



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.ER – Modulo risposte prova scritta**

Dati dello studente e dell'esame

Cognome e nome: Casv Marco Matricola:

Data: 22/05/2024

Corso di laurea e canale di appartenenza:

- ☒ Laurea in Informatica, canale 1 (A-L, Prof. G. Perelli)
☐ Laurea in Informatica, canale 2 (M-Z, Prof.ssa M. De Marsico)
☐ Laurea in Informatica in Modalità Teledidattica Unitelma Sapienza

Firma di un membro della Commissione per
avvenuta identificazione:

.....

Rinuncia alla prova

☐ Desidero rinunciare a questa prova d'esame. Firma:



Questo modulo è ottimizzato per la stampa fronte-retro



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 ER concettuale (*)
 - il relativo dizionario dei dati
 - le specifiche dei domini concettuali non di tipo base
 - eventuali vincoli esterni, espressi utilizzando il linguaggio della logica del primo ordine (*)
- 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)
 - la specifica delle operazioni di use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra) in termini di precondizioni e postcondizioni, utilizzando il linguaggio della logica del primo ordine (*) (30 minuti; 60 minuti al massimo)

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 ER concettuale e dei vincoli esterni (20 minuti; 30 minuti al massimo):
 - * scelta del DBMS da utilizzare
 - * progettazione della corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
 - * ristrutturazione del diagramma ER concettuale e 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 di use-case 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. (*)

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

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 33 in poi possono essere utilizzate per scrivere minute che non verranno valutate.



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

1. Utente

- 1.1 nome
- 1.2 iscrizione
- 1.3 video pubblicati
 - 1.3.1 istante pubblicazione
- 1.4 video visualizzati
 - 1.4.1 commenti [0..*]
 - 1.4.1.2 istante commento
 - 1.4.2 valutazione [0..1]
 - 1.4.3 istante visualizzazione

2. Video

- 2.1 titolo
- 2.2 durata in secondi
- 2.3 descrizione
- 2.4 nome File
- 2.5 categoria
- 2.6 tag [1..*]
- 2.7 risposta?
 - 2.7.1 video alla quale si risponde
- 2.8 operazione visualizzazioni()
- 2.9 Censurato?
 - 2.9.1 istante censura
 - 2.9.2 motivo

