



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:

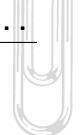
- Laurea in Informatica, canale 1 (Prof. G. Perelli)
- Laurea in Informatica, canale 2 (Prof.ssa M. De Marsico)

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.

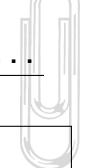


Questa pagina è stata intenzionalmente lasciata vuota

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



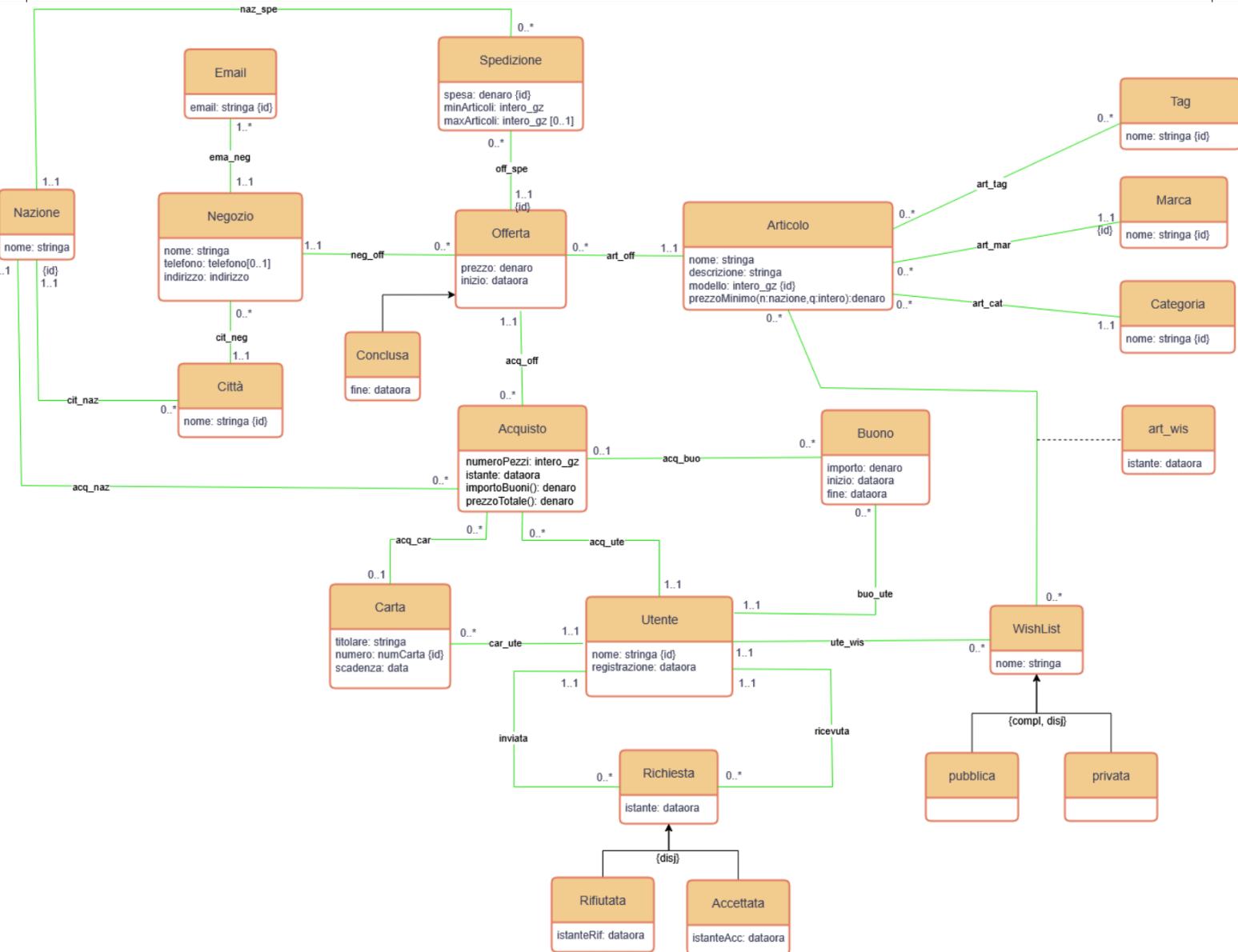
Risposta alla Domanda 1 (segue)

