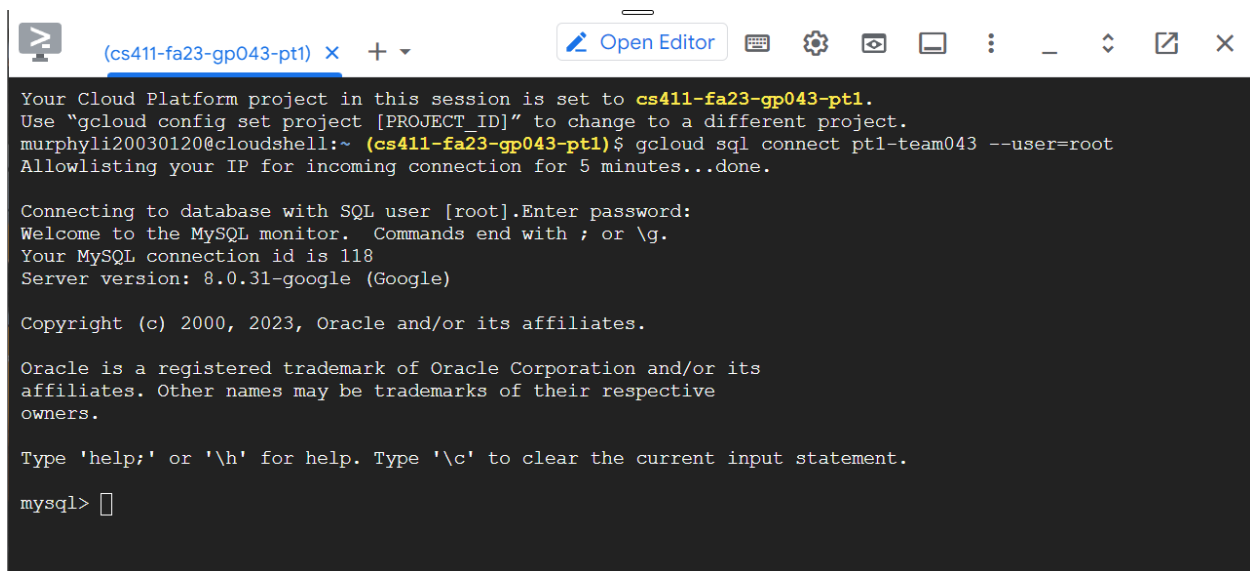


CS411 Stage3

Team043-InfelPhira

Connecting Database on GCP:

Screenshot of connection



```
(cs411-fa23-gp043-pt1) x + ▾ Open Editor
Your Cloud Platform project in this session is set to cs411-fa23-gp043-pt1.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
murphyli20030120@cloudshell:~ (cs411-fa23-gp043-pt1)$ gcloud sql connect pt1-team043 --user=root
Allowlisting your IP for incoming connection for 5 minutes...done.

Connecting to database with SQL user [root].Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 118
Server version: 8.0.31-google (Google)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

DDL Commands:

DDL (Data Definition Language) Commands:

```
CREATE TABLE Channels(  
    channelId VARCHAR(32),  
    channelName VARCHAR(255),  
  
    PRIMARY KEY (channelId)  
);  
  
CREATE TABLE Videos (  
    videoId          VARCHAR(32) PRIMARY KEY,  
    title            VARCHAR(255),  
    publishedAt      DATETIME,  
    channelId        VARCHAR(32) REFERENCES Channels(channelId),  
    categoryId       INT,  
    trendingDate     DATETIME,  
    viewCount        INT,  
    likes            INT,  
    dislikes         INT,  
    commentCount     INT,  
    commentsDisabled BIT,  
    ratingsDisabled  BIT,  
    description      TEXT,  
  
    FOREIGN KEY (channelId) REFERENCES Channels(channelId)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE Bunches(  
    bunchId VARCHAR(32),  
    userId  VARCHAR(32),  
  
    PRIMARY KEY (bunchId),  
    FOREIGN KEY (userId) REFERENCES Users(userId)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);  
  
CREATE TABLE Users(  
    userId  VARCHAR(32),  
    userName VARCHAR(255),
```

```

    passwd VARCHAR(255),

    PRIMARY KEY (userId)
);

CREATE TABLE Comments(
    commentId VARCHAR(32),
    userId VARCHAR(32),
    videoId VARCHAR(32),
    parentcommentId VARCHAR(32),

    PRIMARY KEY (commentId),
    FOREIGN KEY (userId) REFERENCES Users(userId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (videoId) REFERENCES Videos(videoId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE BunchComments(
    bunchCommentId VARCHAR(32),
    userId VARCHAR(32),
    bunchId VARCHAR(32),
    ParentbunchCommentId VARCHAR(32),

    PRIMARY KEY (bunchCommentId),
    FOREIGN KEY (userId) REFERENCES Users(userId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (bunchId) REFERENCES Bunches(bunchId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE BunchContain(
    videoId VARCHAR(32),
    bunchId VARCHAR(32),

    PRIMARY KEY (videoId, bunchId),
    FOREIGN KEY (videoId) REFERENCES Videos(videoId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (bunchId) REFERENCES Bunches(bunchId)
        ON DELETE CASCADE

```

```

        ON UPDATE CASCADE
    );

