



Sapienza Università di Roma
Facoltà di Ing. dell'Informazione, Informatica e Statistica, Laurea in Informatica
Insegnamento di **Basi di Dati, Modulo 2**
Prof. Toni Mancini
Dipartimento di Informatica
<http://tmancini.di.uniroma1.it>

Esame BD2.Esame.Risposte – Modulo risposte prova scritta (diagramma delle classi UML)

Dati dello studente e dell'esame

Cognome e nome: Matricola:

Data:

Corso di laurea e canale di appartenenza:

- Laurea in Informatica, canale 1 (Prof. G. Perelli)
- Laurea in Informatica, canale 2 (Prof.ssa M. De Marsico)

Firma di un membro della Commissione per
avvenuta identificazione:
.....

Rinuncia alla prova

Desidero rinunciare a questa prova d'esame. Firma:





Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

<http://tmancini.di.uniroma1.it>

Progetto 20080401 (P.20080401)

TuTubi

Versione 2024-02-08



Indice

Indice	1
1 Introduzione	2
2 Specifica dei Requisiti	3

1

Introduzione

Si vuole progettare e realizzare *TuTubi*, un sistema informatico da usarsi su un portale Web per la pubblicazione e condivisione di file video da parte degli utenti registrati. Il sistema deve permettere la memorizzazione dei video e la loro visualizzazione da parte degli utenti. Questi devono poter anche esprimere valutazioni e commenti.

2

Specifiche dei Requisiti

- Cloud *Vincoli e altro*
- Cloud *Cose ci serve Giocare*
- Cloud *Con Cose interagisca l'utente*
- Cloud *Cose va monitorato*

TuTubi deve avere la capacità degli utenti, di cui interessa il titolo e la durata. Gli utenti registrati possono utilizzare video visionando quelli disponibili oltre che esprimere su di essi valutazioni.

5 In particolare, di un video interessa conoscere il titolo, la durata e la valutazione (oltre che il titolo dove questo viene memorizzato da parte del sistema). Inoltre, di ogni video interessa conoscere la categorizzazione (unica) a video, 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.

10 TuTubi deve anche permettere ad un utente di pubblicare un nuovo video segnalando che si tratta di una nuova pubblicazione (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 video in risposta ad un video pubblicato da sé stesso.

15 TuTubi deve poi fornire un modo di votare, ricordando le votazioni scritte da ogni singolo utente, relativa a ciascun video. Inoltre, per ogni video interessa conoscere il numero di voti e la percentuale di voti positivi.

Gli utenti di TuTubi devono inoltre avere la facoltà di votare (valutare) i video (pubblicati o non pubblicati) 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 un volto, 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 sulle pubblicazioni (dei quali interessa l'autore). A differenza delle valutazioni, un utente può scrivere tanti commenti.

per lo stesso video. Anche qui, [REDACTED] non è permesso [REDACTED]

30

Ogni utente può creare delle playlist personali, ovvero delle collezioni di tracce che gradisce vedere, oppure vuole condividere con altri utenti. Le playlist infatti (di cui interessa il titolo e la data di creazione) possono essere pubblicate su Internet, rendendo pubbliche per chiunque si accinge a visitare il sito dell'utente. A tal fine, il sistema deve permettere agli utenti di creare delle playlist pubbliche di un altro utente a sua scelta.

35

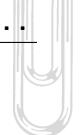
TuTubi deve essere un luogo sicuro, pubblico e aperto, dove tutti possono condividere la loro creatività.

Inoltre, *TuTubi* deve consentire la discussione: in particolare, dato un video, *TuTubi* deve restituire i commenti relativi a quel video, almeno uno tra i tag indicati dalla valutazione associata (se un video non ha ancora alcuna valutazione, deve essere restituito comunque). *TuTubi* deve poi permettere di scrivere, dato uno o più genitori, i commenti relativi a quel video, il numero maggiore di interazioni per categoria, al fine di isolare le discussioni più animate tra gli utenti.

46

La redazione di *TuTubi* ha infine la facoltà di [REDACTED], ad esempio perché di contenuto coperto da copyright, osceno, ecc. Un video censurato non può essere visualizzato, né verranno commentati o aggiunto ad alcun play list, e non riceverà alcuna ricchezza. Un video, una volta censurato, non può tornare più visibile. Il motivo di una censura deve essere mantenuto nel sistema per usi interni.

