

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:

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.

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

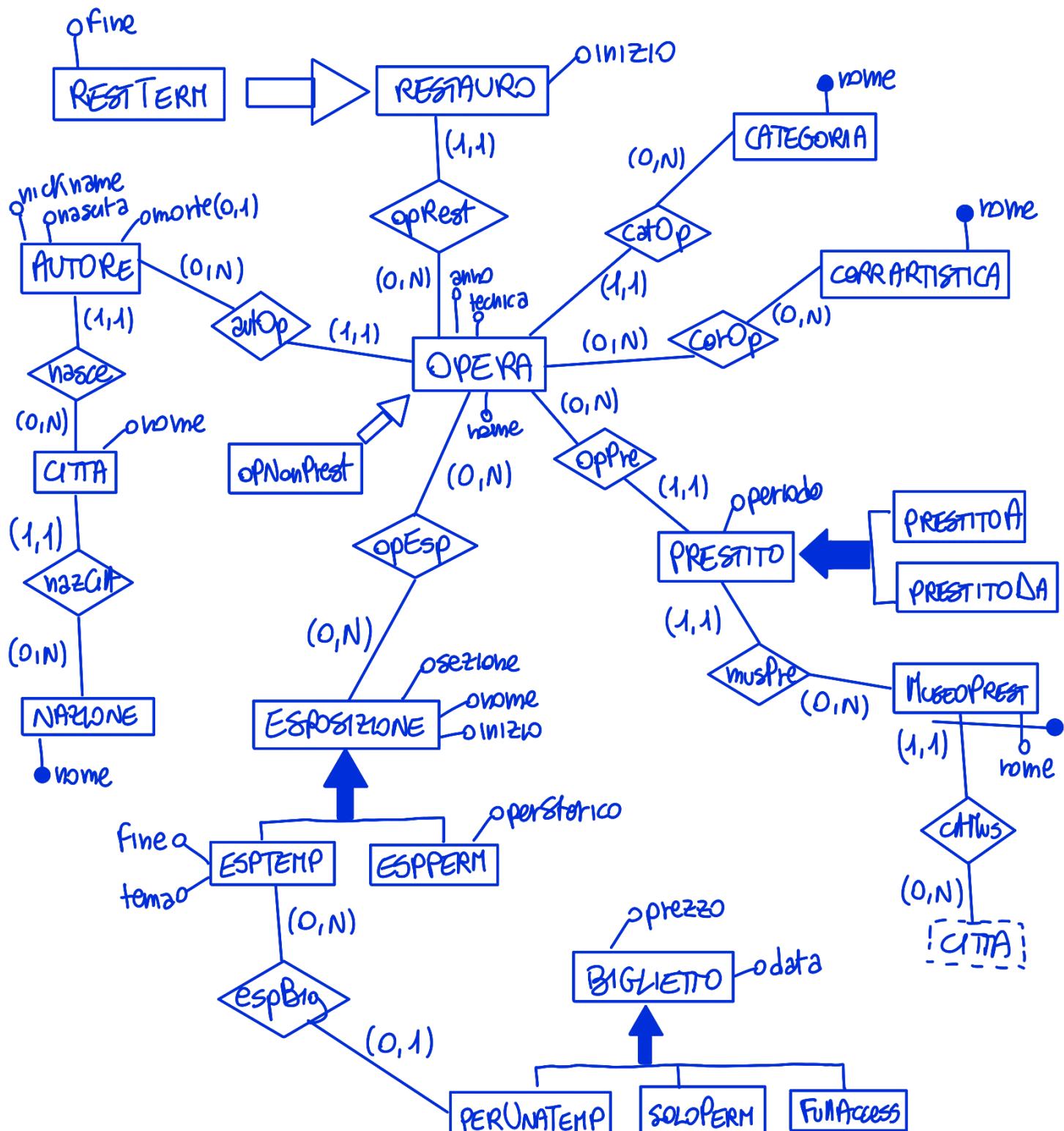
1. inserimento e rimozione di opere d'arte;
2. registrazione del restauro di un'opera d'arte;
3. creazione di un'esposizione, con le opere relative;
4. calcolare il numero dei biglietti venduti in un dato giorno per ognuna delle diverse tipologie;
5. dato un periodo di tempo, il numero medio di biglietti (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.

