



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. Negozio

- 1.1 nome
- 1.2 indirizzo
- 1.3 email [1..*]
- 1.4 numero [0..1]
- 1.5 offerte in vendita

2. Offerta

- 2.1 spedizioni : insieme di Triple (Nazione, Intervallo pezzi, prezzo) Es: (Italia, 7-10, 3.99€)
- 2.2 sped_n.pezzi(naz:Nazione, n:intero ≥ 1): Reale ≥ 0
- 2.3 prezzo
- 2.4 inizio offerta
- 2.5 fine offerta
- 2.6 articolo in questione

3. Articolo

- 3.1 nome
- 3.2 descrizione
- 3.3 marca
- 3.4 numero modello
- 3.5 categoria
- 3.6 tag [1..*]
- 3.7 codice identificativo

4. Buono

- 4.1 utente proprietario
- 4.2 importo
- 4.3 inizio validità
- 4.4 fine //
- 4.5 utilizzato (buono usato parzialmente)

5. Utente

- 5.1 nome
- 5.2 istante registr.
- 5.3 carte associate
 - 5.3.1 nome titolare
 - 5.3.2 numero
 - 5.3.3 data scadenza [0..1] se non presente, e' scaduta
- 5.4 acquisto((articolo, pezzi)[1..*], Nazione): Importo
- 5.5 amici

6. Wishlist

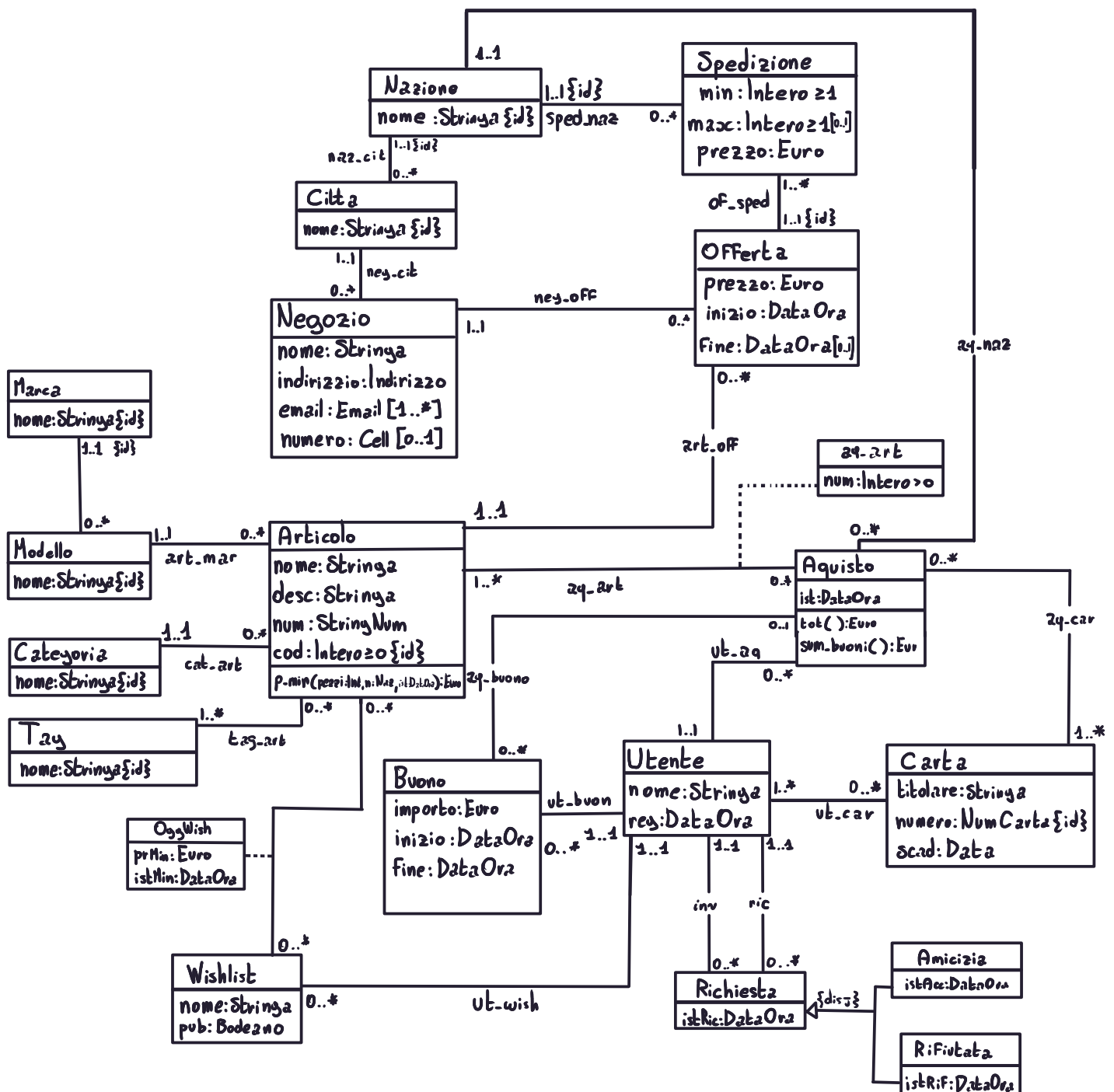
- 6.1 proprietario
- 6.2 nome
- 6.3 articoli

