



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

CLIENTE

- Nome
- Cognome
- Nome utente: EMAIL
- IMPRINT

(NU) RICERCA

- INDIETRÀ: IMPRINT

CARTA

- Nome

FA-TTO-DONO

- Nome
- Cognome
- Nome utente: EMAIL

- Nome Massimo: L

RISTORANTE

- Nome
- Nome utente: EMAIL
- PARTITA IVA: PIVA
- CUCINA
- IMP_MILLI: Euro
- COSTO_COGNOMA: Euro
- OPERI_AP
- GIORNI_AP

CUCINA

- Nome: Scop

OPERE_AP

- DA_OPA
- A_OPA

GODI

- Nome: Scop

CATEGORIA

- Nome

PIRONOTTO

- Nome
- PREZZO

PIZZAIOLA

- INGREDIENTO: L

ORDINE

- RISTORANTE
- CLIENTE
- ISCODE_AP
- PIROTTI
- ISCODE_CNS
- PAGAMENTO: PAGAMENTO
- IMPRINT
- MESSAGGIO

DURATA

- ISDIF

ACCETTATE

- ISAC
- ISACONS_SIN

FEDERAL

- Voto: Voto
- DSC: STAVGA

LAVORAZIONE

- ISL LAU

PPONE

- ISL PPONE

IN_CNS

- ISL PRESA

CONSEGNA

- ISL CNS

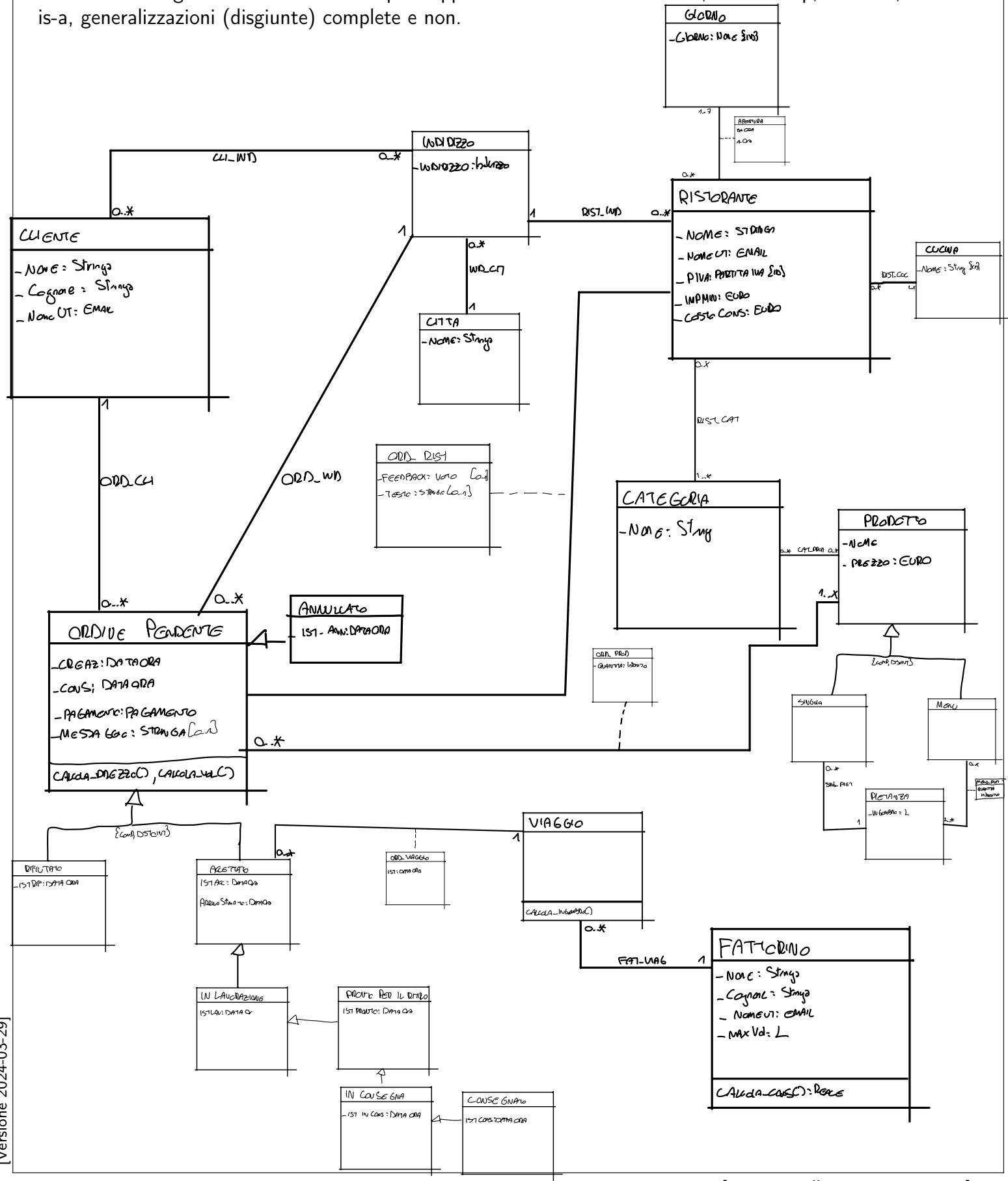
Risposta alla Domanda 1 (segue)

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

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

Diagramma ER

Produrre un diagramma ER concettuale per l'applicazione in termini di entità, relationship, attributi, relazioni is-a, generalizzazioni (disgiunte) complete e non.



