Mini projet :

Data API YouTube

Réalisée par :
Ez-zakkar Mohammed
Oumar Ahmed Mohamedlemine

Encadre par :
Prof: Zahir Jihad

Plan:

1.Introduction :

L’objectif de cette mini projet consiste a crée une Base de données pour faire le stockage et l’analyse des vidéos (et leurs commentaires) correspondant à son mot-clé (ou groupe de mots-clés) et ayant été publiés pendant la plage de dates spécifiée. Et pour arriver à cette objectif on a suggéré un model conceptuel de données (MCD), et après on a transformé cette le MCD au model conceptuel de données (MLD), et on a gérée les requêtes correspondants a créé les tableaux (relations) dans le SGBD(MySql).

Et pour l’insertion et le stockage des données on a utilisé NodeJs et le package Google API pour obtenir les informations des vidéos sous format JSON, et le package MySql pour faire connecte la base de données qu’on a crée au NodeJs.

2.Dictionnaire de Données :

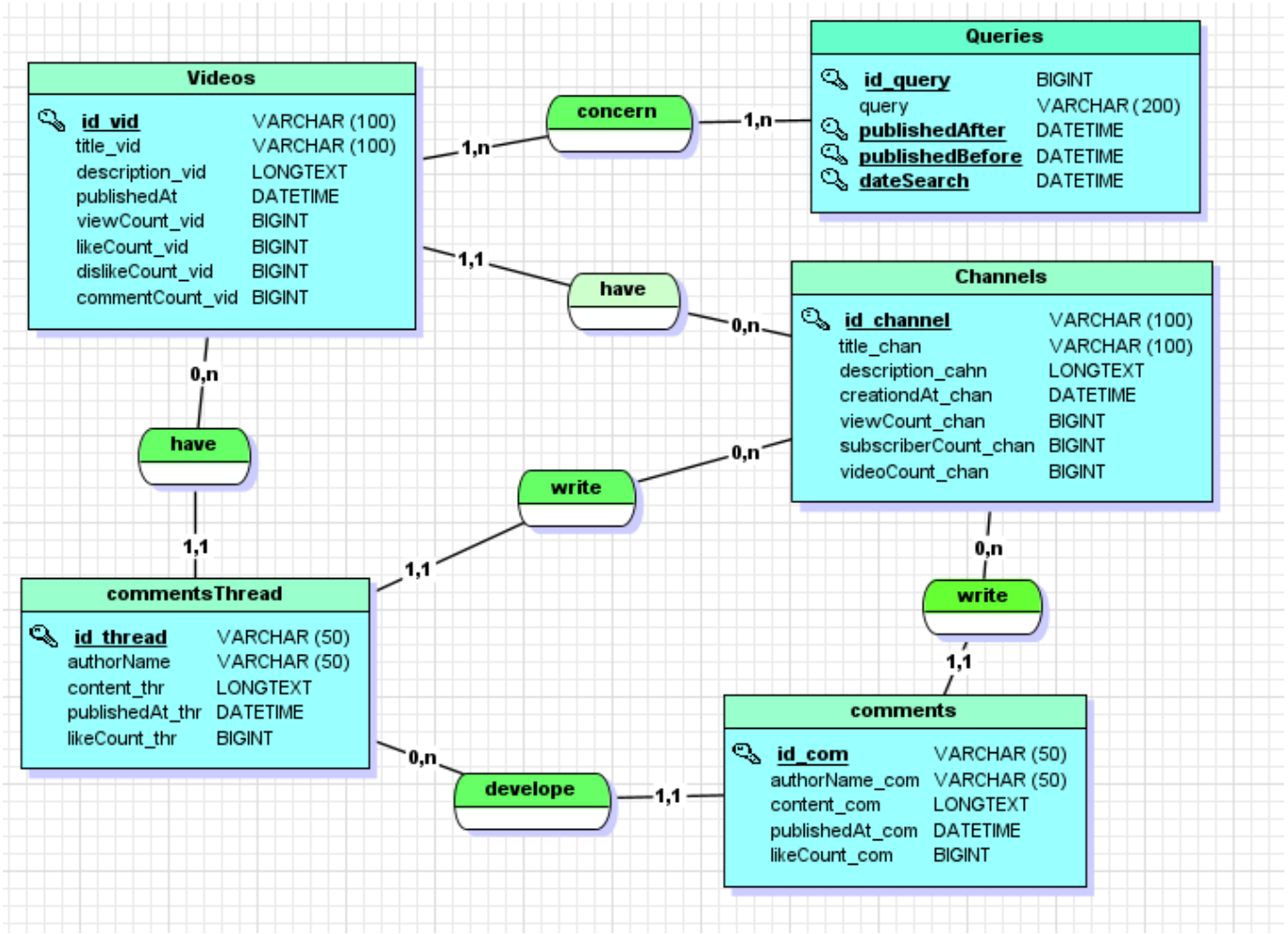
Le Tableau des dictionnaires des données suivant contient les attributs de chaque table avec ses descriptions :

