



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:



Esame Es.20250717 – Prova scritta del 17 luglio 2025

Si vuole progettare *IttieseWeb*, un sistema informativo per la gestione di un ITS (Istituto Tecnologico Superiore), una scuola di eccellenza ad alta specializzazione tecnologica post diploma che permette di conseguire il titolo di *tecnico superiore*.

Il sistema deve permettere al personale amministrativo, ai docenti e agli studenti di gestire corsi, moduli, esami e carriere.

L'ITS offre diversi percorsi di formazione, della durata di due anni (suddivisi in un periodo di lezioni frontali e uno di tirocinio in azienda), chiamati corsi. Ogni corso è identificato dal proprio nome.

Ogni anno, per ogni corso può essere attivata un'edizione (ad es. "Sviluppo Web e Cloud 2025").

Di ogni edizione interessa conoscere l'anno, il numero di crediti che è necessario accumulare, superando esami, per poter accedere al tirocinio in azienda, e il numero di crediti necessari per poter accedere all'esame di stato finale.

Ogni studente (di cui interessa conoscere nome, cognome, data di nascita e codice fiscale) può essere iscritto a una sola edizione alla volta.

Gli insegnamenti erogati in un'edizione di un corso vengono chiamati moduli (ad es. "Basi di Dati 1", "Python 3", ecc.). Di ogni modulo interessa il nome e i docenti che lo insegnano (con nome, cognome, indirizzo email e codice fiscale).

I moduli di un'edizione sono raggruppati in blocchi. Tutti i moduli devono far parte di esattamente un blocco.

I progressi degli studenti nei vari moduli vengono valutati periodicamente per mezzo di esami scritti. Un esame è relativo a un blocco di moduli ed è formato da più prove (un qualsiasi numero, ma almeno una). Di ogni esame interessa conoscere un codice identificativo (una stringa) e il periodo di inizio e di fine.

Di ogni prova interessa conoscere il nome (unico rispetto all'esame relativo) e la data (che deve essere all'interno del periodo dell'esame). Quando uno studente supera l'esame di un blocco, ottiene dei crediti (un numero intero positivo che dipende dal blocco, non dall'esame), i quali serviranno per accedere al tirocinio e, in seguito, all'esame finale.

Ogni studente può sostenere esami di un certo blocco un qualsiasi numero di volte, ma solo fino al superamento (non è possibile "rifiutare" il superamento).

Quando uno studente partecipa ad una prova, ottiene un voto da 0 a 10. Un esame è considerato superato da uno studente quando quest'ultimo ha ottenuto un punteggio pari ad almeno 6 in tutte le prove dell'esame.

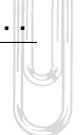
Per evitare che gli studenti seguano un percorso poco strutturato, sono previste delle propedeuticità tra blocchi di moduli. In altre parole, per poter sostenere l'esame di alcuni blocchi è necessario aver prima superato gli esami di altri blocchi. Di ogni blocco di moduli è di interesse quindi conoscere i blocchi i cui esami devono essere sostenuti prima di esso.

Una volta che uno studente ha accumulato un numero di crediti sufficienti (tale numero dipende dall'edizione, si veda sopra), può sostenere il tirocinio in azienda. Di ogni tirocinio interessa conoscere la data di inizio, quella di fine (alla sua conclusione) e l'azienda coinvolta (di cui sono di interesse la partita IVA e la ragione sociale, una stringa).

Quando lo studente ha terminato il tirocinio e accumulato un numero di crediti sufficienti per accedere all'esame di stato finale (anch'esso dipendente dall'edizione, si veda sopra) può accedere a quest'ultimo. L'esame di stato viene considerato superato se si riceve una votazione di almeno 6/10 e non può essere ripetuto.

