



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 (diagramma delle classi UML)

Dati dello studente e dell'esame

Cognome e nome: Matricola:

Data:

Corso di laurea e canale di appartenenza:

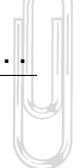
- Laurea in Informatica, canale 1 (Prof. G. Perelli)
- Laurea in Informatica, canale 2 (Prof.ssa M. De Marsico)

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

Rinuncia alla prova

Desidero rinunciare a questa prova d'esame. Firma:





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 UML concettuale delle classi (*)
 - (parte del)le specifiche formali delle classi e delle associazioni
 - le specifiche dei tipi di dato
 - la specifica formale dei vincoli esterni (*)
- 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)
 - (parti del)le specifiche formali degli use-case. (30 minuti; 60 minuti al massimo)

Si richiede *esplicitamente* di modellare le specifiche formali delle operazioni di classe e/o use-case necessarie a modellare i requisiti contrassegnati dalla barra laterale (come quella qui a sinistra), *inclusa* tutte le eventuali operazioni ausiliarie, usando l'estensione della logica del primo ordine studiata nel corso. (*)

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 UML concettuale delle classi e delle specifiche (20 minuti; 30 minuti al massimo):
 - * scelta del DBMS da utilizzare
 - * progettazione della corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
 - * ristrutturazione del diagramma UML concettuale delle classi e delle specifiche 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 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 immersi. (*)

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 31 in poi possono essere utilizzate per scrivere minute che non verranno valutate.

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



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) Clienti:

- + prenotazione [2]
- + nome
- + cognome
- + email

2) PRENOTAZIONE

- ~~+ Istante~~
- ~~+ Giorno~~
- ~~+ ORA~~
- + RISTORANTE [3]
- ~~+ #Commensali~~
- + Confermat.
- + Promozione (0..1) [4]
- + Rifiutato
- + pendente
- + Completata
- + NON utilizzata
- + Annullata

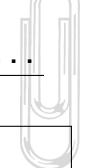
3) RISTORANTE

- + nome
- + partita IVA (Stringa Numerica)
- + indirizzo
- + Città
- + tipi di cucina [4]
- + chiuso per un periodo

4) CUCINE

5) promozione

- + % Sconto
- + Periodo
- + ordine
- + Max #Gperf: 1 giorno



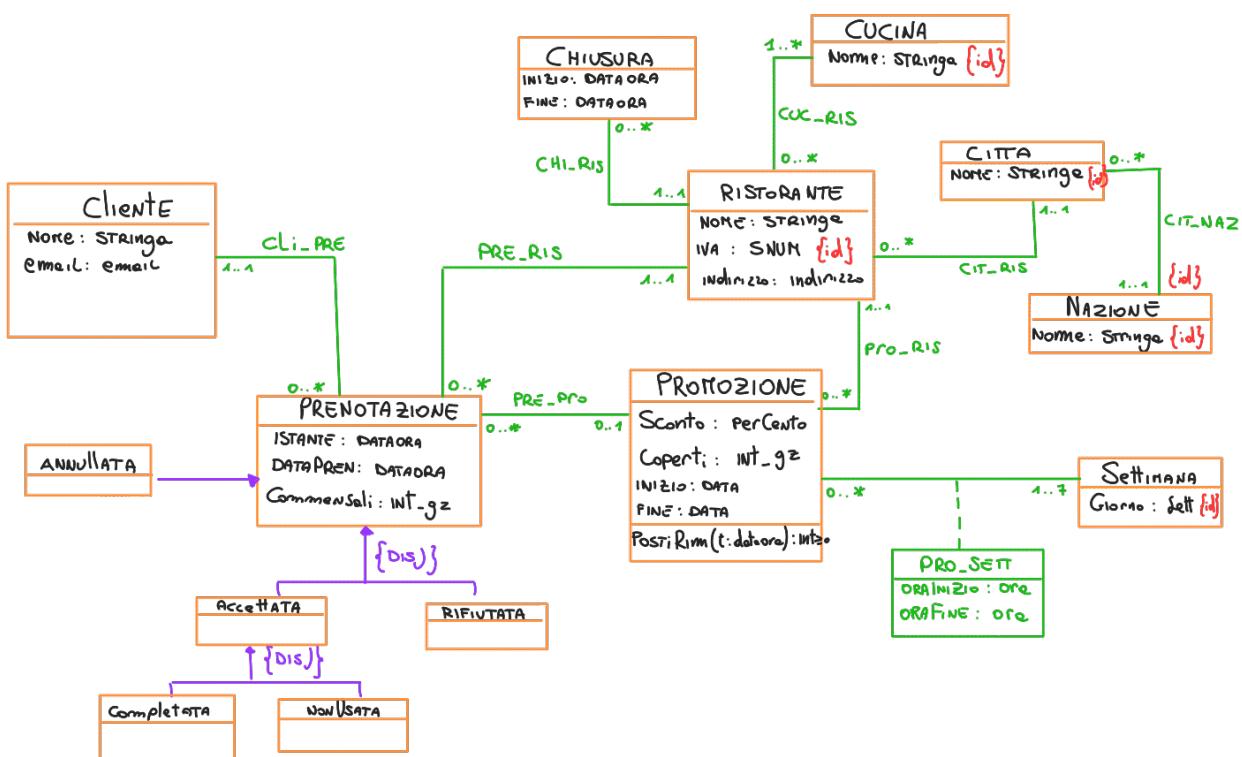
Risposta alla Domanda 1 (segue)