CREATE TABLE ChannelLike(
    userId VARCHAR(32),
    channelId VARCHAR(32),

    PRIMARY KEY(userId, channelId),
    FOREIGN KEY (userId) REFERENCES Users(userId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (channelId) REFERENCES Channels(channelId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE BunchLike(
    userId VARCHAR(32),
    bunchId VARCHAR(32),

    PRIMARY KEY(userId, bunchId),
    FOREIGN KEY (userId) REFERENCES Users(userId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (bunchId) REFERENCES Bunches(bunchId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE CommentLike(
    userId VARCHAR(32),
    commentId VARCHAR(32),

    PRIMARY KEY(userId, commentId),
    FOREIGN KEY (userId) REFERENCES Users(userId)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (commentId) REFERENCES Comments(commentId)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

CREATE TABLE BunchCommentLike(
    userId VARCHAR(32),
    bunchCommentId VARCHAR(32),

```

```

PRIMARY KEY(userId, bunchCommentId),
FOREIGN KEY (userId) REFERENCES Users(userId)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
FOREIGN KEY (bunchCommentId) REFERENCES BunchComments(bunchCommentId)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE VideoTags (
    tagId          INT PRIMARY KEY,
    tagName        VARCHAR(255)
);

CREATE TABLE HasTag (
    videoId        VARCHAR(32),
    tagId          INT PRIMARY KEY,
    PRIMARY KEY (videoId, tagId),
    FOREIGN KEY (videoId) REFERENCES Videos(videoId)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (tagId) REFERENCES VideoTags(tagId)
        ON DELETE CASCADE ON UPDATE CASCADE
);

```

Result:

```

Database changed
mysql> SHOW TABLES
-> ;
+-----+
| Tables_in_test |
+-----+
| BunchCommentLike |
| BunchComments    |
| BunchContain     |
| BunchLike        |
| Bunches          |
| ChannellLike     |
| Channels         |
| CommentLike      |
| Comments         |
| HasTag           |
| Users           |
| VideoTags        |
| Videos          |
+-----+
13 rows in set (0.00 sec)

mysql> 

```

Inserting Rows into Tables:

```
DROP DATABASE IF EXISTS test;
CREATE DATABASE test;
USE test;

CREATE TABLE Channels (
    channelId      VARCHAR(32) PRIMARY KEY,
    channelName    VARCHAR(255)
);

CREATE TABLE Videos (
    videoId        VARCHAR(32) PRIMARY KEY,
    title          VARCHAR(255),
    publishedAt    DATETIME,
    channelId      VARCHAR(32),
    categoryId     INT,
    trendingDate   DATETIME,
    viewCount      INT,
    likes          INT,
    dislikes       INT,
    commentCount   INT,
    commentsDisabled BIT,
    ratingsDisabled BIT,
    description     TEXT,
    FOREIGN KEY (channelId) REFERENCES Channels(channelId)
);

CREATE TABLE VideoTags (
    tagId          INT PRIMARY KEY,
    tagName        VARCHAR(255)
);

CREATE TABLE HasTag (
    videoId        VARCHAR(32),
    tagId          INT,
    PRIMARY KEY (videoId, tagId),
    FOREIGN KEY (videoId) REFERENCES Videos(videoId)
        ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (tagId) REFERENCES VideoTags(tagId)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

