



Basi di Dati, Modulo 2

Sapienza Università di Roma

Facoltà di Ing. dell'Informazione, Informatica e Statistica

Laurea in Informatica

Prof. Toni Mancini

<http://tmancini.di.uniroma1.it>

Progetto 20220506 (P.20220506)

RistoBook

Versione 2022-05-06

2

Specifiche dei Requisiti

Il sistema deve permettere ai **clienti** di effettuare **prenotazioni** presso i **ristoranti** iscritti, usufruendo, eventualmente, di **promozioni**. I **ristoratori**, invece, possono iscriversi per registrare i propri ristoranti e gestire le loro prenotazioni e le loro promozioni.

Dei **clienti** interessa conoscere il **nome** e l'**indirizzo e-mail**, mentre dei **ristoranti** interessa il **nome**, la **partita IVA** (una stringa numerica), l'**indirizzo**, la **città** e l'**insieme di tipologie di cucina** offerte (scelte da una lista tenuta sotto controllo dallo staff di RistoBook).

I **clienti** possono **prenotare** presso un ristorante specificando il **giorno**, l'**ora** e il **numero di commensali**. Le prenotazioni dei clienti devono essere **confermate** (o **rifiutate**) dal personale incaricato dei rispettivi ristoranti, che devono poter accedere ad RistoBook tramite una interfaccia dedicata.

Uno dei punti di forza del modello di business di RistoBook è la possibilità per i ristoranti di offrire e pubblicizzare scontistiche (dette **promozioni**). In particolare, i ristoratori devono poter definire una promozione specificando **una percentuale di sconto** sulle **prenotazioni consumate in un certo periodo di tempo**. Tali promozioni sono valide per al **massimo un certo numero di coperti al giorno**.

Ad esempio, un ristorante potrebbe definire la seguente promozione: sconto del 20% sulle prenotazioni dalle 20 alle 22 da martedì 15 giugno 2022 a venerdì 18 giugno 2022, valido per al massimo 10 coperti al giorno. Un altro ristorante potrebbe definire invece una promozione del tipo: sconto del 25% sulle prenotazioni dalle 18 alle 19 di tutti i martedì e giovedì dal 1 ottobre al 31 dicembre 2022.

Al momento della **prenotazione**, il cliente può scegliere **una delle promozioni ancora disponibili** (anche nessuna, se lo desidera).

Il sistema deve permettere ai ristoratori e ai clienti di gestire lo stato delle prenotazioni. In particolare, quando una prenotazione viene creata, questa è nello stato **"pendente"**. Il ristoratore può scegliere se **accettarla o rifiutarla**. Quando il cliente usufruisce effettivamente della prenotazione, questa viene contrassegnata come **"completata"**. Se, invece, il



cliente non dovesse presentarsi al ristorante, il ristoratore contrasseggerà la prenotazione come “non utilizzata”. Inoltre, in ogni momento (prima del giorno ed ora prenotati) i clienti possono **annullare** le proprie prenotazioni, anche se già accettate. Infine, in caso di tutto esaurito (o per altre ragioni, ad es. giorni di chiusura), il responsabile di un ristorante deve poter **chiudere le prenotazioni** per un **certo lasso di tempo** (ad es., una certa data e fascia oraria, o un’intera settimana): da quel momento in poi, RistoBook **non consentirà più ai clienti di richiedere prenotazioni in quel lasso di tempo** (a meno che il ristoratore non le **riapra**).

Il sistema deve offrire, oltre quelle già descritte, le seguenti funzionalità ai suoi attori:

- Il sistema deve permettere ristoratori di RistoBook di calcolare alcune statistiche di utilizzo delle loro promozioni. In particolare, dato un periodo di tempo, RistoBook deve permettere calcolare, **per ogni promozione del ristorante considerato, il numero medio di clienti al giorno** (dove la media è calcolata sui giorni di validità della promozione) che ha utilizzato quella promozione in una prenotazione.
- **Data una città x , un insieme di tipologie di cucina C , un tasso di sconto minimo s ed una data d ,** i clienti devono poter trovare quali sono i ristoranti nella città x che offrono almeno una delle tipologie di cucina in C e prevedono promozioni con sconti di tasso almeno s nella data d ancora utilizzabili per il numero di coperti che sono interessati a prenotare.



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: 02/09/13

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

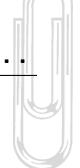
Firma di un membro della Commissione per
avvenuta identificazione:
.....

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

CLIENTE

- NOME, COGNOME
- E-MAIL

RISTORANTE

- NOME
- PARTITA IVA
- INDIRIZZO
- CITTÀ
- TECNOLOGIE
- CUCINA
- SOCIETÀ

RISTORATORE

- CODICE FISCALE

PRENOTAZIONE

- ISTANTE
- GIORNO PRENOTAZIONE
- NUMERO COMMENSALI
- PENDENTE, ACCETTATA
O RIFIUTATA
- COMPLETATA, NON
UTILIZZATA
- ANNULLATA
- TAVOLO