3 Playlist

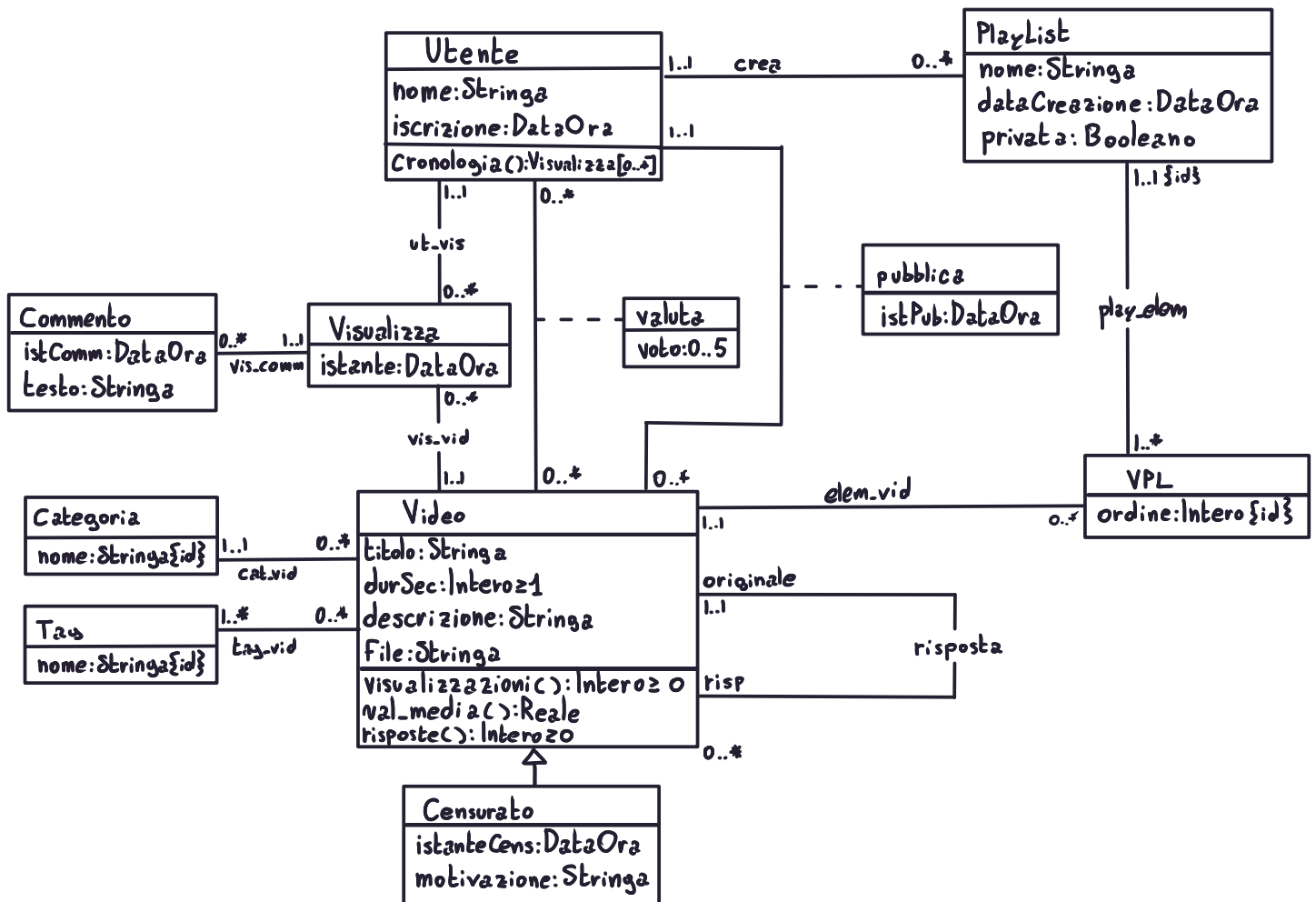
- 3.1 nome
- 3.2 data creazione
- 3.3 visibilità
- 3.4 video coinvolti (almeno 1)
 - 3.4.1 ordine
- 3.5 utente

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

<div>1</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Visualizzazione</div> <table><thead><tr><th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div> <div>[V.visualizzato_dopo-pubblicato]</div> <div>$\forall vis, vid, u, dv, dp [vid_vis(vis, vid) \wedge pubblica(u, vid, dp) \wedge istante(vis, dv)] \rightarrow dv \geq dp$</div> <div>[V.visualizzato-prima-di-censura]</div> <div>$\forall vis, vid, dv, dc [vis_vid(vis, vid) \wedge istante(vis, dv) \wedge Censurato(vid) \wedge istante Cens(vid, dc)] \rightarrow dc > dv$</div>	attributo	dominio	moltepl. (*)				<div>2</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Utente</div> <table><thead><tr><th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr></thead><tbody><tr><td></td><td></td><td></td></tr></tbody></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div> <div>[V.visualizza_dopo.iscrizione]</div> <div>$\forall u, vid, vis, dv, di [ut_vis(u, vis) \wedge vis_vid(vis, vid) \wedge istante(vis, dv) \wedge iscrizione(u, di)] \rightarrow dv \geq di$</div> <div>[V.pubblica_dopo.iscrizione]</div> <div>$\forall u, vid, di, dp [pubblica(u, vid, dp) \wedge iscrizione(u, di)] \rightarrow dp \geq di$</div>	attributo	dominio	moltepl. (*)			
attributo	dominio	moltepl. (*)											
attributo	dominio	moltepl. (*)											

3 Tipo: **Entità** | Relationship (cerchiare)Nome: **Utente**

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.no_valutazioni_a_se_stessi]$ $\forall u, vid [pubblica(u, vid) \rightarrow \neg valuta(u, vid)]$ $[V.valuta_se_visualizzato]$

$$\forall u, vid [valuta(u, vid) \rightarrow [\exists v \text{ ut-vis}(u, v) \wedge vis_vid(v, vid)]]$$
5 Tipo: **Entità** | Relationship (cerchiare)Nome: **Commento**