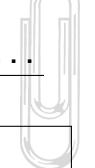
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ..Conclusa.....</p> <p>Operazioni, vincoli:</p> <p>[V. offerta, fine, maggiore, inizio]</p> <p>$\nexists o, i, f \quad \text{Conclusa}(o) \wedge \text{INIZIO}(o, i) \wedge \text{fine}(o, f) \longrightarrow f > i$</p>	<p><input type="checkbox"/> Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ..Richiesta.....</p> <p>Operazioni, vincoli:</p> <p>[V. No, Richiesta, a, sa, stessi]</p> <p>$\nexists r, u, u' \quad \text{Richiesta}(r) \wedge \text{invitata}(u, r) \wedge \text{ricevuta}(u', r) \longrightarrow u \neq u'$</p> <p>[V. rifiutata, istante, minore, istanteRif]</p> <p>$\nexists r, t, t' \quad \text{Rifiutata}(r) \wedge \text{istante}(r, t) \wedge \text{istanteRif}(r, t') \longrightarrow t < t'$</p> <p>[V. accettata, istante, minore, istanteAcc]</p> <p>$\nexists r, t, t' \quad \text{Accettata}(r) \wedge \text{istante}(r, t) \wedge \text{istanteAcc}(r, t') \longrightarrow t < t'$</p>
--	--

<p>3 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>A.C.quisito</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. <u>acquista, con.la.carta, dell'acquirente</u>]</p> <p>$\forall a, u$ $(acquisto(a) \wedge acq-ute(a, u) \wedge (\exists c \ Acq-car(a, c)) \rightarrow car-ute(c, u))$</p> <p>[V. <u>Se...paga...con...buoni...dove...essere...dell'acquirente</u>]</p> <p>$\forall a, u$ $(acquisto(a) \wedge acq-ute(a, u) \wedge (\exists b \ Acq-buo(a, b)) \rightarrow buo-ute(b, u))$</p> <p>[V. <u>almeno...un... pagamento</u>]</p> <p>$\forall a, u$ $(acquisto(a) \rightarrow ((\exists b \ acq-buo(a, b)) \vee (\exists c \ acq-car(a, c))))$</p> <p>[V. <u>Se...buoni...Impoff, uttri...allora...anche...carta</u>]</p> <p>$\forall a \ acquisto(a) \wedge (\exists ib, pt$ $(importoBuoni(a, ib) \wedge preziototale(a, pt) \wedge ib < pt) \leftrightarrow (\exists c \ acq-car(a, c)))$</p>	<p>6 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>A.C.quisito</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. <u>Solo...buoni...validi</u>]</p> <p>$\forall a, b, t, i, f$ $(acquisto(a) \wedge buono(b) \wedge acq-buo(a, b) \wedge istante(a, t) \wedge inizio(b, i) \wedge fine(b, f) \rightarrow i \leq t \wedge t \leq f)$</p> <p>[<u>Acquisti...Solo...in...mazzoni...disponibili</u>]</p> <p>$\forall a, n$ $(acquisto(a) \wedge Acq-neg(a, n) \rightarrow (\exists o, s \ offerta(o) \wedge acq-off(o, o) \wedge off-spe(o, s) \wedge Naz-Spe(n, s)))$</p>
<p>4 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Buono</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. <u>Impoff, fne</u>]</p> <p>$\forall b, i, f \ (Buono(b) \wedge inizio(b, i) \wedge fine(b, f) \rightarrow f > i)$</p>	<p>7 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Offerta</u>.....</p> <p>Operazioni, vincoli:</p> <p>[No. <u>offerte...sottoposte...stessa...negozi...stesso...articoli</u>]</p> <p>$\forall o, o', n, a, i, i'$ $(offerta(o) \wedge offerta(o') \wedge o \neq o' \wedge neg-off(o, o) \wedge neg-off(o, o') \wedge art-off(a, o) \wedge art-off(a, o') \wedge inizio(o, i) \wedge inizio(o, i') \rightarrow (\exists t \ dataora(t) \wedge (i \leq t \wedge (\#fine(o, f) \rightarrow t \leq f) \wedge i' \leq t \wedge (\#fine(o', f') \rightarrow t \leq f'))))$</p>
<p>5 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Spedizione</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. <u>no...Range...surpassi</u>]</p> <p>$\forall s, s', max', min', o$ $(spedizione(s) \wedge spedizione(s') \wedge off-spe(o, s) \wedge off-spe(o, s') \wedge minArticolo(j, min) \wedge minArticolo(j, min') \wedge s \neq s')$</p> <p>$\rightarrow (\exists n \ intero-gz(n) \wedge (min \leq n \wedge (\#maxArticoli(j, max) \rightarrow n \leq max) \wedge (min' \leq n \wedge (\#maxArticoli(j, max') \rightarrow n \leq max'))))$</p> <p>[V. <u>Range min < max</u>]</p> <p>$\forall s, i, f \ spedizione(s) \wedge minArticolo(i) \wedge maxArticolo(f) \rightarrow f \geq i$</p>	<p>8 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ... <u>Spedizione</u>.....</p> <p>Operazioni, vincoli:</p> <p>[V. <u>No...buchi...nei...range</u>]</p> <p>$\forall n \ (intero-gz(n) \rightarrow \exists s, m \ (spedizione(s) \wedge minArticolo(j, m) \wedge m \leq n \wedge ((\exists M \ max(s, M) \wedge n \leq M) \vee \exists N \ max(j, N)))$</p>

