



18 SETTEMBRE 2015

PROGETTO DI BASI DI DATI

ARGOMENTO: YOUTUBE

MARCO BERSELLI E SEBASTIANO ZAMBERLAN

CORSO: BASI DI DATI

Anno Accademico: 2014-2015

Analisi dei Requisiti

1. ABSTRACT

YouTube é una piattaforma web, fondata nel 2005 da Chad Hurley, Steve Chen e Jawed Karim che consente la condivisione e la visualizzazione in rete di video(video sharing).

Di proprietà di Google dall'ottobre 2006, Youtube attualmente ha più di 1 miliardo di utenti(persone iscritte al sito) che ogni giorno guardano centinaia di milioni di video e con 65.00 nuovi filmati aggiunti ogni 24 ore.Questo lo ha reso il terzo sito web più visitato al mondo dopo Google e Facebook.

Gli utenti , non solo ha a disposizione milioni di video, ma ha anche la possibilità di interagire con altri utenti, attraverso commenti e risposte ai video, rendendo il sito un "social network" .

I canali più popolari che caricano video possono attraverso una partnership (ovvero affiliarsi ad un network, cioè delle aziende che inseriscono pubblicità nei video in cambio di una percentuale sul totale guadagnato), per ottenere un compenso.

Non sono solo le persone "comuni" a creare canali su YouTube, ma anche aziende e persone famose che, caricando video di cui detengono il copyright , aumentano così la propria popolarità.

L'obbiettivo che vogliamo raggiungere riguarda di costruire un database, che riproduca nel modo più fedele possibile il database di YouTube.

Concentradoci particolarmente su come un utente attraverso il proprio canale/i gestisce la visione,

il caricamento di video e i feedback su quest'ultimi(commentati , mi piace ecc...) e la varie partnership dei canali.

2. DESCRIZIONE DEI REQUISITI

Si vuole realizzare una base di dati e la relativa applicazione web che contenga e gestisca le informazioni riguardanti il sito web di video sharing YouTube e in particolare si vogliono conoscere i dati relativi ai video inseriti, ai canali che li gestiscono e tutte le interazioni a livello social (Es: Commenti e risposte ai video).

Ciscun Utente è composto dalle seguenti informazioni:

- Una Email che identifica l'utente;
- Un Nome Utente, che indica come si chiama l'utente ;
- Una password, serve per accedere;
- Età, mi indica l'età dell'utente.

Un Utente può condividere un video su un social network. Il video può essere condiviso solo una volta per social.

Un Utente può possedere più canali.

Ciascun Canale è composto dalle seguenti informazioni:

- Codice, univoco che mi identifica il canale;
- Nome, che indica come si chiama il canale;
- Video Totali: il numero totale dei video inseriti nel canale;
- Iscritti: Il numero totale di canali iscritti nel canale;
- Strike: ovvero il quantitativo di ammonimenti ricevuti da youtube;
- Visualizzazioni totali: Le somme di tutte le visualizzazioni dei video.

Un Canale può avere più playlist.

Un canale può segnalare un video solo una volta

Se viene cancellato l'utente viene cancellato anche il canale

Quando viene creato il canale le visualizzazioni totali, gli iscritti, i video totali e gli strike sono a zero.

Dopo tre strike il canale viene cancellato e quindi di conseguenza tutti i dati ad esso associati.

Un canale può iscriversi ad un altro, quindi per vedere i video appena vengono pubblicati dal canale iscritto.

Un Canale ha la facoltà di scrivere un commento sotto un video e di rispondere ad un commento.

Un Canale ha la facoltà di vedere un video quante volte vuole e di votarlo con un Mi Piace se positivo o Non Mi Piace se negativo.

Un Canale può inserire un video.

Un Canale può creare una Playlist in cui vengono inseriti i video.

Dopo tre segnalazioni del video si riceve uno strike, dopo tre strike viene bloccato il Canale.

Ogni canale è associato ad una e-mail.

Dopo 10 visualizzazioni il canale effettua la partnership, quindi diventa un canale partner, ovvero può monetizzare i propri video quindi ricevere un ricavo economico.

Canale Partner è composto dalle seguenti informazioni:

- Un Network, un'azienda che inserisce la pubblicità nel video;
- Una Percentuale, ovvero il guadagno percentuale di ogni singola pubblicità vista;
- Guadagno Totale: Quanto un canale ha guadagnato dall'inizio della partnership.

Partnership può effettuare contratti e visualizzare i contratti passati.

Al inizio quando si diventa partner si ha come network Youtube e come percentuale al 0.6000.

Un Canale Partner può scegliere il tipo di pubblicità e il numero massimo di pubblicità inserite per video (MAX: 3. Scelta Nostra) ma non può inserire una pubblicità specifica.

Network è composto da:

- Nome, il nome del network, che lo identifica.

Può effettuare contratti con più canali partner:

Pubblicità è composta dalle seguenti informazioni:

- Codice, che identifica la pubblicità, ed è chiave.
- Nome, il nome della pubblicità;
- Tipo, dove è inserita nella pagina web.

Pubblicità viene inserita in video, in un certo momento durante la riproduzione (tipo pubblicità overlay, video ignorabili, video non ignorabili).

Tipo di Pubblicità è composto dalle seguenti informazioni:

- Formato Annuncio, che identifica Tipo di Pubblicità, e mi indica che tipo di annuncio;
- Descrizione, mi descrive dove si posiziona e com'è la pubblicità;
- Durata, mi indica quanto dura la pubblicità inserita.

I video ignorabili durano 5 secondi;

I video non ignorabili durano 30 secondi.

Playlist è composta dalle seguenti informazioni:

- Titolo, che insieme al codice del canale identifica la playlist;
- Numero Video, ovvero il quantitativo di video inseriti nella playlist;
- Privata, ovvero se la playlist è privata oppure pubblica che possono vedere tutti.

Ogni Playlist ha dei video che sono inseriti in essa;

Ogni playlist è associata ad un canale

Se viene cancellato il canale allora viene cancellata anche la playlist;

Se viene cancellato il video viene rimosso dalla playlist.

Quando la playlist viene creata il numero di video inseriti è zero.

Ogni canale ha una playlist che si chiama 'Mi Piace' e non può essere cancellata

Commento è composto da queste informazioni:

- Data, ovvero, la data e l'ora in cui è stato scritto;
- Scritto, ovvero, la parte commentata del commento.

Ogni Commento è associato ad un solo video ed può avere più risposte.

Commento è identificato da Data, Url del video e codice del canale che lo ha scritto.

Se viene cancellato il video o il canale che ha creato il commento, viene cancellato.

Risposta è composto da queste informazioni:

- Data ovvero, la data e l'ora in cui è stato scritto;
- Scritto, ovvero, la parte commentata della risposta.

Risposta è associata ad un solo Commento.

Risposta è identificato dalla data della risposta, dal canale che ha risposto e dal commento, cioè dalla data del commento, dal canale che ha scritto il commento e dal video in cui è inserito.

Se viene cancellato il video o il canale che ha scritto la risposta o il commento, quindi il video o il canale che lo ha scritto viene cancellata anche la risposta.

Ciascun video è composto dalle seguenti informazioni:

- URL, ovvero un codice identificativo del video;
- Un Titolo, che mi indica il nome del video;
- Durata: che mi indica in ore, minuti e secondi quanto dura il video;
- DataCaricamento: mi indica il giorno, mese e anno in cui è stato caricato;
- Visualizzazioni: Ovvero il quantitativo di volte che il video è stato visto;
- Mi Piace: indica il numero di gradimento del video;
- Non Mi Piace: indica il numero di non gradimento del video;
- Segnalazioni: ovvero il quantitativo di volte che gli altri canali ritengono inappropriato il video;
- Limite Età: indica l'età minima per vedere il video;
- Nota: indica una nota in descrizione.

Ogni video è associato ad un solo canale.

Al inizio quando viene creato il Video, le visualizzazioni, i Mi piace, i Non mi piace e le segnalazioni sono a zero.

Se l'utente ha età minore rispetto al video allora non può vederlo

Il video possiede anche un vincolo che ,se viene cancellato il Canale viene cancellato anche il video.

Un video può essere segnalato da un Canale e dopo tre segnalazioni viene bloccato.

Se il video riceve un 'Mi Piace', viene inserito nella playlist 'Mi Piace' del canale che ha votato

Il video può essere a pagamento quindi avere un prezzo, deciso dal proprietario del video, quindi un canale deve prima comprarlo se vuole vederlo.

