

Esame Es.20230911 – Prova scritta dell'11 Settembre 2023

Si vuole progettare e realizzare *CozyRooms*, un sistema per la gestione di strutture ricettive e dei soggiorni presso di esse.

Il sistema dovrà servire agli **utenti** per pubblicare offerte riguardo le proprie **strutture** (ad esempio alberghi, appartamenti e case vacanza), gestire le **prenotazioni** e i **soggiorni**, ricercare strutture e sistemazioni, e prenotare soggiorni.

Gli **utenti** si iscrivono a *CozyRooms* fornendo **nome, cognome, indirizzo e-mail e numero di telefono**. Gli utenti **gestori** di strutture devono fornire anche il **codice fiscale** e superare un processo di identificazione (il quale è portato avanti utilizzando un servizio esterno).

I **gestori** di strutture identificati con successo possono **registrare** le proprie strutture. Di una **struttura** interessa conoscere il **nome, l'indirizzo, la tipologia** (albergo, casa vacanze, bed & breakfast, ecc.), il **numero di stelle** (un intero compreso tra 1 e 7), ed i **gestori** (possono essere più d'uno).

Ogni **struttura** dispone di una o più **sistemazioni**. Ad esempio, un hotel dispone in genere di numerose camere, mentre un residence può includere diversi appartamenti. Di ogni sistemazione interessa conoscere il **nome, e il numero di posti letto**.

Il costo del pernottamento non è costante durante l'anno, ma dipende dalla data. Il sistema deve permettere ai gestori delle strutture di specificare, **per ogni sistemazione, la tariffa applicata per ogni giorno dell'anno solare**.

Inoltre, il costo del pernottamento in una sistemazione dipende anche dal numero di occupanti. Ad esempio, una camera doppia di un certo hotel, in una certa data, potrebbe costare 60 Euro a notte per un solo occupante e 80 Euro a notte per due occupanti. Il gestore deve specificare la tariffa di per ogni numero di occupanti da 1 al numero di posti letto presenti nella sistemazione.

Gli utenti di *CozyRooms* che vogliono soggiornare presso una struttura, lo fanno tramite un semplice servizio di **prenotazione**. Al momento della prenotazione, un utente sceglie un **intervallo di date, una o più sistemazioni offerte da una stessa struttura e un numero di occupanti** per ognuna di esse, e fornisce un **metodo di pagamento**. Non deve essere possibile richiedere la prenotazione per una sistemazione che è già prenotata in almeno una data nel periodo richiesto. Il gestore della struttura può quindi **accettare o rifiutare le prenotazioni**. L'utente che effettua la prenotazione può **cancellarla** fino al momento del soggiorno (si assuma che la gestione delle penali per cancellazione tardiva sia gestita da un sistema esterno).

CozyRooms prevede un sistema di fedeltà per i propri utenti, che li premia in base ai soggiorni effettuati. Si vuole che tale sistema di fedeltà sia flessibile, dunque i **gestori** di *CozyRooms* devono poter definire dei **livelli progressivi**, ognuno rappresentato da un **nome univoco** e raggiungibile completando un certo numero minimo di prenotazioni nei due anni solari precedenti. Ad esempio, si potrebbero definire i livelli *bronzo, argento e oro*, raggiungibili completando, rispettivamente, 3, 10 e 20 prenotazioni negli ultimi due anni solari.

I gestori delle strutture possono prevedere (ma non è obbligatorio), **per ogni struttura, un tasso di sconto differente per ogni livello di fedeltà**. Tale tasso di sconto è definito tramite un numero reale compreso tra 0 e 1 (esclusi) e si applica al costo totale della prenotazione.

Il sistema deve permettere ai propri utenti di registrare strutture, visualizzare le strutture ricettive in una certa città, controllare la disponibilità di una sistemazione in un certo intervallo di date,

prenotare una o più sistemazioni presso una struttura, conoscere il proprio livello di fedeltà, ottenere un resoconto di tutte le prenotazioni passate, future ed in corso, e calcolare il costo di un soggiorno.

Inoltre, il sistema deve permettere:

- agli utenti registrati interessati a prenotare soggiorni di ottenere le ~~sistemazioni~~ disponibili in un certo intervallo di date e che rispettino un certo insieme di requisiti. In particolare, le strutture restituite devono trovarsi in una data città, appartenere a una data tipologia, avere un dato numero minimo di stelle, e, per le date richieste, avere un prezzo totale (considerando il livello di fedeltà dell'utente) al di sotto di una data soglia. *e il numero di posti letto richiesti*
- a ogni utente gestore di strutture di calcolare, *data una struttura gestita, per ognuno dei mesi dell'anno solare corrente, il numero* ~~di giorni~~ *in cui tale struttura era piena*. Una struttura è considerata piena in una data se ogni sua sistemazione ha almeno una prenotazione per quella data, indipendentemente dal numero di posti letto prenotati.

