

תאריך הגשה	אתר	מס' קבוצה
12/11/2020	https://www.youtube.com	35
חברי הצוות - מספרי ת.ז.		
315844308	318470242	305730897

פרויקט בסיסי נתונים

רקע כללי על הארגון והאתר

עפ"י הערכות, YouTube הוא מנוע החיפוש השני בעולם מבחינת כמות משתמשים, אחרי Google, ומבחינת כמות מבקרים, האתר השלישי בעולם, אחרי Google ו-Facebook. למרות ש-Google אינה מפרסמת את תרומתה של חברת הבת שלה לשורת הרווח, לא קשה לראות את ההיגיון העסקי ברכישה הזו משנת 2006.

הצהרת הערך של YouTube היא "לתת לאנשים קול ולאפשר להם לראות עולם". ואכן, YouTube הוא מקור מרכזי עבור אנשים רבים לצריכת תכנים רבים, כשהערכות מדברות על 500 שעות סרטונים המועלים בכל יום וסך של מיליארד שעות תוכן הנצפה בכל יום. פתרונות האופטימיזציה, האחסון והתפעול של כמות וידאו כזאת היא גדולה ומורכבת, ומהווה מקור עיקרי למבנה הוצאות החברה.

מודל ההכנסות של האתר הוא פרסומות המופיעות לפני הצפייה בסרטון המבוקש. האתר אוסף מידע על המשתמשים בשירותי הצפייה, על מנת להקרין לצופים פרסומות ממוקדות יותר שיותר סביר שיובילו לרכישות, זאת בשונה מהטלוויזיה או בשלטי החוצות.

בנוסף האתר מוכר שירות פרמיום ללא פרסומות כמקור הכנסה נוסף אך לא עיקרי.

האתגרים שעומדים בפני האתר הינם התמודדות עם נפח מידע ההולך וגובר בכל העולם ומציאת דרכים להשתמש בו כדי לשפר את השירות ולהתאים את הפרסומות, ומענה לעלייה בביקוש לשירותי סטרימינג עקב משבר הקורונה.

מתחרות עיקריות מולן עומדת YouTube בעולם הן לדוגמה ¹Facebook ו-²Netflix. אך בארץ, אין אתר דומה בסדר גודל ובפעילות ל-YouTube.

הבדלים:

- בפייסבוק יש מגוון גדול של סוגי תכנים, בעוד יוטיוב מתמקד בסרטונים.
- בנטפליקס מודל הרווח של האתר הוא אך ורק רכישת שירותי מנוי הנותנים חוויה איכותית, בעוד ב-YouTube אין צורך לשלם על מרבית השירותים אך הם כרוכים בצפייה בפרסומות.

¹ <https://www.facebook.com>

² <https://www.netflix.com>

תהליך עסקי

קהל היעד של החברה הוא כמובן בראש ובראשונה צרכני המדיה, אולם השירותים שהיא מציעה להם, YouTube Premium ו-YouTube Music, אינם מהווים ליבת הפעילות העסקית. המטרה של פעילויות אלה להמשיך למשוך קהל משתמשים רחב ככל האפשר בכדי לתת מענה לליבת העיסוק של החברה – שירותי הפרסום שלה, כשצרכניותיה – יוצרי התוכן וחברות המדיה – הם הלקוחות האמיתיים.

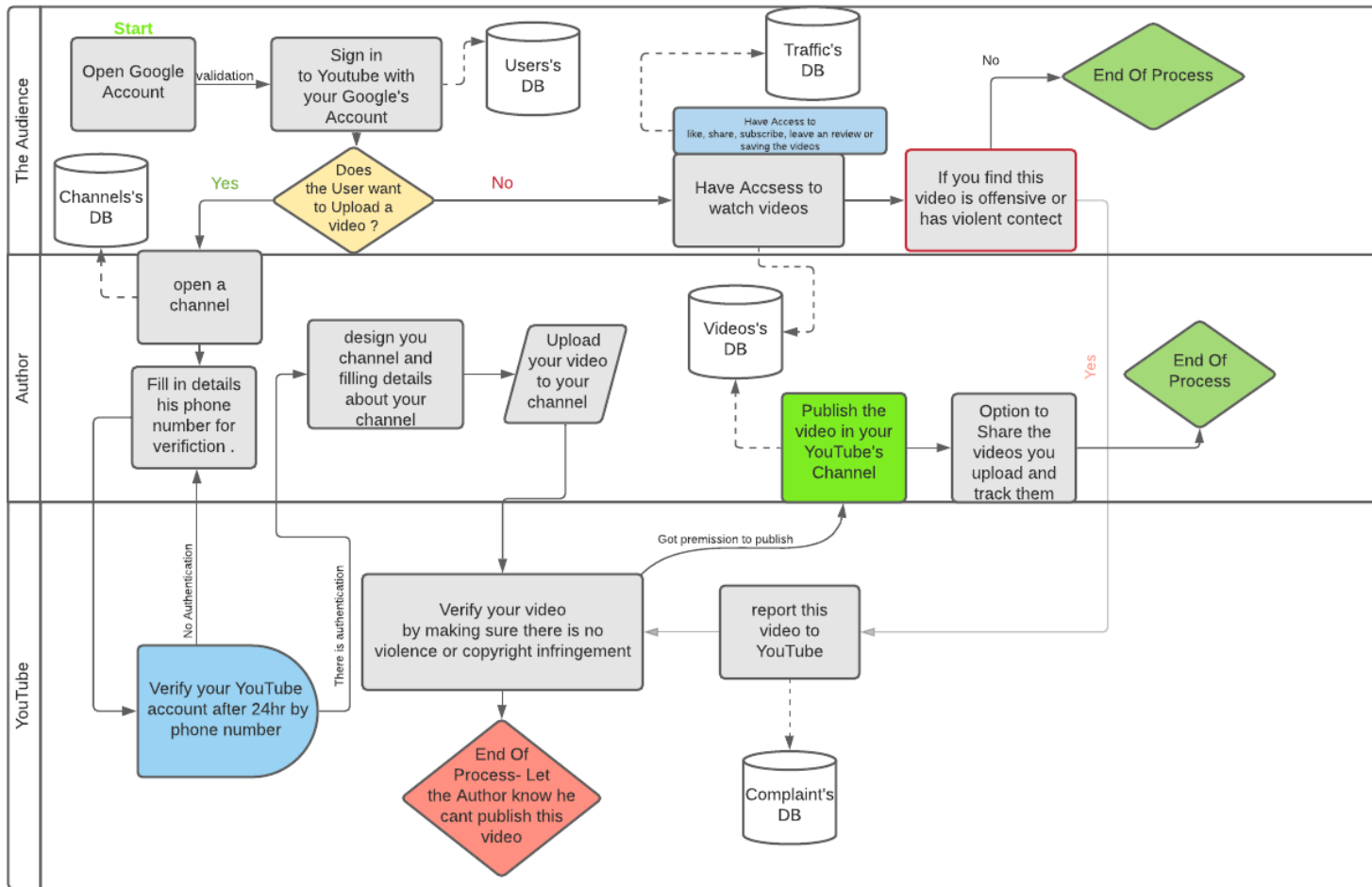
מבחינת המפרסמים, בהשוואה לתהליך המסורתי, לפרסום המקוון יתרון כפול: הן בחשיפת קהל גדול יותר וגלובלי לתכנים, והן פרסום ממוקד יותר לצרכנים פוטנציאליים, זאת באמצעות היסטורית השימוש שנצטברה על העדפות לקוחותיהם. דא עקא, לפרסום מקוון גם עלויות גדולות יותר ומורכבות טכנית גדולה יותר ביצירה ומכאן גם החיסרון המרכזי. מצד החברה, היתרון המרכזי נובע שיתוף הפעולה והסינרגיה עם חברת האם Google, והנגישות הגדולה למידע. אך כאמור החיסרון המרכזי הוא היות המידע כבד יותר ומורכב יותר לעיבוד וניתוח (מידע לא מובנה).

