



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 20220909 (P.20220909)

My Precious

Versione 2024-02-08



Indice

Indice	1
1 Introduzione	2
2 Specifica dei Requisiti	3

1

Introduzione

Si vuole progettare e realizzare *My Precious*, un sistema di gestione di opere d'arte e delle relative esposizioni per conto di un museo, finalizzato all'utilizzo interno.

Tramite il sistema, si vuole che i responsabili del museo possano gestire il patrimonio artistico e organizzare delle esposizioni delle opere e che il management possa ottenere delle statistiche sulle visite.

2

Specifiche dei Requisiti

Gli utenti di *My Precious* sono i responsabili della gestione delle opere d'arte del museo e dell'organizzazione delle esposizioni.

È di interesse rappresentare le opere a disposizione del museo, con il nome, la categoria (p.es., dipinto, scultura, mosaico, manoscritto, ecc.), l'autore, l'anno di realizzazione, la tecnica (se rilevante) e le correnti artistiche alle quali è associata.

Di un'autore interessano il nome d'arte, il luogo e la data di nascita, la data di morte e le opere realizzate che sono di proprietà del museo.

Un'opera di proprietà del museo può essere presa in prestito da o data in prestito a un altro museo. Di un'opera in prestito interessa il museo di origine o destinazione, la data di inizio del prestito e quella di fine. Alcune opere, per via di un'importanza particolare o poiché troppo delicate, non possono essere date in prestito.

Alcune opere di proprietà del museo possono essere oggetto di restauri, nel qual caso interessa il periodo di restauro; un'opera in restauro non può essere esposta nel museo. Il sistema deve mantenere le informazioni riguardo a tutti i restauri ai quali è stata sottoposta ogni opera di proprietà del museo, da quando è di sua proprietà.

I responsabili della gestione del patrimonio del museo organizzano le esposizioni delle opere, le quali possono essere permanenti oppure temporanee.

Le esposizioni permanenti espongono le opere di proprietà del museo e le raggruppano, ad esempio, per periodo storico.

Le esposizioni temporanee, invece, sono esposizioni speciali che si tengono per periodi di tempo più o meno brevi (in genere qualche mese) e hanno un tema particolare, come un artista, un gruppo di artisti, una corrente artistica o un periodo storico.

Di un'esposizione è di interesse conoscere la sezione del museo nella quale ha luogo e le opere coinvolte.

Di un'esposizione temporanea interessa il nome e il periodo di tempo nel quale si tiene.

Il museo vende diversi tipi di biglietto. Ogni biglietto può essere utilizzato solamente nella data scelta al momento del suo acquisto.

Le tipologie di biglietto in vendita sono le seguenti:

30

1. il biglietto *full access*, che dà accesso a tutte le esposizioni, permanenti e temporanee;
2. un biglietto valido per tutte e sole le esposizioni permanenti;
3. un biglietto valido per tutte le esposizioni permanenti ed una sola esposizione temporanea, quest'ultima a scelta del visitatore.

35

Il sistema deve offrire le seguenti funzionalità ai responsabili del patrimonio artistico e al management:

1. inserimento e rimozione di opere d'arte;
2. registrazione del restauro di un'opera d'arte;
3. creazione di un'esposizione, con le opere relative;
4. calcolare il numero dei biglietti venduti in un dato giorno per ognuna delle diverse tipologie;
5. dato un periodo di tempo, il numero medio di biglietti (di qualsivoglia tipologia) venduti (media giornaliera) nel periodo dato;
6. dato un periodo di tempo, calcolare un resoconto che contenga, per ognuna delle correnti artistiche note, il numero di artisti che ha realizzato almeno un'opera di proprietà del museo appartenente a quella corrente, nel periodo dato;
7. dato un periodo di tempo, calcolare l'esposizione temporanea per la quale sono stati venduti più biglietti nel periodo (considerando tutti i tipi di biglietto rilevanti);
8. dato un artista e un periodo di tempo, calcolare le opere realizzate dall'artista ed in possesso del museo che non sono state oggetto di restauri nel periodo dato;
9. trovare le opere attualmente in restauro che non sono esposte da più tempo, insieme alla data di loro ultima esposizione.

45

50



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: SIMONE LIDONNICO Matricola: 2061343.....

Data: 10/06/2024.....

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)

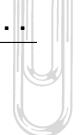
Firma di un membro della Commissione per
avvenuta identificazione:
.....

Rinuncia alla prova

Desidero rinunciare a questa prova d'esame. Firma:



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 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

Opera:

- nomeOp : Stringa
- annoOp : Intero
- associato all'Autore
- associato alla CATEGORIA
- associato alla TECNICA [0..1]
- associato alle Curanti

Prestito:

- initOpPre : Databea
- firePre : Databea
- associato all'Opera
- associato al Museo
- generalizzazione:
 - PrestitoDato
 - PrestitoPerson

Esposizione:

- initExp : Databea
- versione : Stringa
- associato alle Opere
- generalizzazione:
 - ExpPerm:
 - ExpTemp:
 - fireExp : Databea
 - nomeExp : Stringa
 - associato a un Tema

Autore:

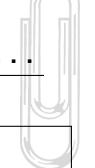
- nomeAut : Stringa
- annoNasc : Intero
- annoNas : Intero [0..1]
- associato alla Città
- associato alle Opere

Restauro:

- initRes : Databea
- fireRes : Databea [0..1]
- associato all'Opera

Biglietto:

- datoBigl : Datas
- generalizzazione:
 - BiglFull
 - BiglPerm
 - BiglTemp
 - associato all'ExpTemp