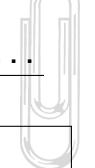
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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

Diagramma UML concettuale delle classi

Produrre un diagramma UML concettuale delle classi per l'applicazione in termini di classi, associazioni, attributi, generalizzazioni, operazioni di classe.





Risposta alla Domanda 2 (segue)

Specifiche delle classi o associazioni Per ogni classe o associazione del diagramma **con** operazioni o vincoli:

- Definire la specifica formale di eventuali operazioni necessarie a modellare i requisiti contrassegnati dalla barra laterale, ed eventuali vincoli esterni. Usare la logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale vista nel corso, usando il seguente alfabeto:
 - Un simbolo di predicato $C/1$ per ogni classe C .
Semantica di $C(x)$: x è una istanza di C .
 - Un simbolo di predicato $T/1$ per ogni tipo di dato T .
Semantica di $T(x)$: x è un valore di T .
 - Un simbolo di predicato $\text{assoc}/2$ per ogni associazione binaria assoc.
Semantica di $\text{assoc}(c_1, c_2)$: (c_1, c_2) è una istanza di assoc.
 - Un simbolo di predicato $\text{attr}/2$ per ogni attributo attr di entità
Semantica di $\text{attr}(c, v)$: uno dei valori dell'attributo attr dell'istanza c è v .
 - Un simbolo di predicato $\text{attr}/3$ per ogni attributo attr di associazione binaria.
Semantica di $\text{attr}(c_1, c_2, v)$: uno dei valori dell'attr. attr del link (c_1, c_2) è v .
 - Un simbolo di predicato $\text{op}/(n+2)$ per ogni operazione di classe ad n argomenti.
Semantica di $\text{op}(c, \arg_1, \dots, \arg_n, v)$: uno dei valori di ritorno di op, quando invocata sull'istanza c e con argomenti \arg_1, \dots, \arg_n è v .
 - Il simbolo di $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...PRENOTAZIONE.....</p> <p>Operazioni, vincoli:</p> <p>[V. Solo Accettate Ependente. Può Essere - Annullato]</p> $\forall p \text{ Prenotazione}(p) \wedge$ $\text{Annullata}(p) \longrightarrow (\neg \text{Completata}(p) \wedge \neg \text{NonUsata}(p) \wedge \neg \text{Rifiutata}(p))$ <p>[V. DATAPREN. maggiore . ISTANTE]</p> $\forall p, i, t \text{ Prenotazione}(p) \wedge \text{ISTANTE}(p, i) \wedge \text{DATAPREN}(p, t) \longrightarrow i \leq t$ <p>[V. Prenotazione. CON. PROMOZIONE. VALIDA]</p> $\forall \text{PRE}, \text{PRO}, \text{D}, \text{o}, \text{T}, \text{i}, \text{f}, \text{r} \left(\text{Prenotazione}(\text{PRE}) \wedge \text{ISTANTE}(\text{PRE}, \text{p}) \wedge \text{Promozione}(\text{PRO}) \wedge \text{Pre_Pro}(\text{PRO}, \text{PRO}) \wedge \text{DATAPREN}(\text{PRE}, \text{t}) \right.