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.



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: **OFFerta**

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

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

Vincoli:

[V.inizio_off_minore_Fine]

$\forall \sigma, i, f [OFFerta(\sigma) \wedge inizio(\sigma, i) \wedge Fine(\sigma, f)] \rightarrow i < f$

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

Nome: **Spedizione**

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

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

Vincoli:

[V.no_intersezioni_pezzi]

$\neg \exists S_1, S_2, n, o, min_1, min_2, max_1, max_2$

$[sped_naz(S_1, n) \wedge sped_naz(S_2, n) \wedge S_1 \neq S_2 \wedge of_sped(S_1, o)$

$\wedge of_sped(S_2, o) \wedge min(S_1, min_1) \wedge min(S_2, min_2) \wedge max(S_1, max_1)$

$\wedge max(S_2, max_2)] \rightarrow$

$[min_1 \leq min_2 \wedge max_1 \geq max_2] \vee [min_2 \leq min_1 \wedge max_2 \geq max_1]$

$\vee [min_1 \leq min_2 \wedge max_1 \leq max_2] \vee [min_2 \leq min_1 \wedge max_2 \leq max_1]$

3 Tipo: **Entità** | Relationship (cerchiare)Nome: Buono

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.validità_corretta]$ $\forall b, i, f [Buono(b) \wedge inizio(b, i) \wedge Fine(b, f)] \rightarrow i < f$ $[V.buono_usato_se_valido]$ $\forall b, a, i, f, i, in [aq_buono(b, a) \wedge Fine(b, f) \wedge ist(a, i) \wedge inizio(b, in)] \rightarrow i \leq f \wedge i \geq in$ $[V.buono_utente]$ $\forall b, u, a [aq_buono(a, b) \wedge ut_aq(u, a)] \rightarrow ut_buono(u, b)$ 5 Tipo: **Entità** | Relationship (cerchiare)Nome: Utente

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.acquisto_dopo_reg]$ $\forall u, a, r, i [ut_aq(a, u) \wedge reg(u, r) \wedge ist(a, i)] \rightarrow r < i$ $[V.amicizia_dopo_reg]$ $\forall u, r, iric, ireg [inv(u, r) \vee ric(u, r)] \wedge reg(u, ireg) \wedge istRic(r, iric) \rightarrow iric > ireg$ 4 Tipo: **Entità** | Relationship (cerchiare)Nome: Richiesta

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.accettazione_dopo_richiesta]$ $\forall r, ir, ia [Amicizia(r) \wedge istRic(r, ir) \wedge istAcc(r, ia)] \rightarrow ia > ir$ $[V.rifiuto_dopo_richiesta]$ $\forall r, ir, i [Rifiutata(r) \wedge istRic(r, ir) \wedge istRiF(r, i)] \rightarrow i > ir$ $[V.no_subo_amicizia]$ $\neg \exists r, u inv(u, r) \wedge ric(u, r)$ 6 Tipo: **Entità** | Relationship (cerchiare)Nome: Carta

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V.usata_se_valida]$ $\forall c, a, i, f, d [aq_car(a, c) \wedge scad(c, f) \wedge ist(a, i) \wedge Data(d, i)] \rightarrow d < f$ $[V.carta_utente]$ $\forall c, u, a [aq_car(a, c) \wedge ut_aq(u, a)] \rightarrow ut_car(u, c)$