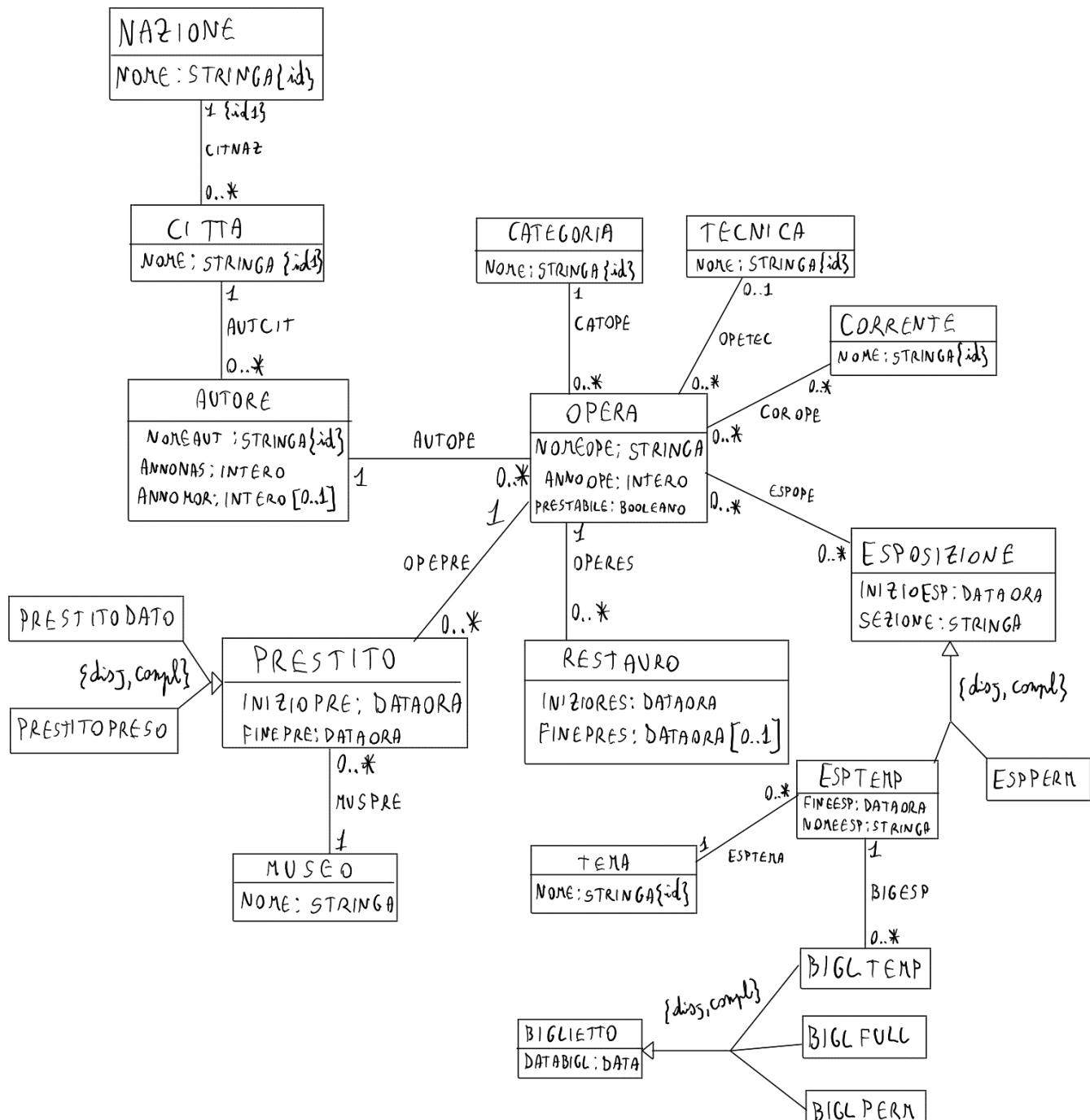
Risposta alla Domanda 1 (segue)

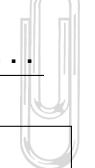
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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

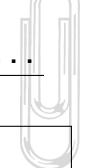
<p>1 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <u>AUTORE</u></p> <p>Operazioni, vincoli:</p> <p>\forall. NASCEPRIMA MUORE</p> <p>$\forall a, an, am \text{ AUTORE}(a) \wedge \text{ANNONASC}(an, a) \wedge \text{ANNOMOR}(a, am)$ $\rightarrow an < am$</p> <p>\forall. OPERAINVITA</p> <p>$\forall a, o, an, ar \text{ AUTOPRE}(a, o) \wedge \text{ANNOOPRE}(o, ar) \wedge \text{ANNONASC}(a, an) \rightarrow$ $an < ar \wedge (\forall am \text{ ANNOMOR}(a, am) \rightarrow ar < am)$</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <u>PRESTITO</u></p> <p>Operazioni, vincoli:</p> <p>\forall. FINE DOPONIZIO</p> <p>$\forall p, i, f \text{ PRESTITO}(p) \wedge \text{INIZIOPRE}(p, i) \wedge \text{FINEPRE}(p, f) \rightarrow i < f$</p> <p>$\forall$. DISGIUNTI</p> <p>$\forall o \text{ OPERA}(o) \rightarrow \exists p, p', i, i', f, f' \text{ id}$ $\text{OPEPRE}(p, o) \wedge \text{OPEPRE}(p', o) \wedge \text{INIZIOPRE}(p, i) \wedge \text{INIZIOPRE}(p', i')$ $\wedge \text{FINEPRE}(p, f) \wedge \text{FINEPRE}(p', f') \wedge \text{DATAORA}(d) \wedge d > i \wedge d > i' \wedge d < f \wedge d < f'$</p> <p>$\forall$. DOPOANNOOPERA</p> <p>$\forall p, i, o, a \text{ OPEPRE}(p, o) \wedge \text{INIZIOPRE}(p, i) \wedge \text{ANNOOPRE}(o, a) \rightarrow \exists d$ $\text{ANNO}(i, d) \wedge a < d$</p>
---	---