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

Utile:

- nome: String
- cognome: String
- email: Mail
- Tel: Telefono
- Specializzazione;
- gestore;
- CF: CodiceFiscale
- associato alle Strutture

Tariffa:

- Pers: Interoso
- giorno: Data
- tariffa: Valuta

Struttura:

- nome: String
- indirizzo: Indirizzo
- stelle: 1..7
- associata ai Gestori
- associata alla Tipologia
- associata alle Listenzioni
- associata a un livello
con sconto: Sconto

livello:

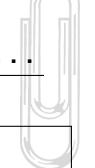
- nome: Stringe {id:1}
- PrezRic: Interoso{id:2}

Listenzione:

- nome: Stringe
- dati: Interoso
- associata alle Tariffe

Prenotazione:

- inizio: Data
- fin: Data
- associata all'Utile
- associata alle Listenzioni
con Pers: Interoso
- associata a un Pagamento
- Specializzazione gestore;
- Accettata
- Rifiutata
- Specializzazione;
- Cancellata



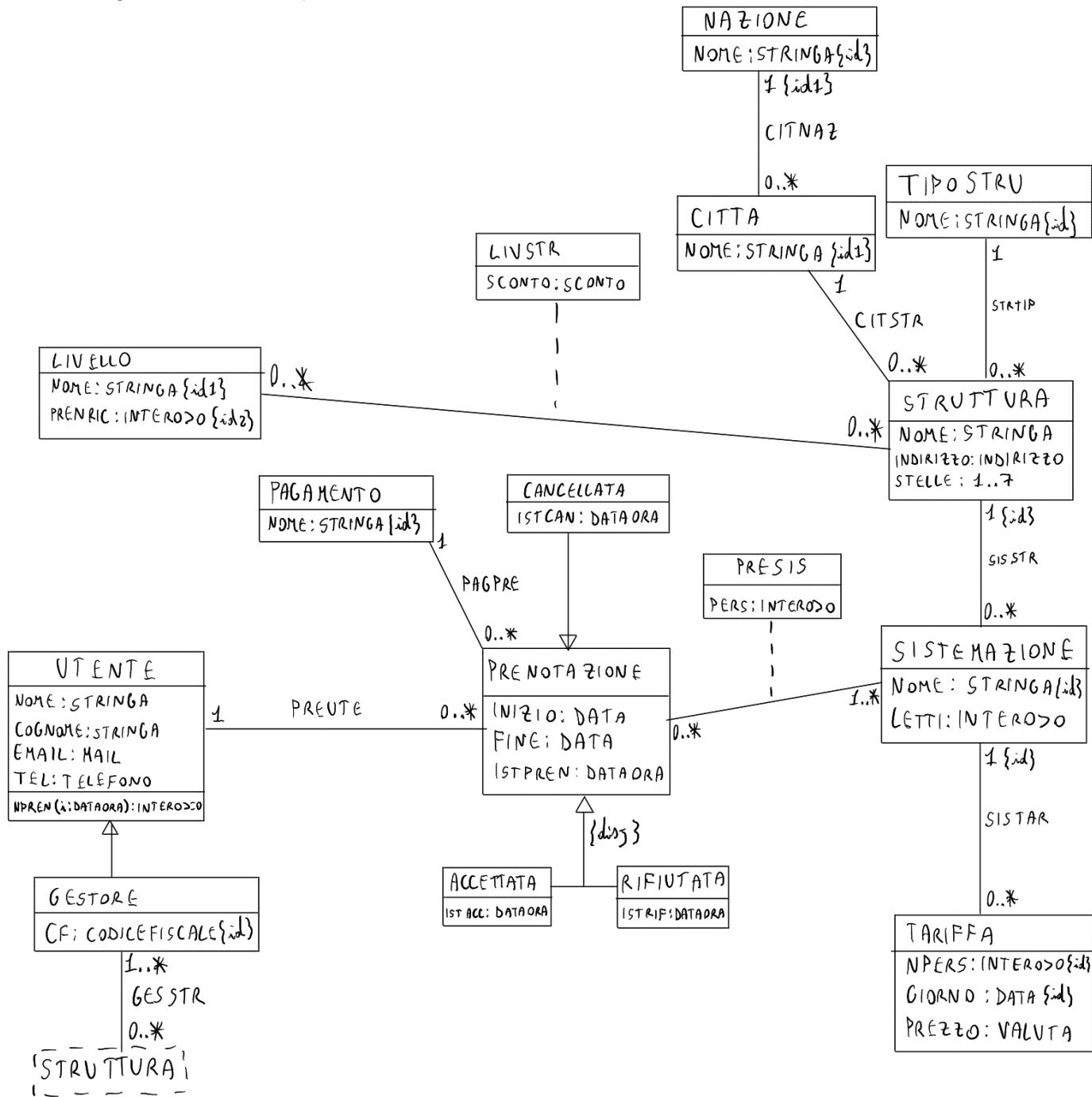
Risposta alla Domanda 1 (segue)

