

Trabajo Nociones de Arquitectura de la Información

Versión: 2021.05.23 05:00

Observación: Cada vez que agregue nuevos elementos al documento, o que modifique algún componente del informe, revise la coherencia y consistencia con los otros elementos que hacen parte del mismo.

RESPONSABLES

Nombre Completo – Documento de Identificación
1. Andrea Stephanya Pino Villegas - C.C 1.144.158.067
2. Daniela Rivera Vargas - C.C. 1.037.636.542
3. Daniel Escobar Ruiz - C.C. 1.040.747.009
4. Edward Darío Mercado Gómez - C.C. 1.038.131.004
5. Simon Pineda Agudelo - C.C. 1.152.220.029
REPO EN GITHUB: https://github.com/skailuxspa-unal/agd-trabajo-1

Realiza este trabajo considerando los datos que generan los sistemas transaccionales e información no estructurada de tu dominio (si trabajas por ejemplo para TCC tu dominio es la mensajería; también puedes explorar en la página <https://www.kaggle.com/datasets> o <https://arxiv.org/>). Considera tener acceso a esta información, de al menos 10 MB (puede ser uno o varios archivos de texto), y **tener al menos cuatro clases conceptuales. Este documento también debe almacenarse en el REPO. Plazo Máximo de Entrega 23 de Mayo, NO SE recibirá por correo electrónico, envíalo por <https://forms.gle/h7ty3yZykaUq5m7y6>**

1 COMPRENSIÓN DEL NEGOCIO

1.1 DESCRIPCIÓN DEL CONTEXTO DEL NEGOCIO.

Describe en máximo 250 palabras el contexto en el cual se generan los datos y cuál es el proceso que los genera.

Somos una agencia de consultoría que ofrece soluciones tecnológicas basadas en el análisis de datos e información para la toma de decisiones inteligentes. En este proyecto en particular, vamos a estar apoyando a un cliente que se encuentra en la industria de la comercialización de artículos electrónicos. Este se encuentra interesado en realizar pautas publicitarias en la red social youtube con target en los

Estados Unidos de América. Por lo tanto, los datos se obtienen de la plataforma de streaming de videos teniendo en cuenta el número de vistas, likes, dislikes, temática, comentarios y categoría del video. Los datos usados fueron extraídos en parte de Kaggle y fueron complementados utilizando la API v3 de YouTube.

Categorías: https://www.kaggle.com/datasnaek/youtube-new?select=US_category_id.json

Datos: <https://www.kaggle.com/datasnaek/youtube-new?select=USvideos.csv>

1.2 IDENTIFICACIÓN DEL PROBLEMA:

Delimite en máximo 150 palabras la problemática, así como identificar los requisitos, supuestos, restricciones y beneficios de la solución de este.

Tomando como referencia las restricciones presupuestales y recursos tecnológicos de nuestro clientes se debe identificar, cuales son los videos con mas reproducciones y sus respectivas categorías para definir cuales son los videos en donde las pautas publicitarias de nuestro cliente traerian mas beneficio a través de su canal de ventas e-commerce, introducción y posicionamiento de marca.

1.3 DETERMINACIÓN DE OBJETIVOS:

Describe en máximo 150 palabras las metas a lograr al proponer una solución basada en un modelo de datos o de analítica (cómo y qué tipo de ventaja competitiva se ganará).

Mediante la estructuración, definición y estudio del modelo de datos de la presente base de datos se logrará ofrecer al cliente una clara estrategia publicitaria a través de la plataforma Youtube, que permitirá obtener mejores resultados. Estos a su vez y como finalidad, se traducirán en mayores ventas para nuestro cliente en específico.

1.4 EVALUACIÓN DE LA SITUACIÓN ACTUAL:

Describe en máximo 150 palabras el estado actual antes de implementar la solución de analítica, a fin de tener un punto de comparación que permita medir el grado de éxito de la solución.

Actualmente, contamos con una base de datos de 20MB. Los datos no se encuentran estructurados, ni tampoco se encuentran en un orden establecido. Por lo tanto, se desconoce cuál sería la mejor estrategia de publicidad visual en la plataforma youtube para nuestro cliente.

2 COMPRENSIÓN DE LOS DATOS

2.1 RECOLECCIÓN DE DATOS

Describe en máximo 150 palabras los datos a utilizar identificando las fuentes, las técnicas empleadas en su recolección, los problemas encontrados en su obtención y la forma como se resolvieron los mismos. Además, adjunte los datos (archivos de texto, etc.) agréguelos en el github (**REPO EN GITHUB**) en un solo archivo, por favor comprímalo(s). Llame el archivo T1.2.1.Datos.zip

Los datos recopilados representan videos de YouTube que estuvieron en trending en 2018. Esto incluye títulos, categorías, número de likes, dislikes, entre otros. Dichos datos fueron obtenidos de Kaggle, sin embargo se decidió complementarlos utilizando el API de YouTube para descargar 20 comentarios por cada video, esto con el fin de entender, grosso modo, qué opiniones tiene la gente de cada video.

