



آزمایشگاه پایگاه داده‌ها

فاز سوم پروژه

استاد

دکتر بصیری

پرnian ملک‌زاده - ۴۰۰۱۷۰۵۳

پاییز ۱۴۰۳

فهرست مطالب

2.....	فهرست مطالب
3.....	توضیح جداول:
8.....	توضیح Triggerها:
12.....	توضیح Functionها:
16.....	توضیح Viewها:
19.....	توضیح Procedureها:
21.....	توضیح برخی از اصول دیتابیس:

توضیح جداول:

جدول Users:

- این جدول اطلاعات مربوط به کاربران سیستم را ذخیره می‌کند.
- UserID: شناسه‌ی منحصر به فرد کاربر که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
 - FirstName: نام کاربر با حداکثر طول 50 کاراکتر.
 - LastName: نام خانوادگی کاربر با حداکثر طول 50 کاراکتر.
 - PhoneNumber: شماره تلفن کاربر با حداکثر طول 15 کاراکتر.
 - EmailAddress: آدرس ایمیل کاربر که منحصر به فرد است.
 - Username: نام کاربری منحصر به فرد کاربر با حداکثر طول 50 کاراکتر.
 - Password: رمز عبور کاربر (هش شده) با حداکثر طول 100 کاراکتر.
 - Gender: جنسیت کاربر با مقادیر محدود به 'Male', 'Female', 'Other'.
 - BirthDate: تاریخ تولد کاربر.

جدول PremiumPlans:

- این جدول اطلاعات مربوط به اشتراک‌های پرمیوم کاربران را ذخیره می‌کند.
- PremiumID: شناسه‌ی منحصر به فرد اشتراک که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
 - UserID: شناسه‌ی کاربری که اشتراک پرمیوم دارد (کلید خارجی به جدول Users).
 - StartDate: تاریخ شروع اشتراک.
 - ExpirationDate: تاریخ انقضای اشتراک.

جدول Channels:

- این جدول اطلاعات کانال‌های متعلق به کاربران را ذخیره می‌کند.
- ChannelID: شناسه‌ی منحصر به فرد کانال که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
 - UserID: شناسه‌ی کاربری که مالک کانال است (کلید خارجی به جدول Users).
 - SubscribersCount: تعداد مشترکین کانال.

- TotalViews: مجموع بازدیدهای ویدیوهای کانال.
- VideosCount: تعداد ویدیوهای کانال.
- Description: توضیحات کانال.
- Region: منطقه جغرافیایی کانال.
- DateJoined: تاریخ ایجاد کانال.

جدول Videos:

این جدول اطلاعات ویدیوها را ذخیره می‌کند.

- VideoID: شناسه‌ی منحصر به فرد ویدیو که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
- ChannelID: شناسه‌ی کانالی که ویدیو به آن تعلق دارد (کلید خارجی به جدول Channels).
- Title: عنوان ویدیو با حداکثر طول 255 کاراکتر.
- Description: توضیحات ویدیو.
- Duration: مدت زمان ویدیو.
- UploadDate: تاریخ بارگذاری ویدیو.
- Views: تعداد بازدیدهای ویدیو.
- LikesCount: تعداد لایک‌های ویدیو.
- CommentsCount: تعداد نظرات ویدیو.

جدول Timestamps:

این جدول نقاط خاصی از ویدیوها (نظیر بوکمارک یا بخش‌های مهم) را ذخیره می‌کند.

- TimestampID: شناسه‌ی منحصر به فرد تایم‌استمپ که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
- VideoID: شناسه‌ی ویدیویی که تایم‌استمپ مربوط به آن است (کلید خارجی به جدول Videos).
- TimeOnVideo: زمان مشخص در ویدیو.
- Title: عنوان توصیفی برای تایم‌استمپ.

جدول History:

این جدول تاریخچه مشاهده ویدیوهای کاربران را ذخیره می‌کند.

- HistoryID: شناسه‌ی منحصر به فرد تاریخچه که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
- UserID: شناسه‌ی کاربر که ویدیو را مشاهده کرده است (کلید خارجی به جدول Users).
- VideoID: شناسه‌ی ویدیو که مشاهده شده است (کلید خارجی به جدول Videos).
- WatchDate: تاریخ مشاهده ویدیو.
- CheckpointTime: آخرین زمانی که ویدیو مشاهده شده است.

جدول Subscriptions:

این جدول اشتراک کاربران در کانال‌ها را ذخیره می‌کند.

- SubscriptionID: شناسه‌ی منحصر به فرد اشتراک که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
- UserID: شناسه‌ی کاربری که مشترک کانال شده است (کلید خارجی به جدول Users).
- ChannelID: شناسه‌ی کانال که کاربر در آن مشترک شده است (کلید خارجی به جدول Channels).
- NotificationType: نوع اعلان‌ها برای اشتراک، مقادیر محدود به 'All', 'Recommended', 'None'.

جدول Playlists:

این جدول اطلاعات پلی‌لیست‌های کاربران را ذخیره می‌کند.

- PlaylistID: شناسه‌ی منحصر به فرد پلی‌لیست که به صورت خودکار مقداردهی می‌شود (کلید اصلی).
- UserID: شناسه‌ی کاربر که مالک پلی‌لیست است (کلید خارجی به جدول Users).
- Name: نام پلی‌لیست.
- IsPublic: وضعیت عمومی بودن یا نبودن پلی‌لیست (1 یا 0).
- LastModified: تاریخ آخرین تغییر پلی‌لیست.
- VideoCount: تعداد ویدیوهای موجود در پلی‌لیست.
- Views: تعداد بازدیدهای پلی‌لیست.

جدول PlaylistVideos:

این جدول ویدیوهای موجود در هر پلی لیست را ذخیره می کند.

- PlaylistVideoID: شناسه ی منحصر به فرد ویدیو در پلی لیست که به صورت خودکار مقداردهی می شود (کلید اصلی).
- PlaylistID: شناسه ی پلی لیستی که ویدیو به آن تعلق دارد (کلید خارجی به جدول Playlists).
- VideoID: شناسه ی ویدیویی که در پلی لیست قرار دارد (کلید خارجی به جدول Videos).

جدول Streams:

این جدول اطلاعات استریم های زنده کانال ها را ذخیره می کند.

- StreamID: شناسه ی منحصر به فرد استریم که به صورت خودکار مقداردهی می شود (کلید اصلی).
- ChannelID: شناسه ی کانالی که استریم متعلق به آن است (کلید خارجی به جدول Channels).
- StartTime: زمان شروع استریم.
- EndTime: زمان پایان استریم (در صورت پایان یافتن).

