

2

Specifiche dei Requisiti

TuTubi deve consentire la registrazione degli utenti di cui interessa il nome e la data di iscrizione. Gli utenti registrati possono pubblicare video, visualizzare quelli disponibili, oltre che esprimere su di essi valutazioni e commenti testuali.

In particolare, di un video pubblicato da un utente interessa conoscere il titolo, la durata e la descrizione (oltre che il nome del file dove questo viene memorizzato da parte del sistema). Inoltre, di ogni video interessa conoscere la categoria (unica) a cui appartiene, oltre che un insieme di parole chiave dette tag (almeno una) che ne descrivano il contenuto in modo più strutturato. Delle categorie e dei tag interessa il nome.

TuTubi deve anche permettere ad un utente di pubblicare un nuovo video segnalando che si tratta di una risposta ad un video già esistente (utile per temi come la politica dove vi possono essere posizioni diverse che stimolano un dibattito). *TuTubi* deve mantenere tale informazione in modo consistente, garantendo che nessun utente possa pubblicare un video in risposta ad un video pubblicato da sé stesso.

TuTubi deve poi fornire un servizio di cronologia, ricordando la sequenza di tutti i video visionati da ogni singolo utente, con relativa data e ora. Inoltre, per ogni video interessa conoscere il numero complessivo di volte che è stato visionato.

Gli utenti di *TuTubi* devono inoltre avere la facoltà di esprimere una valutazione (un valore da 0 –pessimo– a 5 –ottimo) per ogni video che visionano. Tali valutazioni serviranno poi al sistema per promuovere i video più belli (cfr. seguito) e scovare i più brutti. Tuttavia, per evitare degenerazioni nel meccanismo delle votazioni, l'utente che ha pubblicato un video non può votarlo, mentre ogni altro utente può votarlo al più una volta, indipendentemente dal numero di volte che l'ha visionato. In ogni caso però deve essere impossibile per un utente votare un video che non ha mai visionato.

Inoltre, per favorire uno spirito di comunità tra gli utenti del servizio, si prevede che questi possano esprimere commenti testuali ai video che visionano (dei quali interessa anche data e ora). A differenza delle valutazioni, un utente può esprimere più commenti

per lo stesso video. Anche qui, non deve essere permesso ad un utente di scrivere commenti per video che non ha mai visionato.

Ogni utente può creare delle *playlist* personali, ovvero delle collezioni ordinate di video che gradisce vedere, oppure vuole condividere con altri utenti. Le *playlist* infatti (di cui interessa il nome e la data di creazione) possono essere pubbliche o private: solo le *playlist* pubbliche possono essere visualizzate dagli altri utenti. A tal fine, il sistema deve permettere ad ogni utente di ottenere le *playlist* pubbliche di un altro utente a sua scelta.

TuTubi deve permettere ad un utente di iscriversi, pubblicare nuovi video, creare e modificare le sue *playlist*, ed esprimere valutazioni e commenti sui video che visiona.

Inoltre, *TuTubi* deve consentire la ricerca di video: in particolare, data una categoria, un insieme di tag, ed un intero v tra 0 e 5, si vogliono restituire tutti i video disponibili di quella categoria che posseggono almeno uno tra i tag indicati, e che abbiano una valutazione media di almeno v (se un video non ha ancora alcuna valutazione, deve essere restituito comunque). *TuTubi* deve poi permettere di cercare, data una categoria, i video di quella categoria che hanno il numero maggiore di video in risposta, al fine di isolare le discussioni più animate tra gli utenti.