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

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

Diagramma ER

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



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

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

Risposta

<p>[1] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>OPERA</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td>tenuta</td> <td>stringa</td> <td></td> </tr> <tr> <td>anno</td> <td>data</td> <td></td> </tr> <tr> <td>nome</td> <td>stringa</td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p><u>$\forall .\ Opera .\ noPrestito$</u></p> <p><u>$\forall o,p \quad Opera(o) \wedge opPrest(o,p)$</u></p> <p><u>$\wedge prestoA(p) \rightarrow \neg opNonPrest(o)$</u></p>	attributo	dominio	moltepl. (*)	tenuta	stringa		anno	data		nome	stringa		<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>CATEGORIA</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltepl. (*)</th> </tr> </thead> <tbody> <tr> <td>nome</td> <td>stringa</td> <td></td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)	nome	stringa	
attributo	dominio	moltepl. (*)																	
tenuta	stringa																		
anno	data																		
nome	stringa																		
attributo	dominio	moltepl. (*)																	
nome	stringa																		

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

Nome: CORRENTE ARTISTICA

attributo	dominio	moltep. (*)
nome	stringa	

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

Vincoli:

5 Tipo: **Entità** | Relationship (cerchiare)

Nome: PRESTITO

attributo	dominio	moltep. (*)
perPrestito	Periodo	

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

Vincoli:

4 Tipo: **Entità** | Relationship (cerchiare)

Nome: MUSEO ESTERNO

attributo	dominio	moltep. (*)
nome	stringa	

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

Vincoli:

6 Tipo: **Entità** | Relationship (cerchiare)

Nome: RESTAURO

attributo	dominio	moltep. (*)
inizio	date	

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

Vincoli:

7	Tipo: Entità Relationship (cerchiare)	
Nome:	REST TERM	
attributo	dominio	moltep. (*)
fine	dataora	
(*) solo se diversa da (1,1)		
Vincoli:		
$[\forall. RestTerm. \text{inizioFine}]$ $\forall r, rt, i, f$ $\text{RestTerm}(f, rt) \wedge \text{inizio}(i, r)$ $\wedge \text{fine}(f, r) \rightarrow i < f$		

9	Tipo: Entità Relationship (cerchiare)	
Nome:	AUTORE	
attributo	dominio	moltep. (*)
nickname	stringa	
nascita	data	
morte	data	(0,1)
(*) solo se diversa da (1,1)		
Vincoli:		
$[\forall. Autore. \text{Continuità}]$ $\forall a, n \text{ Autore}(a) \wedge \text{nascita}(n, a)$ $\wedge (\forall m \text{ morte}(m, a)) \rightarrow n < m$		

8	Tipo: Entità Relationship (cerchiare)	
Nome:	CITTÀ	
attributo	dominio	moltep. (*)
name	stringa	
(*) solo se diversa da (1,1)		
Vincoli:		

10	Tipo: Entità Relationship (cerchiare)	
Nome:	NAZIONE	
attributo	dominio	moltep. (*)
name	stringa	
(*) solo se diversa da (1,1)		
Vincoli:		

11 Tipo: Entità Relationship (cerchiare)		
Nome: ESPOSIZIONE		
attributo	dominio	moltep. (*)
inizio	dataora	
sezione	$int \geq 0$	
name	stringa	

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

Vincoli:

[V. Esposizione.continuitàOpera]

$$\forall o, e, a, i$$

$$\text{Opera}(o) \wedge \text{opEsp}(o, e) \wedge$$

$$\text{anno}(a, o) \wedge \text{inizio}(i, e)$$

$$\rightarrow a < i$$

13 Tipo: Entità Relationship (cerchiare)		
Nome: ESPTEMP		
attributo	dominio	moltep. (*)
fine	dataora	
tema	stringa	

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

Vincoli:

[V. EspPerm.Opera]

$$\forall e, p, d, a, o, a'$$

$$\text{EspPerm}(p) \wedge \text{opEsp}(o, p) \wedge$$

$$\text{perStorico}(\text{per}, p) \wedge \text{da}(\text{per}, d) \wedge$$

$$a(\text{per}, a) \wedge \text{anno}(o, a')$$

$$\rightarrow d \leq a' \wedge a' \leq a$$

12 Tipo: Entità Relationship (cerchiare)		
Nome: ESPPERM		
attributo	dominio	moltep. (*)
perStorico	Periodo	

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

Vincoli:

14 Tipo: Entità Relationship (cerchiare)		
Nome: BIGLIETTO		
attributo	dominio	moltep. (*)
prezzo	Denaro	
data	data	

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

Vincoli:

[V. Biglietto.valido]

$$\forall b, db, e, i, di, df, f$$

$$\text{permUnaTemp}(b) \wedge \text{data}(db, b) \wedge$$

$$\text{espBig}(e, b) \wedge \text{inizio}(e, i) \wedge$$

$$\text{fine}(f, e) \rightarrow (di \leq d \wedge d \leq df) \wedge \text{EspTemp}(e)$$

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

Dominio Periodo

da: dataora

a: dataora

[V. Periodo. daa]

$$\forall p, x, y \text{ Periodo}(p) \wedge \text{da}(p, x) \wedge a(p, y) \rightarrow x < y$$