Attribut	**Type**	**Description**
id_query	varchar	
publishedAfter	datetime	The **publishedAfter** parameter indicates that the API response should only contain resources created at or after the specified time
PublishedBefore	datetime	The **publishedBefore** parameter indicates that the API response should only contain resources created before or at the specified time.
dateSearch	datetime	
id_channel	varchar	Specifie the channel id
title_chan	varchar	The channel title
description_chan	varchar	The description of the channel
creationdAt_chan	datatime	The creation date
viewCount_chan	bigint	The views counter for the channel
subscriberCount_chan	bigint	The subscribtions counter for the channel
videoCount_chan	bigint	The counter of the channel videos
id_com	varchar	The id of the comment
content_com	text	The content of the comment
publishedAt_com	datetime	The publication date of the comment
likeCount_com	bigint	The counter of the likes
dislikeCount_com	bigint	The counter of the dislikes
id_thread	varchar	The id of the comment thread
content_thr	varchar	The content of the thread
publishedAt_thr	datetime	The date of the thread publication
likeCount_thr	bigint	The counter of the thread likes

dislikeCount_thr	bigint	The counter of the thread dislikes
id_vid	varchar	The video id
title_vid	varchar	The title of the video
description_vid	varchar	The description of the video
publishedAt	datetime	The publication date of the video
viewCount_vid	bigint	The counter of the video views
likeCount_vid	bigint	The counter of the likes of the video
dislikeCount_vid	bigint	The counter of the video dislikes
commentCount_vid	bigint	The counter of the comments on the video

