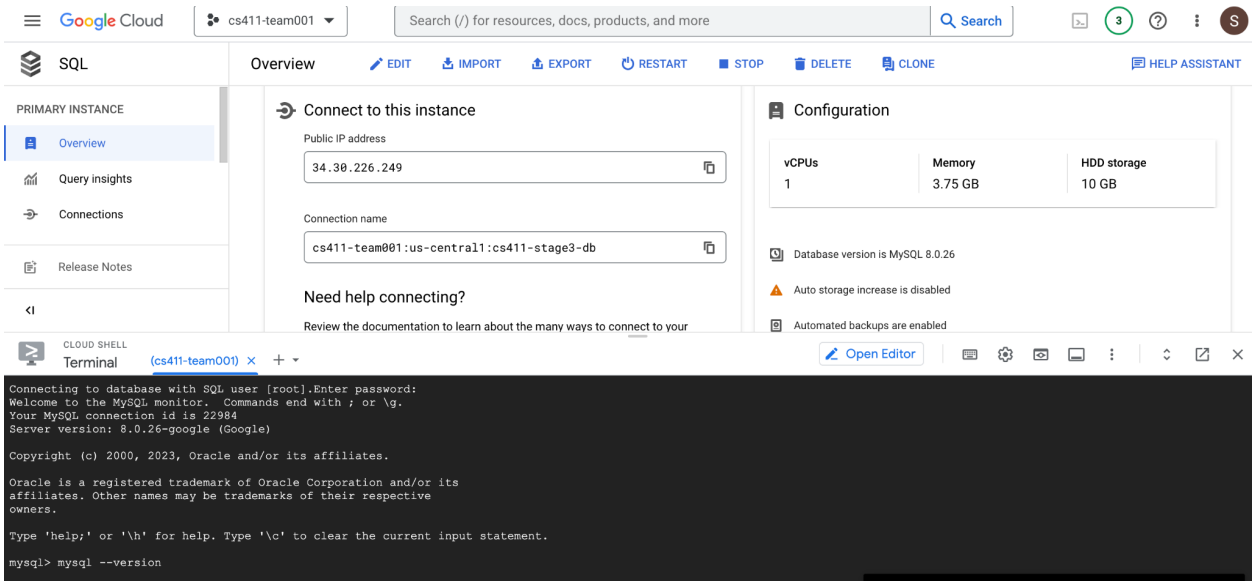
CS 411 Project Stage 3: Database Implementation and Indexing