جدول InChatMessages:

این جدول پیام های چت استریم های زنده را ذخیره می کند.

- MessageID: شناسه ی منحصر به فرد پیام که به صورت خودکار مقداردهی می شود (کلید اصلی).
- StreamID: شناسه ی استریم که پیام مربوط به آن است (کلید خارجی به جدول Streams).
- UserID: شناسه ی کاربری که پیام ارسال کرده است (کلید خارجی به جدول Users).
- Content: محتوای پیام.
- SentTime: زمان ارسال پیام.

جدول Comments:

این جدول نظرات ویدیوها را ذخیره می‌کند.

- CommentID: شناسه‌ی منحصربه‌فرد نظر که به‌صورت خودکار مقداردهی می‌شود (کلید اصلی).
- VideoID: شناسه‌ی ویدیویی که نظر برای آن ثبت شده است (کلید خارجی به جدول Videos).
- UserID: شناسه‌ی کاربری که نظر داده است (کلید خارجی به جدول Users).
- Content: متن نظر.
- PostedTime: زمان ثبت نظر.
- LikesCount: تعداد لایک‌های نظر.
- RepliesCount: تعداد پاسخ‌ها به نظر.

جدول Reactions:

این جدول واکنش‌های کاربران به نظرات را ذخیره می‌کند.

- ReactionID: شناسه‌ی منحصربه‌فرد واکنش که به‌صورت خودکار مقداردهی می‌شود (کلید اصلی).
- CommentID: شناسه‌ی نظری که واکنش به آن داده شده است (کلید خارجی به جدول Comments).
- UserID: شناسه‌ی کاربری که واکنش داده است (کلید خارجی به جدول Users).
- ReactionType: نوع واکنش با مقادیر محدود به 'Like' یا 'Dislike'.
- ReactionTime: زمان ثبت واکنش.

توضیح Triggerها:

trg_IncreaseSubscribersCount:

این تریگر تعداد مشترکان (SubscribersCount) یک کانال را پس از اضافه شدن یک رکورد جدید در جدول Subscriptions افزایش می‌دهد.
مکانیزم این تریگر به این شکل است که با رخ دادن عملیات INSERT روی جدول Subscriptions:

ابتدا بررسی می‌کند که آیا رکوردی در جدول موقت inserted وجود دارد، سپس از ChannelID رکوردهای اضافه‌شده استفاده می‌کند تا مقدار SubscribersCount مربوط به آن کانال در جدول Channels را یک واحد افزایش دهد.

trg_DecreaseSubscribersCount:

این تریگر تعداد مشترکان (SubscribersCount) یک کانال را پس از حذف یک رکورد از جدول Subscriptions کاهش می‌دهد.

مکانیزم این تریگر به این شکل است که با رخ دادن عملیات DELETE روی جدول Subscriptions:

ابتدا بررسی می‌کند که آیا رکوردی در جدول موقت deleted وجود دارد، سپس از ChannelID رکوردهای حذف‌شده استفاده می‌کند تا مقدار SubscribersCount مربوط به آن کانال در جدول Channels را یک واحد کاهش دهد. در واقع این تریگر مشابه trg_IncreaseSubscribersCount است، اما برای حذف استفاده می‌شود.

UpdateVideosViewsAfterView:

این تریگر تعداد بازدیدها (Views) در جدول Videos و همچنین تعداد بازدیدهای مربوط به لیست‌های پخش (Playlists) پس از ثبت یک بازدید جدید در جدول History را به‌روزرسانی می‌کند.

مکانیزم این تریگر به این شکل است که با رخ دادن عملیات INSERT روی جدول History: تعداد بازدیدهای ویدیوهای مربوطه در جدول Videos یک واحد افزایش می‌یابد، تعداد بازدیدهای لیست‌های پخش مرتبط با آن ویدیوها نیز به‌روزرسانی می‌شود. در کوئری‌ها برای جلوگیری از به‌روزرسانی تکراری از DISTINCT استفاده شده است. همچنین کوئری دوم (به‌روزرسانی Playlists) از یک INNER JOIN برای یافتن لیست‌های پخش مرتبط استفاده می‌کند.

UpdateVideoCountAfterInsert:

این تریگر تعداد ویدیوهای مرتبط با یک کانال (VideosCount) پس از اضافه شدن ویدیوی جدید را به‌روزرسانی می‌کند.

مکانیزم این تریگر به این شکل است که با رخ دادن عملیات INSERT روی جدول Videos:

ابتدا بررسی می‌کند که رکوردی در جدول موقت inserted وجود دارد، سپس تعداد ویدیوهای کانال مرتبط در جدول Channels یک واحد افزایش می‌یابد. رابطه یک به چند بین Channels و Videos تضمین می‌کند که هر ویدیو به یک کانال مربوط است.

:trg_CalculateVideoPopularity

این تریگر تضمین مقداردهی اولیه و به‌روزرسانی مقادیر مرتبط با محبوبیت ویدیو (بازدیدها، تعداد لایک‌ها و کامنت‌ها) پس از اضافه شدن یا به‌روزرسانی ویدیو را انجام می‌دهد. مکانیزم این تریگر به این شکل است که با رخ دادن عملیات INSERT یا UPDATE روی جدول Videos:

مقادیر Views، LikesCount و CommentsCount بررسی و در صورت تهی بودن (NULL) مقداردهی می‌شوند. از تابع ISNULL برای مقداردهی پیش‌فرض استفاده شده است. همچنین یکنواختی داده‌ها تضمین می‌شود.

:trg_LogDeletedVideos

این تریگر اطلاعات ویدیوهای حذف‌شده را در جدول VideoDeletionLog ثبت می‌کند. مکانیزم این تریگر به این شکل است که با رخ دادن عملیات DELETE روی جدول Videos: اطلاعات ویدیوهای حذف‌شده از جدول موقت deleted گرفته شده و همراه با زمان حذف در جدول VideoDeletionLog ذخیره می‌شود. زمان حذف به‌صورت خودکار با استفاده از GETDATE() افزوده شده است.

:trg_PreventChannelDeletion

این تریگر از حذف کانال‌هایی که هنوز ویدیو دارند جلوگیری می‌کند. مکانیزم این تریگر به این شکل است که با رخ دادن عملیات DELETE روی جدول Channels:

ابتدا بررسی می‌کند که آیا کانال‌های موردنظر شامل ویدیو هستند، در صورت وجود ویدیو، پیام خطا تولید شده و عملیات متوقف می‌شود (RAISERROR). اگر کانال بدون ویدیو باشد، حذف انجام می‌شود. همچنین برای تغییر رفتار پیش‌فرض، از INSTEAD OF DELETE استفاده شده است.