$ $\wedge \text{DATA}(\text{t}, \text{d}) \wedge \text{ORA}(\text{t}, \text{o}) \wedge \text{INIZIO}(\text{PRO}, \text{i}) \wedge \text{FINE}(\text{PRO}, \text{f}) \longrightarrow i \leq \text{d} \wedge \text{d} \leq f$ $\wedge \exists s, \text{oi}, \text{of}, \text{g}, \text{n}, \text{gg} \text{ Pro_sett}(\text{PRO}, \text{s}) \wedge \text{orarioInizio}(\text{PRO}, \text{s}, \text{oi}) \wedge$ $\text{oraFine}(\text{PRO}, \text{s}, \text{of}) \wedge \text{oi} \leq \text{o} \wedge \text{o} \leq \text{of} \wedge \text{giornoSettimana}(\text{d}, \text{g}) \wedge \text{giorno}(\text{s}, \text{gg})$ $\wedge \text{gg} = \text{g} \wedge \text{Coperti}(\text{PRO}, \text{n}) \wedge \text{n} \leq \text{PostiRim}(\text{p}))$ <p style="text-align: right;">giornoSettimana(a, b) a: date b: giorni b è il giorno dello settimana di a</p> <p style="text-align: right;">13/1/15 14/4/15 giornoSettimana(13/1/15, 'lunedì') ~ TRUE giornoSettimana(13/1/15, 'marc') ~ FALSE</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...PROMOZIONE.....</p> <p>Operazioni, vincoli:</p> <p>O. Posti rimasti. Per. PROMOZIONE</p> <p>PostiRim(t: dataOra): intero ≥ 0</p> <p>Pre: nessuna</p> <p>Post:</p> $P = \{(p, n, i) \mid \text{Pre_Pro}(p, \text{this}) \wedge \text{Commissal}(p, c) \wedge \text{ISTANTE}(p, i)\}$ <p>Sia a + t.c. Coperti(this, a)</p> $\text{Rim}_c = a - \left(\sum_{(p, n, i) \in P} n \right)$ <p>[V. INIZIO - minore - FINE]</p> $\forall p, i, f \text{ Promozione}(p) \wedge \text{inizio}(p, i) \wedge \text{fine}(p, f) \longrightarrow f \geq i$
--	---

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: PRO-SET</p> <p>Operazioni, vincoli:</p> <p>[V. ore_inizio, minore, ore_fine]</p> <p>$\nexists p, s, i, f \quad PRO-SET(p, s) \wedge oreInizio(p, s, i) \wedge oreFine(p, s, f) \rightarrow f \geq i$</p>	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: CHIUSURA</p> <p>Operazioni, vincoli:</p> <p>[V. inizio, minore, fine]</p> <p>$\forall c, i, f \quad chiusura(c) \wedge inizio(c, i) \wedge fine(c, f) \rightarrow f \geq i$</p>
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: PRENOTAZIONE</p> <p>Operazioni, vincoli:</p> <p>[V. PRENOTAZIONE, ad_un_RISTORANTE, CON_PROMOZIONE, Solo_Solo_PromoZIONE, e_dell_RISTORANTE]</p> <p>$\nexists pre, r, pro$ $(Prenotazione(pre) \wedge Ristorante(r) \wedge pre-RIS(pre, r) \wedge Promozione(pro) \wedge pre-Pro(pre, pro) \rightarrow PRO-RIS(PRO, r))$</p> <p>[V. NO_Prenotazioni, CON_CHIUSURE]</p> <p>$\forall r, c, i, f$ $(Ristorante(r) \wedge chiusura(c) \wedge CHI-RIS(c, r) \wedge inizio(c, i) \wedge fine(c, f) \rightarrow \exists p, d \quad prenotazione(p) \wedge dataPren(p, d) \wedge i \leq d \wedge d \leq f \wedge pre-RIS(p, r))$</p>	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>
<p>5 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>	<p>8 Tipo: Classe Associazione (cerchiare)</p> <p>Nome:</p> <p>Operazioni, vincoli:</p>