2.2 DESCRIPCIÓN DE DATOS (DICCIONARIO):

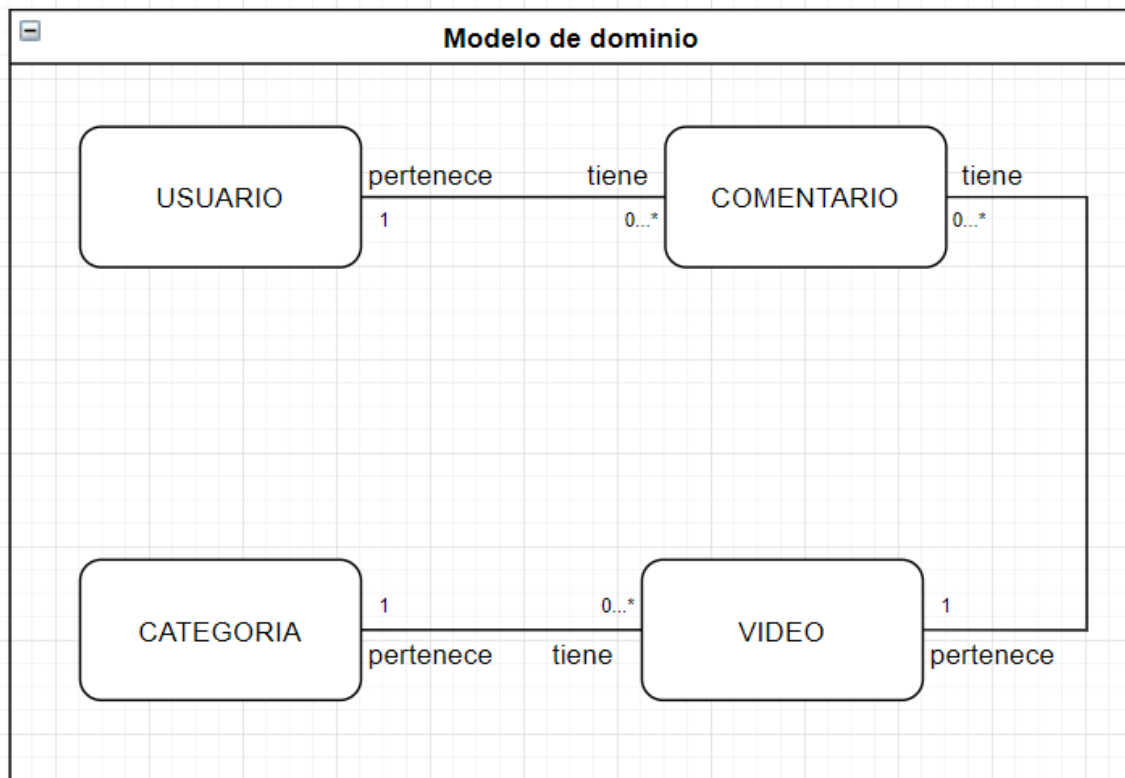
Diligencia la siguiente tabla, puede agregar otra columna si lo considera necesario.

Nombre del atributo / variable	Clase conceptual	Formato o Tipo de Dato	Descripción
category_id	categoría video	Int	Identificador numérico de la categoría.
category	categoría	string	Nombre de la categoría.
viewer_id	usuario comentario	string	Identificador alfanumérico del canal dueño del video.
viewer_display_name	usuario	string	Nombre del autor de un comentario.
video_id	comentario video	string	Identificador alfanumérico del vídeo.
comment_id	comentario		
comment_text	comentario	string	Texto del comentario.
like_count	comentario	Int	Número de pulgares arriba de un comentario.
reply_count	comentario	Int	Número de respuestas que tiene un comentario.
publish_time	video	date	Tiempo de publicación del video.<
channel_title	video	string	Nombre del canal al que pertenece el vídeo.
Title	video	string	Título del vídeo.
Description	video	string	Descripción del vídeo.
trending_date	video	date	Fecha en la cual el video entró a trending.
thumbnail_link	video	string	Link a la imagen de la miniatura del video.
comments_disabled	video	bool	Indica si el video tiene los comentarios deshabilitados.

ratings_disabled	video	bool	Indica si el video tiene las calificaciones desactivadas.
video_error_or_removed	video	bool	Indica si el video presenta error o ha sido removido.
Views	video	Int	Número de veces que ha sido visto el vídeo.
Likes	video	Int	Número de pulgares arriba del vídeo.
Dislikes	video	Int	Número de pulgares abajo del vídeo.
comment_count	video	Int	Número de comentarios que tiene el video.
Tags	video	string	Los tags que tiene el video.

2.3 MODELO DEL DOMINIO

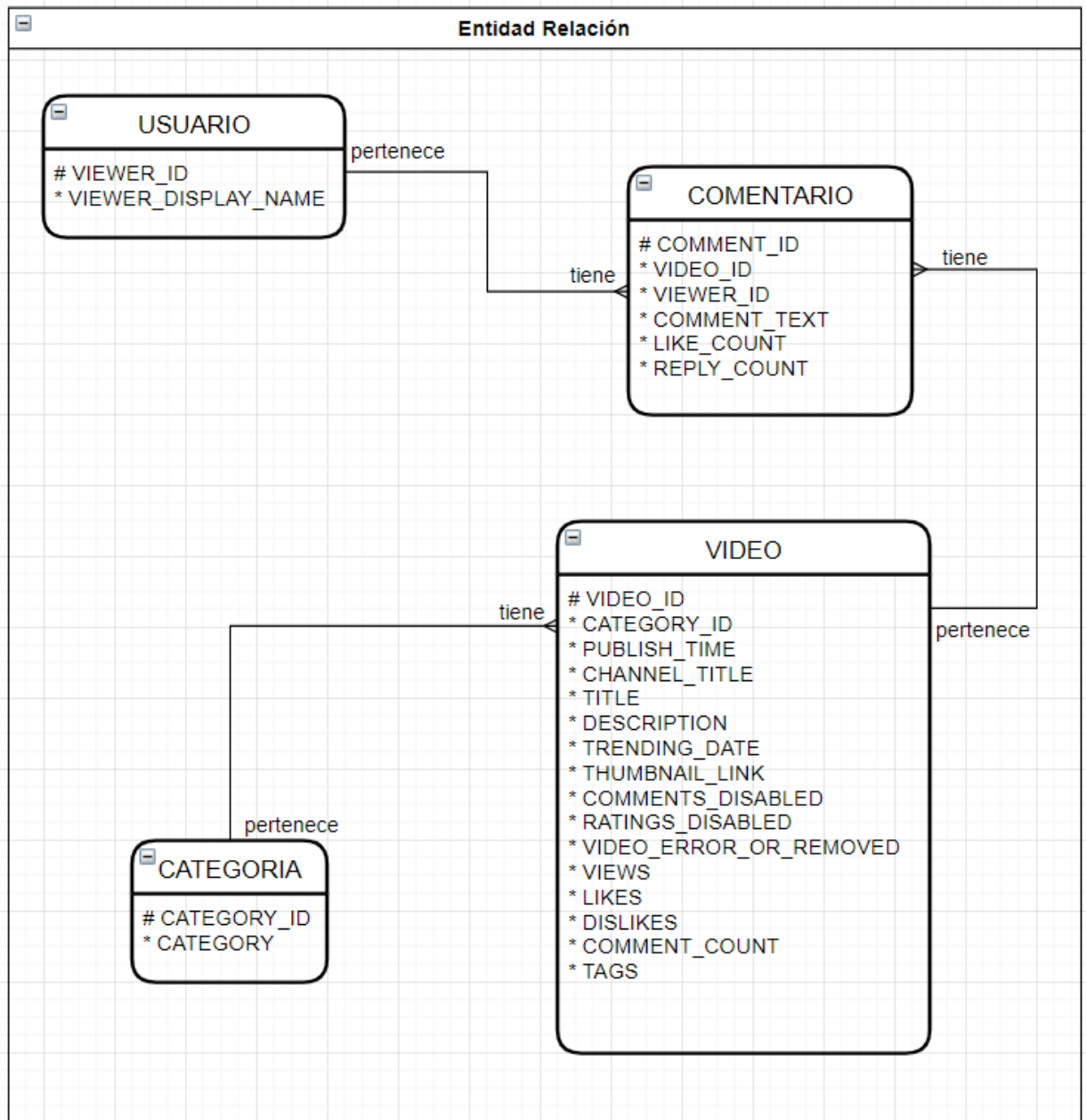
Observación: Incluya el gráfico del modelo del dominio que representa la estructura de datos de su problema.