Using python to generate more than 3000 lines of insertion.

```
build(2).sql
40      ON DELETE CASCADE ON UPDATE CASCADE
41  );
42
43
44
45  INSERT INTO Channels VALUES
46      ('UC-2MJlKSq9_pYk5-bdvMhnw', 'Denver7 - The Denver Channel'),
47      ('UC-2Y8Db0S6DtpXNgAKoJKA', 'PlayStation'),
48      ('UC-8Q-hLdECwQmalNwXitYDw', 'KatyPerryVEVO'),
49      ('UC-J-KZFRV8c13fOckhXdLiQ', 'Dua Lipa'),
50      ('UC-00324c10bi3H-U0bP77dw', 'SnoopDoggTV'),
51      ('UC-PbLusIBd3ECYtvz-f0EcQ', 'audiomack'),
52      ('UC-RxXi2Xws6Uk22vp-sLbGA', 'WCNC'),
53      ('UC-SV8-bUJfXjRmnp7F8Wzw', 'Roman Atwood Vlogs'),
54      ('UC-exjF2T70PJd8jA1050nkW', 'Wavy Boats'),
55      ('UC-1HJZR3Gqxm24_Vd_AJ5Yw', 'PewDiePie'),
56      ('UC-muTcRmDEEBTV8F8Fm82pA', 'The Schmo'),
57      ('UC-yXuc1_OzjwpsJPlxYUCQ', 'WORLDSTARHIPHOP'),
58      ('UC01ly8C-KtixvgSjrrBaRIQ', 'PrincessNokiaVEVO'),
59      ('UC07Kxew-cMIaykMOKzqHtBQ', 'LadyGagaVEVO'),
60      ('UC0BletW9phE4xHFM44q4qKA', 'Juice WRLD'),
61      ('UC0L1suV8pVg04pCAIBNGx5w', '60 Minutes Australia'),
62      ('UC0M0rxSz3IF0CsSour1iWmw', 'Cinemassacre'),
63      ('UC0WP5P-ufpRfjbNrmOWwLBQ', 'The Weeknd'),
64      ('UC0Wwu7r1ybaaR09ANhudTzA', 'Tampa Bay Buccaneers'),
65      ('UC0ZV6M2THA81QT9hrVWJG3A', 'Apex Legends'),
66      ('UC0ZsHHC_frpcqneDpMiteCQ', 'WXII 12 News'),
67      ('UC0biFgrMdkv1hFAFLVePODQ', 'Trixie Mattel'),
68      ('UC0dQFym1TdbTuj4pLZd80uQ', 'Hannah Meloche'),
69      ('UC0jPwNs4B7yJySNewHan5hQ', 'Optus Sport'),
70      ('UC0mUI4iBn1ptBZFj_mc35kg', 'The Perkins'),
71      ('UC0v-tlzs0QZwJnkiaUSJVQ', 'FBE'),
72      ('UC14UlmY1SNiQCB9Eookf_A', 'FC Barcelona'),
73      ('UC14d4SF8PHtsz2W4HJS12jw', 'Nasher'),
74      ('UC1ATY65YMBIF_wsaTgGa8Vw', 'Official Ed and Lorraine Warren'),
75      ('UC1DGEWjCYxjR96FtaI5GFbA', 'Young Merkules'),
76      ('UC1X3TRsCt36QPjF1p5f3HTg', 'LilYachtyVEVO'),
77      ('UC1Ydgfp2x8oLYG66KZHXs1g', 'TheTalko'),
78      ('UC1fjSSxPEEHb7WqAdd6xChQ', 'Hannah Daigle'),
79      ('UC1nJElGcVcTpeZJVyxEbZJw', 'Live Storms Media'),
80      ('UC1sELGmy5jp5fQUugmuYlXQ', 'Minecraft'),
81      ('UC2-0sEOYbQFuaURd_AU6Krg', 'OneFootball English'),
82      ('UC2-BeLxzUBSs0uSrmzWhJuQ', '20th Century Studios'),
83      ('UC21Kozr_K0yDM-VjoiHG9Aw', 'TheStradman'),
84      ('UC21yq4sq8uxTcfGIxxyE9VQ', 'Carli Bybel'),
85      ('UC22QP-rNb5wWYJ0BQOWlXQA', 'YuridiaVEVO'),
86      ('UC23ROAqt3prjnS89uHwcaaQ', 'CNCOVEVO'),
87      ('UC2CCXzC56k0b8L49R_iW-Yw', 'DaBaby'),
88      ('UC2C-fCh+L72Fhukm1anSV9w', 'GCB Group')
Ln 14, Col 34 Spaces: 4 UTF-8 LF SQL MSSQL Disconnected
```

build(2).sql

```
864
865
866 INSERT INTO VideoTags VALUES
867     (0, '#AEWDYNAMITE'),
868     (1, '#BTS'),
869     (2, '#BTSAD'),
870     (3, '#BTSCommercials'),
871     (4, '#BTSFilm'),
872     (5, '#BTSMusicVideo'),
873     (6, '#BTSpurplecar'),
874     (7, '#BTSxHyundai'),
875     (8, '#BTSxIONIQ'),
876     (9, '#BigHit'),
877     (10, '#CleanMobility'),
878     (11, '#Comedy #ComedyCentral #Atlanta #Hoodfights #Sitcom #'),
879     (12, '#Electric'),
880     (13, '#Electricvehicle'),
881     (14, '#FYP'),
882     (15, '#Future'),
883     (16, '#Hybrid'),
884     (17, '#Hyundai'),
885     (18, '#HyundaiElectric'),
886     (19, '#HyundaiIONIQ'),
887     (20, '#HyundaiPresents'),
888     (21, '#HyundaixBTS'),
889     (22, '#IONIQ'),
890     (23, '#IONIQxBTS'),
891     (24, '#Ioniqelectric'),
892     (25, '#Jimin'),
893     (26, '#Jin'),
894     (27, '#JungKook'),
895     (28, '#LyricVideo'),
896     (29, '#MV'),
897     (30, '#MusicVideo'),
898     (31, '#NBA'),
899     (32, '#NFLdropEA'),
900     (33, '#Official'),
901     (34, '#PlugIn'),
902     (35, '#Premium'),
903     (36, '#RM'),
904     (37, '#Review'),
905     (38, '#SUGA'),
906     (39, '#TikTok'),
907     (40, '#V'),
908     (41, '#bangtan'),
909     (42, '#bangtanboys'),
910     (43, '#brisxlife'),
911     (44, '#bts_twt'),
```