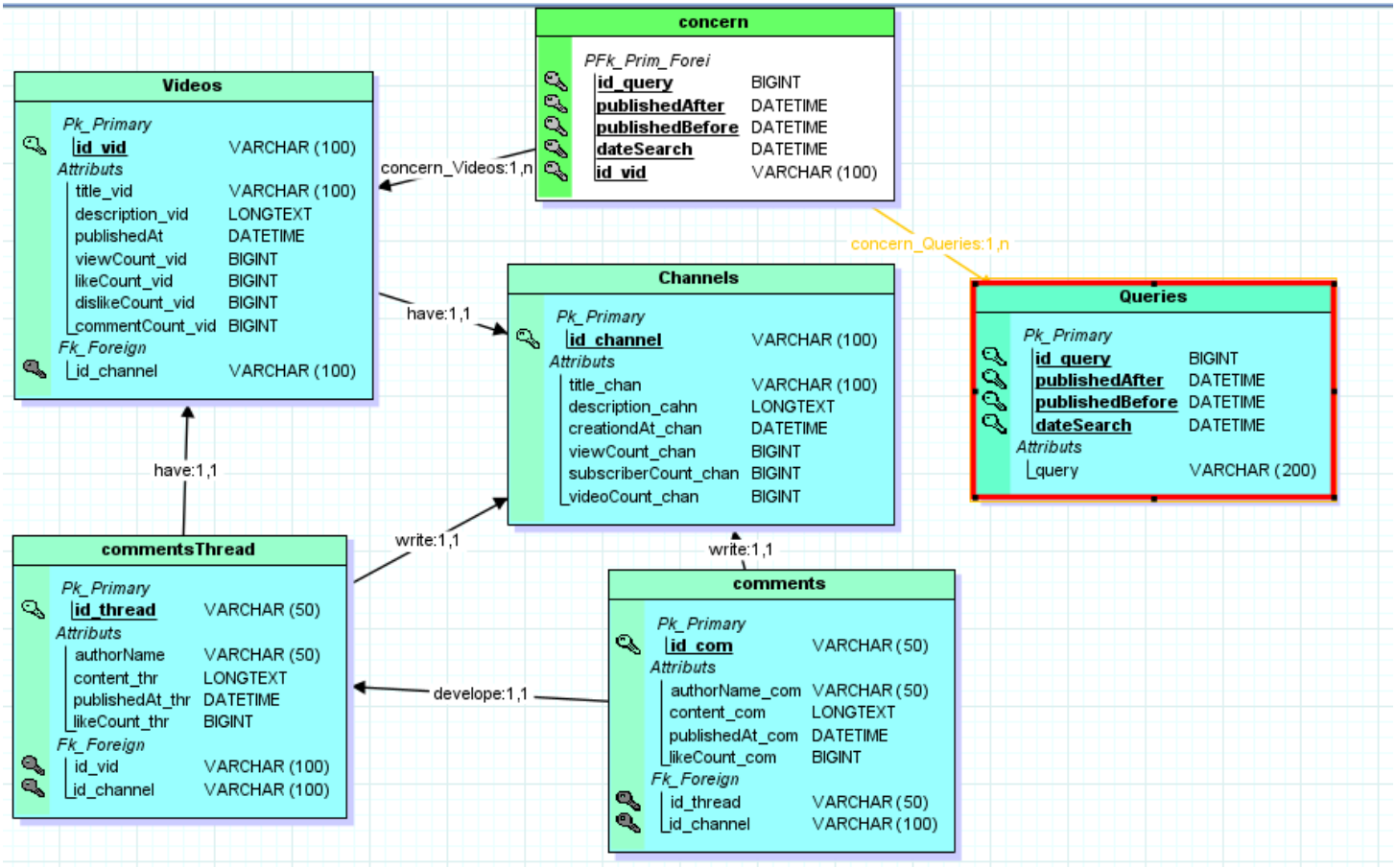
3.MCD :

Voici le model conceptuel de données qu'on a utilisé dans notre projet :



4.MLD

Le model logique de données :