50



Istruzioni e regole d'esame

Prima dell'esame

- Stampare questo modulo, preferibilmente fronte-retro, e rilegarlo con un fermaglio rimovibile, come quello disegnato in alto
- Compilare il frontespizio con i propri dati, come richiesto
- Scrivere la propria matricola nello spazio apposito nella parte alta di tutte le pagine

Durante l'esame

- La prova è dimensionata per essere svolta in circa 3 ore. Tuttavia, data la sua natura fortemente progettuale, la Commissione offre agli studenti la più ampia disponibilità di tempo, al fine ovviare ad eventuali (e limitati) errori di analisi/progettazione rilevati più a valle del ciclo di vita.
Il tempo massimo per la consegna è quindi rilassato a 5 ore (il massimo tempo compatibile con le disponibilità di aule).
- Scrivere le risposte negli spazi predisposti sotto le relative domande. Le ultime pagine sono vuote e possono essere usate come minute oppure, se puntate opportunamente, per contenere risposte in caso gli spazi appositi dovessero risultare insufficienti.
- Non è possibile usare alcun tipo di materiale didattico.
- In caso di necessità di ulteriori fogli (in proprio possesso), chiedere preventivamente alla Commissione una nuova procedura di controllo.
- La Commissione può rispondere solo a brevi domande inerenti al testo dei quesiti.
- Tra la seconda e la quarta ora d'esame, gli studenti possono effettuare **brevi pause** (uno studente alla volta) seguendo la seguente procedura:
 1. Alla lavagna è riportata una coda denominata 'Coda prenotazioni pause'. Sia n (un intero) l'elemento in fondo alla coda (si assuma $n = 0$ in caso di coda vuota).
 2. Recarsi alla lavagna ed aggiungere l'intero $n + 1$ come proprio contrassegno in fondo alla coda, seguito da una stringa a propria scelta (ad es., le proprie iniziali).
 3. Se il proprio contrassegno non è l'elemento affiorante della coda, tornare al lavoro in attesa che lo diventi.
 4. Consegnare tutti i fogli di lavoro e il testo d'esame alla Commissione ed uscire.
 5. Al rientro, cancellare il proprio contrassegno dalla coda di modo da permettere al successivo studente prenotato di uscire, e riprendere i fogli prima consegnati.

Al momento della consegna

- Ordinare tutti i fogli che si vuole far valutare e rilegarli con un fermaglio rimovibile. Non includere fogli che la Commissione non deve valutare (ad es., requisiti, minute), ma includere ovviamente il frontespizio.
- Consegnare i fogli ordinati **nelle mani** di un membro della Commissione. **Non lasciare l'aula senza la conferma, da parte della Commissione, del buon esito delle operazioni di consegna.**

In caso di rinuncia

- È possibile rinunciare alla consegna a partire dalla seconda ora d'esame. In caso di rinuncia, consegnare nelle mani della Commissione solo il frontespizio, dopo aver compilato e firmato la sezione dedicata.

Sommario delle domande

Si richiede di progettare l'applicazione descritta dalla specifica dei requisiti effettuando le fasi di Analisi concettuale dei requisiti e di Progettazione logica della base dati e delle funzionalità, utilizzando la metodologia vista nel corso.

In particolare (vengono indicati i tempi suggeriti per i diversi passi chiave):

Parte 1: Analisi concettuale dei requisiti Effettuare la fase di Analisi concettuale dei requisiti producendo lo schema concettuale per l'applicazione, che includa:

- Analisi dei dati (45 minuti; 75 minuti al massimo):
 - un diagramma UML concettuale delle classi (*)
 - (parte del)le specifiche formali delle classi e delle associazioni
 - le specifiche dei tipi di dato
 - la specifica formale dei vincoli esterni (*)
- Analisi delle funzionalità:
 - un diagramma UML degli use-case (5 minuti; 10 minuti al massimo)
 - la segnatura di tutte le operazioni di use-case (10 minuti)
 - (parti del)le specifiche formali degli use-case. (30 minuti; 60 minuti al massimo)

Si richiede *esplicitamente* di modellare le specifiche formali delle operazioni di classe e/o use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), *inclusa* tutte le eventuali operazioni ausiliarie, usando l'estensione della logica del primo ordine studiata nel corso. (*)