Risposta alla Domanda 2 (segue)

Dizionario dei dati Per ogni entità e relationship del diagramma ER **[con]** attributi o vincoli:

- Definire il dominio e la molteplicità degli attributi (se diversa da (1,1))
- Definire eventuali vincoli esterni in logica del primo ordine estesa con teoria degli insiemi e semantica di mondo reale, usando il seguente alfabeto:
 - Un simbolo di predicato $E/1$ per ogni entità E .
Semantica di $E(x)$: x è una istanza di E .
 - Un simbolo di predicato $D/1$ per ogni dominio D .
Semantica di $D(x)$: x è un valore di D .
 - Un simbolo di predicato r/n ($n > 0$) per ogni relationship n -aria r .
Semantica di $r(x_1, \dots, x_n)$: x_1, \dots, x_n è una istanza di r .
 - Un simbolo di predicato $a/2$ per ogni attributo a di entità
Semantica di $a(x, v)$: uno dei valori dell'attributo a dell'istanza x è v .
 - Un simbolo di predicato $a/(n+1)$ per ogni attributo a di relationship n -aria.
Semantica di $a(x_1, \dots, x_n, v)$: uno dei valori dell'attr. a dell'istanza (x_1, \dots, x_n) della relat. è v .
 - Opportuni simboli di predicato (soggetti a *semantica di mondo reale*) per gestire confronti tra valori di domini numerici o comunque ordinati (tra cui $</2$, $\leq/2$, $>/2$, $\geq/2$).
 - Il predicato di uguaglianza $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso).
 - Opportuni simboli di costante (soggetti a *semantica di mondo reale*), tra cui *adesso*, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>ORDINE</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td>MESSEGAb</td> <td>STRUTT</td> <td>$C_{0,1}$</td> </tr> </tbody> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> [U. ORDINE.CREAZIONE] $\forall c_1, c_2$ $[ord(c_1) \wedge creaz(c_1, c_2) \wedge cons(c_2, c_1)] \rightarrow c_1 = c_2$ [U. ORDINE.DISTORANTE] $\forall c_1, R, b, con, do, a_0$ $[ord(c_1) \wedge cons(c_1, con) \wedge Ristorante(R) \wedge ord.con(R, c_1) \wedge$ $glob(b) \wedge da.con(c_1, b, do) \wedge hora(b, a_0)] \rightarrow do \leq con \wedge a_0$ [U. ORDINE.INDIREZ] $\forall c_1$ $[ord(c_1) \wedge utente(c_1) \wedge ord.con(c_1, c_2) \wedge individuo(c_2) \wedge ord.ind(c_2, c_1)] \rightarrow c_1 = individuo(c_2)$ 	attributo	dominio	moltep. (*)	MESSEGA b	STRUTT	$C_{0,1}$	<p>2 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <u>ORDINE</u></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> </table> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <ul style="list-style-type: none"> [U. ORDINE.ACCESSIONE] $\forall c_1, c_2$ $[Ristorante(c_1) \wedge creaz(c_1, c_2) \wedge ord.con(c_1, c_2)] \rightarrow c_1 = c_2$ [U. ORDINE.ACCESSIONE] $\forall a_1$ $[Acquisto(a_1) \wedge creaz(a_1, c_2) \wedge ord.con(a_1, c_2)] \rightarrow c_1 = a_1$ [U. ORDINE.IN LAU] $\forall a_1$ $[InLaudo(a_1) \wedge creaz(a_1, c_2) \wedge ord.con(a_1, c_2)] \rightarrow a_1 = c_2$ 	attributo	dominio	moltep. (*)
attributo	dominio	moltep. (*)								
MESSEGA b	STRUTT	$C_{0,1}$								
attributo	dominio	moltep. (*)								

<p><input type="checkbox"/> Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...ODDINe.....</p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>[V.ODDINe. PROPR]</p> <p>$\forall P, I^P, IC$ $[PROPR(P) \wedge IS^P PROPR(P, I^P) \wedge IS^I PROPR(I^P, IC)] \rightarrow IC \leq I^P$</p> <p>[V. ODDINe. CAUSE GVA]</p> <p>$\forall IC, II^C, IP$ $[In.CauseGVA(IC) \wedge IS^C In.GVA(IC, II^C) \wedge IS^IP In.GVA(IC, IP)] \rightarrow IP \leq IC$</p> <p>[V. ODDINe. CONSEGUATO]</p> <p>$\forall C, IC, II^C$ $[Conseguato(C) \wedge IS^C Conseguato(C, IC) \wedge IS^{II^C} Conseguato(C, II^C)] \rightarrow IC \leq II^C$</p>	attributo	dominio	moltep. (*)				<p><input type="checkbox"/> Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...ODDING.....</p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>[V.ODDING. VIA GVA]</p> <p>$\forall O, V, IA, ISU$ $[Accettato(O) \wedge IS^O Accettato(O, IA) \wedge IS^IA Accettato(O, IA, ISU)] \rightarrow IA \leq ISU$</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)											
attributo	dominio	moltep. (*)											

<p><input type="checkbox"/> Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...ADD_DIST.....</p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>[V. ADD_DIST. FEEDBACK]</p> <p>$\forall O, D, F, IC$ $[ODDINe(O) \wedge Distante(D) \wedge F FEEDBACK(O, D, F)] \rightarrow$ $Conseguato(F)$</p>	attributo	dominio	moltep. (*)				<p><input type="checkbox"/> Tipo: Entità Relationship (cerchiare)</p> <p>Nome: ...FATTO_MAR...</p> <table border="1"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table> <hr/> <p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p>[V. FATTO_MAR. VAL]</p> <p>$\forall F, V, O, MU$ $[FATTO_MAR(F) \wedge Valore(V) \wedge Freq(V, F) \wedge Accettato(O) \wedge$ $ordine(O, V) \wedge Marca(F, V)] \rightarrow (\exists k \in K \text{ Accettato}(O, k) \wedge k \leq MU)$</p>	attributo	dominio	moltep. (*)			
attributo	dominio	moltep. (*)											
attributo	dominio	moltep. (*)											

<p>7 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <i>ODDINE</i></p> <table border="1" data-bbox="251 263 820 298"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> </table>	attributo	dominio	moltep. (*)	<p>9 Tipo: Entità Relationship (cerchiare)</p> <p>Nome: <i>Anagrafica</i></p> <table border="1" data-bbox="825 263 1538 298"> <thead> <tr> <th>attributo</th> <th>dominio</th> <th>moltep. (*)</th> </tr> </thead> </table>	attributo	dominio	moltep. (*)
attributo	dominio	moltep. (*)					
attributo	dominio	moltep. (*)					
<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p><i>[U-ODDINE-PREZZO]</i></p>	<p>(*) solo se diversa da (1,1)</p> <p>Vincoli:</p> <p><i>[U-Anagrafica-Gesito]</i></p>						