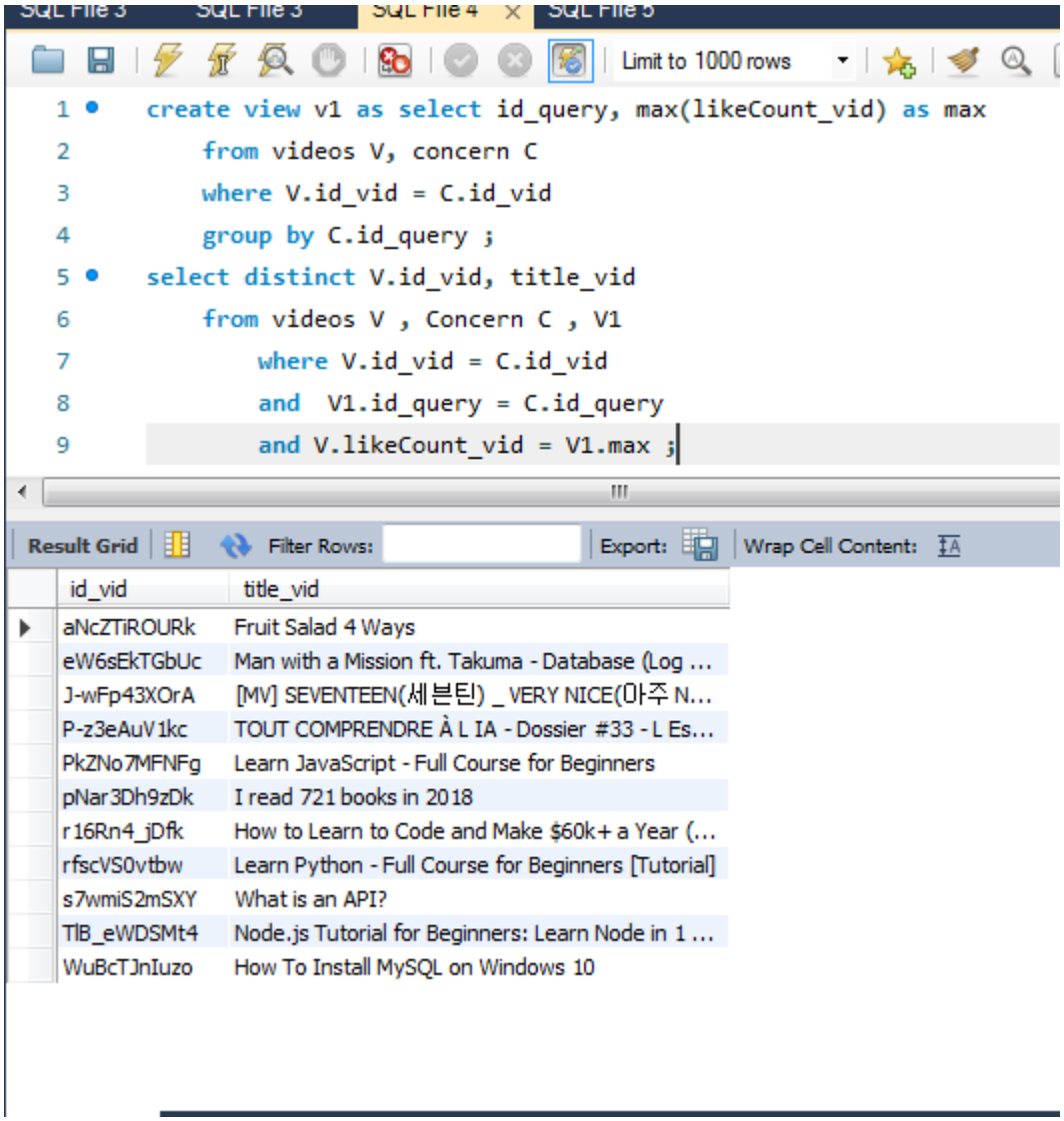


5.Les requêtes :

1. Quels sont le ID et le Titre de la vidéo qui a obtenu le maximum de « Like », pour chacune des « Queries » présentes dans la base de données.

- Create view v1 as select id_query, max(likeCount_vid) as max
  from videos V, concern C
  where V.id_vid = C.id_vid
  group by C.id_query ;
- Select distinct V.id_vid, title_vid
  from videos V , Concern C , V1