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

Stringa = Stringa normale

Email = Stringa secondo STANDARD

DataOra = Stringa per rappresentare in formato standard di: data + ora

Int_gz = intero > 0

DATA = Stringa per rappresentare in formato standard di: data

Percanto = Reale (0..1)

int_p = intero ≥ 0

Se $\Pi = \{\text{'lunedì'}, \text{'martedì'}, \dots, \text{'domenica'}\}$

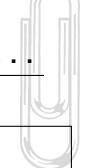
SNUM = Stringa per rappresentare la partita IVA

Indirizzo = (via: Via , civico: Civico , CAP: CAP)

Via = Stringa ...

Civico = Stringa ...

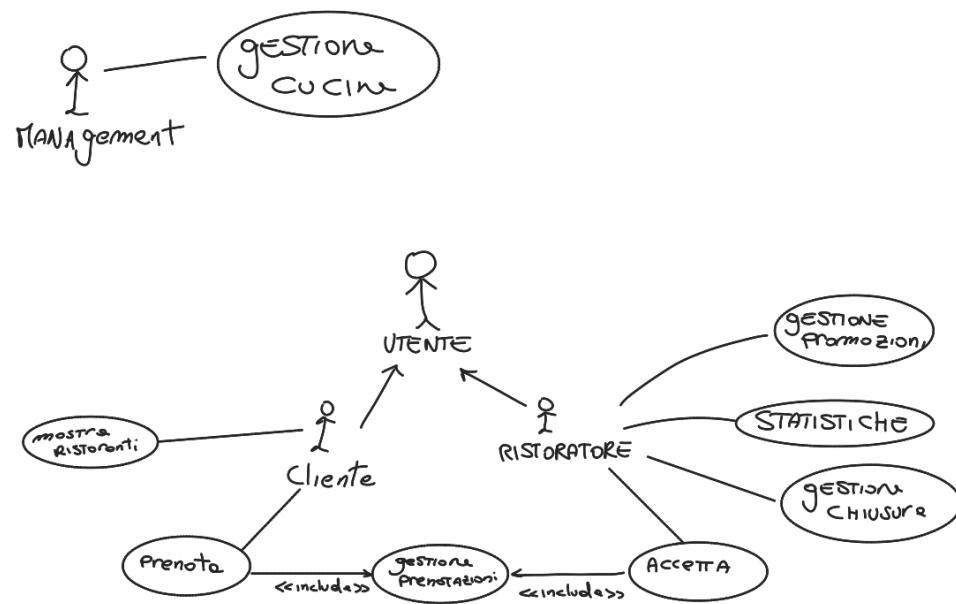
CAP = Stringa ...