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... RESTAURO</p> <p>Operazioni, vincoli:</p> <p>V. FINE DOPOLAVORO $\forall m, n \text{ RESTAURO}(m) \wedge \text{INIZIORES}(m, n) \rightarrow$ $(\forall f \text{ FINERES}(m, f) \rightarrow f > n)$</p> <p>V. DISGIUNTI $\forall \sigma \text{ OPERA}(\sigma) \rightarrow \exists [m, m', i, i', d] \text{ OPERES}(\sigma, m) \wedge \text{OPERES}(\sigma, m') \wedge$ $\wedge \text{INIZIORES}(m, i) \wedge \text{INIZIORES}(m', i') \wedge \text{DATAORA}(d) \wedge d > i \wedge d > i' \wedge$ $\wedge (\forall f \text{ FINERES}(m, f) \rightarrow f > d) \wedge (\forall f' \text{ FINERES}(m', f') \rightarrow f' > d)$</p> <p>V. DOPO ANNOOPERA $\forall m, \sigma, i, a \text{ OPERES}(m, \sigma) \wedge \text{INIZIORES}(i, m) \wedge \text{ANNOOPERA}(\sigma, a) \rightarrow$ $\exists d \text{ ANNO}(i, d) \wedge d > a$</p>	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... ESPOSIZIONE</p> <p>Operazioni, vincoli:</p> <p>V. DOPO ANNOOPERA $\forall e, \sigma, i, a \text{ ESPOPE}(e, \sigma) \wedge \text{INIZIOESP}(i, e) \wedge \text{ANNOOPERA}(\sigma, a) \rightarrow$ $\exists d \text{ ANNO}(i, d) \wedge d > a$</p> <p>V. DISGIUNTE $\forall \sigma \text{ OPERA}(\sigma) \rightarrow \exists [e, e', i, i', d] \text{ ESPOPE}(e, \sigma) \wedge \text{ESPOPE}(e', \sigma) \wedge$ $\wedge \text{INIZIOESP}(e, i) \wedge \text{INIZIOESP}(e', i') \wedge \text{DATAORA}(d) \wedge d > i \wedge d > i' \wedge$ $\wedge (\forall f \text{ FINESP}(e, f) \rightarrow f < d) \wedge (\forall f' \text{ FINESP}(e', f') \rightarrow f' < d)$</p> <p>V. FINE DOPOLAVORO $\forall e, i, f \text{ ESPTEMP}(e) \wedge \text{INIZIOESP}(e, i) \wedge \text{FINEESP}(e, f) \rightarrow i < f$</p>
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... BIGLIETTO</p> <p>Operazioni, vincoli:</p> <p>V. DURANTEESPPOSIZIONE $\forall b, e, i, f, d \text{ BIGLIESP}(b, e) \wedge \text{INIZIOESP}(e, i) \wedge \text{FINEESP}(e, f)$ $\wedge \text{DATABIGL}(b, d) \rightarrow \exists di, df \text{ DATA}(i, di) \wedge \text{DATA}(f, df) \wedge$ $d >= di \wedge d <= df$</p>	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... OPERA</p> <p>Operazioni, vincoli:</p> <p>V. NO PRESTITO SE RESTAURO $\forall \sigma \text{ OPERA}(\sigma) \rightarrow \exists [p, n, ip, ir, fp, d] \text{ OPEPRE}(\sigma, p) \wedge$ $\text{OPERES}(\sigma, n) \wedge \text{INIZIOPRE}(p, ip) \wedge \text{INIZIORES}(n, ir) \wedge \text{FINEPRE}(ip, fp) \wedge$ $\text{DATAORA}(d) \wedge d > ir \wedge d > ip \wedge d < fp \wedge (\forall f \text{ FINERES}(n, f) \rightarrow f < fp)$</p> <p>V. NO ESPTEMP DURANTE RESTAURO $\forall \sigma \text{ OPERA}(\sigma) \rightarrow \exists [e, n, ie, ir, fe, d] \text{ ESPOPE}(\sigma, e) \wedge$ $\text{OPERES}(\sigma, n) \wedge \text{INIZIOESP}(e, ie) \wedge \text{INIZIORES}(n, ir) \wedge \text{FINEESP}(ie, fe) \wedge$ $\text{DATAORA}(d) \wedge d > ir \wedge d > ie \wedge d < fe \wedge (\forall f \text{ FINERES}(n, f) \rightarrow f < fe)$</p> <p>V. NO ESPTEMP DURANTE PRESTITO DATO $\forall \sigma \text{ OPERA}(\sigma) \rightarrow \exists [p, e, ip, il, fp, fe, d] \text{ OPEPRE}(\sigma, p) \wedge$ $\text{PRESTITODATO}(p) \wedge \text{INIZIOPRE}(ip, ip) \wedge \text{INIZIOESP}(il, e) \wedge \text{FINEPRE}(ip, fp) \wedge$ $\text{FINEESP}(fe, e) \wedge \text{DATAORA}(d) \wedge d > ip \wedge d > ie \wedge d < fe \wedge d < fp$</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