Dominio Denaro:

-valuta: stringa di 3 char

-importo: reale ≥ 0

[V. Esposizione. inizFine]

$$\forall e, i, et, f \quad \text{Esposizione}(e) \wedge \text{iniz}(i, e) \wedge \text{EspTerm}(e) \\ \wedge \text{fine}(et, f) \rightarrow i < f$$

[V. Autore. Continuità Opera] T

$$\forall a, o, n, d \quad \text{Autore}(a) \wedge \text{autOp}(a, o) \wedge \text{nascita}(a, n) \wedge \\ \text{data}(d, o) \rightarrow n < d \wedge (\forall m \text{ morte}(d, a) \rightarrow m \geq d)$$

[V. Esposizione. disgiunte]

$$\forall o, e, e', i, i' \quad \text{Opera}(o) \wedge \text{opEsp}(o, e) \wedge \text{opEsp}(o, e') \wedge e \neq e' \\ \wedge \text{iniz}(e, i) \wedge \text{iniz}(e, i') \rightarrow \\ \nexists t \text{ dataora}(t) \wedge (i \leq t \wedge (\forall f \text{ fine}(f, e) \rightarrow t \leq f)) \\ \wedge (i' \leq t \wedge (\forall f' \text{ fine}(f, e) \rightarrow t \leq f'))$$

Risposta alla Domanda 2 (segue)

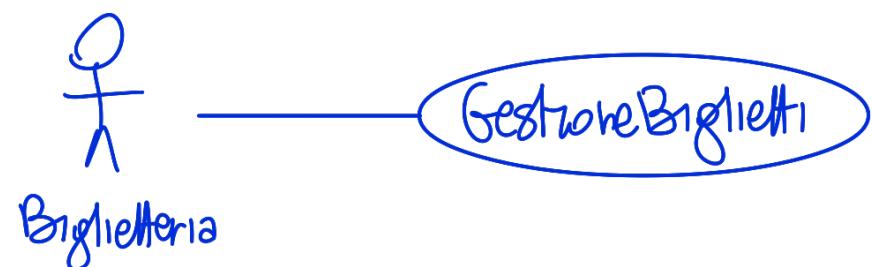
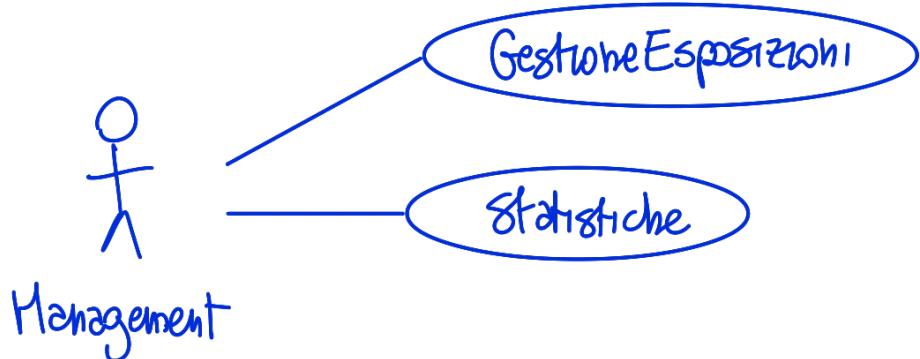
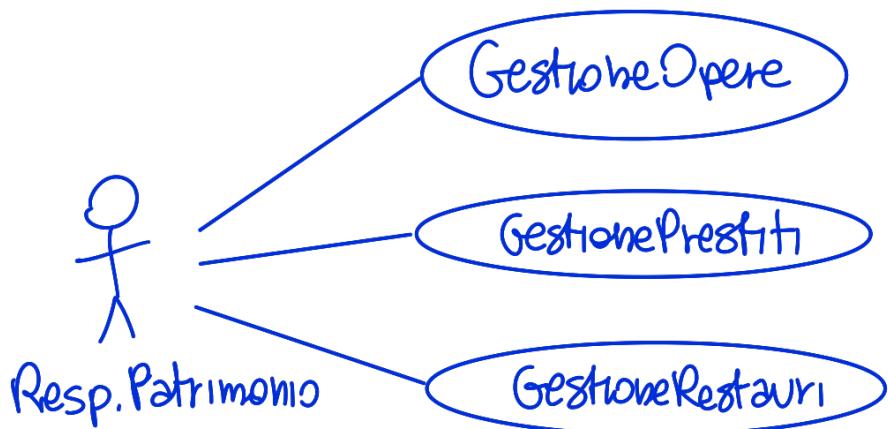
[N. Restauro. continuità Opera] T