Parte 2: Progettazione della base dati e delle funzionalità Effettuare la progettazione della base dati e delle funzionalità a partire dallo schema concettuale prodotto nella Parte 1, ed in particolare eseguire i seguenti passi:

- Progettazione della base dati relazionale con vincoli:
 - Ristrutturazione del diagramma UML concettuale delle classi e delle specifiche (20 minuti; 30 minuti al massimo):
 - * scelta del DBMS da utilizzare
 - * progettazione della corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
 - * ristrutturazione del diagramma UML concettuale delle classi e delle specifiche dei vincoli esterni.
 - Produzione dello schema relazionale della base dati e dei relativi vincoli (*) (30 minuti; 60 minuti al massimo)
- Progettazione delle funzionalità (30 minuti; 45 minuti al massimo):
 - definizione della specifica realizzativa delle operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, in modo conforme alla loro specifica concettuale prodotta nella fase di Analisi, in termini di algoritmi in pseudo-codice e comandi SQL immersi. (*)

Le pagine seguenti contengono le domande specifiche a cui è richiesto rispondere, ulteriori delucidazioni per ogni singolo punto, e spazi per le risposte.

Le pagine da 31 in poi possono essere utilizzate per scrivere minute che non verranno valutate.

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

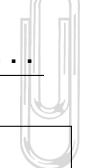


Questa pagina è stata intenzionalmente lasciata vuota

1 Analisi concettuale

Domanda 1 (10 minuti) Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni e ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile. (La risposta a questa domanda non sarà valutata, ma si consiglia di svolgere accuratamente questo passo, in quanto può facilitare di molto le attività di progetto.)

Risposta



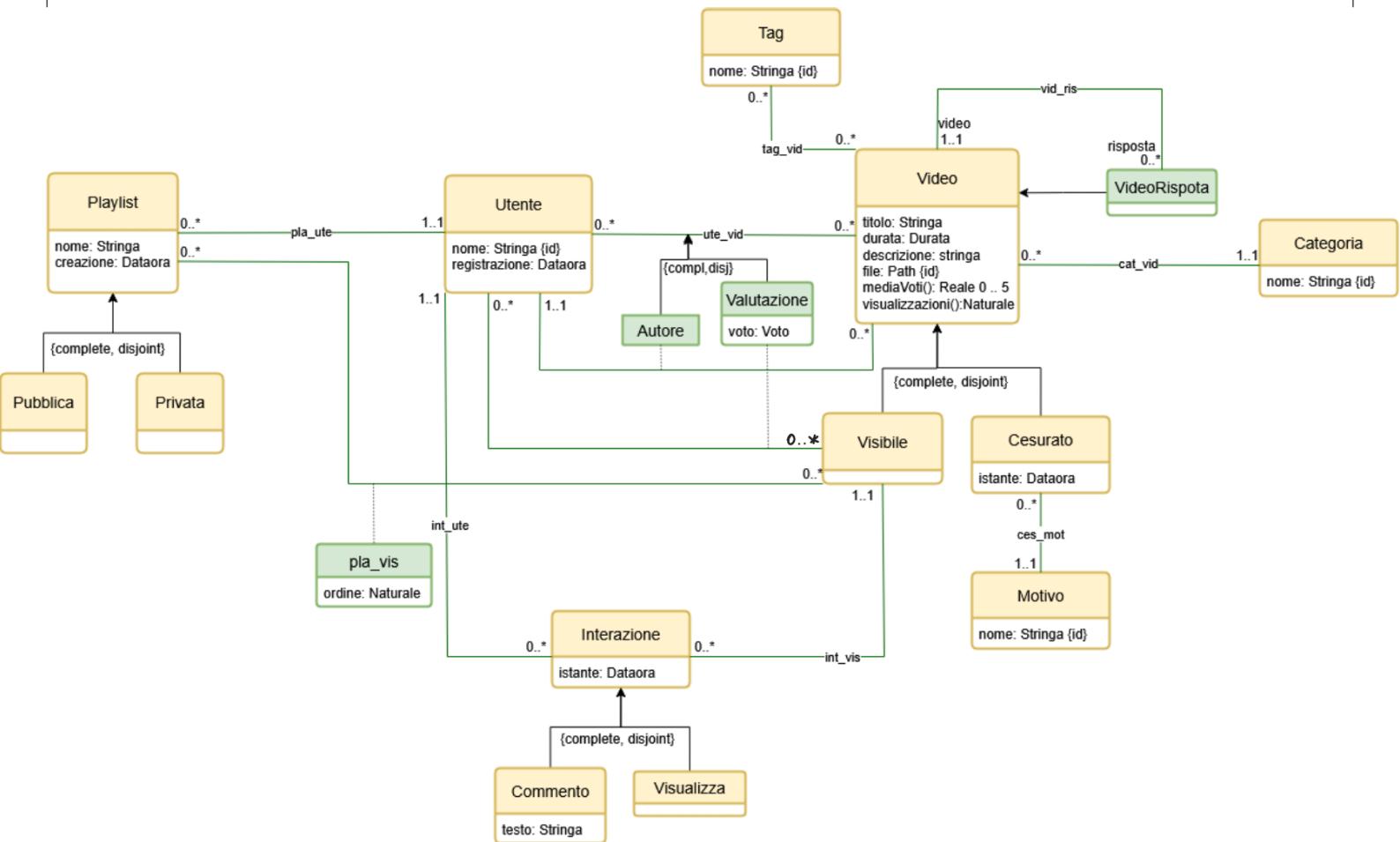
Risposta alla Domanda 1 (segue)