Ln 14, Col 34 Spaces: 4 UTF-8 LF SQL MSSQL Disconnected

build(2).sql

```
15857      (14990, '휴닝카이');
15858
15859
15860
15861  INSERT INTO Videos VALUES
15862      ('3C66w5Z0ixs', 'I ASKED HER TO BE MY GIRLFRIEND...', '2020-08-11 17:15:11',
15863
15864  FOLLOW ME ON SOCIAL
15865      ► Twitter: https://twitter.com/Brawadis
15866      ► Instagram: https://www.instagram.com/brawadis/
15867      ► Snapchat: brawadis
15868
15869  Hi! I'm Brandon Awadis and I like to make dope vlogs, pranks, re
15870      ('M9Pmf9AB4Mo', 'Apex Legends | Stories from the Outlands -
15871      ('J78aPJ3VyNs', 'I left youtube for a month and THIS is what
15872      ('kXLn3HkpjaA', 'XXL 2020 Freshman Class Revealed - Official
15873  XXL presents the 2020 XXL Freshman Class featuring NLE Choppa,
15874  Go here → http://www.xxlmag.com/
15875  XXL on Facebook → https://www.facebook.com/xxlmag
15876  XXL on Twitter → https://twitter.com/XXL
15877  XXL on Instagram → https://instagram.com/xxl/'),
15878      ('VIUo6yapDbc', 'Ultimate DIY Home Movie Theater for The Lab
15879      ('w-aidBdvZo8', 'I Haven't Been Honest About My Injury.. He
15880      ('uet14uf9NsE', 'OUR FIRST FAMILY INTRO!!', '2020-08-12 00:00:00',
15881      ('ua4QMFQATco', 'CGP Grey was WRONG', '2020-08-11 17:15:11',
15882      ('SnsPZj91R7E', 'SURPRISING MY DAD WITH HIS DREAM TRUCK!! |
15883      ('SsWHMAhshPQ', 'Ovi x Natanael Cano x Aleman x Big Soto - \
15884  Suscríbete: http://Rancho.lnk.to/RHYoutubeEscucha Vengo De Nada
15885      ('49Z6Mv4_WCA', 'i don't know what im doing anymore', '2020-08-11 17:15:11',
15886      ('nt3VVyv5pxQ', 'Try Not To Laugh Challenge #51', '2020-08-11 17:15:11',
15887      ('I6hswz4rIrU', 'Rainbow Six Siege: Operation Shadow Legacy
15888      ('W7VK4DUHvKU', 'Lil Yachty & Future - Pardon Me (Official V
15889      ('W9Aen8hG20Y', 'When Our Generation Gets Old and Hears a T
15890      ('BNeDH6UTmXw', 'Ten Minutes with Tyler Cameron | Q&A', '2020-08-11 17:15:11',
15891      ('6TIIsR_7nrNc', 'Kylie Jenner Reacts To "WAP" Music Video
15892      ('gPdUsIIndvVI', 'Our Farm Got Destroyed.', '2020-08-11 23:00:02',
15893      ('GTp-0S82guE', 'Time to Talk..', '2020-08-11 12:04:40', 'U
15894      ('jbGRowa5tIk', 'ITZY "Not Shy" M/V TEASER', '2020-08-11 15:00:00',
15895      ('vePc5V4h_kg', 'Shark Attack Test- Human Blood vs. Fish Blo
15896      ('5WjcDji3xYc', 'Honest Trailers | Avatar: The Last Airbende
15897      ('FopIxceEr8g', 'EXTREME Game of Hide and Seek in my NEW HO
15898      ('p7HGUZwQ_8s', 'Doing Doja Cat's Makeup!!', '2020-08-11 19:00:00',
15899      ('AMXT1okSUBg', 'THIS IS THE END.', '2020-08-11 23:00:02',
15900      ('mLOe7vGI0YI', 'Beerus VS Sailor Galaxia (Dragon Ball VS Sa
15901      ('Que4RnxYLNc', 'iann dior - Prospect ft. Lil Baby (Officia
15902      ('z5l8ovbw_6M', 'Don't be a Tourist', '2020-08-10 21:28:49',
15903      ('2avPJ9TZNmU', 'Juice WRLD & The Weeknd - Smile (Official V
15904      ('PT6hPvOqTYc', 'Being replaced.', '2020-08-10 20:56:04', 'U
```

build(2).sql

```
19205
19206
19207 INSERT INTO HasTag VALUES
19208     ('3C66w5Z0ixs', 5823),
19209     ('3C66w5Z0ixs', 11669),
19210     ('3C66w5Z0ixs', 5403),
19211     ('3C66w5Z0ixs', 12745),
19212     ('3C66w5Z0ixs', 8246),
19213     ('3C66w5Z0ixs', 8099),
19214     ('3C66w5Z0ixs', 14150),
19215     ('3C66w5Z0ixs', 14153),
19216     ('3C66w5Z0ixs', 2995),
19217     ('3C66w5Z0ixs', 5885),
19218     ('3C66w5Z0ixs', 6259),
19219     ('3C66w5Z0ixs', 5738),
19220     ('3C66w5Z0ixs', 7697),
19221     ('3C66w5Z0ixs', 7698),
19222     ('3C66w5Z0ixs', 10167),
19223     ('M9Pmf9AB4Mo', 521),
19224     ('M9Pmf9AB4Mo', 530),
19225     ('M9Pmf9AB4Mo', 10919),
19226     ('M9Pmf9AB4Mo', 526),
19227     ('M9Pmf9AB4Mo', 528),
19228     ('M9Pmf9AB4Mo', 522),
19229     ('M9Pmf9AB4Mo', 706),
19230     ('M9Pmf9AB4Mo', 3779),
19231     ('M9Pmf9AB4Mo', 533),
19232     ('M9Pmf9AB4Mo', 531),
19233     ('M9Pmf9AB4Mo', 3606),
19234     ('M9Pmf9AB4Mo', 709),
19235     ('M9Pmf9AB4Mo', 708),
19236     ('M9Pmf9AB4Mo', 710),
19237     ('M9Pmf9AB4Mo', 520),
19238     ('M9Pmf9AB4Mo', 12977),
19239     ('M9Pmf9AB4Mo', 10678),
19240     ('M9Pmf9AB4Mo', 525),
19241     ('M9Pmf9AB4Mo', 529),
19242     ('M9Pmf9AB4Mo', 524),
19243     ('M9Pmf9AB4Mo', 523),
19244     ('M9Pmf9AB4Mo', 3607),
19245     ('M9Pmf9AB4Mo', 1537),
19246     ('M9Pmf9AB4Mo', 4256),
19247     ('M9Pmf9AB4Mo', 8067),
19248     ('J78aPJ3VyNs', 9111),
19249     ('J78aPJ3VyNs', 8075),
19250     ('J78aPJ3VyNs', 8088),
19251     ('J78aPJ3VyNs', 10352),
19252     ('J78aPJ3VyNs', 9113),
```

```

->
-> ('EpwKNzxH94c', 8304),
-> ('EpwKNzxH94c', 12311),
-> ('EpwKNzxH94c', 6682),
-> ('EpwKNzxH94c', 14451),
-> ('EpwKNzxH94c', 11557),
-> ('EpwKNzxH94c', 11568),
-> ('EpwKNzxH94c', 11489),
-> ('EpwKNzxH94c', 11842),
-> ('EpwKNzxH94c', 13152),
-> ('EpwKNzxH94c', 8646);
Query OK, 21987 rows affected (0.84 sec)
Records: 21987 Duplicates: 0 Warnings: 0

mysql>
mysql>
mysql>
mysql>
mysql>
mysql>