:trg_UpdatePlaylistPopularity

این تریگر تعداد ویدیوها و بازدیدهای یک لیست پخش پس از اضافه یا حذف ویدیو را به روزرسانی می‌کند.

مکانیزم این تریگر به این شکل است که با رخ دادن عملیات INSERT یا DELETE روی جدول PlaylistVideos:

تعداد ویدیوها (VideoCount) و بازدیدهای (Views) لیست‌های پخش مرتبط به روزرسانی می‌شوند. از زیرکوئری‌های SUM و COUNT برای محاسبه مقادیر جدید استفاده شده است.

:trg_ValidatePhoneNumber

این تریگر برای اعتبار سنجی شماره تلفن کاربران طراحی شده است تا اطمینان حاصل شود که شماره تلفن وارد شده فرمت درستی دارد. هدف جلوگیری از درج شماره تلفن‌های نامعتبر در جدول Users است.

این تریگر به صورت INSTEAD OF INSERT تعریف شده و هنگام تلاش برای وارد کردن رکوردهای جدید به جدول Users فعال می‌شود. ابتدا، شماره تلفن وارد شده در رکوردهای جدید بررسی می‌شود تا شرایط زیر را رعایت کند: شماره تلفن باید با رقم "۰" شروع شود، طول شماره تلفن نباید بیشتر از ۱۵ کاراکتر باشد، شماره تلفن باید فقط شامل ارقام باشد و هیچ کاراکتر غیرعددی مانند حروف یا نمادها مجاز نیست. اگر شماره تلفن وارد شده این شرایط را رعایت نکند، خطایی با پیام 'Invalid phone number format. It must contain only digits and be at most 15 characters' ایجاد می‌شود و عمل درج متوقف خواهد شد. در غیر این صورت، رکورد جدید بدون مشکل به جدول Users اضافه می‌شود.

:trg_PreventInvalidChannelForVideo

این تریگر برای جلوگیری از درج ویدیوهایی در جدول Videos طراحی شده است که به کانال‌های نامعتبر یا غیرموجود ارجاع دارند. هدف اصلی این تریگر تضمین یکپارچگی داده‌ها میان جدول Videos و جدول Channels است.

این تریگر به صورت INSTEAD OF INSERT تعریف شده و هر زمان که رکوردی جدید به جدول Videos اضافه می‌شود، فعال می‌گردد و عملکرد آن به این شکل است که: بررسی می‌شود که آیا شناسه کانالی که ویدیو به آن تعلق دارد (فیلد ChannelID) در جدول Channels وجود دارد یا خیر. اگر ChannelID در جدول Channels یافت نشود، خطایی با پیام 'Cannot insert video for a non-existent channel' ایجاد شده و عملیات درج لغو می‌شود. اگر کانال معتبر باشد، رکورد جدید بدون مشکل در جدول Videos درج می‌شود.

:trg_LogUserUpdates

این تریگر برای ثبت تغییرات اعمال شده در جدول Users ایجاد شده است و به صورت AFTER UPDATE عمل می کند. به این معنا که پس از هر به روزرسانی موفق در جدول Users اجرا می شود. هدف این تریگر ذخیره اطلاعات کاربرانی است که نام یا ایمیل آنها تغییر کرده است. این تغییرات در جدول UserUpdateLog ثبت می شوند. ابتدا، تریگر بررسی می کند که آیا رکوردی در مجموعه inserted وجود دارد یا خیر. سپس با استفاده از اتصال بین مجموعه های deleted و inserted، تغییرات اعمال شده را شناسایی می کند. اگر ایمیل یا نام قبلی با مقادیر جدید متفاوت باشد، رکورد مربوطه به جدول UserUpdateLog اضافه می شود. در این جدول، شناسه کاربر، ایمیل قدیمی، ایمیل جدید، نام قدیمی، نام جدید، و زمان تغییرات ثبت می شوند.

:trg_PreventEmptyComments

این تریگر با استفاده از رویکرد INSTEAD OF INSERT ایجاد شده است تا از درج نظرات خالی در جدول Comments جلوگیری کند. زمانی که کاربر تلاش می کند یک نظر را به جدول اضافه کند، این تریگر ابتدا بررسی می کند که محتوای نظر (Content) خالی یا فقط شامل فاصله های خالی نباشد. اگر چنین حالتی وجود داشته باشد، تریگر یک خطا با پیام مناسب برمی گرداند و درج داده لغو می شود. در غیر این صورت، رکوردهای معتبر از مجموعه inserted به جدول Comments منتقل می شوند. این رویکرد تضمین می کند که تنها نظرات معتبر در سیستم ذخیره شوند.

:trg_LogVideoLikes

این تریگر برای ثبت تغییرات در تعداد لایک های ویدیوها طراحی شده است و به صورت AFTER UPDATE عمل می کند. هر زمان که رکوردی در جدول Videos به روزرسانی شود، این تریگر اجرا می شود. ابتدا بررسی می کند که آیا تغییری در تعداد لایک های ویدیو رخ داده است یا خیر. با استفاده از اتصال بین مجموعه های deleted و inserted، رکوردهایی که تعداد لایک های قدیمی و جدیدشان متفاوت است شناسایی می شوند. برای هر رکورد، شناسه ویدیو، تعداد لایک های قدیمی، تعداد لایک های جدید، و زمان به روزرسانی به جدول VideoLikesLog اضافه می شود. این جدول به عنوان یک گزارش کامل از تغییرات لایک های ویدیوها عمل می کند.

:trg_PreventDuplicateLikes

این تریگر از درج واکنش‌های تکراری (به‌ویژه لایک‌های تکراری) در جدول Reactions جلوگیری می‌کند. این تریگر با رویکرد INSTEAD OF INSERT ایجاد شده است و پیش از درج واکنش جدید، بررسی می‌کند که آیا واکنش مشابهی با نوع "لایک" توسط همان کاربر برای همان ویدیو قبلاً وجود داشته است یا خیر. اگر چنین واکنشی وجود داشته باشد، تریگر یک خطا با پیام مناسب برمی‌گرداند و از درج واکنش جلوگیری می‌کند. در غیر این صورت، واکنش‌های جدید از مجموعه inserted به جدول Reactions منتقل می‌شوند. این رویکرد تضمین می‌کند که کاربران نمی‌توانند به‌طور ناخواسته یا عمدی یک ویدیو را چندین بار لایک کنند.

توضیح Function ها:

:dbo.CheckPremiumStatus

هدف این تابع بررسی وضعیت اشتراک یک کاربر (فعال، منقضی‌شده یا بدون اشتراک) است.

