

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 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 massimo 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.



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**

**Dati dello studente e dell'esame**

Cognome e nome: BIANCO SIMONE Matricola: .....

Data: 11/09/2023

Corso di laurea e canale di appartenenza:

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

Firma di un membro della Commissione per  
avvenuta identificazione:

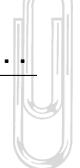
.....

**Rinuncia alla prova**

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



Questo modulo è ottimizzato per la stampa fronte-retro



# Istruzioni e regole d'esame

## Prima dell'esame

- Stampare questo modulo, preferibilmente fronte-retro, e rilegarlo con un fermaglio rimovibile, come quello disegnato in alto
- Compilare il frontespizio con i propri dati, come richiesto
- Scrivere la propria matricola nello spazio apposito nella parte alta di tutte le pagine

## Durante l'esame

- La prova è dimensionata per essere svolta in circa 3 ore. Tuttavia, data la sua natura fortemente progettuale, la Commissione offre agli studenti la più ampia disponibilità di tempo, al fine ovviare ad eventuali (e limitati) errori di analisi/progettazione rilevati più a valle del ciclo di vita.

Il tempo massimo per la consegna è quindi rilassato a 5 ore (il massimo tempo compatibile con le disponibilità di aule).

- Scrivere le risposte negli spazi predisposti sotto le relative domande. Le ultime pagine sono vuote e possono essere usate come minute oppure, se puntate opportunamente, per contenere risposte in caso gli spazi appositi dovessero risultare insufficienti.
- Non è possibile usare alcun tipo di materiale didattico.
- In caso di necessità di ulteriori fogli (in proprio possesso), chiedere preventivamente alla Commissione una nuova procedura di controllo.
- La Commissione può rispondere solo a brevi domande inerenti al testo dei quesiti.
- Tra la seconda e la quarta ora d'esame, gli studenti possono effettuare **brevi pause** (uno studente alla volta) seguendo la seguente procedura:
  1. Alla lavagna è riportata una coda denominata 'Coda prenotazioni pause'. Sia  $n$  (un intero) l'elemento in fondo alla coda (si assuma  $n = 0$  in caso di coda vuota).
  2. Recarsi alla lavagna ed aggiungere l'intero  $n + 1$  come proprio contrassegno in fondo alla coda, seguito da una stringa a propria scelta (ad es., le proprie iniziali).
  3. Se il proprio contrassegno non è l'elemento affiorante della coda, tornare al lavoro in attesa che lo diventi.
  4. Consegnare tutti i fogli di lavoro e il testo d'esame alla Commissione ed uscire.
  5. Al rientro, cancellare il proprio contrassegno dalla coda di modo da permettere al successivo studente prenotato di uscire, e riprendere i fogli prima consegnati.

## Al momento della consegna

- Ordinare tutti i fogli che si vuole far valutare e rilegarli con un fermaglio rimovibile. Non includere fogli che la Commissione non deve valutare (ad es., requisiti, minute), ma includere ovviamente il frontespizio.
- Consegnare i fogli ordinati **nelle mani** di un membro della Commissione. **Non lasciare l'aula senza la conferma, da parte della Commissione, del buon esito delle operazioni di consegna.**

## In caso di rinuncia

- È possibile rinunciare alla consegna a partire dalla seconda ora d'esame. In caso di rinuncia, consegnare nelle mani della Commissione solo il frontespizio, dopo aver compilato e firmato la sezione dedicata.

## Sommario delle domande

Si richiede di progettare l'applicazione descritta dalla specifica dei requisiti effettuando le fasi di Analisi concettuale dei requisiti e di Progettazione logica della base dati e delle funzionalità, utilizzando la metodologia vista nel corso.

In particolare (vengono indicati i tempi suggeriti per i diversi passi chiave):

**Parte 1: Analisi concettuale dei requisiti** Effettuare la fase di Analisi concettuale dei requisiti producendo lo schema concettuale per l'applicazione, che includa:

- Analisi dei dati (45 minuti; 75 minuti al massimo):
  - un diagramma ER concettuale (\*)
  - il relativo dizionario dei dati
  - le specifiche dei domini concettuali non di tipo base
  - eventuali vincoli esterni, espressi utilizzando il linguaggio della logica del primo ordine (\*)
- Analisi delle funzionalità:
  - un diagramma UML degli use-case (5 minuti; 10 minuti al massimo)
  - la segnatura di tutte le operazioni di use-case (10 minuti)
  - la specifica delle operazioni di use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra) in termini di precondizioni e postcondizioni, utilizzando il linguaggio della logica del primo ordine (\*) (30 minuti; 60 minuti al massimo)

**Parte 2: Progettazione della base dati e delle funzionalità** Effettuare la progettazione della base dati e delle funzionalità a partire dallo schema concettuale prodotto nella Parte 1, ed in particolare eseguire i seguenti passi:

- Progettazione della base dati relazionale con vincoli:
  - Ristrutturazione del diagramma ER concettuale e dei vincoli esterni (20 minuti; 30 minuti al massimo):
    - \* scelta del DBMS da utilizzare
    - \* progettazione della corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
    - \* ristrutturazione del diagramma ER concettuale e dei vincoli esterni.
  - Produzione dello schema relazionale della base dati e dei relativi vincoli (\*) (30 minuti; 60 minuti al massimo)
- Progettazione delle funzionalità (30 minuti; 45 minuti al massimo):
  - definizione della specifica realizzativa delle operazioni di use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale, in modo conforme alla loro specifica concettuale prodotta nella fase di Analisi, in termini di algoritmi in pseudo-codice e comandi SQL. (\*)

---

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

Le pagine seguenti contengono le domande specifiche a cui è richiesto rispondere, ulteriori delucidazioni per ogni singolo punto, e spazi per le risposte.

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



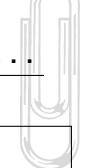
Questa pagina è stata intenzionalmente lasciata vuota

## 1 Analisi concettuale

**Domanda 1 (10 minuti)** Raffinare la specifica dei requisiti eliminando inconsistenze, omissioni e ridondanze e producendo un elenco numerato di requisiti il meno ambiguo possibile. (La risposta a questa domanda non sarà valutata, ma si consiglia di svolgere accuratamente questo passo, in quanto può facilitare di molto le attività di progetto.)