IttesseWeb deve essere utilizzato dal personale amministrativo (che gestisce i corsi, le edizioni, i moduli, i gruppi e gli esami), dai docenti, che somministrano le prove d'esame, e dagli studenti, che devono poter visualizzare le informazioni riguardanti la propria carriera. Il sistema deve offrire le seguenti funzionalità avanzate:

- Dato un corso, calcolare l'edizione che ha avuto il numero minore di studenti che hanno raggiunto i requisiti per sostenere l'esame di stato finale. Uno studente non può accedere all'esame di stato se (1) non ha accumulato il numero totale di crediti necessari e (2) non ha terminato il tirocinio.



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.

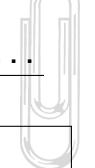


Questa pagina è stata intenzionalmente lasciata vuota

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



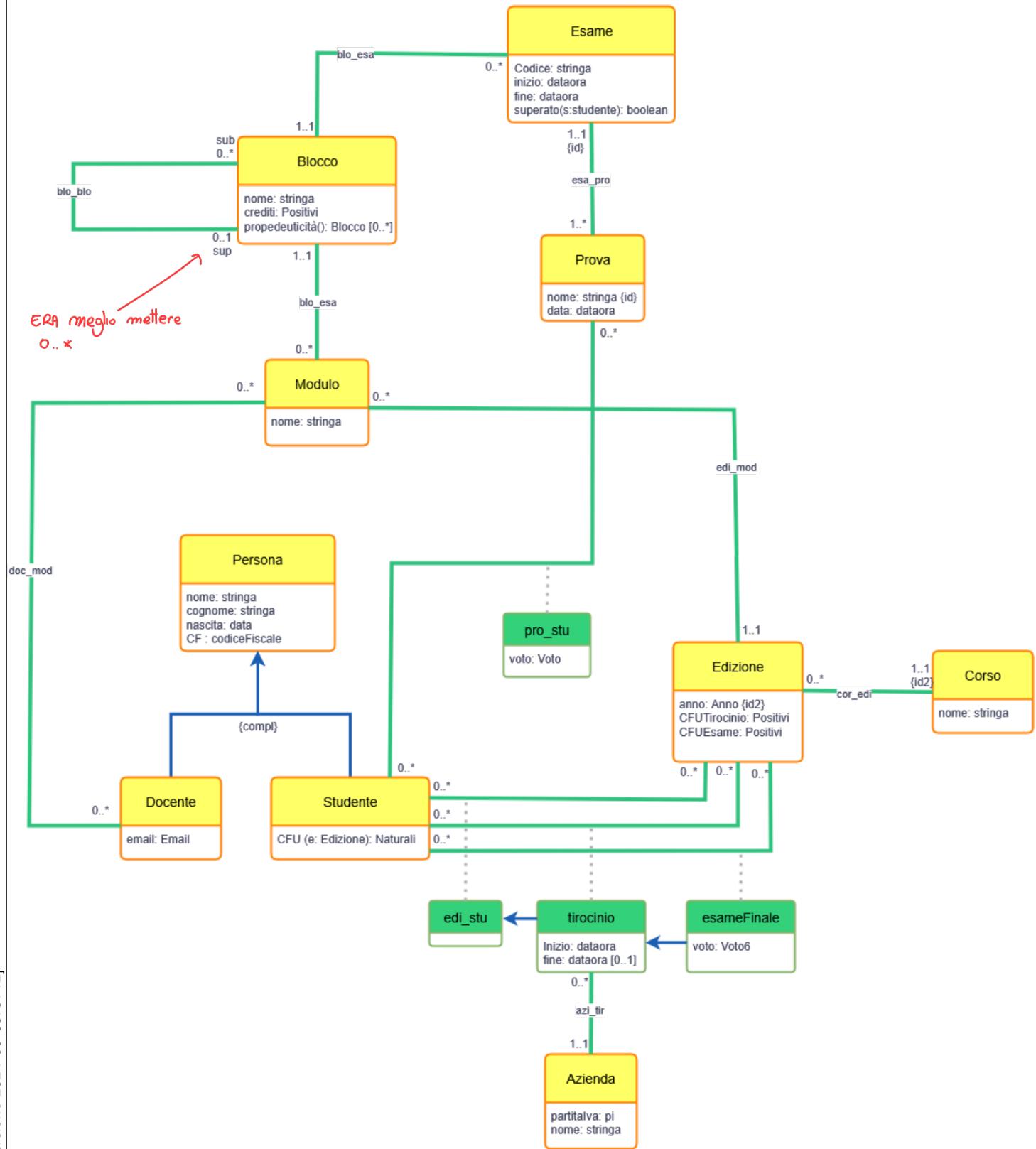
Risposta alla Domanda 1 (segue)