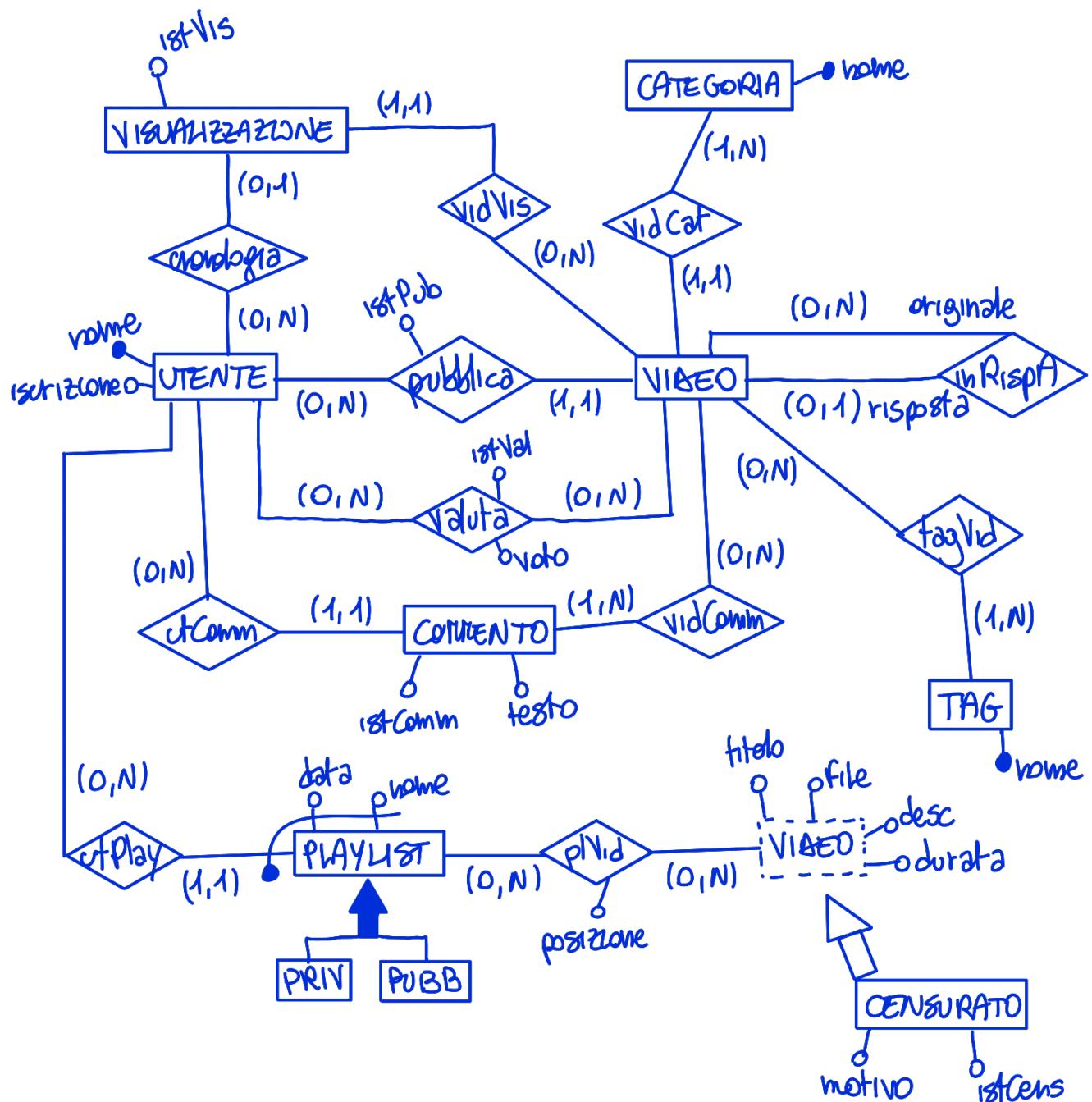
La redazione di *TuTubi* ha infine la facoltà di censurare dei video, ad esempio perché di contenuto coperto da copyright, osceno, ecc. Un video censurato non può essere né visionato, né votato, né commentato, né aggiunto ad alcuna *playlist*, né restituito come risultato di una ricerca. Un video, una volta censurato, non può tornare più visibile. Il motivo di una censura deve essere mantenuto nel sistema per usi interni.

Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Diagramma ER

Produrre un diagramma ER concettuale per l'applicazione in termini di entità, relationship, attributi, relazioni is-a, generalizzazioni (disgiunte) complete e non.



Dizionario dei dati Per ogni entità e relationship del diagramma ER **con** attributi o vincoli:

- Definire il dominio e la molteplicità degli attributi (se diversa da (1,1))
- Definire eventuali vincoli esterni in logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale, usando il seguente alfabeto:
 - Un simbolo di predicato $E/1$ per ogni entità E .
Semantica di $E(x)$: x è una istanza di E .
 - Un simbolo di predicato $D/1$ per ogni dominio D .
Semantica di $D(x)$: x è un valore di D .
 - Un simbolo di predicato r/n ($n > 0$) per ogni relationship n -aria r .
Semantica di $r(x_1, \dots, x_n)$: x_1, \dots, x_n è una istanza di r .
 - Un simbolo di predicato $a/2$ per ogni attributo a di entità
Semantica di $a(x, v)$: uno dei valori dell'attributo a dell'istanza x è v .
 - Un simbolo di predicato $a/(n+1)$ per ogni attributo a di relationship n -aria.
Semantica di $a(x_1, \dots, x_n, v)$: uno dei valori dell'attr. a dell'istanza (x_1, \dots, x_n) della relat. è v .
 - Opportuni simboli di predicato (soggetti a *semantica di mondo reale*) per gestire confronti tra valori di domini numerici o comunque ordinati (tra cui $</2$, $\leq/2$, $>/2$, $\geq/2$).
 - Il predicato di uguaglianza $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso).
 - Opportuni simboli di costante (soggetti a *semantica di mondo reale*), tra cui *adesso*, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>[1] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>CATEGORIA</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td><u>name</u></td><td><u>str</u></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>name</u>	<u>str</u>		<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>UTENTE</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td><u>iscrizione</u></td><td><u>data</u></td><td></td></tr> <tr> <td><u>name</u></td><td><u>str</u></td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>iscrizione</u>	<u>data</u>		<u>name</u>	<u>str</u>	
attributo	dominio	moltep. (*)														
<u>name</u>	<u>str</u>															
attributo	dominio	moltep. (*)														
<u>iscrizione</u>	<u>data</u>															
<u>name</u>	<u>str</u>															

