



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.ER** – Modulo risposte prova scritta

Dati dello studente e dell'esame

Cognome e nome: Matricola:

Data:

Corso di laurea e canale di appartenenza:

- ☐ Laurea in Informatica, canale 1 (A-L, Prof. G. Perelli)
- ☐ 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

1. Cliente

1.1 nome

1.2 email

1.3 ristoranti prenotati

2. Ristorante

2.1 nome

2.2 P.iva {id}

2.3 indirizzo

2.3.1 via

2.3.2 civico

2.3.3 città

2.4 cucine offerte (Esempio: romana)

2.5 prenotazioni

2.5.1 cliente

2.5.2 data e orario

2.5.3 numero commensali

2.5.4 stato (confermata, rifiutata, pendente, non utilizzata)

2.5.5 sconto usufruito

2.6 sconti

2.6.1 percentuale

2.6.2 intervallo temporale [Data-Data]

2.6.3 coperti max al giorno

2.6.4 intervallo orario [Ora-Ora]

2.6.5 giorni validità (lu, mar, mer..., dom)

2.7 chiusura pren

2.7.1 inizio DataOra

2.7.2 fine DataOra

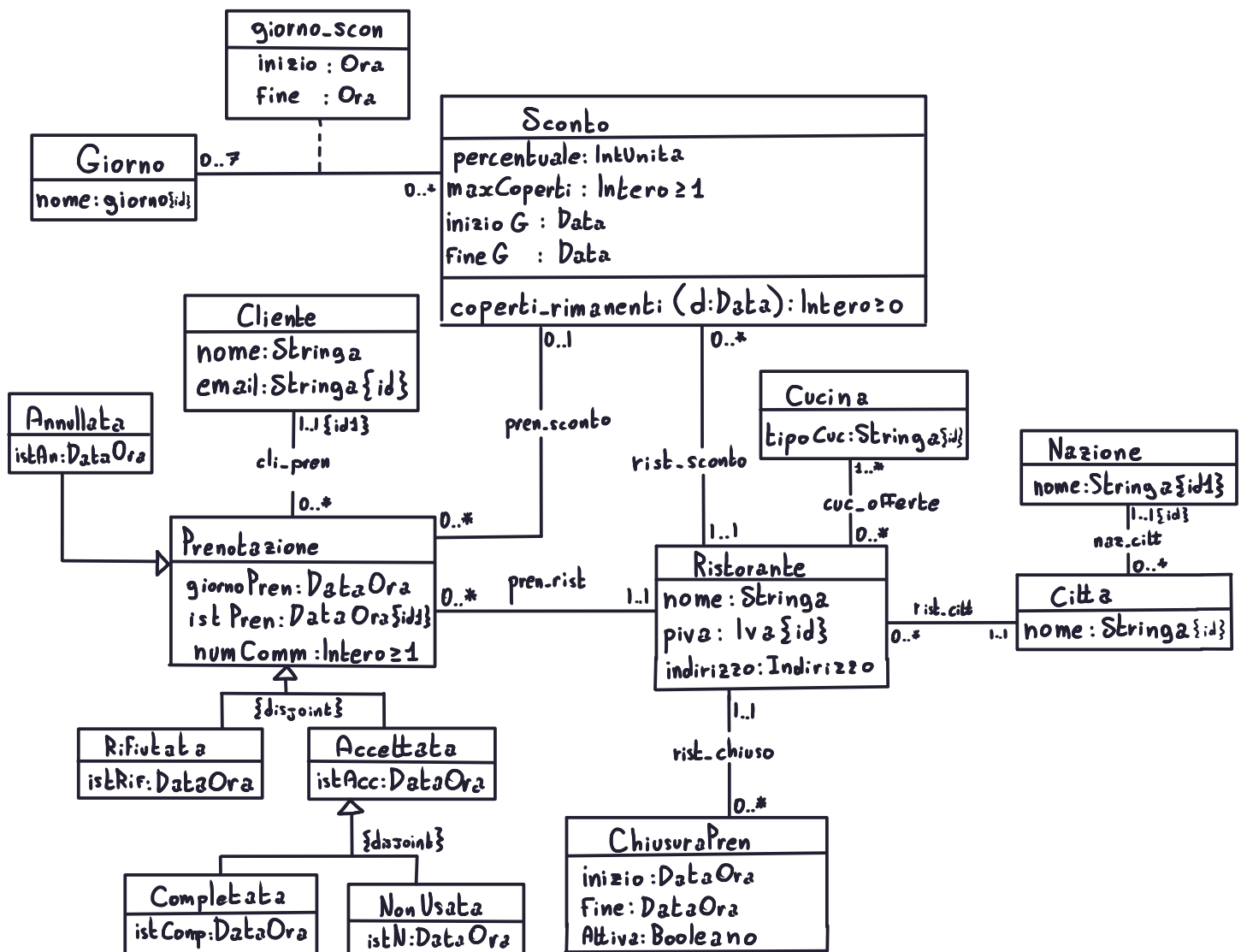
Risposta alla Domanda 1 (segue)