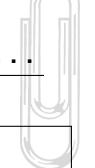
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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'attr. 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>V.T.E.N.T.E</u></p> <p>Operazioni, vincoli: $\text{NPREN}(\text{::DATAORA}) : \text{INTERO} \geq 0$</p> <p>- pre: $P = \left\{ \begin{array}{l} p \mid \text{PREUTE}(p, t) \wedge \exists i \text{ in } \text{INIZIO}(p, i) \wedge \\ \text{in } i \leq i \wedge \text{ACCETTATA}(p) \wedge \neg \text{CANCELLATA}(p) \\ \text{in } i >= i - 2 \text{ ANNI} \end{array} \right\}$</p> <p>RESULT = P</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: <u>P.R.E.N.O.T.A.Z.I.O.N.E</u></p> <p>Operazioni, vincoli: $\text{V.S.T.E.S.S.A.S.T.R.U.T.T.V.R.A}$</p> <p>$\forall p, s_1, s_2 \text{ PRESIS}(p, s_1) \wedge \text{PRESIS}(p, s_2) \rightarrow \exists \text{ str} \text{ SISSTR}(\text{str}, s_1) \wedge \text{SISSTR}(\text{str}, s_2)$</p> <p>V. POSSIBILE</p> <p>$\forall p, s, i, f, t \mid \text{PRESIS}(p, s) \wedge \text{INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge \neg \text{CANCELLATA}(p) \wedge \text{ACCETTATA}(p) \wedge \text{LETTI}(s, t) \wedge \text{PERS}(p, s, p_t) \rightarrow t \geq p_t \wedge \exists p' \mid p' \neq p \wedge \neg \text{PRESIS}(p', s) \wedge \neg \text{INIZIO}(p', i') \wedge \neg \text{FINE}(p', f') \wedge \neg \text{DATA}(t) \wedge i' <= t <= f \wedge i' <= f' \wedge \neg \text{CANCELLATA}(p) \wedge \neg \text{ACCETTATA}(p)$</p>
---	--

3] Tipo: Classe Associazione (cerchiare) Nome: ...ACCETTATA..... Operazioni, vincoli: $\forall, \text{DOPO C REATA PRIMA INIZIO}$ $\forall p, i, ip, ia \quad \text{ACCETTATA}(p) \wedge \text{INIZIO}(p, i) \wedge$ $\text{ISTPREN}(p, ip) \wedge \text{ISTRACC}(p, ia) \rightarrow ip <= ia <= i$

6] Tipo: Classe Associazione (cerchiare) Nome: ...RIFIUTATA..... Operazioni, vincoli: $\forall, \text{DOPO C REATA PRIMA INIZIO}$ $\forall p, i, ip, im \quad \text{RIFIUTATA}(p) \wedge \text{INIZIO}(p, i) \wedge$ $\text{ISTPREN}(p, ip) \wedge \text{ISTRIF}(p, im) \rightarrow ip <= im <= i$
--

4] Tipo: Classe Associazione (cerchiare) Nome: ...CANCELLATA..... Operazioni, vincoli: $\forall, \text{DOPO C REATA PRIMA INIZIO}$ $\forall p, i, ip, ic \quad \text{CANCELLATA}(p) \wedge \text{INIZIO}(p, i) \wedge$ $\text{ISTPREN}(p, ip) \wedge \text{ISTRANC}(p, ic) \rightarrow ip <= ic <= i$ $\forall, \text{NO RIFIUTATA}$ $\forall p \quad \text{CANCELLATA}(p) \rightarrow \neg \text{RIFIUTATA}(p)$
--

7] Tipo: Classe Associazione (cerchiare) Nome: ...PRENOTAZIONE..... Operazioni, vincoli: $\forall, \text{PRIMA INIZIO}$ $\forall p, i, ip, f \quad \text{PRENOTAZIONE}(p) \wedge \text{INIZIO}(p, i) \wedge$ $\text{ISTPRE}(p, ip) \wedge \text{FINE}(p, f) \rightarrow ip < i < f$ $\forall, \text{ACC RIF NON CANC PRIMA}$ $\forall p, i \quad [\text{ACCETTATA}(p) \wedge \text{ISTRACC}(p, i)] \vee$ $[\text{RIFIUTATA}(p) \wedge \text{ISTRIF}(p, i)] \rightarrow \neg (\text{CANCELLATA}(p)$ $\wedge \exists ic \text{ ISTCANC}(p, ic) \wedge ic <= i)$

5] Tipo: Classe Associazione (cerchiare) Nome: ...TARIFA..... Operazioni, vincoli: $\forall s, t, l, n \quad \text{SISTAR}(s, t) \wedge \text{LETTI}(s, l) \wedge$ $\text{NPERS}(t, n) \rightarrow n <= l$
--

8] Tipo: Classe Associazione (cerchiare) Nome: Operazioni, vincoli:

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

SCONTO: REALE ∈ (0,1)

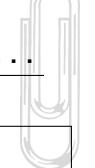
MAIL: STRINGA SECONDO STANDARD

TELEFONO: STRINGA SECONDO STANDARD

INDIRIZZO: (VIA: STRINGA, CIVICO: STRINGA, CAP: [0-9]{5})