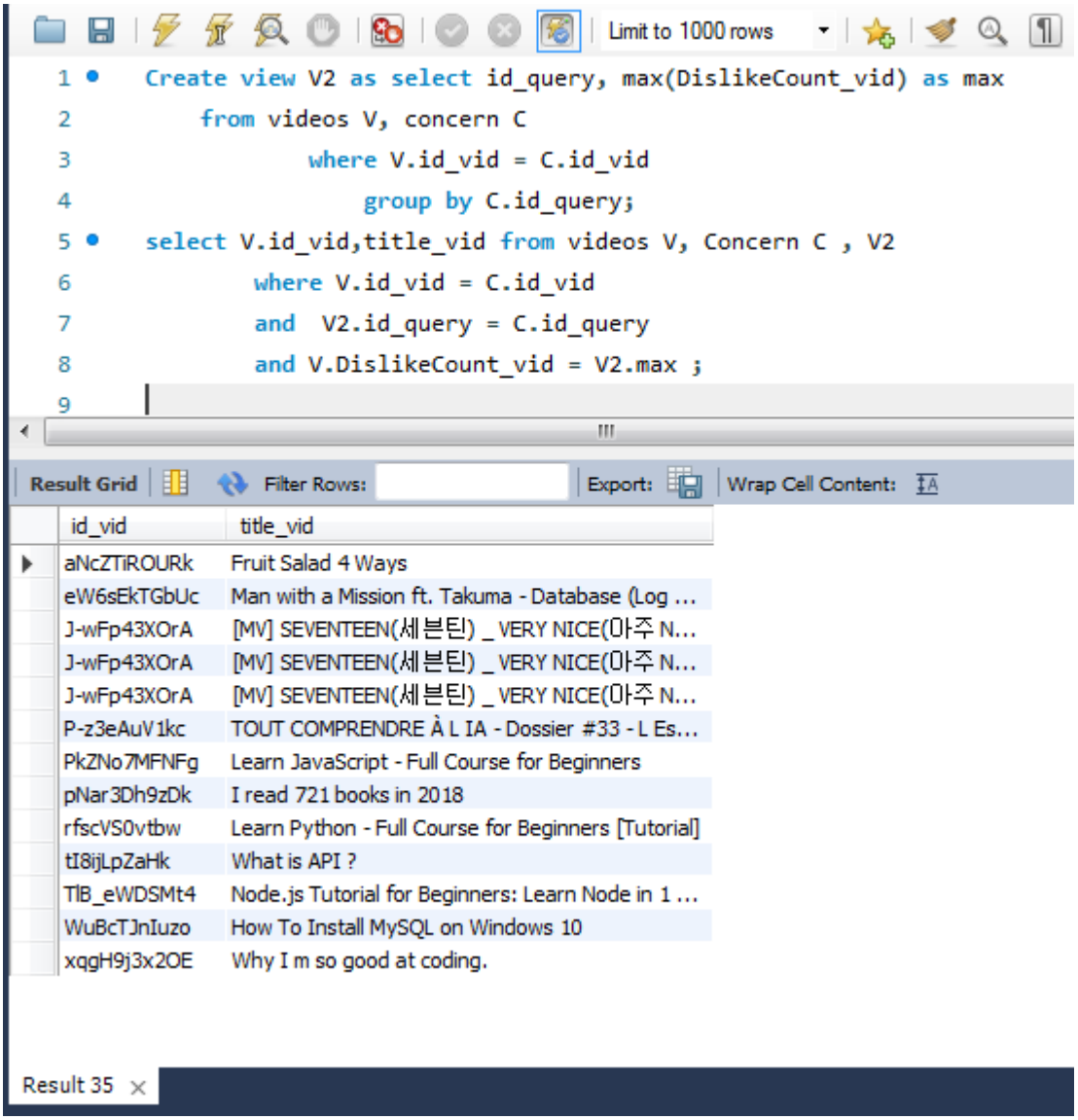
where V.id_vid = C.id_vid
and V1.id_query = C.id_query
and V.likeCount_vid = V1.max ;



2. Quels sont le ID et le Titre de la vidéo qui a obtenu le maximum de « DisLike » au maximum, pour chacune des Queries présente dans la base de données.

- Create view V2 as select id_query, max(DislikeCount_vid) as max
  from videos V, concern C
  where V.id_vid = C.id_vid
  group by C.id_query;

- Select V.id_vid , title_vid from videos V, Concern C , V2
  where V.id_vid = C.id_vid
  and V2.id_query = C.id_query
  and V.DislikeCount_vid = V2.max ;



3. Parmi les vidéos présentes dans la base de données, combien de vidéos ont été publiées par année et par « Query ».

   - SELECT Q.id_query ,
     Query ,

```
        C.publishedAfter ,
        C.publishedBefore ,
        Count(*) as nbVideos
FROM concern C , videos V , queries Q
WHERE C.id_vid = V.id_vid
        and Q.id_query = C.id_query
GROUP BY id_query, publishedAfter, publishedBefore
```

SQL File 3* | SQL File 3 | Administration - Data Export

Limit to 1000 rows

```
1  select Q.id_query,query,C.publishedAfter,C.publishedBefore,count(*)as nbVideos
2          from concern C , videos V , queries Q
3              where C.id_vid = V.id_vid and Q.id_query=C.id_query
4                  group by id_query,publishedAfter,publishedBefore
5
```