Il video può essere una Live in streaming quindi le seguenti informazioni aggiuntive:

- Tempo di Inizio: ovvero l'ora,il minuto e il secondo in cui parte;
- Tempo di Fine: ovvero l'ora,il minuto e il secondo in cui termina.

Quando la live termina diventa un video normale.

Può avere diverse pubblicità inserite.

Ogni Video possiede uno o più Tipi e una o più qualità.

Tipo è composto da:

- Nome, ovvero indica il genere del video, che lo identifica.

Qualità è composta da:

- Codice, ovvero le varie risoluzioni e i quantitativi di frame al secondo(opzionale) possibili, che identifica la classe.

La qualità che ha ogni video è Auto e 144p

3. ANALISI DEI REQUISITI

Utente:

- E-Mail
- Nome
- Password
- Età

Canale:

- Codice
- Nome
- Video Totali
- Iscritti
- Strike
- Visualizzazioni

Un canale è associato ad un solo utente

Dopo tre strike il canale viene cancellato.

Video Totali, Iscritti, Strike e visualizzazioni a 0 se canale appena creato.

Canale Partner:

- Guadagno Totale
- Network
- Percentuale

Network uguale a Youtube e percentuale uguale a 0.6000 se un canale è appena diventato partner.

Generalizzazione di tipo parziale di Canale.

Se cancellato il canale viene cancellata il Canale Partner

Pubblicità:

- Codice
- Nome
- Tipo

Tipo di Pubblicità:

- Titolo;
- Numero Video;
- Privata.

Playlist:

- Titolo
- Numero Video
- Privata

Playlist ha Numero video uguale a zero appena creata.

Una Playlist è associata ad un canale.

Se cancellato il canale viene cancellata la playlist.

Se la playlist è pubblica allora gli altri canali possono vederla.

Network:

- Nome

Video:

- URL
- Titolo
- Durata
- Data di Caricamento
- Visualizzazioni
- Mi Piace
- Non Mi Piace
- Segnalazione
- Limite Età
- Nota

Ogni video è associato ad un canale

Se viene cancellato il canale allora si cancella il video

A Pagamento:

- Prezzo

Il Canale deve prima comprare il video per vederlo.

Generalizzazione di tipo parziale di video.

Dal Vivo:

- Tempo di Inizio
- Tempo di fine

Generalizzazione di tipo parziale di video.

Commento:

- Data
- Scritto

Ad ogni Commento ed associato un video e un canale.

Se Cancellato il Canale che lo ha scritto o il video a cui è associato viene cancellato

Risposta:

- Data
- Scritto

Ad ogni risposta è associato un canale un video e un commento

Se Cancellato il Canale che lo ha scritta o il video a cui è associata o il commento a cui è associata viene cancellata

Tipo

- Nome

Qualità:

- Codice

PROGETTAZIONE

1. PROGETTAZIONE CONCETTUALE

ENTITA' E ATTRIBUTI:

UTENTE:

- E-mail: varchar(50)
- Nome: varchar(100)
- Password: varchar(30)
- Età: decimal(3,0)

CANALE:

- Codice: varchar(24)
- Nome: varchar(100)
- Video totali: int(11)
- Iscritti: int(11)
- Strike: int(11)
- Visualizzazioni Totali: int(11)

CANALE PARTNER:

- Network: varchar(50)
- Percentuale: decimal(5,4)

Generalizzazione di tipo parziale di Canale.

PUBBLICITA':

- Codice: varchar(3)
- Nome: varchar(50)
- Tipo: varchar(30)

TIPO DI PUBBLICITA':

- Formato Annuncio :varchar(50);
- Descrizione: text
- Durata: time

PLAYLIST:

- Titolo: varchar(100)
- Numero Video: int(11)
- Privata: tinyint(1)

NETWORK:

- Nome: varchar(50)

VIDEO:

- URL: varchar(11)
- Titolo: varchar(100)
- Durata: time
- Data di Caricamento: date
- Visualizzazioni: int(11)
- Mi Piace: int(11)
- Non Mi Piace: int(11)
- Segnalazion: int(11)
- Limite Età: decimal(3,0)
- Nota: text

VIDEO A PAGAMENTO:

- Prezzo: decimal(5,2)

Generalizzazione parziale di video.

DAL VIVO:

- Tempo di Inizio: time
- Tempo di fine: time

Generalizzazione parziale di video.

COMMENTO:

- Data: timestamp
- Scritto: text

RISPOSTA:

- Data: timestamp
- Scritto: text

TIPO:

- Nome: varchar(22)

QUALITA':

- Codice: varchar(10)

RELAZIONI:

UTENTE-CANALE: GESTIONE

- Ogni Utente Gestisce uno o più canali.
- Un canale appartiene ad un solo utente.

UTENTE-VIDEO: CONDIVISO

- Un Utente può condividere il video su un social network.
- Il video può essere condiviso solo una volta per social network.

CANALE-CANALE: ISCRITTO

- Canale si iscrive ad un altro canale.

CANALE-COMMENTO: SCRITTO

- Canale scrive un commento.
- Un commento può essere scritto da un solo canale.

CANALE-RISPOSTA: SCRITTO

- Canale scrive una risposta.
- Una risposta può essere scritta da un solo canale

CANALE-PLAYLIST: CONTENUTO

- Solo un canale può creare una o più playlist.
- Una playlist appartiene ad un solo canale.

CANALE-VIDEO: CONTENUTO

- Un video può essere inserito in più di una playlist.
- Una playlist può avere più video.

CANALE-VIDEO: VISTO

- Canale può vedere quante volte il video.
- Il video può essere visto da più canale.

CANALE-VIDEO: VOTI

- Un canale può votare un video con un 'Mi Piace' o 'Non mi Piace'.
- Un video può ricevere più di un 'Mi Piace' o 'Non mi Piace'.

COMMENTO-VIDEO: COMMENTATO

- Un Commento può essere inserito in un solo video.
- Un video può avere più commenti.

COMMENTO-RISPOSTA: RISPOSTA

- Un Commento può avere più risposte
- Una risposta è inserita in un solo commento

PARTNER-NETWORK: CONTRATTO

- Partner ha un solo network
- Un Network ha più Partner

PARTNER-NETWORK: CONTRATTO PASSATO

- Un Partner può avere più Network passati
- Un Network può avere più Partner passati

PUBBLICITA'-VIDEO: INSERITA

- Una pubblicità può essere inserita in più video
- Video può avere più pubblicità

PLAYLIST-VIDEO: LISTA

- Una playlist può avere più video
- Un Video può essere inserito in più playlist

VIDEO-QUALITA': IMPOSTAZIONI

- Un video può avere più qualità
- Una qualità può avere più video

VIDEO-TIPO: GENERE

- Un Video può avere più di un tipo
- Un Tipo può essere associato ad più di un video

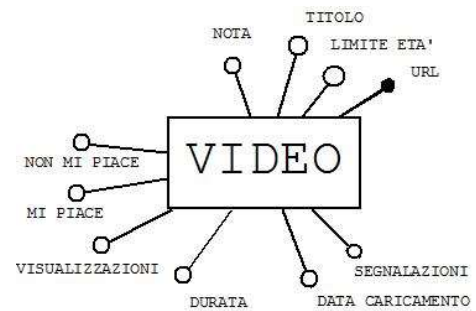
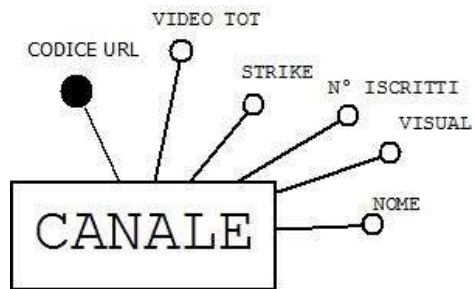
VIDEO-CANALE: SEGNALATO

- Un Video può essere segnalato da più canali
- Un canale può segnalare più video