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

Prodotto 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

1

Tipo: Entità | Relationship (cerchiare)

Nome: ..giorno_scon.....

attributo	dominio	moltepl. (*)
inizio	Ora	
fine	Ora	

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

Vincoli:

•

L'ora di inizio di uno sconto NON può essere maggiore all'ora di fine.

$\forall g, s \quad \text{giorno_scon}(g, s) \rightarrow$
 $\neg [\exists i, f \quad i \geq f \wedge \text{inizio}(g, s, i) \wedge \text{fine}(g, s, i)]$

2

Tipo: Entità | Relationship (cerchiare)

Nome: ..Giorno.....

attributo	dominio	moltepl. (*)
nome	giorno	

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

Vincoli:

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

attributo	dominio	moltepl. (*)
percentuale	IntUnità	
maxCoperti	Intero ≥ 1	
inizio G	Data	
fine G	Data	

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

Vincoli:

• Giorno di inizio di uno sconto NON può essere maggiore al giorno di fine.

$$[\forall s, i, f \text{ Sconto}(s) \wedge \text{inizioG}(s, i) \wedge \text{fineG}(s, f) \rightarrow i < f]$$

• Una prenotazione che usufruisce di uno sconto, cade in un giorno ed orario coperti dallo sconto.

$$\forall p, s, g, gPren, iS, fS, iSo, fSo, gPrenD, gPrenO, gSett, gSettPren$$
5 Tipo: **Entità** | Relationship (cerchiare)Nome: ... **Sconto**

attributo	dominio	moltepl. (*)
Coperti_rimanti(d:Data): Intero ≥ 0		

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

Vincoli:

$$[\text{pren_sconto}(p, s) \wedge \text{giorno_sco}(g, s) \wedge \text{giornoPren}(p, gPren) \wedge \text{inizioG}(s, iS) \wedge \text{fineG}(s, fS) \wedge \text{inizio}(g, s, iSo) \wedge \text{fine}(g, s, fSo) \wedge \text{Data}(gPren, gPrenD) \wedge \text{Ora}(gPren, gPrenO) \wedge \text{giorno_sco}(s, gSett) \wedge \text{giorno}(gSett, gSettPren)]$$

$$\rightarrow [iSo \leq gPrenO \leq fSo \wedge iS \leq gPrenD \leq fS \wedge \text{GiornoSettimana}(gPren, gSettPren)]$$
4 Tipo: **Entità** | Relationship (cerchiare)Nome: ... **Cucina**

attributo	dominio	moltepl. (*)
tipoCuc	Stringa	

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

Vincoli:

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

attributo	dominio	moltepl. (*)
istAn	DataOra	

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

Vincoli:

• L'annullamento avviene dopo la richiesta o accettazione

$$[\forall p, iA, iP, dP \text{ Annullata}(p) \wedge \text{istAn}(p, iA) \wedge \text{giornoPren}(p, dP) \wedge \text{Prenotazione}(p) \wedge \text{istPren}(p, iP)] \rightarrow iA > iP \wedge iA < dP$$

$$[\forall p, iA, iP \text{ Annullata}(p) \wedge \text{istAn}(p, iA) \wedge \text{Accettata}(p) \wedge \text{istAcc}(p, iP)] \rightarrow iA > iP$$

7 Tipo: **Entità** | Relationship (cerchiare)Nome: **Annullata**

attributo	dominio	moltepl. (*)

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

Vincoli:

- Si può annullare una prenotazione solo se accettata o pendente

$$\forall p \text{ Annullata}(p) \rightarrow [Prenotazione(P) \vee Accettata(p)]$$