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

[V. Sds. carte. valhole]

$\forall c, e, s, t \ (acquisto(c) \wedge carta(c) \wedge acq_car(e, c) \wedge utente(a, t) \wedge studente(s, s) \rightarrow t \leq s)$

* ci sono altri vincoli per esempio vincoli temporali sulla creazione utente)

Tipi di dato

Stringa = ...

telefono = Stringa + regex

Indirizzo = (via: Stringa, civico: Stringa, cap: Stringa)

DataOra = (data, ora)

Denaro = reale ≥ 0

Intero_gz = Intero > 0

NumCarta = Stringa + regex

Operazioni di classe

[O. Acquisto]

ImportoBuoni(): Denaro

Pre:

Post:

$$B = \{(b, i) \mid buono(b) \wedge acq_buon(this, b) \wedge importo(b, i)\}$$

Se $B = \emptyset \rightarrow result = 0$

$$\text{Se } B \neq \emptyset \rightarrow result = \sum_{(b, i) \in B} i$$

PrezzoTotale(): Denaro

Pre:

Post:

Sia np t.c. numeroPezzi(this, np)

Sia p t.c. $\exists o \ offerta \wedge acq_off(this, o) \wedge prezzo(o, p)$

Sia Sp t.c. $\exists n, o, s \ nazionale(n) \wedge acq_naz(this, n) \wedge offerta(o) \wedge acq_off(this, o) \wedge off_spe(o, s) \wedge spedizione(s) \wedge spesa(s, sp) \wedge naz_spe(n, s) \wedge (\exists_{min, max} minArticoli(s, min) \wedge maxArticoli(s, max)) \wedge min \leq np \wedge np \leq max$

$$result = p \times np + sp$$

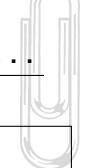
PrezzoMinimo(n: nazionale, q: Intero): Intero > 0

Pre:

Post:

$$A = \left\{ (p, s) \mid \begin{array}{l} \exists o, sp \ art_off(this, o) \wedge prezzo(o, p) \wedge spedizione(sp) \wedge off_spe(o, sp) \wedge spesa(sp, s) \wedge naz_spe(n, sp) \\ (\exists_{min, max} minArticoli(sp, min) \wedge maxArticoli(sp, max)) \wedge min \leq q \wedge (maxArticoli(sp, max) \rightarrow q \leq max) \end{array} \right\}$$