$\forall o, r, a, i \quad \text{Opera}(o) \wedge \text{opRest}(o, r) \wedge \text{inizio}(r, i) \wedge \text{anno}(a, o) \rightarrow a < i$

[N. Restauro. noEsposizione] T

$\forall o, r, e, i, i' \quad \text{Opera}(o) \wedge \text{opRest}(r, o) \wedge \text{opEsp}(o, e) \wedge \text{inizio}(e, i) \wedge \text{inizio}(r, i') \rightarrow [\exists t \quad \text{dataora}(t) \wedge (ist \wedge (\forall f \text{ fine}(f, e) \rightarrow t \leq f)) \wedge (ist \wedge (\forall f' \text{ fine}(f', r) \rightarrow t \leq f'))]$

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: GestioneOpere (nome use-case)

Operazioni dello use-case:

inserimentoOpera(no: stringa, a': data, t: stringa, a: Autore,
c: CATEGORIA, c': CorrenteArtistica): Opera

rimozioneOpera(o: Opera)

2 Specifica use-case: GestioneRestauro (nome use-case)

Operazioni dello use-case:

registraRestauro(o: Opera, d: dataora): Restauro

registraTermineRestauro(r: Restauro)

elencoRestauri(o: Opera): Restauro(0, N)

3 Specifica use-case: GestioneEsposizioni (nome use-case)

Operazioni dello use-case:

registraEspPerm(no: stringa, i: dataora, s: stringa, pr: Periodo): EspPerm

registraEspTemp(no: stringa, i: dataora, s: stringa, f: dataora, t: stringa): EspTemp

aggiungiOpera(e: Esposizione, o: Opera)

rimuoviOpera(e: Esposizione, o: Opera)

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

Operazioni dello use-case:

numBiglietti(d: data) : (F: intero ≥ 0 , p: intero ≥ 0 , t: intero ≥ 0)

numMedioBiglietti(d: data, a: data) : reale ≥ 0

registraFullAccess(d: data, p: Denaro) : FullAccess

registraSoloPerm(d: data, p: Denaro) : SoloPerm

registraPermUnaTemp(d: data, p: Denaro, e: EspTemp) : PermUnaTemp

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

Operazioni dello use-case:

resoContoArtisti(d: data, a: data) : (c: ContArtisti, n: intero ≥ 0) (0, N)

EspTempPopolare(d: dataora, a: dataora) : EspTemp

OperaNonRestaurate(a: Artista, d: dataora, a: dataora) : Opera(0, N)

OperaInRestauroEspMenoRecente(d: dataora, a: dataora) : (o: Opera, i: dataora) (0, N)

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

Operazioni dello use-case:

registraPrestitoDa(o: Opera, m: MuseoEsterno, d: dataora, a: dataora) :
PrestitoDa

registraPrestitoA(o: Opera, m: MuseoEsterno, d: dataora, a: dataora) :
PrestitoA

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

Operazioni dello use-case:

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

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

Risposta

$\text{numBigVend}(d : \text{data}) : (F : \text{intero} \geq 0, p : \text{intero} \geq 0, t : \text{intero} \geq 0)$

precondizioni: nessuna

postcondizioni:

Modif. del liv. estens. dei dati: nessuna

Valore di ritorno: Siano:

$$FL = \{ b \mid \text{FullAccess}(b) \wedge \text{data}(d, b) \}$$

$$SP = \{ b \mid \text{SoloPerm}(b) \wedge \text{data}(d, b) \}$$

$$PUT = \{ b \mid \exists e \text{ permUnatemp}(b) \wedge \text{data}(d, b) \}$$

$$\text{result} = (|FL|, |SP|, |PUT|)$$

$\text{numMedioBig1}(da : \text{data}, a : \text{data}) : \text{reale} \geq 0$

precondizioni: $da < a$

postcondizioni:

Modifica del liv. estens. dei dati: nessuna.

Valore di ritorno:

$$B = \{ b \mid \text{BigLetti}(b) \wedge (\exists d \text{ data}(b, d) \wedge da \leq d \wedge d \leq a) \}$$

$$\text{result} = \frac{|B|}{a - da + 1}$$

Risposta alla Domanda 5 (segue)

`resoContoArtisti (da: data, a: data) : (c: CorrArtisti, n: intero ≥ 0) (O, N)`

precondizioni: $da < a$

postcondizioni:

Modifica del liv. estens. dei dati: nessuna

Valore di ritorno:

$$\text{result} = \left\{ (c, n) \mid \begin{array}{l} \text{CorrenteArtistica}(c) \wedge \\ n = \{x \mid \text{Autore}(x) \wedge (\exists o, a' \text{ Opera}(o) \\ \wedge \text{autOp}(a, o) \wedge \text{corOp}(c, o) \wedge \\ \text{anno}(o, a') \wedge da \leq a' \wedge a' \leq a \} \end{array} \right\}$$