TAVOLO

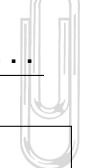
- NUM

CHIUSURA

- RISTORANTE
- INIZIO
- FINE
- PUÒ ESSERE
ANNULLATA

PROMOZIONE

- SCONTO
- PERIODO VALIDITÀ'
- NUM PERSONE
AL GIORNO



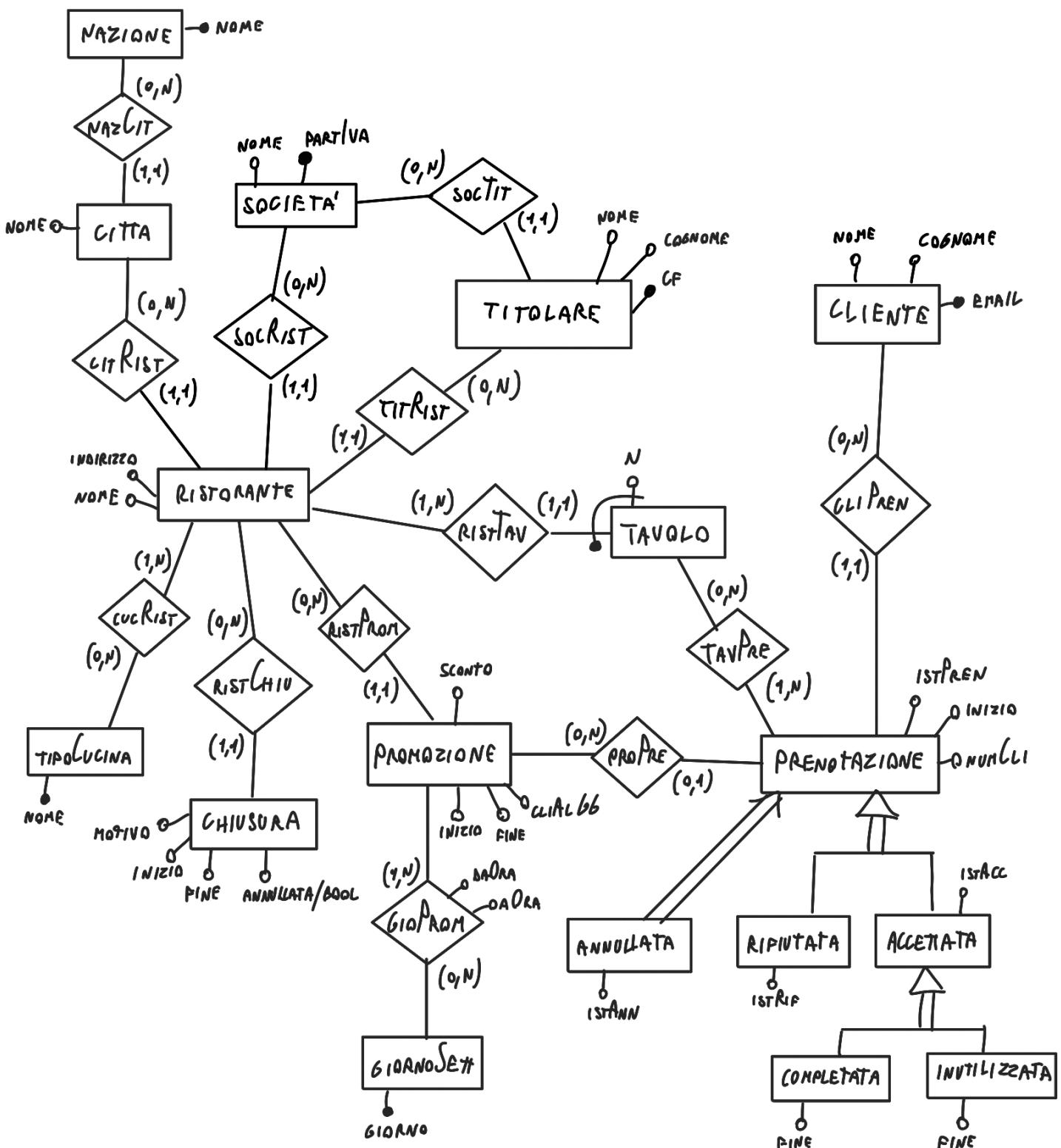
Risposta alla Domanda 1 (segue)

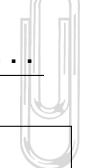
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma ER concettuale per l'applicazione, il dizionario dei dati ed eventuali vincoli esterni.

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.





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: Entità Relationship (cerchiare)</p> <p>Nome: NAZIONE.....</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;">moltepl. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">NAME</td><td style="text-align: center;">STRINGA</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)	NAME	STRINGA		<p>[2] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: CITTÀ.....</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;">moltepl. (*)</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">NAME</td><td style="text-align: center;">STRINGA</td><td></td></tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltepl. (*)	NAME	STRINGA	
attributo	dominio	moltepl. (*)											
NAME	STRINGA												
attributo	dominio	moltepl. (*)											
NAME	STRINGA												

<p>3] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: RISTORANTE</p> <table border="1" data-bbox="95 224 794 426"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td>PART/VA</td> <td>PART/VA</td> <td></td> </tr> <tr> <td>NOME</td> <td>STRINGA</td> <td></td> </tr> <tr> <td>INDRIZZO</td> <td>INDRIZZO</td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>T · V. RISTORANTE. SOCIETÀ</p> $\forall r, t, s \text{ RISTORANTE}(r) \wedge \text{TITRIST}(r, t) \wedge \text{sOCIETÀ}(r, s) \rightarrow \text{sOCITIR}(t, s)$	attributo	dominio	moltep. (*)	PART/VA	PART/VA		NOME	STRINGA		INDRIZZO	INDRIZZO		<p>5] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: TIPOCUCINA</p> <table border="1" data-bbox="794 224 1511 426"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td>NOME</td> <td>STRINGA</td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	NOME	STRINGA	
attributo	dominio	moltep. (*)																	
PART/VA	PART/VA																		
NOME	STRINGA																		
INDRIZZO	INDRIZZO																		
attributo	dominio	moltep. (*)																	
NOME	STRINGA																		