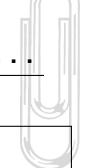
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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: ...STUDENTE.....</p> <p>Operazioni, vincoli:</p> <p>[V.U. EDIZIONE ALLO VOLTO]</p> $\forall s \text{ studente}(s) \rightarrow (\exists e, e', a, a' \quad \text{edizione}(e) \wedge \text{edizione}(e') \wedge e \neq e' \wedge \\ \text{Anno}(e, a) \wedge \text{Anno}(e', a') \wedge \neg(a + 2 \leq a' \vee a' + 2 \leq a))$ <p>[V. NO. ALTRE - PROVE - SE - SUPERATO]</p> $\forall s \text{ studente}(s) \wedge (\exists p, v, t \quad \text{prova}(p) \wedge \text{pro_stu}(r, s) \wedge \text{voto}(r, s, v) \\ \wedge v \geq 6 \wedge \text{DATA}(p, t) \longrightarrow (\exists p', t' \quad \text{Prova}(p') \wedge p' \neq p \wedge \text{DATA}(p', t')) \\ \wedge t' \geq t \wedge \text{pro_stu}(p', s))$	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...ESAME.....</p> <p>Operazioni, vincoli:</p> <p>[O. SUPERATO]</p> <p>Superato(s: utente): Booleano</p> <p>Pre: nessuna</p> <p>Post:</p> <p>Result = True \leftrightarrow $(\forall p, \text{prova}(p) \wedge \text{esa_pro}(this, p) \longrightarrow \\ \text{pro_stu}(p, s) \wedge (\exists v \quad \text{voto}(p, s, v) \wedge v \geq 6))$</p>
---	---

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Blocco</u>.....</p> <p>Operazioni, vincoli:</p> <p>Propedeuticità():</p> <p>Pre: Post: $\text{genitori} = \{ b \mid \text{blocco_blocc}(sop: b, sub: this) \}$</p> <p style="margin-left: 100px;"><i>genitori diretti</i></p> <p>$\text{antenati} = \{ b \mid \exists g \in \text{genitori} \wedge b \in \text{propedeuticità}(g) \}$</p> <p>Result = genitori \cup antenati</p> <p>[V. no. cicli]</p> <p>$\exists b \text{ blocco}(b) \wedge b \in \text{propedeuticità}(b)$</p>	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>STUDENTE</u>.....</p> <p>Operazioni, vincoli:</p> <p>[Tirocinio. solo . se . raggiunto . crediti.]</p> <p>$\forall s, e, ct, c$ $\text{studente}(s) \wedge \text{edizione}(e) \wedge \text{CFU}_\text{tirocinio}(e, ct) \wedge \text{edi_stu}(e, s) \wedge \text{CFU}(s, c) \wedge c < ct \longrightarrow \neg \text{tirocinio}(e, s)$</p> <p>[Op. CFU]</p> <p>CFU(e: Edizione): NATURALI</p> <p>Pre: Post: $\begin{cases} (v, cfv) \mid \text{esame}(v) \wedge (\exists b, m \text{ blocco}(b) \wedge \text{blo_eva}(b, v) \wedge \text{credit}(b, cfv) \wedge \text{blo_mod}(b, m) \wedge \text{edi_mod}(e, m)) \\ \wedge \text{superato}(v, this) \end{cases}$</p> <p>Result = $\sum_{(e, cfv) \in C} \text{CFU}$</p>
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Studente</u>.....</p> <p>Operazioni, vincoli:</p> <p>[esameFinale solo . se . raggiunto . crediti.]</p> <p>$\forall s, e, ct, c$ $\text{studente}(s) \wedge \text{edizione}(e) \wedge \text{CFU}_\text{esame}(e, ct) \wedge \text{edi_stu}(e, s) \wedge \text{CFU}(s, c) \wedge c < ct \longrightarrow \neg \text{EsameFinale}(s, e)$</p>	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>ESAME</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. esame . inizio . fine]</p> <p>$\forall i, f, e \text{ esame}(e) \wedge \text{inizio}(e, i) \wedge \text{fine}(e, f) \longrightarrow f > i$</p>
<p>5 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Prova</u>.....</p> <p>Operazioni, vincoli:</p> <p>[data . dentro . esame]</p> <p>$\forall p, t, e, i, f \text{ prova}(p) \wedge \text{data}(t) \wedge \text{esame}(e) \wedge \text{eva_pro}(e, p) \wedge \text{inizio}(e, i) \wedge \text{fine}(e, f) \longrightarrow t \geq i \wedge t \leq f$</p>	<p>8 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>TIROCINIO</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. inizio . fine]</p> <p>$\forall s, e, i \text{ tirocinio}(s, e) \wedge \text{inizio}(s, e, i) \wedge (\exists f \text{ fine}(s, e, f)) \longrightarrow f > i$</p> <p>[V. tirocinio . dentro . edizione]</p> <p>$\forall e, a \text{ edizione}(e) \wedge \text{Anno}(e, a) \longrightarrow (\forall s, i, f \text{ tirocinio}(s, e) \wedge \text{inizio}(s, e, i) \wedge \text{fine}(s, e, f) \wedge (a \leq i \wedge f \leq a+2))$</p>