3 MODELO ENTIDAD-RELACIÓN

3.1 TOMA DE PANTALLA DEL MODELO E-R

Observación: lo que se pide, puede usar <https://draw.io> o Microsoft Visio® y modele usando la notación de Barker.



3.2 SENTENCIA O CONSULTA DE CREACIÓN DEL TABLA(S)

Observación: Escriba el código en el Sistema de Gestión de Bases de Datos Relacionales de su elección (se recomienda SQLite por simplicidad, mediante <https://sqlitebrowser.org/>) para crear las tablas que corresponda con su conjunto de datos específico. Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de T1.3.2.Creacion_Tablas.sql

Se recomienda repasar SQL en <https://www.w3schools.com/sql/default.asp>

Database Structure		
Browse Data Edit Pragma Execute SQL		
Create Table Create Index Print		
Name	Type	Schema
Tables (4)		
> categorias		CREATE TABLE "categorias" ("category_id" INTEGER NOT NULL UNIQUE, "category" VARCHAR(25) NOT NULL, PRIMARY KEY("category_id"))
> comentarios		CREATE TABLE "comentarios" ("comment_id" INTEGER NOT NULL UNIQUE, "video_id" VARCHAR(15) NOT NULL, "viewer_id" VARCHAR(25) NOT NULL, "comment_text" TEXT NOT NULL, "like_cou
> usuarios		CREATE TABLE "usuarios" ("viewer_id" VARCHAR(25) NOT NULL UNIQUE, "viewer_display_name" TEXT, PRIMARY KEY("viewer_id"))
> videos		CREATE TABLE "videos" ("video_id" VARCHAR(15) NOT NULL UNIQUE, "category_id" INTEGER NOT NULL, "publish_time" VARCHAR(20) NOT NULL, "channel_title" VARCHAR(50) NOT NULL, "title"
Indices (0)		
Views (0)		
Triggers (0)		

?

×

Edit table definition

Table

categorias

▼

Advanced

Fields

Constraints

Add

Remove

Move to top

Move up

Move down

Move to bottom

Name	Type	NN	PK	AI	U	Default
category_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
category	VARCHAR(25)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

<

>

1

[-]

CREATE TABLE "categorias" (

2

"category_id"

INTEGER NOT NULL UNIQUE,

3

"category"

VARCHAR(25) NOT NULL,

4

PRIMARY KEY("category_id")

5

);

