REFERENCES users(user_id),
    subscription_end_date timestamp NOT NULL
);
```

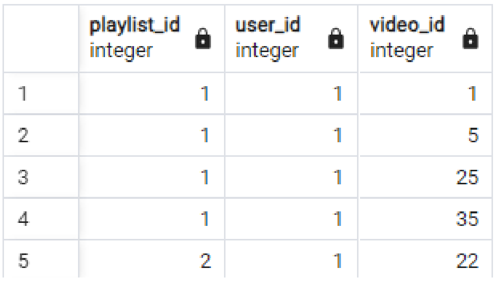


	premium_id [PK] integer	user_id integer	subscription_end_date timestamp without time zone
1	1	1	2024-12-31 23:59:59
2	2	5	2024-10-15 18:30:00
3	3	9	2025-01-30 12:45:00
4	4	13	2024-11-20 08:00:00
5	5	17	2024-09-05 15:20:00

2.9- Create playlist Table

We created channel table and didn't use any primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE playlist (
    playlist_id serial NOT NULL,
    user_id int NOT NULL REFERENCES users(user_id),
    video_id int NOT NULL REFERENCES video(video_id)
);
```

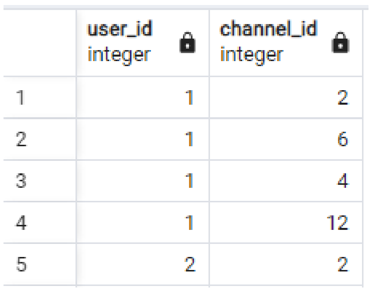


	playlist_id integer	user_id integer	video_id integer
1	1	1	1
2	1	1	5
3	1	1	25
4	1	1	35
5	2	1	22

2.10- Create channel_subscription Table

We created channel table and didn't use any primary key. Also, we added a constraint. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE channel_subscription (
    user_id int NOT NULL REFERENCES users(user_id),
    channel_id int NOT NULL REFERENCES channel(channel_id),
    CONSTRAINT unique_channel_subscription UNIQUE (user_id, channel_id)
);
```

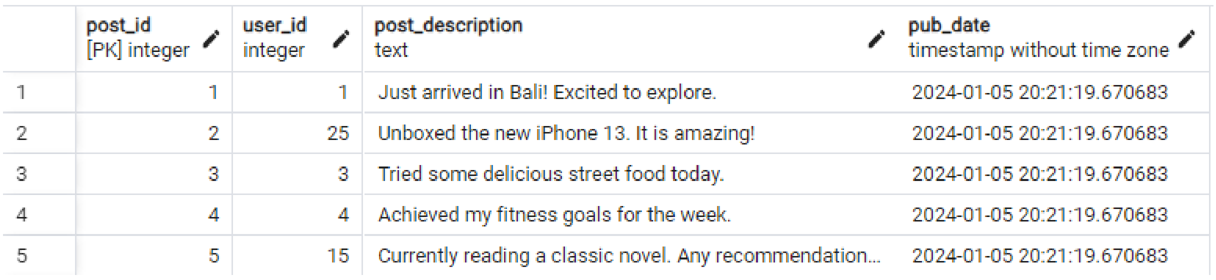


	user_id integer	channel_id integer
1	1	2
2	1	6
3	1	4
4	1	12
5	2	2

2.11- Create post Table

We created channel table use post_id for primary key. Below is the SQL code and 5 sample data of the created table.

```
CREATE TABLE post (
    post_id serial NOT NULL PRIMARY KEY,
    user_id int NOT NULL REFERENCES users(user_id),
    post_description text NOT NULL,
    pub_date timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
);
```



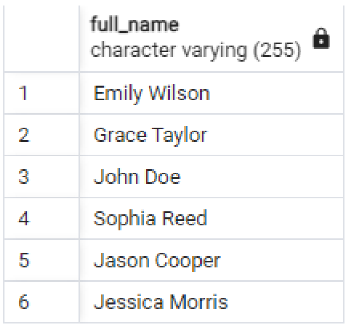
	post_id [PK] integer	user_id integer	post_description text	pub_date timestamp without time zone
1	1	1	Just arrived in Bali! Excited to explore.	2024-01-05 20:21:19.670683
2	2	25	Unboxed the new iPhone 13. It is amazing!	2024-01-05 20:21:19.670683
3	3	3	Tried some delicious street food today.	2024-01-05 20:21:19.670683
4	4	4	Achieved my fitness goals for the week.	2024-01-05 20:21:19.670683
5	5	15	Currently reading a classic novel. Any recommendation...	2024-01-05 20:21:19.670683

3- Queries

3.1- First Query

Shows names of people who liked the 'Artistic Expressions: Mixed Media' video.