این تابع با گرفتن UserID به‌عنوان ورودی:

جدول PremiumPlans را بررسی می‌کند تا تاریخ انقضای اشتراک را بیابد، سپس وضعیت اشتراک را بر اساس شرط‌های زیر مشخص می‌کند: اگر ExpirationDate تهی (NULL) باشد: "No Subscription"، اگر ExpirationDate بزرگ‌تر از تاریخ جاری باشد: "Active" و در غیر این صورت: "Expired". مقدار نهایی با استفاده از ISNULL تضمین می‌شود که حتی در صورت نبود اشتراک، مقدار پیش‌فرض "No Subscription" بازگردانده شود.

:dbo.GetChannelPerformance

هدف این تابع ارزیابی عملکرد یک کانال خاص با جمع‌آوری اطلاعات آماری است.

این تابع یک جدول بازگشتی است که:

تعداد ویدیوها (TotalVideos)، بازدیدها (TotalViews)، لایک‌ها (TotalLikes)، و کامنت‌ها (TotalComments) را برای کانال مشخص‌شده جمع‌آوری می‌کند، میانگین بازدید هر ویدیو (AvgVideoViews) را محاسبه می‌کند، تعداد روزهای سپری‌شده از تاریخ ایجاد

کانال (DaysSinceJoined) را محاسبه می‌کند و از یک LEFT JOIN استفاده می‌کند تا حتی اگر کانال بدون ویدیو باشد، اطلاعات کانال بازگردانده شود.

:dbo.GetTopVideosByChannel

هدف این تابع دریافت محبوب‌ترین ویدیوهای یک کانال مشخص با محدودیت تعداد (TopN) است.

این تابع یک جدول بازگشتی است که:

ویدیوهای کانال مشخص شده را بر اساس یک محاسبه امتیاز محبوبیت مرتب می‌کند: امتیاز محبوبیت = بازدیدها + (2×لایک‌ها) + (3×کامنت‌ها). همچنین محدودیت تعداد ویدیوهای بازگشتی را با (@TopN) TOP اعمال می‌کند و ویدیوها را بر اساس امتیاز محبوبیت به صورت نزولی مرتب می‌کند.

:dbo.GetVideoPopularityScore

هدف این تابع محاسبه امتیاز محبوبیت یک ویدیو است.

این تابع یک مقدار عددی بازمی‌گرداند که امتیاز محبوبیت ویدیو را مشخص می‌کند. فرمول محاسبه به این شکل است که: بازدیدها (40%) + لایک‌ها (30%) + کامنت‌ها (20%). همچنین یک امتیاز اضافی بر اساس مدت زمان سپری‌شده از آپلود در نظر گرفته می‌شود به طوری که: اگر کمتر از 7 روز: 20+، اگر بین 7 و 30 روز: 10+، در غیر این صورت: 0+. مقادیر تهی (NULL) با ISNULL به صفر تبدیل می‌شوند.

:dbo.GetTopRecommendedVideos

هدف این تابع دریافت 5 ویدیوی پیشنهادی برای یک کاربر است.

این تابع یک جدول بازگشتی است که:

ویدیوهایی که کاربر تاکنون تماشا نکرده (با بررسی جدول History) را انتخاب می‌کند، سپس ویدیوها را بر اساس امتیاز محبوبیت (PopularityScore) محاسبه‌شده با استفاده از تابع dbo.GetVideoPopularityScore مرتب می‌کند و حداکثر 5 ویدیو را بازمی‌گرداند.

:dbo.GetPlaylistPopularityScore

هدف این تابع محاسبه امتیاز محبوبیت یک لیست پخش است.

این تابع یک مقدار عددی بازمی‌گرداند که امتیاز محبوبیت یک لیست پخش را مشخص می‌کند.

فرمول محاسبه به این شکل است که: بازدیدها (60%) + تعداد ویدیوها (40%). مقادیر تهی (NULL) با ISNULL به صفر تبدیل می‌شوند.

:dbo.GetActivePremiumUsers

هدف این تابع دریافت کاربران فعالی که اشتراک پریمیوم دارند است.

این تابع یک جدول بازگشتی است که:

تمام کاربران دارای اشتراک با تاریخ انقضا بزرگ‌تر از تاریخ جاری را انتخاب می‌کند، سپس اطلاعاتی مانند نام کاربری، تاریخ شروع و پایان اشتراک را بازمی‌گرداند. از یک INNER JOIN برای اتصال جداول Users و PremiumPlans استفاده می‌کند.

:dbo.GetTotalVideosWatchedByUser

هدف این تابع محاسبه تعداد ویدیوهای منحصربه‌فردی است که یک کاربر تماشا کرده است.

این تابع یک مقدار عددی بازمی‌گرداند که تعداد ویدیوهای منحصربه‌فرد تماشا شده توسط کاربر را مشخص می‌کند. در واقع با استفاده از COUNT(DISTINCT VideoID) تعداد ویدیوها از جدول History شمارش می‌شود. مقادیر تهی (NULL) با ISNULL به صفر تبدیل می‌شوند.

:dbo.GetUserEngagementPercentage

این تابع یک مقدار عددی اعشاری را که نشان‌دهنده درصد تعامل کاربران با یک ویدیو خاص است، بازمی‌گرداند. ورودی تابع، VideoID، شناسه ویدیو است. ابتدا تعداد بازدیدها (Views)، تعداد لایک‌ها (LikesCount) و تعداد نظرات (CommentsCount) از جدول Videos برای ویدیوی مشخص شده استخراج می‌شود. اگر تعداد بازدیدها صفر باشد، مقدار تعامل برابر با صفر در نظر گرفته می‌شود. در غیر این صورت، تعامل کاربران با استفاده از فرمول $100 * ((Comments) / Views + Likes)$ محاسبه می‌شود. این تابع با دقت موارد تهی را بررسی می‌کند و از مقدار پیش‌فرض صفر برای مقادیر نامشخص استفاده می‌کند تا از بروز خطا جلوگیری شود.

:dbo.GetInactiveUsers

این تابع به صورت یک جدول بازگشتی است و شامل کاربرانی است که طی 30 روز گذشته هیچ فعالیتی در پلتفرم نداشته‌اند یا هرگز فعالیتی نداشته‌اند. تابع از اتصال خارجی بین جداول Users و History استفاده می‌کند و بیشترین تاریخ بازدید هر کاربر از جدول History را استخراج می‌کند. اگر حداکثر تاریخ بازدید کمتر از 30 روز پیش باشد یا هیچ تاریخ بازدیدی وجود نداشته باشد، کاربر به عنوان کاربر غیرفعال در نتیجه نمایش داده می‌شود. نتیجه نهایی شامل شناسه کاربر، نام کاربری، و تاریخ آخرین فعالیت است.