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





Questa pagina è stata intenzionalmente lasciata vuota

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

Risposta

- gestione cucina
 - NUOVA_CUCINA (n: Stringe)
 - mostre RISTORANTI:
 - def Cerca(x:CITTÀ, c:CUCINA[1..*], s:PerCento, d:DATA, n:int,gz): RISTORANTI[0..*]
 - Statistiche
 - def CALCOLA(i: dataora, f: dataora, r: RISTORANTE): (Promozione, Reale>0)[0..*]
- : vanno definite tutte le segnature



Questa pagina è stata intenzionalmente lasciata vuota

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

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

Risposta

Cerca($x: \text{Citta}, c: \text{CUCINA}[1..*], s: \text{PerCento}, d: \text{DATA}, n: \text{Int_gz}$): RISTORANTI [$0..*$]
 PRE: $d \geq \text{ADDESSO}$

POST:

$$R = \left\{ r \mid \begin{array}{l} \text{Ristorante}(r) \wedge (\exists p \text{ Promozione}(p) \wedge \text{pro_ris}(p, r) \wedge \text{Sconto}(p, s) \wedge sc \geq s) \\ \wedge (\exists ch, ci, cf \text{ CHISURA}(ch) \wedge \text{CHI_RIS}(ch, r) \wedge \text{INIZIO}(ch, ci) \wedge \text{fine}(ch, cf) \wedge ci \leq d \wedge d \leq cf) \\ \wedge (\exists cuc \text{ CUCINA}(cuc) \wedge \text{cuc_ris}(cuc, r) \wedge c \in cuc) \\ \wedge (\exists p, s, set, gset, i, f, o_i, of, di, df, o \\ \text{Promozione}(p) \wedge \text{pro_ris}(p, r) \wedge \text{Settimana}(s) \wedge \text{pro_set}(p, s) \wedge \text{GiornoSettimana}(d, set) \wedge \text{Giorno}(s, gset) \wedge set = gset \\ \wedge \text{INIZIO}(p, i) \wedge \text{FINE}(p, f) \wedge \text{oraInizio}(p, s, o_i) \wedge \text{oraFine}(p, s, of) \wedge i \leq d \wedge d \leq f \wedge \text{Ora}(d, o) \wedge o_i \leq o \wedge o \leq of \\ \wedge n \leq \text{Postrim}(d)) \end{array} \right\}$$

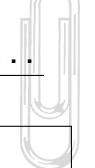
RESULT = R

Calcola($i: \text{dataOra}, f: \text{dataOra}, r: \text{ristorante}$): (Promozione, Reale > 0) [$0..*$]
 PRE: $i \leq f \wedge \exists p \text{ pro_ris}(p, r)$

POST:

$$R = \left\{ (p, m) \mid \begin{array}{l} \text{pro_ris}(p, r) \wedge \\ gg = \left\{ d \mid \begin{array}{l} \exists p, f \\ \text{INIZIO}(p, i) \wedge \text{fine}(p, f) \wedge \text{data}(d) \wedge i \leq d \wedge i \leq d \wedge f \geq d \wedge f \geq d \wedge (\exists gg, gset \text{ pro_set}(p, gg) \wedge \text{giorno}(gg, gset)) \end{array} \right\} \\ \wedge ng = |gg| \\ \wedge pers = \left\{ (pre, n) \mid \begin{array}{l} \text{pre_pro}(pre, r) \wedge \text{commensal:}(pre, n) \wedge \\ \exists g, d \text{ dataPre}(pre, g) \wedge \text{data}(g, d) \wedge d \in gg \end{array} \right\} \\ \wedge np = \sum_{(pre, n) \in pers} n \quad \wedge \quad m = np/ng \end{array} \right\}$$

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 UML delle classi concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

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

Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare **POSTGRESQL**.....