<input type="checkbox"/> 3 Tipo: Entità Relationship (cerchiare)		
Nome: <u>valutazione</u>		
attributo	dominio	moltep. (*)
<u>istVal</u>	<u>dataora</u>	
<u>voto</u>	<u>int[0,5]</u>	

(*) solo se diversa da (1,1)

Vincoli:

[V.valuta.date]

$$\begin{aligned} & \forall u, v, i_p, i_v \text{ pubblica}(u, v) \\ & \wedge \text{istPub}(i_p, u, v) \wedge \text{valuta}(u, v) \\ & \wedge \text{istVal}(i_v, u, v) \rightarrow i_v > i_p \end{aligned}$$

<input type="checkbox"/> 5 Tipo: Entità Relationship (cerchiare)		
Nome: <u>pubblicazione</u>		
attributo	dominio	moltep. (*)
<u>istPub</u>	<u>dataora</u>	

(*) solo se diversa da (1,1)

Vincoli:

<input type="checkbox"/> 4 Tipo: Entità Relationship (cerchiare)		
Nome: <u>VIDEO</u>		
attributo	dominio	moltep. (*)
<u>durataSec</u>	<u>int>0</u>	
<u>titolo</u>	<u>str</u>	
<u>desc</u>	<u>str</u>	
<u>file</u>	<u>str</u>	

(*) solo se diversa da (1,1)

Vincoli:

<input type="checkbox"/> 6 Tipo: Entità Relationship (cerchiare)		
Nome: <u>VIDEO CONSUMATO</u>		
attributo	dominio	moltep. (*)
<u>motivo</u>	<u>str</u>	
<u>istCens</u>	<u>dataora</u>	

(*) solo se diversa da (1,1)

Vincoli:

[V.Censurato.data]

$$\begin{aligned} & \forall v, i_c \\ & \text{Censurato}(v) \wedge \text{istCen}(i_c, v) \rightarrow \\ & \exists u, i_p \text{ Video}(v) \wedge \text{pubblica}(v, u) \\ & \wedge \text{istPub}(i_p, v, u) \wedge i_c < i_p \end{aligned}$$

<p>7 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>TAG</u></p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td><u>name</u></td> <td><u>str</u></td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>name</u>	<u>str</u>		<p>9 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>COMMENTO</u></p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td><u>testo</u></td> <td><u>str</u></td> <td></td> </tr> <tr> <td><u>isCom</u></td> <td><u>dataora</u></td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>testo</u>	<u>str</u>		<u>isCom</u>	<u>dataora</u>	
attributo	dominio	moltep. (*)														
<u>name</u>	<u>str</u>															
attributo	dominio	moltep. (*)														
<u>testo</u>	<u>str</u>															
<u>isCom</u>	<u>dataora</u>															

<p>8 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>PLAYLIST</u></p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td><u>name</u></td> <td><u>str</u></td> <td></td> </tr> <tr> <td><u>data</u></td> <td><u>data</u></td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>name</u>	<u>str</u>		<u>data</u>	<u>data</u>		<p>10 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>vidPlaylist</u></p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td><u>posizione</u></td> <td><u>int > 0</u></td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	<u>posizione</u>	<u>int > 0</u>	
attributo	dominio	moltep. (*)														
<u>name</u>	<u>str</u>															
<u>data</u>	<u>data</u>															
attributo	dominio	moltep. (*)														
<u>posizione</u>	<u>int > 0</u>															

Ulteriori vincoli esterni, specifica di eventuali operazioni ausiliarie invocate da tali vincoli, e specifica dei domini concettuali non di tipo base

[V. Utente. valutaEspresso]

$$\forall u, v \quad \text{Utente}(u) \wedge \text{pubblica}(u, v) \rightarrow \neg \text{valuta}(u, v)$$