- Non si può annullare una prenotazione completata, non usata o rifiutata

$$\forall p \text{ Annullata}(p) \rightarrow [\neg Rifiutata(p) \wedge \neg Completata(p) \wedge \neg NonUsata(p)]$$

9 Tipo: **Entità** | Relationship (cerchiare)Nome: .. **Rifiutata**

attributo	dominio	moltepl. (*)

istRif	DataOra	
--------	---------	--

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

Vincoli:

- istante rifiuto > istante richiesta

$$[\forall p, iR, iP \text{ istRif}(p, iR) \wedge \text{istPren}(p, iP)] \rightarrow iR > iP$$

8 Tipo: **Entità** | Relationship (cerchiare)Nome: .. **Accettata**

attributo	dominio	moltepl. (*)

istAcc	DataOra	
--------	---------	--

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

Vincoli:

- istante accett. > istante richiesta

$$[\forall p, iA, iP \text{ istAcc}(p, iA) \wedge \text{istPren}(p, iP)] \rightarrow iA > iP$$

10 Tipo: **Entità** | Relationship (cerchiare)Nome: ... **Completata**

attributo	dominio	moltepl. (*)

istComp	DataOra	
---------	---------	--

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

Vincoli:

- istante comp > istante accett.

$$[\forall p, iA, iC \text{ istAcc}(p, iA) \wedge \text{istComp}(p, iC)] \rightarrow iC > iA$$

11 Tipo: **Entità** | Relationship (cerchiare)Nome: **NonUsata**

attributo	dominio	moltepl. (*)
istN	DataOra	

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

Vincoli:

• istante di non uso > istante accettazione

$$[\forall p, iA, iN \text{ istAcc}(p, iA) \wedge \text{istN}(p, iN)] \rightarrow iN > iA$$

13 Tipo: **Entità** | Relationship (cerchiare)Nome: **Cliente**

attributo	dominio	moltepl. (*)
nome	Stringa	
email	Stringa	

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

Vincoli:

12 Tipo: **Entità** | Relationship (cerchiare)Nome: **Prenotazione**

attributo	dominio	moltepl. (*)
giornoPren	DataOra	
istPren	DataOra	
numComm	Intero ≥ 1	

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

Vincoli:

• La prenotazione avviene prima del giorno relativo ad essa

$$[\forall p, i, d \text{ Pren}(p) \wedge \text{GiornoPren}(p, d) \wedge \text{istPren}(p, i)] \rightarrow i < d$$

14 Tipo: **Entità** | Relationship (cerchiare)Nome: **Prenotazione**

attributo	dominio	moltepl. (*)
-----------	---------	--------------

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

Vincoli:

• Una prenotazione con m commensali non può usufruire di uno sconto che ha n maxCoperti se $m > n$.

$$[\forall p, s, n, m \text{ pren-sconto}(p, s) \wedge \text{maxCoperti}(s, n) \wedge \text{numComm}(p, m)] \rightarrow m \leq n$$

<div>15 Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Prenotazione</div> <table> <tr> <th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr> <tr> <td></td><td></td><td></td></tr> </table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli: generalizz. del vincolo precedente</div> <ul style="list-style-type: none"> Il numero totale di commensali che usufruiscono di uno sconto in un det. giorno, deve essere \leq al numero di max Coperti. per ogni sconto s e giorno g coperto da s <p>$\forall p, s \quad \text{pren_sconto}(p, s)$</p> <p>da riscrivere, ed aggiungi istante per calcolare coperti_rimanenti()</p>	attributo	dominio	moltepl. (*)				<div>17 Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Chiusura Pren</div> <table> <tr> <th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr> <tr> <td>inizio</td><td>DataOra</td><td></td></tr> <tr> <td>fine</td><td>DataOra</td><td></td></tr> <tr> <td>Attiva</td><td>Booleano</td><td></td></tr> </table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	inizio	DataOra		fine	DataOra		Attiva	Booleano	
attributo	dominio	moltepl. (*)																	
attributo	dominio	moltepl. (*)																	
inizio	DataOra																		
fine	DataOra																		
Attiva	Booleano																		
<div>16 Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Nazione</div> <table> <tr> <th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr> <tr> <td>nome</td><td>Stringa</td><td></td></tr> </table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	nome	Stringa		<div>18 Tipo: Entità Relationship (cerchiare)</div> <div>Nome: Città</div> <table> <tr> <th>attributo</th><th>dominio</th><th>moltepl. (*)</th></tr> <tr> <td>nome</td><td>Stringa</td><td></td></tr> </table> <div>(*) solo se diversa da (1,1)</div> <div>Vincoli:</div>	attributo	dominio	moltepl. (*)	nome	Stringa							
attributo	dominio	moltepl. (*)																	
nome	Stringa																		
attributo	dominio	moltepl. (*)																	
nome	Stringa																		