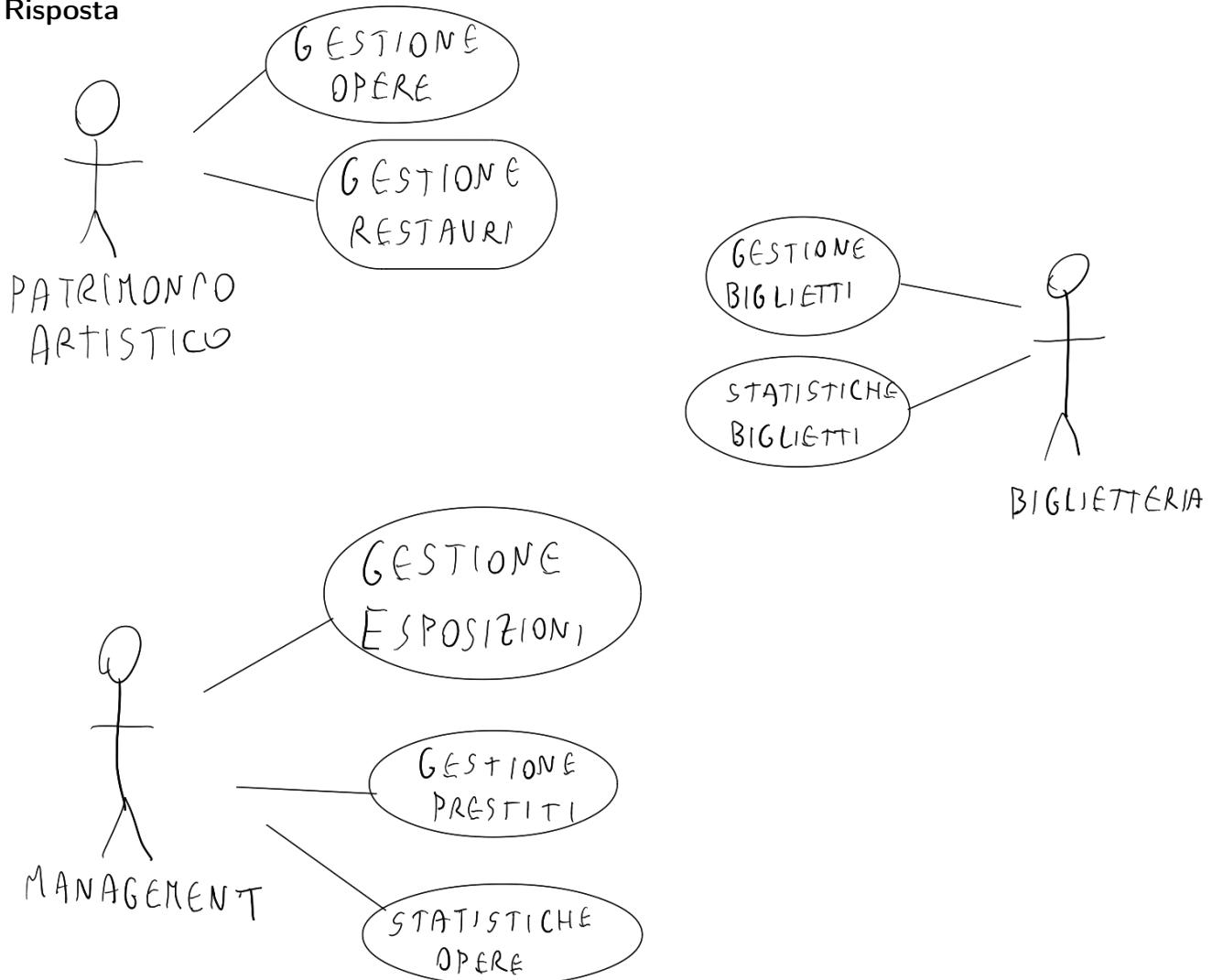
$\forall . \text{PRESTITO} . \text{SE PRESTABILE}$

$\forall \sigma, p \quad \text{OPEPRE}(\sigma, p) \rightarrow \text{PRESTABILE}(\sigma, \text{TRUE})$



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

GESTIONE OPERE:

NEWOPERA (n : STRINGA, a : INTERO, p : BOOLEANO, a : AUTORE): OPERA

RIMUOVOPERA (o : OPERA): OPERA

GESTIONE RESTAURI:

NEWRESTAURO (o : OPERA, i : DATAORA): RESTAURO

FINERESTAURO (r : RESTAURO, f : DATAORA): RESTAURO

GESTIONE PRESTITI:

NEWPRESTITO (o : OPERA, i : DATAORA, f : DATAORA, m : MUSEO): PRESTITO

GESTIONE ESPOSIZIONI:

NEWESPPERM (i : DATAORA, s : STRINGA): ESPPERM

NEWESPTEMP (n : STRINGA, i : DATAORA, f : DATAORA, s : STRINGA, t : TEMA): ESPTEMP

AGGIUNGIOPERA (e : ESPOSIZIONE, o : OPERA)

STATISTICHE OPERE:

RESOCONTOCORRENTI (i : INTERO, f : INTERO): (CORRENTE, INTERO ≥ 0) [0..*]

NORESTARTISTA (a : AUTORE, i : DATA, f : DATA): OPERA [0..*]

OPERERESTVECCHIE (): (OPERA, DATA) [0..*]

GESTIONE BIGLIETTI:

NEWBIGLFULL (d : DATA): BIGLFULL

NEWBIGLPERM (d : DATA): BIGLPERM

NEWBIGLTEMP (d : DATA, e : ESPTEMP): BIGLTEMP

STATISTICHE BIGLIETTI:

BIGLDATA (d : DATA): INTERO ≥ 0 [3]

BIGLMEDIA (i : DATA, f : DATA): REAL ≥ 0

ESPTEMPPOPOLARE (i : DATA, f : DATA): ESPTEMP [0..*]



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

$\text{BIGL DATA}(d; \text{DATA}) : |\text{INTERO}| = 0 [3]$

- pre:

$$\text{BF} = \left\{ b \mid \text{BIGL FULL}(b) \wedge \text{DATA BIGL}(b, d) \right\}$$

$$\text{BP} = \left\{ b \mid \text{BIGL PERM}(b) \wedge \text{DATA BIGL}(b, d) \right\}$$

$$\text{BT} = \left\{ b \mid \text{BIGL TEMP}(b) \wedge \text{DATA BIGL}(b, d) \right\}$$

$$\text{RESULT} = (|\text{BF}|, |\text{BP}|, |\text{BT}|)$$

$\text{BIGL MEDIA}(i; \text{DATA}, f; \text{DATA}) ; |\text{REALC}| = 0$

- pre: $i \leq f$

$$\text{B} = \left\{ b \mid \begin{array}{l} \text{BIGLIETTO}(b) \wedge \exists d \quad \text{DATA BIGL}(b, d) \wedge \\ d \geq i \wedge d \leq f \end{array} \right\}$$