1. Screenshot of Connection:



2. DDL Commands

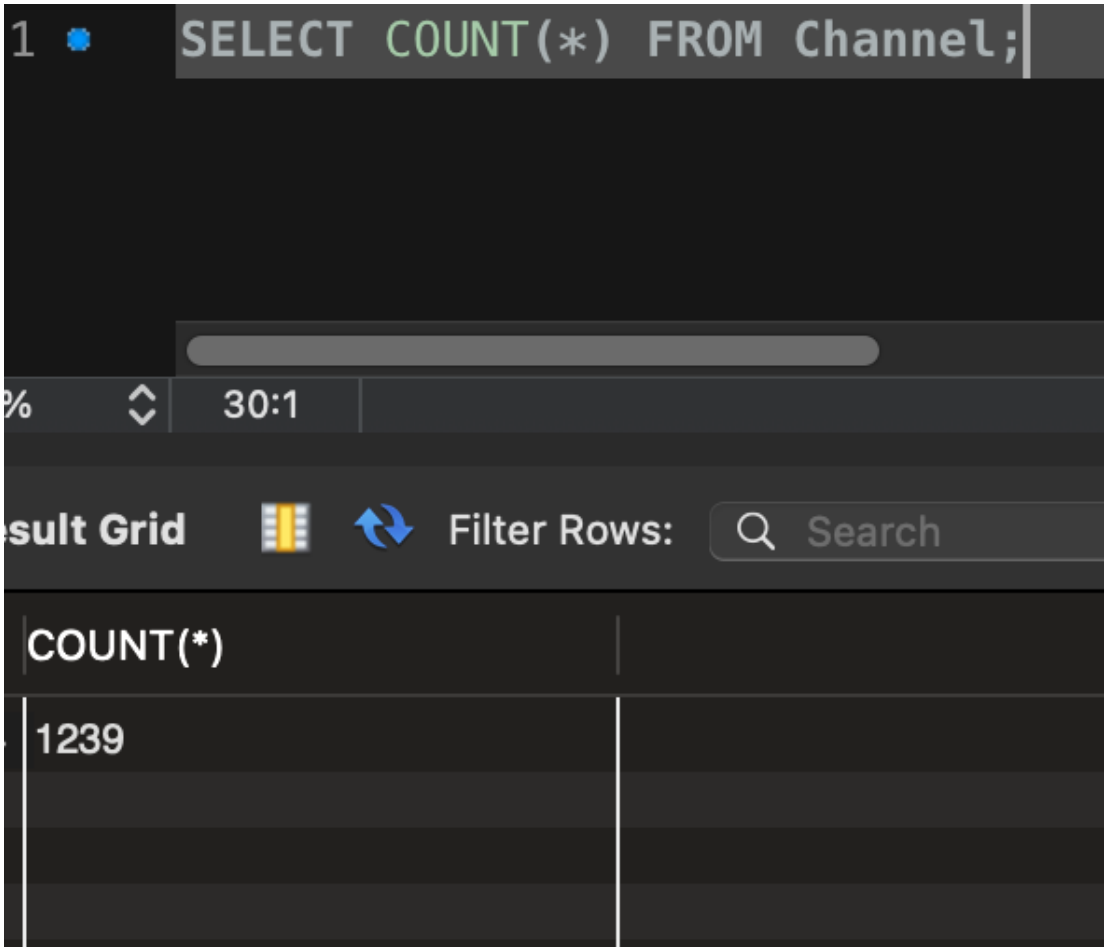
```
CREATE TABLE Channel (
    ChannelId VARCHAR(255) NOT NULL,
    ChannelTitle VARCHAR(255),
    TotalViews INT,
    PRIMARY KEY (ChannelId)
);

CREATE TABLE User (
    ChannelName VARCHAR(255) NOT NULL,
    Email VARCHAR(100),
    Password VARCHAR(255),
```

```
    PRIMARY KEY (ChannelName)
);
CREATE TABLE Video (
    VideoId VARCHAR(255) NOT NULL,
    VideoTitle VARCHAR(255),
    CategoryId  INT NOT NULL,
    ChannelId VARCHAR(255) NOT NULL,
    ChannelName VARCHAR(255) NOT NULL,
    PRIMARY KEY (VideoId),
    FOREIGN KEY (CategoryId) REFERENCES Category(CategoryId),
    FOREIGN KEY (ChannelId) REFERENCES Channel(ChannelId),
    FOREIGN KEY (ChannelName) REFERENCES User(ChannelName)
);

CREATE TABLE Tag (
    TagName VARCHAR(255) PRIMARY KEY
);

CREATE TABLE Category (
    CategoryId INT NOT NULL,
    CategoryName VARCHAR(100),
    PRIMARY KEY (CategoryId)
);
```

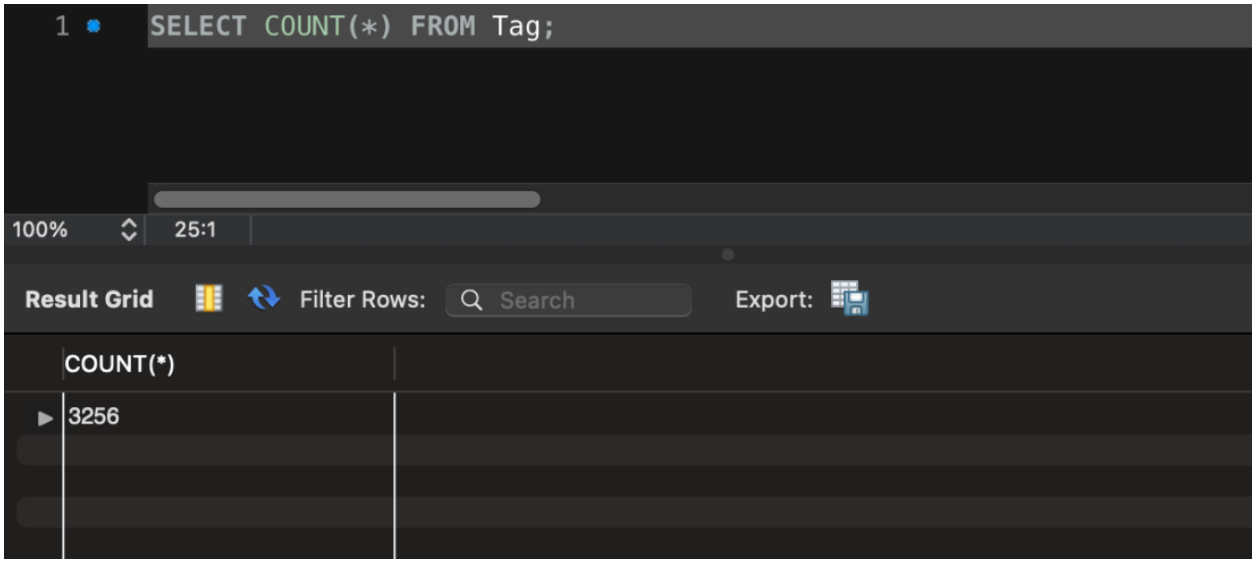
```
CREATE TABLE HasTag (
    VideoId VARCHAR(255) ,
    TagName VARCHAR(255),
    FOREIGN KEY (VideoId) REFERENCES Video ON DELETE CASCADE
    FOREIGN KEY (TagName) REFERENCES Tag ON DELETE CASCADE
);
```