[V. Valuta. istante]

$$\begin{aligned} \forall u, vis, v, iVis, iVal \quad & \text{Utente}(u) \wedge \text{cronologia}(u, vis) \wedge \text{vidVis}(vis, v) \\ & \wedge \text{istVis}(iVis, vis) \wedge \text{valuta}(u, v) \wedge \text{istVal}(iVal, u, v) \rightarrow iVis \leq iVal \end{aligned}$$

[V. Commento. istante]

$$\begin{aligned} \forall u, vis, v, iVis, iComment \quad & \text{Utente}(u) \wedge \text{cronologia}(u, vis) \wedge \text{vidVis}(vis, v) \\ & \wedge \text{istVis}(iVis, vis) \wedge \text{utComm}(u, c) \wedge \text{Commento}(c) \wedge \\ & \text{istComm}(iComment, c) \wedge \text{vidComm}(v, c) \rightarrow iVis < iComment \end{aligned}$$

[V. Visualizz. Video]

$$\begin{aligned} \forall u, u', v, iV, ip, vis \quad & \text{Utente}(u) \wedge \text{pubblica}(u, v) \wedge \text{istPub}(ip, u, v) \\ & \wedge \text{Utente}(u') \wedge \text{cronologia}(u', vis) \wedge \text{istVis}(iV, vis) \wedge u \neq u' \\ & \wedge \text{vidVis}(vis, v) \rightarrow iV > ip \end{aligned}$$

[V. Iscrizione. data]

$$\begin{aligned} \forall u, d, v, ip \quad & \text{Utente}(u) \wedge \text{iscrizione}(d) \wedge \text{pubblica}(u, v) \\ & \wedge \text{istPub}(ip, v, u) \rightarrow d < ip \end{aligned}$$

Risposta alla Domanda 2 (segue)

[V. Video. rispostaSeStesso]

$$\forall u, v \text{ Utente}(u) \wedge \text{pubblica}(v, u) \wedge \text{pubblica}(v_r, u) \wedge v \neq v_r$$

$$\rightarrow \text{originale}(v, v_r)$$

[V. Video. posizionePlaylist]

$$\forall p, v, v', pos, pos' \text{ Playlist}(p) \wedge \text{Video}(v) \wedge \text{Video}(v') \wedge v \neq v'$$

$$\wedge \text{plVid}(p, v) \wedge \text{plVid}(p, v') \wedge \text{posizione}(pos, p, v)$$

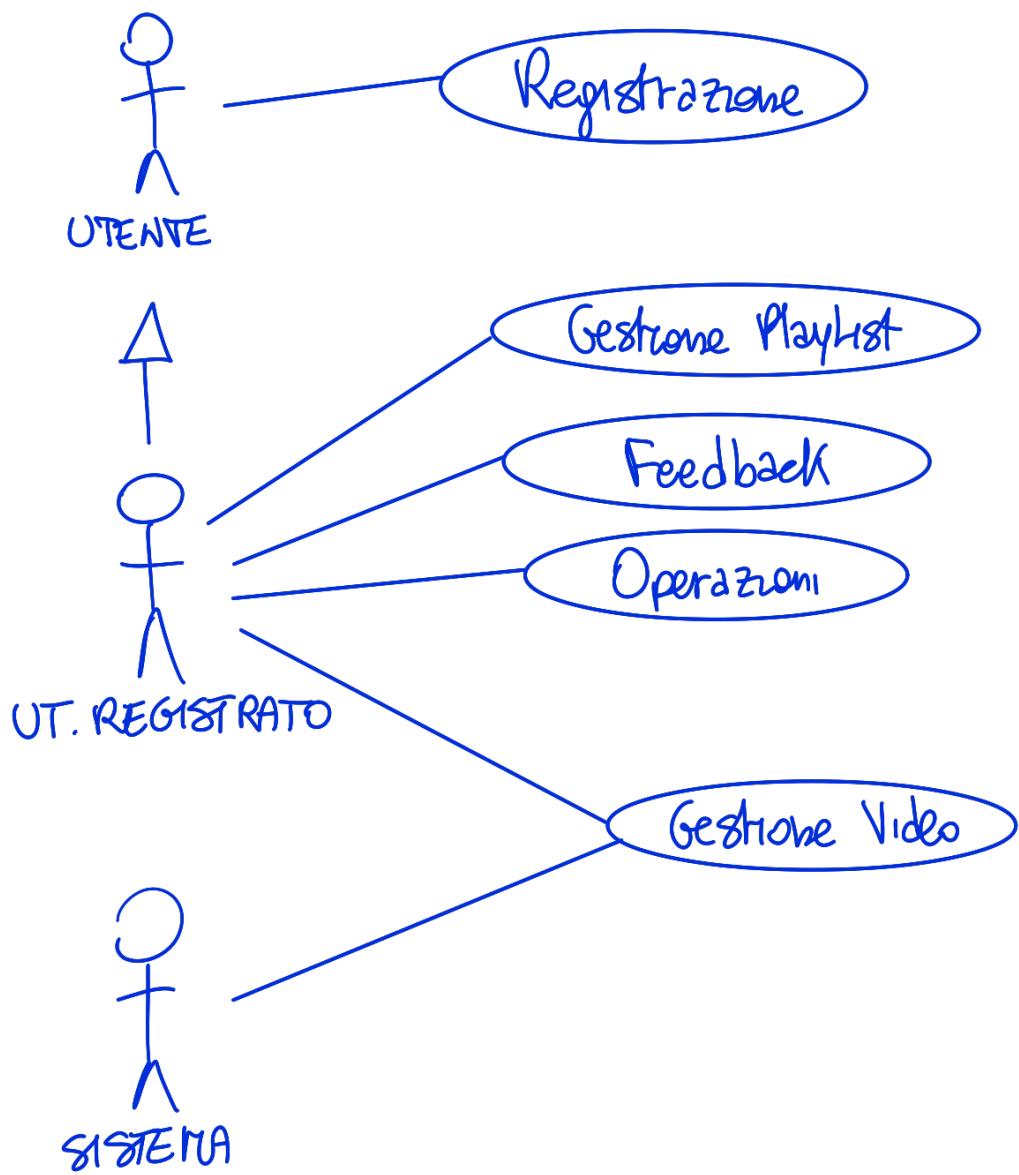
$$\wedge \text{posizione}(pos', p, v') \rightarrow pos > pos'$$