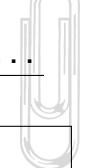
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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

Diagramma UML concettuale delle classi

Produrre un diagramma UML concettuale delle classi per l'applicazione in termini di classi, associazioni, attributi, generalizzazioni, operazioni di classe.





Risposta alla Domanda 2 (segue)

Specifiche delle classi o associazioni Per ogni classe o associazione del diagramma **con** operazioni o vincoli:

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
 - Un simbolo di predicato $C/1$ per ogni classe C .
Semantica di $C(x)$: x è una istanza di C .
 - Un simbolo di predicato $T/1$ per ogni tipo di dato T .
Semantica di $T(x)$: x è un valore di T .
 - Un simbolo di predicato $\text{assoc}/2$ per ogni associazione binaria assoc .
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc .
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità.
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attrib. attr del link (c_1, c_2) è v .
 - Un simbolo di predicato $\text{op}/(n+2)$ per ogni operazione di classe ad n argomenti.
Semantica di $\text{op}(c, \text{arg}_1, \dots, \text{arg}_n, v)$: uno dei valori di ritorno di op , quando invocata sull'istanza c e con argomenti $\text{arg}_1, \dots, \text{arg}_n$ è v .
 - Il simbolo di $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <i>Video</i>.....</p> <p>Operazioni, vincoli:</p> <p>[V. no. risposte. a. se. stessi]</p> <p>$\forall_{v, v'} \exists w \quad (\text{Video}(v) \wedge \text{VideoRisposta}(v') \wedge v \neq v' \wedge \text{Autore}(w, v) \wedge \text{Autore}(w, v'))$</p> <p>$\rightarrow \neg \text{vid_ris}(v, v')$</p> <p>[0. visualizzazioni]</p> <p><i>Visualizzazione</i>; (i): NATURALE</p> <p>Pre:</p> <p>Post:</p> <p>Result = $\left \left\{ v \mid \text{visualizzazione}(v) \wedge \text{int_vis}(v, \text{THIS}) \right\} \right$</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <i>Utente</i>.....</p> <p>Operazioni, vincoli:</p> <p>[V. vota. solo. dopo. V. si. no]</p> <p>$\forall_{u, v} \quad (\text{utente}(u) \wedge \text{video}(v) \wedge (\neg \exists i \quad \text{visualizza}(i) \wedge \text{int_ute}(i, u) \wedge \text{int_vis}(i, v)))$</p> <p>$\rightarrow \neg \text{Valutazione}(u, v)$</p> <p>[V. commenta. solo. dopo. Visione]</p> <p>$\forall_{u, v} \quad (\text{utente}(u) \wedge \text{video}(v) \wedge (\neg \exists i \quad \text{visualizza}(i) \wedge \text{int_ute}(i, u) \wedge \text{int_vis}(i, v)))$</p> <p>$\rightarrow (\neg \exists c \quad \text{commento}(i) \wedge \text{int_ute}(c, u) \wedge \text{int_vis}(c, v))$</p>
--	---