8	Tipo: Entità Relationship (cerchiare)		
Nome:			
attributo	dominio	moltepl. (*)	
<hr/>			
(*) solo se diversa da (1,1)			
Vincoli:			

10	Tipo: Entità Relationship (cerchiare)		
Nome:			
attributo	dominio	moltepl. (*)	
<hr/>			
(*) solo se diversa da (1,1)			
Vincoli:			

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

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

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

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

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

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

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

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

15 Tipo: Entità | Relationship (cerchiare)

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

17 Tipo: Entità | Relationship (cerchiare)

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

16 Tipo: Entità | Relationship (cerchiare)

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

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

Vincoli:

18 Tipo: Entità | Relationship (cerchiare)

Nome:

attributo	dominio	moltep. (*)
-----------	---------	-------------

(*) 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

TIPPI DI DATO

Euro: Intero

Porta lus: String secondo REGEX

String: STRINGA secondo REGEX

L = Intero > 0

Voto = 0..10

PAGAMENTO = econ = { "contanti alla consegna", "carta al momento dell'ordine" }

Indirizzo = { via: String, nro: Intero, CAP: CAP }

CAP: STRINGA secondo REGEX

OPERAZIONI DI CLASSE

-CLASSE ORDINE

- CALCOLA_PREZZO(): Euro

- CALCOLA_VOLUME(): L

- CALCOLA_NDEZZO(): Euro

PDE: Nessuna

Post: SIA $P = \{ (P_i, q_i) \mid \begin{cases} \text{Prodotto}(P_i) \wedge \text{ord_prod}(ths, P_i) \wedge \text{Quantita}'(ths, P_i, q_i) \\ \wedge \text{Prezzo}(P_i, P_i) \end{cases} \}$

$$\text{Tot Prod} = \sum_{(P_i, q_i) \in P} q_i$$

SIA CC | Distintivo(CC) \wedge ord_dest(ths, P) \wedge CostCost(CC)

$$\text{RESULT} = \text{Tot Prod} + CC$$

- CALCOLA_VOLUME(): L

PDE: Nessuna

Post: SIA $IspSum = \{ (I, q) \mid \begin{cases} \text{Prodotto}(P_i) \wedge \text{ord_prod}(ths, P_i) \wedge \text{Quantita}'(ths, P_i, q_i) \\ \wedge \text{Sintesi}(P_i) \wedge \text{Piemonte}(P_i) \wedge \text{Singl. Prez}(P_i, P_i) \wedge \text{NGombro}(P_i, I) \end{cases} \}$

$$\text{Tot Prod Sing} = \sum_{I \in \text{IspSum}} \sum_{q \in Q}$$

$S_A \mid PM_{cost} = \{ (I, qP, qP2) \mid \begin{cases} \text{Produtt.(P)} \wedge \text{ord_prod}(ths, P) \wedge \text{Quantita}'(ths, P, qP) \\ \wedge \text{Menù}(P) \wedge \text{Piemonte}(P) \wedge \text{Menù_Prez}(P, P2) \wedge \text{Quantita}'(P, P2, qP2) \wedge \text{NGombro}(P2, I) \end{cases} \}$

$$\text{Tot Prod Men} = \sum_{\substack{(I, qP, qP2) \\ \in PM_{cost}}} \sum_{P \in \{P1, P2\}} \sum_{P \in \{P1, P2\}}$$

$$\text{RESULT} = \text{Tot Prod Sing} + \text{Tot Prod Men}$$

Risposta alla Domanda 2 (segue)

-CALCOLA-MUOVRPO(C): L

PRE: NCSSNA

$$\text{POST: } SIA \text{ Ins} O = \left\{ i \mid \begin{array}{l} \text{VA_GGIO}(this) \wedge \text{Acquistato}(o) \wedge \text{ord}_i \text{VA_GGIO}(o, this) \wedge \\ (\exists i' \text{ CALCOLA-MUOVRPO}(o, i')) \end{array} \right\}$$