[V. inRispla. non suo]

$$\forall u, u', v_0, v_r \text{ Utente}(u) \wedge \text{Utente}(u') \wedge \text{pubblica}(v_0, u) \wedge \text{pubblica}(v_r, u')$$

$$\wedge \text{originale}(v_0) \wedge \text{inrisposta}(v_r) \rightarrow u \neq u'$$

Domanda 3 (5 minuti; 10 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

Risposta

Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo le operazioni degli use-case.

In particolare, per ogni use-case definito nella risposta alla **Domanda 3** definire la **segnatura** di tutte le operazioni che lo compongono, in termini di nome dell'operazione, nomi e dominio concettuale degli argomenti, dominio concettuale dell'eventuale valore di ritorno.

1 Specifica use-case: *Registrazione* (nome use-case)

Operazioni dello use-case:

iscrviUtente(no: str, d: data) : Utente

2 Specifica use-case: *Gestione Playlist* (nome use-case)

Operazioni dello use-case:

creaPlaylist(u: Utente, d: data, n: str) : Playlist

modificaPlay(p: Playlist)

3 Specifica use-case: *Gestione Video* (nome use-case)

Operazioni dello use-case:

censuraVideo(v: Video)

pubblicaVideo(t: str, f: str, d: str, durS: int > 0, u: Utente, c: CATEGORIA, tg: TAG) : Video

visualizzaVideo(i: dataora, v: Video) : Visualizzazione

4 Specifica use-case: Feedback (nome use-case)

Operazioni dello use-case:

valutazione(U : Utente, V : Video, i : dataora): int [0,5]

commento(U : Utente, V : Video, i : dataora, t : str): Commento

5 Specifica use-case: Operazioni (nome use-case)

Operazioni dello use-case:

creaPlaylist(no : str (0,1)): Pubblica(0,N)

~~█~~ trovaVideoVal(c : Categoria, t : Tag(1,N), V : int [0,5]): Video(0,N)

~~█~~ cercaVideoNumRisp(c : Categoria): Video(0,N)

6 Specifica use-case: Operazioni (nome use-case)

Operazioni dello use-case:

valMedia(V : Video): int [0,5]

ContaRisposte(V : Video): int ≥ 0

7 Specifica use-case: (nome use-case)

Operazioni dello use-case:

Domanda 5 (30 minuti; 60 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali per le operazioni di use-case, **limitandosi** a quelle necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra). In particolare, per ogni operazione, definire segnatura, precondizioni e postcondizioni utilizzando il linguaggio della logica del primo ordine. Si assuma lo stesso vocabolario definito alla [Domanda 2](#).