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <i>playlist</i>.....</p> <p>Operazioni, vincoli:</p> <p>[V. creazione.playlist. dgs. Reg. utente]</p> <p>$\forall p, u, c, r$ $playlist(p) \wedge utente(u) \wedge pl_ute(p, u) \wedge creazione(p, c) \wedge registrazione(u, r)$ $\rightarrow c \geq r$</p>	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <i>interazione</i>.....</p> <p>Operazioni, vincoli:</p> <p>[V. data.interazione. dgs. reg.registration]</p> <p>$\forall n, i, r, t$ $(interazione(i) \wedge istante(i, t) \wedge utente(u) \wedge registrazione(u, r) \wedge int_ute(i, w))$ $\rightarrow i \geq r$</p>
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <i>VIDEO</i>.....</p> <p>Operazioni, vincoli:</p> <p>[O.mediaVot:]</p> <p>mediaVot(): Reale 0..5</p> <p>Pre: $\exists u \text{ valutazione}(u, \text{this})$</p> <p>Post:</p> $V = \{(u, v) \mid \exists u \text{ utente}(u) \wedge \text{valutazione}(u, \text{this}) \wedge \text{voto}(u, \text{this}, v)\}$ $\text{Result} = \frac{\sum_{(u, v) \in V} (v)}{ V }$	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>
<p>5 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>	<p>8 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>

Specifiche dei tipi di dato, specifiche di ulteriori vincoli esterni ed altre specifiche

Stringa = tutti i caratteri

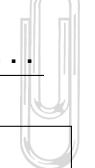
DataOra = valore composto da data + ora

Durata = intero > 0 (rappresenta i secondi)

Naturale = intero ≥ 0

Voto = 0 .. 5

PATH = regex path ' \n .. \$ '



Risposta alla Domanda 2 (segue)

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





Questa pagina è stata intenzionalmente lasciata vuota

Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo la **segnatura** delle operazioni in ogni use-case.

Risposta

- GestionalVideo

CENSURA(v:video, m:motivo): CENSURATO

- REGISTRAZIONE

ISCRIVITI(n:stringa): utente

- VALUTAZIONE E COMMENTI:

VALUTA(v:video, v:voto, u:utente): ute_vis

COMMENTA(v:video, u:utente, t:stringa): INTERAZIONE

- RICERCA

CERCAVIDEO(C:categoria, T:tags[0..*], v:voto): visible [0..*]

CERCA TRENDVIDEO(C:categoria): visible

- PUBBLICAZIONE

PUBBLICA(u:utente, P:PATH, d:stringa, t:durata, n:stringa): video

- VISUALIZZAZIONE

CRONOGRAFIA(u:utente): video [0..*]

- PLAYLIST

CERCAPLAYLIST(u:utente): Playlist [0..*]

AGGIUNGI(P:Playlist, u:utente, v:video, o:Natura): Pla_vis



Questa pagina è stata intenzionalmente lasciata vuota

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), ed includendo eventuali operazioni ausiliarie. 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

Registrazione ($n: \text{stringa}$): Utente

Pre: $\neg \exists u \text{ Utente}(u) \wedge \text{nome}(v, n)$

Post:

IL Livello estensionale dei dati: al termine dell'esecuzione della funzione
differisce da quello di partenza

nuovo elemento α

Utente(α), nome(α, n), registrazione(α , adesso)

Result = α

CercaTRENDVideo ($c: \text{categoria}$): visible

Pre: $\exists v \text{ videoRisposta}(v) \wedge \text{cat_vid}(c, v)$

Post:

$$V = \left\{ (v, n) \mid \text{visible}(v) \wedge \text{cat_vid}(v, c) \wedge n = \left\{ r \mid \text{videoRisposta}(r) \wedge \text{vid_ris}(v, r) \right\} \right\}$$

$$(v_{\max}, n_{\max}) \in \underset{(v, n) \in V}{\operatorname{Argmax}}(n)$$

Result = v_{\max}

CercaViduo ($c: \text{categoria}, t: \text{Tag} [\square..*\square], x: \text{Voto}$): visible [\square..*\square]

Pre: nessuna

Post:

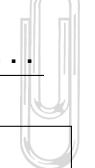
$$\text{Result} = \left\{ v \mid \text{visible}(v) \wedge \text{cat_vid}(c, v) \wedge (\exists t \text{ tag}(t) \wedge \text{tag_vid}(t, v) \wedge t \in T) \wedge \left[\begin{array}{l} \exists u, m \text{ valutazione}(u, v) \wedge \text{mediaVoti}(v, m) \\ \longrightarrow x <= m \end{array} \right] \right\}$$

CercaPlaylist ($u: \text{utente}$): [\square..*\square]

Pre: nessuna

Post:

$$\text{Result} = \left\{ p \mid \text{Pubblica}(p) \wedge \text{pla_ute}(p, u) \right\}$$



Risposta alla Domanda 5 (segue)

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 UML delle classi concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

- progettare una corrispondenza tra i tipi di dato 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 classe
- ristrutturare i vincoli esterni per renderli consistenti con la struttura del nuovo diagramma.