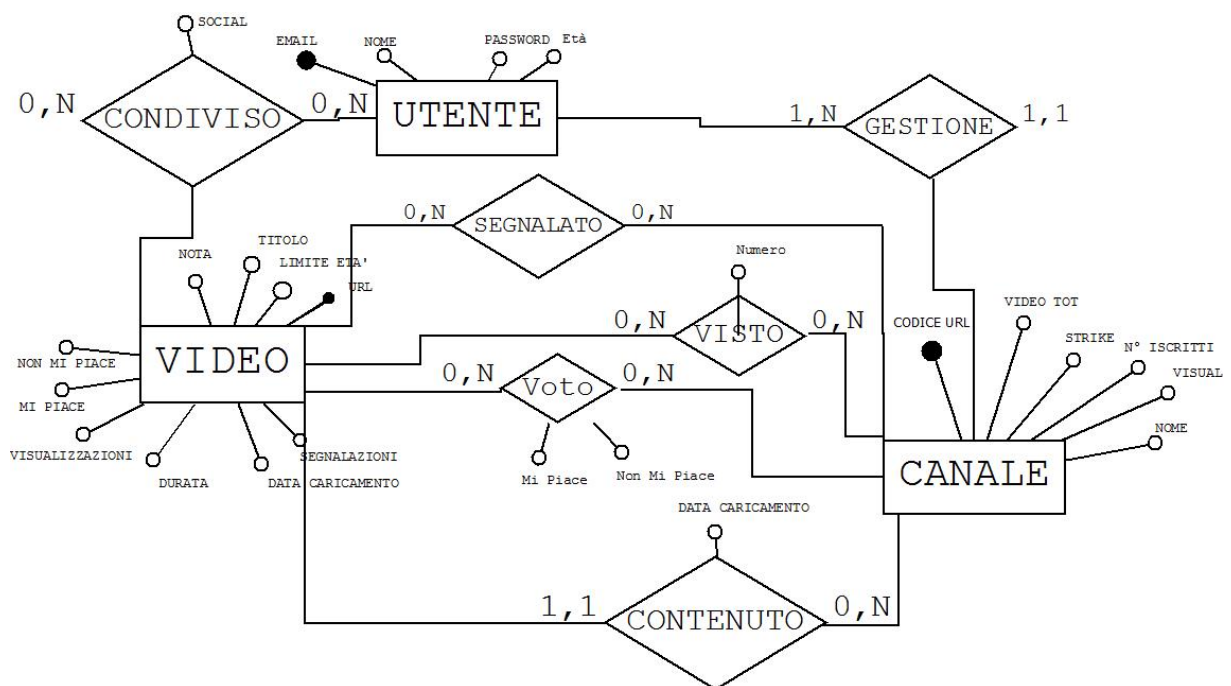
Per lo schema concettuale siamo partiti da un metodo BOTTOM-UP.

Siamo partiti dalle Entità e dalle Relazioni e abbiamo cercato di associarle tra loro.

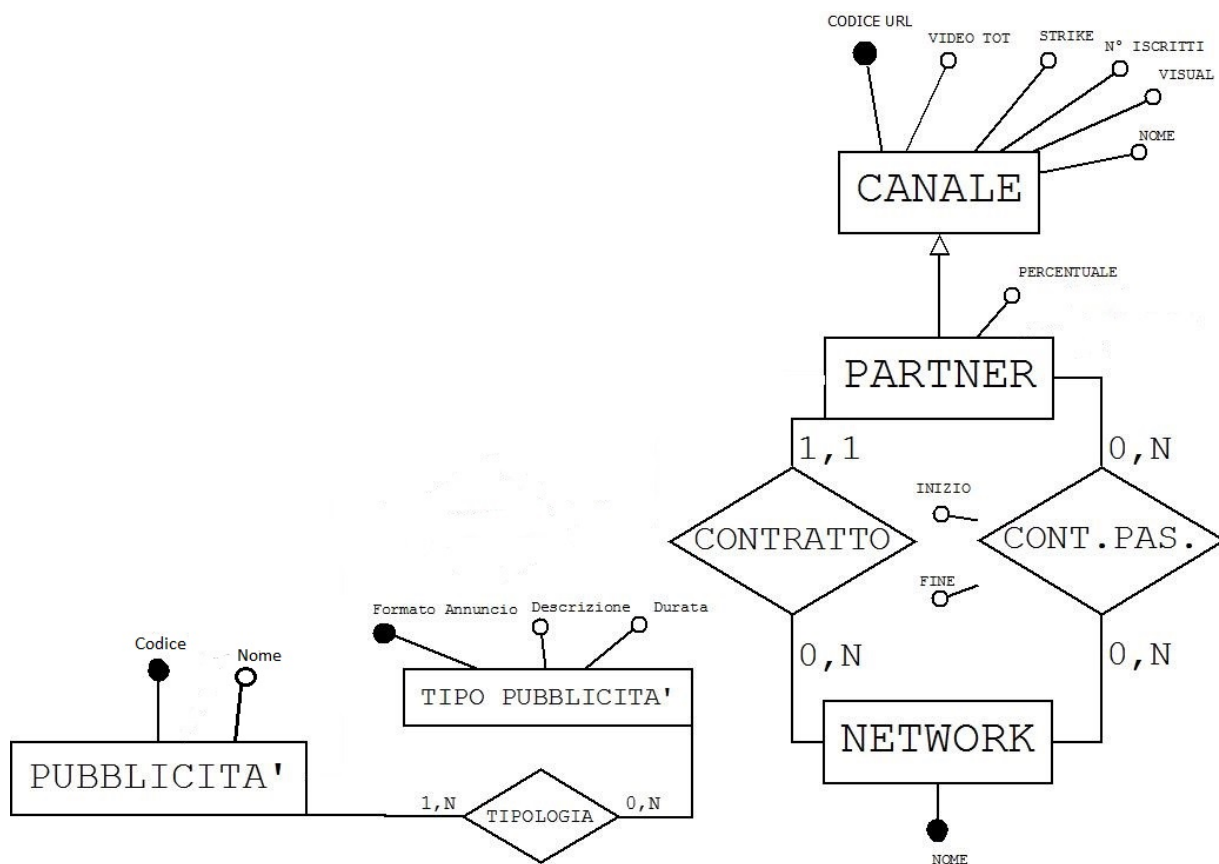
Innanzitutto siamo partiti da Canale e Video e li abbiamo associati tra loro con le relazioni.



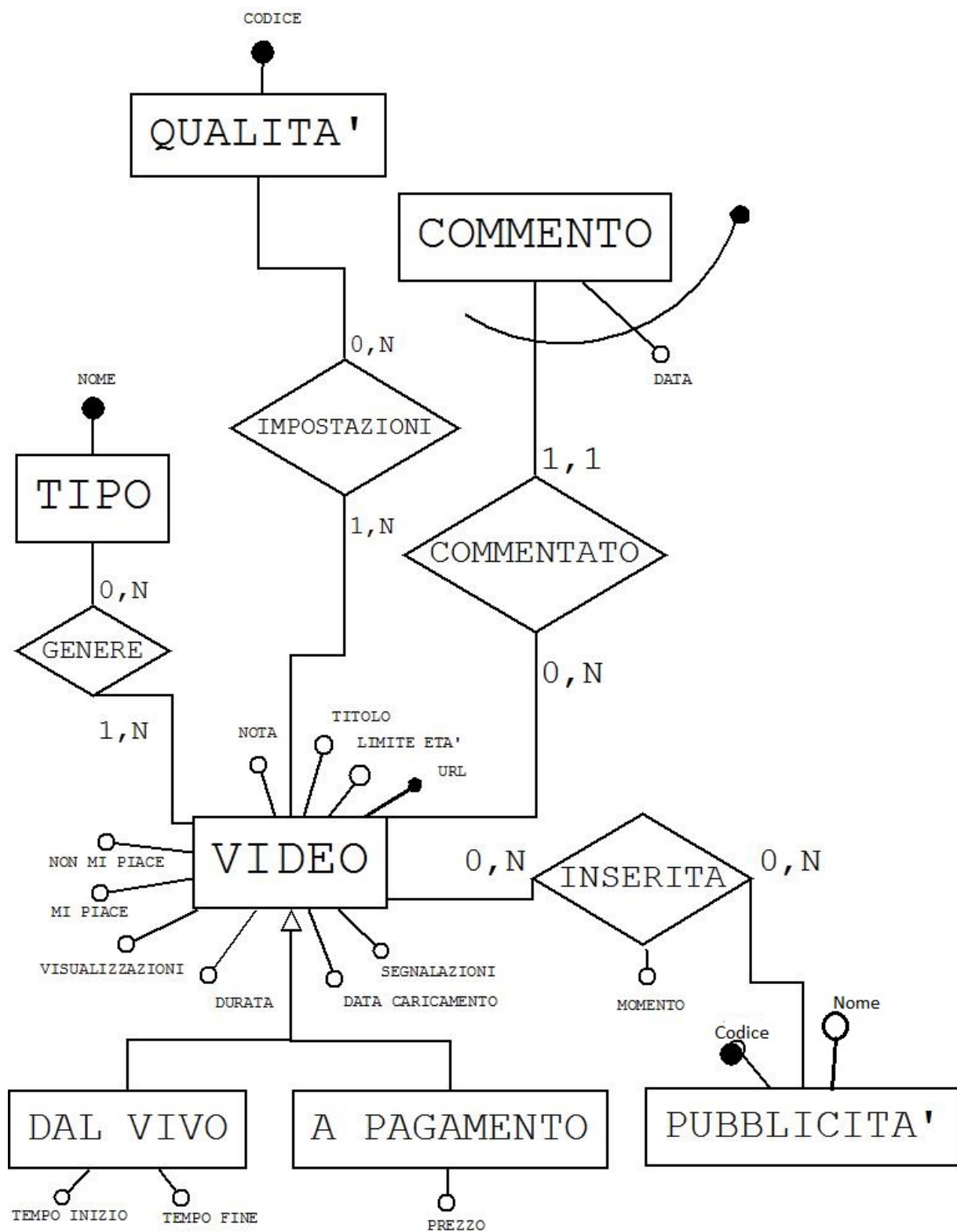
Abbiamo aggiunto le relazioni Contenuto, Voto, Visto e Segnalato, Gestione e Condiviso e abbiamo aggiunto la Entità Utente.