7 Tipo: Entità | Relationship (cerchiare)Nome: Spedizione

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V. pezzi_sped_massimi]$

$$\forall s, o, n, min [sped_naz(s, n) \wedge of_sped(s, o) \wedge min(s, min) \wedge \neg \exists m max(s, m)] \rightarrow [\neg \exists s_2, m \quad s \neq s_2 \wedge sped_naz(s_2, n) \wedge of_sped(s_2, o) \wedge max(s_2, m) \wedge m \geq min]$$
9 Tipo: Entità | Relationship (cerchiare)Nome: Richiesta

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V. no_2_richieste_insieme_se_non_ri_fiutate]$

$$\neg \exists u_1, u_2, r_1, r_2 [r_1 \neq r_2 \wedge inv(u_1, r_1) \wedge ric(u_2, r_2) \wedge inv(u_1, r_2) \wedge ric(u_2, r_1) \wedge \neg Ri_fiutata(r_1) \wedge \neg Ri_fiutata(r_2)]$$
 $[V. no_richieste_incrociate]$

$$\forall u_1, u_2, r_1 [inv(u_1, r_1) \wedge ric(u_2, r_1) \wedge \neg Ri_fiutata(r_1)] \rightarrow \neg \exists r_2 [inv(u_2, r_2) \wedge ric(u_1, r_2)]$$
8 Tipo: Entità | Relationship (cerchiare)Nome: Acquisto

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V. acquisto_articolo_se_c'e_offerta]$

$$\forall aq, ar, iaq [aq_art(aq, ar) \wedge ist(aq, iaq)] \rightarrow [\exists o, in \quad inizio(o, i) \wedge rt_off(ar, o) \wedge [\exists f \quad fine(o, f) \wedge in \leq iaq \leq f] \vee [iaq \geq in \wedge \neg \exists f \quad fine(o, f)]]$$
10 Tipo: Entità | Relationship (cerchiare)Nome: Acquisto

attributo	dominio	moltepl. (*)

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

Vincoli:

 $[V. acquisto_disponibilita_spedizione]$

$$\forall aq, n, ar [aq_art(aq, ar) \wedge aq_naz(aq, n)] \rightarrow \exists \sigma, s [art_off(ar, \sigma) \wedge of_sped(\sigma, s) \wedge sped_naz(s, n)]$$
 $[V. sped_min_minore_max]$

$$\forall s, m_1, m_2 [Spedizione(s) \wedge min(s, m_1) \wedge max(s, m_2)] \rightarrow m_1 \leq m_2$$

11 Tipo: **Entità** | Relationship (cerchiare)Nome: **Offer₂**

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no-intersezioni-offert₂] $\forall \sigma_1, \sigma_2, n, a, in_1, in_2, fi_1, fi_2$

$$[\sigma_1 \neq \sigma_2 \wedge \text{art-off}(a, \sigma_1) \wedge \text{art-off}(a, \sigma_2) \wedge \text{neg-off}(n, \sigma_1) \wedge \text{neg-off}(n, \sigma_2) \wedge \text{inizio}(\sigma_1, in_1) \wedge \text{inizio}(\sigma_2, in_2) \wedge \text{fine}(\sigma_1, fi_1) \wedge \text{fine}(\sigma_2, fi_2)] \rightarrow [fin_1 < in_2] \vee [fin_2 < in_1]$$
13 Tipo: **Entità** | **Relationship** (cerchiare)Nome: **OggWish**

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no-controllo-prezzo-prim₂-di-reg] $\forall w, u, igw, ir, a$

$$[\text{OggWish}(a, w) \wedge \text{atMin}(a, w, igw) \wedge \text{ub-wish}(u, w) \wedge \text{reg}(u, ir)] \rightarrow ir < igw$$
12 Tipo: **Entità** | Relationship (cerchiare)

Nome:

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.buoni-non-suff] $\forall a, vb, t$

$$[\text{Acquisto}(a) \wedge \text{sum-buoni}(a, vb) \wedge \text{tot}(a, t) \wedge vb < t] \rightarrow [\exists c \text{ Carta}(c) \wedge \text{aq-car}(a, c)]$$
[V.buoni-suff] $\forall a, vb, t$

$$[\text{Acquisto}(a) \wedge \text{sum-buoni}(a, vb) \wedge \text{tot}(a, t) \wedge vb \geq t] \rightarrow [\neg \exists c \text{ Carta}(c) \wedge \text{aq-car}(a, c)]$$
14 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