```

Count queries:

SELECT COUNT(*) AS VideoTagsCount FROM VideoTags;

SELECT COUNT(*) AS VideosCount FROM Videos;

SELECT COUNT(*) AS HasTagCount FROM HasTag;

```

mysql>
mysql> SELECT COUNT(*) AS VideoTagsCount FROM VideoTags;
+-----+
| VideoTagsCount |
+-----+
|           14991 |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> SELECT COUNT(*) AS VideosCount FROM Videos;
+-----+
| VideosCount |
+-----+
|           1207 |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> SELECT COUNT(*) AS HasTagCount FROM HasTag;
+-----+
| HasTagCount |
+-----+
|           21987 |
+-----+
1 row in set (0.01 sec)

mysql>
mysql>

```

Two Advanced SQL Queries:

```
--Q1
SELECT DISTINCT tagName,
                  COUNT(videoId) AS appearTimes
FROM    Videos NATURAL JOIN HasTag
        NATURAL JOIN VideoTags
WHERE   viewCount >= 100
        AND likes / viewCount >= 0.001
        AND dislikes / viewCount <= 0.1
        AND commentCount / viewCount >= 0.0001
GROUP BY tagId
ORDER BY appearTimes DESC
LIMIT 15;

--Q2
SELECT DISTINCT title
FROM    Videos NATURAL JOIN BunchContain
WHERE   bunchId IN (
    SELECT bunchId
    FROM    BunchContain
    GROUP BY bunchId
    HAVING  COUNT(videoId) >= ALL(
        SELECT COUNT(videoId)
        FROM    BunchContain
        GROUP BY bunchId
    )
)
LIMIT 15;
```

Q1: This query is designed to retrieve the top 15 video tags that meet designed criteria:

LIMIT 15 clause limits the output to the top 15 rows, effectively showing the top 15 video tags that meet the specified criteria.

```
+-----+-----+
| tagName | appearTimes |
+-----+-----+
| [None]  |          118 |
| funny   |           57 |
| comedy  |           43 |
| NBA     |           43 |
| 2020    |           35 |
| news    |           34 |
| Basketball |          32 |
| G League |          31 |
| animation |          28 |
| vlog    |          28 |
| Disney  |          27 |
| NBA Playoffs |          26 |
| rap     |          25 |
| Highlights |          23 |
| 2019-20 |          22 |
+-----+-----+
15 rows in set (0.08 sec)
```

Q2: This query is designed to retrieve the distinct titles of videos that belong to the bunch with the highest number of videos:

LIMIT 15 clause limits the output to upper 15 rows of that bunch.

```
->
-> LIMIT 15;
+-----+-----+
| title |
+-----+-----+
| What Every Country in Europe Is WORST At Part 1 |
| I Made an Edwardian Wrap Cape...and American Duchess Made Shoes to Match! |
| Rod Wave's 2020 XXL Freshman Freestyle |
| Portsmouth Vice Mayor served summons after calling for police chief's resignation |
| Who will Lionel Messi sign for and how much will he cost? | The Transfer Show |
| Bride Has Unlimited Budget For Her Royal Kentucky Wedding Dress | Say Yes To The Dress |
| Borderlands 3 - Psycho Krieg and the Fantastic Fustercluck Official Reveal Trailer |
| Ammonite - Official Trailer |
| Tame Impala: Tiny Desk (Home) Concert |
| Honest Trailers | Mean Girls |
| Ashe - Save Myself (Official Video) |
| Morgan Wallen - More Than My Hometown (Official Music Video) |
| HOTBOII Feat. Lil Baby Don't Need Time (Remix) (Official Video) |
| NBA 2K21 LEBRON JAMES BUILD - 61 BADGE UPGRADES - DEMIGOD SMALL FORWARD BUILD |
| Fudgy, the Golden Retriever Puppy vs Slow Feeder |
+-----+-----+
15 rows in set (0.00 sec)

mysql> 
```