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.commentato_dopo_visualizzato]$

$$\forall vis, vid, comm, dv, dc [vis_vid(vis, vid) \wedge istante(vis, dv) \wedge ist_comm(comm, dc) \wedge vis_comm(vis, comm)] \rightarrow dc \geq dv$$
4 Tipo: **Entità** | Relationship (cerchiare)Nome: **Playlist**

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.crea_playlist_dopo_iscrizione]$

$$\forall p, u, dc, di [crea(u, p) \wedge iscrizione(u, di) \wedge dataCreazione(p, dc)] \rightarrow dc \geq di$$
 $[V.censurati_no_playlist]$ $\forall v, e [elem_video(e, v)] \rightarrow \neg Censurato(v)$ 6 Tipo: **Entità** | Relationship (cerchiare)Nome: **Video**

attributo	dominio	moltepl. (*)

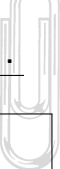
(*) solo se diversa da (1,1)

Vincoli:

 $[V.no_auto_risposte]$

$$\forall v_1, v_2, u_1, u_2 [risposta(v_1, v_2) \wedge pubblica(u_1, v_1) \wedge pubblica(u_2, v_2)] \rightarrow u_1 \neq u_2$$
 $[V.no_risposte_al_futuro]$

$$\forall v_1, v_2, u_1, u_2, dp_1, dp_2 [pubblica(u_1, v_1, dp_1) \wedge pubblica(u_2, v_2, dp_2) \wedge risposta(v_1, v_2)] \rightarrow dp_1 < dp_2$$



<div>11</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome:</div> <table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"><div></div></td></tr></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	<div></div>			<div>13</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome:</div> <table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"><div></div></td></tr></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	<div></div>		
attributo	dominio	moltepl. (*)											
<div></div>													
attributo	dominio	moltepl. (*)											
<div></div>													

<div>12</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome:</div> <table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"><div></div></td></tr></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	<div></div>			<div>14</div> <div>Tipo: Entità Relationship (cerchiare)</div> <div>Nome:</div> <table><tr><td>attributo</td><td>dominio</td><td>moltepl. (*)</td></tr><tr><td colspan="3"><div></div></td></tr></table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	<div></div>		
attributo	dominio	moltepl. (*)											
<div></div>													
attributo	dominio	moltepl. (*)											
<div></div>													

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

$byteFlow = [0..1]\{0..*\}$

UTENTE

$Cronologia(): Video[0..*]$

• pre-cond: nessuna

• post-cond: $V = \{v \mid \exists vid \text{ ut_vis}(this, v) \wedge vis_vid(v, vid) \wedge \neg Censurato(vid)\}$

Result = V

VIDEO

$visualizzazioni(): Intero \geq 0$

pre-cond: nessuna

post-cond: $V = \{v \mid vis_vid(v, this)\}$

Result = |V|

$val_media(): Reale$

• pre-cond: nessuna

• post-cond: $V = \{(u, v) \mid valuta(this, u) \wedge voto(this, u, v)\}$

Result e' tale che:

$|V| = 0 \rightarrow Result = -1$

$|V| > 0 \rightarrow Result = \sum_{(u,v) \in V} v \cdot \frac{1}{|V|}$

$risposte(): Intero \geq 0$

• pre-cond: nessuna

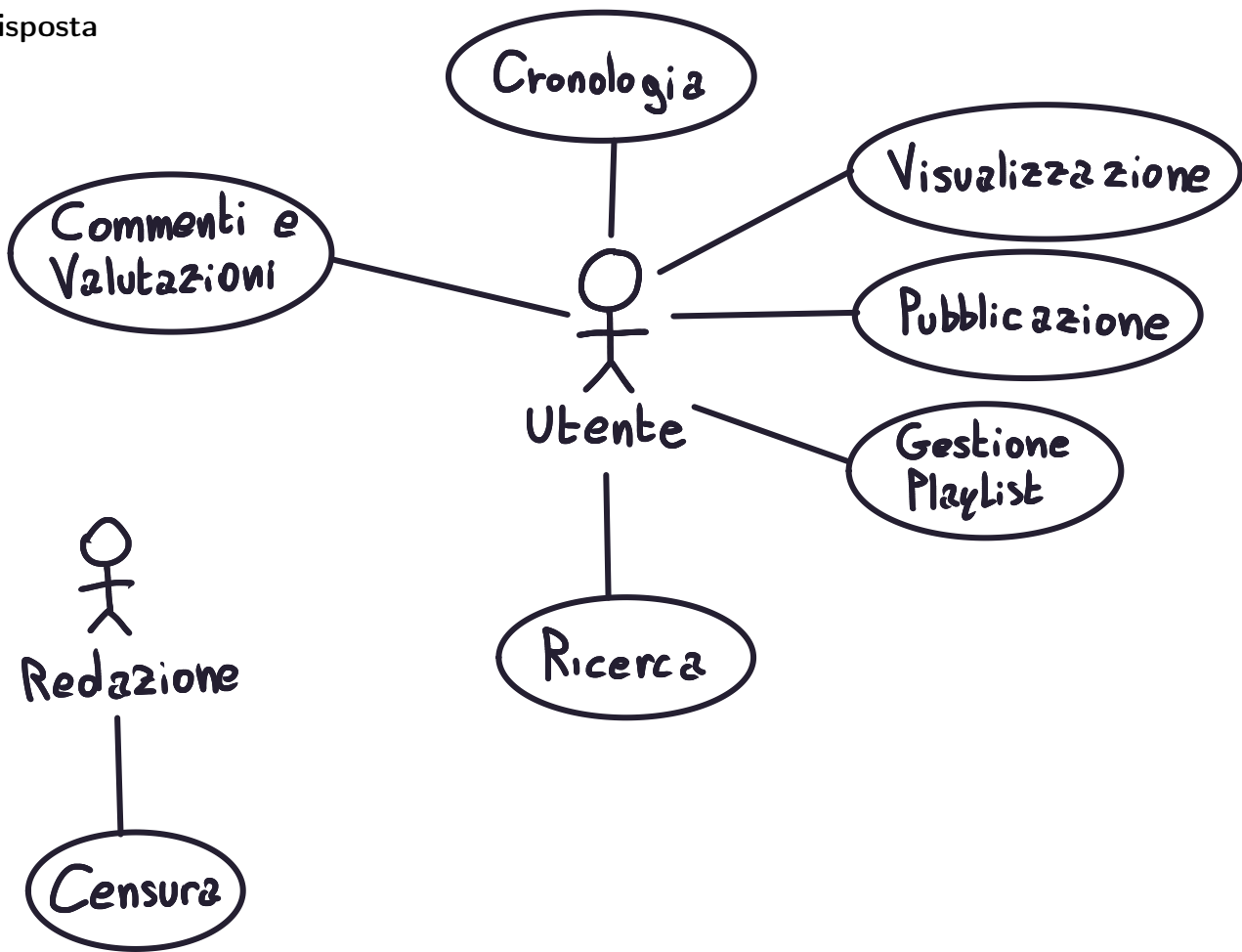
• post-cond: $V = \{v \mid risposta(v, this)\}$

Result = |V|

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



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: *Cronologia* (nome use-case)

Operazioni dello use-case:

visualizza_cronologia(): (Video, DataOra)[0..*]

2 Specifica use-case: *Commenti e Valutazioni* (nome use-case)

Operazioni dello use-case:

commenta(v:Video, c:String):Commento

valuta(v:Video, r:1..5):valutazione

3 Specifica use-case: *Visualizzazione* (nome use-case)

Operazioni dello use-case:

visualizza(v:Video):Visualizza

visualizza_playlist(u:Utente):Playlist[0..]*

4 Specifica use-case: *Pubblicazione* (nome use-case)

Operazioni dello use-case:

pubblica(v:byteFlow, t:Stringa, d:Stringa):Video

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

Operazioni dello use-case:

crea(n:Stringa, p:Booleano):Playlist

rimuovi(v:Video, p:Playlist)

aggiungi(v:Video, p:Playlist):VPL

visibilita'(p:Playlist, b:Booleano)