Edit table definition?×

Table

comentarios

▼ Advanced

Fields

Constraints

Add

Remove

Move to top

Move up

Move down

Move to bottom

Name	Type	NN	PK	AI	U	Default
comment_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
video_id	VARCHAR(15)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
viewer_id	VARCHAR(25)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
comment_text	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
1 CREATE TABLE "comentarios" (  
2     "comment_id"    INTEGER NOT NULL UNIQUE,  
3     "video_id"  VARCHAR(15) NOT NULL,  
4     "viewer_id" VARCHAR(25) NOT NULL,  
5     "comment_text" TEXT NOT NULL,  
6     "like_count"   INTEGER NOT NULL,  
7     "reply_count"  INTEGER NOT NULL,  
8     PRIMARY KEY("comment_id")  
9 );
```


Edit table definition?×

Table

usuarios

▼ Advanced

Fields

Constraints

Add

Remove

Move to top

Move up

Move down

Move to bottom

Name	Type	NN	PK	AI	U	Default
viewer_id	<div>VARCHAR(25) ▼</div>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
viewer_display_name	<div>TEXT ▼</div>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

<

>

1

2

3

4

5

CREATE TABLE "usuarios" (
"viewer_id" VARCHAR(25) NOT NULL UNIQUE,
"viewer_display_name" TEXT,
PRIMARY KEY("viewer_id")
);

Edit table definition?×

Table

videos

▼ Advanced

Fields

Constraints

Add

Remove

Move to top

Move up

Move down

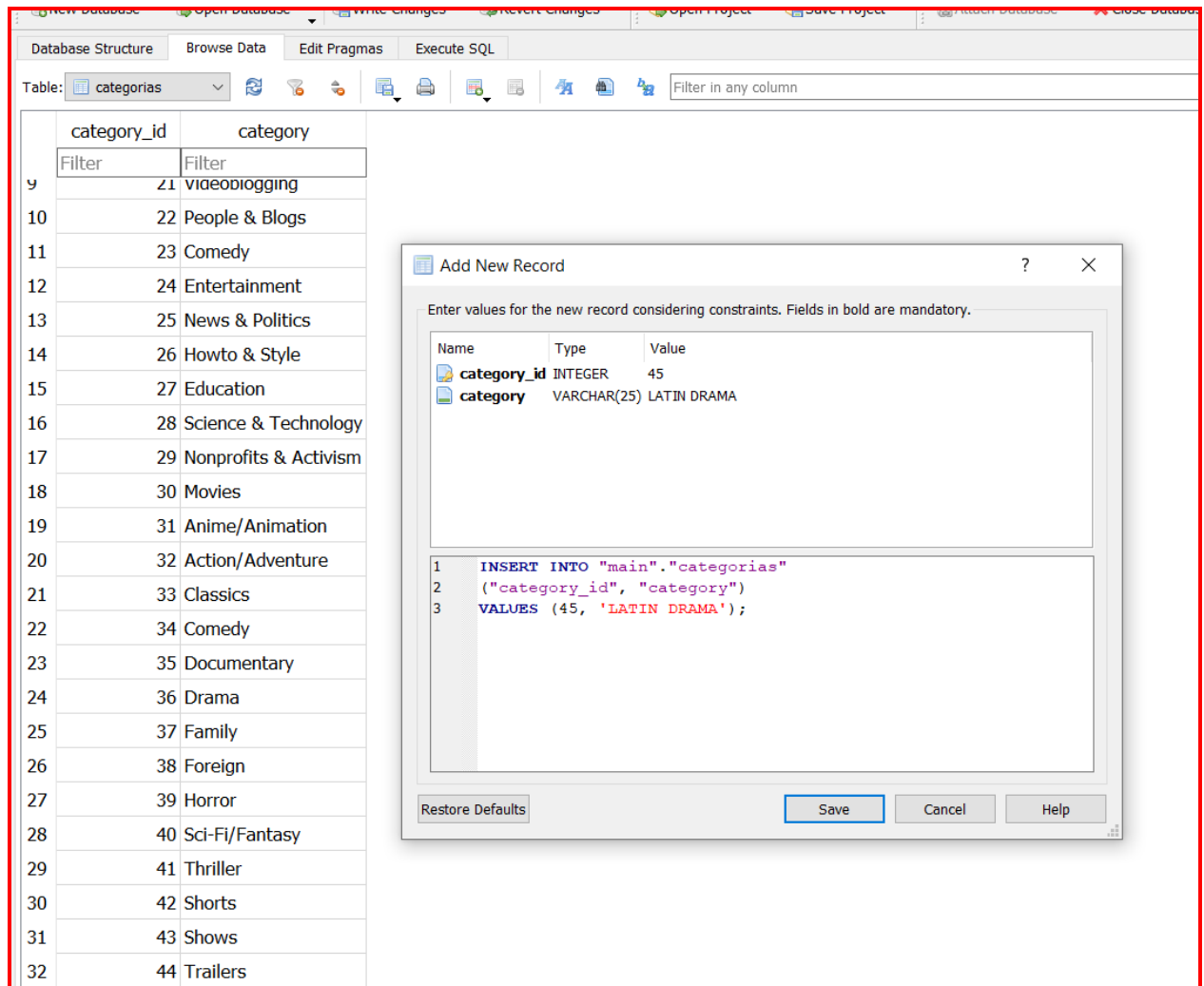
Move to bottom

Name	Type	NN	PK	AI	U	Default
video_id	VARCHAR(15) ▾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
category_id	INTEGER ▾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
publish_time	VARCHAR(20) ▾	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
channel_title	VARCHAR(50) ▾	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
title	VARCHAR(150) ▾	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
description	TEXT ▾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
trending_date	VARCHAR(10) ▾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
1 CREATE TABLE "videos" (  
2     "video_id" VARCHAR(15) NOT NULL UNIQUE,  
3     "category_id" INTEGER NOT NULL,  
4     "publish_time" VARCHAR(20) NOT NULL,  
5     "channel_title" VARCHAR(50) NOT NULL,  
6     "title" VARCHAR(150) NOT NULL,  
7     "description" TEXT,  
8     "trending_date" VARCHAR(10),  
9     "thumbnail_link" TEXT,  
10    "comments_disabled" BIT,  
11    "ratings_disabled" BIT,  
12    "video_error_or_removed" BIT,  
13    "views" INTEGER,  
14    "likes" INTEGER,  
15    "dislikes" INTEGER,  
16    "comment_count" INTEGER,  
17    "tags" TEXT,  
18    PRIMARY KEY("video_id","category_id")  
19 );
```

3.3 SENTENCIAS PARA INSERTAR DATOS

Observación: Escriba el código para insertar los datos en cada una de las tablas creadas. Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de *T1.3.3.Insertar_Datos.sql*



The screenshot shows a database management interface with a table named 'categorias' and an 'Add New Record' dialog box.

Table: categorias

category_id	category
21	videoblogging
22	People & Blogs
23	Comedy
24	Entertainment
25	News & Politics
26	Howto & Style
27	Education
28	Science & Technology
29	Nonprofits & Activism
30	Movies
31	Anime/Animation
32	Action/Adventure
33	Classics
34	Comedy
35	Documentary
36	Drama
37	Family
38	Foreign
39	Horror
40	Sci-Fi/Fantasy
41	Thriller
42	Shorts
43	Shows
44	Trailers

Add New Record Dialog:

Enter values for the new record considering constraints. Fields in bold are mandatory.

Name	Type	Value
category_id	INTEGER	45
category	VARCHAR(25)	LATIN DRAMA

SQL Statement:

```
1 INSERT INTO "main"."categorias"
2 ("category_id", "category")
3 VALUES (45, 'LATIN DRAMA');
```

Buttons: Restore Defaults, Save, Cancel, Help

3.4 SENTENCIA DE CONSULTA

Observación: realice la exploración básica de los datos, conteos totales y por categorías, máximos, promedio y mínimos. Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Use solamente sentencias SQL. Anexe las tomas de pantalla donde evidencie la sentencia SQL y su correspondiente ejecución. Además, Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de *T1.3.4.Consultar_Datos.sql*

Videos con máximo de views por categoría.

SQL 1 SQL 2 SQL 3

```
1 SELECT videos.category_id,
2     videos.title,
3     MAX(videos.views) AS views,
4     videos.likes,
5     videos.dislikes,
6     categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC
7
```

	category_id	title	views	likes	dislikes	category
1	10	BTS (방탄소년단) 'FAKE LOVE' Official MV	103786733	5386959	190567	Music
2	24	Marvel Studios' Avengers: Infinity War Official ...	91933007	2625661	53709	Entertainment
3	22	To Our Daughter	51243149	0	0	People & Blogs
4	23	Poo Bear ft. Justin Bieber & Jay Electronica - Har...	29178096	825258	15353	Comedy
5	26	42 HOLY GRAIL HACKS THAT WILL SAVE YOU A ...	26221058	206111	39209	Howto & Style
6	2	Official Ram Trucks Super Bowl Commercial ...	25193150	6811	1272	Autos & Vehicles

Execution finished without errors.
Result: 16 rows returned in 74ms
At line 1:
SELECT videos.category_id,
videos.title,
MAX(videos.views) AS views,
videos.likes,
videos.dislikes,
categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC

Videos con máximo de likes por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1 SQL 2 SQL 3

```
1 SELECT videos.category_id,
2     videos.title,
3     MAX(videos.likes) AS likes,
4     videos.views,
5     videos.dislikes,
6     categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY cate
7
```

	category_id	title	likes	views	dislikes	category
1	10	BTS (방탄소년단) 'FAKE LOVE' Official MV	5386959	103786733	190567	Music
2	24	Marvel Studios' Avengers: Infinity War Official ...	2625661	91933007	53709	Entertainment
3	22	we broke up	1967904	31505838	96814	People & Blogs
4	29	Suicide: Be Here Tomorrow.	1625402	15808339	334546	Nonprofits & Activism
5	17	Real Life Trick Shots 2 Dude Perfect	926657	23417389	13003	Sports
6	23	Poo Bear ft. Justin Bieber & Jay Electronica - Har...	825258	29178096	15353	Comedy

Execution finished without errors.
Result: 16 rows returned in 89ms
At line 1:
SELECT videos.category_id,
videos.title,
MAX(videos.likes) AS likes,
videos.views,
videos.dislikes,
categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY category
ORDER BY videos.likes DESC

Videos con máximo de dislikes por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1	SQL 2	SQL 3																																										
<pre> 1 SELECT videos.category_id, 2 videos.title, 3 MAX(videos.dislikes) AS dislikes, 4 videos.views, 5 videos.likes, 6 categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY cate 7 </pre>																																												
<table> <tr> <th>category_id</th><th>title</th><th>dislikes</th><th>views</th><th>likes</th><th>category</th></tr> <tr> <td>24</td><td>So Sorry.</td><td>1278887</td><td>28407744</td><td>1204072</td><td>Entertainment</td></tr> <tr> <td>29</td><td>Suicide: Be Here Tomorrow.</td><td>334546</td><td>15808339</td><td>1625402</td><td>Nonprofits & Activism</td></tr> <tr> <td>20</td><td>Official Call of Duty®: Black Ops 4 — Multiplayer ...</td><td>212976</td><td>10306119</td><td>357079</td><td>Gaming</td></tr> <tr> <td>10</td><td>BTS (방탄소년단) 'FAKE LOVE' Official MV</td><td>190567</td><td>103786733</td><td>5386959</td><td>Music</td></tr> <tr> <td>22</td><td>we broke up</td><td>96814</td><td>31505838</td><td>1967904</td><td>People & Blogs</td></tr> <tr> <td>17</td><td>Justin Timberlake's FULL Pepsi Super Bowl LII ...</td><td>46697</td><td>10226518</td><td>143105</td><td>Sports</td></tr> </table>	category_id	title	dislikes	views	likes	category	24	So Sorry.	1278887	28407744	1204072	Entertainment	29	Suicide: Be Here Tomorrow.	334546	15808339	1625402	Nonprofits & Activism	20	Official Call of Duty®: Black Ops 4 — Multiplayer ...	212976	10306119	357079	Gaming	10	BTS (방탄소년단) 'FAKE LOVE' Official MV	190567	103786733	5386959	Music	22	we broke up	96814	31505838	1967904	People & Blogs	17	Justin Timberlake's FULL Pepsi Super Bowl LII ...	46697	10226518	143105	Sports		
category_id	title	dislikes	views	likes	category																																							
24	So Sorry.	1278887	28407744	1204072	Entertainment																																							
29	Suicide: Be Here Tomorrow.	334546	15808339	1625402	Nonprofits & Activism																																							
20	Official Call of Duty®: Black Ops 4 — Multiplayer ...	212976	10306119	357079	Gaming																																							
10	BTS (방탄소년단) 'FAKE LOVE' Official MV	190567	103786733	5386959	Music																																							
22	we broke up	96814	31505838	1967904	People & Blogs																																							
17	Justin Timberlake's FULL Pepsi Super Bowl LII ...	46697	10226518	143105	Sports																																							
<p>Execution finished without errors. Result: 16 rows returned in 77ms At line 1: SELECT videos.category_id, videos.title, MAX(videos.dislikes) AS dislikes, videos.views, videos.likes, categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY category ORDER BY videos.dislikes DESC</p>																																												

Videos con mínimo de views por categoría.

SQL 1

SQL 2

SQL 3

```

1 SELECT videos.category_id,
2     videos.title,
3     MIN(videos.views) AS views,
4     videos.likes,
5     videos.dislikes,
6     categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC
7

```

	category_id	title	views	likes	dislikes	category
1	43	You're not crazy. Apple is slowing down older ...	146353	4276	215	Shows
2	20	Nintendo Switch Now the Fastest-Selling Home ...	8493	309	17	Gaming
3	15	Polar bear cub audio	3577	12	0	Pets & Animals
4	2	old dodge stratus commercial	3312	4	4	Autos & Vehicles
5	23	Peter Bone to Trump: Delete Your Account	3052	7	39	Comedy
6	27	Dan Quayle Speech About Murphy Brown and ...	2302	2	0	Education

Execution finished without errors.

Result: 16 rows returned in 86ms

At line 1:

```

SELECT videos.category_id,
videos.title,
MIN(videos.views) AS views,
videos.likes,
videos.dislikes,
categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC

```

Videos con mínimo de likes por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1SQL 2SQL 3

```
1 SELECT videos.category_id,
2     videos.title,
3     MIN(videos.likes) AS likes,
4     videos.views,
5     videos.dislikes,
6     categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY cate
7
```

	category_id	title	likes	views	dislikes	category
1	43	You're not crazy. Apple is slowing down older ...	4276	146353	215	Shows
2	20	American vs. Brit	148	19459	12	Gaming
3	29	NSC Face to Face with the Prescription Opioid ...	19	1967	0	Nonprofits & Activism
4	26	Disney Princess Hand Pies	16	1515	0	Howto & Style
5	23	Peter Bone to Trump: Delete Your Account	7	3052	39	Comedy
6	15	This fox being a derp...	6	7325	6	Pets & Animals

Execution finished without errors.
Result: 16 rows returned in 78ms
At line 1:
SELECT videos.category_id,
 videos.title,
 MIN(videos.likes) AS likes,
 videos.views,
 videos.dislikes,
 categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY category
ORDER BY videos.likes DESC