<p>4] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: TITOLARE</p> <table border="1" data-bbox="84 1280 794 1459"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td>CP</td> <td>CDFA</td> <td></td> </tr> <tr> <td>NOME</td> <td>STRINGA</td> <td></td> </tr> <tr> <td>COGNOME</td> <td>STRINGA</td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p>	attributo	dominio	moltep. (*)	CP	CDFA		NOME	STRINGA		COGNOME	STRINGA		<p>6] Tipo: Entità Relationship (cerchiare)</p> <p>Nome: CLIENTE</p> <table border="1" data-bbox="794 1280 1511 1459"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td>EMAIL</td> <td>STRINGA</td> <td></td> </tr> <tr> <td>NOME</td> <td>STRINGA</td> <td></td> </tr> <tr> <td>COGNOME</td> <td>STRINGA</td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>R · V. CLIENTE. PRENDISIUNTE</p> $\begin{aligned} & \forall c, p, p', i, i' \text{ CLIENTE}(c) \wedge \text{ACCETTATA}(p) \wedge \\ & \text{ACCETTATA}(p') \wedge \neg \text{ANNULLATA}(p) \wedge \neg \text{ANNULLATA}(p') \wedge \\ & p \neq p' \wedge \text{CLIPRE}(p, c) \wedge \text{CLIPRE}(p', c) \wedge \text{INIZIA}(p, i) \\ & \wedge \text{INIZIA}(p', i') \rightarrow \exists t \text{ DATAORA}(t) \wedge \\ & (i \leq t \wedge (\forall x \text{ PINE}(p, x) \rightarrow t \leq x)) \wedge \\ & (i' \leq t \wedge (\forall x' \text{ PINE}(p', x') \rightarrow t \leq x')) \end{aligned}$	attributo	dominio	moltep. (*)	EMAIL	STRINGA		NOME	STRINGA		COGNOME	STRINGA	
attributo	dominio	moltep. (*)																							
CP	CDFA																								
NOME	STRINGA																								
COGNOME	STRINGA																								
attributo	dominio	moltep. (*)																							
EMAIL	STRINGA																								
NOME	STRINGA																								
COGNOME	STRINGA																								

7	Tipo: Entità Relationship (cerchiare)	
Nome: TAVOLO	
attributo	dominio	moltep. (*)
N	INTEROZO	
(*) solo se diversa da (1,1)		
Vincoli:		
R	<ul style="list-style-type: none"> • V. TAVOLO. PREN DISGIUNTE $\forall t, p, p', i, i' \quad \text{TAVOLO}(t) \wedge \text{ACCEMMATA}(p) \wedge \text{ACCEMMATA}(p') \wedge \neg \text{ANNULLATA}(p) \wedge \neg \text{ANNULLATA}(p') \wedge p \neq p' \wedge \text{TAVPRE}(p, t) \wedge \text{TAVPRE}(p', t) \wedge \text{INIZIO}(p, i) \wedge \text{INIZIO}(p', i') \rightarrow \nexists t' \text{ DATAORA}(t') \wedge (i \leq t \wedge (\forall t' \text{ PINE}(p, t) \rightarrow t \leq t')) \wedge (i' \leq t \wedge (\forall t' \text{ PINE}(p', t') \rightarrow t' \leq t))$	

9	Tipo: Entità Relationship (cerchiare)	
Nome: PROMOZIONE	
attributo	dominio	moltep. (*)
	INIZIO	DATA
	FINE	DATA
	CLI AL66	CLI AL66
	SCONTO	REALE [0,1]
(*) solo se diversa da (1,1)		
Vincoli:		
R	<ul style="list-style-type: none"> • V. PROMOZIONE. RISTORANTE TAVOLO $\forall p, p', t, z \quad \text{PROMOZIONE}(p) \wedge \text{PROPRE}(p', p) \wedge \text{RISTTAV}(p, z) \wedge \text{TAVPRE}(p', t) \rightarrow \text{RISTTAV}(t, z)$	
	<ul style="list-style-type: none"> • V. PROMOZIONE. CLI AL66 $\forall p, n \quad \text{PROMOZIONE}(p) \wedge \text{CLI AL66}(p, n) \wedge \text{DATA}(d) \rightarrow \text{NUM(CLIENTI, PROM}(p, d) \leq n$	
X	<ul style="list-style-type: none"> • V. PROMOZIONE. CONTINUITÀ $\forall p, i, i' \quad \text{PROMOZIONE}(p) \wedge \text{INIZIO}(p, i) \wedge \text{FINE}(p, i') \rightarrow i < i'$	

8	Tipo: Entità Relationship (cerchiare)	
Nome: PRENOTAZIONE	
attributo	dominio	moltep. (*)
NUMLI	INTEROZO	
IST PRE	DATAORA	
INIZIO	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		
R	<ul style="list-style-type: none"> • V. PRENOTAZIONE. PROMOZIONE $\forall p, p', i, d, j, \alpha_i \quad \text{PRENOTAZIONE}(p) \wedge \text{PROPRE}(p, p') \wedge \text{INIZIO}(p, i) \wedge \text{ORA}(i, \alpha_i) \wedge \text{DATA}(i, d) \wedge j = \text{OTTENIBIORNOSETT}(d) \rightarrow \text{GIOVAN}(p', j) \wedge (\exists d_i, d_f \quad \text{INIZIO}(p', d_i) \wedge \text{FINE}(p', d_f) \wedge (i \leq d \wedge d \leq d_f)) \wedge \text{DATA}(p', j, \alpha_d) \wedge \text{DATA}(p', j, \alpha_\alpha) \wedge (\alpha_d \leq \alpha_i \wedge \alpha_i \leq \alpha_\alpha)$	
X	<ul style="list-style-type: none"> • V. PRENOTAZIONE. CONTINUITÀ $\forall p, i, i' \quad \text{PRENOTAZIONE}(p) \wedge \text{ISTPRE}(p, i) \wedge \text{INIZIO}(p, i') \rightarrow i < i'$	

10	Tipo: Entità Relationship (cerchiare)	
Nome:	... ANNULLATA	
attributo	dominio	moltep. (*)
IST ANN	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		
R	<ul style="list-style-type: none"> • V. ANNULLATA. NONCOMPLETATA $\forall p \quad \text{ANNULLATA}(p) \rightarrow \neg (\text{COMPLETATA}(p) \vee \text{INUTILIZZATA}(p))$	

11	Tipo: Entità Relationship (cerchiare)	
Nome:	RIFIUTATA	
attributo	dominio	moltep. (*)
ISTRIP	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		
✓.V. RIFIUTATA . CONTINUITÀ'		
$\forall p, i, i' \text{ PRENOTAZIONE}(p) \wedge \text{ISTRIP}(p, i) \wedge \text{INIZIO}(p, i')$ $\rightarrow (\forall i'' \text{ISTRIP}(p, i'') \rightarrow i < i'' \wedge i'' < i')$		