6 Specifica use-case: *Censura* (nome use-case)

Operazioni dello use-case:

censura(v:Video, m:Stringa):Censurato

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

Operazioni dello use-case:

cerca_video(c:Categoria, t:Tag, v:0..5):Video[0..]*

cerca-discussioni(c:Categoria):Video[0..]*

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

$\text{cerca_video}(c:\text{Categoria}, T:\text{Tag}[1..7], v:0..5):\text{Video}[0..*]$

• pre_cond : nessuna

• post_cond :
$$V = \left\{ \text{vid} \mid \left[\exists k \text{ val_media}(\text{vid}, k) \wedge [k \geq v \vee k = -1] \right] \wedge \begin{array}{l} \text{cat_vid}(\text{vid}, c) \wedge \exists t \text{ tag_vid}(\text{vid}, t) \wedge t \in T \end{array} \right\}$$

$\text{Result} = V$

$\text{cerca_discussioni}(c:\text{Categoria}):\text{Video}[0..*]$

• pre_cond : nessuna

• post_cond : $R = \{nr \mid \exists v \text{ cat_vid}(v, c) \wedge \text{risposte}(v, nr)\}$

m è tale che:

$|R| = 0 \rightarrow m = -1 \wedge$

$|R| \neq 0 \rightarrow m = \max(R)$

$V = \{v \mid \text{cat_video}(v, c) \wedge \text{risposte}(v, m)\}$

$\text{Result} = V$

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 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 multivalore 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.

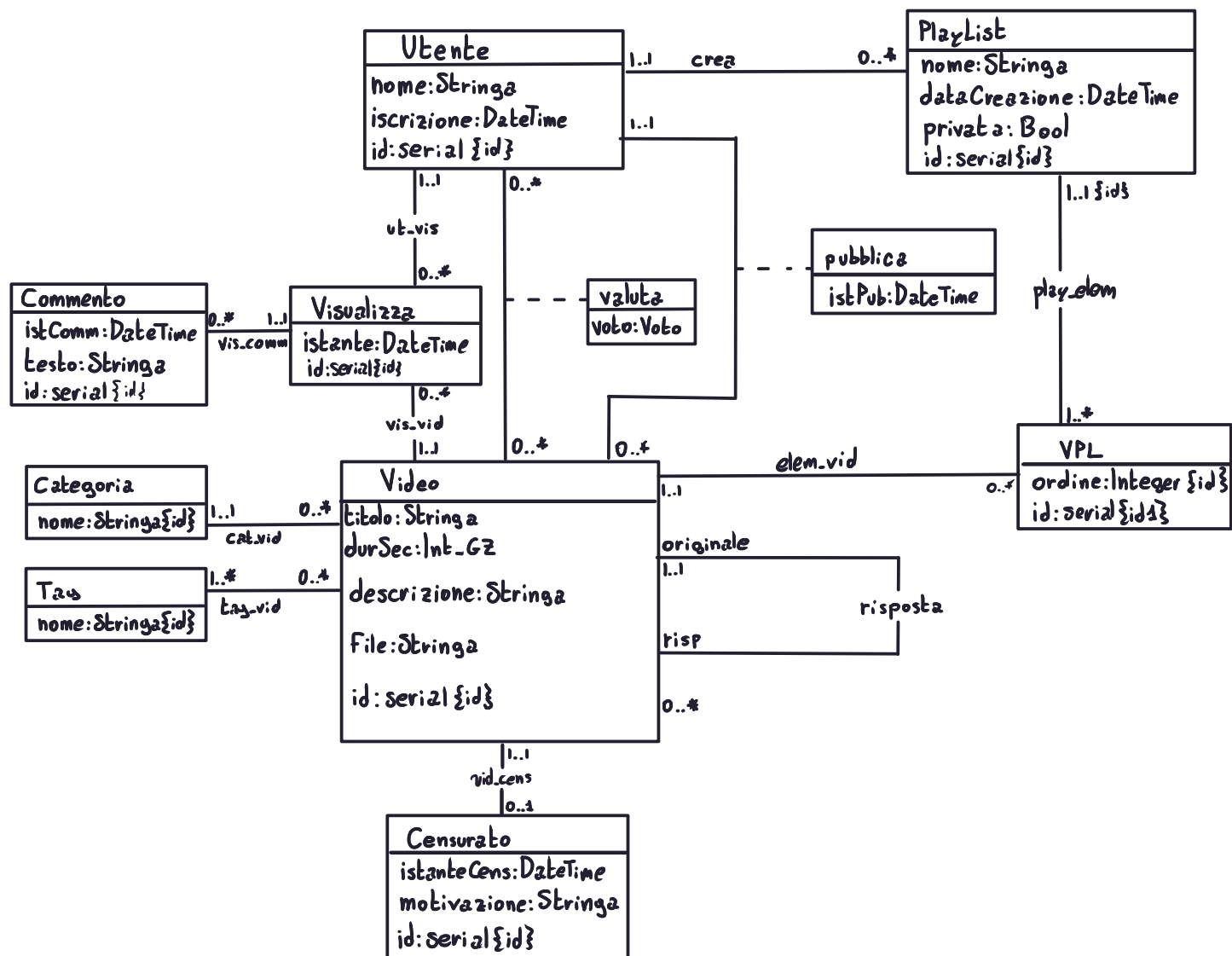
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare PostgreSQL