:dbo.GetChannelRevenue

این تابع میزان درآمد یک کانال خاص را بر اساس تعداد بازدیدهای ویدیوهای آن و مقدار درآمد به ازای هر بازدید محاسبه می‌کند. تابع دو ورودی دریافت می‌کند: ChannelID (شناسه کانال) و RevenuePerView (مقدار درآمد به ازای هر بازدید). ابتدا مجموع بازدیدهای ویدیوهای مربوط به آن کانال از جدول Videos محاسبه می‌شود. سپس درآمد کل با ضرب تعداد کل بازدیدها در درآمد هر بازدید به دست می‌آید. اگر کانال هیچ بازدیدی نداشته باشد، مقدار صفر برگردانده می‌شود.

:dbo.GetBestUploadTime

این تابع به صورت یک جدول بازگشتی است و ساعتی از شبانه‌روز را که بیشترین بازدید را در پلتفرم داشته است، شناسایی می‌کند. این تابع از جدول History استفاده کرده و ساعت مربوط به هر بازدید را استخراج می‌کند. سپس تعداد بازدیدها برای هر ساعت جمع‌آوری شده و نتایج بر اساس تعداد بازدیدها به ترتیب نزولی مرتب می‌شوند. نتیجه نهایی شامل ساعت از شبانه‌روز و تعداد بازدیدهای مربوط به آن ساعت است.

:dbo.GetAverageWatchTimeByUser

این تابع میانگین زمان مشاهده ویدیو توسط یک کاربر خاص را بازمی‌گرداند. ورودی تابع، UserID، شناسه کاربر است. ابتدا مجموع مدت‌زمان مشاهده‌شده و تعداد ویدیوهای مشاهده‌شده توسط کاربر از جدول History استخراج می‌شود. اگر کاربر هیچ ویدیویی را مشاهده نکرده باشد، مقدار صفر بازگردانده می‌شود. در غیر این صورت، میانگین مدت‌زمان مشاهده با تقسیم مجموع مدت‌زمان مشاهده‌شده بر تعداد ویدیوها محاسبه می‌شود. این

تابع از مقادیر پیش‌فرض برای جلوگیری از بروز خطا در شرایطی که هیچ داده‌ای وجود ندارد، استفاده می‌کند.

dbo.GetTopChannelsByViews:

این تابع به صورت یک جدول بازگشتی است و لیست کانال‌هایی را که بیشترین تعداد بازدید را در پلتفرم داشته‌اند، ارائه می‌دهد. ورودی تابع، TopN، تعداد کانال‌های برتر موردنظر است. ابتدا جدول Videos با جدول Channels اتصال داده می‌شود تا تعداد کل بازدیدهای هر کانال محاسبه شود. سپس کانال‌ها بر اساس تعداد کل بازدیدها به ترتیب نزولی مرتب شده و تعداد مشخصی از کانال‌ها به عنوان نتیجه بازگردانده می‌شوند. نتیجه شامل شناسه کانال، نام کانال، و تعداد کل بازدیدها است.

توضیح View ها:

View_PlaylistVideos:

هدف این ویو، دریافت اطلاعات مربوط به پلی‌لیست‌ها به همراه ویدیوهای داخل هر پلی‌لیست است.

اطلاعاتی که بازگردانده می‌شود عبارت‌اند از:

UserID: شناسه کاربری که پلی‌لیست را ساخته است، PlaylistID و PlaylistName: مشخصات پلی‌لیست و اطلاعات ویدیوهای موجود در پلی‌لیست شامل VideoID و VideoTitle و VideoDescription و سایر جزئیات مانند تاریخ آپلود و تعداد بازدید. با استفاده از JOIN جداول Playlists و PlaylistVideos و Videos به یکدیگر متصل شده‌اند.

User_Following_Channels:

هدف این ویو نمایش کانال‌هایی است که کاربران دنبال می‌کنند.

مکانیزم این ویو به این شکل است که:

اطلاعات مربوط به کاربران (UserID, Username) و کانال‌های دنبال‌شده (Description) از جدول Subscriptions استخراج می‌شود. از JOIN جداول Subscriptions و Users و Channels استفاده می‌شود.

:UserWatchHistory

هدف این ویو، نمایش تاریخچه تماشای کاربران همراه با اطلاعات ویدیوها و امتیاز محبوبیت هر ویدیو است.

اطلاعاتی که بازگردانده می‌شود عبارت‌اند از:

کاربر (UserID, Username)، ویدیوهای تماشاشده (VideoTitle, VideoViews) و امتیاز محبوبیت آنها با استفاده از dbo.GetVideoPopularityScore، پلی‌لیستی که ویدیو در آن قرار داشته (PlaylistName) و اطلاعات کانال مربوطه (ChannelName) و جزئیات تاریخ و مدت زمان تماشا (WatchDate, CheckpointTime). از INNER JOIN و LEFT JOIN برای اتصال جداول History و Videos و Channels و PlaylistVideos و Playlists استفاده می‌شود.

:ChannelPerformanceSummary

هدف این ویو ارائه خلاصه عملکرد کانال‌ها با اطلاعات آماری است.

مکانیزم این ویو به این شکل است که:

اطلاعات کانال مانند ChannelID و ChannelName و Region و DateJoined در نظر گرفته می‌شود، سپس تعداد ویدیوها، مجموع بازدیدها، مجموع لایک‌ها، و میانگین بازدید هر ویدیو و همچنین مدت زمان فعالیت کانال از زمان ایجاد تا کنون محاسبه می‌شود. از LEFT JOIN برای اتصال جداول Videos و Channels استفاده می‌شود.

:ActivePremiumUsers

هدف این ویو دریافت کاربران فعال با اشتراک پریمیوم است.

مکانیزم این ویو به این شکل است که:

اطلاعات کاربر (UserID, Username) و تاریخ شروع و پایان اشتراک در نظر گرفته می‌شود، یک ستون محاسباتی (PremiumStatus) اضافه شده که وضعیت اشتراک را مشخص می‌کند: اگر تاریخ انقضا بعد از تاریخ جاری باشد: "Active" و در غیر این صورت: "Expired". از INNER JOIN برای اتصال جداول PremiumPlans و Users استفاده می‌شود.

:RecommendedVideos

هدف این ویو نمایش ویدیوهای پیشنهادی برای هر کاربر است.

اطلاعاتی که بازگردانده می‌شود عبارت‌اند از:

کاربر (UserID, Username)، ویدیوهای پیشنهادی (VideoID, VideoTitle)، که از dbo.GetTopRecommendedVideos استخراج می‌شود. از CROSS APPLY برای استفاده از تابع dbo.GetTopRecommendedVideos استفاده شده و ویدیوهای پیشنهاد شده برای هر کاربر بازگردانده می‌شود.