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Risposta

$\text{trovaVideoVal}(\text{c: Categoria}, \text{t: Tag}(1, N), \text{v: int } [0, 5]): \text{Video}(0, N)$

pre: nessuna

post: No side-effect

Val. ritorno:

$$\text{result} = \{ v \mid \text{Video}(v) \wedge \text{Categoria}(c) \wedge \text{Tag}(t) \wedge \text{vidCat}(v, c) \\ \wedge \text{tagVid}(t, v) \wedge \exists n \quad n = \text{valMedia}(v) \wedge \\ \wedge (n \geq v \vee n == \text{None}) \wedge \neg \text{Censurato}(v) \}$$

$\text{ValMedia}(\text{v: Video}): \text{int } [0, 5]$

pre: nessuna

post: No side-effect

Val. ritorno

$$v = \{ \text{vot} \mid \text{Video}(v) \exists u \quad \text{Utente}(u) \wedge \text{valuta}(v, u) \\ \wedge \text{voto}(v, u, \text{vot}) \}$$

$$\text{result} = \begin{cases} \text{se } v = \emptyset \rightarrow \text{result} = \text{None} \\ \text{altrimenti} \rightarrow \text{result} = \left[\frac{\sum_{\text{vot} \in v} \text{vot}}{|v|} \right] \end{cases}$$

Risposta alla Domanda 5 (segue)

cercaVideoNumRisp (c :Categoria) : Video(O, N)

pre: nessuna

post: No side-effect

Val. ritorno:

$$R = \{(v, n) \mid \text{Video}(v) \wedge \text{Categoria}(c) \wedge \text{vidCat}(v, c) \\ \wedge \exists n = \text{ContaRisposte}(v) \wedge \neg \text{Censurato}(v)\}$$

$$\text{Popolari} = \underset{(v, n) \in R}{\operatorname{argmax}}(n)$$

$$\text{result} = \{v \mid \text{Video}(v) \wedge v \in \text{Popolari}\}$$

ContaRisposte (v : Video) : Int ≥ 0

pre: nessuna

post: No side-effect

Val. ritorno

$$\text{result} = \{vr \mid \text{Video}(vr) \wedge \text{risposta}(v, vr) \wedge \neg \text{Censurato}(v)\}$$

2 Progettazione della base dati e delle funzionalità

Domanda 6 (20 minuti; 30 minuti al massimo) Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema ER concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- progettare una corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario per ogni entità
- valutare se e come aggiungere ridondanza in maniera controllata
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

Descrivere brevemente le principali scelte effettuate.

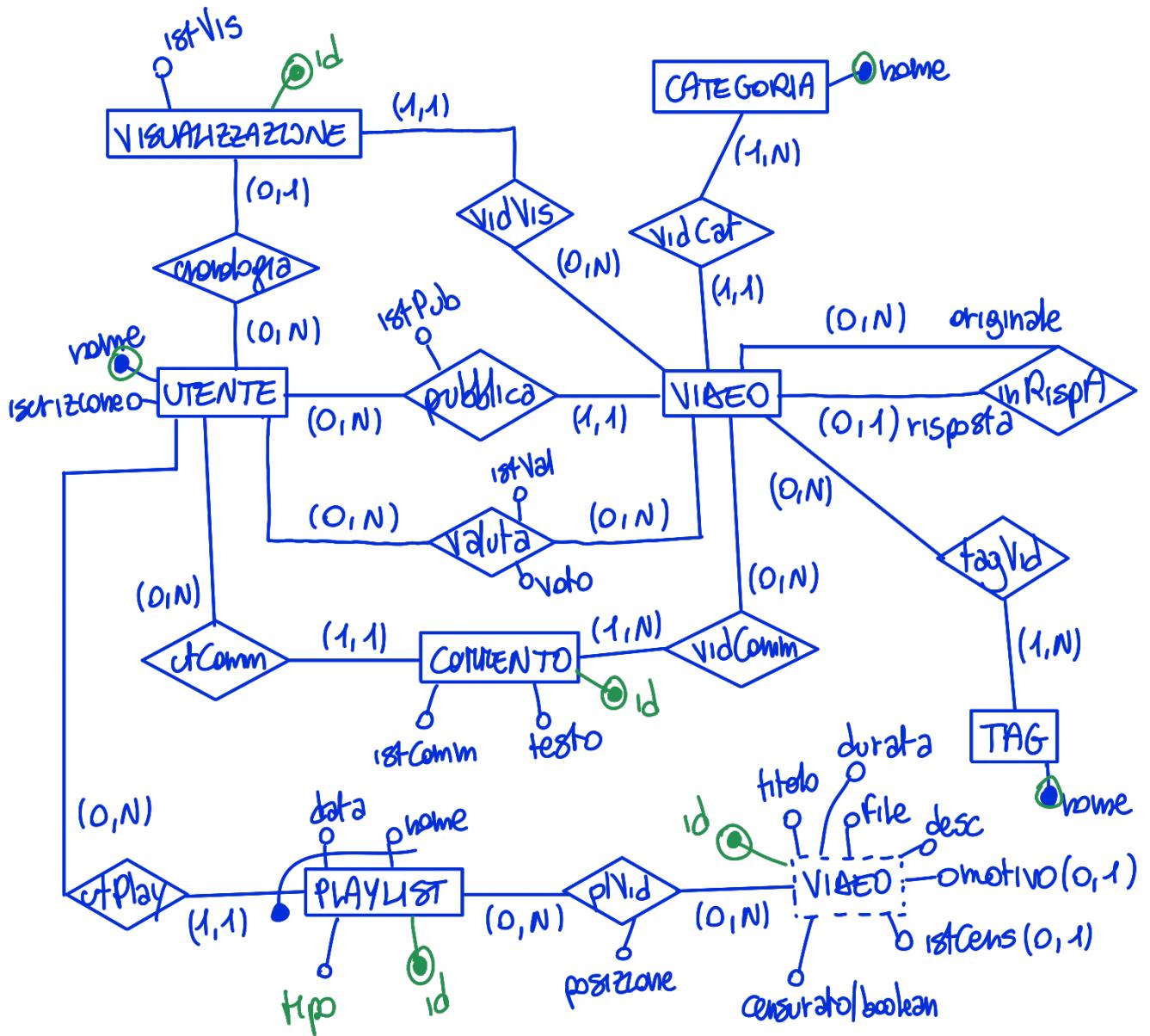
DBMS da utilizzare *PostgreSQL*.....