3. **Screenshots of Count Queries for 3 tables**

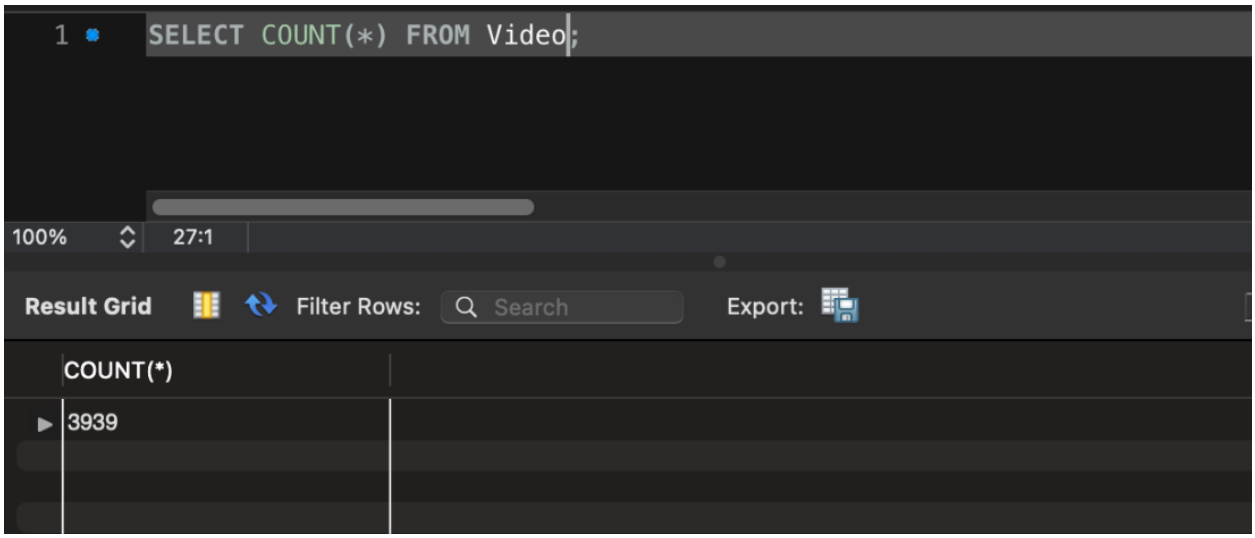
Screenshot for Channel Table:



Screenshot for Tag Table:



Screenshot for Video Table:



4. **Advanced queries and screenshots of Top 15 rows for advanced queries**

Advanced Query 1:

SELECT *
FROM Video v JOIN Category c ON (c.CategoryId = v.CategoryId) JOIN Channel ch ON (ch.ChannelId = v.ChannelId)
WHERE (SELECT TotalViews FROM Channel c2 WHERE (c2.ChannelId = ch.ChannelId)) > 100000 LIMIT 15;

This query selects the top videos per category which have all more than 100,000 in views