$$Result = \sum_{i \in InsO} i$$

-CALCOLA_CON($In: OFFIA, Fin: D$): RGALE

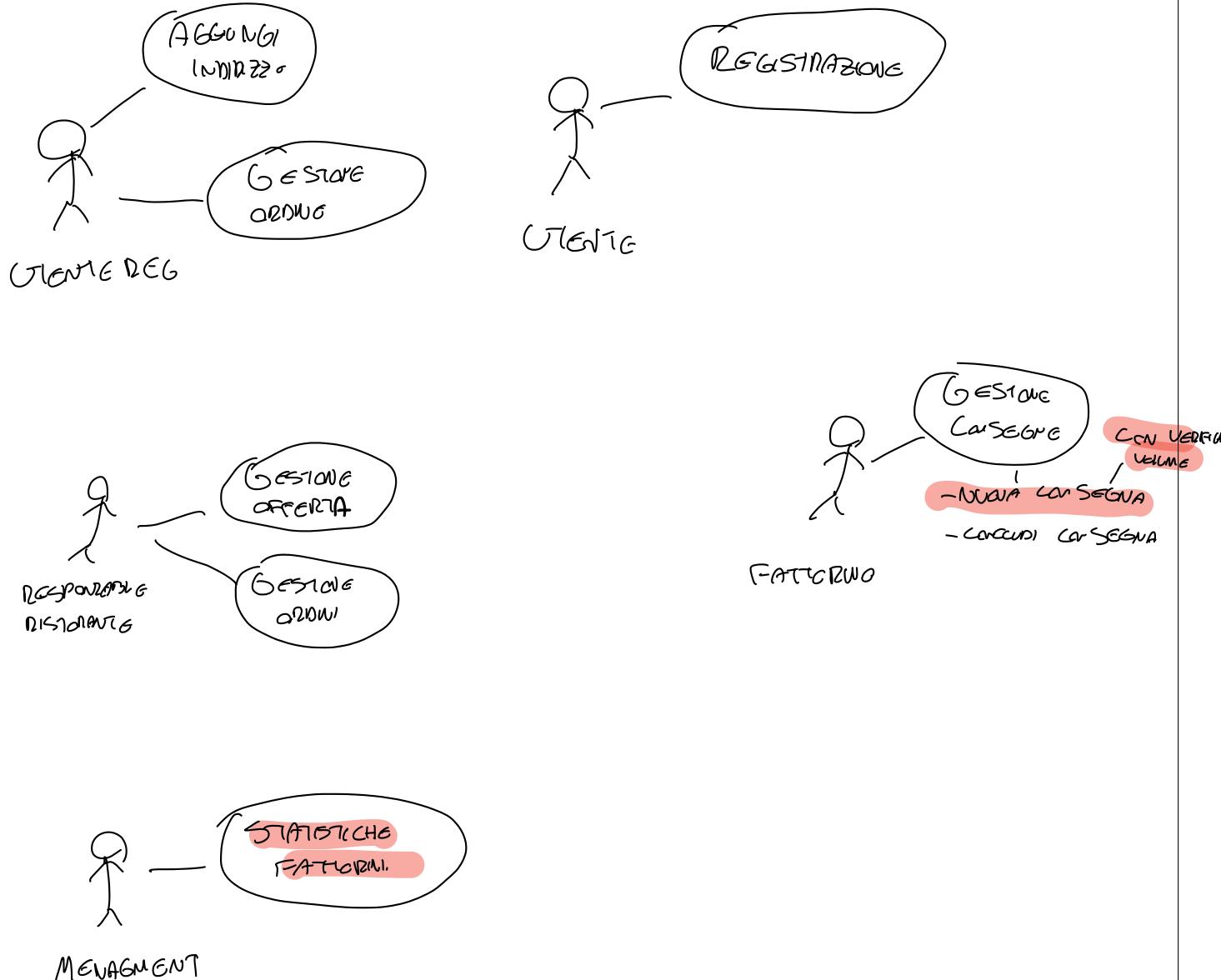
$$SIA \text{ Caus} O = \left\{ c \mid \begin{array}{l} \text{Consegna}(c) \wedge (In \leq \text{ISICONS} \leq Fin) \wedge \text{AcquistaStando}(c, AS) \\ \wedge \text{ISICONS} \leq AS \end{array} \right\}$$

$$SIA \text{ TotCaus} = \left\{ c \mid \text{Consegna}(c) \wedge (In \leq ISICONS \leq Fin) \right\}$$

$$Result = \frac{|CausO|}{|TotCaus|} \cdot 100$$

Domanda 3 (5 minuti; 10 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML degli use-case che definisca ad alto livello tutte le funzionalità richieste al sistema.

Risposta



Domanda 4 (10 minuti) Proseguire la fase di Analisi Concettuale dei requisiti definendo le operazioni degli use-case.

In particolare, per ogni use-case definito nella risposta alla Domanda 3 definire la **segnatura** di tutte le operazioni che lo compongono, in termini di nome dell'operazione, nomi e dominio concettuale degli argomenti, dominio concettuale dell'eventuale valore di ritorno.

1 Specifica use-case: REGISTRAZIONE..... (nome use-case)

Operazioni dello use-case:

REGISTRAZIONE_UTENTE(*N*: STRINGA, *UT*: EMAIL, *C*: STRINGA) : CLIENTE

REGISTRAZIONE_FATTORINO(*N*: STRINGA, *UT*: EMAIL, *C*: STRINGA, *M*: L) : FATTORINO

REGISTRAZIONE_RISPARMIO(*N*: STRINGA, *UT*: EMAIL, *Piu*: PARTIVA, *IM*: Euro, *CC*: Euro, *IN*: NUOVO, *CUC*: CCONVALORE) : RISPARMIO

2 Specifica use-case: Aggiungi -> INDIRIZZO..... (nome use-case)

Operazioni dello use-case:

Aggiungi -> INDIRIZZO(*I*: INDIRIZZO, *C*: CITTA)

3 Specifica use-case: GESTIONE ORDINE..... (nome use-case)

Operazioni dello use-case:

- NUOVO(*Cors*: DATAORA, *P*: PA GAGLIETTO, *M*: STANZA LO 13, *R*: RISTORANTE, *PR*: Ins{PREZZO, QUANTITA}) ,

I: INDIRIZZO) : ORDINE_FORNITORE

- MONITORA(*O*: ORDINE)

- ANNULLA(*C*: ORDINE) : Annulla

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

Operazioni dello use-case:

MUOVA_CATEGORIA (n: STRINGA, p: PRODOTTO [1..*]): CATEGORIA

RIMUOVI_CATEGORIA (c: CATEGORIA)

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

Operazioni dello use-case:

ACCETTA (o: ORDINE): ACCETTATO

RIPUNTA (o: ORDINE): ACCETTATO

MODIFICA_STATO (o: ORDINE)

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

Operazioni dello use-case:

MUOVA_CONSEGNA (o: ACCETTATO): IN CONSEGNA

CONCLUDI_CONSEGNA (o: IN CONSEGNA): CONSEGNAUTO

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

Operazioni dello use-case:

STATISTICHE_FATTORI (IN: DATA, FIN: DATA): [FATTORI, REZ] [0..*]

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

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

Risposta

- NUOVA_CONSEGNA(α : Accettato): In consegna
- STATISTICHE_FATTORINI(In : DATA, Fin : DATA): [FATTORINI, REAZ] [α_{\dots}]

- NUOVA_CONSEGNA(α : Accettato, v : VIA GLO):

PDE: Accettato(α) \wedge Anagrafica(α) \rightarrow In consegna(α)