15 Tipo: Entità | Relationship (cerchiare)

Nome: .. Ristorante

attributo	dominio	moltepl. (*)
nome	Stringa	
piva	lva	
indirizzo	Indirizzo	

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

Vincoli:

• Una pren. può usare sconti per un ristorante, solo se quello sconto è relativo al rist. in questione.

$$\forall p,s,r \text{ pren-sconto}(p,s) \wedge \text{pren-rist}(p,r) \rightarrow \text{rist-sconto}(r,s)$$

17 Tipo: Entità | Relationship (cerchiare)

Nome: .. ChiusuraPren

attributo	dominio	moltepl. (*)

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

Vincoli:

$$\forall c,i,F [\text{ChiusuraPren}(c) \wedge \text{inizio}(c,i) \wedge \text{fine}(c,F)] \rightarrow i < F$$

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

$\text{IntUnita} = 0 \leq \text{Reale} \leq 1$

$\text{Indirizzo} : (\text{via} : \text{Stringa}, \text{civico} : \text{Intero} \geq 1)$

$\text{Iva} : \text{Stringa alfanumerica secondo standard}$

$\text{giorno} = \{\text{lun}, \text{mar}, \text{mer}, \text{gio}, \text{ven}, \text{sab}, \text{dom}\}$

$\text{intGiornaliero} : (\text{g} : \text{giorno}, \text{i} : \text{Ora}, \text{f} : \text{Ora})$

Specifica classe Sconto

OPERAZIONI

$\text{coperti_rimanenti}(\text{d} : \text{Data}) : \text{Intero} \geq 0$

- pre-condizioni:
- post-condizioni: non modifica il livello estensionale degli oggetti.

$$M = \left\{ (p, m) \mid \begin{array}{l} \text{pren-sconto}(\text{this}, p) \wedge \exists dp \text{ giornoPren}(p, dp) \wedge \text{Accettata}(p) \\ \wedge \text{numComm}(p, m) \wedge \exists D \text{ Data}(dp, d) \wedge \neg \text{Annullata}(p) \end{array} \right\}$$

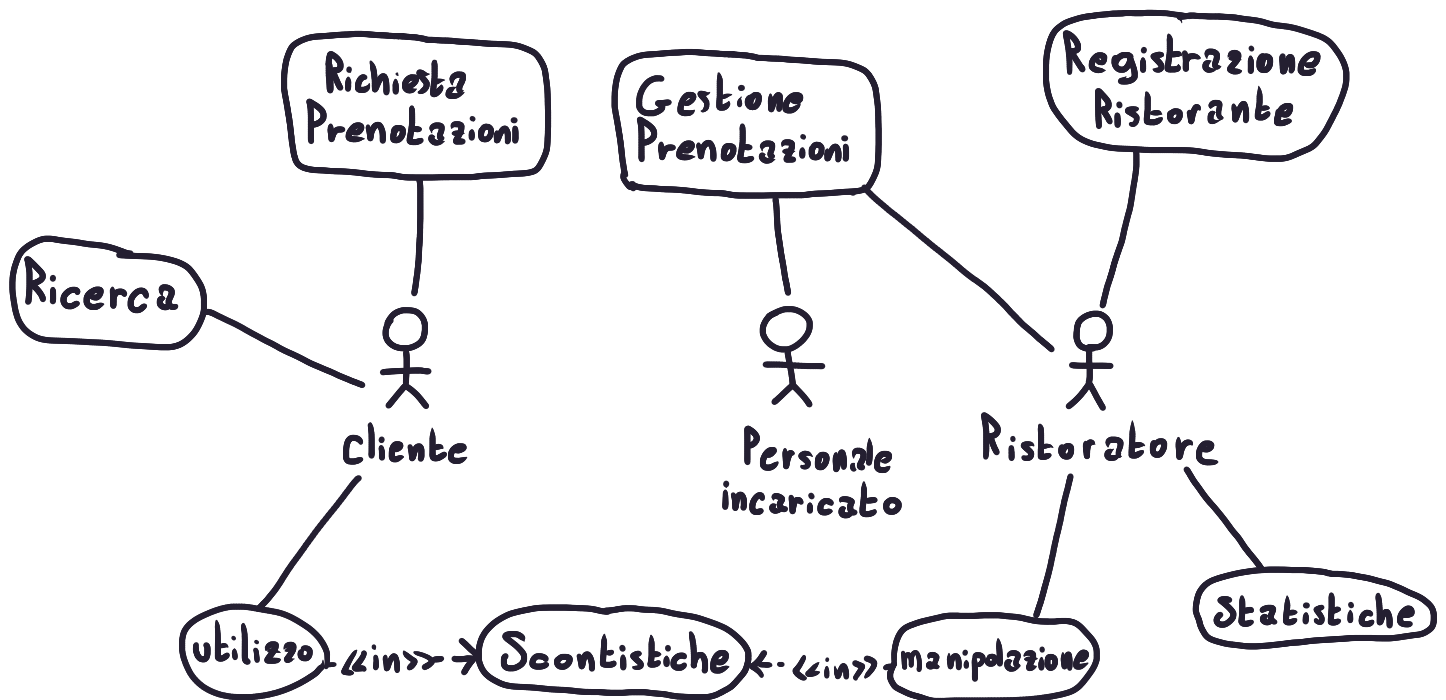
Sia maxC tale che $\text{maxCoperti}(\text{this}, \text{maxC})$