Videos con mínimo de dislikes por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1SQL 2SQL 3

```
1 SELECT videos.category_id,
2     videos.title,
3     MIN(videos.dislikes) AS dislikes,
4     videos.views,
5     videos.likes,
6     categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY cate
7
```

	category_id	title	dislikes	views	likes	category
1	43	You're not crazy. Apple is slowing down older ...	215	146353	4276	Shows
2	20	American vs. Brit	12	19459	148	Gaming
3	2	Small Truck VS Strong Wind	2	13659	7	Autos & Vehicles
4	19	OJO Commuter Scooter®- Light Electric Vehicle f...	1	917	3	Travel & Events
5	17	HIGHLIGHTS: MK Dons U18s 1-0 Cardiff City U18s	0	20618	30	Sports
6	28	exixe clock in action	0	1425	5	Science & Technology

Execution finished without errors.
Result: 16 rows returned in 82ms
At line 1:
SELECT videos.category_id,
 videos.title,
 MIN(videos.dislikes) AS dislikes,
 videos.views,
 videos.likes,
 categorias.category FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False' GROUP BY category
ORDER BY videos.dislikes DESC

Cuenta de videos por categoría

SQL 1SQL 2SQL 3SQL 4

```
1 SELECT videos.category_id,
2 categorias.category,
3 COUNT(*) FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY videos.category_id ORDER BY COUNT(*) DESC
```

	category_id	category	COUNT(*)
1	24	Entertainment	1014
2	10	Music	569
3	26	Howto & Style	396
4	23	Comedy	352
5	22	People & Blogs	333
6	25	News & Politics	295

Execution finished without errors.
Result: 16 rows returned in 25ms
At line 1:
SELECT videos.category_id,
categorias.category,
COUNT(*) FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY videos.category_id ORDER BY COUNT(*) DESC

Promedio de views por categoría.

SQL 1SQL 2SQL 3SQL 4SQL 5SQL 6SQL 7

```
1 SELECT videos.category_id,
2 categorias.category,
3 AVG(videos.views) AS views
4 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC
5
```

	category_id	category	views
1	28	Science & Technology	1057102.2388664
2	22	People & Blogs	1315954.01801802
3	20	Gaming	2179809.97142857
4	10	Music	4359809.14586995
5	26	Howto & Style	867109.477272727
6	2	Autos & Vehicles	1777635.88636364

Execution finished without errors.
Result: 16 rows returned in 76ms
At line 1:
SELECT videos.category_id,
categorias.category,
AVG(videos.views) AS views
FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.views DESC

Promedio de likes por por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1 SQL 2 SQL 3 SQL 4 SQL 5 SQL 6 SQL 7

```
1 SELECT videos.category_id,
2 categorias.category,
3 AVG(videos.likes) AS views
4 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.likes DESC
```

	category_id	category	views
1	28	Science & Technology	27347.3805668016
2	22	People & Blogs	47384.3393393393
3	20	Gaming	68592.2285714286
4	10	Music	180789.492091388
5	26	Howto & Style	33839.946969697
6	23	Comedy	57624.5880681818

Execution finished without errors.
Result: 16 rows returned in 65ms
At line 1:
SELECT videos.category_id,
categorias.category,
AVG(videos.likes) AS views
FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.likes DESC

Promedio de dislikes por por categoría, excluyendo los que tienen 'rating disable' = True.

SQL 1 SQL 2 SQL 3 SQL 4 SQL 5 SQL 6 SQL 7

```
1 SELECT videos.category_id,
2 categorias.category,
3 AVG(videos.dislikes) AS views
4 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.dislikes DESC
```

	category_id	category	views
1	22	People & Blogs	2416.93093093093
2	28	Science & Technology	1368.24696356275
3	20	Gaming	7008.68571428571
4	10	Music	5752.69420035149
5	17	Sports	1564.37974683544
6	26	Howto & Style	1202.12878787879

Execution finished without errors.
Result: 16 rows returned in 74ms
At line 1:
SELECT videos.category_id,
categorias.category,
AVG(videos.dislikes) AS views
FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id GROUP BY category ORDER BY videos.dislikes DESC

Video con el mínimo de views.

The screenshot shows a SQL IDE with a tab labeled 'SQL 8'. The query editor contains the following SQL code:

```
1 SELECT videos.category_id,
2     categorias.category,
3     videos.title,
4     MIN(videos.views) AS views,
5     videos.likes,
6     videos.dislikes
7 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id
8
```

Below the query editor, a table displays the result of the query:

	category_id	category	title	views	likes	dislikes
1	17	Sports	Coach Taggart Monday Presser Ahead of Arizona	687	10	2

At the bottom, the execution status is shown:

Execution finished without errors.
Result: 1 rows returned in 63ms
At line 1:
SELECT videos.category_id,
categorias.category,
videos.title,
MIN(videos.views) AS views,
videos.likes,
videos.dislikes
FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id

Video con el mínimo de likes, excluyendo los que tienen 'rating disable' = True.

The screenshot shows a SQL IDE with a tab labeled 'SQL 9'. The query editor contains the following SQL code:

```
1 SELECT videos.category_id,
2     categorias.category,
3     videos.title,
4     videos.views,
5     MIN(videos.likes) AS likes,
6     videos.dislikes
7 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False'
```