Classe Acquisto

sum-buoni():Euro

• pre-cond: nessuna

• post-cond: $B = \{(b,i) \mid \text{aq-buono}(\text{this}, b) \wedge \text{importo}(b, i)\}$

$$\text{tot} = \sum_{(b,i) \in B} i$$

$$\text{Result} = \text{tot}$$

tot():Euro

• pre-cond: nessuna

• post-cond: $P = \left\{ pr \mid \begin{array}{l} \exists a, n, num, ist \quad \text{aq-art}(a, \text{this}, num) \wedge \text{aq-naz}(\text{this}, n) \\ \wedge \text{ist}(\text{this}, ist) \wedge p_min(a, num, n, ist, pr) \end{array} \right\}$

$$\text{Result} = \sum_{pr \in P} pr$$

Classe Articolo

p-minimo(pezzi:intero, n:Nazione, ist:DataOra):Euro

• pre-cond: $\exists \sigma, s \quad [\text{art-of}(\text{this}, \sigma) \wedge \text{of-sped}(\sigma, s) \wedge \text{sped-naz}(s, n)]$

• post-cond: $\text{Prezzi} = \left\{ (prO, prS) \mid \begin{array}{l} \exists \sigma, s, mi, m2, in \quad \text{art-of}(\text{this}, \sigma) \wedge \text{prezzo}(\sigma, prO) \wedge \text{of-sped}(\sigma, s) \\ \wedge \text{prezzo}(pr, prS) \wedge \text{sped-naz}(s, n) \wedge \text{min}(s, mi) \wedge \text{inizio}(\sigma, in) \\ [\exists m \max(s, m) \wedge mi \leq \text{pezzi} \leq m] \vee [\neg \exists m \max(s, m) \wedge mi \leq \text{pezzi}] \wedge \\ [\exists f \text{fine}(\sigma, f) \wedge in \leq ist \leq f] \vee [\neg \exists f \text{fine}(\sigma, f) \wedge ist \geq in] \end{array} \right\}$

$$\text{MinPre} = \min \left(\bigcup_{\substack{(prO, prS) \\ \in \text{Prezzi}}} \{prO + prS\} \right)$$

$$\text{Result} = \text{MinPre} \cdot \text{pezzi}$$

Risposta alla Domanda 2 (segue)