במסגרת התהליך העסקי של העלאה, שמירה ומעקב אחר סרטונים וביקורת עליהם, YouTube אוספת מידע מגוון לאורך השלבים השונים. הרמה הבסיסית של משתמשי האתר מוגדרים "אורחים" ולהם הרשאות פעילות בסיסיות בלבד, המסתכמות בצפייה בתכנים החינמיים. אלו מהם המעוניינים גם בהרשאות לפעילות אקטיבית יותר (סימון "אהבתי", יצירת רשימות השמעה ו"מנוי" לערוצים, השארת תגובות ועוד). נדרשים לפתוח חשבון Google אשר מהווה חשבון הפעילות בחברה. במסגרת ההרשמה הלקוח מתבקש לציין שם (פרטי ומשפחה) כתובת דוא"ל (רצוי גם של Google), תאריך לידה ולבחור סיסמא. טלפון ומגדר הם רשות בשלב זה. מכאן גם ניתן גם לרכוש את השירותים המשלימים בתשלום, שפורטו לעיל. כאן גם נשמרים נתוני הפעילות השונים.

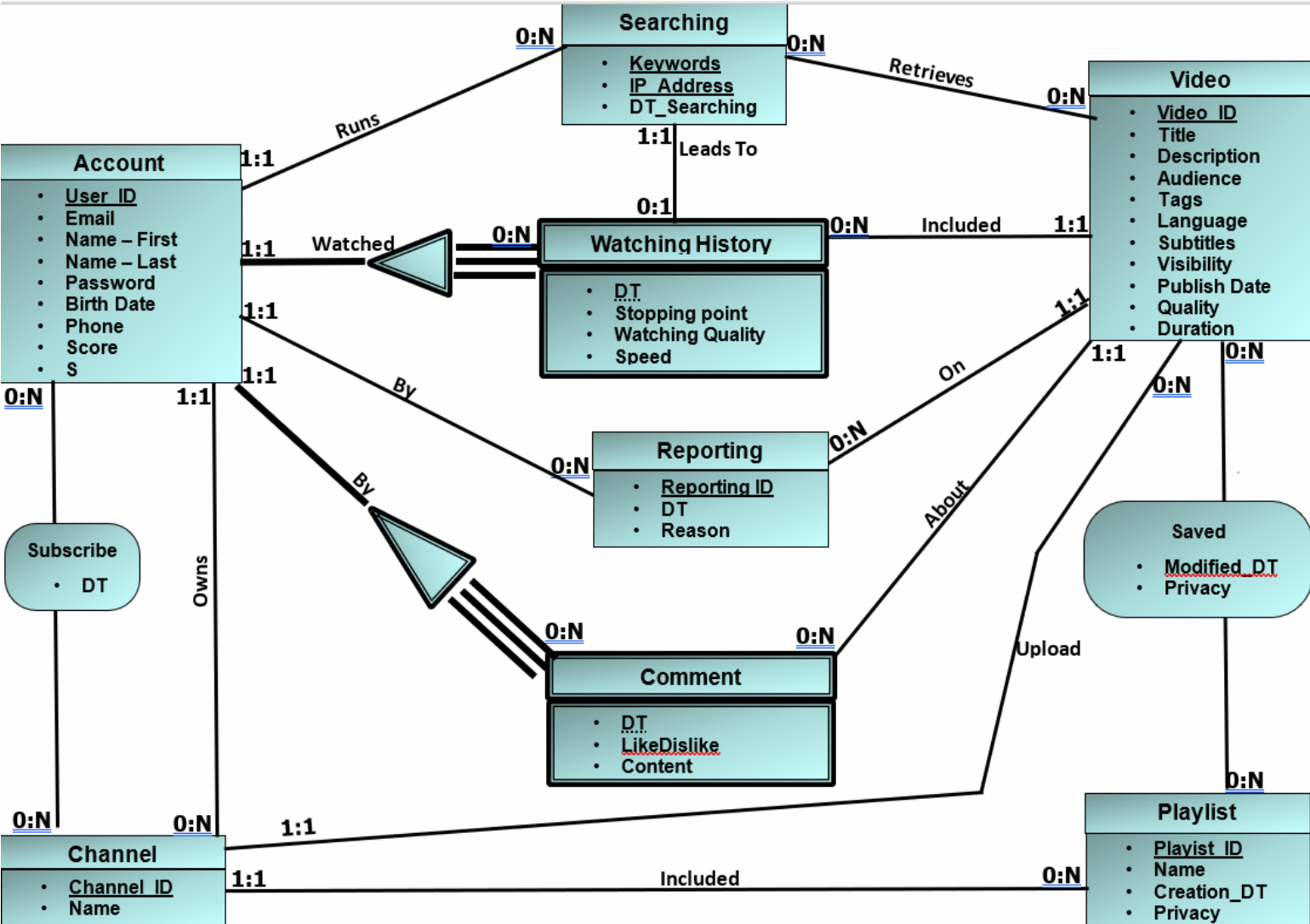
דרגת מעורבות שלישית מוגדרת פתיחת "ערוץ" עבור המשתמשים בעלי החשבון, ומיועדת להעלאה, שמירה ומעקב אחר הסרטונים במה שנקרא YouTube Studio. כאן מספר הטלפון נדרש לאימות בעל החשבון, ואחרי 24 שעות ניתן לפרסם. בשלב הבא והאחרון, לערוצים מסוימים ניתנת האפשרות לנצל את YouTube ואת Google (באמצעות Google AdSense) לצרכי מונטיזציה ע"י הוספת פרסומות, צ"ט עם אפשרות לגיוס תרומות ("סופר-צ'ט") והוספת פונקציה של "חברות בערוץ" למנויים המעוניינים בתכנים נוספים בתשלום.

כאמור מידע רב נדרש להשלמת שלבי התהליך. הוא, בתוספת עם הרגלי הגלישה שנאספים ע"י Google, משמש את אלגוריתמי הבינה המלאכותית לצורך פרסום יעיל יותר ("טירגוט" הצופה) והתאמה אישית של הפרסומות, ומבחינה זו - אין מידע מיותר.