13	Tipo: Entità Relationship (cerchiare)	
Nome:	ACCETTATA	
attributo	dominio	moltep. (*)
ISTRACC	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		
✓.V. ACCETTATA . CONTINUITÀ'		
$\forall p, i, i' \text{ PRENOTAZIONE}(p) \wedge \text{ISTRIP}(p, i) \wedge \text{INIZIO}(p, i')$ $\rightarrow (\forall i'' \text{ISTRACC}(p, i'') \rightarrow i < i'' \wedge i'' < i')$		
✓.V. ACCETTATA . CONTINUITÀ' FINE		
$\forall p, i \text{ PRENOTAZIONE}(p) \wedge \text{INIZIO}(p, i')$ $\rightarrow (\forall f \text{FINE}(p, f) \rightarrow i < f)$		

12	Tipo: Entità Relationship (cerchiare)	
Nome:	COMPLETATA	
attributo	dominio	moltep. (*)
FINE	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		

14	Tipo: Entità Relationship (cerchiare)	
Nome:	UTILIZZATA	
attributo	dominio	moltep. (*)
FINE	DATAORA	
(*) solo se diversa da (1,1)		
Vincoli:		

15	Tipo: Entità Relationship (cerchiare)
	Nome: CHIUSURA
attributo	dominio
INIZIO	DATAORA
FINE	DATAORA
ANNULLATA	BOOLEANO
MOTIVO	STRINGA

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

Vincoli:

- X** • V. CHIUSURA. CONTINUITÀ
 $\forall c, i, f \ CHIUSURA(c) \wedge INIZIO(c, i) \wedge FINE(c, f) \rightarrow i < f$
- T** • V. CHIUSURA. NO PRENOTAZIONI
 $\forall c, r, i, f \ CHIUSURA(c) \wedge ANNULLATA(c, FALSE) \wedge RISCHIUSURA(c, r) \wedge INIZIO(c, i) \wedge FINE(c, f) \rightarrow \nexists p, t, i' \ PRENOTAZIONE(p) \wedge TAVPRE(p, t) \wedge RISRTAV(t, r) \wedge INIZIO(p, i') \wedge i \leq i' \wedge i' \leq f$

17	Tipo: Entità Relationship (cerchiare)
	Nome: GIORNO.SEST
attributo	dominio
GIORNO	GIORNO

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

Vincoli:

16	Tipo: Entità Relationship (cerchiare)
	Nome: SOCIETA
attributo	dominio
PARTIVA	PARTIVA
NOME	STRINGA

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

Vincoli:

18	Tipo: Entità Relationship (cerchiare)
	Nome: GIO PROM
attributo	dominio
DAORA	ORA
AORA	ORA

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

Vincoli:

- X** • V. GIO PROM. CONTINUITÀ
 $\forall p, g, o, o' \ PROMOZIONE(p) \wedge GIO_PROM(p, g) \wedge DAORA(p, g, o) \wedge AORA(p, g, o') \rightarrow o < o'$

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

DOMINIO PART/VA: STRINGA ALFANUMERICA SECONDO STANDARD

DOMINIO codfis: STRINGA 16 CARATTERI SECONDO STANDARD

DOMINIO EMAIL: STRINGA SECONDO STANDARD

DOMINIO GIORNO: { 'LUN', 'MAR', 'MER', 'GIO', 'VER', 'SAB', 'DOM' }

DOMINIO INDIRIZZO:

VIA: STRINGA

CIVICO: INTERO>0

CAP: INTERO>0

$\text{NUMLINETIPROM}(\text{P: PROMOZIONE}, \text{d: DATA}): \text{INTERO}>0$

PRE:

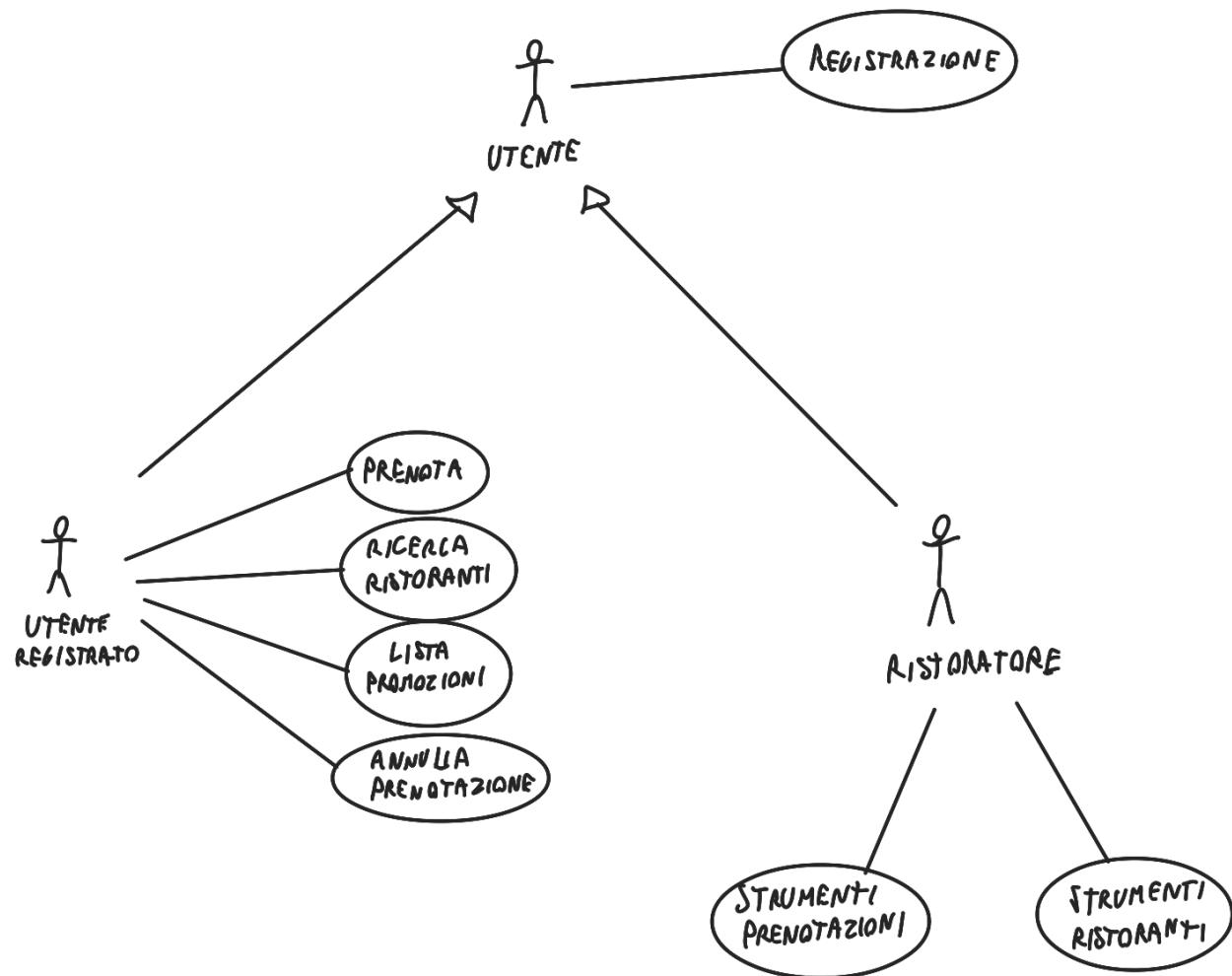
POST:

$$\text{P} = \left\{ (\text{P}', \text{n}) \mid \text{PREMNOTAZIONE}(\text{P}') \wedge \text{PROPRE}(\text{P}', \text{P}) \wedge \text{NUMCLI}(\text{P}', \text{n}) \wedge (\exists i \text{ INIZIA}(\text{P}', i) \wedge \text{DATA}(i, \text{d})) \right\}$$

$$\text{RESULT} = \sum_{(\text{P}', \text{n}) \in \text{P}} \text{n}$$

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: REGISTRAZIONE (nome use-case)

Operazioni dello use-case:

REGISTRA CLIENTE (cf: CCFIS, n: STRINGA, c: STRINGA, e: EMAIL) : CLIENTE

REGISTRA RISTORATORE (cf: CCFIS, n: STRINGA, c: STRINGA) : RISTORATORE

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

Operazioni dello use-case:

PRENOTA (c: CLIENTE, r: RISTORANTE, n: INTERVALLO, i: DATAORA, p: PIANOZIONE(0,1)) : PRENOTAZIONE

3 Specifica use-case: RICERCA RISTORANTI (nome use-case)

Operazioni dello use-case:

RICERCA RISTORANTI (x: Utente, t: TIPOCUINA(1,N), a: REALE[0,1], d: DATA, n: INTERVALLO) : RISTORANTE(0,N)

4 Specifica use-case: LISTA PROMOZIONI (nome use-case)

Operazioni dello use-case:

LISTA PROMOZIONI (r: RISTORANTE) : PROMOZIONE(0,N)

5 Specifica use-case: ANNULLA PRENOTAZIONE (nome use-case)

Operazioni dello use-case:

ANNULLA PRENOTAZIONE (p: PRENOTAZIONE)

6 Specifica use-case: STRUMENTI PRENOTAZIONE (nome use-case)

Operazioni dello use-case:

ACCETTA PRENOTAZIONE (p: PRENOTAZIONE)

RIFIUTA PRENOTAZIONE (p: PRENOTAZIONE)

CONFERMA PRENOTAZIONE (p: ACCETTATA)

PRENOTAZIONE UTILIZZATA (p: ACCETTATA)

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

Operazioni dello use-case:

AGGIUNGI PROMOZIONE (r: RISTORANTE, i: DATAORA, f: DATAORA, a: REALE [0,1], b: (g: Giornate, od: ORA, or: ORA) (1,N)) : PROMOZIONE

REGISTRA CHIUSURA (r: RISTORANTE, i: DATAORA, f: DATAORA, m: STRINGA) : CHIUSURA

ANNULLA CHIUSURA (c: CHIUSURA)

NUMERI MEDI CLIENTI PROMOZIONI (r: RISTORANTE, i: DATA, f: DATA) : (p: PROMOZIONE, m: REALE [0,1]) (0,N)

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

$\text{NUMMEDIOCLIENTIPRENOTAZIONI}(\tau: \text{RISTORANTE}, i: \text{DATA}, f: \text{DATA}): (\rho: \text{PRENOTAZIONE}, m: \text{REAL}^{>0}) (0, n)$

PRE: $i < f$

POST:

$$\rho = \left\{ (\rho, m) \mid \begin{array}{l} \text{PRENOTAZIONE}(\rho) \wedge \text{RISTPROM}(\rho, \tau) \wedge \left(\exists i, f \text{ INIZIO}(\rho, i) \wedge \text{FINE}(\rho, f) \right) \\ \wedge i \leq i' \wedge i' \leq f \wedge m = \text{NUMMEDIOCLIENTIPRENOTAZIONE}(\rho) \end{array} \right\}$$

RESULT = ρ

$\text{NUMMEDIOCLIENTIPRENOTAZIONE}(\rho: \text{PRENOTAZIONE}): \text{REAL}^{>0}$

PRE:

POST:

Esistono i, f tali che $\text{INIZIO}(\rho, i) \wedge \text{FINE}(\rho, f)$

$$\rho = \left\{ (\rho', n) \mid \begin{array}{l} \text{PRENOTAZIONE}(\rho') \wedge \text{PROPRE}(\rho', \rho) \wedge \text{NUMLI}(\rho', n) \\ \left(\rho = \emptyset \rightarrow \text{RESULT} = 0 \right) \wedge \left(\rho \neq \emptyset \rightarrow \text{RESULT} = \frac{\sum n}{f - i + 1} \right) \end{array} \right\}$$

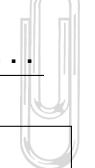
$\text{RICERCARISTORANTI}(x: \text{CITTÀ}, c: \text{TITOLUCINA}(1, n), \alpha: \text{REAL}[0, 1], d: \text{DATA}, n: \text{INTERO} > 0): \text{RISTORANTE}(0, n)$

PRE:

POST:

$$R = \left\{ x \mid \begin{array}{l} \text{RISTORANTE}(x) \wedge \text{CITRIST}(x, x) \wedge \left(\exists c, \rho, \alpha', i, f, n' \text{ CUCRIST}(x, c) \wedge c \in C \wedge \right. \right. \\ \left. \left. \text{RISTPROM}(\rho, x) \wedge \text{INIZIO}(\rho, i) \wedge \text{FINE}(\rho, f) \wedge i \leq d \wedge d \leq f \wedge \text{SCONTO}(\rho, \alpha') \right. \right. \\ \left. \left. \wedge \alpha \leq \alpha' \wedge \text{CUTALGB}(\rho, n') \wedge (n' - \text{NUMLI}(\rho, d)) \geq n \right) \right\}$$

RESULT = R



Risposta alla Domanda 5 (segue)

2 Progettazione della base dati e delle funzionalità

Domanda 6 (20 minuti; 30 minuti al massimo) Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema 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.