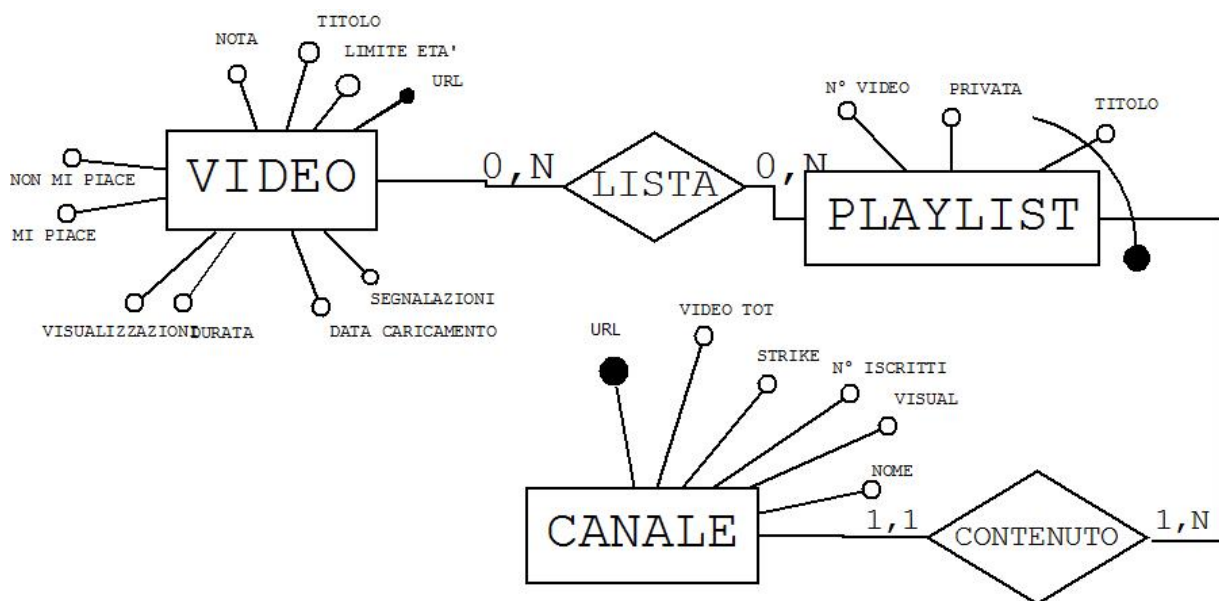
Successivamente abbiamo creato una componente con le entità: Canale, Partner, Network, pubblicità e Tipo di pubblicità e abbiamo aggiunto le relazioni: contratto, contratto passato e tipologia.



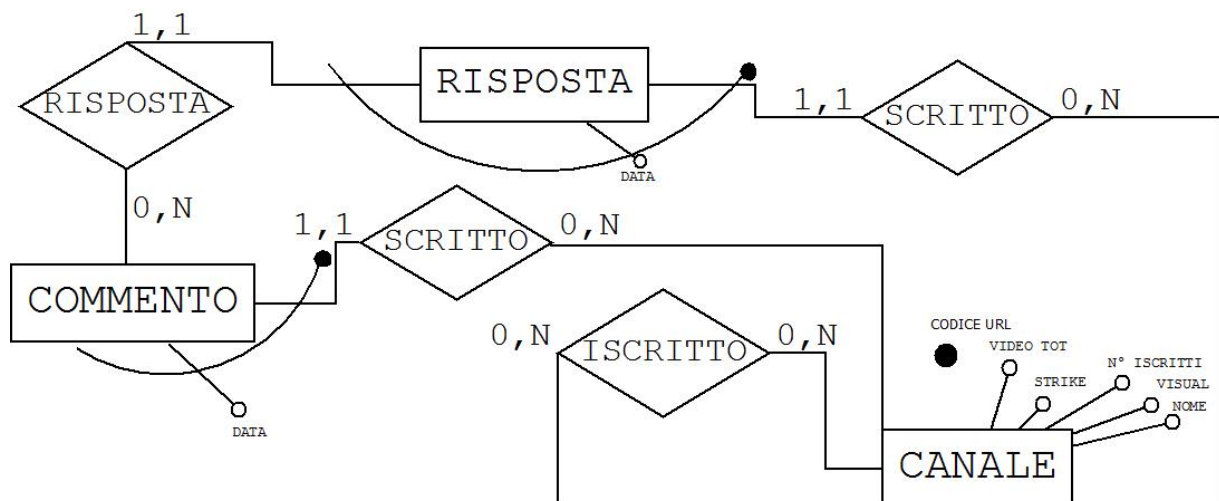
Dopo creiamo un componente che utilizza le Entità: Video, Dal Vivo, A Pagamento, Pubblicità, Commento Qualità e Tipo e le Relazioni sono: impostazioni, commentato, genere e inserita.



Successivamente abbiamo creato una componente utilizzando le Entità: Video, Playlist e Canale e abbiamo usato le relazioni: Lista e Contenuto.



Abbiamo creato un componente con le entità Commento Risposta e Canale e le Relazioni Risposta, scritto, scritto e iscritto.



Il diagramma ER illustra la struttura di un sistema di video sharing. Le entità e le loro attributi sono:

- UTENTE**: EMAIL, NOME, PASSWORD, ETÀ
- QUALITA'**: CODICE
- TIPO**: NOME
- GENERE**: (relazione con TIPO)
- IMPOSTAZIONI**: (relazione con QUALITA')
- COMMENTO**: DATA
- RISPOSTA**: DATA
- SCRITTO**: (relazione con COMMENTO e RISPOSTA)
- SEGNALATO**: (relazione con COMMENTO e VIDEO)
- LISTA**: (relazione con COMMENTO e VISTO)
- VOTO**: MI PIACE, NON MI PIACE
- VISTO**: NUMERO
- CANALE**: CODICE URL, VIDEO TOT, STRIKE, N° ISCRITTI VISUAL, NOME
- VIDEO**: NOTA, TITOLO, LIMITE ETÀ, URL, DURATA, SEGNALAZIONI, DATA CARICAMENTO, DAL VIVO, A PAGAMENTO, PREZZO
- PARTNER**: PERCENTUALE
- CONTRATTO**: INIZIO, FINE
- CONT.PAS.**: (relazione con PARTNER e NETWORK)
- PUBBLICITA'**: CODICE, NOME
- TIPO PUBBLICITA'**: FORMATO ANNUNCIO, DESCRIZIONE, DURATA
- TIPOLOGIA**: (relazione con TIPO PUBBLICITA')
- NETWORK**: NOME

Le relazioni e i vincoli cardinali sono:

- CONDIVISO** (UTENTE - QUALITA'): 0,N a 0,N
- GESTIONE** (UTENTE - CANALE): 1,N a 1,1
- RISPOSTA** (UTENTE - COMMENTO): 1,1 a 0,N
- SCRITTO** (UTENTE - RISPOSTA): 1,1 a 0,N
- SCRITTO** (COMMENTO - RISPOSTA): 1,1 a 0,N
- SEGNALATO** (COMMENTO - VIDEO): 0,N a 0,N
- LISTA** (COMMENTO - VISTO): 0,N a 0,N
- VOTO** (COMMENTO - VISTO): 0,N a 0,N
- VISTO** (COMMENTO - CANALE): 0,N a 0,N
- CANALE** (VIDEO): 1,N a 0,N
- PARTNER** (CANALE): 1,1 a 0,N
- CONTRATTO** (PARTNER - NETWORK): 1,1 a 0,N
- CONT.PAS.** (PARTNER - NETWORK): 0,N a 0,N
- PUBBLICITA'** (TIPO PUBBLICITA'): 1,1 a 0,N
- TIPOLOGIA** (TIPO PUBBLICITA'): 1,1 a 0,N

Dopo aver compiuto queste operazioni possiamo ottenere lo schema relazionale, attraverso l'accorpamento delle associazioni.