תרשים זרימה של התהליך העסקי -



עיצוב קונספטואלי (תרשים ישויות-קשרים)



הנחות-

מרכיב	הנחות	הערות נוספות
Entity : History <ul style="list-style-type: none"> Watching_DT Stopping Point Quality Language Subtitles Speed 	<ul style="list-style-type: none"> היסטוריית צפייה – משתמש יכול לצפות בזמן נתון בסרטון אחד ויחיד. צירוף המזהים זמן הצפייה, ומזהה המשתמש, או ID, יזהה את הצפייה בסרטון ברגע נתון. מצירוף הישויות "History" ו- "Videos", נוכל לחשב את כמות הצפיות לסרטון. 	היסטורית הצפייה מאפשרת לנו לקבל מידע לגבי הגדרות שהמשתמש בחר לצפות באותו סרטון.
הקשר בין "Account" – "Channel" Subscribes <ul style="list-style-type: none"> DT 	<ul style="list-style-type: none"> בנוסף להיותו בעל ערוץ, משתמש יכול להיות מנוי לכמה ערוצים אחרים (או לא בכלל). המעקב אחר כמות המנויים שיש לכל ערוץ נעשית בעזרת התכונה "Subscribe" כך שניתן לדעת את תאריך הוספת מנוי לאותו ערוץ 	
Entity : Comment	<ul style="list-style-type: none"> בכדי לעקוב אחר זהות המגיב ומושא התגובה נגדיר את הישות כישות חלשה ובכך נאפשר חזרתיות על מתן התגובה (נציין כי: בזמן נתון ניתן להגיב על סרטון יחיד). 	בנוסף הישות תתאר האם בוצע like או dislike ואת תוכן התגובה שנעשית על אותו סרטון.
Entity: Reporting	<ul style="list-style-type: none"> Reporting תחשב ישות עצמאית ולא תכונה משום שאנו רוצים לבצע מעקב אחר דיווחים מפי המשתמש על סרטון מסוים. קיימת אופציה לבצע דיווח גם על תגובה בסרטון אך החלטנו להזניח בטבלה. 	הישות תקושר או לסרטון או לתגובה מסוימת (מהסרטון) בעזרת ID מתאים, כך גם הקרדינליות מציינת זאת.
Entity: Playlist	<ul style="list-style-type: none"> לכל משתמש יש אופציה לפתוח מספר playlist משלו עם אופציות כמו שם, רמת פרטיות הצפייה. כך שלכל playlist יש ID משלו ותאריך יצירת הplaylist. 	הישות תהיה עצמאית משום שלכל playlist יש זיהוי על ידי ID.
Feature :Saved <ul style="list-style-type: none"> Privacy Modified_DT 	<ul style="list-style-type: none"> התכונה נותנת לנו אופציה לעקוב על תאריך שינוי הפלייליסט, מתי המשתמש שינה את הפלייליסט שברשותו, איך הוא ביצע את השינוי(איזה סרטונים הוסיף) ואיזו רמת פרטיות תהיה לסרטונים. 	

עיצוב לוגי (מודל נתונים טבלאי)

מודל טבלאי

Accounts (User ID, Email, Name-first, Name-last, Password, Birth Date, Phone, Score, S)

Subscribes (User ID(ACCOUNTS), Channel ID(CHANNELS), DT)

Channels (Channel ID, User ID(ACCOUNTS), Name)

History (User ID(ACCOUNTS), DT, Stopping point, Watching_Quality, speed, {Keywords, IP_Address} (Searching), Video ID (VIDEOS))

Reportings (Reporting ID, DT, Reason, User ID(ACCOUNTS), Video ID(VIDEOS))

Comments (User ID(ACCOUNTS), DT, LikeDislike, Content, Video ID(VIDEOS))

Playlists (Playlist ID, Name, Creation_DT, Privacy, Channel ID(CHANNELS))

Saved (Video ID(Videos), Playlist ID(Playlists), Privacy, Modified_DT)

Videos (Video ID, Title, Description, Audience, Tags, Language, Subtitles, Visibility, Publish date, Quality, Duration, Channel ID(CHANNELS))

Comments (User ID(ACCOUNTS), DT, LikeDislike, Content, Video ID(VIDEOS))

Searching (Keywords, IP Address, DT_Searching, User ID(ACCOUNTS))

Retrieves ({Keywords, IP Address} (Searching), Video ID (VIDEOS))

פירוט שדות בסיס הנתונים והערכת גודלם

Table	Field	Data Type	Size (Bytes)
Accounts	<u>User ID</u>	Nvarchar(50)	50
	Email	Varchar(50)	40
	Name – First	Varchar(20)	20
	Name - Last	Varchar(20)	20
	Password	Varchar(20)	20
	Birth Date	Date	3
	Phone	Char(10)	10
	Total Record Size (Bytes)		164
History	<u>User ID</u>	Nvarchar(50)	50
	<u>DT</u>	DateTime	8
	Video ID	Nvarchar(50)	50
	Stopping Point	Time	5
	Watching_Quality	Varchar(20)	20
	Language_Subtitles	Varchar(20)	20
	Speed	Real	4
	Total Record Size (Bytes)		152
Videos	<u>Video ID</u>	Nvarchar(50)	50
	Title	Varchar(500)	500
	Description	Varchar(500)	500
	Audience	Varchar(50)	50
	Tags	Varchar(150)	150
	Language	Varchar(20)	20
	Subtitles	Varchar(20)	20
	Visibility	Varchar(20)	20
	PublishDate	Datetime	8
	Quality	Varchar(20)	20
	Duration	Time	5
	Channel ID	Nvarchar(50)	50
	Total Record Size (Bytes)		1392
Reporting	<u>Reporting ID</u>	Nvarchar(10)	30
	DT	DateTime	8
	Reason	Varchar(150)	150
	User ID	Nvarchar(50)	50
	Video ID	Nvarchar(50)	50
	Total Record Size (Bytes)		288
Comments	<u>Video ID</u>	Nvarchar(50)	50
	<u>User ID</u>	Nvarchar(50)	50
	<u>DT</u>	DateTime	8
	Content	Varchar(500)	500
	Likedislike	bit	10
	Total Record Size (Bytes)		614
Channels	<u>Channel ID</u>	Nvarchar(50)	50
	User ID	Nvarchar(50)	50
	Name	Varchar(100)	100
	Total Record Size (Bytes)		200

Playlists	<u>Playlist ID</u>	Nvarchar(50)	50
	Name	Varchar(100)	100
	Creation_DT	DateTime	8
	Privacy	Varchar(20)	20
	Channel ID	Nvarchar(50)	50
	Total Record Size (Bytes)		228
Subscribe	<u>User ID</u>	Nvarchar(50)	50
	<u>Channel ID</u>	Nvarchar(50)	50
	DT	DateTime	8
	Total Record Size (Bytes)		108
Saved	Video ID	Nvarchar(50)	50
	Playlist ID	Nvarchar(50)	50
	Privacy	Varchar(20)	20
	Modified_DT	DateTime	8
	Total Record Size (Bytes)		128

עיצוב פיסי (בסיס נתונים פעיל על שרת RDBMS)

אילוצי מרחב ערכים

אילוצי בדיקה (CHECK):

- **אילוץ על Password:** הוספנו אילוץ על השדה Password בטבלה. על הסיסמא להיות באורך מינימלי של 8 תווים כדי לדרוש סיסמאות בדרגת מורכבות מסוימת.

```
Password Varchar(20) CONSTRAINT Ck_Password CHECK (len>Password) >= 8),
```

- **אילוץ על Phone:** הוספנו אילוץ על השדה Phone בטבלה. מספר הטלפון תקני הוא באורך 10 תווים, כך שכל תו בין הספרה 0 ל 9. התייחסנו למספרים מקומיים בלבד.

```
Phone Char(10) CONSTRAINT Ck_Phone CHECK (Phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))
```

טבלאות חיפוש (Lookup Tables):