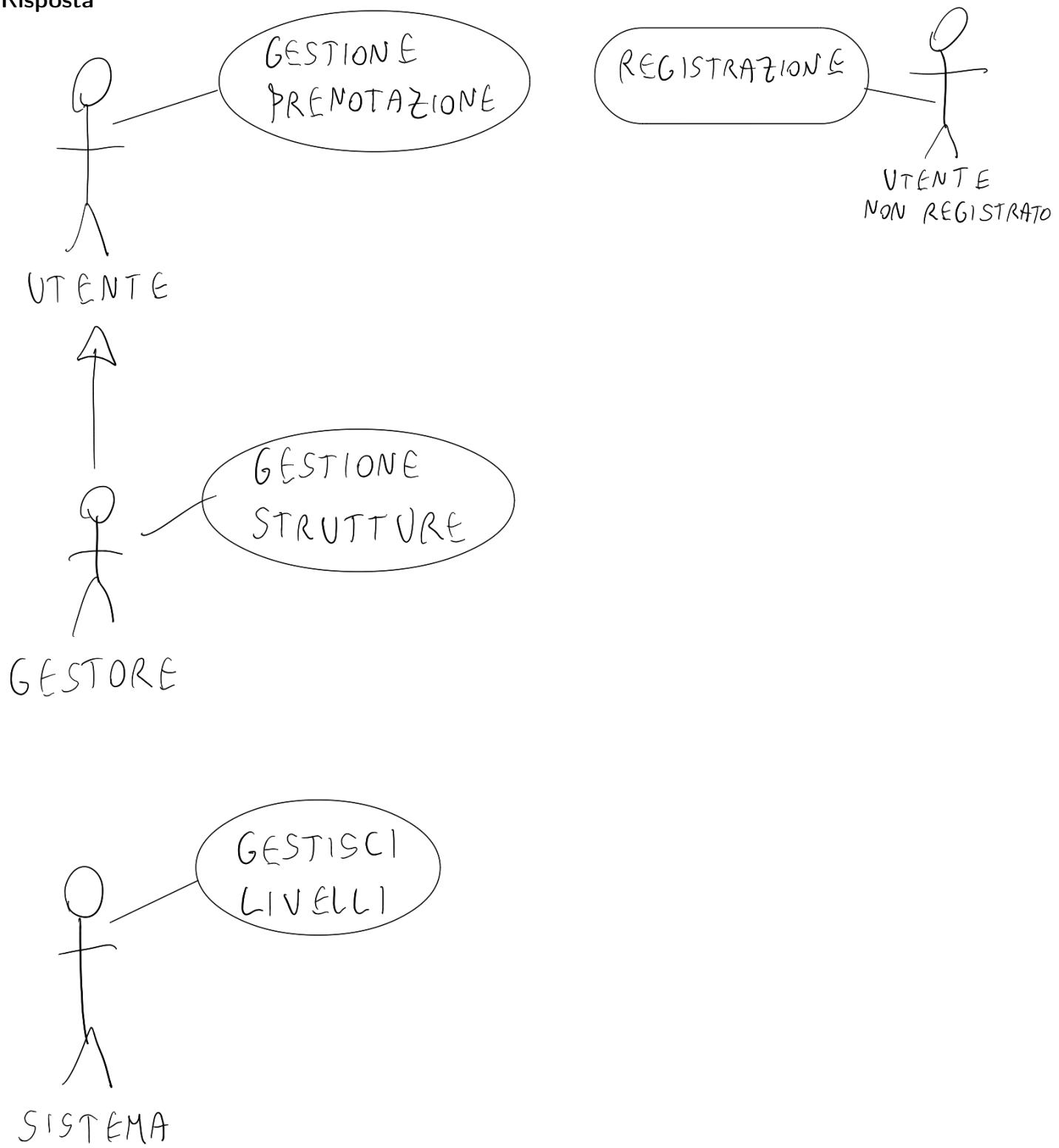
CODICE FISCALE: STRINGA SECONDO STANDARD {16}

VALUTA: REALE > 0



Risposta alla Domanda 2 (segue)

Domanda 3 (5 minuti; 10 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

Risposta

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

Risposta

REGISTRAZIONE:

- NEW_UTENTE (n : STRINGA, c : STRINGA, e : MAIL, t : TELEFONO): UTENTE

GESTIONE PRENOTAZIONE:

- NEW_PREN (\tilde{n} : DATA, f : DATA, u : UTENTE, p : PAGAMENTO, s : SISTEMAZIONE, m : INTERO>0): PREN

- AGGIUNGISIS (p : PRENOTAZIONE, s : SISTEMAZIONE, mp : INTERO>0):

- SISTEMAZIONI DISPONIBILI (\tilde{n} : DATA, f : DATA, c : CITTA, p : VALUTA, t : TIPOSTR, s : 1..7, m : INTERO>0, u : UTENTE): SISTEMAZIONE [0..*]

GESTIONE STRUTTURE

- NEWSTR (n : STRINGA, i : INDIRIZZO, c : CITTA, t : TIPOSTR, s : 1..7, g : GESTORE): STRUTTURA

- NEWSIS (n : STRINGA, l : INTERO>0, s : STRUTTURA): SISTEMAZIONI

- GIORNI PIENI MESE (s : STRUTTURA): (1..12, INTERO>0) [0..*]