Corrispondenza tra domini concettuali e domini supportati dal DBMS

```
create domain StringS as varchar(50);
"      "      StringM as varchar(200);
"      "      StringL as varchar(500);
"      "      IntegerGZ as integer check (value > 0);
"      "      IntegerGEZ as integer check (value ≥ 0);
"      "      Int_Voto as integer check (value ≥ 0 and value ≤ 5);
```

```
create type Tipol as enum ('Public','Private')
```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Fusione dell'is-a in Playlist

Fusione dell'is-a in Video

Vincoli esterni introdotti o modificati durante la fase di ristrutturazione
(si omettano i vincoli esterni la cui formulazione è rimasta identica a seguito della ristrutturazione)

[V. Playlist.tipo]

$\forall p \text{ Playlist}(p) \rightarrow [\text{tipo}(p, \text{public}) \vee \text{tipo}(p, \text{private})]$

[V. Video.censurato]

$\forall v \text{ Video}(v) \wedge \text{censurato}(v, \text{true})$

$\rightarrow \exists m, i \text{ motivo}(m, v) \wedge \text{motCens}(i, v)$

Domanda 7 (30 minuti; 60 minuti al massimo) Proseguire la fase di progettazione logica della base di dati producendo lo schema relazionale della base dati e i relativi vincoli a partire dallo schema ER ristrutturato.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

1 Relazione	CATEGORIA	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>name</u>				
Domini	<u>StringS</u>				

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

Inclusione: name ⊆ vidCat(categoria)

La relazione accorda le relazioni che implementano le seguenti relationship:

2 Relazione	VISUALIZZAZIONE	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>1stVis</u>	<u>video</u>	<u>utente*</u>	
Domini	<u>integer</u>	<u>timestamp</u>	<u>integer</u>	<u>StringS</u>	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): **seriale:id**

PK: video refer Video(id)

PK: utente refer Utente(name)

La relazione accorda le relazioni che implementano le seguenti relationship: **vidVis, cronologica**

3 Relazione	UTENTE	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>name</u>	<u>descrizione</u>			
Domini	<u>StringS</u>	<u>timestamp</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti relationship:

4 Relazione	VIDEO	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	<u>titolo</u>	<u>durata</u>	<u>descr</u>	<u>censurato</u> <u>motivo*</u> <u>1stCens*</u>
Domini	<u>integer</u>	<u>StringM</u>	<u>IntegerG</u>	<u>StringL</u>	<u>boolean</u> <u>StringL</u> <u>timestamp</u>

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): **seriale:id**

ennupla: censurato = TRUE → motivo ≠ NULL AND 1stCENS ≠ NULL

La relazione accorda le relazioni che implementano le seguenti relationship:

5 Relazione	VIDEO	(nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>categoria</u>	<u>utente</u>	<u>1stPub</u>		
Domini	<u>StringS</u>	<u>StringS</u>	<u>timestamp</u>		

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

PK: categoria refer Categoria(name)

PK: utente refer Utente(name)

La relazione accorda le relazioni che implementano le seguenti relationship: **vidCat, pubblica**