$$\text{result} = \text{maxC} - \sum_{(p, m) \in M} m$$

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: Richiesta..... (nome use-case)

Operazioni dello use-case:

$\text{prenota}(g:\text{DataOra}, r:\text{Ristorante}, n:\text{Intero}): \text{Prenotazione}$

$\text{Annulla-pren}(p:\text{Prenotazione}): \text{Annullata}$

2 Specifica use-case: Gestione..... (nome use-case)

Operazioni dello use-case:

$\text{accetta-pren}(p:\text{Prenotazione}): \text{Accettata}$

$\text{rifiuta-pren}(p:\text{Prenotazione}): \text{Rifiutata}$

$\text{completa-pren}(p:\text{Prenotazione}): \text{Completata}$

$\text{chiudi-prenotazioni}(r:\text{Ristorante}, i:\text{DataOra}, f:\text{DataOra}): \text{ChiusuraPren}$

$\text{cambia-stato-prenotazioni-chiuse}(c:\text{ChiusuraPren}, b:\text{Booleano}): \text{ChiusuraPren}$

3 Specifica use-case: Statistiche..... (nome use-case)

Operazioni dello use-case:

$\text{calcola-stat}(r:\text{Ristorante}, s:\text{Sconto}): \text{Reale}$

$\text{calc-stat-glob}(r:\text{Ristorante}, in:\text{Data}, fin:\text{Data}): (\text{Sconto}, \text{Reale})[0..*]$

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

Operazioni dello use-case:

`applica_sconto(p:Prenotazione)`

5 Specifica use-case: Registrazione Ristorante (nome use-case)

Operazioni dello use-case:

`registra(n:String a, i:Iva, ind:Indirizzo): Ristorante`

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

Operazioni dello use-case:

`crea_sconto(r:Ristorante, p:IntUnita, m:Intero ≥ 1, inG:Data, finG:Data, intGiornaliero [1..7]): Sconto`

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

Operazioni dello use-case:

`cerca(x:Citta', c:Cucina [1..*], tasso:IntUnita, d:Data, cop:Intero ≥ 1): Ristorante [0..*]`

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

Ricerca

$\text{cerca}(x:\text{Città}, c:\text{Cucina}[1..*], \text{tasso}:\text{IntUnità}, d:\text{Data}, \text{cop}:\text{Intero} \geq 1) : \text{Ristorante}[0..*]$