Domanda 5 (30 minuti; 60 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti producendo le specifiche concettuali per le operazioni di use-case, **limitandosi** a quelle necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), 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

SISTEMAZIONI DISPONIBILI (i : DATA, f : DATA, c : CITTA, p : VALUTA,
 t : TIPOSTR, s : 1..7, m : INTERO>0, u : UTENTE) : SISTEMAZIONE [0..*]

- pre: $i < f$

- post:

$$S = \left\{ \begin{array}{l} \text{SISTEMAZIONE}(sis) \wedge \exists l \text{ LETTI}(sis, l) \wedge l >= m \wedge \exists atm, str \text{ SISSTR}(sis, atm) \wedge \\ \text{STELLE}(str, atm) \wedge atm >= s \wedge \text{STARTIP}(str, t) \wedge \text{CITSTR}(atm, c) \\ \wedge p = \left\{ \begin{array}{l} \text{DATA}(g) \wedge g >= i \wedge g <= f \wedge \exists tar \text{ PREZZO}(tar, p) \wedge \\ \text{SISTAR}(sis, tar) \wedge \text{NPERS}(tar, n) \wedge \text{GIORNO}(tar, g) \wedge \end{array} \right\} \\ \wedge l = \left\{ \begin{array}{l} (lin, sc) \mid \text{LIVSTR}(lin, atm) \wedge \text{SCONTO}(lin, sc) \wedge \exists mp, pric \text{ PRENRIC}(lin, pric) \\ \wedge \text{NPREN}(u, \text{ADESSO}, mp) \wedge pric <= mp \end{array} \right\} \\ \wedge \left(\sum_{(g, p) \in P} p \cdot \text{MIN}(g) \right) <= p \wedge \exists pu, i', f', t \text{ PRESIS}(pu, sis) \wedge \\ \text{INIZIO}(pu, i') \wedge \text{FINE}(pu, f') \wedge \text{DATA}(t) \wedge i' <= t <= f' \end{array} \right\}$$

RESULT = S

Risposta alla Domanda 5 (segue)

GIORNI PIENI MESE (s ; STRUTTURA): ($t..12$, INTERO) [0..*]

- pre:

- post:

$$\text{RESULT} = \left\{ (m, n) \mid \begin{array}{l} 1 \leq m \leq 12 \wedge \\ n = \left\{ \begin{array}{l} g \mid \text{MESE}(g, m) \wedge \text{ANNO}(g, \text{ADESSO}-1 \text{ ANNO}) \wedge \\ \forall \text{sis } \text{SISSTR}(\text{sis}, s) \rightarrow \\ \exists p, i, f \text{ PRENOTAZIONE}(p) \wedge \text{INIZIO}(p, i) \wedge \\ \text{FINE}(p, f) \wedge i <= g <= f \wedge \text{PRESIS}(p, \text{sis}) \wedge \\ \text{ACCETTATA}(p) \wedge \neg \text{CANCELLATA}(p) \end{array} \right\} \end{array} \right\}$$