- **טבלת חיפוש ל-History, Videos:** הוספנו טבלת חיפוש עבור איכויות הצפייה השונות האפשריות.

```
CREATE TABLE Qualities (  
    Quality Varchar(20) NOT NULL PRIMARY KEY)
```

```
INSERT INTO Qualities  
VALUES ('2160p'),  
        ('1440p'),  
        ('1080p'),  
        ('720p'),  
        ('480p'),  
        ('360p'),  
        ('240p'),  
        ('144p')
```

```
ALTER TABLE Videos  
ADD CONSTRAINT FK_qual FOREIGN KEY (Quality)  
REFERENCES Qualities (Quality)
```

```
CONSTRAINT FK_qual2 FOREIGN KEY (Watching_Quality)  
REFERENCES Qualities (Quality)
```

- **טבלת חיפוש ל-Comments:** הוספנו טבלת חיפוש עבור אפשרויות התגובה השונות.

```
CREATE TABLE Qualities (
    Quality Varchar(20) NOT NULL PRIMARY KEY)

INSERT INTO LikeDislike
VALUES ('Like'),
      ('Dislike')

Constraint FK_LDL FOREIGN KEY ( LikeDislike )
REFERENCES LikeDislike (LikeDislike)
```

- **טבלת חיפוש ל-Comments:** הוספנו טבלת חיפוש עבור אפשרויות התגובה השונות.

```
CREATE TABLE Privacies (
    Privacy Varchar(20) NOT NULL PRIMARY KEY)

INSERT INTO Privacies
VALUES ('Private'),
      ('Unlisted'),
      ('Public')

CREATE TABLE Saved (
    Constraint Fk_Priv FOREIGN KEY (Privacy)
REFERENCES Privacies (Privacy))

CREATE TABLE Playlists (
    Constraint Fk_Priv2 FOREIGN KEY (Privacy)
REFERENCES Privacies (Privacy))
```

SCRIPT לבניית הטבלאות

```
CREATE TABLE Accounts (
    UserID Nvarchar(50) NOT NULL PRIMARY
    KEY,
    Email Varchar(50) CONSTRAINT Ck_Mail CHECK (Email LIKE '%@%'),
    Name_First Varchar(20) NOT NULL,
    Name_Last Varchar(20) NOT NULL,
    Password Varchar(20) CONSTRAINT Ck_Password CHECK (len>Password) >= 8),
    BirthDate Date NOT Null,
    Phone Char(10) CONSTRAINT Ck_Phone CHECK (Phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

CREATE TABLE Channels (
    ChannelID Nvarchar(50) NOT NULL PRIMARY KEY,
    UserID Nvarchar(50) NOT NULL,
    Name Varchar(100) NOT NULL,

    Constraint Fk_User FOREIGN KEY (UserID)
REFERENCES Accounts (UserID))
```

```

CREATE TABLE Subscribes (
  UserID Nvarchar(50) NOT NULL,
  ChannelID Nvarchar(50) NOT NULL,
  DT DATETIME Not Null

  Constraint PK_subscribes PRIMARY KEY (UserID,ChannelID),
  Constraint Fk_User2 FOREIGN KEY (UserID)
                                     REFERENCES Accounts (UserID),
  Constraint Fk_Cnl FOREIGN KEY (ChannelID)
                                     REFERENCES Channels (ChannelID))

```

```

CREATE TABLE Qualities (
  Quality Varchar(20) NOT NULL PRIMARY KEY)

```

```

CREATE TABLE Videos (
  VideoID Nvarchar(50) NOT NULL PRIMARY KEY,
  Title Varchar(500) Not Null,
  Description Varchar(500) Null,
  Audience Varchar(50) NOT NULL,
  Tags Varchar(150) Null,
  Language Varchar(20) Null,
  Subtitles Varchar(20) Null,
  Visibility Varchar(20) Null,
  PublishDate DATETIME NOT NULL,
  Quality Varchar(20) NOT NULL,
  Duration TIME NOT NULL,
  ChannelID Nvarchar(50) NOT NULL)

```

```

CREATE TABLE History (
  UserID Nvarchar(50) NOT NULL,
  VideoID Nvarchar(50) NOT NULL,
  DT DATETIME Not Null,
  Stopping_Point TIME Not Null,
  Watching_Quality Varchar(20) Not Null,
  Language_Subtitles Varchar(20) Null,
  Speed Real Not Null

  Constraint PK_History PRIMARY KEY (UserID,DT),
  Constraint Fk_User3 FOREIGN KEY (UserID)
                                     REFERENCES Accounts (UserID),
  Constraint Fk_VID5 FOREIGN KEY (VideoID)
                                     REFERENCES VIDEOS(VideoID),
  CONSTRAINT FK_qual2 FOREIGN KEY (Watching_Quality)
                                     REFERENCES Qualities (Quality))

```

```

ALTER TABLE Videos
ADD CONSTRAINT FK_qual FOREIGN KEY (Quality)
                                     REFERENCES Qualities (Quality)

```

```

CREATE TABLE Privacies (
  Privacy Varchar(20) NOT NULL PRIMARY KEY)

```

```

CREATE TABLE Playlists (
    PlaylistID Nvarchar(50) NOT NULL PRIMARY KEY,
    Name Varchar(100) NOT NULL,
    Creation_DT DateTime NOT NULL,
    Privacy Varchar(20) NOT NULL,
    ChannelID Nvarchar(50) NOT NULL

    Constraint Fk_Cn14 FOREIGN KEY (ChannelID)
        REFERENCES Channels (ChannelID),
    Constraint Fk_Priv2 FOREIGN KEY (Privacy)
        REFERENCES Privacies (Privacy))

CREATE TABLE Saved (
    VideoID Nvarchar(50) NOT NULL,
    PlaylistID Nvarchar(50) NOT NULL,
    Privacy Varchar(20) NOT NULL,
    Modified_DT DATETIME NOT NULL,

    Constraint PK_SAVED PRIMARY KEY (VideoID, PlaylistID),
    Constraint Fk_plst FOREIGN KEY (PlaylistID)
        REFERENCES Playlists (PlaylistID),
    Constraint Fk_VID FOREIGN KEY (VideoID)
        REFERENCES Videos (VideoID),
    Constraint Fk_Priv FOREIGN KEY (Privacy)
        REFERENCES Privacies (Privacy))

Create Table LikeDislike (
    LikeDislike Varchar(10) NOT NULL PRIMARY KEY)

CREATE TABLE Comments (
    VideoID Nvarchar(50) NOT NULL,
    UserID Nvarchar(50) NOT NULL,
    DT DATETIME NOT NULL,
    Content Varchar(500) NOT NULL,
    LikeDislike Varchar(10) Null,

    PRIMARY KEY (VideoID, UserID, DT),
    Constraint Fk_VID2 FOREIGN KEY (VideoID)
        REFERENCES Videos (VideoID),
    Constraint Fk_User4 FOREIGN KEY (UserID)
        REFERENCES Accounts (UserID),
    Constraint FK_LDL FOREIGN KEY ( LikeDislike )
        REFERENCES LikeDislike (LikeDislike))

CREATE TABLE Reportings (
    ReportingID Nvarchar(10) NOT NULL PRIMARY KEY,
    DT DATETIME NOT NULL,
    Reason Varchar(150) Null,
    UserID Nvarchar(50) NOT NULL,
    VideoID Nvarchar(50) NULL

    Constraint Fk_VID3 FOREIGN KEY (VideoID)
        REFERENCES Videos (VideoID),
    Constraint Fk_User5 FOREIGN KEY (UserID)
        REFERENCES Accounts (UserID))