• pre-cond: Sia y tale che $\text{Adesso}(y)$

Sia z tale che $\text{Data}(y, z)$

deve essere che $d \geq z$

• post-cond: non modifica il livello estensionale degli oggetti

$$R = \left\{ r \mid \begin{array}{l} \text{rist_citt}(r, x) \wedge \exists t \text{ cuc_offerite}(r, t) \wedge t \in c \wedge \\ [\exists \text{in, fin, s, v, gSett, gS, m} \quad \text{Sconto}(s) \wedge \text{inizioG}(s, \text{in}) \wedge \text{fineG}(s, \text{fin}) \\ \wedge \text{in} \leq d \leq \text{fin} \wedge \text{percentuale}(s, v) \wedge v \geq \text{tasso} \wedge \text{rist_sconto}(r, s)] \\ \text{giorno_scon}(s, \text{gSett}) \wedge \text{nome}(\text{gSett}, \text{gS}) \wedge \text{GiornoSettimana}(d, \text{gS}) \\ \wedge \text{cop_rimanenti}(s, d, m) \wedge n \leq m \end{array} \right\}$$

result=R

Risposta alla **Domanda 5** (segue)

Statistiche

$\text{calcola_stat}(r:\text{Ristorante}, s:\text{Sconto}): \text{Reale}$

• pre-cond: deve essere che $\text{rist_sconto}(r, s)$

• post-cond: $M = \{(p, m) \mid \text{pren_sconto}(p, s) \wedge \text{numComm}(m)\}$

$$\text{tot} = \sum_{(p, m) \in M} m$$

Siano i_G, F_G tali che $\text{inizioG}(s, i_G) \wedge \text{fineG}(s, F_G)$

$$GV = \left\{ g \mid \begin{array}{l} \text{Data}(g) \wedge i_G \leq g \leq F_G \wedge [\exists g_s, \text{clasG} \text{ } \text{GiornoSettimana}(g, g_s) \\ \wedge \text{giornoScon}(s, \text{clasG}) \wedge \text{nome}(\text{clasG}, g_s) \end{array} \right\}$$

$$\text{result} = \text{tot} \cdot \frac{1}{|GV|}$$

$\text{calc_stat_glob}(r:\text{Ristorante}, \text{in}:\text{Data}, \text{fin}:\text{Data}): (\text{Sconto}, \text{Reale})[0..*]$

• pre-cond: $\text{in} \leq \text{fin}$

• post-cond: $SN = \left\{ (s, n) \mid \begin{array}{l} \text{calcola_stat}(r, s, n) \wedge \text{rist_sconto}(r, s) \wedge \\ [\exists i, f \text{ } \text{inizioG}(s, i) \wedge \text{fineG}(s, f) \wedge \text{in} \leq i \leq f \leq \text{fin}] \end{array} \right\}$

$$\text{result} = SN$$

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 multivalore 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

Ora → time

Data → date

Stringa → varchar

DataOra → DateTime

Booleano → bool

Create type giorno as enum('lun','mar','mer','gio','ven','sab','dom');

Create domain IntUnita as real check(value ≥ 0 and value ≤ 1);

Create domain Int_GE2 as integer check(value ≥ 0);

Create domain Int_GEO as integer check(value ≥ 1);

Create type IntGiornaliero (g giorno, i time, f time);

Create type indirizzo (via varchar, civico Int_GEO);

Create type Iva as varchar check value ~ '[0-9]{11}';

■



Breve descrizione delle scelte effettuate durante la ristrutturazione

Vincoli esterni introdotti o modificati durante la fase di ristrutturazione

(si omettono i vincoli esterni la cui formulazione è rimasta identica a seguito della ristrutturazione)

- L'annullamento viene dopo la richiesta o accettazione

$$\forall p, iA, iP, a \quad [an_isa_pren(a, p) \wedge istAn(iA) \wedge istPren(iP)] \rightarrow iA > iP$$

$$\forall p, iA, iP, a, acc \quad [an_isa_pren(a, p) \wedge istAn(iA) \wedge acc_isa_pren(acc, p) \wedge istAcc(iP)] \rightarrow iA > iP$$

- Non si annullano prenotazioni che sono: non usate, completate o rifiutate

$$\forall p, a \quad [an_isa_pren(a, p)] \rightarrow [\neg \exists r \quad rif_isa_pren(p, r)]$$

$$\forall p, a, acc \quad [an_isa_pren(a, p) \wedge acc_isa_pren(acc, p)] \rightarrow [\neg \exists c \quad comp_isa_acc(acc, c)] \wedge [\neg \exists n \quad n_isa_acc(a, n)]$$

Vincoli {disjoint, complete} persi eliminando le generalizzazioni

- rifiutata $\Rightarrow \neg$ accettata

$$\forall p, r \quad [rif_isa_pren(p, r)] \rightarrow [\neg \exists a \quad acc_isa_pren(a, p)]$$