`EspTempPopolare (d: dataora, a: dataora) : EspTemp`

precondizioni: $d < a$

postcondizioni:

Modifica del liv. estens. dei dati: nessuna

Valore di ritorno: Sono

$$E = \left\{ (e, n) \mid \begin{array}{l} \text{espTemp}(e) \wedge \text{data}(d) \wedge (\exists i, f, d_i, d_f, f' \\ \text{inizio}(e, i) \wedge \text{fine}(e, f) \wedge \text{data}(i, d_i) \wedge \\ \text{data}(f, d_f) \wedge d_i \leq d \wedge d \leq d_f) \wedge \\ n = \left| \begin{array}{l} b \mid \text{FullAccess}(b) \wedge (\text{PerUnaTemp}(b) \wedge \\ \text{espBig}(e, b)) \wedge (\exists d' \text{ data}(b, d')) \\ \wedge d \leq d' \wedge d' \leq a \end{array} \right| \end{array} \right\}$$

$(e_{\max}, n_{\max}) \in \text{ARGMAX}(n)$

$(e, n) \in E$

$\text{result} = e_{\max}$

Risposta alla Domanda 5 (segue)

$\text{OpereNonRestaurate } (a': \text{Artista}, d: \text{dataora}, a: \text{dataora}) : \text{Opera}(O, N)$

precondizioni: $d < a$

postcondizioni:

Modifica del l.v. est. dei dati: nessuna

Valore di ritorno:

$$\text{result} = \{ o \mid \text{Opera}(o) \wedge \text{autOp}(a', o) \wedge \\ (\exists r, i \text{ opRest}(o, r) \wedge \text{inizio}(r, i) \wedge d \leq i \\ \wedge i \leq a \}$$

$\text{OperelnRestauroEspMenoRecente } (d: \text{dataora}, a: \text{dataora}) : (o: \text{Opera}, i: \text{dataora})(O, N)$

precondizioni: $d < a$

postcondizioni:

Modifica del l.v. est. dei dati: nessuna

Valore di ritorno:

$$O = \{ (o, i) \mid \text{Opera}(o) \wedge (\exists r, e, i' \text{ Restauro}(e) \wedge \\ \neg \text{RestTerm}(r) \wedge \text{opRest}(o, r) \wedge \text{inizio}(r, i') \\ \wedge d \leq i' \wedge i' \leq a \wedge \text{opEsp}(e, o) \wedge \\ \text{inizio}(i, e) \wedge (\exists e', i'' \text{ opEsp}(e', o) \\ \wedge \text{inizio}(i'', e') \wedge i < i'') \}$$

$$\text{result} = \text{ARG-MIN}(i) \\ (o, i) \in O$$

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 ER concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