```

SCRIPT להזנת רשומות

```
INSERT INTO Accounts
VALUES ('bcMVpCRxwcqaT1wiWM_ocg', 'alaana@post.bgu.ac.il', 'Bar',
'Ganor', '95731297', '1992-06-15', '0528000009'),
      ('TCMk91_YW-xjeHEHCXZ6AQ', 'idogola@post.bgu.ac.il',
'Ido', 'Golan', '315842ty', '1991-02-04', '0544757771'),
      ('gqemvxKnPPQEnmK4_3nGDQ', 'ronhaza@post.bgu.ac.il',
'Ron', 'Hazan', '147951go', '1986-05-17', '0534300264'),
      ('UCgqemvxKnPPQEnmK4_3nGDQ', 'lowy@post.bgu.ac.il',
'Neta', 'Levy', '991438ev', '1956-06-03', '0503642311'),
      ('UCTCMk91_YW-xjeHEHCXZ6AQ', 'galyav@post.bgu.ac.il',
'Galy', 'Aviv', '142or578', '1971-04-01', '0523113390')

INSERT INTO Channels
VALUES ('cdERqDWyzbhiJ20KE_dvr', 'bcMVpCRxwcqaT1wiWM_ocg',
'HeyImNotAlice'),
      ('RQJi63_RS-hisOPSBTC6US', 'TCMk91_YW-xjeHEHCXZ6AQ',
'AhnaIdT101'),
      ('ehKDtLSypfymaU8QPG_aiv', 'gqemvxKnPPQEnmK4_3nGDQ',
'StatQuest with Josh Starmer'),
      ('UCtYLUTtgS3k1Fg4y5tAhLbw', 'UCgqemvxKnPPQEnmK4_3nGDQ',
'929'),
      ('UCoqzhxjJIPoJbeuAsVnvM5Q', 'UCTCMk91_YW-xjeHEHCXZ6AQ',
'Q-dance')

INSERT INTO Privacies
VALUES ('Private'),
      ('Unlisted'),
      ('Public')

INSERT INTO Qualities
VALUES ('2160p'),
      ('1440p'),
      ('1080p'),
      ('720p'),
      ('480p'),
      ('360p'),
      ('240p'),
      ('144p')

INSERT INTO Videos
VALUES ('1BYrHIBs6ro', 'Tomorrowland 2020', 'Best Songs, Remixes &
Mashups - Festival Mix 2020', 'Yes, made for kids', NULL, 'English', NULL,
'Private', '2010-08-13 13:23:44', '1080p', '01:01:19',
'cdERqDWyzbhiJ20KE_dvr'),
      ('aTTx5WsswPM', 'Feeling Me #5', 'Dua Lipa, Coldplay,
Martin Garrix & Kygo, The Chainsmokers', 'Yes, made for kids',
'#SummerMusicMix', 'Spanish', NULL, 'Public', '2006-11-23 11:16:21', '720p',
'00:22:35', 'RQJi63_RS-hisOPSBTC6US'),
      ('v2qbbq2QCZkc', 'Jedi Music for relaxation', 'Feel the
Force and focus', 'Not for kids', NULL, 'English', NULL, 'Private', '1998-04-23
17:58:21', '480p', '00:33:59', 'ehKDtLSypfymaU8QPG_aiv'),
      ('kRrIAqCdjCI', 'ZEN MUSIC FOR BALANCE AND RELAXATION',
'incredible assistance in reaching a deep meditation', 'Not for kids', NULL,
'French', NULL, 'Unlisted', '1998-01-01 22:17:06', '360p', '01:05:01',
'UCtYLUTtgS3k1Fg4y5tAhLbw'),
      ('iG_lNuNUVd4', 'om mani padme hum', 'mantra', 'Not made
for kids', NULL, 'English', NULL, 'Public', '2011-06-04 17:58:21', '360p',
'00:41:27', 'UCoqzhxjJIPoJbeuAsVnvM5Q')
```

```

INSERT INTO History
VALUES ('bcMVpCRxwcqaT1wiWM_ocg', '1BYrHIBs6ro', '2020-11-11 13:23:44',
'00:05:33', '360p', NULL, 1.5),
('TCMk91_YW-xjeHEHCXZ6AQ', 'aTTx5WsswPM', '2020-11-09
15:45:21', '00:01:33', '240p', 'Hebrew', 2.0),
('gqemvxKnPPQEnmK4_3nGDQ', 'v2qbq2QCZkc', '2020-11-11
11:12:01', '00:10:11', '720p', 'Arabic', 1.0),
('UCgqemvxKnPPQEnmK4_3nGDQ', 'kRrIAqCdjCI', '2020-10-29
14:56:59', '00:02:10', '480p', NULL, 1.25),
('UCTCMk91_YW-xjeHEHCXZ6AQ', 'iG_lNuNUVd4', '2012-12-25
16:44:12', '00:13:52', '480p', 'English', 1.75)

```

```

INSERT INTO Playlists
VALUES ('Uj4Mg3FVGgSV38Cxo5r', 'Meditations', '2020-11-11 14:23:00',
'Public', 'cdERqDWyzbhiJ20KE_dvr'),
('oW-4xxER15GyuFq', 'Podcast Episodes', '2020-11-09
16:01:05', 'Public', 'RQJi63_RS-hisOPSBTC6US'),
('Oyt32r4alo6S4Hs', 'Developing Mindfulness Workshop',
'2020-11-11 11:15:07', 'Public', 'ehKDtLSypfymaU8QPG_aiv'),
('UrQcS6gN_NluRpi', 'Original Songs', '2020-10-29
15:02:59', 'Private', 'RQJi63_RS-hisOPSBTC6US'),
('Q6xK4tZEIbIvAAEEykp', 'High Throughput Sequencing',
'2012-12-25 16:59:17', 'Unlisted', 'RQJi63_RS-hisOPSBTC6US')

```

```

INSERT INTO Saved
VALUES ('1BYrHIBs6ro', 'Uj4Mg3FVGgSV38Cxo5r', 'Public', 1),
('aTTx5WsswPM', 'oW-4xxER15GyuFq', 'Public', 5),
('v2qbq2QCZkc', 'Oyt32r4alo6S4Hs', 'Public', 7),
('kRrIAqCdjCI', 'UrQcS6gN_NluRpi', 'Private', 20),
('iG_lNuNUVd4', 'Q6xK4tZEIbIvAAEEykp', 'Unlisted', 0)

```

```

INSERT INTO Subscribes
VALUES ('bcMVpCRxwcqaT1wiWM_ocg', 'cdERqDWyzbhiJ20KE_dvr', '2020-11-12
11:13:00'),
('bcMVpCRxwcqaT1wiWM_ocg', 'RQJi63_RS-hisOPSBTC6US',
'2020-12-01 15:20:00'),
('bcMVpCRxwcqaT1wiWM_ocg', 'UCoqzhxjJIPOJbeuAsVnvM5Q',
'2020-09-15 08:35:02'),
('UCgqemvxKnPPQEnmK4_3nGDQ', 'cdERqDWyzbhiJ20KE_dvr',
'2011-05-17 18:42:07'),
('UCgqemvxKnPPQEnmK4_3nGDQ', 'RQJi63_RS-hisOPSBTC6US',
'2008-07-14 12:44:05')