- completata \Rightarrow usata

$$\forall a, c \quad [comp_isa_acc(a, c)] \rightarrow [\neg \exists n \quad n_isa_acc(a, n)]$$

[continua alla pagina seguente]

Risposta alla **Domanda 6** (segue)

- istante rifiuto > istante richiesta

$$\forall p, r, iP, iR [rif_isa_pren(p, r) \wedge istRif(r, iR) \wedge istPren(p, iP)] \rightarrow iR > iP$$

- istante accett. > istante richiesta

$$\forall p, a, iP, iR [acc_isa_pren(p, a) \wedge istAcc(a, iR) \wedge istPren(p, iP)] \rightarrow iR > iP$$

- istante completamento > istante accettazione

$$\forall a, c, iA, iC [comp_isa_acc(a, c) \wedge istAcc(a, iA) \wedge istComp(c, iC)] \rightarrow iC > iA$$

- istante non uso > istante accettazione

$$\forall a, c, iA, iC [n_isa_acc(a, c) \wedge istAcc(a, iA) \wedge istN(c, iC)] \rightarrow iC > iA$$

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 .Giorno..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</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 ennuola, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

2 Relazione .Sconto..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	percentuale	maxCoperbi	inizioG	FineG	<u>id</u>	ristorante		
Domini	IntUnità	Int-GEO	Date	Date	serial	Iva		

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

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

foreign key ristorante references Ristorante(piva)
 check(inizioG < FineG)

La relazione accorpa le relazioni che implementano le seguenti relationship: rist-sconto.....

3 Relazione .giorno_scon... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>g</u>	<u>sconto</u>	inizio	Fine				
Domini	giorno	serial	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 ennuola, di dominio):

unique(g, sconto)
 foreign key g references Giorno(giorno)
 foreign key sconto references Sconto(id)
 check(inizio < fine)

La relazione accorpa le relazioni che implementano le seguenti relationship:

4 Relazione .Ristorante..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	nome	indirizzo	<u>piva</u>	citta				
Domini	varchar	Indirizzo	Iva	serial				

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

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

foreign key citta references Citta(id)

La relazione accorpa le relazioni che implementano le seguenti relationship:

5 Relazione .Citta..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	nome	nazione	<u>id</u>					
Domini	varchar	varchar	serial					

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

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

unique(nome, nazione)
 foreign key nazione references Nazione(nome)

La relazione accorpa le relazioni che implementano le seguenti relationship:

6 Relazione Nazione (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>							
Domini	<u>varchar</u>							

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

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

La relazione accorpa le relazioni che implementano le seguenti relationship:

7 Relazione Cucina (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>tipoCuc</u>							
Domini	<u>varchar</u>							

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

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

La relazione accorpa le relazioni che implementano le seguenti relationship:

8 Relazione cuc-offerte (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>cucina</u>	<u>ristorante</u>						
Domini	<u>varchar</u>	<u>Iva</u>						

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

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

ristorante occorre in Ristorante(piva)

La relazione accorpa le relazioni che implementano le seguenti relationship:

9 Relazione Cliente (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>	<u>email</u>						
Domini	<u>varchar</u>	<u>varchar</u>						

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

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

La relazione accorpa le relazioni che implementano le seguenti relationship:

10 Relazione Prenotazione (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>giornoPren</u>	<u>istPren</u>	<u>numComm</u>	<u>id</u>	<u>cliente</u>	<u>annullata*</u>	<u>rifutata*</u>	<u>accettata*</u>
Domini	<u>DateTime</u>	<u>DateTime</u>	<u>Int.GEO</u>	<u>serial</u>	<u>varchar</u>	<u>serial</u>	<u>serial</u>	<u>serial</u>

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

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

check(giornoPren > istPren)

unique(cliente, istPren)

foreign key cliente references Cliente(email)

foreign key annullata

foreign key rifiutata

foreign key accettata

references Annullata(id)

references Rifiutata(id)

references Accettata(id)

La relazione accorpa le relazioni che implementano le seguenti relationship: cli_pren, an_isz_pren, rif_isz_pren

11 Relazione Prenotazione.... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>sconto*</u>	<u>ristorante</u>						
Domini	<u>serial</u>	<u>iva</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):

unique(annullata) unique(rifiutata) unique(acceptata)

Foreign key sconto references Sconto(id)