```
SELECT full_name
FROM video NATURAL JOIN likes NATURAL JOIN users
WHERE title = 'Artistic Expressions: Mixed Media'
```

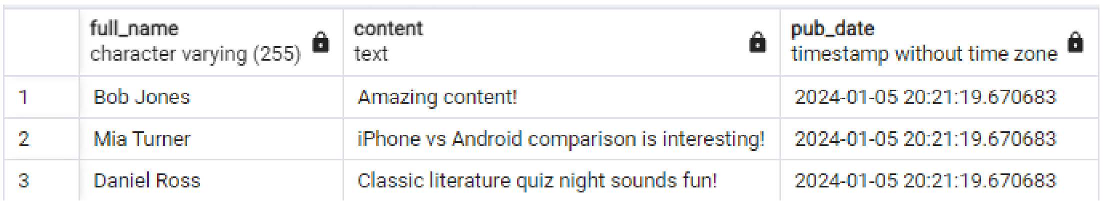


	full_name character varying (255)
1	Emily Wilson
2	Grace Taylor
3	John Doe
4	Sophia Reed
5	Jason Cooper
6	Jessica Morris

3.2- Second Query

Shows comments on the 'iPhone vs Android: The Ultimate Comparison' video, the author of the comments and the date they were published.

```
SELECT full_name , content, pub_date
FROM video NATURAL JOIN comment NATURAL JOIN users
WHERE title = 'iPhone vs Android: The Ultimate Comparison'
```

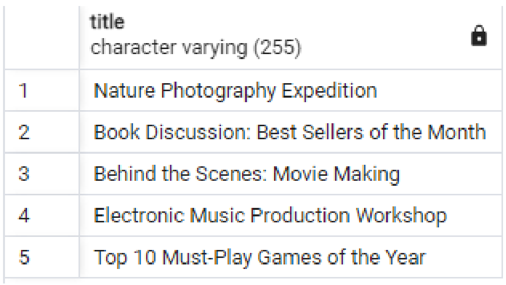


	full_name character varying (255)	content text	pub_date timestamp without time zone
1	Bob Jones	Amazing content!	2024-01-05 20:21:19.670683
2	Mia Turner	iPhone vs Android comparison is interesting!	2024-01-05 20:21:19.670683
3	Daniel Ross	Classic literature quiz night sounds fun!	2024-01-05 20:21:19.670683

3.3- Third Query

Names of videos that the user 'Sarah Martin' has watched in the past.

```
SELECT title
FROM view NATURAL JOIN video NATURAL JOIN users
WHERE full_name = 'Sarah Martin'
```

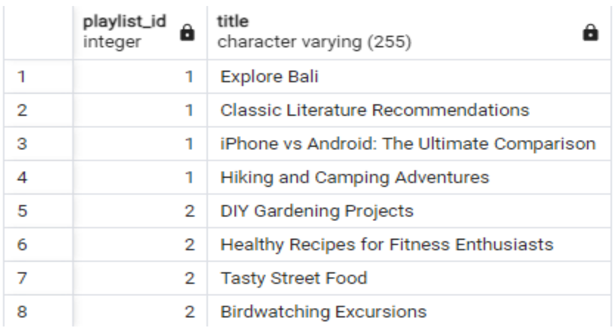


	title character varying (255)
1	Nature Photography Expedition
2	Book Discussion: Best Sellers of the Month
3	Behind the Scenes: Movie Making
4	Electronic Music Production Workshop
5	Top 10 Must-Play Games of the Year

3.4- Fourth Query

Playlist videos of user 'John Doe' and which playlist these videos belong to.