SIA g TALE DA SODDISFARE:

$$\text{GIORNO}(f - i, g)$$

$$\text{RESULT} = \frac{|\text{B}|}{g}$$

$\text{ESPTEMP POPOLARE}(i; \text{DATA}, f; \text{DATA}) : \text{ESPTEMP}[0..*]$

- pre: $i \leq f$

- post:

$$\text{E} = \left\{ (e, n) \mid \begin{array}{l} \text{ESPTEMP}(e) \wedge \exists i_e, f_e, d_e, d_f \quad \text{INIZIOESP}(e, i_e) \wedge \text{FINEESP}(e, f_e) \\ \wedge \text{DATA}(i_e, d_e) \wedge \text{DATA}(f_e, d_f) \wedge \\ n = \left| \left\{ b \mid \begin{array}{l} [\text{BIGL FULL}(b) \vee \text{BIGL TEMP}(b) \wedge \text{BIGL ESP}(b, e)] \\ \wedge \exists d \quad \text{DATA BIGL}(b, d) \wedge d \geq i_e \wedge d \leq f_e \wedge d \leq d_f \end{array} \right\} \right| \end{array} \right\}$$

$$\text{RESULT} = \text{ARG MAX}(n) \\ (e, n) \in \text{E}$$

Risposta alla Domanda 5 (segue)

RESOCONTO CORRENTI (i : INTERO, f : INTERO): $(CORRENTE, INTERO \geq 0) [0..*]$

- pre: $i <= f$

- post:

$$\text{RESULT} = \left\{ (c, a) \middle| \begin{array}{l} \text{CORRENTE}(c) \wedge \\ a = \left\{ \begin{array}{l} \text{aut} \mid \text{AUTORE}(\text{aut}) \wedge \exists o, \text{an} \quad \text{OPERA}(o) \wedge \text{COPROPE}(c, o) \wedge \\ \text{AUTORE}(\text{aut}, o) \wedge \text{ANNOOPRE}(o, \text{an}) \wedge \text{an} \geq i \wedge \text{an} \leq f \end{array} \right\} \end{array} \right\}$$

NORESTARTISTA (σ : AUTORE, i : DATA, f : DATA): $\text{OPERA} [0..*]$

- pre: $i <= f$

- post:

$$\text{RESULT} = \left\{ \sigma \mid \begin{array}{l} \text{AUTORE}(\sigma, o) \wedge \exists m, im, d, di \quad \text{OPERES}(\sigma, m) \wedge \text{INIZIORES}(m, im) \wedge \\ \text{DATA}(d) \wedge \text{DATA}(im, di) \wedge d > i \wedge d > di \wedge d < f \wedge \\ (\forall fr, df \quad \text{FINERES}(fr, m) \wedge \text{DATA}(df, fr) \rightarrow d < df) \end{array} \right\}$$

OPERE REST VECCHIE(): $(\text{OPERA}, \text{DATA}) [0..*]$

- pre:

- post:

$$\text{OP} = \left\{ (\sigma, d) \mid \begin{array}{l} \exists r \quad \text{OPERES}(\sigma, m) \wedge \exists f \quad \text{FINERES}(m, f) \wedge \\ \exists im \quad \text{INIZIORES}(m, im) \wedge im < \text{ADESSO} \wedge \\ \exists e, i \quad \text{ESPPERM}(e, m) \wedge \text{ESPOPE}(e, \sigma) \wedge \text{INIZIOESP}(e, i) \\ \wedge i < im \wedge \text{DATA}(d, im) \end{array} \right\}$$

$$\text{OT} = \left\{ (\sigma, d) \mid \begin{array}{l} \exists r \quad \text{OPERES}(\sigma, m) \wedge \exists f \quad \text{FINERES}(m, f) \wedge \\ \exists im \quad \text{INIZIORES}(m, im) \wedge im < \text{ADESSO} \wedge \\ \exists e, fe \quad \text{ESPTEMP}(e) \wedge \text{ESPOPE}(e, \sigma) \wedge \text{FINEESP}(e, fe) \wedge \\ \exists e', ie' \quad \text{ESPOSIZIONE}(e') \wedge \text{INIZIOESP}(e, ie') \wedge ie' >= fe \wedge ie' < im \\ \wedge \text{ESPOPE}(e', \sigma) \wedge \text{DATA}(fe, d) \end{array} \right\}$$

$$O = OT \cup OP$$

$$\text{RESULT} = \text{ARGMIN}(d) \\ (\sigma, d) \in O$$

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.