Foreign key ristorante references Ristorante(iva)

La relazione accorpa le relazioni che implementano le seguenti relationship: ..acc-isa-pren, pren-sconto, pren-rist

12 Relazione Annullata..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>id</u>	<u>istan</u>						
Domini	<u>serial</u>	<u>DateTime</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 accorpa le relazioni che implementano le seguenti relationship:

13 Relazione Accettata..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>id</u>	<u>istAcc</u>	<u>completata*</u>	<u>nonusata*</u>				
Domini	<u>serial</u>	<u>DateTime</u>	<u>serial</u>	<u>serial</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):

unique(completata) unique(nonusata)

Foreign key completata references Completata(id)

Foreign key nonusata references NonUsata(id)

La relazione accorpa le relazioni che implementano le seguenti relationship: comp-isa-acc, n-isa-acc.....

14 Relazione Rifiutata..... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>id</u>	<u>istRif</u>						
Domini	<u>serial</u>	<u>DateTime</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 accorpa le relazioni che implementano le seguenti relationship:

15 Relazione Completata.... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>istComp</u>	<u>id</u>						
Domini	<u>DateTime</u>	<u>serial</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 accorpa le relazioni che implementano le seguenti relationship:

16 Relazione NonUsata (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>istN</u>	<u>id</u>						
Domini	<u>DateTime</u>	<u>serial</u>						

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

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

La relazione accorpa le relazioni che implementano le seguenti relationship:

17 Relazione Chiusura Pren ... (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>inizio</u>	<u>fine</u>	<u>attiva</u>	<u>id</u>	<u>ristorante</u>			
Domini	<u>DateTime</u>	<u>DateTime</u>	<u>bool</u>	<u>serial</u>	<u>lva</u>			

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

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

check(inizio < fine)

foreign key ristorante references Ristorante(piva)

La relazione accorpa le relazioni che implementano le seguenti relationship: ... rist_chiuso

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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa 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 enupla, di dominio):

La relazione accorpa 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 ennupe); (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.

- Una prenotazione che usufruisce di uno sconto, cade in un giorno ed orario coperti dallo sconto.

Trigger: Insert Prenotazione

sia id l'id della prenotazione da aggiungere

Q = Select * From

(Select g.nome

From Giorno g, Sconto s, giorno_scon gs

where s.id=id \wedge g.id=gs.g \wedge gs.sconto=id), Prenotazione

where to_char(istPren)=giorno

Error se Q=NULL

- Annullata \Rightarrow Pren oppure Accettata

Trigger: Insert Annullata

Q = SELECT *

FROM Prenotazione p, Annullata a, RiFiutata r, Accettata ac, Completata c, NonUsata n

WHERE (p.annullata=a.id AND p.rifiutata=r.id)

OR (p.annullata=a.id AND p.accettata=ac.id AND (ac.completata=c.id OR ac.nonusata=n.id));

Error se Q \neq NULL

- Una pren. può usare sconti per un ristorante, solo se quello sconto è relativo al rist. in questione.

Trigger: Insert Prenotazione

Q =

SELECT

FROM Prenotazione p, Ristorante r, Sconto s

WHERE p.ristorante=r.iva

AND p.sconto=s.id AND s.ristorante \neq r.iva;

ERROR se Q \neq NULL

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

Cerca(x :varchar, c :varchar[1..*], t :intUnità, d :Date, cop :int_GEO): lva [0..*]

• pre-cond: $d \geq \text{NOW}()$

• post-cond:

```
for( $dc:c$ ) {
    Q=QU(SELECT * FROM Cucina cuc WHERE cuc.tipoCuc =  $dc$ )
}
```

```
Result = ( SELECT DISTINCT r.piva
            FROM Ristorante r, Città c, Sconto s, Q cuc, cuc.offerte, Giorno g, giorno_scon
            WHERE s.ristorante=r.piva AND r.città=c.id
            AND c.nome=x AND cuc.offerte.cucina=cuc.tipoCuc
            AND cuc.offerte.ristorante=r.piva AND s.percentuale>=t.asso
            AND s.inizioG<=d AND s.fineG>=d AND giorno_scon.sconto=s.id
            AND extract(dow from d)=g.nome AND giorno_scon.g=g);
```


Risposta alla **Domanda 8** (segue)

Matricola:

Minute

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.]





Matricola:

Minute

[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.]