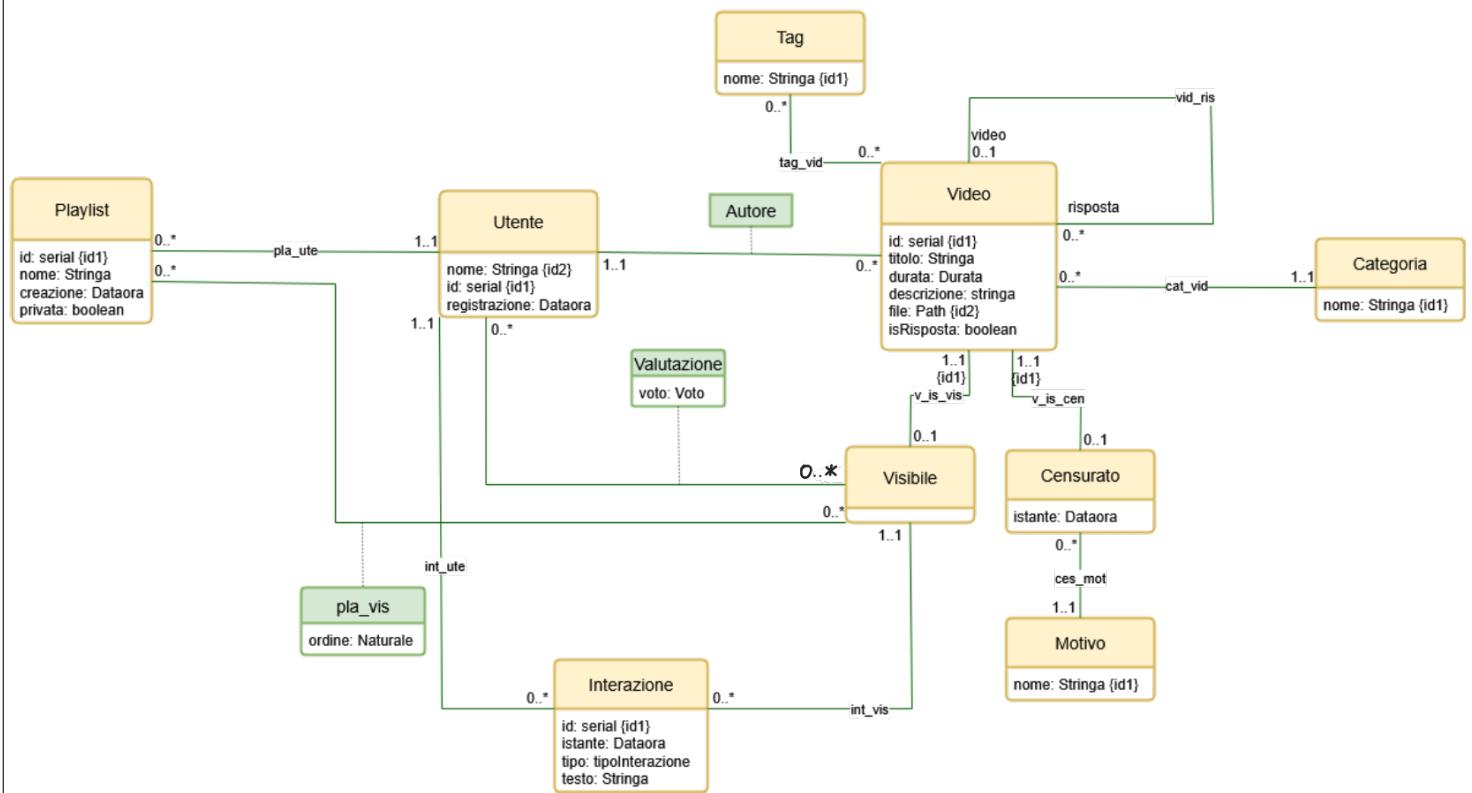
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare PostgreSQL.....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```
Create Domain Stringe AS TEXT
Create Domain Dataore AS timestamp
Create Domain Durata AS INT check (value > 0)
Create Domain Naturale AS INT check (value >= 0)
Create Domain Voto AS INT check (value >= 0 AND value <= 5)
Create Domain PATH AS TEXT check (value ~'..')
Create type Tipolinterazione AS ENUM ('commento', 'visualizzazione')
```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Rimosso IS-A playlist : fusione
 Rimosso IS-A interazione : fusione
 Rimosso IS-A video-Risposta : fusione
 Rimosso IS-A visible-Censurato : sostituzione
 Rimosso IS-A valutazione-autore : divisione
 Aggiunte chiavi primarie
 Rimosse operazioni dal diagramma

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. IS-A Risposta-video]

$\nexists v \text{ video}(v) \rightarrow (\text{isRisposta}(v, \text{true}) \leftrightarrow \exists r \text{ video}(r) \wedge \text{vid-ris}(v, r))$

[V. IS-A Visible & Censurato. Completato]

$\forall v \text{ video}(v) \rightarrow (\exists vs \text{ v-is-vs}(v, vs) \vee \exists c \text{ v-is-cen}(v, c))$

[V. IS-A Visible & Censurato. disinolti]

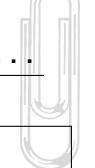
$\rightarrow \exists v, vs, c \text{ video}(v) \wedge \text{v-is-vs}(v, vs) \wedge \text{v-is-cen}(v, c) \wedge \text{censurato}(c) \wedge \text{visible}(vs)$

[V. IS-A Interazione]

$\forall i \text{ interazione}(i) \wedge \text{tipo}(i, \text{'visualizzazione'}) \rightarrow (\neg \exists t \text{ Testo}(i, t))$

[V. IS-A Valutazione-Autore]

$\forall u, v \text{ utente}(u) \wedge \text{video}(v) \wedge \text{Autore}(u, v) \rightarrow (\neg \text{VALUTAZIONE}(u, v))$



Risposta alla Domanda 6 (segue)

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 UML delle classi ristrutturato.

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

[1] Relazione ..UTENTE..... (nome)	Derivante da: classe associazione (cerchiare)
---	---

Attributi <u>id</u> nome registrato	
Domini serial Stringe Data/ore	

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):

Unique (nome)

La relazione accorda le relazioni che implementano le seguenti associazioni:

[2] Relazione ..PLAYLIST..... (nome)	Derivante da: classe associazione (cerchiare)
---	---

Attributi <u>id</u> nome creazione privata utente	
Domini serial Stringe Date/ore Boolean int	

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):

FK utente Nat utente (id)

La relazione accorda le relazioni che implementano le seguenti associazioni:

[3] Relazione ..CATEGORIA..... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>nome</u>	
Domini Stringe	

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 associazioni:

[4] Relazione ..TAG..... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>nome</u>	
Domini Stringe	

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 associazioni:

[5] Relazione ..tag-video..... (nome)	Derivante da: classe associazione (cerchiare)
--	---

Attributi <u>tag</u> <u>video</u>	
Domini Stringe int	

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):

FK tag Nat Tag (name)

FK video Nat video (id)

La relazione accorda le relazioni che implementano le seguenti associazioni:

6 Relazione .VIDEO..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>id</u> titolo durata descrizion file isRisposta Categorie Risposta *	
Domini Stringa Stringa Durata Stringa Path Boolean Stringa int	