	VideoId	VideoTitle	Categ...	ChannelId	Channel
▶	_3DHGXgZ-_s	Game Theory: Sonic is TOO Powerful! (Sonic th…	20	UCo_IB5145EVNcf8hw1Kku7w	The Gan
	_J9Z74dQXrQ	Leipzig vs. Atlético Madrid I Champions League…	17	UCET00YnetHT7tOpu12v8jxg	Champic
	_jj_-ElKrdM	BITTEN - by a GIANT CATFISH!	15	UC6E2mP01ZLH_kbAyeazCNdg	Brave W
	_rJYc_k-w84	ANNABELLE THE DOLL ESCAPED?	24	UCrp8aFu6VjkZAY9Hhj6IrXA	billschar
	_yKoFS4mgHI	SHOOTING AGAINST 16 KEEPERS I IMPOSSI…	17	UCKvn9VBLAiLiYL4FFJHri6g	F2Frees
	-bCiD4QGSvQ	The Weeknd Experience LIVE - Director's Cut (…	10	UC0WP5P-ufpRfjbNrmOWwLBQ	The Wee
	-N0x_Tt7WVc	A Zombie Extra's First Day on Set - Key & Peele	23	UCdN4aXTrHAtfgbVG9HjBmxQ	Key & P
	-n8lrFPSFCU	LOGAN PAUL ADDRESSES JAKE PAUL'S FBI…	24	UCE9ZKI1b_PhVm3gejYuilhw	Impaulsi
	-rn_R1aheK8	Lapiz Conciente - 9 Dias	10	UCtYkcye4qLtsfODz17_PHdg	UANLOF
	0a5YhsnITFE	LFR13 - QR, Game 5 - Done - CBJ 3, Tor 0	17	UCkUjSzthJUlO0uyUpiJfnxg	SteveDa
	0C80BSgjb8M	YoungBoy Never Broke Again - Kacey talk	10	UCIW4jraMKz6Qj69lJf-tODA	YoungBo
	0La8uz7WJiU	Stephen A. reacts to Damian Lillard dropping 61…	17	UCiWLfSweyRNmLpgEHekhoAg	ESPN
	0opZqh_TprM	Internet Money - Lemonade ft. Don Toliver, Gun…	24	UCtylTUUVIGY_i5afsQYeBZA	Lyrical L
	0VJ37Gsmq4Y	BRASA ❌ CHANTAL MEDINA - EX (Video Ofic…	10	UCHB2HyAz_s3iyNrjTjQTL5w	Brasa
	0wurtsQ31TE	Why Everest Isn't Earth's Highest Mountain... so…	27	UCP5tjEmvPItGyLhmjdwP7Ww	RealLife

Advanced Query 2:

```
SELECT Channel.ChannelTitle, COUNT(*) AS NumOfVideos, SUM(TotalViews) AS
TotalViews
FROM Channel INNER JOIN Video ON Channel.ChannelId = Video.ChannelId
GROUP BY Channel.ChannelTitle
ORDER BY NumOfVideos DESC;
```

This query joins the Channel and Video tables based on the ChannelId column, groups the result by ChannelTitle and calculates the count of videos and the total views for each channel, and then orders the result by the number of videos in descending order.

ChannelTitle	NumOfVideos	TotalViews
Champions League on CBS Sports	7	5236693
NBA on TNT	4	1457508
XXL	3	1490313
Apex Legends	3	7145064
Brooklyn and Bailey	3	1817034
NBA	3	2812707
MrBeast	3	63068748
UFC - Ultimate Fighting Championship	3	1898874
JYP Entertainment	3	17999196
The Game Theorists	2	5315782
Dua Lipa	2	4050634
Linkin Park	2	661746
Mama Rug and Papa Rug	2	997640
TREASURE (트레저)	2	4588464
ESPN	2	1582978

5. **Screenshot of EXPLAIN ANALYZE COMMAND**

Original Advanced Query 1

```
EXPLAIN ANALYZE SELECT *
FROM Video v JOIN Category c ON (c.CategoryId = v.CategoryId) JOIN Channel ch (
WHERE (SELECT TotalViews FROM Channel c2 WHERE (c2.ChannelId = ch.ChannelId)) >
```

1:1

itor Navigate: 1 / 1

```
-> Nested loop inner join  (cost=2691.95 rows=3843) (actual time=0.214..32.520 rows=3089 loops=1)
    -> Nested loop inner join  (cost=1346.90 rows=3843) (actual time=0.070..7.280 rows=3939 loops=1)
```

Form Editor

Field Types

Query 1 after 1st indexing design:

```
-- CREATE INDEX cat_idx on Video(CategoryId);
EXPLAIN ANALYZE SELECT *
FROM Video v JOIN Category c ON (c.CategoryId = v.CategoryId) JOIN Channel ch (
WHERE (SELECT TotalViews FROM Channel c2 WHERE (c2.ChannelId = ch.ChannelId)) >
```

1:5

itor Navigate: 1 / 1

```
-> Nested loop inner join  (cost=2691.95 rows=3843) (actual time=0.207..32.789 rows=3089 loops=1)
    -> Nested loop inner join  (cost=1346.90 rows=3843) (actual time=0.088..7.299 rows=3939 loops=1)
```

Form Editor

Field Types

Query 1 after 2nd indexing design:

```
-- CREATE INDEX chan_idx on Channel(ChannelId);
EXPLAIN ANALYZE SELECT *
FROM Video v JOIN Category c ON (c.CategoryId = v.CategoryId) JOIN Channel ch (
WHERE (SELECT TotalViews FROM Channel c2 WHERE (c2.ChannelId = ch.ChannelId)) >
```