Indexing:

Q1:

```
EXPLAIN ANALYZE
SELECT DISTINCT tagName,
                COUNT(videoId) AS appearTimes
FROM    Videos NATURAL JOIN HasTag
        NATURAL JOIN VideoTags
WHERE   viewCount >= 100
        AND likes / viewCount >= 0.001
        AND dislikes / viewCount <= 0.1
        AND commentCount / viewCount >= 0.0001
GROUP BY tagName
ORDER BY appearTimes DESC
LIMIT 15;
```

```
CREATE INDEX idx_viewcount ON Videos (viewCount);
```

[illegible]

```
CREATE INDEX idx_likes_viewcount ON Videos (likes, viewCount);
```

```

-----+
| -> Limit: 15 row(s) (actual time=72.538..72.540 rows=15 loops=1)
   -> Sort with duplicate removal: appearTimes DESC, VideoTags.tagName (actual time=72.537..72.538 rows=15 loops=1)
       -> Table scan on <temporary> (actual time=57.686..60.183 rows=14587 loops=1)
           -> Aggregate using temporary table (actual time=57.681..57.681 rows=14587 loops=1)
               -> Nested loop inner join (cost=9274.96 rows=19257) (actual time=0.112..40.449 rows=21567 loops=1)
                   -> Nested loop inner join (cost=2535.13 rows=19257) (actual time=0.104..17.076 rows=21567 loops=1)
                       -> Filter: ((Videos.viewCount >= 100) and ((Videos.likes / Videos.viewCount) >= 0.001) and ((Videos
.dislikes / Videos.viewCount) <= 0.1) and ((Videos.commentCount / Videos.viewCount) >= 0.0001)) (cost=162.15 rows=1207) (a
ctual time=0.084..1.772 rows=1179 loops=1)
                           -> Table scan on Videos (cost=162.15 rows=1219) (actual time=0.078..0.871 rows=1207 loops=1)
                               -> Covering index lookup on HasTag using PRIMARY (videoId=Videos.videoId) (cost=0.37 rows=16) (act
ual time=0.006..0.012 rows=18 loops=1179)
                                   -> Single-row index lookup on VideoTags using PRIMARY (tagId=HasTag.tagId) (cost=0.25 rows=1) (actual
time=0.001..0.001 rows=1 loops=21567)
                                       |
                                           +-----+
-----+
1 row in set (0.08 sec)

```

CREATE INDEX idx_dislikes_viewcount ON Videos (dislikes, viewCount);

```

-----+
| -> Limit: 15 row(s) (actual time=78.383..78.385 rows=15 loops=1)
   -> Sort with duplicate removal: appearTimes DESC, VideoTags.tagName (actual time=78.382..78.383 rows=15 loops=1)
       -> Table scan on <temporary> (actual time=61.576..64.553 rows=14587 loops=1)
           -> Aggregate using temporary table (actual time=61.574..61.574 rows=14587 loops=1)
               -> Nested loop inner join (cost=9274.96 rows=19257) (actual time=0.124..42.398 rows=21567 loops=1)
                   -> Nested loop inner join (cost=2535.13 rows=19257) (actual time=0.116..17.892 rows=21567 loops=1)
                       -> Filter: ((Videos.viewCount >= 100) and ((Videos.likes / Videos.viewCount) >= 0.001) and ((Videos
.dislikes / Videos.viewCount) <= 0.1) and ((Videos.commentCount / Videos.viewCount) >= 0.0001)) (cost=162.15 rows=1207) (a
ctual time=0.084..1.880 rows=1179 loops=1)
                           -> Table scan on Videos (cost=162.15 rows=1219) (actual time=0.075..0.931 rows=1207 loops=1)
                               -> Covering index lookup on HasTag using PRIMARY (videoId=Videos.videoId) (cost=0.37 rows=16) (act
ual time=0.007..0.012 rows=18 loops=1179)
                                   -> Single-row index lookup on VideoTags using PRIMARY (tagId=HasTag.tagId) (cost=0.25 rows=1) (actual
time=0.001..0.001 rows=1 loops=21567)
                                       |
                                           +-----+
-----+
1 row in set (0.09 sec)

```

CREATE INDEX idx_commentcount_viewcount ON Videos (commentCount, viewCount);

Analyses:

idx_likes_viewcount on Videos (likes, viewCount): This is a combined index that considers the query criteria for both the likes and viewCount columns, such as likes/viewCount ≥ 0.001 .