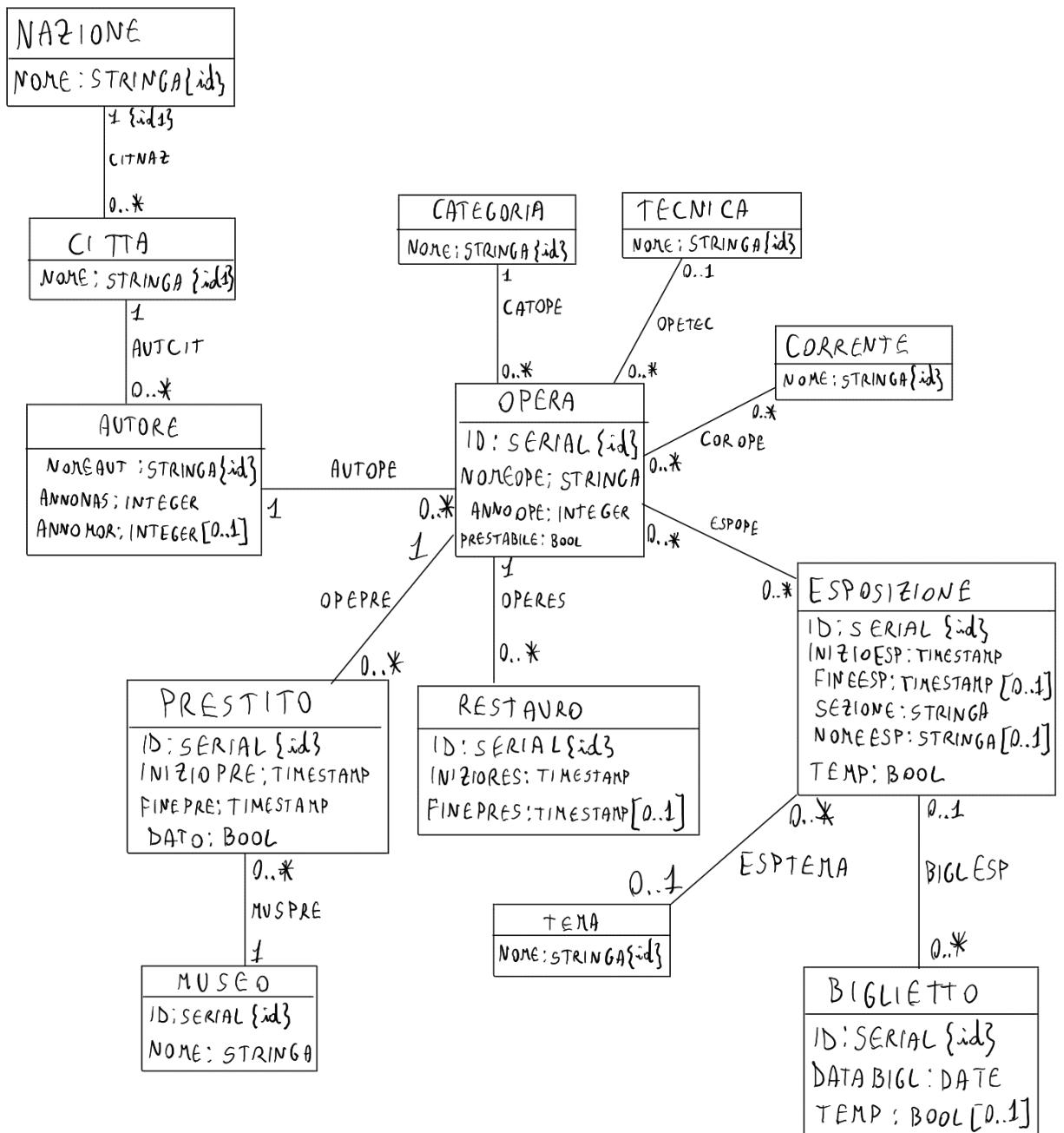
Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare POSTGRESQL.....

Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```
CREATE DOMAIN STRINGA AS VARCHAR NOT NULL;  
CREATE DOMAIN REALGEZ AS REAL CHECK(VALUE>=0);  
CREATE DOMAIN INTGEZ AS INTEGER CHECK(VALUE>=0);
```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

FUSIONE SU ESPOSIZIONE

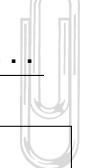
FUSIONE SU BIGLIETTO

FUSIONE SU PRESTITO

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)

 $\forall V, BIGLIETTO. ESP \text{ SOLOTEMP DURANTE}$ $\forall b, e \text{ BIGLESP}(b, e) \rightarrow TEMP(b, \text{TRUE}) \wedge TEMP(e, \text{TRUE}) \wedge$ $\exists i, f, d \text{ INIZIOESP}(i, e) \wedge FINEESP(f, e) \wedge DATABIGL(d, b) \wedge d >= i \wedge d <= f$ $\forall V, ESPOSIZIONE. TEMPINFO$ $\forall e \text{ ESPOSIZIONE}(e) \wedge TEMP(e, \text{TRUE}) \rightarrow \exists f, n \text{ NOMEESP}(e, n) \wedge FINEESP(e, f)$ $\forall V, ESPOSIZIONE. FINE DOPO INIZIO$ $\forall e, i, f \text{ ESPOSIZIONE}(e) \wedge INIZIOESP(e, i) \wedge FINEESP(e, f) \rightarrow i < f$



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

Attributi | ID | NOMEOPERA| ANNOOPERA| PRESTABILE | AUTORE | CAT | TEC *

Domini | SERIAL | STRINGA | INTEGER | BOOL | STRINGA | STRINGA | 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(AUTORE) REF AUTORE(NOMEAUT) FK(TEC) REF TECNICA(NOME)

FK(CAT) REF CATEGORIA(NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni: AUTOPERA, CATOPERA, OPERATEC.

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

Attributi | NOMEAUT | ANNONAS | ANNOMOR * | CITTA | NAZIONE | | |

Domini | STRINGA | INTEGER | INTEGER | STRINGA | 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(CITTA,NAZIONE) REF CITTA(NOME,NAZIONE) CHECK(ANNONASC ANNO MOR)

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

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

Attributi | NOME | NAZIONE | | | | | |

Domini | STRINGA | 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(NAZIONE) REF NAZIONE(NOME)

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

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

Attributi | NOME | | | | | | | |

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:

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

Attributi | NOME | | | | | | | |

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:

6 Relazione .T.E.C.N.ICA.... (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:

7 Relazione .C.O.R.R.E.N.T.E.. (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:

8 Relazione .C.O.R.O.P.E.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>OPERA</u> CORRENTE	
Domini INTEGER 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(OPER) REF OPERA(ID)

FK(CORRENTE) REF CORRENTE(NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

9 Relazione .R.E.S.T.A.V.R.O.. (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> INIZIRES FINERES OPERA	
Domini SERIAL TIMESTAMP TIMESTAMP INTEGER	

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(OPERA) REF OPERA(ID) CHECK(INIZIRES < FINERES)

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione .P.R.E.S.T.I.T.O.. (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> INIZIOPRE FINEPRE DATO OPERA MUSEO	
Domini SERIAL TIMESTAMP TIMESTAMP BOOL INTEGER INTEGER	

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(OPERA) REF OPERA(ID) CHECK(INIZIOPRE < FINEPRE)

FK(MUSEO) REF MUSEO(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione ...MUSEO..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> NOME	
Domini SERIAL 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:

12 Relazione ...ESPOSIZIONE. (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> INIZIOESP FINEESP * SEZIONE NOMEESP * TEMP TEMA	
Domini SERIAL TIMESTAMP TIMESTAMP STRINGA STRINGA BOOL 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(TEMA) REF TEMA(NOME) CHECK((TEMP IS NULL) = (FINEESP IS NULL)) AND

(CHECK(INIZIOESP < FINEESP) (TEMP IS NULL) = (NOMEESP IS NULL))

La relazione accorda le relazioni che implementano le seguenti associazioni: ...ESP.TEMA.....

13 Relazione ...ESP.OPER... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ESP</u> OPERA	
Domini INTEGER INTEGER	

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(ESP) REF ESPOSIZIONE(ID)

FK(OPERA) REF OPERA(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni:

14 Relazione .BIGLIETTO. (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> DATA TEMP * ESPTEMP *	
Domini SERIAL DATE BOOL INTEGER	

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(ESPTEMP) REF ESPOSIZIONE(ID) CHECK((TEMP IS NULL) = (ESPTEMP IS NULL))

La relazione accorda le relazioni che implementano le seguenti associazioni: ...BIGL.ESP.....

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. AUTORE. OPERA IN VITA

- INSERT OR UPDATE OPERA

ERROR = EXIST (SELECT * FROM AUTORE
 WHERE (ANNO_NAS > NEW_ANNO_OPE OR ANNO_MOR < NEW_ANNO_OPE)
 AND NOME = NEW_AUTORE)

IF ERROR; ERRORE

ELSE; COMMIT

T. PRESTITO. DISGIUNTI

- INSERT OR UPDATE PRESTITO

VALID = NOT EXIST (SELECT * FROM PRESTITO
 WHERE ID <> NEW_ID AND OPERA = NEW_OPERA AND
 (INIZIO_PRE, FINE_PRE) OVERLAPS (NEW_INIZIO_PRE, NEW_FINE_PRE))

IF VALID; COMMIT

ELSE; ERRORE

T. RESTAURO. DISGIUNTI

- INSERT OR UPDATE RESTAURO

VALID = NOT EXIST (SELECT * FROM RESTAURO
 WHERE ID <> NEW_ID AND OPERA = NEW_OPERA AND
 (INIZIO_RES, FINE_RES) OVERLAPS (NEW_INIZIO_RES, NEW_FINE_RES))

IF VALID; COMMIT

ELSE; ERRORE

Risposta alla Domanda 7 (segue)

T. ESPOSIZIONE, DOPO ANNO OPERA

- INSERT OR UPDATE ESP OPE

VALID = EXIST (SELECT * FROM ESPOSIZIONE o, OPERA o
WHERE o.ID = NEW.OPERID AND o.ID = NEW.ESP AND
EXTRACT(YEAR FROM e.INIZIOESP) >= o.ANNO OPE)

IF VALID: COMMIT

ELSE: ERRORE

T. BIGLIETTO, ESP SOLO TEMP DURANTE

- INSERT OR UPDATE BIGLIETTO

ERROR = EXIST (SELECT * FROM ESPOSIZIONE
WHERE NEW.ESP = ID AND (e.TEMP = FALSE OR
NEW.DATABIGL > e.FINEESP OR NEW.DATABIGL < e.INIZIOESP))

IF ERROR: ERRORE

ELSE: COMMIT

T. OPERA, NO PRESTITO DURANTE RESTAURO

- INSERT OR UPDATE PRESTITO OR RESTAURO

ERROR = EXIST (SELECT * FROM RESTAURO r, PRESTITO p
WHERE r.OPERA = p.OPERA AND p.OPERA = NEW.OPERA
AND (r.INIZIRES, r.FINERES) OVERLAPS (p.INIZIOPRE, p.FINEPRE))

IF ERROR: ERRORE

ELSE: COMMIT

CONTINUA A MINUTE 2,3

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

BIGL DATA(d : DATE): INTEGER [3]

$R = \text{WITH BF AS}(\text{SELECT * FROM BIGLIETTO WHERE TEMP=NULL}),$
 $\text{BP AS}(\text{SELECT * FROM BIGLIETTO WHERE TEMP=FALSE}),$
 $\text{BT AS}(\text{SELECT * FROM BIGLIETTO WHERE TEMP=TRUE})$

$\text{SELECT COUNT(DISTINCT BF.ID)}, \text{COUNT(DISTINCT BP.ID)}, \text{COUNT(DISTINCT BT.ID)}$
 FROM BF, BP, BT

RETURN R

BIGL MEDIA(i : DATE, f : DATE): REALGET

ERROR = $i > f$

$R = \text{SELECT COUNT(ID) / } i - f \text{ FROM BIGLIETTO}$
 $\text{WHERE DATABIGL} \geq i \text{ AND DATABIGL} \leq f$

RETURN R

ESP TEMP POPOLARE(i : DATE, f : DATE): INTEGER [0..*]

ERROR = $i > f$

$R = \text{WITH E AS}(\text{SELECT e.ID, COUNT(b.ID) C FROM ESPOSIZIONE e, BIGLIETTO b}$
 $\text{WHERE b.DATABIGL} \geq i \text{ AND b.DATABIGL} \leq f \text{ AND}$
 $b.DATABIGL} \geq e.\text{INIZIOESP} \text{ AND b.DATABIGL} \leq e.\text{FINEESP} \text{ AND}$
 $(b.TEMP IS NULL OR b.TEMP=TRUE AND b.ESP=e.ID)}$
 $\text{GROUP BY e.ID})$

SELECT ID FROM E

$\text{WHERE C} = (\text{SELECT MAX(C) FROM E})$

RETURN R

Risposta alla Domanda 8 (segue)

RESOCONTO CORRENTI (i ; INTEGER, f ; INTEGER): (STRINGA, INTEGER)[0..*]
 ERROR: $i > f$

$R = \text{SELECT CO.CORRENTE, COUNT(DISTINCT O.AUTORE) FROM COROPERA CO, OPERA O WHERE CO.OPERA = O.ID AND O.ANNOOPERA} \geq i \text{ AND O.ANNOOPERA} \leq f \text{ GROUP BY CO.CORRENTE}$
 $\text{RETURN } R$

NORESTARTISTA (α ; STRINGA, i ; DATE, f ; DATE); INTEGER[0..*]
 ERROR: $i > f$

$R = (\text{SELECT ID FROM OPERA WHERE AUTORE} = \alpha)$
 EXCEPT
 $(\text{SELECT O.ID FROM OPERA O, RESTAVROM M WHERE O.AUTORE} = \alpha \text{ AND M.OPERA} = O.ID \text{ AND}$
 $(i, f) \text{ OVERLAPS (EXTRACT(DATE FROM M.INIZIRES), EXTRACT(DATE FROM M.FINERES))})$
 $\text{RETURN } R$

CONTINUA MINUTE 1

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.]

OPEREREST VECCHIE(): (INTEGER, DATE)[0..*]