```

```

INSERT INTO LikeDislike
VALUES ('Like'),
('Dislike')

```

```

INSERT INTO Comments
VALUES ('1BYrHIBs6ro', 'bcMVpCRxwcqaT1wiWM_ocg', '2020-11-13 11:13:00',
'very important video, KUDOS', 'Like'),
('aTTx5WsswPM', 'bcMVpCRxwcqaT1wiWM_ocg', '2020-11-09
19:15:00', 'Not cool dude', 'Dislike'),
('v2qbq2QCZkc', 'bcMVpCRxwcqaT1wiWM_ocg', '2020-11-15
16:17:00', 'Respect.', 'Like'),
('kRrIAqCdjCI', 'UCgqemvxKnPPQEnmK4_3nGDQ', '2020-11-18
10:19:00', 'i like what u do!', NULL),
('iG_lNuNUVd4', 'UCTCMk91_YW-xjeHEHCXZ6AQ', '2020-11-19
12:21:00', 'That looks dangerous!', 'Dislike')

```

```

INSERT INTO Reportings
VALUES ('nk2CQITm', '2020-11-09 19:15:00', 'Violent or repulsive
content', 'bcMVpCRxwcqaT1wiWM_ocg', 'aTTx5WsswPM'),
('IlzDU6aQ', '2020-11-19 12:21:00', 'Hateful or abusive
content', 'bcMVpCRxwcqaT1wiWM_ocg', 'iG_lNuNUVd4'),
('6vJHC3C2', '2001-10-05 11:07:35', 'Promotes terrorism',
'bcMVpCRxwcqaT1wiWM_ocg', 'v2qbq2QCZkc'),
('BI5JNDsA', '2015-09-04 18:33:42', 'Infringes my rights',
'bcMVpCRxwcqaT1wiWM_ocg', 'kRrIAqCdjCI'),
('7CBCW6T8', '2017-05-11 20:21:56', 'Captions issue',
'bcMVpCRxwcqaT1wiWM_ocg', '1BYrHIBs6ro')

```

SCRIPT להסרת טבלאות ואילוצים

בהתחשב במפתחות הזרים הקובעים את התלויות, סדר הסרת הטבלאות (והאילוצים בתוכן) הוא כלהלן :

```

DROP TABLE REPORTINGS
DROP TABLE COMMENTS
Drop Table LikeDislike
DROP TABLE SUBSCRIBES
DROP TABLE SAVED
DROP TABLE Playlists
Drop Table History
Drop Table Videos
Drop table Qualities
Drop Table Privacies
Drop Table Channels
Drop Table Accounts

```

שאלות

שתי שאלות SELECT ללא קינון –

- מתוך כלל הערוצים שהעלו סרטונים ב-2020, אלו ערוצים זכו למעל 50 חיפושים?

```

Select c.ChannelID, c.Name, Numnew=count(DISTINCT
v.VideoID), countsearches=count(R.IP_Adderss)
From Channels as C join Videos as v on c.ChannelID=v.ChannelID Join Retrives as
R on v.VideoID=r.VideoID
where Year(v.PublishDate)='2020'
Group by c.ChannelID, c.Name
Having count(R.IP_Adderss)>50
Order By countsearches desc

```

(אלו הערוצים הפופולריים ביותר ואפשר שייתכן שת"פ עתידי). הפלט המתקבל -

	ChannelID	Name	Numnew	countsearches
1	wmdQkr0l	Zoonder	13	204
2	t57P7xR	Gevee	8	152
3	zZUyNaRVC0A	Voolia	7	106
4	KiLZ9wcfR	Thoughtstom	5	103
5	KmsxXKg8pIJ	Skinder	2	64
6	RzilActjaLi	Ntags	2	57
7	8XdaVkJAn	Avamm	4	56

- אילו לקוחות צפו בסרטון שכותרתו "טויוטה"?

```
select A.UserID, Name= a.Name_First+' '+a.Name_Last
From History as H join Videos as v on v.VideoID=H.VideoID Join Accounts as a on
a.UserID=H.UserID
Where Title='Toyota'
Order By 1 desc
```

(ייתכן כחלק מהסכם לא חוקי כלשהו עם היבואן להעברת שמות המחפשים). הפלט -

	UserID	Name
1	xEZfZRmfP	Gabbie Scamwell
2	uVF8qXjin	Bam Perelli
3	tRXsRbmbx4K	Padraig Farrall
4	sH9IR7emFD	Nil Tomaselli
5	ruUDlmsIG	Elmer Eunson
6	o6GyUw	Ferdie Aven
7	MBmf8gXV	Cordie Mecchi
8	MBmf8gXV	Cordie Mecchi
9	mbDMfa	Dunc Graben
10	KFxvhnMf	Noam Zukieman

שתי שאילות SELECT מקוננות -

- לאיזה משתמשים יש יותר מ-2 ערוצים פעילים (שהעלו לפחות סרטון אשתקד)?

```
select a.UserID, Numchannels= count(Distinct z.channelID)
From accounts as a join Channels as c on c.UserID=a.UserID join(
Select c2.ChannelID
from channels as c2 join videos as V on v.ChannelID=c2.ChannelID
Where Year(v.PublishDate)='2020'
Group By c2.ChannelID
)
as z on z.ChannelID=c.ChannelID
Group by a.UserID
Having count(Distinct z.channelID)>2
Order by 2
```

(בהנחה וכל ערוץ משמש לפרסום תוכן מסוג מסוים – אלו משתמשים מגוונים ומעניינים).
הפלט -

	UserID	Numchannels
1	FUHSeuMn	3
2	mbDMfa	3
3	i15IDLgtsC	4
4	2lJSZxiMbk	4
5	tuyI6gWwvquG	5
6	JDyfbnL92	8

- לאיזה סרטונים "לא לילדים" יש יותר חיפושים מהמוצע?

```

Select v.VideoID, Title, NumSearches= count(distinct R.IP_Adderss)
From Videos as v join Retrives as r on v.VideoID=r.VideoID
Where Audience='Not Made for Kids'
Group by v.VideoID, Title
Having count(distinct R.IP_Adderss)>
(
Select avgviews= Cast(Count( r2.IP_Adderss)/count( distinct r2.VideoID) as
decimal(10,3))
From Retrives as r2
)
order by NumSearches

```

(חלק מפילוח אוכלוסיית הלקוחות - המשלים הוא כמובן צרכני הסרטונים "לילדים").
הפלט -

	VideoID	Title	NumSearches
1	7RujEEed19T	Mercedes-Benz	19
2	ccLVV9sgm	Mercedes-Benz	19
3	cpqNjz	Mercedes-Benz	19
4	DlaVUYWNj	BMW	19
5	dpG3L3bu	BMW	19
6	fBeESd	BMW	19
7	JFjiT7	Mercedes-Benz	19
8	jGofNyvAPk	BMW	19
9	Nvqq2m	BMW	19
10	ObgGaYQ8kCMt	Mercedes-Benz	19

- שתי שאילות מקוננות תוך שימוש במרכיבים נוספים -
- יצירת שאילתה אחת של Update או Delete מקוננת.
(נמחק, כמובן, רשימות השמעה ריקות).

```

Delete From Playlists
Where PlaylistID Not in
(
Select PlaylistID
From Saved
Group By PlaylistID
)