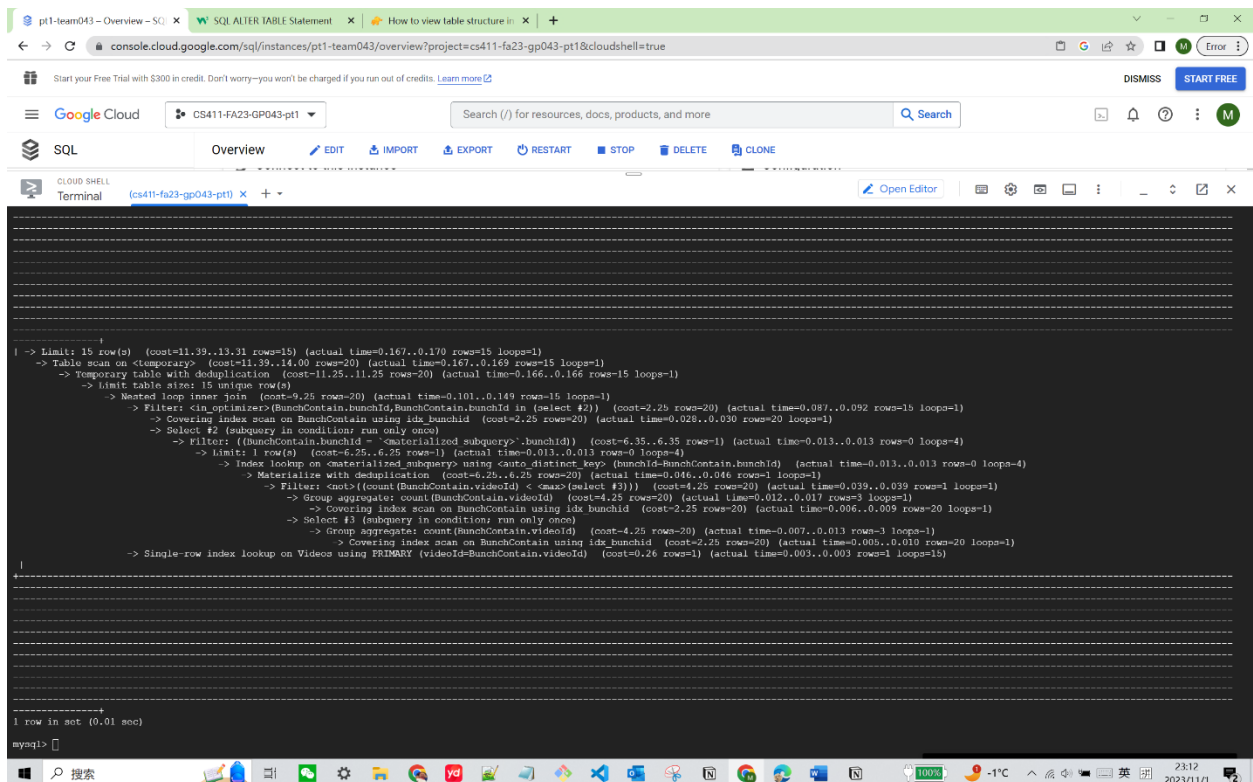
idx_commentcount_viewcount on Videos (commentCount, viewCount): This combined index is suitable for queries that consider both commentCount and viewCount columns, such as commentCount/viewCount >= 0.0001.

CREATE INDEX idx_likes_viewcount ON Videos (likes, viewCount); is the best, maybe that's because the condition of like is more general.

Q2:

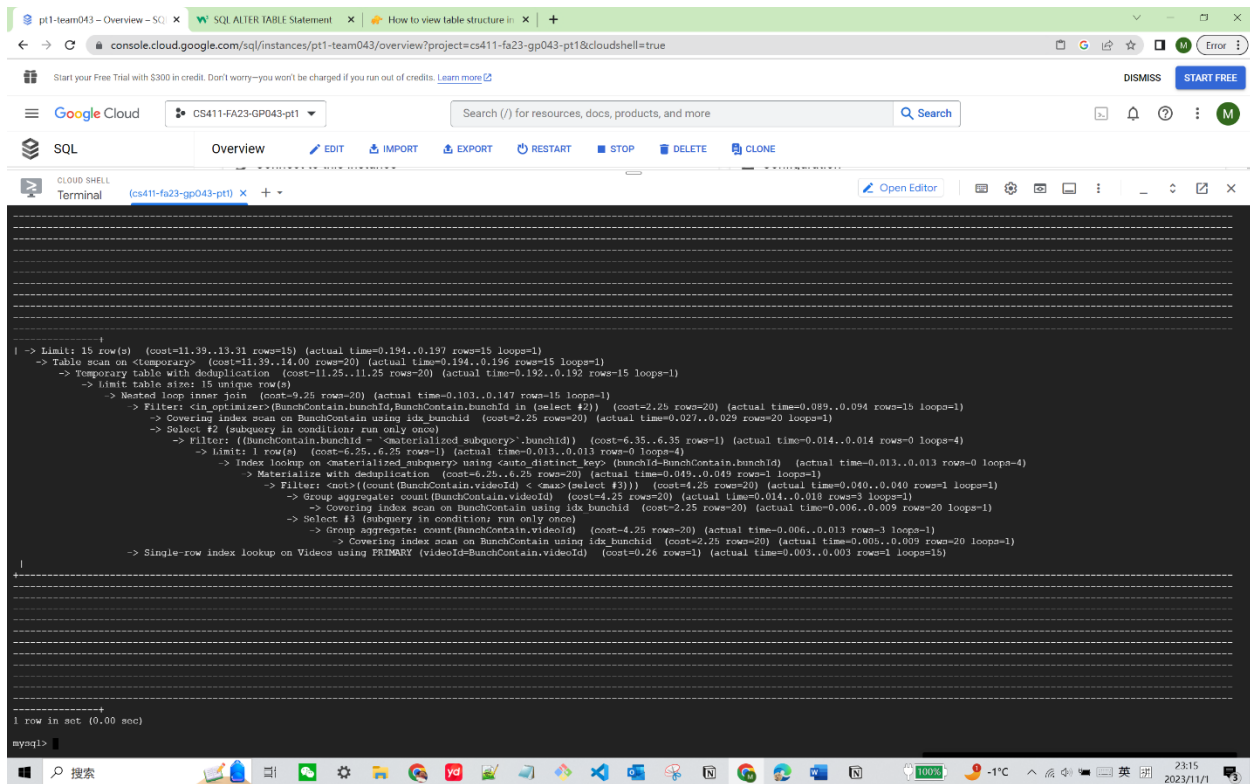
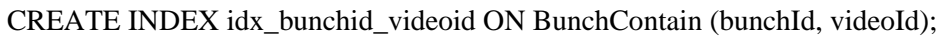
```
EXPLAIN ANALYZE
SELECT DISTINCT title
FROM   Videos NATURAL JOIN BunchContain
WHERE  bunchId IN (
    SELECT bunchId
    FROM   BunchContain
    GROUP BY bunchId
    HAVING COUNT(videoId) >= ALL(
        SELECT COUNT(videoId)
        FROM   BunchContain
        GROUP BY bunchId
    )
)
LIMIT 15;
```