Corrispondenza tra domini concettuali e domini supportati dal DBMS

```
create domain Stringa as varchar NOT NULL;  
create domain Int_G2 as integer check(value > 0);  
create domain Int_GE2 as integer check(value >= 0);  
create domain Voto as integer check(value >= 0 and value <= 5);
```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

La generalizzazione fra Video e Censurato e'
diventata una associazione

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)

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 ER ristrutturato.

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

1 Relazione Utente (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>	<u>iscrizione</u>	<u>id</u>					
Domini	Stringa	DateTime	serial					

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 accorpa le relazioni che implementano le seguenti relationship:

2 Relazione Visualizza (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>istante</u>	<u>id</u>	<u>utente</u>	<u>video</u>				
Domini	DateTime	serial	serial	serial				

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):

Foreign key utente references Utente(id);

Foreign key video references Video(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: vis-vid, ut-vis

3 Relazione Commento (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>istCom</u>	<u>testo</u>	<u>id</u>	<u>vis</u>				
Domini	DateTime	Stringa	serial	serial				

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):

Foreign key vis references Visualizza(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ...vis-comm

4 Relazione Video (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>titolo</u>	<u>durSec</u>	<u>descrizione</u>	<u>id</u>	<u>cat</u>	<u>risposta*</u>	<u>creator</u>	<u>istPub</u>
Domini	Stringa	Int_G2	Stringa	serial	Stringa	serial	serial	DateTime

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):

Foreign key cat references Categoria(nome); check(risposta <> id);

Foreign key risposta references Video(id); Foreign key creator references Utente(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ...cat-vid, risposta_pubblica

5 Relazione Categoria (nome) Derivante da: entità | relationship (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 accorpa le relazioni che implementano le seguenti relationship:

6 Relazione Tay (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>							
Domini	<u>Stringa</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 accorpa le relazioni che implementano le seguenti relationship:

7 Relazione Tag-vid (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>tag</u>	<u>video</u>						
Domini	<u>Stringa</u>	<u>serial</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):

video occorre in Video(id)

La relazione accorpa le relazioni che implementano le seguenti relationship:

8 Relazione Censurato (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>istanteCens</u>	<u>motivazione</u>	<u>video</u>					
Domini	<u>DateTime</u>	<u>Stringa</u>	<u>serial</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 accorpa le relazioni che implementano le seguenti relationship: vid_cens

9 Relazione valuta (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>utente</u>	<u>video</u>	<u>voto</u>					
Domini	<u>serial</u>	<u>serial</u>	<u>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):

La relazione accorpa le relazioni che implementano le seguenti relationship:

10 Relazione Playlist (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>id</u>	<u>nome</u>	<u>dataCreazione</u>	<u>privata</u>	<u>creator</u>			
Domini	<u>serial</u>	<u>Stringa</u>	<u>DateTime</u>	<u>Bool</u>	<u>serial</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):

Foreign key creator references Utente(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: crea