```
R := WITH O AS((SELECT o.ID, EXTRACT(DATE FROM M.INIZIORES) d
  FROM OPERA o, RESTAVRO R, ESPOPE eo, ESPOSIZIONE e
  WHERE M.OPERA=o.ID AND eo.OPERA=o.ID AND eo.ESP=e.ID AND
  M.INIZIORES<NOW() AND R.FINERES ISNULL AND e.INIZIOESP<M.INIZIORES AND
  e.TEMP=FALSE)
UNION
(SELECT o.ID, EXTRACT(DATE FROM e.FINEESP) d
  FROM OPERA o, RESTAVRO R, ESPOPE eo, ESPOSIZIONE e
  WHERE M.OPERA=o.ID AND eo.OPERA=o.ID AND eo.ESP=e.ID AND
  M.INIZIORES<NOW() AND R.FINERES ISNULL AND e.INIZIOESP<M.INIZIORES AND
  e.TEMP=TRUE AND
  e.FINEESP=(SELECT MAX(e.FINEESP) FROM ESPOSIZIONE e, ESPOPE eo
  WHERE eo.ESP=e.ID AND eo.OPERA=o.ID))
SELECT * FROM O
WHERE d=(SELECT MIN(d) FROM O)
```

RETURN R

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

T, OPERA, NOESPTEMP DURANTE RESTAURO

- INSERT OR UPDATE ESPOPE OR RESTAURO

ERROR = EXIST (SELECT * FROM ESPOPE e₀, PRESTITO p, ESPOSIZIONE e
WHERE e₀.OPERA = p.OPERA AND p.OPERA = NEW.OPERA AND
e.ID = e₀.ESP AND e.TEMP = TRUE AND
AND (e.INIZIOESP, e.FINEESP) OVERLAPS (p.INIZIOPRE, p.FINEPRE))

IF ERROR; ERRORE

ELSE; COMMIT

T, OPERA, NOESPTEMP DURANTE PRESTITO DATO

- INSERT OR UPDATE ESPOPE OR PRESTITO

ERROR = EXIST (SELECT * FROM ESPOPE e₀, PRESTITO p, ESPOSIZIONE e
WHERE e₀.OPERA = p.OPERA AND p.OPERA = NEW.OPERA AND
e.ID = e₀.ESP AND e.TEMP = TRUE AND p.DATO = TRUE
AND (e.INIZIOESP, e.FINEESP) OVERLAPS (p.INIZIOPRE, p.FINEPRE))

IF ERROR; ERRORE

ELSE; COMMIT

T, PRESTITO, SE PRESTABILE DOPO ANNOOPERA

- INSERT OR UPDATE PRESTITO

VALID = EXIST (SELECT * FROM OPERA
WHERE ID = NEW.OPERA AND PRESTABILE = TRUE
AND ANNOOPERA <= EXTRACT(YEAR FROM NEW.INIZIOPRE))

IF VALID; COMMIT

ELSE; ERRORE

T, RESTAURO, DOPO ANNOOPERA

- INSERT OR UPDATE PRESTITO

VALID = EXISTS (SELECT * FROM OPERA
WHERE ID = NEW.OPERA AND
ANNOOPERA <= EXTRACT(YEAR FROM NEW.INIZIOPRE))

IF VALID; COMMIT

ELSE; ERRORE

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

T. ESPOSIZIONE. DISGIUNTE

- INSERT OR UPDATE ESPOPE

VALID = NOT EXIST (WITH ESP AS (SELECT * FROM ESPOSIZIONE
WHERE ID = NEW.ESP)

SELECT * FROM ESPOPE e0, ESPOSIZIONE e1, ESP

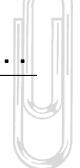
WHERE e0.ESP = e1.ID AND e1.ID <> NEW.ESP AND

e0.OPERA = NEW.OPERA AND

(e1.INIZIOESP, e1.FINEESP) OVERLAPS (ESP.INIZIOESP, ESP.FINEESP))

IF VALID, COMMIT

ELSE, ERRORE



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