:MostActiveUsers

هدف این ویو نمایش کاربران فعال بر اساس تعاملات آنها است. اطلاعاتی که بازگردانده می‌شود عبارت‌اند از: شناسه و نام کاربر، تعداد ویدیوهای تماشا شده (VideosWatched)، کامنت‌های ارسال شده (CommentsMade)، و واکنش‌های داده شده (ReactionsGiven). مجموع این تعاملات به عنوان TotalInteractions در نظر گرفته می‌شود. از LEFT JOIN برای اتصال جداول Users، History، Comments و Reactions استفاده شده است.

:AdvancedVideoPerformance

هدف این ویو ارائه جزئیات عملکرد پیشرفته برای هر ویدیو است. اطلاعاتی که بازگردانده می‌شود عبارت‌اند از: جزئیات ویدیو (VideoID, Title, Views, PopularityScore)، یک ستون محاسباتی (VideoStatus) که وضعیت ویدیو را مشخص می‌کند: اگر کمتر از 7 روز از آپلود گذشته باشد: "Trending"، اگر بین 7 تا 30 روز باشد: "New" و در غیر این صورت: "Archived". همچنین امتیاز محبوبیت هر ویدیو با استفاده از dbo.GetVideoPopularityScore محاسبه می‌شود.

:FastestGrowingChannels

هدف این ویو شناسایی کانال‌هایی است که سریع‌ترین رشد را داشته‌اند. مکانیزم این ویو به این شکل است که: اطلاعات کانال مانند تعداد مشترکان (SubscribersCount) و بازدیدها (TotalViews)، نرخ رشد روزانه بر اساس تعداد مشترکان و مدت زمان فعالیت محاسبه می‌شود: $\text{نرخ رشد} = \frac{\text{تعداد مشترکان}}{\text{تعداد روزهای فعالیت}}$. از LEFT JOIN برای اتصال جداول Channels و Videos استفاده شده است.

PremiumUsersPerformance:

هدف این ویو ارزیابی عملکرد کاربران با اشتراک پریمیوم است.

اطلاعاتی که بازگردانده می‌شود عبارت‌اند از:

اطلاعات کاربر (UserID, Username)، تاریخ شروع و پایان اشتراک، تعداد ویدیوهای تماشا شده، کامنت‌های ارسال شده، و واکنش‌های داده شده، یک ستون محاسباتی (PremiumStatus) که وضعیت اشتراک را مشخص می‌کند و تعداد کل ویدیوهای تماشا شده توسط کاربر با استفاده از dbo.GetTotalVideosWatchedByUser. از LEFT JOIN برای اتصال جداول Users و PremiumPlans و History و Comments و Reactions استفاده شده است.

TrendingVideos:

این ویو برای شناسایی و نمایش ویدیوهایی که در هفت روز گذشته آپلود شده‌اند و به لحاظ محبوبیت در میان کاربران برجسته هستند، طراحی شده است. این ویو اطلاعات مهمی از ویدیوها مانند شناسه، عنوان، توضیحات، تاریخ آپلود، تعداد بازدیدها، تعداد لایک‌ها، و تعداد نظرات را ارائه می‌دهد. علاوه بر این، از یک تابع محاسباتی به نام GetVideoPopularityScore استفاده می‌کند که امتیاز محبوبیت ویدیو را بر اساس معیارهای مختلف محاسبه می‌کند. فیلتر WHERE تضمین می‌کند که تنها ویدیوهایی که در هفت روز گذشته آپلود شده‌اند وارد ویو شوند. همچنین، مقدار روزهای سپری شده از زمان آپلود هر ویدیو نیز محاسبه و نمایش داده می‌شود. ویدیوها بر اساس امتیاز محبوبیت به ترتیب نزولی مرتب می‌شوند تا محبوب‌ترین ویدیوها در صدر لیست قرار بگیرند. این ویو ابزاری مناسب برای تحلیل سریع محتوای پرتیرفردار در بازه زمانی اخیر است.

PlaylistEngagement:

این ویو برای ارائه اطلاعات تحلیلی در مورد لیست‌های پخش و سطح تعامل کاربران با آن‌ها طراحی شده است. این ویو شامل اطلاعاتی از قبیل شناسه لیست پخش، نام لیست، شناسه و نام کاربر مالک لیست، تعداد کل بازدیدها، تعداد دفعات مشاهده شده ویدیوهای لیست توسط کاربران مختلف، تعداد نظرات ثبت شده روی ویدیوهای موجود در لیست، و امتیاز محبوبیت لیست پخش است که توسط تابع GetPlaylistPopularityScore محاسبه می‌شود. اتصال جداول Users و Playlists، PlaylistVideos، History، Comments و Reactions صورت چپ‌جایی، اطمینان می‌دهد که تمامی لیست‌های پخش حتی در صورت نبود تعامل

کاربر نمایش داده شوند. در نهایت، این اطلاعات به صورت گروه‌بندی شده توسط شناسه لیست پخش نمایش داده می‌شود. این ویو به‌عنوان یک ابزار کلیدی برای تحلیل محبوبیت و میزان تعامل کاربران با لیست‌های پخش عمل می‌کند.

:RecentActivitySummary

این ویو فعالیت‌های اخیر کاربران در پلتفرم را طی 30 روز گذشته به صورت جامع و خلاصه نمایش می‌دهد. این ویو با اتصال جداول Users، History، Comments، Reactions و Videos اطلاعاتی نظیر شناسه کاربر، نام کاربری، شناسه و عنوان ویدیو، تاریخ مشاهده ویدیو، شناسه و متن نظرات ثبت‌شده، و نوع واکنش کاربران ارائه می‌کند. با استفاده از یک شرط در بخش WHERE، تنها فعالیت‌هایی که طی 30 روز اخیر رخ داده‌اند، شامل مشاهده ویدیو، ثبت نظر و واکنش به محتوا، در نظر گرفته می‌شوند. همچنین، یک ستون محاسباتی به نام ActivityType اضافه شده است که نوع فعالیت هر کاربر را به صورت متنی مشخص می‌کند. این ویو ابزاری مفید برای تحلیل رفتار کاربران فعال در بازه زمانی اخیر است و می‌تواند به بهبود تجربه کاربری و شناسایی الگوهای تعامل کاربران کمک کند.