```

- יצירת איחוד/חיתוך/הפרש בין 2 שאילות, כך שלפחות אחת מהן מקוננת.
(משתמשים העלו למעלה מ-10 סרטונים וכתבו למעלה מ-10 תגובות. אלו למעשה משתמשים בעלי דירוג פעילות גבוה).

```

select UserID
From Comments
Group by UserID
having Count(*)>9
Intersect
(
select UserID
From Channels as c join(
Select ChannelID, videonum=Count(VideoID)
From Videos
Group By ChannelID
) as x on x.ChannelID=c.ChannelID
Group By UserID
having sum(Videonum)>9)

```

	UserID
1	i15IDLgtsC
2	JDyfbnL92
3	tuyI6gWwvquG

יישומי כלים מתקדמים

– View

```
create view Watchers as
select DAY(h.DT) as 'Day in December 2020', COUNT(distinct h.UserID) as '#
number of viewers'
from History h
where YEAR(h.DT) = 2020 and MONTH(h.DT) = 12
group by DAY(h.DT)
```

```
select *
from Watchers c
order by 2 desc
```

נציג את התפלגות מספר הצופים לפי יום בחודש שעבר. אלו הימים בעלי פוטנציאל ההכנסות הגבוה בחודש. הפלט -

	Day in December 2020	# number of viewers
1	23	21
2	10	20
3	16	19
4	5	19
5	9	18
6	1	18
7	19	18
8	17	17
9	14	17
10	22	17

- Functions

```
create function NumberViewed (@VID Nvarchar(50))
returns int
as
begin
    declare @Total int
    select @Total = COUNT(distinct h.UserID)
    from History h
    where h.VideoID = @VID
    group by h.VideoID
    return @Total
end

select v.Title, isnull(dbo.NumberViewed(v.VideoID), 0) as '# of Users viewed'
from Videos v
order by 2 desc
```

- נציג את הסרטונים הנצפים ביותר בעזרת פונקציה המקבלת מזהה הסרטון ומחשבת את מספר צופיו. אלו הסרטונים מהם נרצה ללמוד על העדפותיהם של לקוחותיהם וטעמיהם. הפלט -

	Title	# of Users viewed
1	Chevrolet	32
2	Ford	29
3	Mitsubishi	25
4	Mazda	22
5	Dodge	21
6	Toyota	21
7	Nissan	21
8	Mercedes-Benz	18
9	BMW	17
10	GMC	16

- נציג את פרטי כל הסרטונים שצפה בהם המשתמש המסוים, בעזרת פונקציה המקבלת את המזהה שלו ומחזירה את כל הטבלה. אם נרצה לאפיין את מאפייני השימוש, זו הדרך להתחיל.

```
create function Mylist (@UID Nvarchar(50))
returns table
as
    return
    select v.Title, v.Description, v.Language, v.Subtitles,
v.Duration
    from History h join Videos v on h.VideoID = v.VideoID
    where h.UserID = @UID

select *
from dbo.Mylist('1Fouru')
```

	Title	Description	Language	Subtitles	Duration
1	Mazda	benchmark viral mindshare	Lao	FALSE	01:05:01.0000000
2	Hyundai	evolve impactful niches	Dutch	FALSE	00:22:35.0000000
3	Toyota	exploit collaborative e-business	Bislama	TRUE	01:05:01.0000000
4	Saab	e-enable mission-critical e-services	English	FALSE	01:05:01.0000000
5	Volkswagen	iterate interactive vortals	Geman	FALSE	00:22:35.0000000
6	Mercedes-Benz	cultivate holistic vortals	Belarusian	FALSE	01:05:01.0000000

- Trigger

```
create trigger Retrive
on dbo.Searching for insert as
insert into dbo.Retrides
    select s.KeyWords, s.IP_Adderss, v.VideoID
    from Videos v join inserted s on v.Title=s.KeyWords
```

נרצה למלא את טבלת Retrive בצורה אוטומטית.

10 רשומות לדוגמה בטבלה הן –

	KeyWords	IP_Address	VideoID
1	Acura	139.182.160.153	7cKVe7AgWd
2	Acura	139.182.160.153	budNbDBm
3	Acura	139.182.160.153	gx4UxxxHi2Zk
4	Acura	139.182.160.153	hZEYMhp
5	Acura	139.182.160.153	IM35vPpvY
6	Acura	139.182.160.153	xMBDNn2zpqwb
7	Acura	173.23.137.224	7cKVe7AgWd
8	Acura	173.23.137.224	budNbDBm
9	Acura	173.23.137.224	gx4UxxxHi2Zk
10	Acura	173.23.137.224	hZEYMhp

- Stored Procedure

```
Create Procedure UpdateDescriptionVideo
@VideoID varchar(20) ,@new Varchar(500)
as
Update Videos Set Description=@new
Where VideoID=@VideoID
```

```
Exec UpdateDescriptionVideo '0FVQuPCBseD' , 'Chips'
```

מעלי הסרטונים בעלי אפשרות לעדכון תיאור הסרטונים שלהם. נרצה לאפשר את היכולת הזו.

לפני –

	VideoID	Title	Description	Audience	Tags	Language	Subtitles	Visibility	PublishDate	Quality	Duration	ChannelID
1	0FVQuPCBseD	Dodge	mesh real-time models	Not Made for Kids		Khmer	FALSE	Public	2019-12-28 11:50:00.000	2160P	01:05:01.0000000	W0uaYPIOXfX

ואחרי –

	VideoID	Title	Description	Audience	Tags	Language	Subtitles	Visibility	PublishDate	Quality	Duration	ChannelID
1	0FVQuPCBseD	Dodge	Chips	Not Made for Kids		Khmer	FALSE	Public	2019-12-28 11:50:00.000	2160P	01:05:01.0000000	W0uaYPIOXfX

כלים להצגת נתונים

מצורף למסמך זה קובץ PowerBI כמבוקש. להלן עיקריו:

דו"ח עסקי -

מטרת הדו"ח הוא לספק מידע מהימן ללקוחות עסקיים המבקשים למכור פרסומות להצגה באתר.

הדו"ח מציג פילוחים מעניינים לגבי היוצרים המובילים והמאפיינים של הסרטונים באתר, למשל: פילוח של האיכות היוזאלית וקהל היעד של הסרטונים הנצפים ביותר. ניתן לפלח פילוח נוסף את כל הכלים לפי מילות החיפוש.

לוח מחוונים -

לוח המחוונים הזה מיועד למנהלי האתר ומטרתו להציג את היוצרים הפעילים והערוצים הפעילים יותר באתר. למשל, הערוצים שהכי מדווחים, הערוצים שמקבלים הכי הרבה תגובות וסך הצפיות של משתמשים פעילים.

המדדים מוצגים לפי ערוצים, אך ניתן גם לפלח לפי משתמשים בלחיצה נוספת.