CREATE INDEX idx_bunchid ON BunchContain (bunchId);



```
pt1-team043 - Overview - SQL | SQL ALTER TABLE Statement | How to view table structure in | +
console.cloud.google.com/sql/instances/pt1-team043/overview?project=cs411-fa23-gp043-pt1&cloudshell=true
Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more
Google Cloud CS411-FA23-GP043-pt1 Search (/) for resources, docs, products, and more
SQL Overview EDIT IMPORT EXPORT RESTART STOP DELETE CLONE
CLOUD SHELL Terminal (cs411-fa23-gp043-pt1) X + Open Editor
I -> Limit: 15 row(s) (cost=11.39..13.31 rows=15) (actual time=0.167..0.170 rows=15 loops=1)
-> Table scan on <temporary> (cost=11.39..14.00 rows=20) (actual time=0.167..0.169 rows=15 loops=1)
-> Temporary table with deduplication (cost=11.25..11.25 rows=20) (actual time=0.166..0.166 rows=15 loops=1)
-> Limit table size: 15 unique row(s)
-> Nested loop inner join (cost=9.25 rows=20) (actual time=0.101..0.149 rows=15 loops=1)
-> Filter: <in optimizer> (BunchContain.bunchId, BunchContain.bunchId in (select #2)) (cost=2.25 rows=20) (actual time=0.087..0.092 rows=15 loops=1)
-> Covering index scan on BunchContain using idx_bunchid (cost=2.25 rows=20) (actual time=0.029..0.030 rows=20 loops=1)
-> Select #2 (subquery in condition; run only once)
-> Filter: ((BunchContain.bunchid = <materialized subquery> bunchid) (cost=6.35..6.35 rows=1) (actual time=0.013..0.013 rows=0 loops=4)
-> Limit: 1 row(s) (cost=6.25..6.25 rows=1) (actual time=0.013..0.013 rows=0 loops=4)
-> Index lookup on <materialized subquery> using <auto distinct key> (bunchid=BunchContain.bunchid) (actual time=0.013..0.013 rows=0 loops=4)
-> Materialize with deduplication (cost=6.25..6.25 rows=20) (actual time=0.046..0.046 rows=1 loops=1)
-> Filter: <not> ((count(BunchContain.videoId) < <max> (select #3))) (cost=1.25 rows=20) (actual time=0.039..0.039 rows=1 loops=1)
-> Group aggregate: count(BunchContain.videoId) (cost=4.25 rows=20) (actual time=0.012..0.017 rows=3 loops=1)
-> Covering index scan on BunchContain using idx_bunchid (cost=2.25 rows=20) (actual time=0.006..0.009 rows=20 loops=1)
-> Select #3 (subquery in condition; run only once)
-> Group aggregate: count(BunchContain.videoId) (cost=4.25 rows=20) (actual time=0.007..0.013 rows=3 loops=1)
-> Covering index scan on BunchContain using idx_bunchid (cost=2.25 rows=20) (actual time=0.005..0.010 rows=20 loops=1)
-> Single-row index lookup on Videos using PRIMARY (videoId=BunchContain.videoId) (cost=0.26 rows=1) (actual time=0.003..0.003 rows=1 loops=15)
1 row in set (0.01 sec)
mysql>
```

CREATE INDEX idx_videoId ON BunchContain (videoId);



Analyses:

idx_bunchid on BunchContain (bunchId): This index is used to speed up queries on bunchId columns in the BunchContain table. It is often helpful for querying and filtering Bunchid-based data.

idx_videoid on BunchContain (videoId): This index is used to speed up queries on the videoId column in the BunchContain table. It is often helpful for querying and filtering Videoid-based data.

idx_bunchid_videoid on BunchContain (bunchId, videoId): This is a combined index that covers the bunchId and videoId columns. This index is suitable for situations where both bunchId and videoId columns are involved in a query, because it can locate data that meets both criteria more quickly.

Considering actual time they consumed,

CREATE INDEX idx_videoid ON BunchContain (videoId); is the best. Because videoId appears more in having conditions.