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):

Unique(file) check ((risposta is true AND risposta is not null) OR (risposta is false AND risposta is null))

FK Risposta Ref Video(id)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... cat-video, vid-ru...

7 Relazione .Video(Gont.).... (nome)	Derivante da: classe associazione (cerchiare)
Attributi Autore	
Domini int	

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 Autore Ref Utente (id)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... Autore

8 Relazione ..VISIBLE..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Video</u>	
Domini int	

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):

FK video Ref video (id)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... V-!-vis

9 Relazione ..CENSURATO..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>video</u> istante motivo	
Domini int Dataora Stringa	

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):

FK video Ref video (id)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... v-is-cen, cens-mot

10 Relazione ..Motivo..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Nome</u>	
Domini Stringa	

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 associazioni:

11 Relazione ...HA_VU..... (nome)	Derivante da: classe associazione (cerchiare)							
Attributi <u>Playlist</u> <u>visibile</u> ordine								
Domini int int Natural								

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):

Unique (Playlist, ordine)
 FK Playlist ref Playlist(id)
 FK visibile ref visibile(video)

La relazione accorda le relazioni che implementano le seguenti associazioni:

12 Relazione ...INTERAZIONE.... (nome)	Derivante da: classe associazione (cerchiare)							
Attributi <u>id</u> istante TIPO testo playlist visibile								
Domini ferial Datazione tipointerazione Stringe int int								

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):

FK Playlist ref Playlist(id)
 FK visibile ref visibile(video)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...int-ute, ...int-vis

13 Relazione .VALUTAZIONE.... (nome)	Derivante da: classe associazione (cerchiare)							
Attributi <u>utente</u> <u>visibile</u> voto								
Domini int int voto								

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):

FK utente ref Utente(id)
 FK visibile ref visibile(video)

La relazione accorda le relazioni che implementano le seguenti associazioni:

14 Relazione (nome)	Derivante da: classe associazione (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 associazioni:

15 Relazione (nome)	Derivante da: classe associazione (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 associazioni:

16	Relazione (nome)	Derivante da: classe associazione (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 associazioni:

17	Relazione (nome)	Derivante da: classe associazione (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 associazioni:

18	Relazione (nome)	Derivante da: classe associazione (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 associazioni:

19	Relazione (nome)	Derivante da: classe associazione (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 associazioni:

20	Relazione (nome)	Derivante da: classe associazione (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 associazioni:

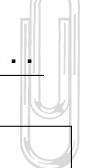
Ulteriori vincoli esterni

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.

[T. VN_Autore_nono_Votare_1c-nu-velvo]

Before insert or update on Valutazione