SIA v UNOGGIO DEL MUNDO CHE INVOCA IL CASO
FATTORINI(α) \wedge MAXVAL(α, m) \wedge VIAGGIO(α, v) \wedge FATTORIA(α, n) \wedge
 $(\exists k \in \text{CALCOLA-INGRESSO}(\alpha, m)) \wedge (\exists k \in \text{CALCOLA-VIAGGIO}(\alpha, k)) \rightarrow k \leq m - 1$

Post

$$R = \left(\begin{array}{l} \text{VIAGGIO}(\alpha) \wedge \text{ADESSO}(\alpha) \wedge \text{ODD VIAGGIO}(\alpha, v) \wedge \text{STATO}(\alpha, v, \alpha) \\ \text{ODD VIAGGIO}(\alpha, v) \end{array} \right)$$

$$\text{Result} = R$$

- STATISTICHE_FATTORINI(In : DATA, Fin : DATA): [FATTORINI, REAZ] [α_{\dots}]

PDE: $v \leq \text{FIN}$

Post:

$$R = \left\{ F, OC \right\} \mid \left\{ \begin{array}{l} \text{FATTORINO}(F) \wedge \left(\exists \alpha \in \text{CALCOLA-COS}(F, IN, FIN, \alpha) \right) \end{array} \right\}$$

Risposta alla Domanda 5 (segue)

2 Progettazione della base dati e delle funzionalità

Domanda 6 (20 minuti; 30 minuti al massimo) Iniziare la fase di progettazione logica della base di dati decidendo il DBMS da utilizzare e ristrutturando lo schema ER concettuale, il dizionario dei dati e i vincoli esterni. In particolare:

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

Descrivere brevemente le principali scelte effettuate.

DBMS da utilizzare ... *PostgreSQL*

Corrispondenza tra domini concettuali e domini supportati dal DBMS

```

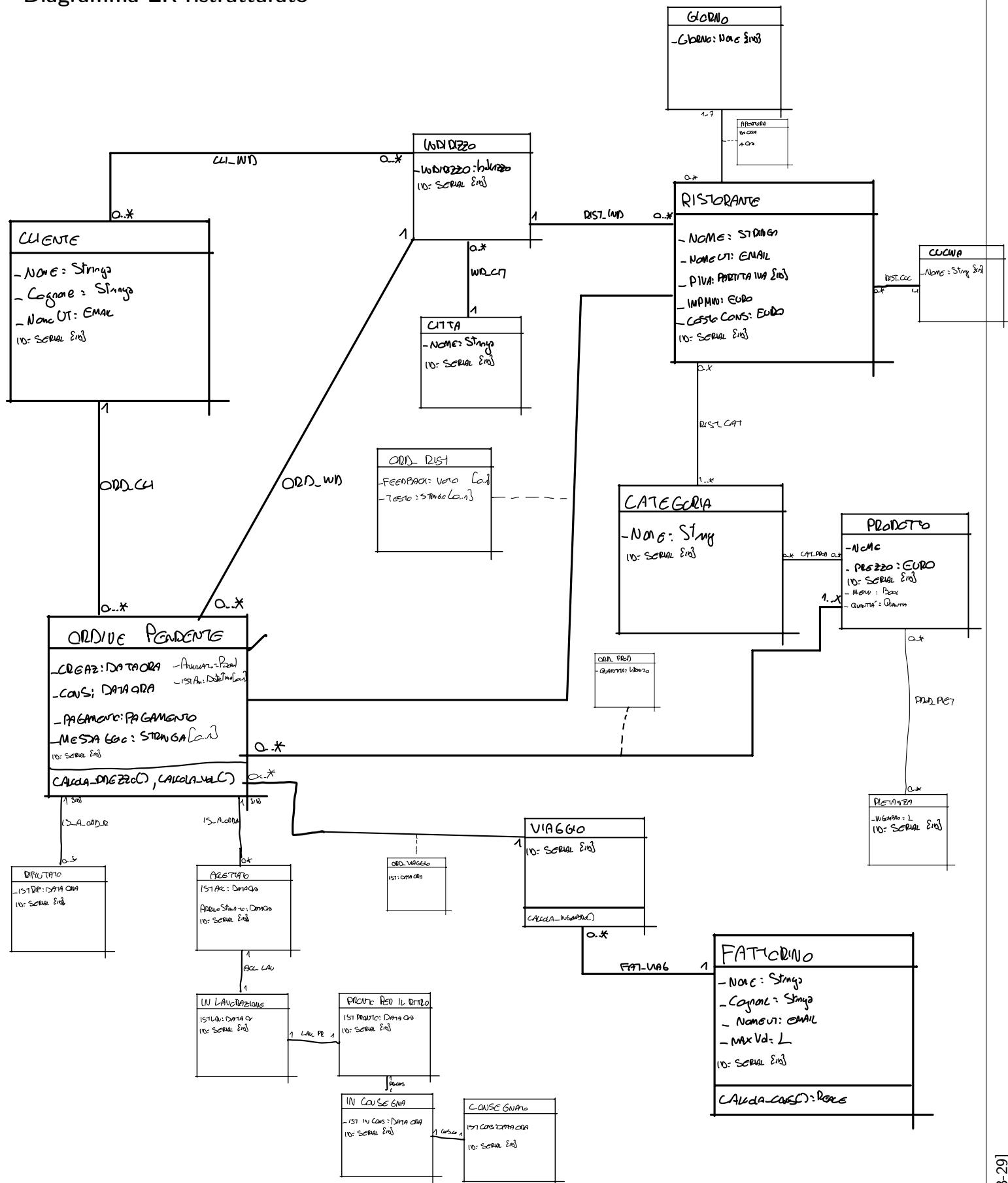
CREATE DOMAIN STRINGA AS VARCHAR CHECK (VALUE NOT NULL)
CREATE DOMAIN INTERO AS INTEGER CHECK (VALUE NOT NULL AND VALUE > 0)
CREATE DOMAIN INTERO AS INTEGER CHECK (VALUE NOT NULL AND VALUE >= 0)

CREATE DOMAIN EURO AS INTEGER
CREATE DOMAIN PARITATIVA AS STRING CHECK (Value <= 100)
CREATE DOMAIN DECIMAL AS STRING CHECK (Value <= 100)
CREATE DOMAIN VOTO AS INTEGER CHECK (Value <= 100)
CREATE DOMAIN CAP AS STRING CHECK (Value <= 100)
CREATE TYPE PAGAMENTO AS ENUM ('CARTA DI CREDITO', 'CARTA AL MONTANTE DEL DIRETTORE')
CREATE TYPE INDIRIZZO AS (NAZIONE STRING, PROVINCIA STRING, CITTÀ STRING, VIA STRING, NUMEROINTERO, CAP STRING)

CREATE DOMAIN QUANTIA AS INTEGER (DEFAULT = 1)