Tipi di Dato

Euro = Reale ≥ 0

Cell = $[0-9]\{10\}$

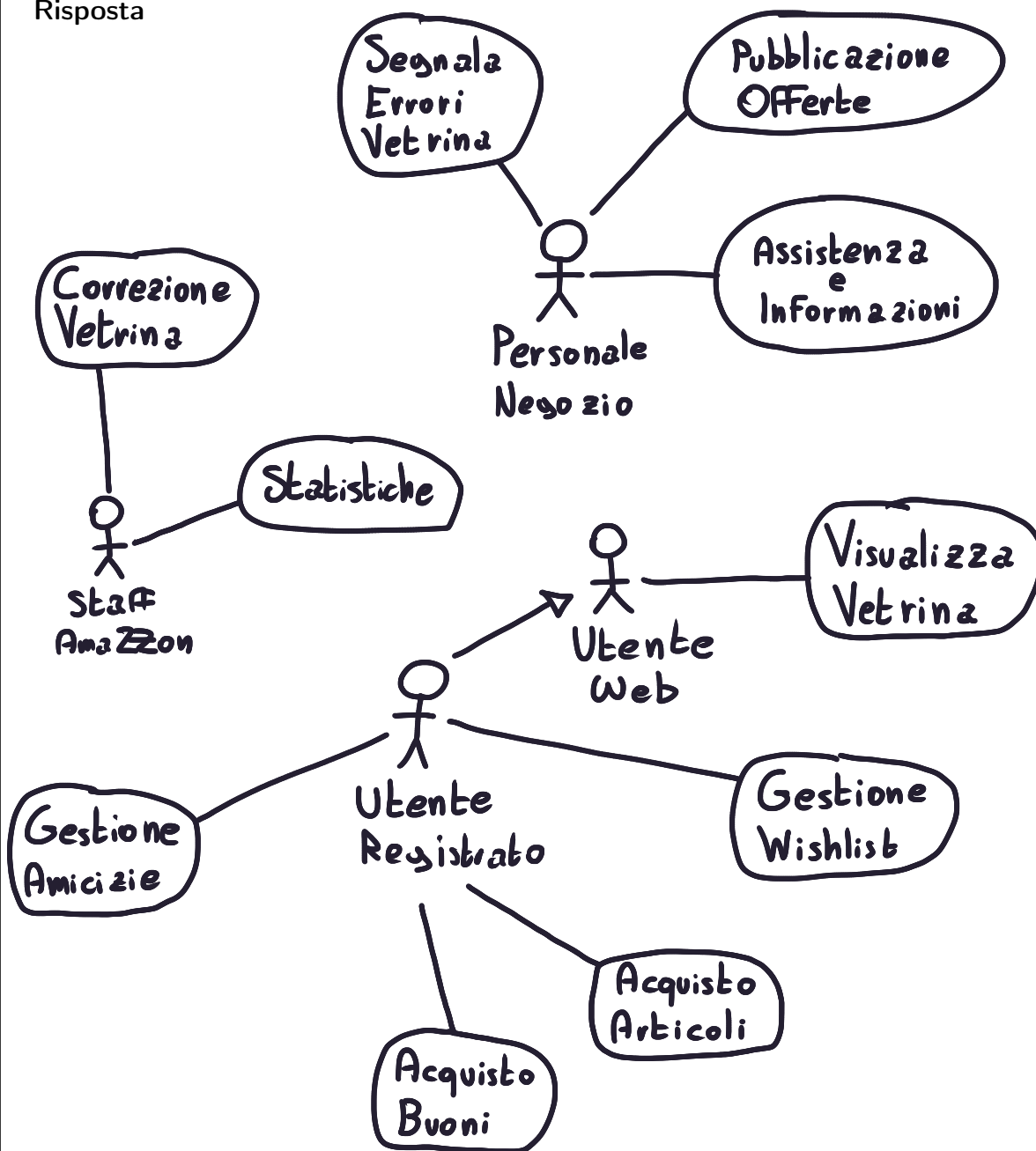
Email = $[a-zA-Z0-9]+\@[a-zA-Z0-9]+\.[a-z]\{2,6\}$

NumCarta = $[0-9]\{16\}$

Indirizzo = (via:Stringa, civico:Interzzo)

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: *Acquisto Articoli* (nome use-case)

Operazioni dello use-case:

compra(a:Articolo, p:Pezzi, n:Nazione, u:Utente): Acquisto

2 Specifica use-case: *Acquista Buoni* (nome use-case)

Operazioni dello use-case:

compra-buono(u:Utente, i:Euro): Buono

3 Specifica use-case: *Gestione Amicizie* (nome use-case)

Operazioni dello use-case:

richiedi(r:Richiesta, richiesto:Utente): Richiesta

accetta(r:Richiesta): Amicizia

rifiuta(r:Richiesta): Rifiutata

cancella(a:Amicizia)

4 Specifica use-case: Gestione Wishlist (nome use-case)

Operazioni dello use-case:

crea($n:String, p:Boolean$):Wishlist
 aggiungi($w:Wishlist, a:Articolo$)
 rimuovi($w:Wishlist, a:Articolo$)
 cambia_visibilita'($w:Wishlist$)

5 Specifica use-case: Statistiche (nome use-case)

Operazioni dello use-case:

paese_migliore($i:DataOra, f:DataOra$):Nazione[1..*]
 vendite_paese($i:DataOra, f:DataOra, n:Nazione$):Intero ≥ 0

6 Specifica use-case: Pubblicazione Offerte (nome use-case)

Operazioni dello use-case:

rimuovi_off($o:Offerta$)
 pubblica($p:Euro, i:DataOra, f:DataOra, a:Articolo, s:Spedizione[1..*]$):Offerta

7 Specifica use-case: Visualizza Vetrina (nome use-case)

Operazioni dello use-case:

visualizza_per_cat($c:Categoria, T:Tag[1..*], n:Nazione$):(Articolo, Euro)[0..*]
 articoli_moda():Articolo[0..*]

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

Operazioni dello use-case:

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

Operazioni dello use-case:

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

Operazioni dello use-case:

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

Operazioni dello use-case:

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

Visualizza Vetrina

$visualizza_per_cat(c: Categoria, T: Tag[1..*], n: Nazione) : (Articolo, Euro) [0..*]$

• pre-cond: nessuna

• post-cond:

$$A = \left\{ (a, pr) \mid \begin{array}{l} Articolo(a) \wedge cat_art(a, c) \wedge \\ \exists t \in T \quad tag_art(a, t) \wedge \\ p_min(a, 1, n, adesso, pr) \end{array} \right\}$$

Result = A

$articoli_moda() : Articolo [0..*]$

• pre-cond: nessuna

• post-cond:

$$A = \left\{ (a, \left(\frac{vmp - \frac{1}{2} - v\sigma}{v\sigma} \right) \cdot 100) \mid \begin{array}{l} Articolo(a) \\ \wedge vendite_periodo(a, adesso - '1 mese', adesso, v\sigma) \\ \wedge vendite_periodo(a, adesso - '3 mesi', adesso - '1 mese', vmp) \end{array} \right\}$$

Result = $\underset{(a,b) \in A}{argmax} (b)$

$vendite_periodo(a: Articolo, i: DataOra, f: DataOra): Intero \geq 0$

• pre-cond: $i < f \leq adesso$

• post-cond: $A = \{(aq, n) \mid aq_art(a, aq, n) \wedge \exists is \ is_t(aq, i) \wedge i \leq is \leq f\}$

Result = $\sum_{(aq, n) \in A} n$

Risposta alla Domanda 5 (segue)

Statistiche

$\text{paese_migliore}(i:\text{DataOra}, f:\text{DataOra}): \text{Nazione}[1..*]$

• pre-cond: $i < f \leq \text{adesso}$

• post-cond: $NV = \{ (n, v) \mid \text{Nazione}(v) \wedge \text{vendite_paese}(i, f, n) \}$

$$\text{Result} = \underset{(n, v) \in NV}{\text{argmax}}(v)$$

$\text{vendite_paese}(i:\text{DataOra}, f:\text{DataOra}, n:\text{Nazione}): \text{Intero} \geq 0$

• pre-cond: $i < f \leq \text{adesso}$

• post-cond: $V = \left\{ (aq, num) \mid \begin{array}{l} \text{Acquisto}(aq) \wedge aq_naz(aq, n) \wedge \exists is \text{ ist}(aq, is) \\ \wedge i \leq is \leq f \wedge \exists av \text{ aq_art}(aq, av, num) \end{array} \right\}$

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

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

```
CREATE DOMAIN Stringa AS varchar NOT NULL;
```

```
CREATE DOMAIN Euro AS Real CHECK(value ≥ 0);
```

```
CREATE DOMAIN Int-GZ AS Integer CHECK(value > 0);
```

```
CREATE DOMAIN Int-GEZ AS Integer CHECK(value ≥ 0);
```

```
CREATE DOMAIN NumCarta AS varchar ~ '[0-9]{16}'
```

```
CREATE DOMAIN Cell AS varchar ~ '[0-9]{10}'
```

```
CREATE TYPE Indirizzo AS (
  via: Stringa,
  civico: Int-GEZ );
```

```
CREATE TYPE Email AS varchar ~ '[a-zA-Z0-9]+@[a-zA-Z0-9]+\.[a-z]{2,6}';
```

■



Breve descrizione delle scelte effettuate durante la ristrutturazione

Fusione su Richiesta

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.richiesta2_istanti]

$\forall r [Richiesta(r) \rightarrow [\exists b accettazione(r,b) \leftrightarrow \exists i istRisp(r,i)]]$

[V.no_2_richieste_insieme_se_non_rifiutate]

da fare

[V.no_richieste-incrociate]

da fare

Risposta alla **Domanda 6** (segue)

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

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

1 Relazione Marca (nome) Derivante da: entità | relationship (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 accorpa le relazioni che implementano le seguenti relationship:

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

Attributi	<u>nome</u>	<u>id</u>	<u>marca</u>					
Domini	<u>Stringa</u>	<u>serial</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):

unique(nome, marca);

FK marca references Marca(nome);

La relazione accorpa le relazioni che implementano le seguenti relationship: mod-mar

3 Relazione Categoria (nome) Derivante da: entità | relationship (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 accorpa le relazioni che implementano le seguenti relationship:

4 Relazione Tau (nome) Derivante da: entità | relationship (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 accorpa le relazioni che implementano le seguenti relationship:

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

Attributi	<u>nome</u>	<u>desc</u>	<u>num</u>	<u>cod</u>	<u>modello</u>	<u>cat</u>		
Domini	<u>Stringa</u>	<u>Stringa</u>	<u>StringNum</u>	<u>serial</u>	<u>Integer</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 modello references Modello(id);

FK cat references Categoria(nome);

La relazione accorpa le relazioni che implementano le seguenti relationship: art-mod, cat-art

6 Relazione Tag-art (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>articolo</u>	<u>tag</u>						
Domini	Integer	Stringa						

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):

FK articolo references Articolo(cod);

FK tag references Tag(nome);

Articolo(cod) occorre in articolo

La relazione accorpa le relazioni che implementano le seguenti relationship:

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

Attributi	<u>nome</u>							
Domini	Stringa							

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:

8 Relazione Città (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>	<u>id</u>	<u>naz</u>					
Domini	Stringa	Serial	Stringa					

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):

unique(nome, naz);

FK naz references Nazione(nome);

La relazione accorpa le relazioni che implementano le seguenti relationship: naz-cit

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

Attributi	<u>nome</u>	<u>indirizzo</u>	<u>numero</u> *	<u>id</u>	<u>città</u>			
Domini	Stringa	Indirizzo	Cell	serial	Integer			

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):

FK città references Città(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: neg-cit

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

Attributi	<u>email</u>							
Domini	Email							

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:

11 Relazione neg-mail (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>neg</u>	<u>mail</u>						
Domini	Integer	Email						

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):

Fk mail references Mail(email); Mail.email ≤ mail
 Fk neg references Negozio(id); Negozio.id ≤ neg

La relazione accorpa le relazioni che implementano le seguenti relationship:

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

Attributi	<u>prezzo</u>	<u>inizio</u>	<u>fine</u> *	<u>id</u>	<u>neg</u>	<u>art</u>		
Domini	Euro	DateTime	DateTime	serial	Integer	Integer		

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

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

Fk neg references Negozio(id); check(inizio < fine);
 Fk art references Articolo(cod);

La relazione accorpa le relazioni che implementano le seguenti relationship: neg-off, art-off13 Relazione Spedizione (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>min</u>	<u>max</u> *	<u>prezzo</u>	<u>id</u>	<u>naz</u>	<u>off</u>		
Domini	Int-G2	Int-G2	Euro	serial	Stringa	Integer		

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(naz, off);
 Fk naz references Nazione(nome); check(min < max);
 Fk off references Offerta(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: off-sped, sped-naz14 Relazione Acquisto (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ist</u>	<u>id</u>	<u>utente</u>	<u>naz</u>	<u>car</u> *			
Domini	DateTime	serial	Integer	Stringa	NumCarb			

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):

Fk utente references Utente(id); Fk car ref Carta(Numero);
 Fk naz references Nazione(nome);

La relazione accorpa le relazioni che implementano le seguenti relationship: ut-az, az-naz15 Relazione Utente (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>	<u>neg</u>	<u>id</u>					
Domini	Stringa	DateTime	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):

La relazione accorpa le relazioni che implementano le seguenti relationship:

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

Attributi	<u>importo</u>	<u>inizio</u>	<u>fine</u>	<u>id</u>	<u>utente</u>	<u>ag *</u>		
Domini	<u>Euro</u>	<u>DateTime</u>	<u>DateTime</u>	<u>serial</u>	<u>Integer</u>	<u>Integer</u>		

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

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

check(inizio < Fine);

FK utente References Utente(id); FK ag References Acquisto(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ag-buono, ut-buono

17 Relazione ag-art (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ag</u>	<u>art</u>	<u>num</u>					
Domini	<u>Integer</u>	<u>Integer</u>	<u>Int_62</u>					

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

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

FK ag References Acquisto(id);

Acquisto.id < ag; FK art References Articolo(card);

La relazione accorpa le relazioni che implementano le seguenti relationship:

18 Relazione Carta (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>titolare</u>	<u>numero</u>	<u>scad</u>					
Domini	<u>Stringa</u>	<u>numCarta</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 enupla, di dominio):

La relazione accorpa le relazioni che implementano le seguenti relationship:

19 Relazione ut-car (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ut</u>	<u>car</u>						
Domini	<u>Integer</u>	<u>NumCarta</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):

FK ut ref Utente(id);

FK car ref Carta(numero); Carta.numero < car;

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:

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

Attributi	<u>istRic</u>	<u>istRis *</u>	<u>accettazione*</u>	<u>id</u>	<u>ric</u>	<u>inv</u>		
Domini	<u>DateTime</u>	<u>DateTime</u>	<u>bool</u>	<u>serial</u>	<u>Integer</u>	<u>Integer</u>		

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

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

check(istRis > istRic); fu inv ref Utente(id);

check(inv <> ric); fu ric ref Utente(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: inv, ric

17 Relazione Wishlist (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>nome</u>	<u>pub</u>	<u>id</u>	<u>ut</u>				
Domini	<u>Stringa</u>	<u>bool</u>	<u>serial</u>	<u>Integer</u>				

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

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

fu ut ref Utente(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ut-wish

18 Relazione ovwWish (nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>zv</u>	<u>w</u>	<u>prMin</u>	<u>istMin</u>				
Domini	<u>Integer</u>	<u>Integer</u>	<u>Euro</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 ennuola, di dominio):

FK zv ref Articolo(cod);

FK w ref Wishlist(id);

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 ennuola, 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 ennuola, 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.

[V.no.intersezione.pezzi]

Trigger on Insert o Update Spedizione

```
Error = SELECT *
FROM Spedizione s, Nazione n, Offerta o
WHERE s.<>new AND n.nome=new.naz
AND o.id=new.off AND
int4range(s.min,s.max)*int4range(new.min,new.max)<>'empty'
AND s.max<>NULL AND new.max<>NULL
```

Se Error e' vuoto, commit, altrimenti segnala errore.

[V.buono-usato-se-valido]

Trigger Insert o Update Buono

```
Error = SELECT *
FROM Acquisto aq
WHERE aq.id=new.aq
AND ( new.inizio ≥ aq.ist
OR new.fine ≤ aq.ist)
```

Se Error e' vuoto, commit, altrimenti segnala errore.

[V.buono.utente]

Trigger Insert o Update Buono

```
Error = SELECT *
FROM Utente ut, Acquisto aq
WHERE new.ut=ut.id
AND new.aq=aq.id
AND aq.utente <> ut.id
```

Se Error e' vuoto, commit, altrimenti segnala errore.

Risposta alla Domanda 7 (segue)

[V.acquisto-dopo-reg]

Trigger Insert Acquisto

```
Error = SELECT *  
        FROM Utente u  
        WHERE new.utente = u.id  
        AND u.reg > new.ist
```

Se Error e' vuoto, commit, altrimenti segnala errore.

[V.amicizia-dopo-reg]

Trigger Insert Richiesta

```
Error = SELECT *  
        FROM Utente u  
        WHERE (u.id = new.ric  
        OR u.id = new.inv)  
        AND new.istRic < u.reg
```

[V.cartta-utente]

Trigger on Insert Acquisto

controlla solo se new.car <> NULL

```
OK = SELECT *  
      FROM ut_car r  
      WHERE r.car = new.car  
      AND r.ut = new.utente
```

Se OK=Vuoto, errore.

[V.usata-se-valida]

Trigger on Insert Acquisto

```
Error = SELECT *  
        FROM Carta c  
        WHERE c.numero = new.car  
        AND c.scad < new.ist
```

`vendite_paese (n:Stringa, i:DateTime, f:DateTime):Int - GF2`

- pre-cond: $i < f \leq \text{NOW}()$
- post-cond $Q = \text{SELECT sum}(aq.art.num)$
FROM `Acquisto aq, aq_art`
WHERE `aq.naz = n AND aq.id = aq_art.aq`
AND `aq.ist ≥ i`
AND `aq.ist ≤ f`;

Result = Q

`paese_migliore (i:DateTime, f:DateTime):Stringa`

- pre-cond: $i < f \leq \text{NOW}()$
- post-cond: $\text{SELECT } n.\text{nome}, \text{vendite_paese}(n, i, f) \text{ as } c$
FROM `Nazione n`
WHERE $c = \text{SELECT max}(\text{vendite_paese}(n, i, f))$
FROM `Nazione n`)

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

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.]