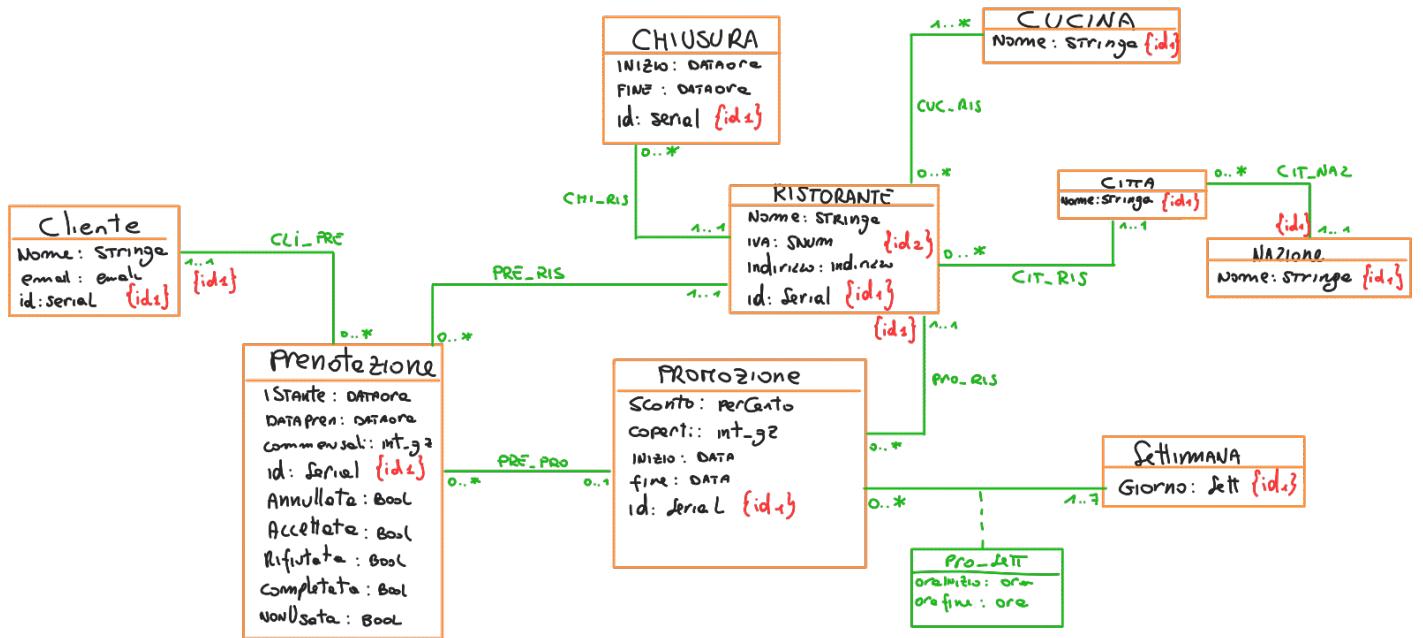
Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```