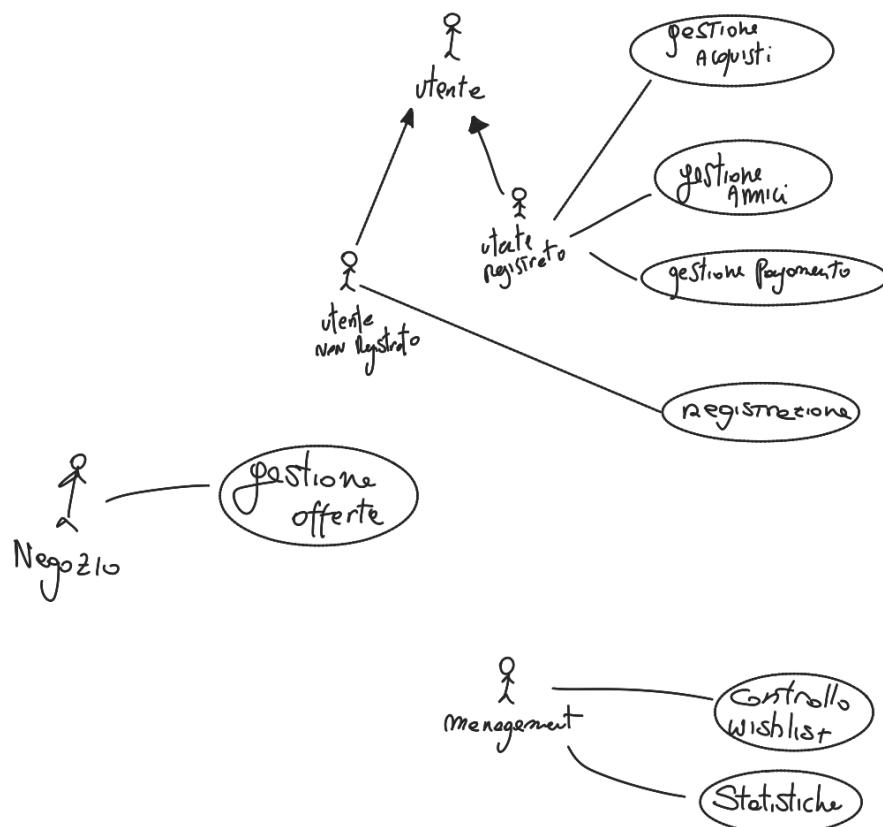
$$result = \min_{(p, s) \in A} (p + s)$$



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 **segnatura** delle operazioni in ogni use-case.

Risposta

Bisogna scrivere tutte

le segnature

Da fare



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

CalcolaPAEse (*i: dataora, f: dataora*): Nazione

Pre: $\exists a \text{ acquisto}(a)$

Post:

$$A = \left\{ (p, n) \mid \begin{array}{l} \text{NAZIONE}(p) \wedge n = \left\{ \begin{array}{l} \exists t \text{ acquisto}(a) \wedge \\ \text{ACQ-NAT}(a, p) \wedge \text{ISTANTE}(a, t) \wedge i \leq t \wedge t \leq f \end{array} \right\} \end{array} \right\}$$

$$\text{Result} = \underset{(p, n) \in A}{\text{Argmax}}(n)$$

CERCAOGGETTI (*c: categoria, T: TAG[0..*], n: nazione, q: intero > 0*) : (Articolo, Denaro)[0..*]

Pre:

Post:

$$A = \left\{ (a, d) \mid \begin{array}{l} \text{articolo}(a) \wedge \text{ART-CAT}(a, c) \wedge (\forall x \in T \text{ ART-TAG}(a, x)) \\ \wedge \text{PREZZO-MINIMO}(a, n, q, d) \wedge (\exists o, \text{OFFERTA}(o) \wedge \text{ART-OFF}(a, o) \wedge \text{OFF-SPEC}(o, s) \wedge \text{NAZ-SPEC}(n, s)) \end{array} \right\}$$

$$\text{Result} = A$$

moda(): Articoli[0..*]

Pre:

Post:

$$\text{Result} = \left\{ a \mid \text{articolo}(a) \wedge (\exists c \text{ CRESITA}(a, c) \wedge c > 0) \right\}$$

(Aux)

Cresita (*ar: articolo*): Reale

Pre: deve esistere almeno un acquisto negli ultimi 3 mesi (divisione per zero!)