### Risposta

*Bozza del professore : /*



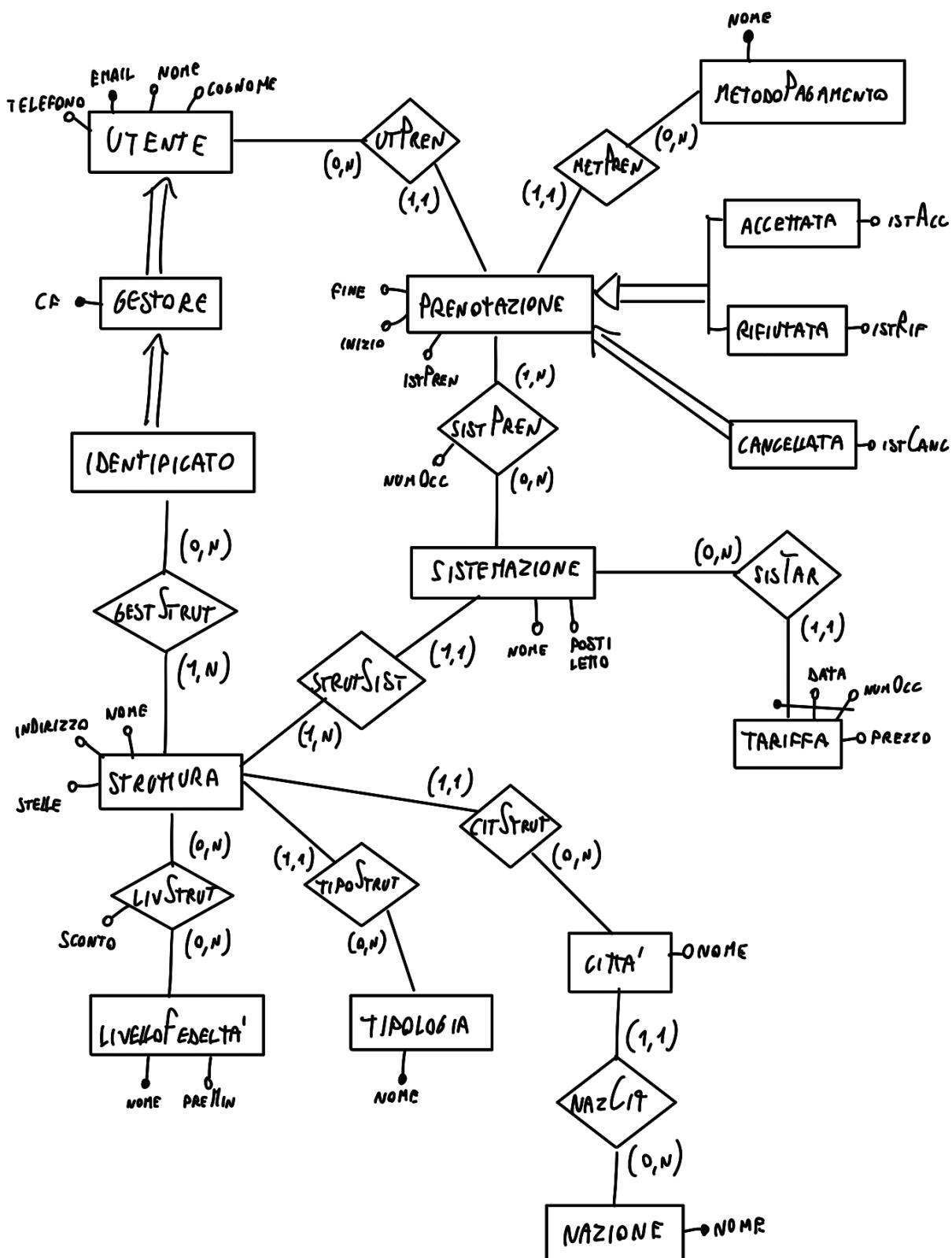
**Risposta alla Domanda 1 (segue)**

**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.

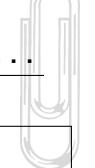
Una risposta soddisfacente a questa domanda è condizione *necessaria* (ma non sufficiente) per superare la prova.

### Diagramma ER

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



[continua alla pagina seguente]



**Risposta alla Domanda 2 (segue)**

**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: <b>Entità</b>   Relationship (cerchiare)</p> <p>Nome: ..... <b>UTENTE</b> .....</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td>EMAIL</td><td>EMAIL</td><td></td></tr> <tr> <td>NOME</td><td>STRINGA</td><td></td></tr> <tr> <td>COGNOME</td><td>STRINGA</td><td></td></tr> <tr> <td>TELEFONO</td><td>TELEFONO</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	EMAIL	EMAIL		NOME	STRINGA		COGNOME	STRINGA		TELEFONO	TELEFONO		<p>[2] Tipo: <b>Entità</b>   Relationship (cerchiare)</p> <p>Nome: ..... <b>GESTORE</b> .....</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">attributo</th><th style="text-align: left;">dominio</th><th style="text-align: left;">moltep. (*)</th></tr> </thead> <tbody> <tr> <td>CF</td><td>codfis</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	CF	codfis	
attributo	dominio	moltep. (*)																				
EMAIL	EMAIL																					
NOME	STRINGA																					
COGNOME	STRINGA																					
TELEFONO	TELEFONO																					
attributo	dominio	moltep. (*)																				
CF	codfis																					

3 Tipo: **Entità** | Relationship (cerchiare)  
Nome: .....  
**CITTÀ**

attributo	dominio	moltep. (*)
<b>NOME</b>	<b>STRINGA</b>	

---

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

Vincoli:

5 Tipo: **Entità** | Relationship (cerchiare)  
Nome: .....  
**NAZIONE**

attributo	dominio	moltep. (*)
<b>NOME</b>	<b>STRINGA</b>	

---

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

Vincoli:

4 Tipo: **Entità** | Relationship (cerchiare)  
Nome: .....  
**TIPOLOGIA**

attributo	dominio	moltep. (*)
<b>NOME</b>	<b>STRINGA</b>	

---

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

Vincoli:

6 Tipo: **Entità** | Relationship (cerchiare)  
Nome: .....  
**LIVELLOFEDELTA'**

attributo	dominio	moltep. (*)
<b>NOME</b>	<b>STRINGA</b>	
<b>PREMIS</b>	<b>INTEROPO</b>	

---

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

Vincoli:

- V. LIVELLOFEDELTA'. PROGRESSIVO

$\forall l, l', n, n', nc, nc' \text{ struttura}(n) \wedge \text{livstrut}(n, l) \wedge \text{livstrut}(n, l') \wedge l \neq l' \wedge \text{prem}(l, n) \wedge \text{prem}(l', n') \wedge \text{scontro}(n, l, nc) \wedge \text{scontro}(n, l', nc') \rightarrow (n < n' \rightarrow nc < nc')$