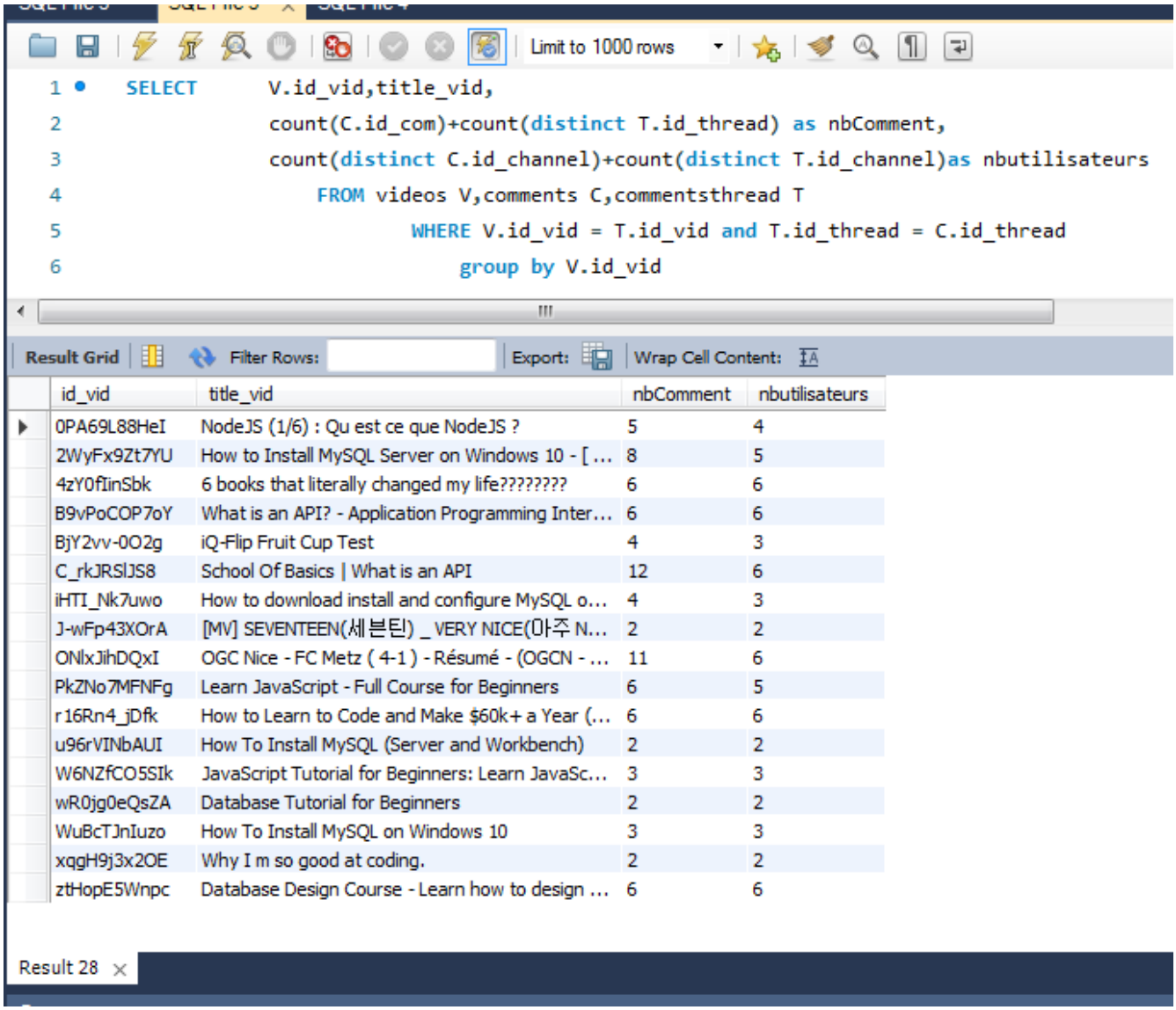
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id_query	query	publishedAfter	publishedBefore	nbVideos
0	javascript	2019-01-01 02:00:00	2019-12-16 02:00:00	2
1	nice	2017-01-01 00:00:00	2018-00-02 02:00:00	15
1	nice	2018-01-01 00:00:00	2018-12-02 02:00:00	15
1	nice	2019-01-01 02:00:00	2019-12-16 02:00:00	15
2	node js	2019-01-01 02:00:00	2019-12-16 02:00:00	3
3	books	2019-01-01 02:00:00	2019-11-16 02:00:00	10
4	IA	2019-01-01 02:00:00	2019-11-16 02:00:00	1
5	python	2018-01-01 00:00:00	2018-12-02 02:00:00	1
6	database	2018-01-01 00:00:00	2018-12-02 02:00:00	5
7	cup fruit	2018-01-01 00:00:00	2018-12-02 02:00:00	5
9	mysql ins...	2017-01-01 00:00:00	2018-00-02 02:00:00	5
10	what is API	2017-01-01 00:00:00	2018-00-02 02:00:00	5
11	coding	2017-01-01 00:00:00	2018-00-02 02:00:00	4

4. Pour chacune des vidéos présentes dans la base de données, quel est le nombre total des commentaires (y compris les réponses aux commentaires), et le nombre total des utilisateurs ayant rédigé ou répondu un commentaire.