Post:

$$A = \left\{ (a, q) \mid \begin{array}{l} \text{acquisto}(a) \wedge \text{NUMEROPEZZI}(a, q) \wedge (\exists o, \text{OFFERTA}(o) \wedge \text{ACQ-OFF}(a, o) \wedge \text{ART-OFF}(ar, o)) \\ \wedge (\exists i, \text{diff}, \text{mesidiff} \text{ ISTANTE}(a, i) \wedge \text{diff} = \text{adesso} - i \wedge \text{MESI}(diff, mesidiff) \wedge \text{mesidiff} \leq 3) \end{array} \right\}$$

Acquisti
di un Articolo
nell'ultimo
mese

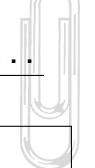
$$V = \sum_{(a, q) \in A} q$$

$$B = \left\{ (a, q) \mid \begin{array}{l} \text{acquisto}(a) \wedge \text{NUMEROPEZZI}(a, q) \wedge (\exists o, \text{OFFERTA}(o) \wedge \text{ACQ-OFF}(a, o) \wedge \text{ART-OFF}(ar, o)) \\ \wedge (\exists i, \text{diff}, \text{mesidiff} \text{ ISTANTE}(a, i) \wedge \text{diff} = \text{adesso} - i \wedge \text{MESI}(diff, mesidiff) \wedge \text{mesidiff} \leq 3 \wedge \text{mesidiff} \geq 1) \end{array} \right\}$$

$$M = \left(\sum_{(a, q) \in B} q \right) \times \frac{1}{2}$$

$$\text{Result} = \frac{V - M}{M} \times 100$$

[continua alla pagina seguente]



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:

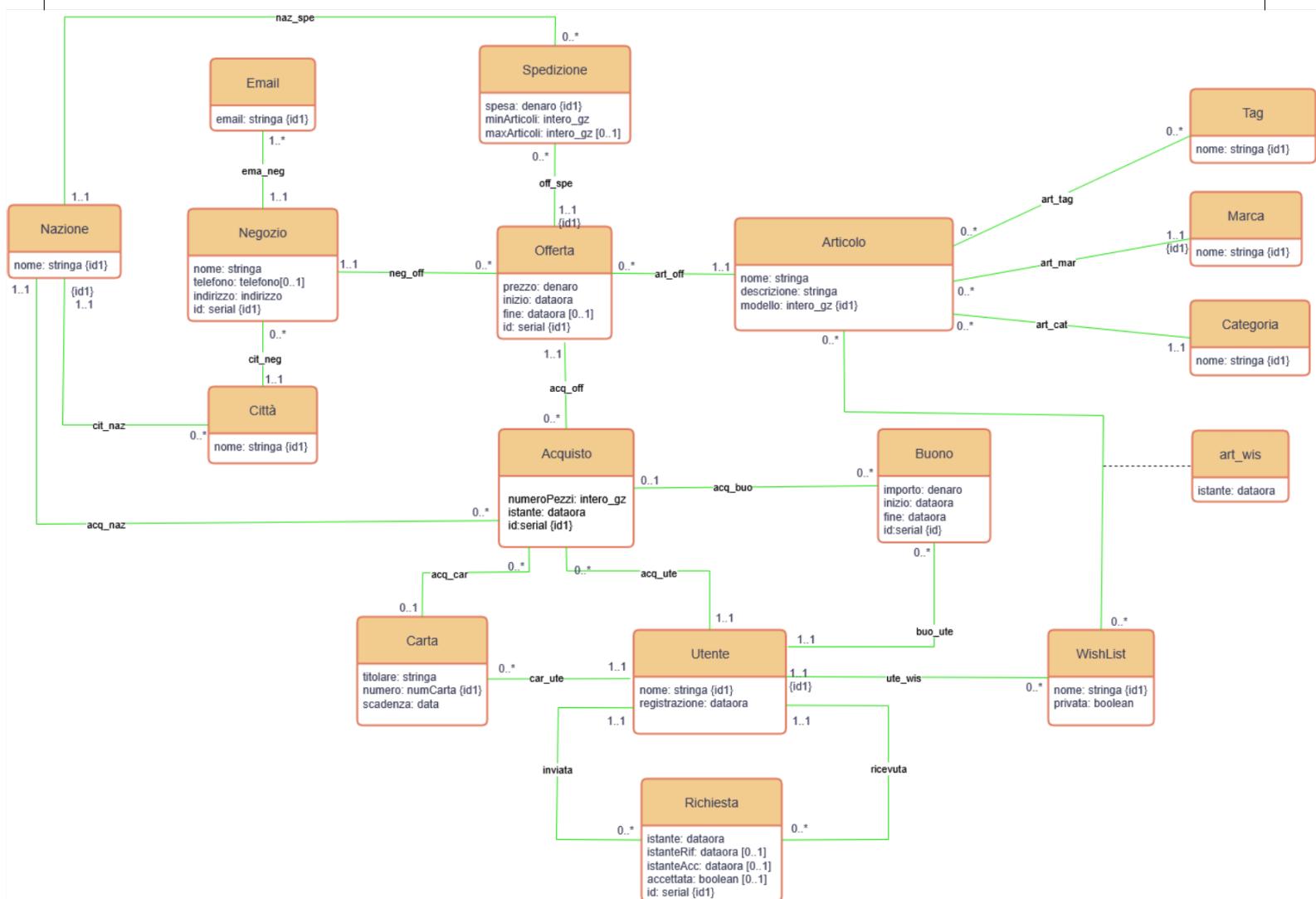
- progettare una corrispondenza tra i tipi di dato concettuali ed opportuni domini SQL (domini base o utente, oppure realizzati mediante relazioni aggiuntive) supportati dal DBMS scelto
- eliminare attributi multivale o composti
- eliminare relazioni is-a e generalizzazioni
- definire un identificatore 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
CREATE DOMAIN Telefono AS TEXT CHECK(VALUE ~'^...$')
CREATE TYPE indirizzo AS (
    VIA: STRINGA CHECK...
    Civico: STRINGA CHECK...
    CAP: STRINGA CHECK...
)
CREATE DOMAIN DATOORE AS TIMESTAMP
CREATE DOMAIN Denaro AS REAL CHECK(VALUE >= 0)
CREATE DOMAIN intero_ogr AS INT CHECK(VALUE >= 0)
CREATE DOMAIN numCarta AS TEXT CHECK(VALUE ~'^...$')
```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Rimossa le is-a tutte per fusioni

Aggiunti: chiavi primarie

modificati/aggiunti: classi: vincoli esterni

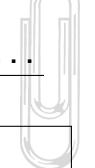
Vincoli esterni introdotti o modificati durante la fase di ristrutturazione

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

[V. richiesta disp]

$$\nexists r, i \text{ Richiesta}(r) \wedge \text{istante}(r, i) \longrightarrow (\text{accettata}(r, \text{true}) \iff (\exists t \text{ istanteAcc}(r, t) \wedge \neg \exists t' \text{ istanteRif}(r, t') \wedge i < t))$$

$$\nexists r, i \text{ Richiesta}(r) \wedge \text{istante}(r, i) \longrightarrow (\text{accettata}(r, \text{false}) \iff (\exists t \text{ istanteRif}(r, t) \wedge \forall t' \text{ istanteAcc}(r, t')))$$



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 .Utente..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>Nome</u>	Registrazione						
Domini	Stringa	Dataora						

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 .RICHIEDA.... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>id</u>	ISTANTE	istanteRef *	istanteAcc *	Accettata *	Inviate	Ricevute	
Domini	Serial	Dataora	Dataora	Dataora	Boolean	Stringe	Stringa	

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

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

Check (istante < istanteAcc) Check ((Accettata IS TRUE AND istanteAcc IS NOT NULL AND istanteRef IS NULL) OR (Accettata IS FALSE AND istanteRef IS NOT NULL AND istanteAcc IS NULL)) OR (Accettata IS NULL AND istanteAcc IS NULL AND istanteRef IS NULL))
 FK Inviate ref Utente(Nome)
 FK Ricevute ref Utente(Nome)

La relazione accorda le relazioni che implementano le seguenti associazioni: .. Inviate, Ricevuta

[3] Relazione .Carta..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>titolare</u>	numero	Scadenze	Utente				
Domini	Stringa	numCarta	Data	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 titolare Utente(nome)

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

[4] Relazione .Acquisto..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>id</u>	ISTANTE	numeroPezzi	Utente	Carta *	offerta	NAZIONE	
Domini	Serial	Dataora	Intero-gz	Stringe	numCarta	INT	Stringa	

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

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

FK NAZIONE ref NAZIONE(nome) FK utente ref Utente(nome)

FK offerta ref offerte (id)

FK Carta ref Carta(numero)

La relazione accorda le relazioni che implementano le seguenti associazioni: .. Acq-off, Acq-Naz, Acq-Car, Acq-Ute

[5] Relazione .Buono..... (nome) Derivante da: classe | associazione (cerchiare)

Attributi	<u>id</u>	importo	inizio	fine	Acquisto *	utente		
Domini	Serial	Decaro	Dataora	Dataora	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 Acquisto ref Acquisto(nome)

FK utente ref Utente(id)

La relazione accorda le relazioni che implementano le seguenti associazioni: .. Acq-Buo, Buo-Ute