SCHEMA RELAZIONALE:

UTENTE (Email, Nome, Password, Età);

VIDEO (URL, Titolo, Canale, Durata, Data Caricamento, Visualizzazioni, Mi Piace, Non Mi Piace, Segnalazioni, Limite di Età, Prezzo, Tempo di Inizio, Tempo di Fine, Nota);

CONDIVISO (Email, Video, Social);

CANALE (Canale, Email, Nome, Video Totali, Iscritti, Strike, Visualizzazioni Totali);

ISCRITTO (Canale, Iscritto);

COMMENTO (Canale, Video, Data);

RISPOSTA (Canale, Video, Data, Canale Commento, Data Commento);

SEGNALATO (Canale, Video);

PLAYLIST (Canale, Titolo, Privata, Numero Video);

LISTA (Canale, Video, Playlist);

QUALITA' (Codice);

IMPOSTAZIONI (Video, Qualità);

TIPO (Nome);

GENERE (Video, Tipo);

VISTO (Canale, Video, Numero);

VOTO (Video, Canale, Mi Piace, Non Mi Piace);

NETWORK (Nome);

PARTNER (Canale, Network, Percentuale);

CONTRATTO PASSATO (Canale, Network, Inizio, Fine);

PUBBLICITA' (Codice, Nome, Tipo);

INSERITA (Video, Pubblicità, Momento);

TIPO PUBBLICITA' (Formato Annuncio, Descrizione, Durata);

QUERY SIGNIFICATIVE

QUERY 1

Seleziona i video più popolari di ogni canale e li mette in ordine per visualizzazioni.

```
SELECT Titolo, Canale, URL, Visualizzazioni
FROM video V
WHERE Visualizzazioni >= ALL
(SELECT Visualizzazioni
FROM video V1
WHERE V1.Canale=v.Canale)
AND
V.URL <> ALL
(SELECT V1.URL
FROM video V1, video V2
WHERE V1.URL<>V2.URL AND V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND
V1.MiPiacce <= V2.MiPiacce)
ORDER by Visualizzazioni DESC
```

Document Manager Query 1:

Descrizione:

Questa interrogazione parte in automatico quando si entra in home (ovvero la pagina iniziale) quando non sei loggato.

La richiesta al database è quella di restituire il Titolo del video, il Canale del video , l' URL del video e le visualizzazioni.

Questo tipo di interrogazione viene scelta in base a due condizioni:

1. Le Visualizzazioni del video maggiori o uguali rispetto a tutte le visualizzazioni dello stesso canale.

Visualizzazioni >= ALL(SELECT Visualizzazioni FROM video V1 WHERE V1.Canale=v.Canale)

2. L' URL del Video diverso da tutti gli altri URL che vengo scelti in base ai Mi Piacce maggiori o uguali agli altri con visualizzazioni uguali e canale uguale.

```
V.URL <> ALL(SELECT V1.URL FROM video V1, video V2 WHERE V1.URL<>V2.URL AND
V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND V1.MiPiacce <= V2.MiPiacce)
```

RISULTATI QUERY 1: Pagina Successiva

Titolo	Canale	URL	Visualizzazioni
Hitler vs Vader 3	erb [->]	BpqNvskS_kM	144
THE WORST GAME EVER CREATED.	PewDiePie [->]	zdwY0J3coG0	107
Spyro - Did You Know Gaming? Feat. Caddicarus	DYKGaming [->]	KCJJHwmB9G4	104
Lady Gaga - Bad Romance	LadyGagaVEVO [->]	qrO4YZeyl0l	71
Eminem - Love The Way You Lie ft. Rihanna	EminemVEVO [->]	uelHwf8o7_U	49
Il Gigante di Ferro: il film che TUTTI dovrebbero vedere - Nerd Cultura	DarioMocciaChannel [->]	EMurDPVB4zQ	20
MOSTARDA - Fuga Di Cervelli	yotobi [->]	JaFewo9ml0s	17
Facebook sapra riconoscere quando SEI UBRIACO!	breakingitaly [->]	ltVR9laoxso	16
Iron Maiden -The Number Of The Beast / Remember Tomorrow	ironmaiden [->]	WD4PdLEldLg	15
Buying Your First Telescope	tdarnell [->]	gzMsjGulmUE	13
UN TUFFO ALLA LAGUNA BLU VLOG ISLANDA	ClioMakeUp [->]	e8LluGQVIKQ	13
TraileRant #2: Dragonball Evolution	croix89 [->]	fSCCP1p20R8	11
ONE PIECE: Usop, coraggioso guerriero del mare	OnePieceReport [->]	0CBKQTpYJEo	10
ITALIA URUGUAY - IL COMMENTO Parodia	Giumenta87 [->]	i0cnGX-sSf8	10
Gioventu Giocata #3 - AGE OF EMPIRES	parliamodivg [->]	bYpITTKZR5Q	9
Lawless	ItaKochMediaFilm [->]	CDBOd-lujl4	9
Mazinger Edition Z - Ep. 1 ITA - Fino alla fine	YamatoEntertainment [->]	kxER5YekkZg	9
The Hunger Games: Mockingjay - Part 2 Official Teaser Trailer #1 (2015)	movieclipsTRAILERS [->]	KmYNkasYthg	9
Cristiano Ronaldo wins FIFA Ballon d'Or 2014	FIFATV [->]	3kJwVazTitg	8
Ingegneria chimica e dei materiali - Università di Padova	UniPadova [->]	9lQ6dBksmA4	7
Chi vincer� il prossimo campionato ?	UCctiECw-dnJpSW8E8PX9oEQ [->]	fusaigbigas	5

QUERY 2

Seleziona i video di una tipologia più popolari di ogni canale.

In questo esempio abbiamo scelto tre categorie: "Film e Intrattenimento", "Animazione" e "Umorismo"

```
SELECT Titolo,Canale,URL,Visualizzazioni
FROM video as V join genere as G on (V.URL=G.Video)
WHERE G.Tipo='Animazione' AND
V.Visualizzazioni >= ALL(SELECT V1.Visualizzazioni FROM video V1 WHERE V1.Canale=v.Canale)
AND
V.URL <>ALL(SELECT V1.URL FROM video V1, video V2 WHERE V1.URL<>V2.URL AND
V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND V1.MiPiace <= V2.MiPiace)
UNION
SELECT Titolo,Canale,URL,Visualizzazioni
FROM video as V join genere as G on (V.URL=G.Video)
WHERE G.Tipo='Film e Intrattenimento'AND
V.Visualizzazioni >= ALL
(SELECT V1.Visualizzazioni
FROM video V1
WHERE V1.Canale=v.Canale)
AND
V.URL <>ALL
(SELECT V1.URL
FROM video V1, video V2
WHERE V1.URL<>V2.URL AND V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND
V1.MiPiace <= V2.MiPiace)
UNION
SELECT Titolo,Canale,URL,Visualizzazioni
FROM video as V join genere as G on (V.URL=G.Video)
WHERE G.Tipo='Umorismo' AND
V.Visualizzazioni >= ALL
(SELECT V1.Visualizzazioni
FROM video V1
WHERE V1.Canale=v.Canale)
AND
V.URL <>ALL
(SELECT V1.URL
FROM video V1, video V2
WHERE V1.URL<>V2.URL AND V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND
V1.MiPiace <= V2.MiPiace)
ORDER by Visualizzazioni DESC
```

Document Manager Query 2:

Descrizione:

Questa interrogazione parte in automatico in Home quando no sei loggato .
Essa restituisce i Titolo, Canale, URL e Visualizzazioni dei video per ogni categoria scelta e successivamente unite con la funzione Union. Prendo in considerazione due 2 condizioni perché ci sono delle query simili che si unisco con union in cui l'unica differenza è il passaggio del parametro categoria.