```
ISERROR = EXISTS (Select 1 FROM video  
                    Where autore = NEW.utente  
                      AND ID = NEW.visible )  
IF ISERROR Then RAISE exception  
ELSE permit  
  
Non ti faccio tutti i Trigger
```



Risposta alla Domanda 7 (segue)

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di classe e/o 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. Specificare, per ogni operazione, se debba essere implementata nel DBMS o nel *back-end*.

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

Risposta

CercaTRENDVideo(*C*: Stringa): int

```
ISERROR = EXISTS ( SELECT 1 FROM video
                    WHERE categoria = C AND isRisposta = TRUE )
IF ISERROR THEN
    RAISE exception
```

```
SELECT v.RISPOSTA
FROM Video v JOIN visible vs ON vs.video = v.id
WHERE v.categoria = C AND v.isRisposta = TRUE
GROUP BY v.RISPOSTA
ORDER BY COUNT(*) DESC
LIMIT 1
```

VERSIONE CHE GESTISCE PIÙ MASSIMI

```
CercaTRENDVideo(C: Stringa): INSIEME(<INT>)
ISERROR = EXISTS ( SELECT 1 FROM video
                    WHERE categoria = C AND isRisposta = TRUE )
IF ISERROR THEN
    RAISE exception
WITH RISP_Count AS (
    SELECT v.RISPOSTA, COUNT(*) AS NUM_RISPOSTE
    FROM Video v JOIN visible vs ON vs.video = v.id
    WHERE v.categoria = C AND v.isRisposta = TRUE
    GROUP BY v.RISPOSTA
)
SELECT RISPOSTA
FROM RISP_Count
WHERE num_risposte = (SELECT MAX(num_risposte)
                      FROM RISP_Count)
```

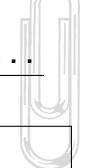
mediaVoti(*v*: int): REAL

```
ISERROR = SELECT 1 FROM valutazione
            WHERE visible = v
IF ISERROR THEN
    RAISE exception
```

```
SELECT AVG(f.voto)
FROM valutazione f
WHERE f.visible = v
```

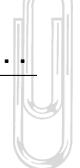
CercaVideo(*C*: Stringa, *T*: INSIEME(Stringa), *x*: VOTO): INSIEME(<INT>)

```
Select v.id
FROM video v JOIN tag_vid tv ON tv.video = v.id
WHERE v.categoria = C AND tv.tag = ANY(T)
AND MEDIAVOTI(v.id) >= x
```



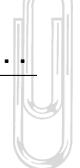
Risposta alla Domanda 8 (segue)

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.]