6 Relazione .NAZIONE..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Nome</u>	
Domini <u>Stringa</u>	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

7 Relazione ...Citt..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Nome</u>	<u>Nazione</u>
Domini <u>Stringa</u>	<u>Stringa</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 NAZIONE ref NAZIONE(Nome)

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

8 Relazione .Negozio..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Nome</u>	<u>id</u> <u>telefono</u> * <u>indiriz</u>
Domini <u>Stringa</u>	<u>Serial</u> <u>telefono</u> <u>indiriz</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):

id <--> Negozio(Negozio)

La relazione accorda le relazioni che implementano le seguenti associazioni:

9 Relazione .email..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Email</u>	
Domini <u>Stringa</u>	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione .eme-neg..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>Negozio</u>	<u>email</u>
Domini <u>int</u>	<u>Stringa</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 Negozio ref Negozio(id)

FK email ref email(email)

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione <u>Spedizione</u> ... (nome)	Derivante da: classe associazione (cerchiare)					
Attributi <u>Spese</u> <u>offerto</u> minArticoli maxArticoli * Nazione						
Domini denaro int intero-ge intero-ge 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):

Check (minArticoli <= maxArticoli)

FK Nazione ref Nazione(nome)

PK offerto ref offerto(id)

La relazione accorda le relazioni che implementano le seguenti associazioni: . off-Spe, max-Spe

12 Relazione <u>Articolo</u> (nome)	Derivante da: classe associazione (cerchiare)					
Attributi <u>Nome</u> <u>descrizione</u> <u>moduli</u> <u>marca</u> <u>Categoria</u>						
Domini Stringe Stringe intero-ge 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 marca ref marca(nome)

FK Categoria ref Categoria(nome)

La relazione accorda le relazioni che implementano le seguenti associazioni: Art-mar, Art-cat

13 Relazione <u>Tag</u> (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:

14 Relazione <u>Art-tag</u> (nome)	Derivante da: classe associazione (cerchiare)					
Attributi <u>modello</u> <u>marca</u> <u>Tag</u>						
Domini intero-ge 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 (marca, modello) ref Articolo (marca, modello)

FK Tag ref Tag (name)

La relazione accorda le relazioni che implementano le seguenti associazioni:

15 Relazione <u>NishList</u> (nome)	Derivante da: classe associazione (cerchiare)					
Attributi <u>Nome</u> <u>privata</u> <u>utente</u>						
Domini Stringe Boolean 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:

16 Relazione .ART_WIS..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>wishList</u> <u>marca</u> <u>modello</u> <u>istante</u>	
Domini <u>String</u> <u>String</u> <u>interv-y</u> <u>dataOne</u>	

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

17 Relazione .offerto..... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>id</u> <u>WIZIO</u> <u>fine*</u> <u>prezzo</u> <u>modello</u> <u>Negozi</u> <u>marca</u>	
Domini <u>Serie</u> <u>dataOne</u> <u>dataOne</u> <u>denaro</u> <u>String</u> <u>int</u> <u>interv-y</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 Negozi Ref Negozi(id)

FK (modello, marca) Ref Articoli(modello, marca)

clock (fine > WIZIO)

La relazione accorda le relazioni che implementano le seguenti associazioni: .Art-off, neg-off

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.

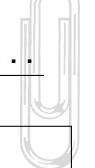
[T. Acquisti solo se p.tre disponibile]

Before Insert o Update on Acquisto