Questo tipo di interrogazione viene scelta in base a due condizioni:

1. Le Visualizzazioni del video maggiori o uguali rispetto a tutte le visualizzazioni dello stesso canale.

Visualizzazioni >= ALL(SELECT Visualizzazioni FROM video V1 WHERE V1.Canale=v.Canale)

2. L' URL del Video diverso da tutti gli altri URL che vengo scelti in base ai Mi Piace maggiori o uguali agli altri con visualizzazioni uguali e canale uguale.

V.URL <>ALL(SELECT V1.URL FROM video V1, video V2 WHERE V1.URL<>V2.URL AND V1.Canale=V2.Canale AND V1.Visualizzazioni=V2.Visualizzazioni AND V1.MiPiacce <= V2.MiPiacce)

RISULTATI QUERY 2:

Titolo	Canale	URL	Visualizzazioni
Lady Gaga - Bad Romance	LadyGagaVEVO [->]	qrO4YZeyl0I	71
Il Gigante di Ferro: il film che TUTTI dovrebbero vedere - Nerd Cultura	DarioMocciaChannel [->]	EMurDPVB4zQ	20
MOSTARDA - Fuga Di Cervelli	yotobi [->]	JaFewo9ml0s	17
TraileRant #2: Dragonball Evolution	croix89 [->]	fSCCP1p20R8	11
ONE PIECE: Usop, coraggioso guerriero del mare	OnePieceReport [->]	0CBKQTpYJEO	10
Mazinger Edition Z - Ep. 1 ITA - Fino alla fine	YamatoEntertainment [->]	kxER5YekkZg	9
The Hunger Games: Mockingjay - Part 2 Official Teaser Trailer #1 (2015)	movieclipsTRAILERS [->]	KmYNkasYthg	9
Lawless	ItaKochMediaFilm [->]	CDBOd-lujl4	9
Chi vincerÃ il prossimo campionato ?	UCctiECw-dnJpSW8E8PX9oEQ [->]	fusaigbigas	5

QUERY 3

Seleziona la tipologia dei video dei tuoi iscritti e trova i video di Canali a cui non sei iscritto della stessa tipologia e li mettere in ordine in base alle visualizzazioni.

Abbiamo scelto per la variabile \$Codice = 'DYKGaming'

```
SELECT DISTINCT v2.URL,v2.Titolo,v2.Canale,v2.Visualizzazioni
FROM iscritto as i2 JOIN video as v2 ON (i2.Canale=v2.Canale) JOIN genere as g2 ON (v2.URL=g2.Video)
WHERE i2.Canale NOT IN(SELECT Canale FROM iscritto WHERE Iscritto=\"$Codice\" )
AND g2.Tipo IN(SELECT g1.Tipo
FROM iscritto as i1 JOIN video as v1 ON (i1.Canale=v1.Canale) JOIN genere as g1 ON (v1.URL=g1.Video)
WHERE i1.Iscritto=\"$Codice\"
GROUP BY g1.Tipo
ORDER BY COUNT(v1.URL) DESC)
AND v2.Visualizzazioni>=ALL(SELECT v3.Visualizzazioni
FROM video as v3
WHERE v3.Canale=v2.Canale)

ORDER BY v2.Visualizzazioni DESC
```

Document Manager Query 3:

Questa interrogazione ha una variabile dipendente ovvero \$Codice che mi indica il codice del canale che ha fatto l'accesso al sito.

Questa interrogazione parte in home quando sei loggato al sito.

Essa restituisce l' URL, Titolo, Canale, e Visualizzazioni di un video di un canale in cui non sei iscritto che ha come tipologia uguale a quella dei video dei tuoi iscritti.

Utilizziamo tre condizioni:

1. Seleziona i Canali diversi dai tuoi iscritti

```
i2.Canale NOT IN(SELECT Canale FROM iscritto WHERE Iscritto=\"$Codice\" )
```

2. Il Genere di video è uguale al tipo di genere degli iscritti.

```
g2.Tipo IN(SELECT g1.Tipo FROM iscritto as i1 JOIN video as v1 ON (i1.Canale=v1.Canale) JOIN genere as g1 ON (v1.URL=g1.Video) WHERE i1.Iscritto=\"$Codice\";
```

3. Seleziona i video con le visualizzazioni maggiori

```
v2.Visualizzazioni>=ALL(SELECT v3.Visualizzazioni FROM video as v3 WHERE v3.Canale=v2.Canale)
```

RISULTATI QUERY 3:pagina successiva

URL	Titolo	Canale	Visualizzazioni
KCJJHwmB9G4	Spyro - Did You Know Gaming? Feat. Caddicarus	DYKGaming [->]	104
qrO4YZeyl0I	Lady Gaga - Bad Romance	LadyGagaVEVO [->]	71
bYpITTKZR5Q	Gioventu Giocata #3 - AGE OF EMPIRES	parliamodivg [->]	9

QUERY 4

Seleziona in base alla tipologia dei tuoi video che i tuoi iscritti non hanno, e cerca i migliori video dei canali a cui non sei iscritto e li mostra.

Abbiamo scelto per la variabile \$Codice = 'DYKGaming'

```

SELECT DISTINCT v2.URL,v2.Titolo,v2.Visualizzazioni, v2.Canale
FROM iscritto as i2 JOIN video as v2 ON (i2.Canale=v2.Canale) JOIN genere as g2 ON (v2.URL=g2.Video)
WHERE i2.Canale NOT IN(SELECT Canale FROM iscritto WHERE Iscritto=\"$Codice\")
AND g2.Tipo NOT IN(SELECT g1.Tipo
FROM iscritto as i1 JOIN video as v1 ON (i1.Canale=v1.Canale) JOIN genere as g1 ON (v1.URL=g1.Video)
WHERE i1.Iscritto=\"$Codice\")
GROUP BY g1.Tipo
ORDER BY COUNT(v1.URL) DESC)
AND v2.Visualizzazioni>=ALL(SELECT v3.Visualizzazioni
FROM video as v3
WHERE v3.Canale=v2.Canale)

ORDER BY v2.Visualizzazioni DESC

```

Document Manager Query 4:

Questa interrogazione ha una variabile dipendente ovvero \$Codice che mi indica il codice del canale che ha fatto l'accesso al sito.

Questa interrogazione parte in home quando sei loggato al sito.

Essa restituisce l' URL, Titolo, Canale, e Visualizzazioni di un video di un canale in cui non sei iscritto che ha come tipologia diversa rispetto a quella dei video dei tuoi iscritti.

Utilizziamo due condizioni:

1. Seleziona i Canali diversi dai tuoi iscritti

i2.Canale NOT IN(SELECT Canale FROM iscritto WHERE Iscritto=\"\$Codice\")

2. Il Genere di video è diverso al tipo di genere degli iscritti.

g2.Tipo NOT IN(SELECT g1.TipoFROM iscritto as i1 JOIN video as v1 ON (i1.Canale=v1.Canale) JOIN genere as g1 ON (v1.URL=g1.Video) WHERE i1.Iscritto=\"\$Codice\" GROUP BY g1.TipoORDER BY COUNT(v1.URL) DESC)

3. Seleziona i video con le visualizzazioni maggiori

v2.Visualizzazioni>=ALL(SELECT v3.Visualizzazioni FROM video as v3 WHERE v3.Canale=v2.Canale)

RISULTATI QUERY4:Questa Pagina e successiva

URL	Titolo	Visualizzazioni	Canale
qrO4YZeyl0I	Lady Gaga - Bad Romance	71	LadyGagaVEVO [->]
EMurDPVB4zQ	Il Gigante di Ferro: il film che TUTTI dovrebbero ...	20	DarioMocciaChannel [->]
JaFewo9ml0s	MOSTARDA - Fuga Di Cervelli	17	yotobi [->]
ltVR9laoxso	Facebook sapra riconoscere quando SEI UBRIACO!	16	breakingitaly [->]
e8LluGQVIKQ	UN TUFFO ALLA LAGUNA BLU VLOG ISLANDA	13	ClioMakeUp [->]