```
SELECT playlist_id, title
FROM playlist NATURAL JOIN video NATURAL JOIN users
WHERE full_name = 'John Doe'
ORDER BY playlist_id ASC
```



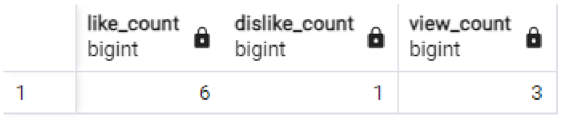
	playlist_id integer	title character varying (255)
1	1	Explore Bali
2	1	Classic Literature Recommendations
3	1	iPhone vs Android: The Ultimate Comparison
4	1	Hiking and Camping Adventures
5	2	DIY Gardening Projects
6	2	Healthy Recipes for Fitness Enthusiasts
7	2	Tasty Street Food
8	2	Birdwatching Excursions

3.5- Fifth Query

Number of likes, dislikes and views on the 'Artistic Expressions: Mixed Media' video.

```
SELECT

    (SELECT COUNT(*) FROM likes
     JOIN video ON likes.video_id = video.video_id
     WHERE title = 'Artistic Expressions: Mixed Media') AS like_count,
    (SELECT COUNT(*) FROM dislike
     JOIN video ON dislike.video_id = video.video_id
     WHERE title = 'Artistic Expressions: Mixed Media') AS dislike_count,
    (SELECT COUNT(*) FROM view
     JOIN video ON view.video_id = video.video_id
     WHERE title = 'Artistic Expressions: Mixed Media') AS view_count;
```

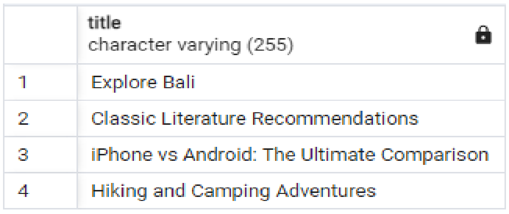


	like_count bigint	dislike_count bigint	view_count bigint
1	6	1	3

3.6- Sixth Query

Show videos from a specific (playlist_id=1) playlist by user 'John Doe'.

```
SELECT title
FROM playlist p JOIN video v ON p.video_id = v.video_id
JOIN users u ON p.user_id = u.user_id
WHERE full_name = 'John Doe' and playlist_id = 1
```

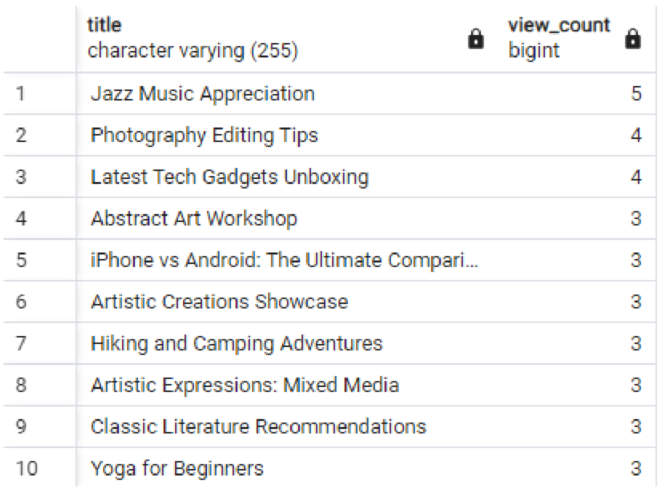


	title character varying (255)
1	Explore Bali
2	Classic Literature Recommendations
3	iPhone vs Android: The Ultimate Comparison
4	Hiking and Camping Adventures

3.7- Seventh Query

Sort the names and views of the videos in descending order