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

Positivi = Intero > 0

Naturali = Intero ≥ 0

DataOra = (DATA, ora)

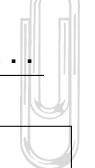
Voto = Intero 0..10

Voto6 = Intero 6..10

CodiceFiscale = Stringa secondo STANDARD

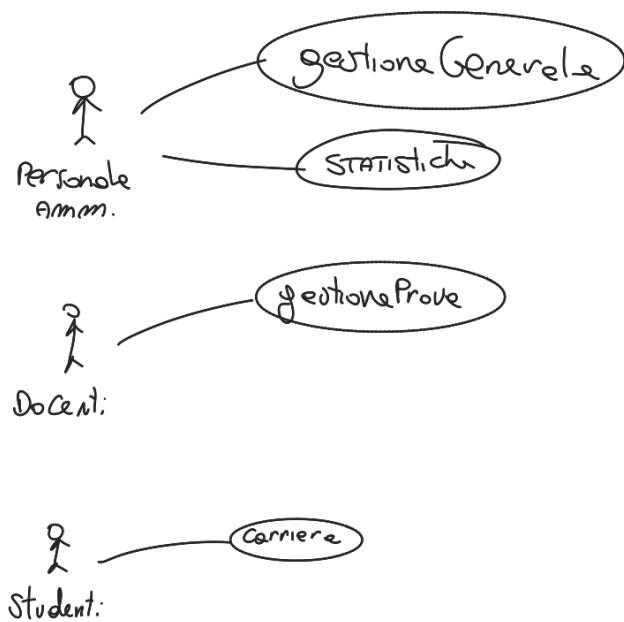
E-mail = Stringa secondo STANDARD

Pi = Stringa secondo STANDARD



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

- **Statistiche**

PeggiorEdizione (c: corso): NATURALI;

|
|
| Vanno scritti anche gli altri:



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

PeggiorEdizione(c: Corso): EDIZIONE

Pre: $\exists e \text{ edizione}(e) \text{ cor_edi}(c, e)$

Post:

$$E = \left\{ (e, n) \mid \text{edizione}(e) \wedge \text{cor_edi}(c, e) \wedge n = \text{Student:EsameFinale}(e: \text{edizione}) \right\}$$

$$(E_m, n_m) = \underset{(e, n) \in E}{\text{ArgMax}}(n)$$

Result = E_m

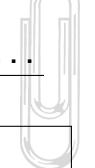
(Aux)

Student:EsameFinale(e: EDIZIONE): NATURALI;

Pre: nessuna

Post:

$$\text{Result} = \left\{ s \mid \begin{array}{l} \text{Studente}(s) \wedge (\exists n, c \quad \text{CFU}(s, e, n) \wedge \text{CFUEsame}(e, c) \wedge n \geq c) \\ \wedge \text{Tirocinio}(s, e) \wedge (\exists f \quad \text{FINE}(s, e, f)) \end{array} \right\}$$



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.