11 Relazione VPL.....(nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ordine</u>	<u>id</u>	<u>pl</u>	<u>video</u>				
Domini	<u>Integer</u>	<u>serial</u>	<u>serial</u>	<u>serial</u>				

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

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

unique(ordine, pl);Foreign key video references Video(id);Foreign key pl references Playlist(id);La relazione accorpa le relazioni che implementano le seguenti relationship: play-elen.....12 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

16 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 ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

17 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 ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

18 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 ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

19 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 ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

20 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 ennuola, di dominio):

La relazione accorpa 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 ennupe); (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.visualizzato-dopo-pubblicato]

Trigger: Insert Visualizza

```
Error = SELECT *
        FROM Video vid, Visualizza vis
        WHERE vis.video = vid.id
        AND vis.istante < vid.istPub;
```

[V.visualizzato-prima-di-censura]

Trigger: Insert Visualizza

```
Error = SELECT *
        FROM Visualizza vis, Censurato c
        WHERE vis.video = c.video
        AND c.istCens < vis.istante;
```

[V.visualizza-dopo-iscrizione]

Trigger: Insert Visualizza

```
ERROR = SELECT *
        FROM Visualizza vis, Utente u
        WHERE vis.utente = u.id
        AND vis.istante < u.iscrizione;
```

[V.pubblicazione-dopo-iscrizione]

Trigger: Insert Video

```
Error = SELECT *
        FROM Video v, Utente u
        WHERE v.creator = u.id
        AND v.istPub < u.iscrizione;
```

[V.censurabi-no-playlist]

Trigger: Insert VPL

```
Error: SELECT *
        FROM Censurato c, VPL v
        WHERE v.video = c.video;
```

[V.no-auto-risposte]

Trigger: Insert Video

```
Error: SELECT *
        FROM Video v1, Video v2
        WHERE v1.risposta = v2.id
        AND v1.creator = v2.creator;
```

[V.no-risposte-al-Futuro]

Trigger: Insert Video

```
Error = SELECT
        FROM Video v1, Video v2
        WHERE v1.risposta = v2.id
        AND v1.istPub < v2.istPub;
```

Risposta alla Domanda 7 (segue)

[V.valutazioni_a_se_stessi]

Trigger: Insert Valuta

```
Error = SELECT *  
        FROM Valuta val, Video vid, Utente u  
        WHERE val.video = vid.id AND val.utente = u.id  
        AND vid.creator = u.id;
```

[V.valuta_se_visualizzato]

Trigger: Insert Valuta

```
For all u:Utente {  
    Q = SELECT DISTINCT v.video FROM Visualizza WHERE v.utente = u;  
    P = SELECT v.video FROM Valuta WHERE v.utente = u;  
    IF (P ≠ Q) { ERRORE }  
}  
Commit
```

[V.commento_dopo_visualizzato]

Trigger: Insert Commento

```
Error = SELECT *  
        FROM Commento c, Visualizza v  
        WHERE c.vis = v.id  
        AND c.istComm < v.istante;
```

[V.crea_playlist_dopo_iscrizione]

Trigger: Insert Playlist

```
Error: SELECT *  
        FROM Playlist p, Utente u  
        WHERE p.creator = u.id  
        AND p.dataCreazione < u.iscrizione;
```

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

cerca_video(c: Stringa, T: Tag[1..*], v: Voto)

```
Q = SELECT v.id  
      FROM Video v, tag_video t  
      WHERE val_media(v) ≥ v  
      AND v.cat = c  
      AND t.video = v.id  
      AND t in T;
```

Result = Q;

val_media(id: serial)

```
Q: SELECT SUM(val.voto)/COUNT(val.voto)  
      FROM Video vid, Valuta val  
      WHERE vid.id = id  
      AND val.video = id
```

Result = Q

cerca_discussioni(c: Stringa)

```
SELECT *  
FROM (SELECT v1.id, count(v2.id) as risposte  
      FROM Video v1, Video v2  
      WHERE v1.risposta = v2 AND v1.cat = c  
      GROUP BY v1.id)  
WHERE risposte = MAX(risposte)
```

Risposta alla **Domanda 8** (segue)

Matricola:

Minute

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.]





Matricola:

Minute

[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.]