```
SELECT title, COUNT(*) as view_count
FROM view NATURAL JOIN video
GROUP BY video_id, title
ORDER BY COUNT(*) desc
```



	title character varying (255)	view_count bigint
1	Jazz Music Appreciation	5
2	Photography Editing Tips	4
3	Latest Tech Gadgets Unboxing	4
4	Abstract Art Workshop	3
5	iPhone vs Android: The Ultimate Compari...	3
6	Artistic Creations Showcase	3
7	Hiking and Camping Adventures	3
8	Artistic Expressions: Mixed Media	3
9	Classic Literature Recommendations	3
10	Yoga for Beginners	3

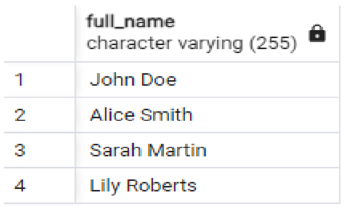
...

Total rows: 40 of 40

3.8- Eighth Query

Subscribers of the 'Tech Enthusiasts' channel.

```
SELECT full_name
FROM users JOIN channel_subscription ON users.user_id =
channel_subscription.user_id
JOIN channel ON channel_subscription.channel_id = channel.channel_id
WHERE channel_name = 'Tech Enthusiasts'
```

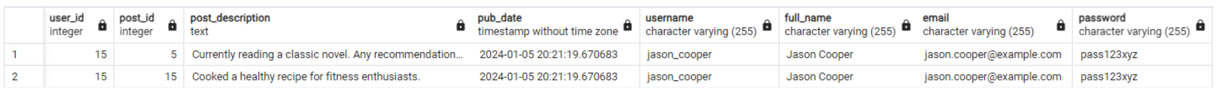


	full_name character varying (255)
1	John Doe
2	Alice Smith
3	Sarah Martin
4	Lily Roberts

3.9- Ninth Query

Posts by user 'Jason Cooper'.

```
SELECT *
FROM post NATURAL JOIN users
WHERE full_name = 'Jason Cooper'
```

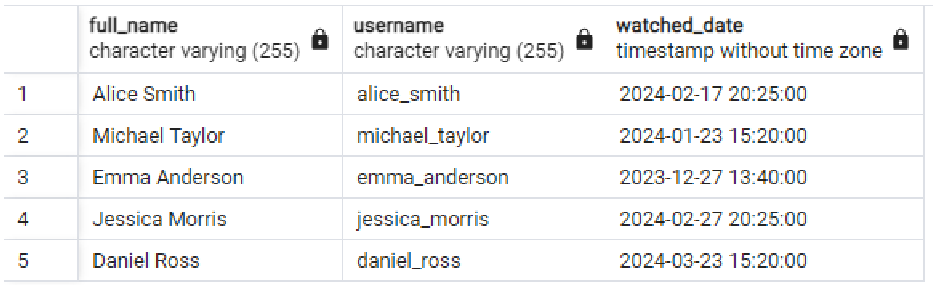


	user_id integer	post_id integer	post_description text	pub_date timestamp without time zone	username character varying (255)	full_name character varying (255)	email character varying (255)	password character varying (255)
1	15	5	Currently reading a classic novel. Any recommendation...	2024-01-05 20:21:19.670683	jason_cooper	Jason Cooper	jason.cooper@example.com	pass123xyz
2	15	15	Cooked a healthy recipe for fitness enthusiasts.	2024-01-05 20:21:19.670683	jason_cooper	Jason Cooper	jason.cooper@example.com	pass123xyz

3.10- Tenth Query

User information who watched the video with video_id 20.