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

Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare **PostgreSQL**

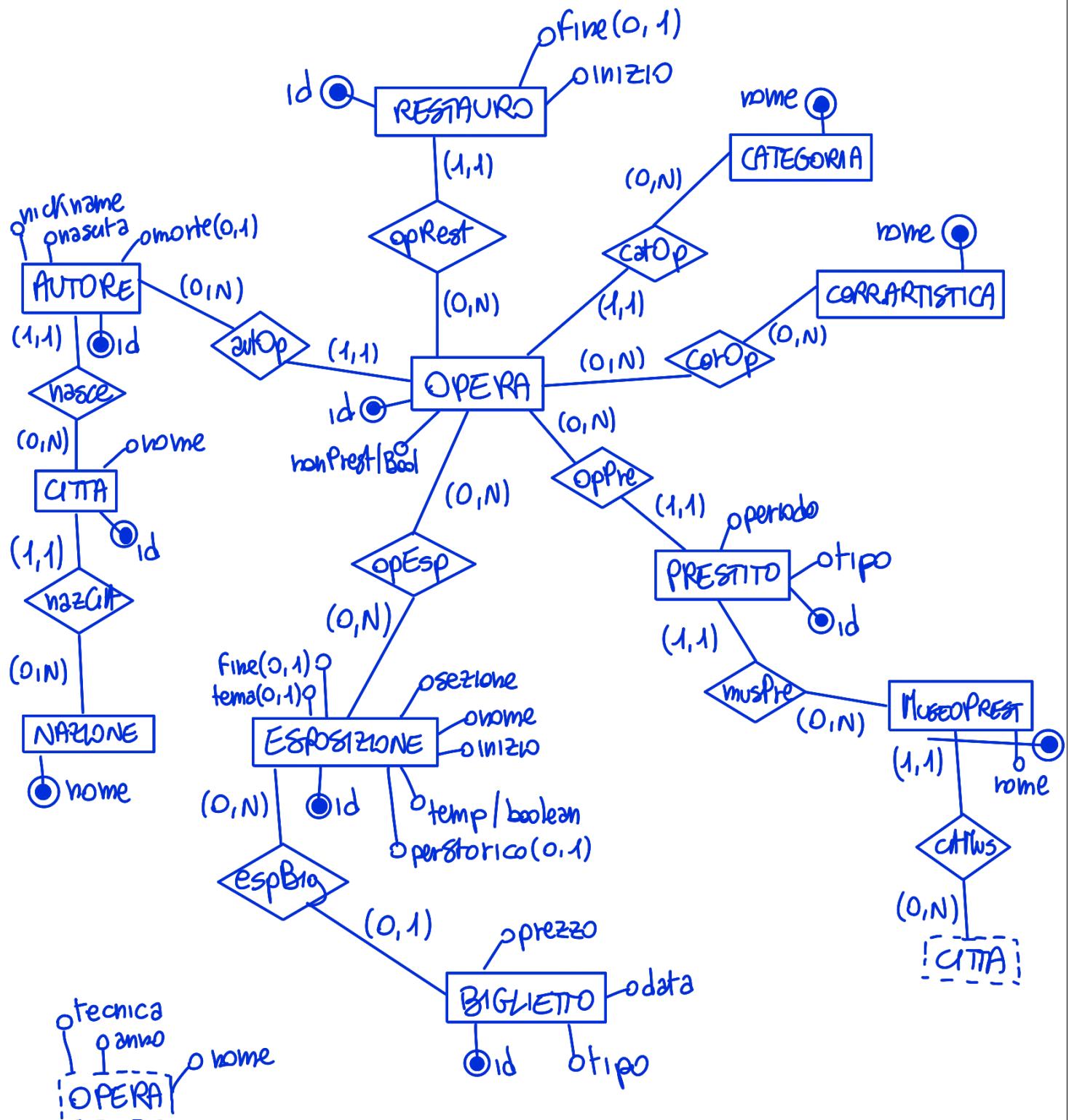
Corrispondenza tra domini concettuali e domini supportati dal DBMS

```

create domain StringS as varchar(50);
create domain StringM as varchar(200);
create domain StringL as varchar(600);
create type Periodo as(
    da: timestamp
    a: timestamp
);
create domain RealGEZ as real check(value >= 0)
create type Denaro as(
    Valuta: char(3)
    importo: RealGEZ );
create type Tipo_Bigli as enum ('PUT','SP','FA')
create type Tipo_Prest as enum('DA','A')
create domain IntegerGEZ as integer check(value >= 0)

```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

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. Esposizione.espTemp]

$\forall e \text{ Esposizione}(e) \rightarrow (\exists f, t \text{ fine}(e, t) \wedge \text{tema}(t, e) \leftrightarrow \text{temp}(e, \text{True}))$

[V. Biglietto.PermUnaTemp]

$\forall b \text{ Biglietto}(b) \rightarrow (\exists e \text{ espBig}(e, p) \leftrightarrow \text{tipo}(b, 'PUT'))$

[V. Prestito.disg.compl]

$\forall p \text{ Prestito} \rightarrow \text{tipo}(p, 'A') \vee \text{tipo}(p, 'DA')$

[V. Opera.noPrestito] T

$\forall o, p \text{ Opera}(o) \wedge \text{opPrest}(o, p) \wedge \text{tipo}(p, 'A') \rightarrow \text{nonPrest}(o, \text{false})$

Risposta alla Domanda 6 (segue)

[V. Esposizione.espPerm]

$\forall e \text{ Esposizione}(e) \rightarrow (\exists p \text{ persistente}(e,p) \leftrightarrow \text{temp}(e,\text{false}))$

[V. Biglietto.Valido] T

$\forall b,d,e,i,f \text{ Biglietto}(b) \wedge \text{tipo}(B, \text{'PUT'}) \wedge \text{espBig}(e,b)$
 $\wedge \text{data}(b,d) \wedge \text{inizio}(e,i) \wedge \text{fine}(e,f) \rightarrow (i \leq d \wedge d \leq f \wedge \text{temp}=(e,\text{true}))$

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	<u>CITTÀ</u> (nome)	Derivante da:	entità	relationship (cerchiare)				
Attributi	<u>id</u>	nome	nazione						
Domini	integer	stringM	stringM						

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

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

seriale: id

PK: nazione refer Nazione(name)

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

2	Relazione	<u>Nazione</u> (nome)	Derivante da:	entità	relationship (cerchiare)				
Attributi	<u> nome</u>								
Domini	stringM								