CREATE DOMAIN STRINGA AS TEXT check(VALUE ~'..')
: DOMAIN EMAIL : TEXT check(VALUE ~'..')
: DOMAIN DATAORA : timestamp
: DOMAIN INT_GZ : INT check(VALUE > 0)
: DOMAIN DATA : DATE
: DOMAIN Percento : REAL check(VALUE ≥ 0 AND VALUE ≤ 1)
: DOMAIN INT_P : INT check(VALUE ≥ 0)
: type SETT : ENUM ('lunedì',...)
: DOMAIN SNUM : STRING check(VALUE ~'..')
: type Indirizzo : (
    VIA: TEXT check(VALUE ..)
    CIVICO: TEXT check(VALUE ..)
    CAP: TEXT check(VALUE ..)
)

```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Aggiunte Chiavi primarie

Rimosse IS-A su Prenotazione per Fusione

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. DISJoint. Accettata. Rifiutata]

$$\nexists p \text{ Prenotazione}(p) \rightarrow [\neg(\text{Accettata}(p, 'TRUE') \wedge \text{Rifiutata}(p, 'TRUE'))]$$

[V. DISJoint. Completata. NonUsata]

$$\forall p \text{ Prenotata}(p) \rightarrow [\neg(\text{Completata}(p, 'TRUE') \wedge \text{NonUsata}(p, 'TRUE'))]$$

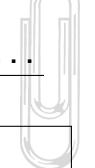
[Dipendenza tra IS-A]

$$\forall p \text{ Prenotazione}(p) \wedge (\text{Completata}(p, 'TRUE') \vee \text{NonUsata}(p, 'TRUE')) \rightarrow \text{Accettata}(p, 'TRUE')$$

[V. Annullata. NO. Completa. nonUsata. rifiutata]

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

$$(\text{che} \wedge (\text{NOT}(\text{Annullata} \text{ IS } \text{TRUE}) \vee (\text{NOT}(\text{Completa} \text{ IS } \text{TRUE}) \wedge \text{NOT}(\text{Rifiutata} \text{ IS } \text{TRUE}) \wedge \text{NOT}(\text{NonUsata} \text{ IS } \text{TRUE})))$$



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 UML delle classi ristrutturato.

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

1 Relazione .CLIENTE..... (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>Nome</u>	EMAIL	<u>ID</u>					
Domini	Stringa	email	SERIAL					

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

2 Relazione PRENOTAZIONE... (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>ID</u>	ISTANTE	DATAPREN	Commensali	Annnullata	Rifiutata	Accettata	Completata
Domini	SERIAL	DATA ora	DATA ora	INT-gz	Boolean	Boolean	Boolean	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):

FK Cliente REF Cliente(id)
 Check (NOT(NonUsato IS TRUE AND Completata IS TRUE))
 Check (DataPren >= Istante)
 Check (NOT(Completata IS TRUE OR NonUsato IS TRUE) OR (Accettata IS TRUE))
 Check (NOT(Accettata IS TRUE AND Rifiutata IS TRUE))
 Check (NOT(Annnullata IS TRUE) OR (NOT(NonUsato IS TRUE) AND NOT(Completata IS TRUE) AND NOT(Rifiutata IS TRUE)))

La relazione accorda le relazioni che implementano le seguenti associazioni: ...cli-pre.....

3 Relazione Prenotazione. (cont.) (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>Non Usato</u>	<u>Cliente</u>	Prenotazione*	Ristorante				
Domini	Boolean	INT	INT	INT				

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 Ristorante REF Ristorante (ID)
 FK (Promozione, Ristorante) REF Promozione (ID, Ristorante)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...PRE-RIS, ...PRE-PRO.....

4 Relazione RISTORANTE..... (nome)	Derivante da: classe associazione (cerchiare)
---	--

Attributi	<u>ID</u>	Nome	IVA	Indirizzo	CITTÀ			
Domini	SERIAL	Stringa	SNUM	Indirizzo	Stringa			

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

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

Unique (IVA)
 $id \subseteq CUCINA(RISTORANTE)$

La relazione accorda le relazioni che implementano le seguenti associazioni: ...CIT-RIS.....

5 Relazione ...PROMOZIONE... (nome)	Derivante da: classe associazione (cerchiare)
--	--

Attributi	<u>ID</u>	Ristorante	Sconto	Coperti	INIZIO	FINE		
Domini	SERIAL	INT	PERCENTO	INT-gz	DATA	DATA		

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

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

Check (Inizio <= Fine)
 $id \subseteq PRO-SETT(PROMOZIONE)$
 $Ristorante \subseteq PRO-SETT(PROMOZIONE)$

La relazione accorda le relazioni che implementano le seguenti associazioni: ...PRO-RIS.....

6 Relazione .PROMO-FETT..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Promozione</u> <u>Ristorante</u> <u>Settimana</u> <u>oraInizio</u> <u>oraFine</u>	
Domini int int Sett ora ore	

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 (Promozione, Ristorante) REF Promozione (id, Ristorante)

fk Settimana ref Settimana (giorno)

La relazione accorda le relazioni che implementano le seguenti associazioni:

7 Relazione .SETTIMANA..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Giorno</u>	
Domini Sett	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

8 Relazione .CHIUSURA.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>id</u> <u>Inizio</u> <u>Fine</u> <u>Ristorante</u>	
Domini int Date/ore Date/ore int	

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 Ristorante REF Ristorante (id)