```

Diagramma ER ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

Ho fuso le generalizzazioni sui Prodotti, quindi il menu' Aggiornamento

Unico

Ho disc. la gen tra RF e Accettato

Ho fuso Anagrafe Dentro Cogn

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. ORDING. DIFUSO/ACC]

$\forall o, p, A$
 $[Ordine(o) \wedge Riferito(o) \wedge Accettato(A) \wedge IsA_ord(o, p)] \rightarrow \exists A_ord(o, A)$

[V. ORDINE. ACC/RIFORTA]

$\forall o, p, A$
 $[Ordine(o) \wedge Riferito(o) \wedge Accettato(A) \wedge IsA_ord(o, p)] \rightarrow \exists A_ord(o, A)$

[V. Annullato. TRUE]

$\forall o$
 $[Ordine(o) \wedge Annullato(o, "true") \rightarrow \exists Istanza(o, Nuc)}$

[V. Annullato. FALSE]

$\forall o$
 $[Ordine(o) \wedge Annullato(o, "false") \rightarrow \exists Istanza(o, Nuc)}$

[V. VIAGGIO. ORDINE]

$\forall v, o, A$
 $[Viaggio(v) \wedge Ordine(o) \wedge Ordine(v, o) \wedge Accettato(A)]$
 $\rightarrow IsA_ord(o, A)$

[V. VIAGGIO. DIFUSO]

$\forall v, o, p$
 $[Viaggio(v) \wedge Ordine(o) \wedge Difusivo(o) \wedge IsA_ord(o, p)]$
 $\rightarrow \exists A_ord(o, p)$

[V. PRODOTTO. MENU]

$\forall p$
 $[Prodotto(p) \wedge Menu[p, "true"] \rightarrow \exists Quantita(p, Nuc)]$

[V. PRODOTTO. NO MENU]

$\forall p$
 $[Prodotto(p) \wedge Menu[p, "false"] \rightarrow \exists Quantita(p, Nuc)]$

[V. Annullato. IST]

$\forall o, i, c, IA$
 $[Ordine(o) \wedge Annullato(o, "true") \wedge IstAnnullato(o, IA) \wedge IsCognato(c, IA)] \rightarrow \exists Istanza(o, Nuc)$

[V. ANNULLATO. IN LAU]

$\forall o, A, IN$
 $[Ordine(o) \wedge Annullato(o, "true") \wedge Accettato(A) \wedge IsA_ord(o, A)]$
 $\wedge InLaudozione(IN) \rightarrow \exists AccLaudo(A, IN)]$

[V. IN LAUDAZIONE. NO AN]

$\forall IN, A, O$
 $[(InLaudazione(IN) \wedge Accettato(A) \wedge AccLaudo(A, IN) \wedge Ordine(O) \wedge IsA_ord(o, A))]$
 $\rightarrow Annullato(o, "false")]$

Risposta alla Domanda 6 (segue)

SONO DA DISCUTERE I VIGLI MODIFICATI.

[V. ORDINE.DIFERITO]

[V. ORDINE.ACCEPATO]

[V. ORDINE.[N LAW]]

[V. ORDINE.ACCEPATO]

[V. ORDINE.[N LAW]]

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 <u>CLEVER</u> (nome)	Derivante da: <u>entità</u> relationship (cerchiare)
--	--

Attributi <u>id</u> Nome Cognome Nome Ut						
--	--	--	--	--	--	--

Attributi <u>id</u> Nome Cognome Nome Ut						
--	--	--	--	--	--	--

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

2 Relazione <u>INDRIZZO</u> (nome)	Derivante da: <u>entità</u> relationship (cerchiare)
--	--

Attributi <u>id</u> <u>INDRIZZO</u> Città						
---	--	--	--	--	--	--

Attributi <u>id</u> <u>INDRIZZO</u> Città						
---	--	--	--	--	--	--

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: città references città(id)

La relazione accorda le relazioni che implementano le seguenti relationship: IND_CIT

3 Relazione <u>CITTA</u> (nome)	Derivante da: <u>entità</u> relationship (cerchiare)
---------------------------------------	--

Attributi <u>id</u> <u>NOME</u>						
-------------------------------------	--	--	--	--	--	--

Attributi <u>id</u> <u>NOME</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: uore references uore(id)

fk: INDRIZZO references INDRIZZO(id)

La relazione accorda le relazioni che implementano le seguenti relationship:

4 Relazione <u>UORE</u> (nome)	Derivante da: <u>entità</u> relationship (cerchiare)
--------------------------------------	--

Attributi <u>id</u> <u>INDRIZZO</u>						
---	--	--	--	--	--	--

Attributi <u>id</u> <u>INDRIZZO</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: INDRIZZO references INDRIZZO(id)

UNIQUE (PIVA)

La relazione accorda le relazioni che implementano le seguenti relationship: PIVA

6 Relazione	<u>Globo</u> (nome)	Derivante da: <u>entità</u> <u>relationship</u> (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 relationship:

7 Relazione	<u>Apertura</u> (nome)	Derivante da: <u>entità</u> <u>relationship</u> (cerchiare)
Attributi	<u>RISTORANTE</u> <u>Globo</u> <u>DA Ora</u> <u>A Ora</u>	
Domini	<u>Intero</u> <u>Stringa</u> <u>TIME</u> <u>TIME</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):

ck: $(DA\ Ora \leq A\ Ora)$ FK: RISTORANTE Resto Ristorante(C)
 FK: Globo Resto Globo(C)