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

3	Relazione	<u>Autore</u> (nome)	Derivante da:	entità	relationship (cerchiare)				
Attributi	<u>id</u>	nickname	nascita	morte*	città				
Domini	integer	stringM	date	date	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):

seriale: id
ennupla: morte ≠ NULL → nascita < morte

PK: città references Città(id)

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

4	Relazione	<u>MuseoEsterno</u> (nome)	Derivante da:	entità	relationship (cerchiare)				
Attributi	<u> nome</u>	città							
Domini	stringM	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):

PK: città refer Città(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

5	Relazione	<u>Prestito</u> (nome)	Derivante da:	entità	relationship (cerchiare)				
Attributi	<u>id</u>	tipo	periodo	nomeMuseo	altMuseo	opera			
Domini	integer	Tipo_Prest	Periodo	stringM	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):

seriale: id
PK: opera refer Opera(id)

FK: (nomeMuseo,altMuseo) refer MuseoEsterno(nome,id)

La relazione accorda le relazioni che implementano le seguenti relationship: ...musPre, Opera

6 Relazione	Restauro (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	inizio	fine*	opera	
Domini	integer	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):

seriale: id
ennupla: fine ≠ NULL → inizio < fine
FK: opera refer Opera(id)
La relazione accorda le relazioni che implementano le seguenti relationship: ... opRest

7 Relazione	Biglietto (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	prezzo	data	tipo	esposizion*
Domini	integer	Denaro	date	Tipo_Bigl	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):

seriale: id ennupla: Esposizione ≠ NULL ↔ tipo = 'PUT'
FK: esposizione refer Esposizione(id) [ennupla: tipo = 'PUT' ∨ tipo = 'FA' ∨ tipo = 'SP']
La relazione accorda le relazioni che implementano le seguenti relationship: ... espBig

8 Relazione	Opera (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>id</u>	nonPrest	tecnica	anno	autore nome categoria
Domini	integer	boolean	StringM	date	integer StringM StringM

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

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): seriale: id
FK: autore refer Autore(id)
PK: categoria refer Categoria(StringM)
La relazione accorda le relazioni che implementano le seguenti relationship: ... autOp, catOp

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

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

10 Relazione	CARR_ARTISTICA (nome)	Derivante da:	entità	relationship (cerchiare)
Attributi	<u>name</u>				
Domini	StringM				

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

11 Relazione <u>Esposizione</u> (nome)	Derivante da: entità relationship (cerchiare)
--	---

Attributi	<u>id</u>	<u>temp</u>	<u>perstorico*</u>	<u>inizio</u>	<u>settore</u>	<u>name</u>	<u>fine*</u>	<u>tema*</u>
Domini	<u>integer</u>	<u>boolean</u>	<u>Periodo</u>	<u>timestamp</u>	<u>StringM</u>	<u>StringM</u>	<u>timestamp</u>	<u>StringL</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): settore: id

ennupla: perstorico ≠ NULL ↔ temp = False

ennupla: fine ≠ NULL ∧ tema ≠ NULL ↔ temp = TRUE

La relazione accorda le relazioni che implementano le seguenti relationship:

12 Relazione <u>corrOp</u> (nome)	Derivante da: entità relationship (cerchiare)
-----------------------------------	---

Attributi	<u>opera</u>	<u>corrArtis</u>						
Domini	<u>integer</u>	<u>StringM</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: opera refer Opera(id)

PK: corrArtis refer CorrArtistica(name)

La relazione accorda le relazioni che implementano le seguenti relationship:

13 Relazione <u>opEsp</u> (nome)	Derivante da: entità relationship (cerchiare)
----------------------------------	---

Attributi	<u>opera</u>	<u>esposizione</u>						
Domini	<u>integer</u>	<u>integer</u>						

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

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

PK: opera refer Opera(id)

PK: esposizione refer Esposizione(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

14 Relazione (nome)	Derivante da: entità relationship (cerchiare)
---------------------------	---

Attributi								
Domini								

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

15 Relazione (nome)	Derivante da: entità relationship (cerchiare)
---------------------------	---

Attributi								
Domini								

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennupla); (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.