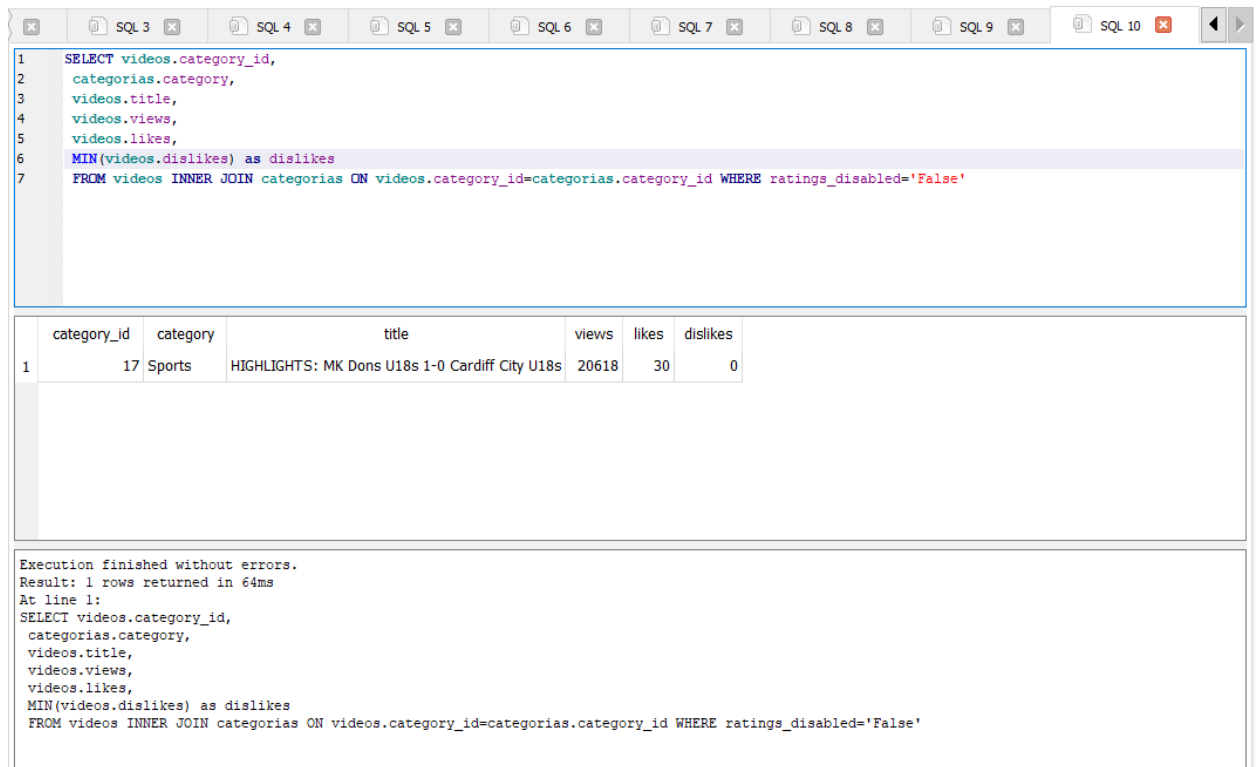
Below the query editor, a table displays the result of the query:

	category_id	category	title	views	likes	dislikes
1	22	People & Blogs	Apple Clips sample	2259	0	0

At the bottom, the execution status is shown:

Execution finished without errors.
Result: 1 rows returned in 57ms
At line 1:
SELECT videos.category_id,
categorias.category,
videos.title,
videos.views,
MIN(videos.likes) AS likes,
videos.dislikes
FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False'

Video con el mínimo de dislikes, excluyendo los que tienen 'rating disable' = True.



The screenshot shows a SQL IDE with multiple tabs labeled SQL 3 through SQL 10. The active tab, SQL 10, contains the following SQL query:

```
1 SELECT videos.category_id,  
2     categorias.category,  
3     videos.title,  
4     videos.views,  
5     videos.likes,  
6     MIN(videos.dislikes) as dislikes  
7 FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False'
```

Below the query editor, the results are displayed in a table with 7 columns: category_id, category, title, views, likes, and dislikes. The table contains one row of data:

category_id	category	title	views	likes	dislikes	
1	17	Sports	HIGHLIGHTS: MK Dons U18s 1-0 Cardiff City U18s	20618	30	0

At the bottom of the IDE, a status bar indicates: "Execution finished without errors. Result: 1 rows returned in 64ms. At line 1: SELECT videos.category_id, categorias.category, videos.title, videos.views, videos.likes, MIN(videos.dislikes) as dislikes FROM videos INNER JOIN categorias ON videos.category_id=categorias.category_id WHERE ratings_disabled='False'".

4 MongoDB

4.1 SENTENCIA O CONSULTA DE CREACIÓN DEL DOCUMENTO(S)

Observación: Escriba el código en MongoDB para crear al menos 20 documentos que correspondan a su conjunto de datos específico. Almacene en el repositorio **(REPO EN GITHUB)** el script con el nombre de T1.4.1.Creacion_Documentos.sql

Trabajo.Videosdata

DOCUMENTS 4.1k TOTAL SIZE 7.0MB AVG. SIZE 1.7KB INDEXES 1 TOTAL SIZE 120.0KB AVG. SIZE 120.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ↺ ...

ADD DATA VIEW {}

Displaying documents 1 - 20 of 4118 C REFRESH

```

_id: ObjectId("60ac62f970b15503d28fb861")
video_id: "4_aR3HqFrr0"
category_id: "24"
publish_time: "2018-05-15T12:59:50.000Z"
channel_title: "E! Entertainment"
title: "Nikki Bella's Love for Niece Makes Her Want Kids Even More | Total Bel..."
description: "Feeling excluded from all the baby talk at the dinner table, the Total..."
trending_date: "18.02.06"
thumbnail_link: "https://i.ytimg.com/vi/4_aR3HqFrr0/default.jpg"
comments_disabled: "False"
ratings_disabled: "False"
video_error_or_removed: "False"
views: "770131"
likes: "5585"
dislikes: "424"
comment_count: "727"
tags: "Real Time|Total Bellas|\"WWE\"|\"wrestler\"|\"wrestle\"|\"superstars\"|\"worl..."

_id: ObjectId("60ac62f970b15503d28fb862")
video_id: "vyqCSzFzyXg"
category_id: "10"
publish_time: "2018-04-26T06:02:51.000Z"
channel_title: "Shawn Mendes"
title: "Shawn Mendes: The Album"
description: "Pre-order Shawn Mendes: The Album here now: https://IslandRecs.lnk.to/..."
trending_date: "18.27.04"
thumbnail_link: "https://i.ytimg.com/vi/vyqCSzFzyXg/default.jpg"
comments_disabled: "False"
ratings_disabled: "False"
video_error_or_removed: "False"
views: "1093538"
likes: "96286"

```

4.2 SENTENCIA DE CONSULTA

Observación: Realice la exploración básica de los datos, conteos totales y por categorías, máximos y mínimos. Es decir, aplique estadística descriptiva con el fin de conocer las propiedades de los datos y entenderlos lo mejor posible. Anexe las tomas de pantalla donde evidencie la sentencia SQL y su correspondiente ejecución. Además, **Almacene en el repositorio (REPO EN GITHUB) script con el nombre de T1.4.2.Consultar_Datos.sql**

||| \$project [ON] [] [] Output after \$project stage (Sample of 20 documents)

```

1- /**
2-  * specifications: The fields to
3-  * include or exclude.
4-  */
5- {
6-   category_id: 1,
7-   title: 1,
8-   trending_date: 1,
9-   views: 1,
10-  likes: 1,
11-  dislikes: 1,
12-  comment_count: 1,
13- }

```

```

_id: ObjectId("60ac62f970b15503d28fb861")
category_id: "24"
title: "Nikki Bella's Love for Niece Makes Her Want Kids Even More | Total Bel..."
trending_date: "18.02.06"
views: "770131"
likes: "5585"
dislikes: "424"
comment_count: "727"

_id: ObjectId("60ac62f970b15503d28fb862")
category_id: "10"
title: "Shawn Mendes: The Album"
trending_date: "18.27.04"
views: "1093538"
likes: "96286"
dislikes: "1315"
comment_count: "1241"

```

Refinamiento de la base de datos.