יישומים מתקדמים

כלי מורכב (SP או Trigger) העושה שימוש בסמן (Cursor) & שילוב מערכתי של מספר כלים -

בכדי להדגים שימוש מורכב ומתקדם ב-SQL נאחד המימוש של 2 סעיפים אלו:

```
create trigger Update_Status
on Accounts for update as

declare @uid Varchar
declare @score int
declare @s Varchar(10)
declare d cursor
                for select UserID, Score, S from Accounts
                for update

open d
fetch next from d into @uid, @score, @s;

while (@@FETCH_STATUS = 0)
begin
    if (@score > 10) set @s = 'active'
                    else set @s = 'regular'

    --print @s
    update Accounts
    set Accounts.S = @s
    from Accounts
    where current of d

    fetch next from d into @uid, @score, @s
end
close d
deallocate d
```

```

create function myScore (@UID Nvarchar(50), @Year int, @MONTH int)
returns int
as
    begin
        declare @Score int
        select @Score =
            (select COUNT(c.Content)
             from Comments c
             where UserID = @UID and YEAR(c.DT) = @Year and
MONTH(c.DT) = @MONTH) * 3
            +
            (select COUNT(r.ReporingID)
             from Reportings r
             where UserID = @UID and YEAR(r.DT) = @Year and
MONTH(r.DT) = @MONTH) * 2
        return @Score
    end

create procedure AllScores @Year int, @MONTH int
as begin
    Update Accounts set Score = dbo.myScore(Accounts.UserID, @Year, @MONTH)
end

exec AllScores 2020, 7

```

נרצה לאפיין משתמשים על בסיס היקף פעילותם באתר. לכל אחד נתבסס על כמות התגובות והדיווחים על הסרטונים שביצע, ונשייך לכל סוג פעילות ציון מתאים. הציונים ישוקללו לפי המשקולות לעיל לכדי ציון משוקלל, תגובה תזכה ליותר נקודות מאחר ובכוחה לעודד פעילות באתר. פרטים אלה יעודכנו בטבלת המשתמשים שלנו בתוספת אינדיקטור בינארי בהתאם להיקף הפעילות.

ראוי לציין כי ציונים אלה ניתנים על בסיס חודשי בלבד, זאת מאחר ונרצה לשמור על נתוני הפעילות העדכניים ולא על אלו ההיסטוריים שעלולים לאבד רלוונטיות עם הזמן. וכך, אחריות המשתמש להעביר פרמטרים מתאים לפרוצדורה. אופן הפעולה –

1. SP המקבלת חודש ושנה תעדכן (עפ"י קריאת המשתמש) את טבלת Accounts כאמור לעיל.
2. העדכון יתבצע בעזרת קריאה לפונקציה המקבלת פרמטרים דומים.
3. העדכון יפעיל Trigger לעדכון השדה S בטבלה.
4. העדכון יתבע על כל רשומה בתורה בעזרת הסמן, תוך שימוש בלולאת While.
5. ונקבל -

	UserID	Email	Name_First	Name_Last	Password	BirthDate	Phone	Score	S
1	1Fouru	eburgyn16@cbsnews.com	Everett	Burgyn	A1f47AmuC	1952-05-25	9931416647	3	regular
2	1NhuV5tsA	awanley9@mozilla.org	Angel	Wanley	MZKCU4	2001-10-08	1799223415	6	regular
3	2lJSZxiMbk	cgazeh@oracle.com	Clementius	Gaze	Fz4fUlbut	1995-12-30	3881146865	9	regular
4	3HG2wc	acoulitshm@blogspot.com	Ardine	Coultish	1Lq6MpkV	1975-02-04	8970920254	3	regular
5	3HOEEPD	lfairpoole@abc.net.au	Lucinda	Fairpool	USWCrD5pWkCe	1951-12-12	6280516877	14	active
6	3lXCuV	gterbrugge1r@irs.gov	Gardy	Terbrugge	ScTE3Tkni	1984-03-26	9017018825	15	active

דו"ח המושתת על שאילתה מקוננת מורכבת -

```
Create Function AvgViewsCn12(@Cn1ID varchar(20))
Returns Real
As Begin
Declare @AverageViews real
Select @AverageViews=(case when count(Distinct v.VideoID)=0 then 0 Else Count(
IP_Adderss)/count(Distinct h.VideoID) eND)
From History AS H JOIN Videos as v on v.VideoID= h.VideoID
Where h.VideoID in
(Select VideoID
From Videos
where ChannelID=@Cn1ID
)
Return @AverageViews
END
```

```
Create Function Net2(@Cn1ID varchar(20))
Returns int
As Begin
Declare @Net int
Select @Net=(Case When Count(*)=0 Then 0 Else
(Select Count(c.Content)
From Comments AS c JOIN Videos as v on v.VideoID= C.VideoID
Where c.VideoID in
(Select VideoID
From Videos
where ChannelID=@Cn1ID
) and LikeDislike='Like'
Group By ChannelID
)
-(Select Count(c.Content)
From Comments AS c JOIN Videos as v on v.VideoID= C.VideoID
Where c.VideoID in
(Select VideoID
From Videos
where ChannelID=@Cn1ID
) and LikeDislike='DisLike'
Group By ChannelID
)End)
Return @Net
END
```

```
select accounts.UserID, Name=Accounts.Name_First+' '+ Accounts.Name_Last,
AverageViews=(case when Sum(dbo.AvgViewsCn12(ChannelID))=0 then 0 Else
Sum(dbo.AvgViewsCn12(ChannelID))/count(Distinct ChannelID) eND)
, NetInteractions=sum(case when dbo.Net2(channelID) is null then 0 Else
dbo.Net2(ChannelID) end)
From Channels join Accounts on Channels.UserID= Accounts.UserID
Group By accounts.UserID,Name_First,Name_Last
Having (Sum(dbo.AvgViewsCn12(ChannelID))>0)
Order By AverageViews desc
```

דו"ח שיכלול את הנתונים הבאים על משתמש: מספר צפיות ממוצע לסרטון,
"נטו פידבק" (כמות הלייקים על הסרטונים שלו בניכוי ה-Dislikes).

נרצה לראות את המשתמשים הפופולריים ביותר (אלו שיש להם הכי הרבה צפיות לסרטון) ולראות האם הצופים אוהבים את הסרטונים או לא.

ההיגיון הוא שסרטון יכול לקבל הרבה צפיות גם מסיבות חיוביות וגם משליליות, ולכן נרצה לראות לפי הפידבק שהמשתמשים נותנים, לאיזה קטגוריה משתייכים הסרטונים ולפי זה לראות האם לקדם את הסרטון למבקרים ומפרסמים באתר או לא.

המימוש באמצעות שאילתה העושה שימוש ב-2 פונקציות עזר, בשתיהן קינן ובאחת אף קינן כפול. כמו כן, יש פעולת חיסור בין 2 שאילתות פנימיות.

	UserID	Name	AverageViews	NetInteractions
1	ALtYi3d9yEtN	Cody Baskeyfied	20	3
2	3YXLWohstYM	Inna Ashe	12.6666666666667	-3
3	imi2w5w5	Dene Bill	9	-4
4	AGv3gS	Neron McEneny	6.5	0
5	JDyfbnL92	Madonna Keary	5.625	6
6	EeNpnPG0	Joline Coie	5	0
7	FUHSeuMn	Bartlet Hassett	4.5	-5
8	mbDMfa	Dunc Graben	4.33333333333333	5
9	o6GyUw	Ferdie Aven	3.5	1
10	i15IDLgtsC	Derwin Pattle	3.41666666666667	-12
11	4JHUQI86pz	Vanna Castiglio	3	0
12	2IJSZxiMbk	Clementius Gaze	2.33333333333333	0
13	tuyl6gWwvquG	Aldric Maciejewski	1.44444444444444	-7
14	ruUDlmsIG	Elmer Eunson	0.25	0