URL	Titolo	Visualizzazioni	Canale
gzMsjGuImUE	Buying Your First Telescope	13	tdarnell [->]
fSCCP1p20R8	TraileRant #2: Dragonball Evolution	11	croix89 [->]
0CBKQTpYJEo	ONE PIECE: Usop, coraggioso guerriero del mare	10	OnePieceReport [->]
i0cnGX-sSf8	ITALIA URUGUAY - IL COMMENTO Parodia	10	Giumenta87 [->]
kxER5YekkZg	Mazinger Edition Z - Ep. 1 ITA - Fino alla fine	9	YamatoEntertainment [->]
KmYNkasYthg	The Hunger Games: Mockingjay - Part 2 Official Tea...	9	movieclipsTRAILERS [->]
CDBOd-IujI4	Lawless	9	ItaKochMediaFilm [->]
3kJwVazTitg	Cristiano Ronaldo wins FIFA Ballon d'Or 2014	8	FIFATV [->]
9lQ6dBksmA4	Ingegneria chimica e dei materiali - Universita di...	7	UniPadova [->]

QUERY 5

Selezione dei canali la cui età media degli iscritti è compresa fra +10 e -10 anni rispetto all'età dell'utente del canale

Abbiamo scelto per la variabile \$Codice = 'yotobi'

```
select iscritto.Canale, avg(utente.Eta)
from iscritto join canale on (iscritto.Iscritto=canale.Codice) join utente on
(canale.Email_Utente=utente.Email)
GROUP BY iscritto.Canale
HAVING ((avg(utente.Eta))>(SELECT utente.Eta -10
      FROM canale join utente ON(canale.Email_Utente=utente.Email)
      WHERE canale.Codice=\"$Codice\"
    )
  AND (avg(utente.Eta))<(SELECT utente.Eta +10
      FROM canale join utente ON(canale.Email_Utente=utente.Email)
      WHERE canale.Codice=\"$Codice\" ))
```

Document Manager Query 5:

Questa query dipende dalla variabile dipendente \$Codice che è il codice della tabella canale di chi ha fatto l'accesso al sito.

Questa interrogazione parte in home quando sei loggato al sito.

Essa restituisce il canale degli iscritti di \$Codice, e l'età media dell'utente compresa tra +10 e -10 anni rispetto all'età dell'utente del canale.

Utilizziamo due condizioni inserite nel having:

1. L'età media degli utenti deve essere maggiore dell'età dell'utente del canale preso in esame

```
(avg(utente.Eta))>(SELECT utente.Eta -10 FROM canale join utente N(canale.Email_Utente=utente.Email)
      WHERE canale.Codice=\"$Codice\")
```

2. L'età media degli utenti deve essere minore dell'età dell'utente del canale preso in esame

```
(avg(utente.Eta))<(SELECT utente.Eta +10 FROM canale join utente
ON(canale.Email_Utente=utente.Email) WHERE canale.Codice=\"$Codice\" )
```

RISULTATI QUERY 5:

Canale	avg(utente.Eta)
breakingitaly [->]	34.0000
croix89 [->]	24.2857
DarioMocciaChannel [->]	22.4444
DYKGaming [->]	29.0000
EminemVEVO [->]	31.8333
erb [->]	32.9259
ironmaiden [->]	25.0000
ItaKochMediaFilm [->]	27.5000
LadyGagaVEVO [->]	30.3636
movieclipsTRAILERS [->]	29.5000
OnePieceReport [->]	23.2000
PewDiePie [->]	30.0000
tdarnell [->]	24.5000
YamatoEntertainment [->]	34.0000
yotobi [->]	30.2222

QUERY 6

Seleziona il tipo di video più visualizzato da chi ha una età media inferiore ai 18 anni e da chi ha più di 18 anni.

```
SELECT genere.Tipo, COUNT(visto.Numero)
FROM visto join canale ON (visto.Canale=canale.Codice) join utente on
(canale.Email_Utente=utente.Email) join genere on (visto.Video=genere.Video)
GROUP by genere.Tipo
HAVING ((AVG(utente.Eta)>18) AND( COUNT(visto.Numero)>=ALL(SELECT COUNT(visto.Numero)
FROM visto join canale ON (visto.Canale=canale.Codice) join utente on
(canale.Email_Utente=utente.Email) join genere on (visto.Video=genere.Video)
GROUP by genere.Tipo
HAVING (AVG(utente.Eta)>18))))))
UNION
SELECT genere.Tipo, COUNT(visto.Numero)
FROM visto join canale ON (visto.Canale=canale.Codice) join utente on
(canale.Email_Utente=utente.Email) join genere on (visto.Video=genere.Video)
GROUP by genere.Tipo
HAVING ((AVG(utente.Eta)<18) AND( COUNT(visto.Numero)>=ALL(SELECT COUNT(visto.Numero)
FROM visto join canale ON (visto.Canale=canale.Codice) join utente on
(canale.Email_Utente=utente.Email) join genere on (visto.Video=genere.Video)
GROUP by genere.Tipo
HAVING (AVG(utente.Eta)<18))))))
```

Document Manager Query 6:

Questa interrogazione parte in home quando non sei loggato al sito.

Essa restituisce il tipo di video e quanto hanno visto di quel tipo di video quelli di età media maggiore di 18 e quelli di età media inferiore ai 18

Prendo in considerazione solo due condizioni perché c'è una union di due query simili tra loro il cui cambia solo una uguaglianza.

Considero le condizioni:

1. L'età media dell'utente è maggiore o minore di 18

Maggiore: (AVG(utente.Eta)>18), Minore: (AVG(utente.Eta)<18)

2. Il video ha le visualizzazioni maggiori rispetto a tutti gli altri video.

COUNT(visto.Numero)>=ALL(SELECT COUNT(visto.Numero) FROM visto join canale ON
(visto.Canale=canale.Codice) join utente on (canale.Email_Utente=utente.Email) join genere on
(visto.Video=genere.Video))

RISULTATI QUERY 6:

Tipo	COUNT(visto.Numero)
Musica [->]	239
Scienza e Istruzione [->]	14

Funzioni

Function:

1.

Questa funzione restituisce la media del Rapporto tra visualizzazioni e il numero di iscritti.

```
CREATE DEFINER=`root`@`localhost` FUNCTION `MedialiscrittiVisualTOT`(`canaleTest` VARCHAR(24))
RETURNS int(11)
    NO SQL
BEGIN
DECLARE MedialsVis integer;
DECLARE VideoTest integer;
SELECT canale.Iscritti INTO VideoTest
FROM canale
WHERE canale.Codice=canaleTest;
IF(VideoTest=0)
THEN
SET MedialsVis=0;
ELSE
SELECT canale.Visualizzazioni_TOT/canale.Iscritti INTO MedialsVis
FROM canale
WHERE canale.Codice=canaleTest;
END IF;
RETURN MedialsVis;
END
```

Descrizione: la funzione permette di calcolare il rapporti tra visualizzazioni e il numero di iscritti passando il codice del canale (Varchar(24)) come input. Se la media degli iscritti è zero metto il rapporto a zero.

2.

Questa funzione restituisce la media del rapporto tra visualizzazioni totali e video totali.

```
CREATE DEFINER='root'@'localhost' FUNCTION `MediaVisualTOTVideoTOT`(`canaleTest` VARCHAR(24))
RETURNS int(11)
    NO SQL
BEGIN
DECLARE MediaVisVid integer;
DECLARE VideoTest integer;
SELECT canale.Video_TOT INTO VideoTest
FROM canale
WHERE canale.Codice=canaleTest;
IF(VideoTest=0)
THEN
SET MediaVisVid=0;
ELSE
SELECT canale.Visualizzazioni_TOT/canale.Video_TOT INTO MediaVisVid
FROM canale
WHERE canale.Codice=canaleTest;
END IF;
RETURN MediaVisVid;
END
```

Descrizione: la funzione permette di calcolare il rapporto tra visualizzazioni totali e i video totali passando il codice del canale (Varchar(24)) come input. Se la media dei video è zero metto il rapporto a zero.

Procedure:

1.