La relazione accorda le relazioni che implementano le seguenti relationship:

8 Relazione	<u>Cocina</u> (nome)	Derivante da: <u>entità</u> <u>relationship</u> (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 relationship:

9 Relazione	<u>RISI_COC</u> (nome)	Derivante da: <u>entità</u> <u>relationship</u> (cerchiare)
Attributi	<u>RISTORANTE</u> <u>CICLIA</u>	
Domini	<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):

fk: RISTORANTE Resto Ristorante(C)
 fk: CICLIA Resto Ciclia(C)

La relazione accorda le relazioni che implementano le seguenti relationship:

10 Relazione	<u>CATEGORIA</u> (nome)	Derivante da: <u>entità</u> <u>relationship</u> (cerchiare)
Attributi	<u>ID</u> <u>Nome</u>	
Domini	<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):

La relazione accorda le relazioni che implementano le seguenti relationship:

11 Relazione <u>RIS1_CAT</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>RISORSA</u> <u>CATEGORIA</u>	
Domini <u>Integ</u> <u>Integ</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: RISORSA ref. RISORSA(n)

fk: CATEGORIA ref. CATEGORIA(n)

La relazione accorda le relazioni che implementano le seguenti relationship:

12 Relazione <u>PRODOTTI</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> <u>Nome</u> <u>Prezzo</u> <u>Metro</u> <u>Quantita</u>	
Domini <u>Serial</u> <u>Stringa</u> <u>euro</u> <u>Piccol</u> <u>Quantiq</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):

Check (Metro=False AND Value QUANTITA=1)

La relazione accorda le relazioni che implementano le seguenti relationship:

13 Relazione <u>CAT_PRCH</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>CATEGORIA</u> <u>PRODOTTO</u>	
Domini <u>Integ</u> <u>Integ</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: CATEGORIA ref. CATEGORIA(n)

fk: PRODOTTO ref. PRODOTTO(n)

La relazione accorda le relazioni che implementano le seguenti relationship:

14 Relazione <u>PICTURA</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ID</u> <u>INFORMATIVO</u>	
Domini <u>Serial</u> <u>L</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 relationship:

15 Relazione <u>PROD_PIC</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>PRODOTTO</u> <u>PICTURA</u>	
Domini <u>Integ</u> <u>Integ</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: PRODOTTO

fk: PICTURA

La relazione accorda le relazioni che implementano le seguenti relationship:

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

Attributi	<u>ID</u>	CREAZ	CONS:	PAGA.	MESS.*	ANNULLA	ISTANZA*	CLIENTE
Domini	<u>SERIAL</u>	DATETIME	DATETIME	PAGAMENTO	STANGA	BOOL	DATETIME	INDRIZZO

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: CREA
 CHECK (CREAZ >= CONS)
 CHECK (ANNULLA = TRUE AND Istanza NOT NULL OR
 ANNULLATO = FALSE AND Istanza NULL)

La relazione accorda le relazioni che implementano le seguenti relationship: ... ADDCLI, ... ADDVN

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

Attributi	<u>ORDINE</u>	<u>DISTANZA</u>	FEEDBACK*	TEST*				
Domini	Integer	Float	Voto	STANGA				

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: ORDINE
 PK: DISTANZA

La relazione accorda le relazioni che implementano le seguenti relationship:

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

Attributi	<u>ORDINE</u>	<u>ID</u>	<u>LISTA</u>					
Domini	Integer	<u>SERIAL</u>	DATETIME					

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: ORDINE
 UNICO (ID)

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

19 Relazione ACCETTAZIONE.....(nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ORDINE</u>	<u>ID</u>	<u>LISTA</u>	<u>ACCETTAZIONE</u>				
Domini	Integer	<u>SERIAL</u>	DATETIME	DATETIME				

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 ACCETTAZIONE > LISTA
 FK: ORDINE
 UNICO (ID)

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

20 Relazione IN.LAVORAZIONE ..(nome) Derivante da: entità | relationship (cerchiare)

Attributi	<u>ORDINE</u>	<u>LISTA</u>	<u>ID</u>					
Domini	Integer	DATETIME	<u>SERIAL</u>					

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

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

UNICO (ID)
 PK: ORDINE

La relazione accorda le relazioni che implementano le seguenti relationship:

16 Relazione <u>PDO</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>codice</u> <u>id</u> <u>CSR_PDO</u>	
Domini <u>Integro</u> <u>Serial</u> <u>RangeInt</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):

valore (n)

FK: ODDNE

La relazione accorda le relazioni che implementano le seguenti relationship: Wn

17 Relazione <u>In_Cassegno</u> ...(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>codice</u> <u>id</u> <u>CSR_Cass</u>	
Domini <u>Integro</u> <u>Serial</u> <u>RangeInt</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):

valore (n)

FK: ODDNE

La relazione accorda le relazioni che implementano le seguenti relationship:

18 Relazione <u>Cars_Grafo</u> ...(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>codice</u> <u>id</u> <u>CSR_Cars</u>	
Domini <u>Integro</u> <u>Serial</u> <u>RangeInt</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):

valore (n)

FK: ODDNE

La relazione accorda le relazioni che implementano le seguenti relationship:

19 Relazione <u>...Fattura</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>id</u> <u>Nome</u> <u>Colore</u> <u>Valore UT</u> <u>Max Val</u>	
Domini <u>Serial</u> <u>Stringa</u> <u>Stringa</u> <u>Small</u> <u>L</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 relationship:

20 Relazione <u>...Viaggio</u>(nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>id</u> <u>Fattura</u>	
Domini <u>Serial</u> <u>Integro</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: Fattura

La relazione accorda le relazioni che implementano le seguenti relationship: Fatt_ViaG

16 Relazione .ORD_VIA60.. (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ORDNO</u> <u>VIA60</u> <u>ST_VIA6</u>	
Domini <u>Integ</u> <u>Integ</u> <u>DATA-TM</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: ORDNO REF