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 POSTGRESSQL

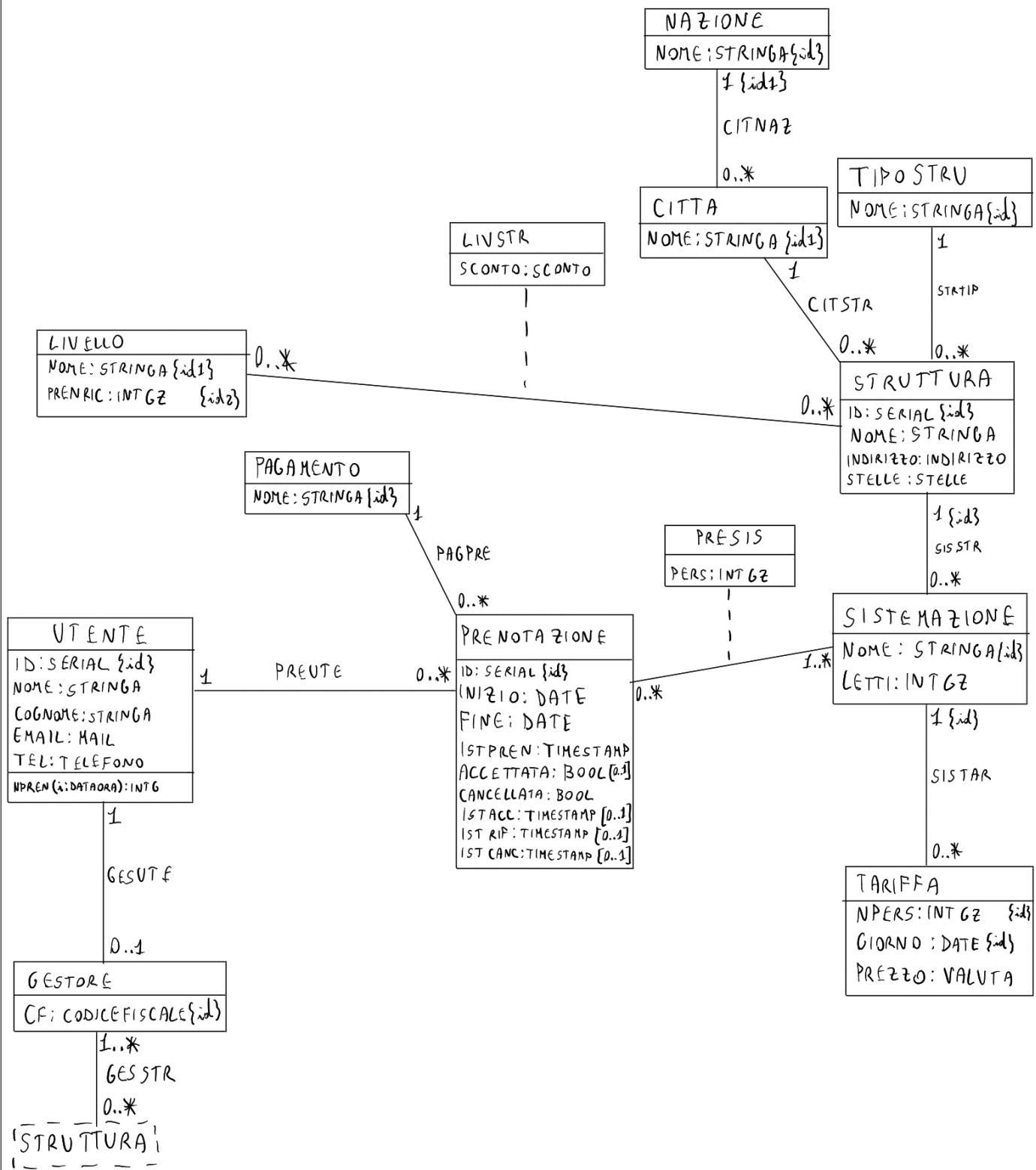
Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```