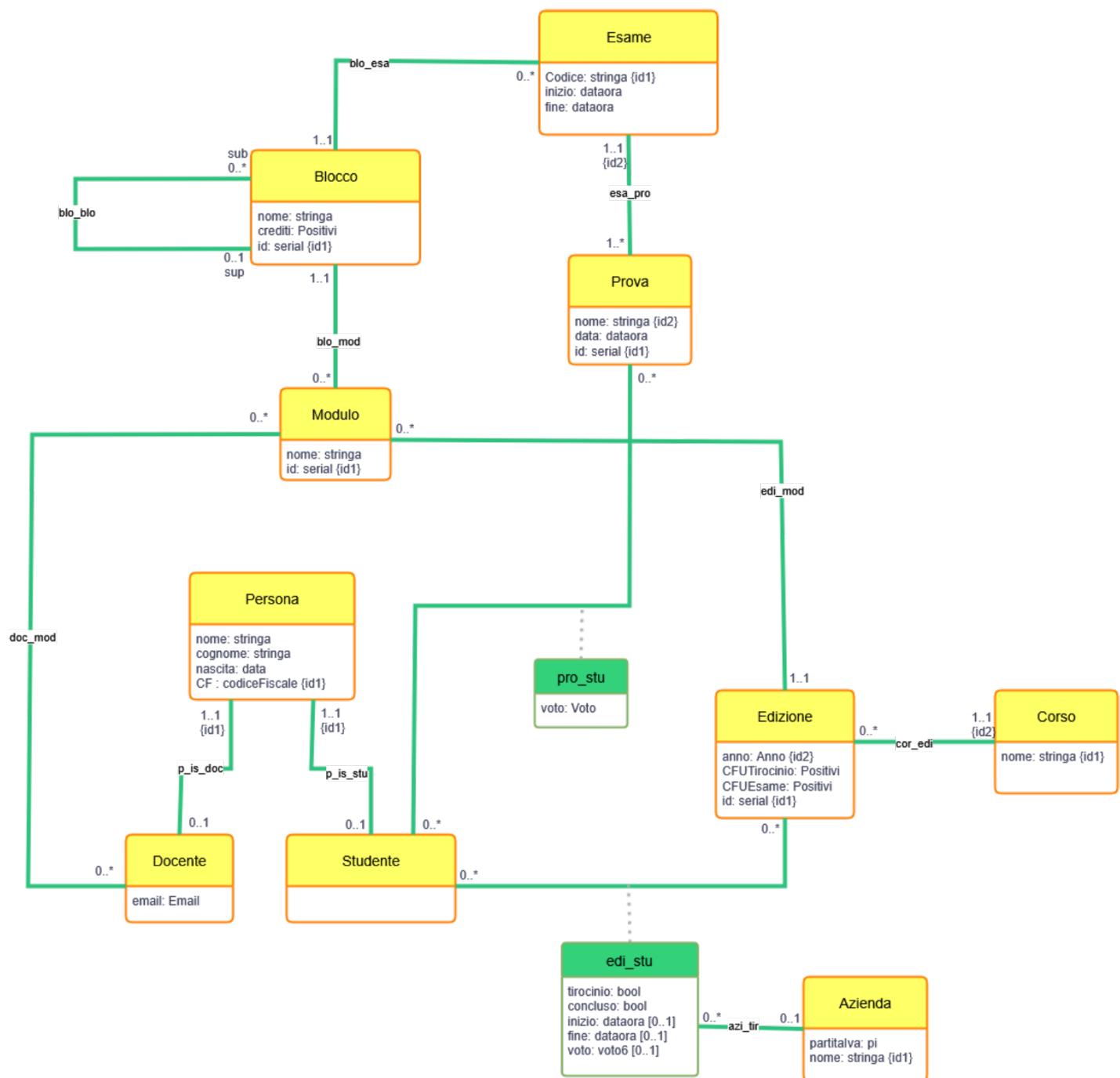
DBMS da utilizzare POSTGRESQL.....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```
CREATE DOMAIN Stringe AS TEXT
    " "          Positivi AS INT CHECK(value > 0)
    Naturali AS INT CHECK(value >= 0)
    CodiceFiscale AS TEXT CHECK(value ~ '...-...-...-...-...-...')
    Email AS TEXT CHECK(value ~ '...@...')