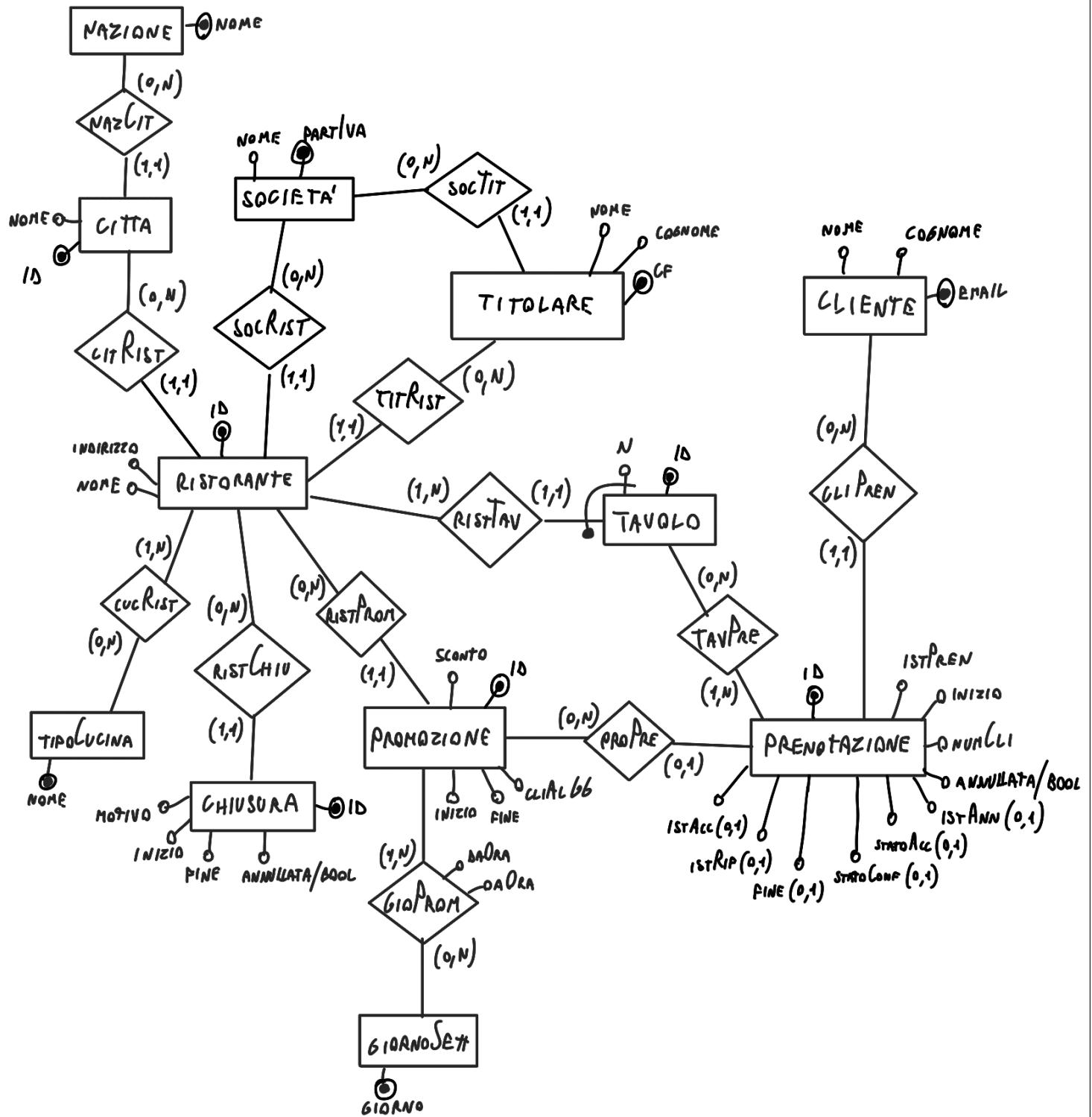
DBMS da utilizzare **POSTGRESQL**

Corrispondenza tra domini concettuali e domini supportati dal DBMS

- CREATE DOMAIN INT_GEZ AS INTEGER CHECK(VALUE >= 0)
- CREATE DOMAIN INT_GZ AS INTEGER CHECK(VALUE > 0)
- CREATE DOMAIN REAL_GEZ AS REAL CHECK(VALUE >= 0)
- CREATE DOMAIN REAL_GEZ 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 PARLIVA AS STR CHECK (isValidPV(value))
- CREATE DOMAIN CODFIS AS CHAR(16) CHECK (isValidCF(value))
- CREATE DOMAIN EMAIL AS STR CHECK (isValidEmail(value))
- CREATE TYPE GIORNA AS ENUM('LUN', 'MAR', 'MER', 'GIO', 'VER', 'SAB', 'DOM')
- CREATE TYPE INDIRIZZO AS (

 VIA STR,
 CIVICO INT_GZ,
 CAP INT_GZ
)
- CREATE TYPE TIPO_ACC ('ACC', 'RIF')
- CREATE TYPE TIPO_COMP ('COMP', 'INUT')

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. PRENOTAZIONE. ANNULLATA

$$\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists i \text{ ISRANN}(p,i) \Rightarrow \text{ANNULLATA}(p, \text{TRUE}))$$