توضیح Procedure ها:

:AddVideoToChannel

این پرسیجر با هدف افزودن یک ویدیو جدید به یک کانال و به‌روزرسانی تعداد ویدیوهای آن طراحی شده است. ابتدا ورودی‌های مورد نیاز شامل شناسه کانال، عنوان ویدیو، توضیحات ویدیو، مدت زمان و تاریخ آپلود را دریافت می‌کند. سپس یک رکورد جدید در جدول Videos با مقادیر اولیه صفر برای بازدیدها، تعداد لایک‌ها و تعداد کامنت‌ها اضافه می‌کند. در پایان، تعداد ویدیوهای کانال در جدول Channels به‌روزرسانی شده و یک واحد افزایش داده می‌شود.

:RemoveUserSubscription

هدف این پرسیجر حذف اشتراک کاربر از یک کانال و کاهش تعداد مشترکین آن کانال است. ابتدا شناسه کاربر و شناسه کانال به عنوان ورودی دریافت می‌شود. سپس اشتراک کاربر از

جدول Subscriptions حذف می‌شود و تعداد مشترکین کانال در جدول Channels یک واحد کاهش می‌یابد.

:RecommendVideosForUser

این پرسیجر برای نمایش لیستی از ویدیوهای پیشنهادی برای یک کاربر طراحی شده است. ورودی‌های آن شامل شناسه کاربر و تعداد ویدیوهای پیشنهادی برتر است. این پرسیجر از نتیجه تابع `dbo.GetTopRecommendedVideos` برای استخراج ویدیوهای پیشنهادی استفاده کرده و لیست ویدیوها را بر اساس امتیاز محبوبیت به ترتیب نزولی مرتب می‌کند تا بهترین پیشنهادها نمایش داده شوند.

:AddCommentToVideo

این پرسیجر وظیفه افزودن یک نظر جدید به یک ویدیو را بر عهده دارد. ورودی‌ها شامل شناسه ویدیو، شناسه کاربر، و محتوای نظر است. پس از دریافت ورودی‌ها، یک رکورد جدید به جدول Comments با مقادیر اولیه صفر برای تعداد لایک‌ها و پاسخ‌ها اضافه می‌شود و تاریخ ارسال نظر به زمان جاری تنظیم می‌شود.

:DeleteComment

پرسیجر حذف یک نظر و تمامی واکنش‌های مرتبط با آن است. این پرسیجر ابتدا شناسه نظر را به عنوان ورودی دریافت می‌کند، سپس تمامی واکنش‌های مربوطه از جدول Reactions حذف می‌شوند. در نهایت، خود نظر از جدول Comments حذف می‌شود.

:AddReactionToComment

این پرسیجر برای افزودن یک واکنش (مانند لایک یا دیسلایک) به یک نظر استفاده می‌شود. ورودی‌های آن شامل شناسه نظر، شناسه کاربر، و نوع واکنش است. با دریافت این ورودی‌ها، واکنش جدید به جدول Reactions اضافه می‌شود و زمان ثبت واکنش به زمان جاری تنظیم می‌شود.

:AddVideoToPlaylist

هدف این پرسیجر افزودن یک ویدیو به یک پلی‌لیست و به‌روزرسانی تعداد ویدیوهای موجود در آن است. ورودی‌ها شامل شناسه پلی‌لیست و شناسه ویدیو هستند. ابتدا بررسی می‌شود

که آیا این ویدیو قبلاً در پلی‌لیست وجود دارد یا خیر. اگر ویدیو از قبل موجود نباشد، به جدول PlaylistVideos اضافه می‌شود و تعداد ویدیوهای پلی‌لیست در جدول Playlists افزایش می‌یابد.

:StartLiveStream

این پرسیجر برای شروع یک پخش زنده برای یک کانال استفاده می‌شود. ورودی آن شناسه کانال است. با اجرای این پرسیجر، یک رکورد جدید در جدول Streams با زمان شروع تنظیم‌شده به زمان جاری و مقدار اولیه NULL برای زمان پایان اضافه می‌شود.

:EndLiveStream

هدف این پرسیجر پایان دادن به یک پخش زنده است. ورودی آن شناسه پخش زنده است. این پرسیجر زمان پایان پخش زنده را در جدول Streams به زمان جاری به‌روزرسانی می‌کند و پخش زنده خاتمه می‌یابد.

:AddPremiumSubscription

این پرسیجر برای افزودن یک اشتراک پریمیوم جدید به کاربر طراحی شده است. این پرسیجر سه ورودی اصلی دارد: شناسه کاربر (@UserID)، تاریخ شروع اشتراک (@StartDate)، و مدت زمان اشتراک به ماه (@DurationInMonths). ابتدا تاریخ انقضای اشتراک با استفاده از تابع DATEADD محاسبه می‌شود، که ماه‌های مشخص‌شده را به تاریخ شروع اضافه می‌کند. سپس اطلاعات مربوط به اشتراک شامل شناسه کاربر، تاریخ شروع، و تاریخ انقضا در جدول PremiumPlans ذخیره می‌شود. این پرسیجر امکان مدیریت آسان اشتراک‌های پریمیوم کاربران را فراهم می‌کند و تضمین می‌کند که تاریخ انقضا به درستی محاسبه و ثبت شود.

:RemoveVideoFromPlaylist

این پرسیجر برای حذف یک ویدیو از یک لیست پخش خاص طراحی شده است. ورودی‌های این پرسیجر شامل شناسه لیست پخش (@PlaylistID) و شناسه ویدیو (@VideoID) است. ابتدا ویدیو مشخص‌شده از جدول PlaylistVideos حذف می‌شود. سپس، تعداد ویدیوهای موجود در لیست پخش با کاهش یک واحد در جدول Playlists به‌روزرسانی

می‌شود. این پرسیجر با حذف صحیح ویدیو و مدیریت تعداد ویدیوها، یکپارچگی داده‌ها در لیست‌های پخش را حفظ می‌کند و برای مدیریت محتوا در لیست‌های پخش کاربردی است.

FetchUserRecommendations:

این پرسیجر برای ارائه لیستی از ویدیوهای پیشنهادی به کاربر طراحی شده است. این پرسیجر دو ورودی دارد: شناسه کاربر (UserID@) و تعداد حداکثری نتایج بازگشتی (MaxResults@). ابتدا از تابع GetTopRecommendedVideos استفاده می‌شود که بر اساس شناسه کاربر، لیستی از ویدیوهای پیشنهادی ارائه می‌دهد. سپس این لیست با جدول Videos مرتبط می‌شود تا اطلاعات ویدیو شامل شناسه، عنوان، توضیحات، تعداد بازدیدها، تعداد لایک‌ها، و امتیاز محبوبیت محاسبه‌شده توسط فانکشن GetVideoPopularityScore به دست آید. نتایج بر اساس امتیاز محبوبیت به ترتیب نزولی مرتب شده و تعداد مشخص‌شده از نتایج بازگردانده می‌شود. این پرسیجر ابزاری قدرتمند برای پیشنهاد ویدیوهای شخصی‌سازی‌شده بر اساس رفتار و علایق کاربران است.