- rendere update on Restauro from User
- grant update(fine) on Restauro to User

[T. Restauro. Disgunti]

- inserimento o modifica in Restauro
- pre-operazione

```
isError = (exists (select *
                   from Restauro r
                  where r.opera = new.opera and
                        ((r.fine is not null and new.fine is not null)
                         and (r.inizio,r.fine)overlaps(new.inizio,new.fine))
                     or (r.fine is not null and new.fine is not null)
                         and new.inizio <= r.fine) and (r.fine is null
                         and new.fine is not null and r.inizio <= new.fine)))
```

```
if isError: genera errore
else commit
```

Risposta alla Domanda 7 (segue)

- rende update on Opera from User
- rende update on Prestito from User

[T. Opera, noPrestito]

- inserimento in prestito
- pre-operazione

```
isError=(exists (select *
                  from Opera o
                 where newr.opera=o.id and newr.tipo='A'
                   and o.noPrest=True))
```

if isError: genera errore
else commit

- grant update (fine) on Esposizione to User

[T. Biglietto, Valido]

- inserimento in biglietto
- pre-operazione

```
isValid=(exists (select *
                  from Esposizione
                 where newr.tipo='PUT' and newr.esp=e.id
                   and e.valid='TRUE' and e.inizio < newr.data
                     and newr.data <= e.fine))
```

if isError: genera errore
else commit

Risposta alla Domanda 7 (segue)

- renvoie update on Opera

[T. Restauro. ContinuitàOpera]

- inserimento in Restauro

- pre-operazione:

isValid : (exists (select *
From Opera o
where new.opera = o.id
and o.anno < new.inizio))

if isError: genera errore
else commit

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

$\text{numBigliettiVenduti}(\text{d}: \text{date}) : (\text{f: IntegerGEZ}, \text{p: IntegerGEZ}, \text{t: IntegerGEZ})$

$Q \leftarrow$ risultato della query

```
select sum(case when tipo = 'FA' then 1 else 0 end) as f,
       sum(case when tipo = 'SP' then 1 else 0 end) as p,
       sum(case when tipo = 'PUT' then 1 else 0 end) as t
```

from Biglietto b

where b.data = :d

return Q

$\text{resoContoArtisti}(\text{d}: \text{date}, \text{a}: \text{date}) : \text{Insieme}(\text{c: StringM}, \text{m: IntegerGEZ})$

if :d >= :a : genera errore

1 $Q \leftarrow$ risultato della query

```
select co.corrente, count(a.id)
```

from corOp co, Autore a, Opera o

where o.autore = a.id and co.opera = o.id

and :d <= o.anno and o.anno <= :a

return Q

Risposta alla Domanda 8 (segue)

$\text{numMedioBaglietto}(\text{d} : \text{date}, \text{a} : \text{date}) : \text{RealGEZ}$

if: $:d \geq :a$: genera errore

1 $Q \leftarrow$ risultato della query

select $(\text{count}(\text{b.id}) / (:a + :d - 1))$

from Baglietto b

where $:d \leq \text{b.data}$ and $\text{b.data} \leq :a$

return Q

$\text{espTempPopolare}(\text{d} : \text{timestamp}, \text{a} : \text{dataora}) : \text{integer}$

1 if $:d \geq :a$: genera errore

2 $\text{Tmp}(\text{esp} : \text{integer}, \text{num} : \text{integerGEZ})$

fk: esp references Esposizione(id)

3 insert into $\text{Tmp}(\text{esp}, \text{num})$

4 select $e.id, \text{count}(\text{b.id})$

from $\text{Esposizione e, Baglietto b}$

where $e.temp = e.id$ and $(:d \leq e.inizio \text{ and } e.fine} \leq :a)$

and $(\text{b.tipo} = 'FIA' \text{ or } (\text{b.tipo} = 'PUT' \text{ and } \text{b.esposizione} = e.id))$

and $:d \leq \text{b.data}$ and $\text{b.data} \leq :a$

group by $e.id$

5 $Q = (\text{select esp from Tmp}$

where $\text{num} = (\text{select max(num) from Tmp})$)

6 return Q

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

OpereNonRestaurate ($a^1:\text{integer}$, $d:\text{timestamp}$,
 $a:\text{timestamp}$): Insieme ($< o:\text{integer} >$)

IF $:d \geq :a$: genera errore

$Q \leftarrow$ risultato della query

(select $o.id$
from Opera o
where $o.autore =: a^1$)

except

(select $o.id$
From Opera o , Restauro r
where $o.autore =: a$ and $r.opera = o.id$
and $:d \leq r.inizio$ and $r.fine \leq :a$)

return Q

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

opereInPrestitoEspMenoRecente(d:timestamp, d:timestamp):

:insieme (<0:integer, i:timestamp>)

1F :d ≥ :d : genera errore

1 Tmp(op:integer, ist:timestamp)

PK: op references Opera(id)

2 insert into Tmp(op, ist)

3 select o.id, max(e.inizio)

from Opera o, Restauro r, Esposizione e, opEsp oe

where r.fine is null and r.opera = o.id

and :d ≤ r.inizio and r.inizio <= :d

and oe.esp = e.id and oe.opera = o.id

4 Q = select op, ist from Tmp

where ist = (select min(ist) from Tmp)

5 return Q

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