```
OK = EXISTS (Select 1
    From offerta o Join Spedizione s on o.id = s.offerta
    where new.offerta = o.id AND s.NAZIONE = NEW.NAZIONE)
)
if OK then
    Permit
else
    Raise Exception
```

Non scrivo tutti
i trigger

DA fare



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

CalcolaPaese(i: dataora, f: dataora) : Stringa

```
Select NAZIONE
FROM Acquisto WHERE BETWEEN i AND f
Group by NAZIONE
order by Count(*) DESC
Limit 1
```

CercaOggetti(c: stringa, T: insieme<String>, n: stringa, q: int) : insieme<(int, Denaro)>

```
Select DISTINCT(a.modello, a.marca), PrezzoMinimo(o.modello, o.marca, n, q)
From Articoli a Join offerte o on o.modello = a.modello AND o.marca = a.marca
Join Spedizioni s on s.offerta = o.id
Join Art-tag at on at.modello = a.modello AND at.marca = a.marca

Where a.categoria = c AND NOT EXISTS(
    Select nome FROM T
    Except Select tag FROM Art-tag Where modello = o.modello AND marca = o.marca
)
```

PrezzoMinimo(modello: int, marca: stringa, n: stringa, q: int) : Denaro

```
Select MIN(o.prezzo + s.spesa)
From offerte o Join spedizioni s on s.offerta = o.id
Where s.nazione = n AND q BETWEEN s.minArticoli AND s.maxArticoli
```

Risposta alla Domanda 8 (segue)

*QUESTO UECHE NON
MI CONVINCÈ

$\text{MODA}(\text{ }) : (\text{Intero-}\varphi_2, \text{Stringe})$

WITH MESE1 (mod, mar, TOT) AS (

```
Select A.modello, A.marca, Coalesce(Sum(O.numeropezi), 0) AS TOT
FROM Articolo A left join Acquisto O ON O.modello=A.modello AND O.marca = A.marca
WHERE O.ISTANTE Between Now() - Interval '3 months' AND Now() - 'months 1'
GROUP by A.modello, A.marca
```

)

WITH MESE1-3 (mod, mar, TOT) AS (

```
Select A.modello, A.marca, Coalesce(Sum(O.numeropezi), 0) AS TOT
FROM Articolo A left join Acquisto O ON O.modello=A.modello AND O.marca = A.marca
WHERE O.ISTANTE Between Now() - Interval '3 months' AND Now() - 'months 1'
GROUP by A.modello, A.marca
```

)

Select m1.mod, m1.mar

FROM meset m1, meset m2

Where m1.mod = m2.mod AND m1.mar = m2.mar

AND $(m1.TOT - (m2.TOT/2)) / (m2.TOT/2) * 100 > 0$

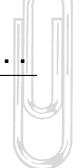
)

)

DA fare

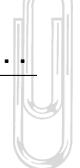
VANNO scritte tutte le operazioni

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.]