    DataOra AS TIMESTAMP
    Voto AS INT CHECK(value >= 0 AND value <= 10)
    Voto6 AS INT CHECK(value >= 6 AND value <= 10)
```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Rimossa is-a docente-studente per sostituzione

Rimossa is-a edi-stu per fusione

Rimossa operazioni di Classe dal diagramma UML_{ris}
definite le chiavi primarie

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)

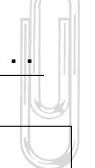
[V. is-a.doc-stu_complete]

$\forall p \text{ persona}(p) \rightarrow ((\exists d \text{ p-is-doc}(p, d)) \vee (\exists s \text{ p-is-stu}(p, s)))$

[V. is-a.edi-stu]

$\forall s, e \text{ edi-stu}(e, s) \wedge \text{Tirocinio}(e, s, \text{TRUE}) \rightarrow (\exists i, f, a \text{ INIZIO}(e, s, i) \wedge \text{fina}(e, s, f) \wedge \text{AZI-tir}(e, s, a))$

$\forall s, e \text{ edi-stu}(e, s) \wedge \text{concluso}(e, s, \text{TRUE}) \rightarrow (\text{Tirocinio}(e, s, \text{TRUE}) \wedge (\exists v \text{ voto}(e, s, v)))$



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 .PERSONA..... (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>Nome</u>	Gcognome	Nasce	<u>CF</u>				
Domini	Stringe	Stringe	date	CodiceFiscale				

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:

2 Relazione .DOCENTE..... (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>email</u>	<u>PERSONA</u>						
Domini	Email	CodiceFiscale						

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 Persone Ref Persone (CF);

La relazione accorda le relazioni che implementano le seguenti associazioni: P_U-doc.....

3 Relazione .Studente..... (nome)	Derivante da: classe associazione (cerchiare)
--	--

Attributi	<u>Persone</u>							
Domini	CodiceFiscale							

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 Persone Ref Persone (CF);

La relazione accorda le relazioni che implementano le seguenti associazioni: ...P_U-STU.....

4 Relazione EDI-STU..... (nome)	Derivante da: classe associazione (cerchiare)
--	--

Attributi	<u>TIROCINIO</u>	concluso	<u>Studente</u>	<u>edizione</u>	<u>inizio</u> *	<u>fine</u> *	<u>voto</u> *	Azienda *
Domini	bool	bool	CodiceFiscale	int	dataore	dataore	voto	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): CHECK(inizio <= fine)

FK Studente Ref Studente(person) check((Tirocinio IS TRUE AND inizio IS NOT NULL AND fine IS NOT NULL AND Azienda IS NOT NULL)
FK Azienda Ref Azienda (name) OR (Tirocinio IS FALSE AND inizio IS NULL AND fine IS NULL AND Azienda IS NULL))
FK edizione Ref edizioni (id) check((concluso IS TRUE AND voto IS NOT NULL AND TIROCINIO IS TRUE) OR (concluso IS FALSE
AND VOTO IS NULL))

La relazione accorda le relazioni che implementano le seguenti associazioni: ...AZI-TIR.....

5 Relazione ..AZIENDA..... (nome)	Derivante da: classe associazione (cerchiare)
--	--

Attributi	<u>PartitaIva</u>	<u>Nome</u>						
Domini	P;	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:

6 Relazione	<u>edizione</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>id</u>	Anno
Domini	serial	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):

UNIQUE(Anno, corso);

La relazione accorda le relazioni che implementano le seguenti associazioni: Cor-ed:

7 Relazione	<u>corso</u> (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:

8 Relazione	<u>Modulo</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>name</u>	<u>id</u>
Domini	Stringe	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):

FK edizione ref edizione (id)

PK blocco ref blocca (id)

La relazione accorda le relazioni che implementano le seguenti associazioni: .edi-mod, blo-mod

9 Relazione	<u>doc-mod</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>docente</u>	<u>modulo</u>
Domini	codicefiscale	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 docente ref docente (Personne)

Fk modulo ref modulo (id)

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione	<u>Blocco</u> (nome)	Derivante da: classe associazione (cerchiare)
Attributi	<u>name</u>	<u>crediti</u>
Domini	Stringe	positivi

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 sup ref blocca (id)

La relazione accorda le relazioni che implementano le seguenti associazioni: blo-bl

11 Relazione	<u>esame</u>	(nome)	Derivante da:	classe	associazione (cerchiare)
Attributi	<u>codice</u>	<u>INIZIO</u>	<u>fnr</u>	<u>bloco</u>	
Domini	<u>Stringe</u>	<u>dataora</u>	<u>data ora</u>	<u>int</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 bloco ref bloca(id)

check(WIZIO < fnr)

codice = Prova(esame)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... blo - eva

12 Relazione	<u>Prova</u>	(nome)	Derivante da:	classe	associazione (cerchiare)
Attributi	<u>id</u>	<u>Nome</u>	<u>esame</u>	<u>date</u>	
Domini	<u>Serial</u>	<u>Stringe</u>	<u>Stringe</u>	<u>dataora</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 esame ref esame(codice)

Unique(nome, esam)

La relazione accorda le relazioni che implementano le seguenti associazioni: ... elo - Pro

13 Relazione	<u>pro_stu</u>	(nome)	Derivante da:	classe	associazione (cerchiare)
Attributi	<u>Studente</u>	<u>Prove</u>	<u>voto</u>		
Domini	<u>CodiceFiscale</u>	<u>int</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 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 ennupple); (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.

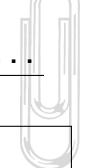
[T. 11.a. complete_persona]

Before insert e update on PERSONA