```sql
SELECT full_name,username, watched_date
FROM view NATURAL JOIN users
WHERE video_id = 20
```



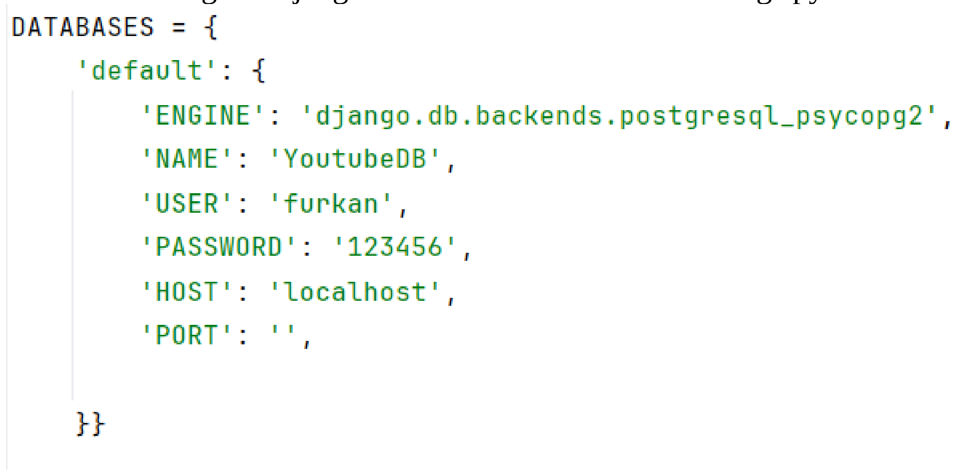
	full_name character varying (255)	username character varying (255)	watched_date timestamp without time zone
1	Alice Smith	alice_smith	2024-02-17 20:25:00
2	Michael Taylor	michael_taylor	2024-01-23 15:20:00
3	Emma Anderson	emma_anderson	2023-12-27 13:40:00
4	Jessica Morris	jessica_morris	2024-02-27 20:25:00
5	Daniel Ross	daniel_ross	2024-03-23 15:20:00

4 - Connection With Django Project

Database integration with a project is crucial and necessary for efficient data management and seamless functionality. Databases serve as structured repositories where relevant information is stored, organized, and retrieved as needed. By establishing a connection between a project and a database, developers can ensure that data is accurately captured, updated, and accessed in real-time, enhancing the overall performance and reliability of the system. This integration enables better decision-making, scalability, and adaptability as the project evolves, making it an essential component for a robust and dynamic application.

4.1 Connection String

We made this project concrete by connecting the database with Django. Here is connection string for Django Database Connection in settings.py file.



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'YoutubeDB',
        'USER': 'furkan',
        'PASSWORD': '123456',
        'HOST': 'localhost',
        'PORT': '',

    }}
```

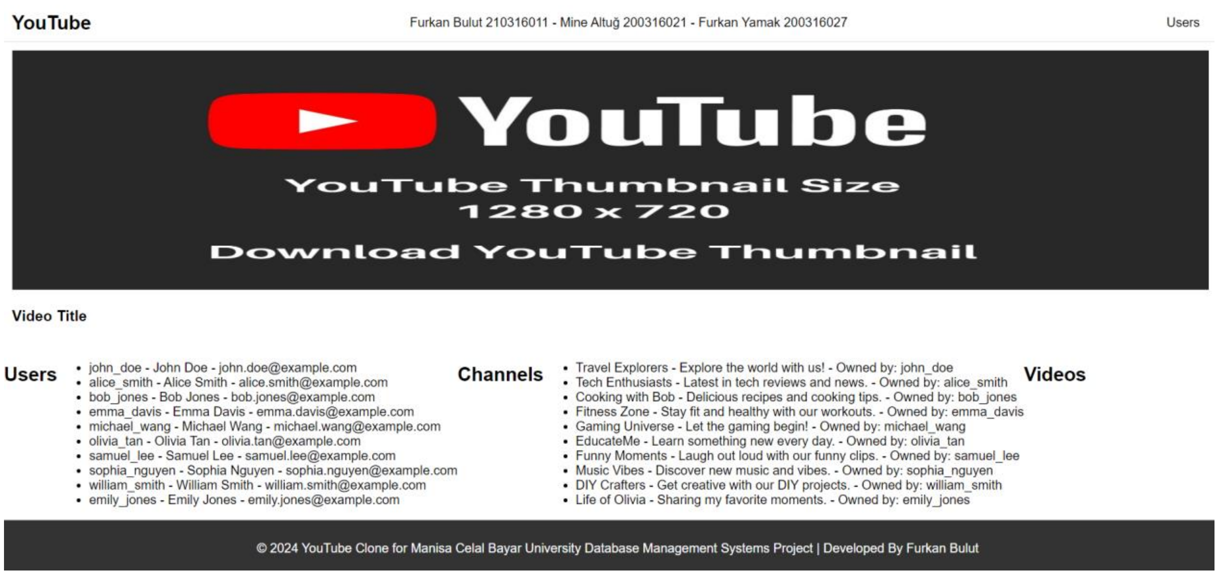
In this Django project, a database configuration has been specified to use a PostgreSQL database. The key-value pairs used in this configuration are as follows:

- **ENGINE**: The engine to be used in the background for the database. In this case, **django.db.backends.postgresql_psycopg2** is a driver recommended by Django for PostgreSQL databases.
- **NAME**: The name of the database. In this example, it is named "YoutubeDB".
- **USER**: The username used for accessing the database. In this example, it is set to "furkan".
- **PASSWORD**: The password used for accessing the database. In this example, it is set to "123456".
- **HOST**: The name or IP address of the computer where the database is located. In this case, "localhost" is used, indicating that the database is running on the local machine.
- **PORT**: The port number used to connect to the database. It is left blank in this example because the default PostgreSQL port will be used.

This configuration is defined in the Django project settings in the **DATABASES** variable. Since this configuration is defined there, the Django project uses this database connection to perform tasks such as creating models, executing queries, and handling database operations.

4.2 Demo

We successfully implemented the process of displaying data on a simple HTML page for testing purposes. Here is a screenshot depicting this accomplishment.



Furkan Bulut 210316011 - Mine Altuğ 200316021 - Furkan Yamak 200316027

Video Title

Users

- john_doe - John Doe - john.doe@example.com
- alice_smith - Alice Smith - alice.smith@example.com
- bob_jones - Bob Jones - bob.jones@example.com
- emma_davis - Emma Davis - emma.davis@example.com
- michael_wang - Michael Wang - michael.wang@example.com
- olivia_tan - Olivia Tan - olivia.tan@example.com
- samuel_lee - Samuel Lee - samuel.lee@example.com
- sophia_nguyen - Sophia Nguyen - sophia.nguyen@example.com
- william_smith - William Smith - william.smith@example.com
- emily_jones - Emily Jones - emily.jones@example.com

Channels

- Travel Explorers - Explore the world with us! - Owned by: john_doe
- Tech Enthusiasts - Latest in tech reviews and news. - Owned by: alice_smith
- Cooking with Bob - Delicious recipes and cooking tips. - Owned by: bob_jones
- Fitness Zone - Stay fit and healthy with our workouts. - Owned by: emma_davis
- Gaming Universe - Let the gaming begin! - Owned by: michael_wang
- EducateMe - Learn something new every day. - Owned by: olivia_tan
- Funny Moments - Laugh out loud with our funny clips. - Owned by: samuel_lee
- Music Vibes - Discover new music and vibes. - Owned by: sophia_nguyen
- DIY Crafters - Get creative with our DIY projects. - Owned by: william_smith
- Life of Olivia - Sharing my favorite moments. - Owned by: emily_jones

Videos

© 2024 YouTube Clone for Manisa Celal Bayar University Database Management Systems Project | Developed By Furkan Bulut

To run a Django project, follow these simple steps:

1. **Navigate to Project Directory:** Open your command prompt or terminal and navigate to the directory where your Django project is located using the **cd** command. For example:

```
cd path/to/your/django/project
```

2. **Activate Virtual Environment (Optional but Recommended):** If you are using a virtual environment (which is recommended), activate it. For example:

```
source venv/bin/activate    # For Linux or macOS
venv\Scripts\activate       # For Windows
```

After activating your virtual environment installation of requirements.txt is required. Make sure you are in the correct directory where your **requirements.txt** file is located. This command is used to install all the dependencies listed in the **requirements.txt** file for a Python project. If you have a different file name or path, adjust the command accordingly.

```
pip install -r requirements.txt
```

3. **Run the Development Server:** Start the Django development server:

```
python manage.py runserver
```

4. **Access the Project:** Open your web browser and go to **http://127.0.0.1:8000/** or **http://localhost:8000/**. You should see your Django project running.

To see this project on a github repository here are the fallowing link:
https://github.com/Furk4nBulut/Youtube