- SELECT V.id_vid,title_vid,
  count(C.id_com)+count(distinct T.id_thread) as nbComment,
  count(distinct C.id_channel)+count(distinct T.id_channel)as nbutilisateurs

FROM videos V,comments C,commentsthread T
WHERE V.id_vid = T.id_vid and T.id_thread = C.id_thread
Group by V.id_vid



```
SELECT      V.id_vid,title_vid,
            count(C.id_com)+count(distinct T.id_thread) as nbComment,
            count(distinct C.id_channel)+count(distinct T.id_channel)as nbutilisateurs
                FROM videos V,comments C,commentsthread T
                        WHERE V.id_vid = T.id_vid and T.id_thread = C.id_thread
                            group by V.id_vid
```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

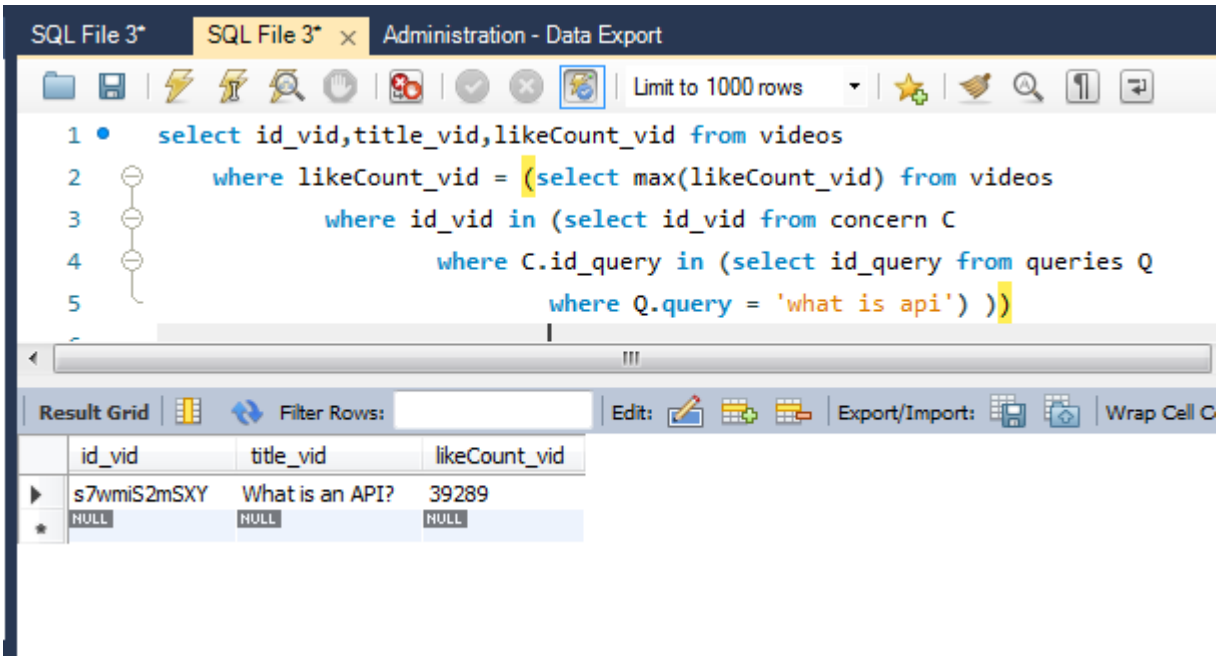
id_vid	title_vid	nbComment	nbutilisateurs
0PA69L88HeI	NodeJS (1/6) : Qu est ce que NodeJS ?	5	4
2WyFx9Zt7YU	How to Install MySQL Server on Windows 10 - [...	8	5
4zY0fIinSbk	6 books that literally changed my life????????	6	6
B9vPoCOP7oY	What is an API? - Application Programming Inter...	6	6
BjY2vv-0O2g	iQ-Flip Fruit Cup Test	4	3
C_rkJRSlJS8	School Of Basics | What is an API	12	6
iHTI_Nk7uwo	How to download install and configure MySQL o...	4	3
J-wFp43XOrA	[MV] SEVENTEEN(세븐틴) _ VERY NICE(아주 N...	2	2
ONlxJihDQxI	OGC Nice - FC Metz (4-1) - Résumé - (OGCN - ...	11	6
PkZNo7MFNFg	Learn JavaScript - Full Course for Beginners	6	5
r16Rn4_jDfk	How to Learn to Code and Make $60k+ a Year (...	6	6
u96rVINbAUI	How To Install MySQL (Server and Workbench)	2	2
W6NZfCO5SIk	JavaScript Tutorial for Beginners: Learn JavaSc...	3	3
wR0jg0eQsZA	Database Tutorial for Beginners	2	2
WuBcTJnIuzo	How To Install MySQL on Windows 10	3	3
xqgH9j3x2OE	Why I m so good at coding.	2	2
ztHopE5Wnpc	Database Design Course - Learn how to design ...	6	6