```
IF NOT EXISTS (SELECT 1 FROM docente d, studente s  
WHERE spersona = new.cf OR d.personee = new.cf)
```

```
IF ISError THEN  
    RAISE EXCEPTION
```

```
ELSE  
    permit
```



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

[USE-CASE]

PeggiorEdizione(c: String): INT

```
ISERROR = NOT EXISTS ( Select 1 FROM edizione e WHERE e.corso = c )
IF ISERROR THEN
    RAISE exception
```

SELECT e.id

```
FROM edizione e JOIN edi-stu es ON es.edizione = e.id
WHERE e.corso = c AND CFU(es.studente, e.id) >= e.CFUEdema
GROUP BY e.id
ORDER BY COUNT(*) ASC
LIMIT 1
```

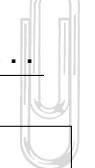
[Operazioni classi]

CFU(stud: codiceFiscale, eo: int): Naturali

```
Select SUM(b.crediti)
From esame es JOIN blocco b ON es.blocco = b.id
    JOIN modulo m ON m.blocco = b.id
Where m.edizione = eo AND Superato(es.codice, stud.)
```

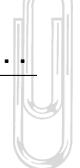
Superato(es: string, s: codiceFiscale): Bool

```
Select exists (Select 1
    From prova p, pro-stu ps
    Where p.esame = es AND
        ps.prova = p.id AND ps.studente = s
        AND p.voto >= 6 )
```



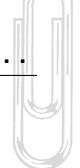
Risposta alla Domanda 8 (segue)

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.]