Procedura che inserisce la pubblicità nella tabella inserita in base al tipo di pubblicità scelta.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `inserimentoPubblicita` (IN `VideoScelto` VARCHAR(24)  
CHARSET latin1, IN `TipoPubblicita1` VARCHAR(50) CHARSET latin1, IN `Orario` TIME, OUT `Testo` TEXT)  
    NO SQL  
BEGIN  
    DECLARE numPub int;  
    DECLARE codRandom varchar(3);  
    SELECT COUNT( DISTINCT inserita.Codice,inserita.Momento) INTO numPub  
    FROM inserita  
    WHERE inserita.Video=VideoScelto  
    GROUP BY inserita.Video;  
    IF(numPub<3)  
    THEN  
        INSERT INTO inserita  
        (SELECT pubblicita.Codice, VideoScelto AS Video, Orario AS Momento  
        FROM pubblicita  
        WHERE pubblicita.Tipo=TipoPubblicita1 AND  
        pubblicita.Codice<>ALL(SELECT inserita.Codice FROM inserita WHERE inserita.Video=VideoScelto)  
        ORDER BY RAND()  
        limit 1);  
  
    ELSE  
        SET Testo ='errore numero max pubblicita a video gia inserito';  
    END IF;  
END
```

Descrizione: Il Primo select restituisce il numero di pubblicità inserite nel video.

Se il numero di pubblicità è minore di 3 allora si inserisce con un' altra select il codice delle pubblicità e l'URL del video dove il tipo di pubblicità è uguale a quello selezionato e il codice della pubblicità è diverso il codice delle altre pubblicità inserite(3 Select).

2.

Restituisce il canale che ha visto di più i video del canale cercato.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `canalepiùvisual`(IN `CanaleRicerca` VARCHAR(24)
CHARSET latin1)
    NO SQL
BEGIN
SELECT c1.Codice,COUNT(visto.Numero)
FROM visto JOIN canale as c1 ON (visto.Canale=c1.Codice)JOIN video ON (video.URL=visto.Video)
WHERE video.Canale=CanaleRicerca
GROUP BY c1.Codice
HAVING COUNT(visto.Numero)>=ALL(SELECT COUNT(visto.Numero)
FROM visto JOIN canale as c1 ON (visto.Canale=c1.Codice)JOIN video ON (video.URL=visto.Video)
WHERE video.Canale=CanaleRicerca
GROUP BY c1.Codice);
END
```

Descrizione: Il primo select restituisce il codice del canale e il numero di visualizzazioni da parte dei canali, dove il canale del video è quello di ricerca. Poi sono le soluzioni sono raggruppate attraverso il codice del canale e ha il numero di visualizzazioni maggiore degli altri canali(2° select).

3.

Seleziono il canale con più Mi Piace.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `canalepiuMP`(IN `CanaleRicerca` VARCHAR(24)
CHARSET latin1)
    NO SQL
SELECT c1.Codice,COUNT(voti.MiPiace)
FROM voti JOIN canale as c1 ON (voti.Canale=c1.Codice)JOIN video ON (video.URL=voti.Video)
WHERE video.Canale=CanaleRicerca
GROUP BY c1.Codice
HAVING COUNT(voti.MiPiace)>=ALL(SELECT COUNT(voti.MiPiace)
FROM voti JOIN canale as c1 ON (voti.Canale=c1.Codice)JOIN video ON (video.URL=voti.Video)
WHERE video.Canale=CanaleRicerca
GROUP BY c1.Codice)
```

Descrizione: Ritorno il canale con più mi piace (1° select) dove il canale del video è uguale a quello di ricerca. Il canale ritornato è raggruppato in base al codice e in base ai Mi piace che devono essere maggiori di tutti gli altri(2° select).

Trigger

1. Aggiorna strike: inserisce gli strike al canale.

```
CREATE TRIGGER `aggiornaStrike_Visual` AFTER UPDATE ON `video`  
  FOR EACH ROW BEGIN  
    DECLARE Y int; DECLARE X int;  
  
    IF(NEW.Visualizzazioni>OLD.Visualizzazioni)  
    THEN  
      UPDATE canale SET canale.Visualizzazioni_TOT=canale.Visualizzazioni_TOT+1 WHERE  
      canale.Codice=NEW.Canale;  
    END IF;  
  
    SELECT canale.Strike INTO Y FROM canale WHERE canale.Codice=NEW.Canale;  
    IF(Y=0)  
    THEN  
      SELECT SUM(video.Segnalazioni) INTO X FROM video WHERE video.Canale=NEW.Canale;  
      IF(X=3)  
      THEN  
        UPDATE canale SET canale.Strike=canale.Strike+1 WHERE canale.Codice=NEW.Canale;  
      END IF;  
    ELSEIF(Y=1)  
    THEN  
      SELECT SUM(video.Segnalazioni) INTO X FROM video WHERE video.Canale=NEW.Canale;  
      IF (X=6)  
      THEN  
        UPDATE canale SET canale.Strike=canale.Strike+1 WHERE canale.Codice=NEW.Canale;  
      END IF;  
    ELSEIF(Y=2)  
    THEN  
      SELECT SUM(video.Segnalazioni) INTO X FROM video WHERE video.Canale=NEW.Canale;  
      IF (X=9)  
      THEN  
        UPDATE canale SET canale.Strike=canale.Strike+1 WHERE canale.Codice=NEW.Canale;  
      END IF; END IF;  
    END
```

Descrizione:

AggiornaStrike , se le nuove visualizzazioni del video sono maggiori di quelle vecchie si aggiorna visualizzazioni totali del canale aggiungendo un +1. dichiara 2 variabili int , Y e X. Y viene inserito dalla query che seleziona gli strike dal canale. Se Y= 0 allora preleva la somma di tutte le segnalazioni di ogni video appartenente al canale e le inserisce in X se X=3 allora inserisce uno strike, se X=6 inserisce un altro strike e se X=9 inserisce un altro strike.

2.UpdatePartnerInsertContrattoPassato

```
CREATE TRIGGER `UpdatePartnerInsertContrattoPass` AFTER UPDATE ON `partner`  
  FOR EACH ROW BEGIN  
  DECLARE Inz date; DECLARE conta integer;  
  if(NEW.Network!=OLD.Network)  
  THEN  
  SELECT count(NetworkPassati) into conta  
  FROM contrattopassato  
  WHERE canale=OLD.Canale;  
  IF(conta!=0) then  
  SELECT max(Fine) into Inz  
  FROM contrattopassato  
  WHERE canale=OLD.Canale;  
  ELSE  
  SELECT max(DataCaricamento) into Inz  
  FROM video  
  WHERE canale=OLD.Canale;  
  END IF;  
  insert into contrattopassato(Canale,NetworkPassati,Inizio,Fine) values  
  (OLD.Canale,OLD.Network,Inz,CURRENT_DATE);  
  END IF;  
  END
```

Descrizione:

Si dichiarano due variabili Inz di tipo date e conta di tipo integer.

Se il nuovo network è diverso da quello vecchio allora si contano i network passati e se li inseriscono in conta. Se conta è diverso da zero allora si seleziona la data di fine massima del contratto passato e la si inserisce in Inz. Altrimenti si sceglie la data massima fra i video appartenenti al canale e si inserisce il valore in Inz. Dopo si inserisce in contratto passato il canale e il network passato Inz come data di inizio e la data corrente come fine(CURRENT_DATE).

3. aggiornaCaratteristicheVideo

```
CREATE TRIGGER `aggiornaCaratteristicheVideo` AFTER INSERT ON `video`  
  FOR EACH ROW BEGIN  
  DECLARE X integer;  
  UPDATE canale  
  SET canale.Video_TOT=canale.Video_TOT+1  
  WHERE  canale.Codice=NEW.Canale;  
  
  SELECT COUNT(*) INTO X FROM impostazioni WHERE Video=NEW.URL;  
  
  IF X=0  
  THEN  
  INSERT INTO impostazioni(Video,Qualita)VALUES (NEW.URL,'144p');  
  END IF;  
  
  INSERT INTO impostazioni(Video,Qualita)VALUES (NEW.URL,'Auto');  
  END
```

Descrizione:

Si dichiara una variabile X integer . Si aggiornano i video totali del canale sommando 1.

Dopo di che si inserisce in x tutti i video di Impostazioni con l'URL uguale a quello della nuova tabella

Se X e uguale zero allora inserisci in impostazioni il video con le qualità 144p e Auto