CREATE DOMAIN STRINGA AS VARCHAR;
CREATE DOMAIN SCONTTO AS REAL CHECK(VALUE>0 AND VALUE<1);
CREATE TYPE INDIRIZZO AS(
    VIA VARCHAR,
    CIVICO VARCHAR,
    CAP CAP);
CREATE DOMAIN CAP AS VARCHAR ILIKE '[0-9]{5}';
CREATE DOMAIN TELEFONO AS VARCHAR;
CREATE DOMAIN MAIL AS VARCHAR;
CREATE DOMAIN CODICEFISCALE AS VARCHAR ILIKE '{16}';
CREATE DOMAIN INTGZ AS INTEGER CHECK(VALUE>0);
CREATE DOMAIN STELLE AS INTEGER CHECK(VALUE>=1 AND VALUE<=7);
CREATE DOMAIN MESE AS INTEGER CHECK(VALUE>=1 AND VALUE<=12);
CREATE DOMAIN INTGET AS INTEGER CHECK(VALUE>=0);
CREATE DOMAIN VALUTA AS REAL CHECK(VALUE>=0);

```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione**FUSIONE SU PRENOTAZIONE****DIVISIONE SU GESTORE****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. PRENOTAZIONE POSSIBILE

$$\forall p, s, i, f \text{ PRESIS}(p, s) \wedge \text{INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge \text{CANCELLATA}(p, \text{FALSE}) \wedge \text{ACCETTATA}(p, \text{TRUE}) \\ \rightarrow \exists p', i', f' \text{ t PRESIS}(p', s) \wedge \text{INIZIO}(p', i') \wedge \text{FINE}(p', f') \wedge \text{DATA}(t) \wedge i < t <= f \wedge i' < t <= f' \wedge \\ \text{CANCELLATA}(p, \text{FALSE}) \wedge \text{ACCETTATA}(p, \text{TRUE})$$
V. ACCETTATA, DOPO CREATURA PRIMA INIZIO

$$\forall p, i, ip, ia \text{ PRENOTAZIONE}(p) \wedge \text{ACCETTATA}(p, \text{TRUE}) \wedge \text{INIZIO}(p, i) \wedge \text{ISTPREN}(p, ip) \wedge \text{ISTRACC}(p, ia) \rightarrow ip <= ia < i$$
V. RIFINTATA, DOPO CREATURA PRIMA INIZIO

$$\forall p, i, ip, ir \text{ PRENOTAZIONE}(p) \wedge \text{ACCETTATA}(p, \text{FALSE}) \wedge \text{INIZIO}(p, i) \wedge \text{ISTPREN}(p, ip) \wedge \text{ISTRIF}(p, ir) \rightarrow ip <= ir < i$$
V. CANCELLATA, DOPO CREATURA PRIMA INIZIO

$$\forall p, i, ip, ic \text{ PRENOTAZIONE}(p) \wedge \text{CANCELLATA}(p, \text{TRUE}) \wedge \text{INIZIO}(p, i) \wedge \text{ISTPREN}(p, ip) \wedge \text{ISTCANC}(p, ic) \rightarrow ip <= ic < i$$
V. CANCELLATA, NON RIFINTATA

$$\forall p \text{ PRENOTAZIONE}(p) \wedge \text{CANCELLATA}(p, \text{TRUE}) \rightarrow \neg \text{ACCETTATA}(p, \text{FALSE})$$

Risposta alla Domanda 6 (segue)
$$\forall p \text{ PRENOTAZIONE}(p) \wedge \text{ACCETTATA}(p, \text{TRUE}) \Leftrightarrow \exists i \text{ ISTACC}(p, i)$$
$$\forall p \text{ PRENOTAZIONE}(p) \wedge \text{RIFINTATA}(p, \text{FALSE}) \Leftrightarrow \exists i \text{ISTRIF}(p, i)$$
$$\forall p \text{ PRENOTAZIONE}(p) \wedge \text{CANCELLATA}(p, \text{TRUE}) \Leftrightarrow \exists i \text{ISTCANC}(p, i)$$

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	<u>UTENTE</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	----------------------	---

Attributi	<u>ID</u>	INOME	COGNOME	EMAIL	TEL			
-----------	-----------	-------	---------	-------	-----	--	--	--

Domini	SERIAL	STRINGA	STRINGA	MAIL	TELEFONO			
--------	--------	---------	---------	------	----------	--	--	--

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	<u>GESTORE</u> (nome)	Derivante da: classe associazione (cerchiare)
--------------------	-----------------------	---

Attributi	<u>CF</u>	UTENTE						
-----------	-----------	--------	--	--	--	--	--	--

Domini	CODICE FISCALE	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 (UTENTE) REF UTENTE (ID)

UNIQUE (UTENTE)

La relazione accorda le relazioni che implementano le seguenti associazioni: GESUTE

3 Relazione	<u>PRENOTAZIONE</u> (nome)	1	Derivante da: classe associazione (cerchiare)
--------------------	----------------------------	---	---

Attributi	<u>ID</u>	INIZIO	FINE	ISTPREN	ACCETTATA*	ISTACC*	ISTRIF*	
-----------	-----------	--------	------	---------	------------	---------	---------	--

Domini	SERIAL	DATE	DATE	TIME STAMP	BOOL	TIMESTAMP	TIMESTAMP	
--------	--------	------	------	------------	------	-----------	-----------	--

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

Vincoli (foreign key, inclusione, altra chiave, di ennupla, di dominio): FK (UTENTE) REF UTENTE (ID)

CHECK (EXTRACT(DATE FROM ISTPREN) < INIZIO) CHECK ((ACCETTATA = TRUE) = (ISTACC IS NOTNULL))

CHECK (EXTRACT(DATE FROM ISTPREN) <= EXTRACT(DATE FROM ISTACC) < INIZIO) CHECK (ISTACC < ISTANC)

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

4 Relazione	<u>PRENOTAZIONE</u> (nome)	2	Derivante da: classe associazione (cerchiare)
--------------------	----------------------------	---	---

Attributi	CANCELLATA	ISTCANC*	UTENTE	PAGAMENTO				
-----------	------------	----------	--------	-----------	--	--	--	--

Domini	BOOL	TIMESTAMP	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): CHECK ((ACCETTATA = FALSE) = (ISTRIF IS NOTNULL))

CHECK (EXTRACT(DATE FROM ISTPREN) <= EXTRACT(DATE FROM ISTRIF) < INIZIO) CHECK ((CANCELLATA = TRUE) = (ISTCANC IS NOTNULL))

CHECK (EXTRACT(DATE FROM ISTPREN) <= EXTRACT(DATE FROM ISTCANC) < INIZIO) CHECK ((CANCELLATA = TRUE) > (ACCETTATA = FALSE))

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

5 Relazione	<u>PAGAMENTO</u> (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:

6 Relazione SISTEMAZIONE (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>NOME</u> LETTERI <u>STR</u>	
Domini STRINGA 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(STR) REF STRUTTURA(ID)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...SIS...STR.....

7 Relazione PREVISIONE (nome)	Derivante da: classe associazione (cerchiare)
Attributi PREN SIS STR PERS	
Domini INTEGER STRINGA 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(PREN) REF PRENOTAZIONE(ID)

FK(SIS,STR) REF SISTEMAZIONE(NOME,STR)

La relazione accorda le relazioni che implementano le seguenti associazioni:

8 Relazione TARIFFE (nome)	Derivante da: classe associazione (cerchiare)
Attributi NPERS GIORNO PREZZO SIG STR	
Domini INTEGER DATE VALUTA STRINGA 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(SIG,STR) REF SISTEMAZIONE(NOME,STR)

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

9 Relazione STRUTTURA (nome)	Derivante da: classe associazione (cerchiare)
Attributi ID NOME <u>INDIRIZZO</u> STELLE TIPO CITTA NAZIONE	
Domini SERIAL STRINGA STRINGA STRINGA 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(TIPO) REF TIPOSTR(NOME)

FK(CITTA,NAZIONE) REF CITTA(NOME,NAZIONE)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...TIPSTR...CITSTR.....

10 Relazione TIPOSTR (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:

11 Relazione ...CITTÀ.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>NOME</u> <u>NAZIONE</u>	
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....

12 Relazione ...NAZIONE... (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:

13 Relazione ...GESSTR... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>GES</u> <u>STR</u>	
Domini CODICEFISCALE 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(STR) REF STRUTTURA(ID)

FK(GES) REF GESTORE(CF)

La relazione accorda le relazioni che implementano le seguenti associazioni:

14 Relazione ...LIVELLO.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>NOME</u> PRENOME	
Domini STRINGA INT62	

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(PRENOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

15 Relazione ...LIVSTR.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>LIV</u> <u>STR</u> SCONT0	
Domini STRINGA INTEGER SCONT0	

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(LIV) REF LIVELLO(NOME)

FK(STR) REF STRUTTURA(ID)

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. PRENOTAZIONE. STESSA STRUTTURA

- INSERT OR UPDATE PRESIS

ERROR := EXISTS (SELECT * FROM PRESIS ps1, PRESIS ps2
 WHERE ps1.PRE = ps2.PRE AND ps1.ID = NEW.ID
 ps1.STR <> ps2.STR)

IF ERROR: ERRORE
 ELSE: COMMIT

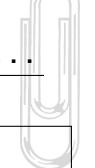
T. PRENOTAZIONE. POSSIBILE

- INSERT OR UPDATE PRESIS

ERROR := EXISTS (WITH PREINFO AS (SELECT * FROM PRENOTAZIONE
 WHERE ID = NEW.ID))

SELECT * FROM PRENOTAZIONE p, PRESIS ps, PREINFO pi
 WHERE p.ID = ps.ID AND ps.SIS = NEW.SIS AND ps.STR = NEW.STR
 AND (pi.INIZIO, pi.FINE) OVERLAPS (p.INIZIO, p.FINE) AND p.ID <> pi.ID)

IF ERROR: ERRORE
 ELSE: COMMIT



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

SISTEMAZIONI DISPONIBILI ($i: DATE, f: DATE, c: STRINGA, n: STRINGA, p: VALUTA, t: STRINGA, s: STELLE, np: INTEGER, u: INTEGER): (STRINGA, INTEGER) [0..*]$
 $| IF(i > f): ERRORE$

$R = ($ SELECT sis.NOME, sis.STR FROM SISTEMAZIONE sis, STRUTTURA str
 WHERE sis.STR = str.ID
 AND sis.LETTI >= np AND str.STELLE >= s AND str.TIPO = t AND str.CITTA = c AND str.NAZIONE = n
 AND NOT EXISTS (SELECT * FROM PRESIS ps, PRENOTAZIONE pre
 WHERE ps.SIS = sis.NOME AND ps.STR = str.ID AND ps.PRE = pre.ID AND
 (pre.INIZIO, pre.FINE) OVERLAPS (i, f))
 AND (SELECT SUM(PREZZO) FROM TARIFFA
 WHERE GIORNO >= n AND GIORNO <= f AND NPERS = np AND SIS = sis.NOME
 AND STR = str.ID)
 $*$
 (SELECT MIN(SCONTO) FROM LIVESTR ls, LIVELLO l
 WHERE ls.LIV = l.NOME AND ls.STR = str.ID AND
 l.PREN_RIC <= NPREN(u, now()) <= p)

RETURN R

Risposta alla Domanda 8 (segue)

GIORNI PIENI MESE (>: INTEGER): (MESE, INTGEZ) [0..*]

