

Documentation de la structure de tvideos.db

les tables

Table `videos`

Cette table contient des informations sur les vidéos. Elle possède les champs suivants :

- `id` : un entier qui représente l'identifiant unique de la vidéo, et qui est utilisé comme clé primaire. Il est également défini comme étant auto-incrémenté.
- `titre` : un champ de type texte qui contient le titre de la vidéo.
- `lien` : un champ de type texte qui contient le lien vers la vidéo.
- `description` : un champ de type texte qui contient la description de la vidéo.
- `like` : un champ de type entier qui contient le nombre de "likes" de la vidéo.
- `dislike` : un champ de type entier qui contient le nombre de "dislikes" de la vidéo.
- `id_chaine` : un champ de type entier qui contient l'identifiant de la chaîne à laquelle appartient la vidéo. Il est utilisé comme clé étrangère pour faire référence à la table `chaines`.
- `vue` : un champ de type entier qui contient le nombre de vues de la vidéo.
- `timestamp` : un champ de type entier qui contient la date de publication de la vidéo.
- `PRIMARY KEY` : un champ qui définit le champ `id` comme clé primaire.
- `FOREIGN KEY` : un champ qui définit le champ `id_chaine` comme clé étrangère pour faire référence à la table `chaines`.

Table `chaines`

Cette table contient des informations sur les chaînes. Elle possède les champs suivants :

- `id` : un entier qui représente l'identifiant unique de la chaîne, et qui est utilisé comme clé primaire. Il est également défini comme étant auto-incrémenté.
- `pseudo` : un champ de type texte qui contient le pseudo de la chaîne.
- `abonnes` : un champ de type entier qui contient le nombre d'abonnés de la chaîne.
- `bio` : un champ de type texte qui contient la biographie de la chaîne.
- `timestamp` : un champ de type entier qui contient la date de création de la chaîne.
- `email` : un champ de type texte qui contient l'email de la chaîne.
- `PRIMARY KEY` : un champ qui définit le champ `id` comme clé primaire.

- **FOREIGN KEY** : un champ qui définit le champ `id_chaine` comme clé étrangère pour faire référence à la table `videos`.

Table `commentaires`

Cette table contient des informations sur les commentaires. Elle possède les champs suivants :

- `id` : un entier qui représente l'identifiant unique du commentaire, et qui est utilisé comme clé primaire. Il est également défini comme étant auto-incrémenté.
- `contenue` : un champ de type texte qui contient le contenu du commentaire.
- `like` : un champ de type entier qui contient le nombre de "likes" du commentaire.
- `id_chaine` : un champ de type entier qui contient l'identifiant de la chaîne à laquelle appartient le commentaire. Il est utilisé comme clé étrangère pour faire référence à la table `chaines`.
- `id_video` : un champ de type entier qui contient l'identifiant de la vidéo à laquelle appartient le commentaire. Il est utilisé comme clé étrangère pour faire référence à la table `videos`.
- `timestamp` : un champ de type entier qui contient la date de publication du commentaire.
- **PRIMARY KEY** : un champ qui définit le champ `id` comme clé primaire.
- **FOREIGN KEY** : un champ qui définit les champs `id_chaine` et `id_video` comme clé étrangère pour faire référence à la table `chaines` et `videos`.

Table `##n_com``

Cette table contient des informations sur les relations entre les vidéos et les commentaires. Elle possède les champs suivants :

- `id_video` : un champ de type entier qui contient l'identifiant de la vidéo. Il est utilisé comme clé étrangère pour faire référence à la table `videos`.
- `id_commentaire` : un champ de type entier qui contient l'identifiant du commentaire. Il est utilisé comme clé étrangère pour faire référence à la table `commentaires`.
- **PRIMARY KEY** : un champ qui définit les champs `id_video` et `id_commentaire` comme clé primaire.
- **FOREIGN KEY** : un champ qui définit les champs `id_video` et `id_commentaire` comme clé étrangère pour faire référence à la table `videos` et `commentaires`.

Exemples de requêtes SQL pour interagir avec les tables

Requête pour sélectionner toutes les vidéos d'une chaîne donnée

```
SELECT * FROM videos
WHERE id_chaine = [id_de_la_chaine]
```

Requête pour sélectionner les 3 dernières vidéos publiées

```
SELECT * FROM videos
ORDER BY timestamp DESC
LIMIT 3
```

Requête pour sélectionner les commentaires d'une vidéo donnée

```
SELECT * FROM commentaires
WHERE id_video = [id_de_la_video]
```

Requête pour mettre à jour le nombre de vues d'une vidéo

```
UPDATE videos
SET vue = [nombre_de_vues]
WHERE id = [id_de_la_video]
```

Requête pour supprimer un commentaire

```
DELETE FROM commentaires
WHERE id = [id_du_commentaire]
```

Requête pour sélectionner les informations sur une vidéo et la chaîne à laquelle elle appartient

```
SELECT videos.*, chaines.*
FROM videos
JOIN chaines ON videos.id_chaine = chaines.id
WHERE videos.id = [id_de_la_video]
```

Requête pour sélectionner les commentaires d'une vidéo et les informations sur la chaîne qui les a publiés

```
SELECT commentaires.*, chaines.*
FROM commentaires
JOIN chaines ON commentaires.id_chaine = chaines.id
WHERE commentaires.id_video = [id_de_la_video]
```

Requête pour sélectionner les informations sur les vidéos d'une chaîne avec le nombre de commentaires pour chaque vidéo

```
SELECT videos.*, COUNT(n_com.id_video) AS nb_commentaires
FROM videos
JOIN n_com ON videos.id = n_com.id_video
WHERE videos.id_chaine = [id_de_la_chaine]
GROUP BY videos.id
```