1:5

m Editor Navigate: 1 / 1

```
-> Nested loop inner join  (cost=3098.65 rows=3843) (actual time=0.103..30.145 rows=3089 loops=1)
    -> Nested loop inner join  (cost=1753.60 rows=3843) (actual time=0.068..5.065 rows=3939 loops=1)
```

Form Editor

Field Types

Query 1 after 3rd indexing design:

```
-- CREATE INDEX cath_idx on Category(CategoryId);
EXPLAIN ANALYZE SELECT *
FROM Video v JOIN Category c ON (c.CategoryId = v.CategoryId) JOIN Channel ch (
WHERE (SELECT TotalViews FROM Channel c2 WHERE (c2.ChannelId = ch.ChannelId)) >
```

1:5

Editor Navigate: 1 / 1

Form Editor

Field Types

```
-> Nested loop inner join  (cost=3098.65 rows=3843) (actual time=0.100..30.049 rows=3089 loops=1)
    -> Nested loop inner join  (cost=1753.60 rows=3843) (actual time=0.060..5.034 rows=3939 loops=1)
```

Query 1 Indexing Analysis:

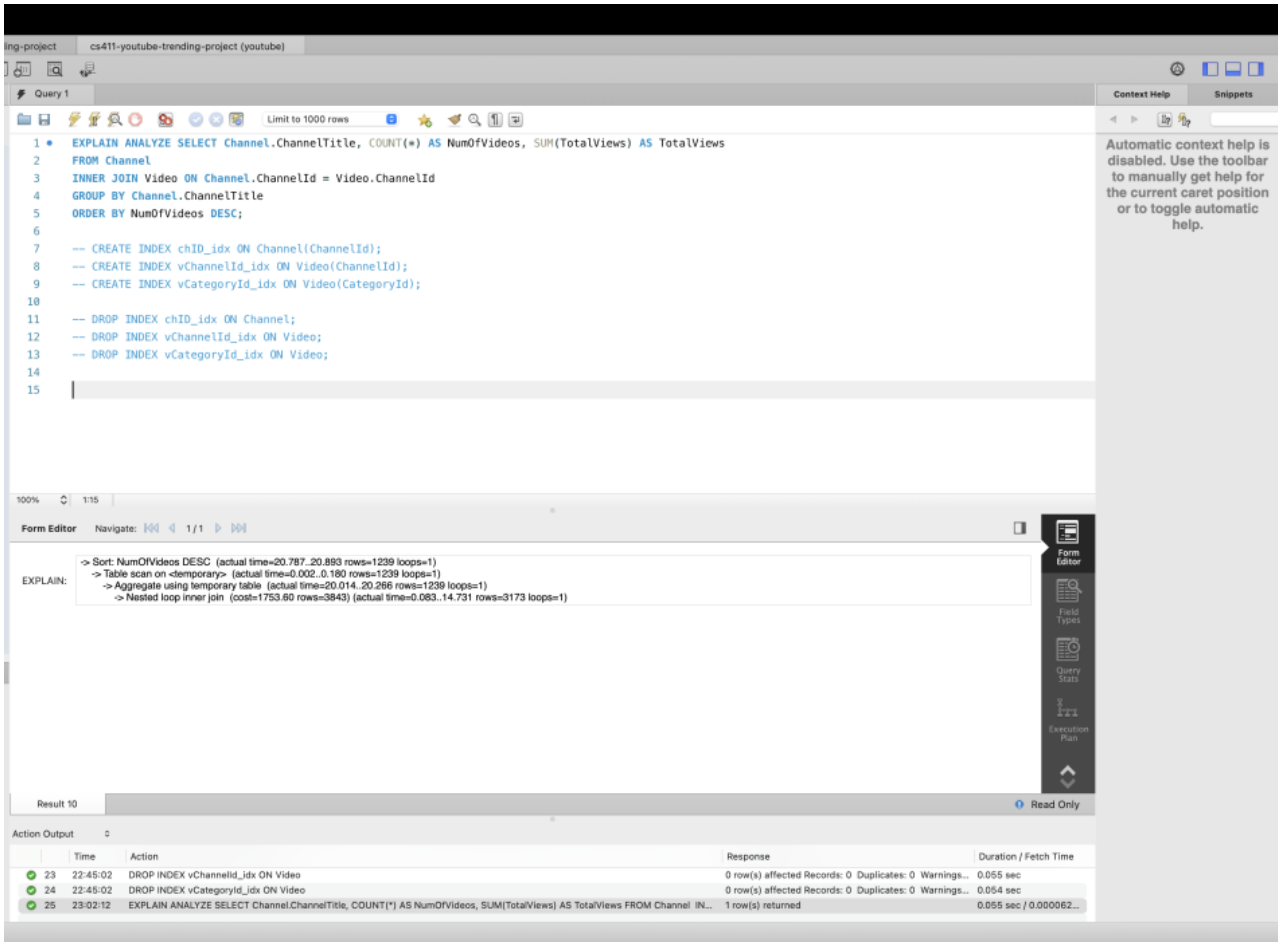
The reason why creating an index on the CategoryId column in the Video table did not change the cost or performance of the query is because the index was not used by the query optimizer in the execution plan.