```

R = WITH DAYS(d) AS (
    SELECT * FROM
    GENERATE_SERIES(DATE_TRUNC(YEAR, NOW()) - 1 YEAR,
                      DATE_TRUNC(YEAR, NOW()) - 1 DAY, '1 DAY')
)
SELECT EXTRACT(MONTH FROM d) M, COUNT(d) FROM DAYS
WHERE NOT EXISTS (
    SELECT * FROM SISTEMAZIONE sis
    WHERE sis.STR = >
    EXCEPT
    SELECT sis.* FROM SISTEMAZIONE sis, PRESIS ps, PRENOTAZIONE p
    WHERE sis.STR = > AND ps.SIS = sis.NOME AND ps.STR = > AND
          ps.PRE = p.ID AND p.ACCETTATA = TRUE AND p.CANCELLATA = FALSE AND
          p.INIZIOC = > AND p.FINE >= >
    GROUP BY M)
)

```

RETURN R

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.]

```
CREATE FUNCTION NPREN (u:INTEGER, i:TIMESTAMP): INTEGER  
R=SELECT COUNT(id) FROM PRENOTAZIONE  
WHERE UTENTE=u AND ACCETTATA=TRUE AND CANCELLATA=FALSE  
AND INIZIO <= EXTRACT(DATE FROM i) AND  
INIZIO >= EXTRACT(DATE FROM i) - '2 YEAR'  
RETURN R
```