ResetUserPassword:

این پرسیجر برای تنظیم مجدد رمز عبور کاربران طراحی شده است. این پرسیجر دو ورودی دارد: شناسه کاربر (UserID@) و هش رمز عبور جدید (NewPasswordHash@). هنگامی که کاربر نیاز به بازبینی رمز عبور دارد یا مدیریت سیستم بخواهد رمز عبور را تغییر دهد، این پرسیجر هش رمز عبور جدید را به‌روزرسانی می‌کند. دستور UPDATE جدول Users را برای کاربری که شناسه آن با UserID@ مطابقت دارد، به‌روزرسانی کرده و مقدار جدید هش را ثبت می‌کند. این پرسیجر امنیت حساب کاربران را با مدیریت دقیق تغییر رمز عبور تضمین می‌کند و فرآیند تغییر رمز عبور را به شکلی کارآمد و امن پیاده‌سازی می‌کند.

توضیح برخی از اصول دیتابیس:

روابط ضعیف:

روابط ضعیف (Weak Relations) به روابطی گفته می‌شود که موجودیت فرزند (Child Entity) بدون موجودیت والد (Parent Entity) وجود خارجی ندارد. در این روابط، موجودیت فرزند برای شناسایی به موجودیت والد وابسته است و به عبارتی موجودیت فرزند، به تنهایی

معنای مستقلی ندارد. برای مدیریت این وابستگی، معمولاً کلید خارجی موجودیت فرزند به کلید اصلی موجودیت والد متصل می‌شود. مثال‌های روابط ضعیف در این طراحی عبارت‌اند از:

PremiumPlans وابسته به Users است: PremiumPlans یک موجودیت ضعیف است، زیرا بدون وجود یک کاربر در جدول Users، رکوردهای این جدول بی‌معنی هستند. کلید خارجی UserID به Users(UserID) متصل است.

Timestamps وابسته به Videos است: Timestamps نقاط زمانی یک ویدیو را ذخیره می‌کند و بدون وجود ویدیوی والد، رکوردهای آن بی‌معنی خواهند بود. بنابراین Timestamp یک موجودیت ضعیف است. کلید خارجی VideoID به Videos(VideoID) متصل است.

PlaylistVideos وابسته به Videos و Playlists است: این جدول رابطه‌ای است که لیست پخش و ویدیوها را به یکدیگر متصل می‌کند. وجود رکورد در این جدول بدون وجود هر دو Playlist و Video بی‌معنی است. بنابراین PlaylistVideos یک موجودیت ضعیف است. کلیدهای خارجی PlaylistID به Playlists(PlaylistID) و VideoID به Videos(VideoID) متصل هستند.

InChatMessages وابسته به Streams است: پیام‌های چت تنها در استریم‌های زنده معنا پیدا می‌کنند. بنابراین InChatMessages یک موجودیت ضعیف است. کلید خارجی StreamID به Streams(StreamID) متصل است.

انواع اتریبیوت‌ها (فیلدها):

اتریبیوت‌های ساده (Simple Attributes): این اتریبیوت‌ها غیرقابل تفکیک هستند و نمی‌توان آنها را به اجزای کوچک‌تر تقسیم کرد.

مثال‌هایی از آن‌ها عبارت‌اند از: FirstName و LastName و Username در جدول Users و Region در جدول Channels و...

اتریبیوت‌های مرکب (Composite Attributes): اتریبیوت‌هایی هستند که می‌توانند به اجزای کوچک‌تر تقسیم شوند. مانند Name به صورت ترکیب FirstName و LastName در جدول Users.

اتریبیوت‌های مشتق‌شده یا استنتاجی (Derived Attributes): مقادیری هستند که می‌توان آنها را از روی سایر داده‌ها محاسبه کرد. این اتریبیوت‌ها نیازی به ذخیره‌سازی در پایگاه داده ندارند اما ممکن است برای بهبود عملکرد ذخیره شوند.

مثالهایی از آن‌ها عبارت‌اند از: TotalViews و SubscribersCount و VideosCount در جدول Channels. همچنین VideoCount در جدول Playlists نیز مشابه است. این مقادیر می‌توانند از روابط مرتبط محاسبه شوند، مثلاً می‌توانیم این فیلدها را حذف کنیم و در زمان نیاز، از کوئری‌های COUNT() استفاده کنیم.

اتریبیوت‌های چند مقدار (Multi-Valued Attributes): اتریبیوت‌هایی هستند که می‌توانند چندین مقدار داشته باشند. این اتریبیوت‌ها به صورت مستقیم در طراحی رابطه‌ای ذخیره نمی‌شوند و باید از جداول کمکی استفاده شود. مثال‌هایی از آن‌ها عبارت‌اند از: ویدیوهای موجود در یک لیست پخش (Videos ↔ Playlists)، کاربران دنبال‌کننده یک کانال (Users ↔ Channels).

انواع روابط از نظر کاردینالیتی:

روابط یک به یک (One-to-One): مثالی از این نوع رابطه در دیتابیس: هر کاربر می‌تواند یک پلن پریمیوم داشته باشد و یک پلن پریمیوم فقط برای یک کاربر است.

روابط یک به چند (One-to-Many): مثال‌هایی از این نوع رابطه در دیتابیس:

Users → Channels: یک کاربر می‌تواند چندین کانال داشته باشد.

Channels → Videos: یک کانال می‌تواند چندین ویدیو داشته باشد.

Videos → Timestamps: یک ویدیو می‌تواند چندین تایم‌استمپ داشته باشد.

Videos → Comments: یک ویدیو می‌تواند چندین کامنت داشته باشد.

Comments → Reactions: یک کامنت می‌تواند چندین واکنش داشته باشد.

روابط چند به چند (Many-to-Many): مثال‌هایی از این نوع رابطه در دیتابیس:

Users ↔ Channels: کاربران می‌توانند چندین کانال را دنبال کنند و یک کانال ممکن است چندین کاربر دنبال‌کننده داشته باشد.

این رابطه توسط جدول Subscriptions مدیریت می‌شود.

Playlists ↔ Videos: یک لیست پخش می‌تواند شامل چندین ویدیو باشد و یک ویدیو

می‌تواند در چندین لیست پخش وجود داشته باشد.

این رابطه توسط جدول PlaylistVideos مدیریت می‌شود.