Although the CategoryId column is used in the Video table to join with the Category table, it is not used in the SELECT or GROUP BY clauses of the query. Therefore, the query optimizer may have chosen to use the index on the ChannelId column in both the Channel and Video tables to perform the JOIN operation, and then perform the aggregation using an in-memory hash table instead of the index on the CategoryId column.

Additionally, the index on the CategoryId column may not be very useful if there are many duplicate values in the column, or if the number of distinct values in the column is small. In this case, the query optimizer may determine that a full table scan is more efficient than using the index.

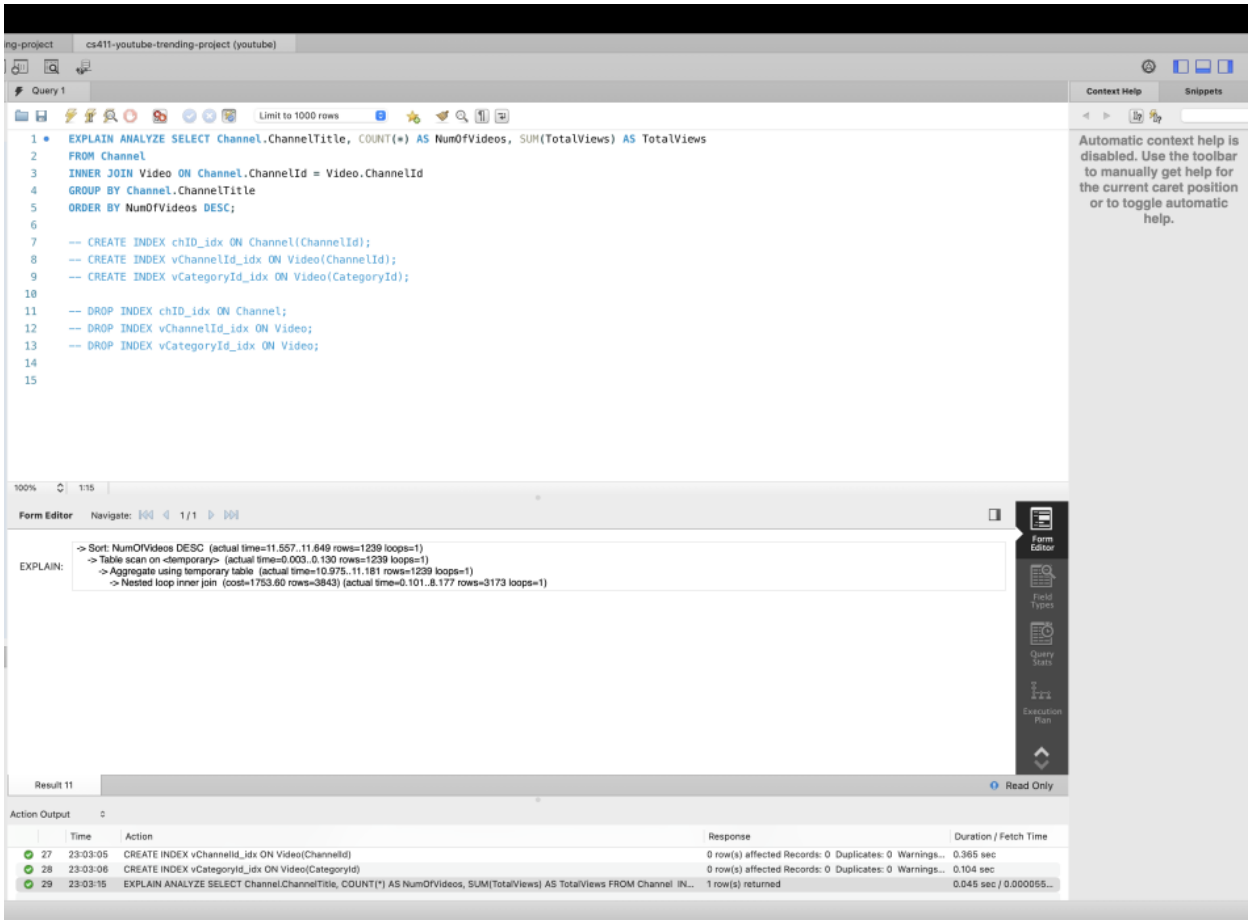
Overall, it's important to carefully analyze the query and understand which columns are most critical for performance, as not all indexes will be useful for all queries.