<p>7 Tipo: <b>Entità</b> Relationship (cerchiare)</p> <p>Nome: ..... <b>STRUTTURA</b> .....</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td><b>NOME</b></td><td><b>STRINGA</b></td><td></td></tr> <tr> <td><b>INDIRIZZO</b></td><td><b>INDIRIZZO</b></td><td></td></tr> <tr> <td><b>STELLE</b></td><td><b>INTERO[1,7]</b></td><td></td></tr> </tbody> </table>	attributo	dominio	moltep. (*)	<b>NOME</b>	<b>STRINGA</b>		<b>INDIRIZZO</b>	<b>INDIRIZZO</b>		<b>STELLE</b>	<b>INTERO[1,7]</b>		<p>9 Tipo: <b>Entità</b> Relationship (cerchiare)</p> <p>Nome: ..... <b>TARIFFE</b> .....</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td><b>DATA</b></td><td><b>DATA</b></td><td></td></tr> <tr> <td><b>NUMOcc</b></td><td><b>INTERO</b></td><td></td></tr> <tr> <td><b>PREZZO</b></td><td><b>DENARO</b></td><td></td></tr> </tbody> </table>	attributo	dominio	moltep. (*)	<b>DATA</b>	<b>DATA</b>		<b>NUMOcc</b>	<b>INTERO</b>		<b>PREZZO</b>	<b>DENARO</b>	
attributo	dominio	moltep. (*)																							
<b>NOME</b>	<b>STRINGA</b>																								
<b>INDIRIZZO</b>	<b>INDIRIZZO</b>																								
<b>STELLE</b>	<b>INTERO[1,7]</b>																								
attributo	dominio	moltep. (*)																							
<b>DATA</b>	<b>DATA</b>																								
<b>NUMOcc</b>	<b>INTERO</b>																								
<b>PREZZO</b>	<b>DENARO</b>																								
<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> <li>• <b>V. PRENOTAZIONE.</b></li> </ul>	<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> <li>• <b>V. TARIFFE. PRENOTAZIONE</b></li> </ul> $\forall p, t, i, f, di, df, d, n \text{ PRENOTAZIONE}(p) \wedge \\ \text{INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge \text{DATA}(i, di) \wedge \text{DATA}(f, df) \wedge \\ \text{DATA}(d) \wedge di \leq d \wedge d \leq df \wedge \text{SISTPREN}(p, t) \wedge \\ \text{NUMOcc}(p, n, n) \rightarrow \exists t \text{ TARIFFE}(t) \wedge \text{SISTTAR}(t, n) \wedge \\ \text{DATA}(t, d) \wedge \text{NUMOcc}(t, n)$																								

<p>8 Tipo: <b>Entità</b> Relationship (cerchiare)</p> <p>Nome: ..... <b>PRENOTAZIONE</b> .....</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td><b>ISTPREN</b></td><td><b>DATAORA</b></td><td></td></tr> <tr> <td><b>INIZIO</b></td><td><b>DATAORA</b></td><td></td></tr> <tr> <td><b>FINE</b></td><td><b>DATAORA</b></td><td></td></tr> </tbody> </table>	attributo	dominio	moltep. (*)	<b>ISTPREN</b>	<b>DATAORA</b>		<b>INIZIO</b>	<b>DATAORA</b>		<b>FINE</b>	<b>DATAORA</b>		<p>10 Tipo: <b>Entità</b> Relationship (cerchiare)</p> <p>Nome: ..... <b>ACCETTATA</b> .....</p> <table border="1"> <thead> <tr> <th>attributo</th><th>dominio</th><th>moltep. (*)</th></tr> </thead> <tbody> <tr> <td><b>ISTAcc</b></td><td><b>DATAORA</b></td><td></td></tr> </tbody> </table>	attributo	dominio	moltep. (*)	<b>ISTAcc</b>	<b>DATAORA</b>	
attributo	dominio	moltep. (*)																	
<b>ISTPREN</b>	<b>DATAORA</b>																		
<b>INIZIO</b>	<b>DATAORA</b>																		
<b>FINE</b>	<b>DATAORA</b>																		
attributo	dominio	moltep. (*)																	
<b>ISTAcc</b>	<b>DATAORA</b>																		
<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> <li>• <b>V. PRENOTAZIONE. CONTINUITA'</b></li> </ul> $\forall p, ip, i, f \text{ PRENOTAZIONE}(p) \wedge \text{ISTPREN}(p, ip) \wedge \text{INIZIO}(p, i) \wedge \\ \text{FINE}(p, f) \rightarrow ip < i \wedge i < f$	<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> <li>• <b>V. ACCETTATA. CONTINUITA'</b></li> </ul> $\forall p, ip, i, f \text{ PRENOTAZIONE}(p) \wedge \text{ISTPREN}(p, ip) \wedge \text{INIZIO}(p, i) \wedge \\ \text{FINE}(p, f) \rightarrow ip < i \wedge i < f$																		

11	Tipo: <b>Entità</b>   Relationship (cerchiare)
Nome: .....	<b>RIFIUTATA</b>
attributo	dominio
ISTRIP	MOLTEPL. (*)
	<b>NATAORA</b>

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

Vincoli:

•  $\forall RIFIUTATA . CONTINUITA'$

$$\begin{aligned} & \forall p, i, i, \nexists PRENOTAZIONE(p) \wedge ISPREN(p, i) \wedge INIZIO(p, i) \\ \rightarrow & (\exists i' ISRIP(p, i') \rightarrow i_p < i' \wedge i' < i) \end{aligned}$$

13	Tipo: <b>Entità</b>   Relationship (cerchiare)
Nome: .....	<b>CANCELLATA</b>
attributo	dominio
ISLANC	MOLTEPL. (*)
	<b>NATAORA</b>

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

Vincoli:

•  $\forall CANCELLATA . CONTINUITA'$

$$\begin{aligned} & \forall p, i, i, \nexists PRENOTAZIONE(p) \wedge ISPREN(p, i) \wedge INIZIO(p, i) \\ \rightarrow & (\exists i' ISLANC(p, i') \rightarrow i_p < i' \wedge i' < i) \end{aligned}$$

12	Tipo: <b>Entità</b>   Relationship (cerchiare)
Nome: .....	<b>SISTPREN</b>
attributo	dominio
NUMOcc	MOLTEPL. (*)
	<b>INTERO70</b>

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

Vincoli:

•  $\forall SISTPREN . DISGIUNTE$

$$\begin{aligned} & \forall p, p', n, i, i', \nexists SISTPREN(p, n) \wedge SISTPREN(p', n) \\ \wedge & SISTPREN(p, n) \wedge p \neq p' \wedge ACCETTATA(p) \wedge ACCETTATA(p') \\ \wedge & \neg CANCELLATA(p) \wedge \neg CANCELLATA(p') \wedge INIZIO(p, i) \wedge \\ & INIZIO(p', i') \wedge PINE(p, i) \wedge FINE(p', i') \rightarrow \\ & \neg NATAORA(i) \wedge (i \leq t \wedge t \leq i') \wedge (i \leq t \wedge t \leq i') \end{aligned}$$

14	Tipo: <b>Entità</b>   Relationship (cerchiare)
Nome: .....	<b>SISTEMAZIONE</b>
attributo	dominio
NOME	MOLTEPL. (*)
POSTIETTO	<b>INTERO70</b>
	STRINBA

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

Vincoli:

•  $\forall SISTEMAZIONE . STRUTTUAA$

$$\begin{aligned} & \forall p, n, n', n'' \exists PRENOTAZIONE(p) \wedge SISTPREN(p, n) \wedge \\ & SISTPREN(p, n') \wedge STRUTSIS(p, n) \wedge STRUTSIS(n', n'') \rightarrow \\ & n = n'' \end{aligned}$$

•  $\forall SISTEMAZIONE . POSTIETTO$

$$\begin{aligned} & \forall n, p, n, n' \exists SISTEMAZIONE(n) \wedge SISTPREN(p, n) \wedge \\ & NUMOcc(p, n, n') \wedge POSTIETTO(p, n') \rightarrow n \leq n' \end{aligned}$$

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

Nome: ..... **LIVSTRUT**

attributo	dominio	moltepl. (*)
<b>SCANTO</b>	<b>REALE(0,1)</b>	

---

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

Vincoli:

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

Nome: ..... **METODO PAGAMENTO**

attributo	dominio	moltepl. (*)
<b>NOME</b>	<b>STRINGA</b>	

---

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

Vincoli:

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

Nome: .....

attributo	dominio	moltepl. (*)

---

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

Vincoli:

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

Nome: .....

attributo	dominio	moltepl. (*)

---

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

Vincoli:

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

DOMINIO EMAIL : STRINGA SECONDO STANDARD

DOMINIO CODFIS : STRINGA 10 CARATTERI SECONDO STANDARD

DOMINIO TELEFONO :

PREF: STRINGA NUMERICA SECONDO STANDARD

POST: STRINGA NUMERICA SECONDO STANDARD

DOMINIO DENARO:

VALUTA: STRINGA 3 CARATTERI

IMPORTO: REALE 7,0

DOMINIO INDIRIZZO:

VIA: STRINGA

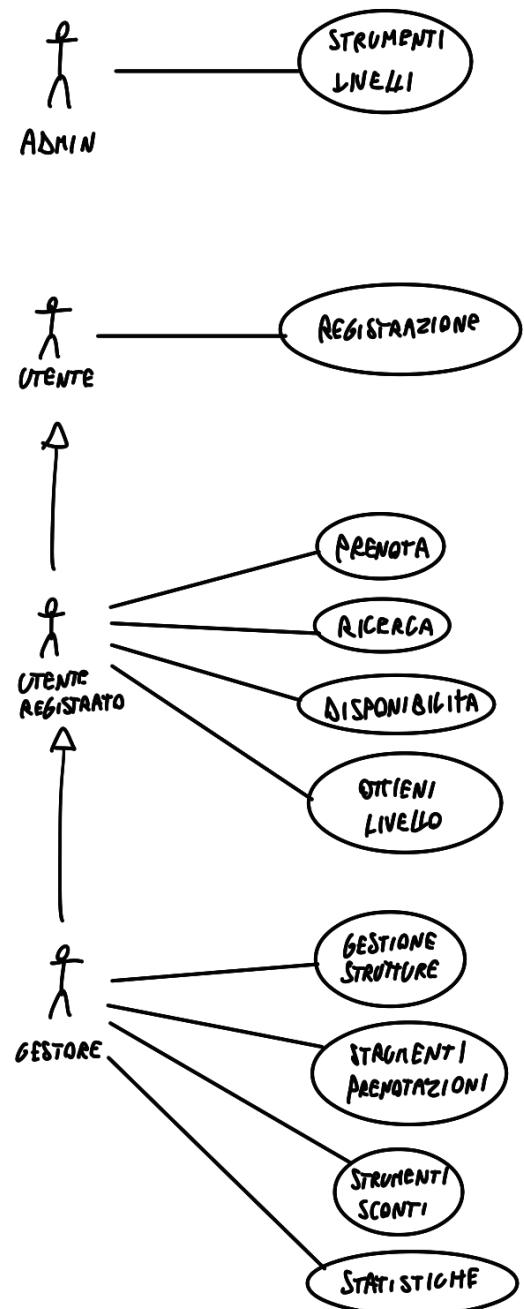
CIVICO: INTERO 70

CAP: INTERO 70

**Risposta alla Domanda 2 (segue)**

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

### Risposta



**Domanda 4 (10 minuti)** Proseguire la fase di Analisi Concettuale dei requisiti definendo le operazioni degli use-case.

In particolare, per ogni use-case definito nella risposta alla **Domanda 3** definire la **segnatura** di tutte le operazioni che lo compongono, in termini di nome dell'operazione, nomi e dominio concettuale degli argomenti, dominio concettuale dell'eventuale valore di ritorno.

**1 Specifica use-case:** ..... **STRUMENTI LINEA!** ..... (nome use-case)

Operazioni dello use-case:

**AGGIUNGI LIVELLO (n: STRINGA, n': INTERO70) : LIVELLO**

**2 Specifica use-case:** ..... **REGISTRAZIONE** ..... (nome use-case)

Operazioni dello use-case:

**REGISTRATUTENTE (e: EMAIL, n: STRINGA, c: STRINGA, t: TELEFONO) : UTENTE**

**3 Specifica use-case:** ..... **PRENOTA** ..... (nome use-case)

Operazioni dello use-case:

**PRENOTA (u: UTENTE, i: DATAORA, f: DATAORA, m: METODOPAGAMENTO, (s: SISTEMAZIONE, n: INTERO70)(1, n)) : PRENOTAZIONE**

**CANCELLA\_PRENOTAZIONE (p: PRENOTAZIONE) : CANCELLATA**

4 Specifica use-case: ..... **RICERCA** ..... (nome use-case)

Operazioni dello use-case:

**RICERCA SIST/ALIDE** ( $u: Utente, c: Città, m: intero[1,7], i: DATA, f: DATA, p: denaro, n: intero[0]$ ): **SISTEMAZIONE(0,N)**

**RICERCA STRUTTURE** ( $c: Città$ ): **STRUTTURA(0,N)**

5 Specifica use-case: ..... **DISPONIBILE** ..... (nome use-case)

Operazioni dello use-case:

**DISPONIBILE** ( $a: SISTEMAZIONE, i: DATAORA, x: DATAORA$ ) : **BOOLEAN**

6 Specifica use-case: ..... **OBTIENE LIVELLO** ..... (nome use-case)

Operazioni dello use-case:

**CALCOLA LIV(Utente)** ( $u: Utente$ ): **LIVELLOFEDELTA**

7 Specifica use-case: ..... **GESTIONE STRUTTURE** ..... (nome use-case)

Operazioni dello use-case:

**AGGIUNGI STRUTTURA** ( $n: STRINGA, i: INDIRIZZO, m: intero[1,4], s: SISTEMAZIONE(1,N), c: Città, t: TIPOLOGIA$ ): **STRUTTURA**

**AGGIUNGI GESTORE** ( $a: STRUTTURA, u: IDENTIFICATO$ )

**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

**CALCOLALIVUTENTE (u: UTENTE): LIVELLOFDELTA'**

PRE:

POST:

- NO SIDE EFFECT
- VAL RITORNO

Esister  $a, a'$  tali che ANNO(ALESSO, a)  $\wedge$   $a' = a - 1$

$$L = \left\{ (l, n) \mid \begin{array}{l} \text{livello}(l) \wedge \text{PREMIN}(l, n) \wedge \\ n \leq \left| \left\{ p \mid \begin{array}{l} \text{PRENOTAZIONE}(p) \wedge \text{UTPRE}(p, u) \wedge \text{ACCETTATA}(p) \wedge \neg \text{CANCELLATA}(p) \wedge \\ (\exists i, f, ai, af) \text{ INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge \text{ANNO}(i, ai) \wedge \text{ANNO}(f, af) \\ \wedge ai < af \wedge i \leq n \end{array} \right\} \right| \end{array} \right\}$$

$$(l', n') \in \text{ANOMAX}(n) \\ (l, n) \in L$$

$$\text{RESULT} = l'$$

**DISPONIBILE (A: SISTEMAZIONE, i: DATAORA, f: DATAORA): BOOLEAN**

PRE:  $i < f$

POST:

- NO SIDE EFFECT
- VAL RITORNO

$$P = \left\{ p \mid \begin{array}{l} \text{PRENOTAZIONE}(p) \wedge \text{ACCETTATA}(p) \wedge \neg \text{CANCELLATA}(p) \wedge \\ (\exists i, f, t) \text{ inizio}(p, i) \wedge \text{FINE}(p, f) \wedge \text{DATAORA}(t) \wedge i \leq i' \leq t \wedge t \leq f' \wedge f' \leq f \end{array} \right\}$$

$$(P = \emptyset \rightarrow \text{RESULT} = \text{TRUE}) \wedge (P \neq \emptyset \rightarrow \text{RESULT} = \text{FALSE})$$

**COSTOSISTEMAZIONE (A: SISTEMAZIONE, i: DATA, f: DATA, l: LIVELLOFDELTA', n: INTERO>0): DENARO**

PRE:  $i < f \wedge (\forall d \text{ DATA}(d) \wedge i \leq d \wedge d \leq f \rightarrow \exists t \text{ TARIFFE}(t) \wedge \text{DATA}(t, d) \wedge \text{NUMOCC}(t, n) \wedge \text{SISTAR}(t, A))$

POST:

- N.S.E.
- VAL RITORNO

$$T = \left\{ (t, p) \mid \begin{array}{l} \text{TARIFFE}(t) \wedge \text{PREZZO}(t, p) \wedge \text{NUMOCC}(t, n) \wedge \\ (\exists d \text{ DATA}(t, d) \wedge i \leq d \wedge d \leq f) \end{array} \right\}$$

$$(\exists ac, ac' \text{ SISTAR}(t, ac') \wedge \text{STURFLIV}(l, ac') \wedge \text{scanno}(l, ac', ac) \rightarrow \text{RESULT} = \sum_{(t, p) \in T} p \cdot (1 - ac))$$

$$(\exists ac, ac' \text{ SISTAR}(t, ac') \wedge \text{STURFLIV}(l, ac') \wedge \text{scanno}(l, ac', ac) \rightarrow \text{RESULT} = \sum_{(t, p) \in T} p)$$

[continua alla pagina seguente]

## Risposta alla Domanda 5 (segue)

**RICERCA SIST/VALIDE** ( $u: UTENTE, c: CITTÀ, n: \text{INTERO}[1,7], l: DATA, f: DATA, p: DENARO, m: \text{INTERO}>0$ ):  $\text{SISTEMAZIONE}(n, n)$

PRE:  $i < f$

POST:

- N. S. E.

- VAL RITORNO

Sia  $l$  tale che  $l = \text{CALCOLA\_L}(u)$

$$S = \left\{ a \mid \begin{array}{l} \text{SISTEMAZIONE}(n) \wedge (\exists n', n', n'' \text{ POSTLETTRO}(n, n') \wedge n' \geq n \wedge \text{STRUCT}(n, n') \wedge \\ \text{CIT\_STRUT}(n', c) \wedge \text{TIPOSTRUT}(n', t) \wedge \text{costoSISTEMAZIONE}(n, i, f, l, n) \leq p \wedge \\ \text{DISPONIBILE}(n, i, f) \wedge \text{STELLE}(n', n'') \wedge n'' \geq n) \end{array} \right\}$$

**NUM\_GIORNI\_PENI** ( $a: \text{STRUTTURA}$ ):  $(m: \text{INTERO}[1,12], n: \text{INTERO}>0)$  ( $0, n$ )

PRE:

POST:

Finora  $di, dx$  tali che  $\exists a \text{ ANNO}(\text{ADESSO}, a) \wedge \text{DATA}(di) \wedge \text{DATA}(dx) \wedge \\ \text{ANNO}(di, a) \wedge \text{ANNO}(dx, a) \wedge \text{MESE}(di, 1) \wedge \text{MESE}(dx, 12) \wedge \text{giorno}(di, 1) \wedge \text{giorno}(dx, 31)$

$$M = \left\{ (m, k) \mid \begin{array}{l} \text{MESE}(m) \wedge \\ k = \left| \begin{array}{l} d \mid \begin{array}{l} \text{DATA}(d) \wedge (di \leq d \wedge d \leq dx) \wedge \text{MESE}(d, m) \wedge (\forall a' \text{ SISTEMAZIONE}(a')) \\ \text{STRUCT}(a', a) \rightarrow \exists p, i, f \text{ PRENOTAZIONE}(p) \wedge \text{ACCETTATA}(p) \\ \wedge \neg \text{CANCELLATA}(p) \wedge \text{INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge i \leq d \wedge d \leq f \end{array} \right| \end{array} \right\}$$

RESULT = M

## 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:

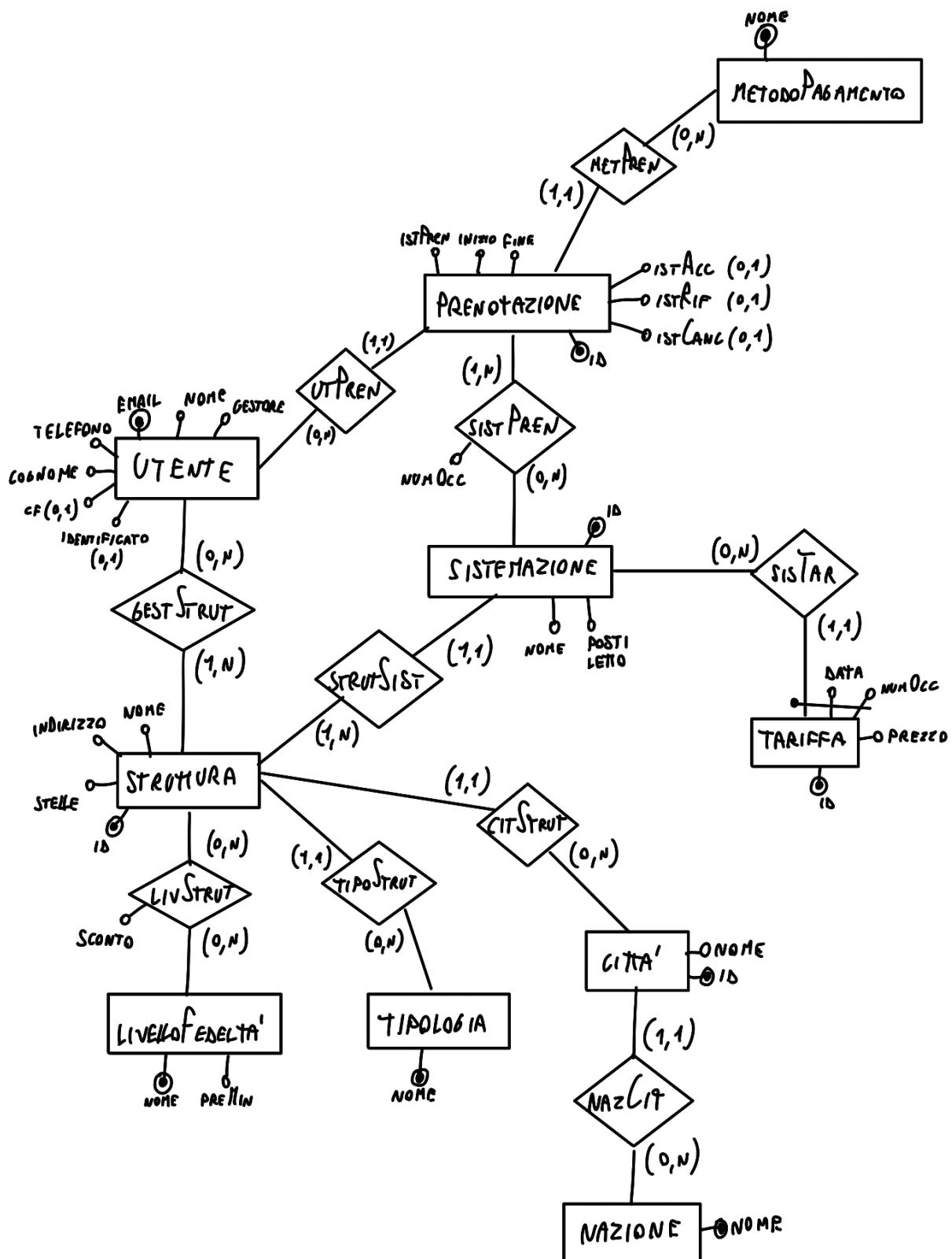
- progettare una corrispondenza tra i domini concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore primario 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 INT-6Z AS INTEGER CHECK (VALUE >= 0)
- CREATE DOMAIN INT-6EZ AS INTEGER CHECK (VALUE >= 0)
- CREATE DOMAIN INT-1-7 AS INTEGER CHECK (VALUE >= 1 AND VALUE <= 7 )
- CREATE DOMAIN REAL-6EZ AS REAL CHECK (VALUE >= 0)
- CREATE DOMAIN REAL-0-1 AS REAL CHECK ( VALUE > 0 AND VALUE < 1 )
- CREATE DOMAIN STR AS VARCHAR(200)
- CREATE DOMAIN EMAIL AS STR CHECK (ISVALIDEMAIL(VALUE))
- CREATE DOMAIN CONFIS AS CHAR(16) CHECK (ISVALID(F(VALUE)))
- CREATE TYPE TELEFONO AS (
   