check (Inizio <= fine)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...CHI-RIS.....

9 Relazione .CUCINA..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Nome</u>	
Domini Stringe	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione .CU.C.-RIS.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Ristorante</u> <u>Cucina</u>	
Domini int Stringe	

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 Ristorante REF Ristorante (id)

FK Cucina ref cucina (Nome)

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione ...CITTÀ..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi	<u>Nome</u>	<u>Nazione</u>						
Domini	Stringe	Stringe						

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 UTAZIONE REF Nazione(Nome)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...CIT-NAZ...

12 Relazione ...NAZIONE..... (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi	<u>Nome</u>							
Domini	Jtngy							

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

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

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

13 Relazione (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

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

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

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

14 Relazione (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

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

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

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

15 Relazione (nome) Derivante da: **classe** | **associazione** (cerchiare)

Attributi								
Domini								

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

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

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

16	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

17	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

18	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

19	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

20	Relazione (nome)	Derivante da: classe associazione (cerchiare)
Attributi		
Domini		

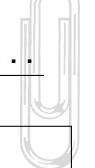
Gli attributi chiave primaria sono sottolineati, quelli i cui valori possono essere NULL sono contrassegnati con *

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

Ulteriori vincoli esterni

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



Risposta alla Domanda 7 (segue)

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di classe e/o use-case definite per modellare i requisiti contrassegnati dalla barra laterale della specifica dei requisiti.

In particolare, per ogni operazione definire la segnatura, in termini di nome dell'operazione, nomi e dominio SQL degli argomenti, dominio SQL dell'eventuale valore di ritorno, e un algoritmo in pseudo-codice con SQL immerso che verifichi le precondizioni e garantisca il raggiungimento delle postcondizioni definite in fase di Analisi. Specificare, per ogni operazione, se debba essere implementata nel DBMS o nel *back-end*.

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

Risposta

CERCA($x: \text{int}$, $c: \text{insieme}(\text{stringe})$, $s: \text{percent}$, $d: \text{DATA}$, $n: \text{int}_{\geq 0}$, $na: \text{int}$): $\text{insieme}(\text{int})$

IF $d < \text{NOW}()$ THEN
RAISE EXCEPTION

Select Distinct(r.id)

FROM Ristorante r Join CUC_RIS cr ON r.id = cr.Ristorante
Join Promozioni p ON p.Ristorante = r.id
Join PRO_SETT ps ON ps.Promozione = p.id AND ps.Ristorante = p.Ristorante
Join CITTA ci ON x = ci.Nome AND na = ci.Nazione
Join Prenotazioni pr ON pr.Promozione = p.id AND pr.Ristorante = r.id
WHERE cr.Cucina IN c AND p.Sconto >= s AND p.Inizio <= d AND d <= p.Fine AND
EXTRACT(DOW FROM d) = ps.Settimana
Group by p.id HAVING ($N \leq (p.Coperti - \text{SUM}(pr.Comensal))$)

CALCOLA(:date, f: date, r: int): $\text{insieme}(\text{int}, \text{int}, \text{REAL})$

OK = Select 1 From Promozione Where Ristorante = r

IF i > f OR NOT OK THEN
RAISE EXCEPTION

R = Select sub.id, AVG(sub.pers)

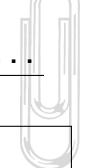
FROM (SELECT Pro.id, CAST(pre.datapren AS DATA) AS DT, SUM(pr.Comensal) AS pers
FROM Promozione pro Join Prenotazione pre ON pre.Promozione = pro.id AND pre.Ristorante = pro.Ristorante
Join PRO_SETT ps ON ps.Promozione = pro.id AND ps.Ristorante = pro.Ristorante

Where CAST(pre.datapren AS DATA) Between i AND f AND Cast(datapren AS SETT) = ps.Settimana

Group by Pro.id, pre.datapren) AS sub

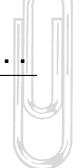
Group by sub.id

RETURN R



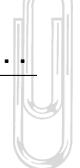
Risposta alla Domanda 8 (segue)

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.]