Original Advanced Query 2



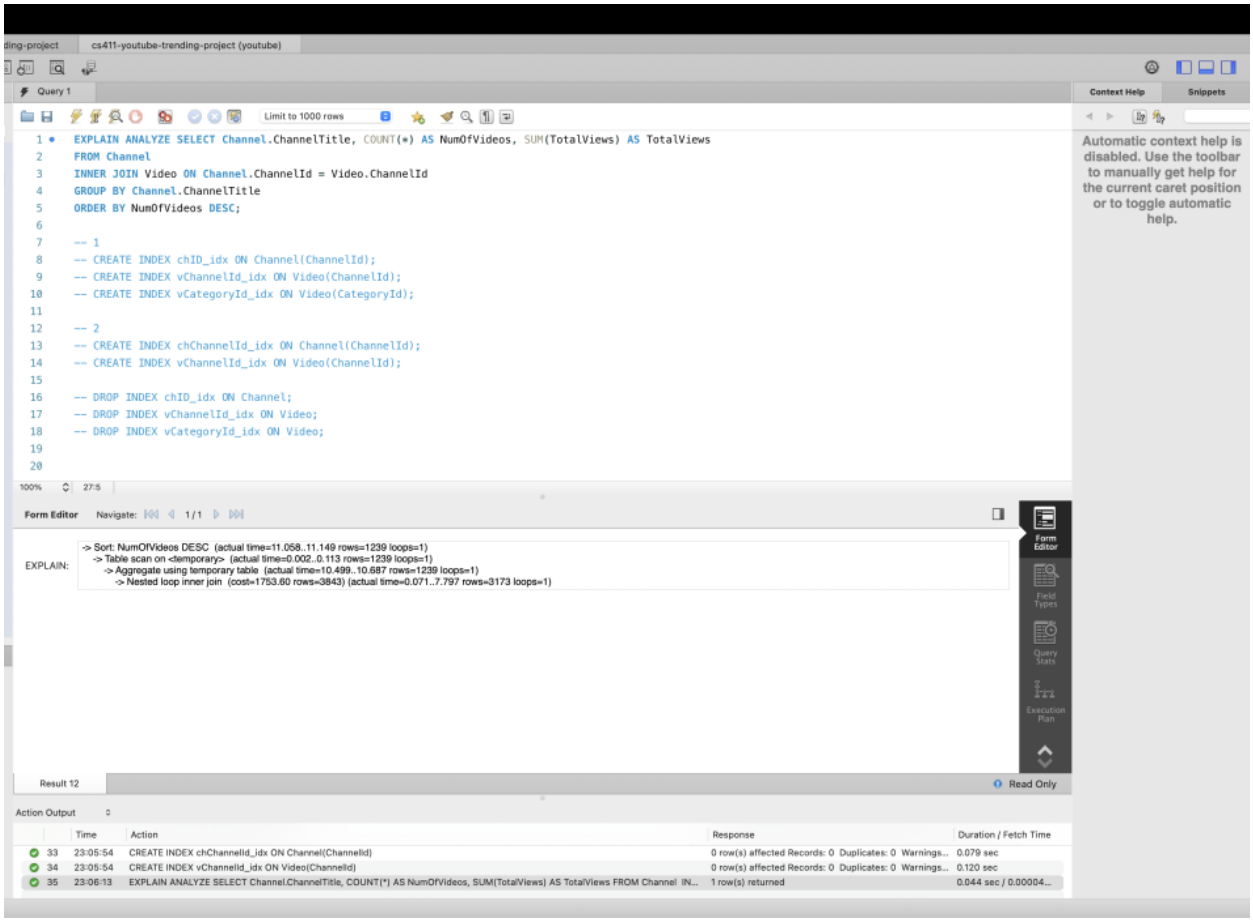
Query 2 after 1st indexing design:

CREATE INDEX chID_idx ON Channel(ChannelId);
CREATE INDEX vChannelId_idx ON Video(ChannelId);
CREATE INDEX vCategoryId_idx ON Video(CategoryId);