6 Relazione	<u>INRISPA</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>Voriguale</u> <u>VinRisposta</u>			
Domini	<u>integer</u> <u>integer</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

PK: Voriguale refer Video(id)

PK: VinRisposta refer Video(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

7 Relazione	<u>votata</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>utente</u> <u>video</u> <u>1stVal</u> <u>voto</u>			
Domini	<u>string</u> <u>integer</u> <u>timestamp</u> <u>int_Voto</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

PK: utente refer Utente(name)

PK: video refer Video(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

8 Relazione	<u>COMMENTO</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u> <u>testo</u> <u>1stComm</u> <u>utente</u>			
Domini	<u>integer</u> <u>string</u> <u>timestamp</u> <u>string</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): seriale: id

PK: utente refer Utente(name)

Inclusione: id ⊆ vidComm(commento)

La relazione accorda le relazioni che implementano le seguenti relationship: utComm

9 Relazione	<u>TAG</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>name</u>			
Domini	<u>string</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

Inclusione: name ⊆ tagVid(Tag)

La relazione accorda le relazioni che implementano le seguenti relationship:

10 Relazione	<u>PLAYLIST</u> (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>name</u> <u>data</u> <u>id</u> <u>tipo</u> <u>utente</u>			
Domini	<u>string</u> <u>date</u> <u>integer</u> <u>Tipop</u> <u>string</u>			

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): seriale: id

enumpl: tipo = PUBLIC ∨ tipo = PRIVATE PK: utente refer Utente(name)

chiave(name, data, utente)

La relazione accorda le relazioni che implementano le seguenti relationship: utPlay

11 Relazione ... <u>p1Nid</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>PlayList</u> <u>Video</u> <u>posizione</u>	
Domini <u>integer</u> <u>integer</u> <u>integer62</u>	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

PK: playList refer PlayList(id)

PK: video refer Video(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

12 Relazione ... <u>v1dComm</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>Video</u> <u>Commento</u>	
Domini <u>integer</u> <u>integer</u>	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

PK: video refer Video(id)

PK: commento refer Commento(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

13 Relazione(nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti relationship:

14 Relazione(nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti relationship:

15 Relazione(nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti relationship:

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennuple); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

[V. Valuta. istante]

Operaz: insert o modify in Video

Istante: before

Funzione:

1 isError = exists(select *

```
from Video v, Utente u
where v.nome = new.utente
and v.censurato = False
and new.istPub > v.istCens )
```

2 if isError: blocca l'operazione

3 else: permetti l'operazione

Risposta alla Domanda 7 (segue)

[V.Valuta.date]

Operaz: insert o modify in Video

Istante: before

Funzione:

1 isError = exists(select *

from Video v, Utente u, Valuta val

where v.nome = newv.utente

and val.nome = newv.utente

and val.video = v.id

and newv.istPub > val.istValuta)

2 if isError: blocca l'operazione

3 else: permetti l'operazione

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di use-case definite per modellare i requisiti contrassegnati dalla barra laterale della specifica dei requisiti.

In particolare, per ogni operazione definire la segnatura, in termini di nome dell'operazione, nomi e dominio SQL degli argomenti, dominio SQL dell'eventuale valore di ritorno, e un algoritmo in pseudo-codice con SQL immerso che verifichi le precondizioni e garantisca il raggiungimento delle postcondizioni definite in fase di Analisi.

Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

Risposta

`trovaVideoVal (c:StringS, <t:StringS>, v:Int_Voto):(<vid:Integer>)`

`pre: nessuna`

`post:`

`Q = select vid.id`

`from Video vid, tagVid tg,`

`(select avg(v.voto) as media,`

`vid.id as video`

`from valuta v, Video vid, Utente u`

`where v.video = vid.id`

`and v.utente = u.nome)\ Valutazione val`

`where vid.categoria = :c and tg.video = vid.id`

`and tg.tag = :t and v.censurato = false`

`and (val.media >= :v or val.media = NULL)`

`return Q`

Risposta alla Domanda 8 (segue)

cercaVideo NumRisposte (c: StringS) : (<vid: integer>)

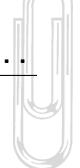
pre: nessuna

post:

Q = select vid.id, max(ris.num_risposte)
From Video vid,
(select count(r.risposta) as num_risposte
from inRisposta r, Video vid
where r.risposta=vid.id) Risposte ris
where vid.categoria=:c and vid.censurato=false
group by vid.id

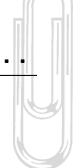
return Q

Tempo totale stimato per svolgere questa prova: 180 minuti (tempo totale concesso: 300 minuti).
[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]



[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]

[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]



[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]