- V. PRENOTAZIONE. ACCETTATA

$$\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists i \text{ ISRACC}(p,i) \Rightarrow \exists n \text{ STATOACC}(p,n) \wedge n = 'acc')$$

- V. PRENOTAZIONE. RIFIUTATA

$$\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists i \text{ ISRIF}(p,i) \Rightarrow \exists n \text{ STATOACC}(p,n) \wedge n = 'rif')$$

- V. PRENOTAZIONE. COMPLETATA / INUTILIZZATA

$$\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists n \text{ STATOCONF}(p,n) \rightarrow \exists n' \text{ STATOACC}(p,n') \wedge n' = 'acc')$$

- V. PRENOTAZIONE. FINE

$$\forall p \text{ PRENOTAZIONE}(p) \rightarrow (\exists f \text{ FINE}(p,f) \Leftrightarrow \exists n \text{ STATOCONF}(p,n))$$

- V. ANNULLATA. NON COMPLETATA

$$\nexists p \text{ PRENOTAZIONE}(p) \wedge \text{ANNULLATA}(p, \text{TRUE}) \rightarrow \nexists n \text{ STATOCONF}(p,n)$$

Risposta alla Domanda 6 (segue)

• V. CLIENTE . PREN DISGIUNTE

$$\forall c, p, p', i, i' \text{ CLIENTE}(c) \wedge \text{PRENOTAZIONE}(p) \wedge \text{PRENOTAZIONE}(p') \wedge \text{ANNULLATA}(p, \text{FALSE}) \wedge \text{ANNULLATA}(p', \text{FALSE}) \wedge (\exists_{\alpha} \text{STATOAcc}(p, \alpha) \wedge \alpha = 'ACC') \wedge (\exists_{\alpha'} \text{STATOAcc}(p', \alpha') \wedge \alpha' = 'ACC') \wedge p \neq p' \wedge \text{CLI_PRE}(p, c) \wedge \text{CLI_PRE}(p', c) \wedge \text{INIZIO}(p, i) \wedge \text{INIZIO}(p', i') \rightarrow \nexists t \text{ DATAORA}(t) \wedge (i \leq t \wedge (\forall t' \text{ PINE}(p, t) \rightarrow t \leq t')) \wedge (i' \leq t \wedge (\forall t' \text{ PINE}(p', t') \rightarrow t \leq t'))$$

• V. TAVOLO . PREN DISGIUNTE

$$\forall t, p, p', i, i' \text{ TAVOLO}(t) \wedge \text{PRENOTAZIONE}(p) \wedge \text{PRENOTAZIONE}(p') \wedge \text{ANNULLATA}(p, \text{FALSE}) \wedge \text{ANNULLATA}(p', \text{FALSE}) \wedge (\exists_{\alpha} \text{STATOAcc}(p, \alpha) \wedge \alpha = 'ACC') \wedge (\exists_{\alpha'} \text{STATOAcc}(p', \alpha') \wedge \alpha' = 'ACC') \wedge p \neq p' \wedge \text{TAV_PRE}(p, t) \wedge \text{TAV_PRE}(p', t) \wedge \text{INIZIO}(p, i) \wedge \text{INIZIO}(p', i') \rightarrow \nexists t \text{ DATAORA}(t) \wedge (i \leq t \wedge (\forall t' \text{ PINE}(p, t) \rightarrow t \leq t')) \wedge (i' \leq t \wedge (\forall t' \text{ PINE}(p', t') \rightarrow t \leq t'))$$

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 .. NAZIONE ..	(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:

2 Relazione CITTÀ ..	(nome)	Derivante da: entità relationship (cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>
Domini	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: (NAZ) REF NAZIONE(NOME)

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

3 Relazione ... RISTORANTE ..	(nome)	Derivante da: entità relationship (cerchiare)
Attributi	<u>ID</u>	<u>NOME</u>
Domini	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: (SOCIETÀ) REF SOCIETÀ(PARTIVA)

INC : (ID) ⊆ CUCRIST(RIST)

FK: (CITTÀ) REF CITTÀ(ID)

FK: (TITOLARE) REF TITOLARE(CF)

La relazione accorda le relazioni che implementano le seguenti relationship: ... CIRIST, TITLST, SOCIST

4 Relazione ... SOCIETÀ ..	(nome)	Derivante da: entità relationship (cerchiare)
Attributi	<u>PARTIVA</u>	<u>NOME</u>
Domini	PARTIVA	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:

5 Relazione ... TITOLARE ..	(nome)	Derivante da: entità relationship (cerchiare)
Attributi	<u>CF</u>	<u>NOME</u>
Domini	CODFIS	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):

FK: (SOCIETÀ) REF SOCIETÀ(PARTIVA)

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

6 Relazione ...<u>TIPOCUINA</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:

7 Relazione ...<u>CUCRIST</u>... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>RIST</u> <u>CUC</u>	
Domini 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):

FK: (RIST)REF RISTANTE (ID)

FK: (CUC)REF TIPOCUINA (NOME)

La relazione accorda le relazioni che implementano le seguenti relationship:

8 Relazione ...<u>CHIUSURA</u>... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> RIST INIZIO FINE MOTIVO ANNULLATA	
Domini INTEGER INTEGER TIMESTAMP TIMESTAMP STR BOOLEAN	

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

ENN: INIZIO < FINE

FK: (RIST)REF RISTANTE (ID)

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

9 Relazione ...<u>PROMOZIONE</u>... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> RIST INIZIO FINE SCONTO	
Domini INTEGER INTEGER DATE DATE REAL-0..1	

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

ENN: INIZIO < FINE

FK: (RIST)REF RISTANTE (ID)

INC: (ID) ⊑ GIORNO(PROM)

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

10 Relazione ...<u>GIORNOSET</u>... (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>GIORNO</u>	
Domini GIORNO	

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:

<input checked="" type="checkbox"/> Relazione ... <u>PROM</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>PROM</u> <u>610</u> DAORA AORA	
Domini INTEGER GIORNO TIME TIME	