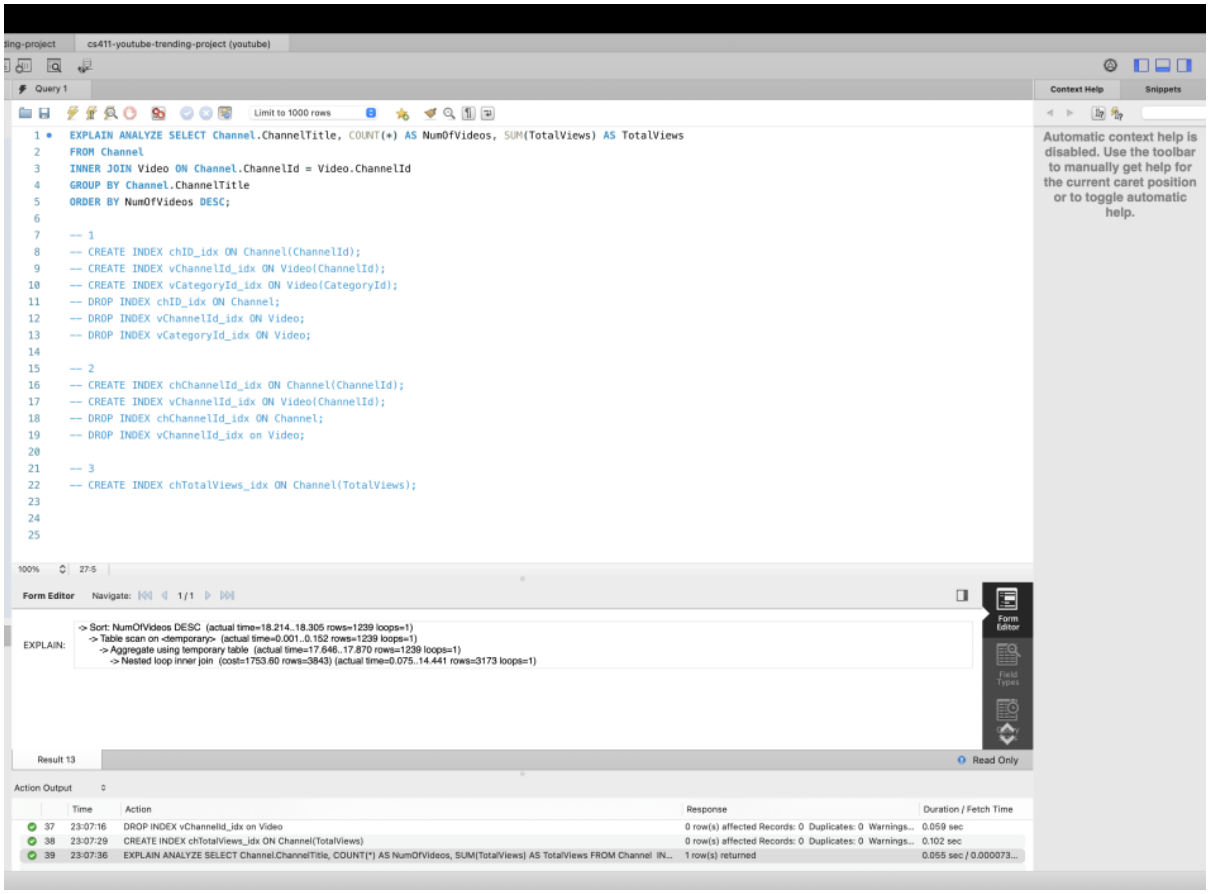
Query 2 after 2nd indexing design:

CREATE INDEX chChannelId_idx ON Channel(ChannelId);
CREATE INDEX vChannelId_idx ON Video(ChannelId);



Query 2 after 3rd indexing design:

CREATE INDEX chTotalViews_idx ON Channel(TotalViews);



Query 2 Indexing Analysis:

Reflecting on the three index designs that we analyzed, we determined that our second index design would be selected for query 2. This is because the second index design had the lowest runtimes and best performance using the "EXPLAIN ANALYZE" command.

Specifically, the 2nd index design is effective because it includes indexes on the ChannelId column in both the Channel and Video tables. Since the query performs an INNER JOIN between the two tables on the ChannelId column, having an index on this column can significantly improve the performance of the query by reducing the number of disk reads required to match the rows in the two tables. By creating the indexes on the ChannelId column in both tables, the query optimizer can efficiently access the relevant data in each table, reducing the amount of time required to perform the JOIN operation. This index design is especially effective when the Channel and Video tables are large and contain a significant number of rows, as it can help avoid full table scans and make the query run much faster.