 PAE VARCHAR(5),
   
 SUFF VARCHAR(15)
 )
- CREATE TYPE DENARO AS (
   
 VALUTA CHAR(3),
   
 IMPATO REAL-6EZ
 )
- CREATE TYPE INDIRIZZO AS (
   
 VIA STR,
   
 CIVICO INT-6Z,
   
 CAP INT-6Z
 )

## Diagramma ER ristrutturato

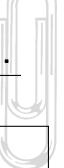


### Breve descrizione delle scelte effettuate durante la ristrutturazione

- ACCORPAMENTO UTENTE
- ACCORPAMENTO PRENOTAZIONE

**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. UTENTE.CF UNICO  
 $\forall u, u' \text{ UTENTE}(u) \wedge \text{UTENTE}(u') \wedge u \neq u' \rightarrow (\exists c, c' \text{ CF}(u, c) \wedge \text{CF}(u', c') \rightarrow c \neq c')$
- V. UTENTE.GESTORE  
 $\forall u \text{ UTENTE}(u) \rightarrow (\exists c \text{ CF}(u, c) \leftrightarrow \text{GESTORE}(u, \text{TRUE}))$
- V. UTENTE.IDENTIFICATO  
 $\forall u \text{ UTENTE}(u) \rightarrow (\text{IDENTIFICATO}(u, \text{TRUE}) \rightarrow \text{GESTORE}(u, \text{TRUE}))$
- V. STRUTTURA.GESTORE  
 $\forall a, u \text{ STRUTTURA}(a) \wedge \text{GESTSTRUT}(a, u) \rightarrow \text{GESTORE}(u, \text{TRUE})$
- V. PRENOTAZIONE.ACCETTATA - O - RIFIUTATA  
 $\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists i \text{ ISTAcc}(p, i) \rightarrow \exists i' \text{ ISTRif}(p, i'))$
- V. SISTPREN. DISGIUNTE  
 $\forall p, p', a, i, i', x, x' \text{ SISTEMAZIONE}(a) \wedge \text{SISTPREN}(p, a) \wedge \text{SISTPREN}(p', a) \wedge p \neq p' \wedge (\exists z, z' \text{ ISTAcc}(p, z) \wedge \text{ISTAcc}(p', z')) \wedge (\exists c, c' \text{ ISTANC}(p, c) \wedge \text{INSTANC}(p', c')) \wedge \text{INIZIO}(p, i) \wedge \text{INIZIO}(p', i') \wedge \text{FINE}(p, x) \wedge \text{FINE}(p', x') \rightarrow \nexists t \text{ DATAORA}(t) \wedge (i \leq t \leq x) \wedge (i' \leq t \leq x')$



**Risposta alla Domanda 6 (segue)**

**Domanda 7 (30 minuti; 60 minuti al massimo)** Proseguire la fase di progettazione logica della base di dati producendo lo schema relazionale della base dati e i relativi vincoli a partire dallo schema ER ristrutturato.

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

<b>1 Relazione</b>	<u>UTENTE</u>	(nome)	Derivante da: entità   relationship (cerchiare)
Attributi	<u>EMAIL</u>	<u>NOME</u>	<u>COGNOME</u> <u>TELEFONO</u> <u>GESTORE</u> <u>CF*</u> <u>IDENTIFICATO*</u>
Domini	<u>EMAIL</u>	<u>STR</u>	<u>STR</u> <u>TELEFONO</u> <u>BOOLEAN</u> <u>coefs</u> <u>BOOLEAN</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):

$\text{ENN: GESTORE} = \text{TRUE} \Leftrightarrow \text{CF} \neq \text{NULL}$

$\text{ENN: IDENTIFICATO} \neq \text{NULL} \rightarrow \text{GESTORE} = \text{TRUE}$

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

<b>2 Relazione</b>	<u>STRUTTURA</u>	(nome)	Derivante da: entità   relationship (cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>	<u>INIZI</u> <u>STELLE</u> <u>TIPOLOGIA</u> <u>CITTÀ</u>
Domini	<u>INTEGER</u>	<u>STR</u>	<u>INDIRIZZO</u> <u>INT-1-7</u> <u>STR</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):

$\text{SERIAL: ID}$   $\text{FK: (CITTÀ)} \text{ REF CRM}(\text{ID})$   $\text{INC: (ID)} \subseteq \text{SISTEMAZIONE(STRUT)}$   
 $\text{FK: (TIPOLOGIA)} \text{ REF TIPOLOGIA(NOME)}$   $\text{INC: (ID)} \subseteq \text{GESTSTRUT(STRUT)}$

La relazione accorda le relazioni che implementano le seguenti relationship: .... TIPOSTRUT, CITSTRUT .....