Output after `$group` stage (Sample of 16 documents)

```

1 // **
2 * _id: The id of the group.
3 * fieldN: The first field name.
4 */
5 {
6   _id: '$category_id',
7   total: {
8     $sum: 1
9   },
10  vistasmin: {
11    $min: '$views',
12  },
13  vistasmax: {
14    $max: '$views',
15  }
16 }

```

Sample documents from output:

```

{ "_id": "19", "total": 37, "vistasmin": "103082", "vistasmax": "979217" }
{ "_id": "22", "total": 333, "vistasmin": "101737", "vistasmax": "996318" }

```

Maximos y minimos por vistas de videos por categoría. Adicionalmente, videos totales por categoría.

Resultados para las primeras 3 categorías:

1. `_id: "1"`
2. `total: 238`
3. `vistasmin: "1004815"`
4. `vistasmax: "99653"`
5. `_id: "2"`
6. `total: 44`
7. `vistasmin: "25193150"`
8. `vistasmax: "95509"`
9. `_id: "10"`
10. `total: 569`
11. `vistasmin: "103786733"`
12. `vistasmax: "9998724"`

Output after `$group` stage (Sample of 16 documents)

```

1 // **
2 * _id: The id of the group.
3 * fieldN: The first field name.
4 */
5 {
6   _id: '$category_id',
7   total: {
8     $sum: 1
9   },
10  Likesmin: {
11    $min: '$likes',
12  },
13  Likesmax: {
14    $max: '$likes',
15  },
16  Dislikesmax: {
17    $max: '$dislikes',
18  },
19  Dislikesmin: {
20    $min: '$dislikes',
21  }
22 }

```

Sample documents from output:

```

{ "_id": "20", "total": 70, "Likesmin": "10278", "Likesmax": "99594", "Dislikesmax": "9956", "Dislikesmin": "1018" }
{ "_id": "27", "total": 179, "Likesmin": "0", "Likesmax": "9965", "Dislikesmax": "980", "Dislikesmin": "0" }

```

Máximos y mínimos tanto para likes como dislikes por categoría. Además de videos totales por categoría.

Resultados para las primeras 3 categorías:

1. `_id: "1"`
2. `total: 238`
3. `Likesmin: "0"`
4. `Likesmax: "9978"`
5. `Dislikesmax: "9917"`
6. `Dislikesmin: "0"`
7. `_id: "2"`
8. `total: 44`

```
9. Likesmin: "0"
10. Likesmax: "9826"
11. Dislikesmax: "93"
12. Dislikesmin: "2"
13. _id: "10"
14. total: 569
15. Likesmin: "0"
16. Likesmax: "5386959"
17. Dislikesmax: "190567"
18. Dislikesmin: "0"
```

5 ANÁLISIS DE LECTURA

Observación: Considerando el artículo: “The Definitive Guide to Graph Databases for the RDBMS Developer” de Neo4J. Compartido en las carpeta de lecturas recomendadas. Analice y responda cada pregunta en máximo 150 palabras:

1. ¿Cuáles son las limitaciones, que se pueden inferir de la lectura, para migrar los conjuntos de datos relacionales a NoSQL?

Las nuevas tecnologías generan diferentes retos como resistencias al cambio tanto de las empresas como en los equipos de trabajo. A su vez, en la actualidad se encuentran pocos especialistas en estas nuevas tecnologías, por lo tanto es un campo inexplorado con pocos recursos para autogestión del personal técnico.

Por otro lado, el tratamiento de los conjuntos de datos relacionales es totalmente diferente a los no relacionales. Así, manipular y migrar los mismos como no relacionales podrían acarrear problemas, ya que se tendrían propiedades que no podrían migrar de la misma manera.

2. ¿Cuáles limitaciones adicionales se deben considerar, a parte de las mencionadas en el artículo?
 - a. Dada la misma novedad de la tecnología, la comunidad, Bibliografía, librerías y apoyos online de la que se puede disponer es relativamente escasa en comparación a las existentes para las tecnologías tradicionales.
 - b. Las bases de datos relacionales tienen estándares bien definidos para la creación, actualización y realización de consultas, cosa que puede no ocurrir en muchas ocasiones con un desarrollo reciente como lo son las bases no relacionales.
 - c. Migrar el modelo de almacenamiento implica un costo de inversión. Educar a los programadores, adquisición de nuevas licencias y costos adicionales en capital humano.
3. ¿Cuáles son las razones (criterios) que se deben considerar para migrar un conjunto de datos relacionados a NoSQL?
 - a. Un criterio de costo beneficio. Como se expresó anteriormente, migrar a esta nueva tecnología implica un costo que no en todas la empresas se traduce en un beneficio extra. En definitiva, hay un beneficio adicional al migrar a las BD NoSQL si la compañía

requiere procesamiento en tiempo real o gestionar contenidos de datos de tipo audiovisuales, pdf, entre otros archivos que las RDBM no son capaces de almacenar.

- b. Tener muy en claro el objetivo por el cual se está migrando la información y el tipo de consultas a realizar en la misma que empuje la necesidad de cambiar la base relacional a NoSQL.
- c. El tiempo y consumo de recursos computacionales para la migración. Adicionalmente, tener en cuenta que si se piensa migrar una base de datos relacional con muchos joints a NoSQL esto puede causar un detrimento en eficiencia. Por lo tanto, las características mismas de la base como volumen de la información deben ser tenidas en cuenta.