Result 28

5. Pour une « Query » donnée, quels sont le ID et le Titre de la vidéo qui a été « Liké » au maximum.

- SELECT id_vid,title_vid,likeCount_vid from videos
  where likeCount_vid = (select max(likeCount_vid)
  From videos
  where id_vid in (select id_vid from concern C
  where C.id_query in (select id_query from queries Q

where Q.query = 'what is api')))



6. Quels sont le ID et le Titre de la vidéo qui a reçu le maximum de commentaires, pour chacune des « Queries » présentes dans la base de données.

- Create view vi As select Q.id_query,query,max(CommentCount_vid)As nbcomment

  from videos V ,concern C ,queries Q
  where V.id_vid = C.id_vid
  and Q.id_query = C.id_query
  group by Q.id_query;

- select distinct title_vid,V.id_vid from videos V,concern C ,vi
  where V.id_vid = C.id_vid
  and vi.id_query = C.id_query
  and V.CommentCount_vid = vi.nbcomment

```
1  create view vi as select Q.id_query,query,max(CommentCount_vid)as nbcomment
2                          from videos V ,concern C ,queries Q
3                              where V.id_vid = C.id_vid
4                              and Q.id_query = C.id_query
5                              group by Q.id_query;
6  select distinct title_vid,V.id_vid from videos V,concern C ,vi
7  where V.id_vid = C.id_vid
8  and vi.id_query = C.id_query
9  and V.CommentCount_vid = vi.nbcomment
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

title_vid	id_vid
Fruit Salad 4 Ways	aNcZTiROURk
Man with a Mission ft. Takuma - Database (Log ...	eW6sEkTGbUc
[MV] SEVENTEEN(세븐틴) _ VERY NICE(아주 N...	J-wFp43XOrA
TOUT COMPRENDRE À L IA - Dossier #33 - L Es...	P-z3eAuV1kc
Learn JavaScript - Full Course for Beginners	PkZNo7MFNFg
I read 721 books in 2018	pNar3Dh9zDk
How to Learn to Code and Make $60k+ a Year (...	r16Rn4_jDfk
Learn Python - Full Course for Beginners [Tutorial]	rfscVS0vtbw
What is API ?	tI8ijLpZaHk
Node.js Tutorial for Beginners: Learn Node in 1 ...	TlB_eWDSMt4
How To Install MySQL on Windows 10	WuBcTJnIuzo

7. Pour une requête donnée, afficher l'utilisateur ayant rédigé le commentaire le plus populaire concernant la vidéo la plus « Likée »

- create view V11 as select distinct V.id_vid
  from videos V , concern C , queries Q
  where C.id_vid = V.id_vid
  and C.id_query = Q.id_query
  and query = 'nice'
  and likeCount_vid = (select max(V.likeCount_vid)
  from videos V , concern C , queries Q
  where C.id_vid = V.id_vid
  and C.id_query = Q.id_query

and query = 'nice');

- select authorName,id_thread,V.likeCount_vid,T.likeCount_thr
  from commentsThread T,videos V
  where T.id_vid = V.id_vid and V.id_vid = (select id_vid from V11)
  and likeCount_thr=(select max(likeCount_thr)
  from commentsThread T
  where T.id_vid=(select id_vid from V11));