<b>3 Relazione</b>	<u>GESTSTRUT</u>	(nome)	Derivante da: entità   relationship (cerchiare)
Attributi	<u>UTENTE</u>	<u>STATUT</u>	
Domini	<u>EMAIL</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):

$\text{FK: (UTENTE)} \text{ REF UTENTE(EMAIL)}$   
 $\text{FK: (STATUT)} \text{ REF STRUTURA(ID)}$

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

<b>4 Relazione</b>	<u>LIVELLOFEDELTA'</u>	(nome)	Derivante da: entità   relationship (cerchiare)
Attributi	<u>NOME</u>	<u>PROMIN</u>	
Domini	<u>STR</u>	<u>INT-62</u>	

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

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

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

<b>5 Relazione</b>	<u>LIVSTRUT</u>	(nome)	Derivante da: entità   relationship (cerchiare)
Attributi	<u>STRUT</u>	<u>LIV</u>	<u>SCONTO</u>
Domini	<u>INTEGER</u>	<u>STR</u>	<u>REAL-0-1</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):

$\text{FK: (LIV)} \text{ REF LIVELLOFEDELTA'(NOME)}$   
 $\text{FK: (STRUT)} \text{ REF STRUTURA(ID)}$

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

<b>6 Relazione ...<u>TIMOLOSLIA</u>.....</b> (nome)	Derivante da: <b>entità</b>   relationship (cerchiare)
Attributi   <u>ID</u>   <u>NOME</u>   <u>STR</u>	
Domini   <u>STR</u>	

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

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

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

<b>7 Relazione .....<u>CITTÀ</u>'.....</b> (nome)	Derivante da: <b>entità</b>   relationship (cerchiare)
Attributi   <u>ID</u>   <u>NOME</u>   <u>NAZ</u>	
Domini   <u>INTEGER</u>   <u>STR</u>   <u>STR</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):

SERIAL: ID  
FK: (NAZ) REF NAZIONE (NONE)

La relazione accorda le relazioni che implementano le seguenti relationship: ..... NazCit

<b>8 Relazione .....<u>NAZIONE</u>.....</b> (nome)	Derivante da: <b>entità</b>   relationship (cerchiare)
Attributi   <u>NOME</u>	
Domini   <u>STR</u>	

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

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

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

<b>9 Relazione ...<u>SISTENAZIONE</u>..</b> (nome)	Derivante da: <b>entità</b>   relationship (cerchiare)
Attributi   <u>ID</u>   <u>NOME</u>   <u>POSTILETTI</u>   <u>STRUT</u>	
Domini   <u>INTEGER</u>   <u>STR</u>   <u>INT-62</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):

SERIAL: ID  
FK: (STRUT) REF STRUTTURA (ID)

La relazione accorda le relazioni che implementano le seguenti relationship: ..... STRUT-SIST

<b>10 Relazione ...<u>TARIFFE</u>.....</b> (nome)	Derivante da: <b>entità</b>   relationship (cerchiare)
Attributi   <u>ID</u>   <u>DATA</u>   <u>NUMOcc</u>   <u>PREZZO</u>   <u>SIST</u>	
Domini   <u>INTEGER</u>   <u>DATA</u>   <u>INT-62</u>   <u>DENARO</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):

SERIAL: ID  
CHIAVE: (SIST, DATA, NUMOcc)  
FK: (SIST) REF SISTENAZIONE (ID)

La relazione accorda le relazioni che implementano le seguenti relationship: ..... SIST-TAR

11 Relazione <u>METODOPAGAMENTO</u> (nome)	Derivante da: entità   relationship (cerchiare)
Attributi   <u>NOME</u>	
Domini   STR	

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: .....

12 Relazione <u>SISTPREN</u> (nome)	Derivante da: entità   relationship (cerchiare)
Attributi   <u>PREN</u>   <u>SIST</u>	
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: (PREN) REF METODOPAGAMENTO (ID)

FK: (SIST) REF SISTEMAZIONE (ID)

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

13 Relazione <u>PRENOTAZIONE</u> (nome)	Derivante da: entità   relationship (cerchiare)
Attributi   <u>ID</u>   <u>UTENTE</u>   <u>ISTPREN</u>   <u>INIZIO</u>   <u>FINE</u>   <u>STRUT</u>   <u>METPAS</u>	
Domini   INTEGER   STR   TIMESTAMP   TIMESTAMP   TIMESTAMP   INTEGER   STR	

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):

SERIAL: ID

FK: (UTENTE) REF UTENTE (EMAIL)

FK: (STRUT) REF STRUTTURA (ID)

FK: (METPAS) REF METODOPAGAMENTO (NONE)

ENN: (ISTPREN) REF SISTPREN (PREN)

ENN: ISTPREN < INIZIO & INIZIO < FINE

INC: (ID) ⊆ SISTPREN (PREN)

UTPREN, METPEN

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

14 Relazione <u>PRENOTAZIONE</u> (nome)	Derivante da: entità   relationship (cerchiare)
Attributi   <u>ISTACC</u> *   <u>ISTRIP</u> *   <u>ISTRANC</u> *	
Domini   TIMESTAMP   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):

ENN: ISTACC ≠ NULL → ISTPREN < ISTACC & ISTACC < INIZIO

ENN: ISTRANC ≠ NULL → ISTPREN < ISTRANC & ISTRANC < INIZIO

ENN: ISTRIP ≠ NULL → ISTPREN < ISTRIP & ISTRIP < INIZIO

ENN: ISTACC ≠ NULL → ISTRIP = NULL

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: .....

<b>16</b>	<b>Relazione</b> ..... (nome)	Derivante da: <b>entità</b>   <b>relationship</b> (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: .....

<b>17</b>	<b>Relazione</b> ..... (nome)	Derivante da: <b>entità</b>   <b>relationship</b> (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: .....

<b>18</b>	<b>Relazione</b> ..... (nome)	Derivante da: <b>entità</b>   <b>relationship</b> (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: .....

<b>19</b>	<b>Relazione</b> ..... (nome)	Derivante da: <b>entità</b>   <b>relationship</b> (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: .....

<b>20</b>	<b>Relazione</b> ..... (nome)	Derivante da: <b>entità</b>   <b>relationship</b> (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 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. SISTPREN. BISGIUNTE

- INSERIMENTO IN SISTPREN
- PRE-OPERAZIONE

$ISVALID = (\text{NOT EXISTS} (\text{SELECT * FROM PRENOTAZIONE } p, \text{PRENOTAZIONE } p', \text{SISTPREN } m \text{ WHERE } p.ID = m.PREN \text{ AND } p'.ID = NEW.PREN \text{ AND } p.ID \neq p'.ID \text{ AND } p.ISTACC \text{ IS NOT NULL AND } p'.ISTACC \text{ IS NOT NULL AND } p.ISTANC \text{ IS NULL AND } p'.ISTANC \text{ IS NULL AND } m.SIST = NEW.SIST \text{ AND } (p.INIZIO, p.FINE) OVERLAAS (p'.INIZIO, p'.FINE)))$

IF ISVALID

COMMIT

ELSE

GENERA ERRORE

REVERT

#### • T. UTENTE. CF UNICO

- INSERIMENTO E MODIFICA IN UTENTE
- PRE-OPERAZIONE

$ISVALID = (\text{NOT EXISTS} (\text{SELECT * FROM UTENTE } u \text{ WHERE NEW.CF} = u.CF \text{ AND NEW.CF IS NOT NULL AND } u.CF IS NOT NULL))$

IF ISVALID

COMMIT

ELSE

GENERA ERRORE

REVERT

#### • T. STRUTTURA. GESTORE

- INSERIMENTO IN GESTSTRUT
- PRE-OPERAZIONE

$ISVALID = (\text{EXISTS} (\text{SELECT * FROM UTENTE } u \text{ WHERE } u.EMAIL = NEW.UTENTE \text{ AND } u.GESTORE = TRUE))$