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: (PROM) REF PROMOZIONE(1D) ENN: DAORA < AORA
FK: (610) REF GIORNOSETT(GIORNO)

La relazione accorda le relazioni che implementano le seguenti relationship:

<input checked="" type="checkbox"/> Relazione ... <u>TAVOLI</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> RIST N	
Domini INTEGER INTEGER INT-62	

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: (RIST, N)
FK: (RIST) REF RISTORANTE(1D)

La relazione accorda le relazioni che implementano le seguenti relationship:

<input checked="" type="checkbox"/> Relazione ... <u>CLIENTE</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>EMAIL</u> Nome COGNOME	
Domini EMAIL STR 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:

<input checked="" type="checkbox"/> Relazione ... <u>PRENOTAZIONE</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> RIST ISTPREN INIZIO NUMCLI ANNULLATA ISTANN* PROM*	
Domini INTEGER INTEGER TIMESTAMP TIMESTAMP INT-62 BOOLEAN TIMESTAMP INTEGER	

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

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

FK: (RIST) REF RISTORANTE(1D) SERIAL:ID ENN: ANNNULLATA = TRUE \Leftrightarrow ISTANN = NULL
FK: (PROM) REF PROMOZIONE(1D) ENN: ISTPREN < INIZIO ENN: ANNNULLATA = TRUE \rightarrow STATOCONF \neq NULL
INC: (ID) \subseteq TAVPREN(PREN) ENN: ISTPREN < INIZIO ENN: ANNNULLATA = TRUE \rightarrow STATOCONF \neq NULL

La relazione accorda le relazioni che implementano le seguenti relationship:

<input checked="" type="checkbox"/> Relazione ... <u>PRENOTAZIONE</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi STATOACC* STATOCONF* ISTACC* ISTRIF* FINE*	
Domini STATO_ACC STATO_CONF 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: STATOACC = 'ACC' \Leftrightarrow ISTACC \neq NULL ENN: STATOCONF \neq NULL \rightarrow FINE \neq NULL
ENN: STATOACC = 'RIF' \Leftrightarrow ISTRIF \neq NULL ENN: ISTACC \neq NULL \rightarrow ISTPREN < ISTALL < INIZIO
ENN: STATOCONF \neq NULL \rightarrow STATOACC = 'ACC' ENN: ISTRIF \neq NULL \rightarrow ISTPREN < ISTRIPC < INIZIO

La relazione accorda le relazioni che implementano le seguenti relationship:

16	Relazione ... <u>TAVPREN</u> (nome)	Derivante da: entità relationship (cerchiare)
Attributi	<u>PREN</u> <u>TAV</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):

FK: (PREN) REF PRENOTAZIONE(ID)

Fk: (TAV) REF TAVOLA(ID)

La relazione accorda le relazioni che implementano le seguenti relationship:

17	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:

18	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:

19	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:

20	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 ennuple); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

• T. RISTORANTE. SOCIETA'

- INSERIMENTO IN RISTORANTE
- PRE-OPERAZIONE

$$\text{ISVALID} = \left(\text{EXISTS} \left(\text{SELECT * FROM TITOLARE } t \right. \right. \\ \left. \left. \text{WHERE NEW.TITOLARE} = t.CF \text{ AND} \right. \right. \\ \left. \left. \text{NEW.SOCIETA'} = t.SOC \right) \right)$$

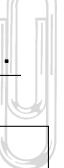
```
IF ISVALID
    COMMIT
ELSE
    GENERA ERRORE
    REVERT
```

• V. CHIUSURA NO PRENOTAZIONI

- INSERIMENTO IN CHIUSURA

$$\text{ISERROR} = \left(\text{EXISTS} \left(\text{SELECT *} \right. \right. \\ \left. \left. \text{FROM PRENOTAZIONE p, TAVOLO t, TAVPRE tp} \right. \right. \\ \left. \left. \text{WHERE NEW.ANNULLATA} = \text{FALSE AND} \right. \right. \\ \left. \left. \text{NEW.INIZIO} \leq p.INIZIO \text{ AND } p.FINE} \geq \text{NEW.FINE} \right. \right. \\ \left. \left. \text{AND tp.PREN} = p.ID \text{ AND tp.TAV} = t.ID \text{ AND} \right. \right. \\ \left. \left. t.RIST} = \text{NEW.RIST} \right) \right)$$

```
IF ISERROR
    GENERA ERRORE
    REVERT
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 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

NUMMEDIOCLIENTIPROMOZIONI (τ : INTEGER, i : DATE, f : DATE) : INSIEME (p : INTEGER, m : REAL-6Z)

IF $f \leq i$
GENERAL ERRORE

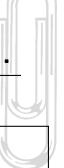
$Q = \left(\text{SELECT } p.ID, (\text{SUM}(p'.NUM(CL)) / (\text{EXTRACT(DAY FROM } p.f - p.i) + 1)) \right.$
FROM PROMOZIONE p, PRENOTAZIONE p'
WHERE p.RIST = : τ AND i <= p.INIZIO AND p.FINE <= : f AND p'.PROM = p.ID
GROUP BY p.ID)

RETURN Q

RICERCARISTORANTI (x : INTEGER, C : INSIEME (c : STR), a : REAL-0-1, d : DATE, n : INT-6Z) : INSIEME (τ : INTEGER)

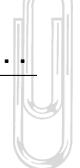
$Q = \left(\text{SELECT } \tau.ID \right.$
FROM PROMOZIONE p, RISTORANTE r, CUCRIST cr, PRENOTAZIONE p'
WHERE p.RIST = $\tau.ID$ AND r.CITTÀ = : x AND cr.RIST = $\tau.ID$ AND
cr.CUC IN (:C) AND p.INIZIO <= : d AND : d <= p.FINE
AND p.SCONTO >= : a AND p'.PROM = p.ID
AND EXTRACT(DATE FROM p'.INIZIO) = : d
GROUP BY p.ID
HAVING (p.CLIENTELB - SUM(p'.NUM(CL))) >= : n)

RETURN Q



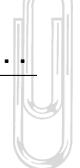
Risposta alla Domanda 8 (segue)

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



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

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



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