IF ISVALID

COMMIT

ELSE

GENERA ERRORE

REVERT

## Risposta alla Domanda 7 (segue)

## • T. SISTEMAZIONE. POSTILETTO

## • INSERIMENTO IN SISTPREN

## • PRE-OPERAZIONE

$$\text{isValid} = (\exists \text{NEW} \in \text{SISTEMAZIONE} \text{ WHERE NEW.STRUT} = A.ID \text{ AND NEW.NHOCC} \leq A.POSTILETTO)$$

IF isValid  
    COMMIT

ELSE  
    GENERA ERRORE  
    REPEAT

## • T. SISTEMAZIONE. STRUTURA

## • INSERIMENTO IN SISTPREN

## • PRE-OPERAZIONE

$$\text{isValid} = (\text{NOT EXISTS} (\text{SELECT * FROM SISTPREN } M, \text{SISTEMAZIONE } A, \text{SISTEMAZIONE } A' \text{ WHERE } M.PREN} = \text{NEW.PREN} \text{ AND } A.ID} = M.SIST \text{ AND } A'.ID} = \text{NEW.SIST} \text{ AND } A.STRUT} < A'.STRUT))$$

IF isValid  
    COMMIT

ELSE  
    GENERA ERRORE  
    REPEAT

**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

*CALCOLA\_LIV\_UTENTE (u: EMAIL): STR*

*TMAPTABLE (l: STR, n: INT-GEZ)*

*FK: (l) REF LIVELOFDELTA' (NOME)*

*INSERT INTO TMAPTABLE (l, n)*

*SELECT l.NOME, COUNT(p.ID) AS n*

*FROM LIVELOFDELTA' l, PRENOTAZIONE p*

*WHERE p.UTENTE = :u AND p.ISTACE IS NOT NULL*

*AND p.ISTANC IS NULL AND*

*EXTRACT(YEAR FROM p.INIZIO) >= CURRENT\_YEAR - 1 AND*

*EXTRACT(YEAR FROM p.FINE) <= CURRENT\_YEAR*

*GROUP BY l.NOME*

*HAVING n >= l.MEMLIN*

*Q = (SELECT t.l AS livr*

*FROM TMAPTABLE t WHERE t.n = (SELECT MAX(t'.n) FROM TMAPTABLE t'))*

*RETURN Q.livr*

## Risposta alla Domanda 8 (segue)

RICENASISTVALIDE ( $u: EMAIL, c: INTEGER, n: INT-1-9, i: DATE, f: DATE, p: DECIMAL, n: INT-62$ ): INSERIRE ( $c: INTEGER$ )

IF  $:f < :i$   
GENERA ERRORE

TMAPDATE ( $d: DATE$ )

WITH RECURSIVE GENDATE ( $d$ ) AS (

VALUES (:i)

UNION ALL

SELECT  $d + '1 DAY':: INTERVAL$   
FROM GENDATE WHERE  $d < :f$

INSERT INTO TMAPDATE ( $d$ )

SELECT  $d$  FROM GENDATE

ISERROR = (EXISTS (SELECT  $d$  FROM TMAPDATE  
EXCEPT  
SELECT t.DATA FROM TARIFFA t  
WHERE t.NUMOCC = :n AND t.SIST = :a))

IF ISERROR

GENERA ERRORE

Q = (SELECT A.ID

FROM SISTEMAZIONE A LEFT OUTER JOIN STRUTLIV AL ON A.ID = AL.STRUT,  
STRUTTURA t', TIPOLOGIA t, TARIFFA t'

WHERE A.STRUT = A'.ID AND A'.CITTA = :c AND A'.TIPOLOGIA = :t'  
AND A'.STELLE >= :n AND t'.SIST = A.ID AND t'.NUMOCC = :n  
AND :i <= t'.DATA AND t'.DATA <= :f AND A.MOSTILETTO >= :n  
AND AL.LIV = DB.CALCOLALIVELLO (UTENTE (:u))

AND NOT EXISTS (SELECT \* FROM PRENOTAZIONE P, SISTPREN AP  
WHERE P.ISTACC IS NOT NULL AND P.IST(ANC) IS NULL  
AND A.ID = AP.PREN AND A.ID = AL.SIST  
AND (:i, :f) OVERLAES (P.INIZIO, P.FINE))

GROUP BY A.ID  
HAVING (SUM(t'.PREZZO) \* (CASE WHEN AL.SCONTI IS NOT NULL THEN 1 - AL.SCONTI  
ELSE 1) <= :p)

RETURN Q

[CONTINUA A  
MINUTA 33]

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.]

*NUM\_GIORNI\_PENI (d: INTEGER) : /NSIEME (< m: INTEGER, n: INT\_627)*

*+ TMP\_MESI (m: INTEGER)*

*WITH RECURSIVE GEN\_MESI (n) AS (*

*VALUES (1)*

*UNION ALL*

*SELECT n+1 FROM GEN\_MESI WHERE n < 12)*

*INSERT INTO TMP\_MESI (m)*

*SELECT n FROM GEN\_MESI*

*+ TMP\_DATE (d: DATE)*

*WITH RECURSIVE GEN\_DATE (d) AS (*

*SELECT DATE\_TRUNC('YEAR', CURRENT\_TIMESTAMP)*

*UNION ALL*

*SELECT d + '1 DAY'::INTERVAL*

*FROM GEN\_DATE*

*WHERE d < DATE\_TRUNC('YEAR', CURRENT\_TIMESTAMP + '1 YEAR'::INTERVAL) - '1 DAY'::INTERVAL)*

*INSERT INTO TMP\_DATE (d)*

*SELECT d FROM GEN\_DATE*

*Q = (SELECT f.m, COUNT(t'.d)*

*FROM TMP\_MESI f, TMP\_DATE t'*

*WHERE NOT EXISTS (SELECT \* FROM SISTEMAZIONE g' WHERE g'.STRUT = :g*

*EXCEPT*

*SELECT \* FROM SISTEMAZIONE g', SISTREN gr, PRENOTAZIONE p*

*WHERE g'.STRUT = :g AND gr.STRUT = g'.ID AND*

*gr.PREN = p.ID AND p.ISTACC IS NOT NULL AND*

*p.ISTANC IS NULL AND p.INIZIO <= t'.d AND*

*t'.d <= p.FINE))*

*RETURN Q*

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

### [USE-CASE STRUMENTI PRENOTAZIONI]

ACCETTA PREN ( $P: \text{PRENOTAZIONE}$ ): ACCETTATA

RIFIUTA PREN ( $P: \text{PRENOTAZIONE}$ ): RIFIUTATA

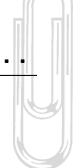
### [USE-CASE STRUMENTI SCONTI]

ASSIGNA SCONTO ( $s: \text{STRUTTURA}, l: \text{LIVELLO}, n: \text{REALE}(0,1) (0,1)$ )

### [USE-CASE STATISTICHE]

■ NUMBIORNIPIENI ( $s: \text{STRUTTURA}$ ):  $(m: \text{INTERO}[1, +\infty], n: \text{INTERO} \geq 0) (0, n)$

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



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