Fk: VIA60

La relazione accorda le relazioni che implementano le seguenti relationship:

17 Relazione .ORD_PDO.. (nome)	Derivante da: entità relationship (cerchiare)
Attributi <u>ORDNO</u> <u>PDO-TC</u> <u>QUANTIT</u>	
Domini <u>Integ</u> <u>Integ</u> <u>Integ</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 relationship:

18 Relazione (nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

19 Relazione (nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

20 Relazione (nome)	Derivante da: entità relationship (cerchiare)
Attributi	
Domini	

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

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

La relazione accorda le relazioni che implementano le seguenti relationship:

Ulteriori vincoli esterni

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

VINCOLI DA IMPLEMENTARE:

[V. ORDINE. PREZZO]

[V. ORDINE. INDIRIZZO]

[V. ORDINE. RISTORANTE]

T. ORDINE. PREZZO

IN SERVIMENTO: ORDINE

PDC OPERAZIONE

WITH TOTALE = (SELECT CALCOLA_PREZZO (a.ID)
FROM ORDINE a
WHERE a.ID = NEW (a.ID))

ISVALID := NOT EXISTS (SELECT *
FROM ORDINE a, DISTANTE C R, ORD RST OR
WHERE a.ID = NEW (a.ID) AND a.RISTORANTE = R.ID
AND a.ORDINE = R.ID AND R.IMPMINLTOTALE)

IF ISVALID:
COMMIT

ELSE:
GENERA ERRORE
REVERT

T. ORDINE. INDIRIZZO

IN SERVIMENTO IN ORDINE

PDC OPERAZIONE

ISVALID := NOT EXISTS (SELECT 1
FROM ORDINE a, CLIENTE c, INDIRIZZO i, CLEIND CI
WHERE a.ID = NEW (a.ID) AND a.CLIENTE = c.ID AND c.CLIENTE = CI.ID
AND CI.INDIRIZZO = i.ID AND a.INDIRIZZO = i.ID)

IF ISVALID:
COMMIT ELSE: ERRORE E REVERT

Risposta alla Domanda 7 (segue)

$\forall c, r, b, co, do, ao$

$[ordine(c), cors(c, co) \wedge Ristorante(r), ordine(c, r) \wedge$
 $giorno(c) \wedge data(c, do) \wedge ora(c, do)] \rightarrow do \leq co \leq ao$

T. ORDINE. RISTORANTE

INSGRIMENTO N ORDINE

PRE OPERAZIONE

isValue := Cexists (SELECT *

FROM ORDINE O, DBO.RST R, RISTORANTE R, APERTURA A

WHERE O.ID = NEW.ID AND O.ID = OLE.ORDINE AND O.RISTORANTE = R.ID
 AND A.RISTORANTE = R.ID AND (A.DARA <= EXTRACT(HOUR FROM O.CORS) <= A.AORA)
 AND (EXTRACT(DAY FROM O.CORS) = A.GIORNO))

;} ISVALID;

comm

else: GENERA ERRORE

REVERT

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

OPERAZIONI DI CLASSE

CLASSE ORDINE

-CALCOLA_PREZZO (in: integer): Euro

Mandizza in Q LA QUERY:

```
SELECT SUM ((P. PREZZO * P. QUANTITA) * QP. QUANTITA)
FROM ORDINE O, PRODOTTO P, ORD_PPQD OP
WHERE O.ID = OP.ORDINE AND O.ID = P.ID AND OP.PRODOTTO = P.ID
```

RESULT = Q

-CALCOLA_VL (in: integer): L

Mandizza in Q LA QUERY:

```
SELECT SUM ((P2. INGRED * P. QUANTITA) * QP. QUANTITA)
FROM ORDINE O, PRODOTTO P, ORD_PPQD OP, PIZZETTA P2, DOLCE DLT P3
WHERE O.ID = OP.ORDINE AND O.ID = P.ID AND OP.PRODOTTO = P.ID AND P.PIZZETTA = P2.ID
AND P.DOLCE = DLT.ID
```

RESULT = Q

CLASSE VIAGGIO

-CALCOLA_INGOMPA(V1: integer): L

Mandizza in Q LA QUERY

```
SELECT SUM(CALCOLA_VOL(O.ID))
FROM VIAGGIO V, ORD_VIA6 OV, ORDINAT O
WHERE V.ID = V1 AND OV.VIAGGIO = V.ID AND OV.ORDINAT = O.ID
```

RESULT = Q

Risposta alla Domanda 8 (segue)

CLASS OF FATTO RAC

-CALCDA_CARS (F1: INTEGER, IN: DATETIME, FIN: DATETIME): DEAL

MEMORIZZA IN Q IL RISULTATO DELLA QUERY

SELECT COUNT(CC.ID)

FROM CONSEGNA C, ORDINE O, VIAGGIO V, ORD_VIAGGIO OV, FATTO_RACCO F
WHERE F.ID=FR AND V.FATTO_RACCO=F.ID AND OV.VIAGGIO=V.ID
AND OV.ORDING=O.ID AND C.ORDINE=O.ID AND
IN <= C.LISTCARS <= FIN AND C.IDCARS <= O.CARS

Memorizza IN S IL RISULTATO DELLA QUERY

SELECT COUNT(CC.ID)

FROM CONSEGNA C, ORDINE O, VIAGGIO V, ORD_VIAGGIO OV, FATTO_RACCO F
WHERE F.ID=FR AND V.FATTO_RACCO=F.ID AND OV.VIAGGIO=V.ID
AND OV.ORDING=O.ID AND C.ORDINE=O.ID AND
IN < C.LISTCARS <= FIN

RESULT = Q/S

USE CASE

-STATISTICHE_FATTO_RACCO (IN: DATA, FIN: DATA): [FATTO_RACCO, REALE] [0..*]

MEMORIZZA IN Q IL RISULTATO DELLA QUERY

SELECT F.ID, CALCDACARS(F.ID, IN, FIN)

FROM FATTO_RACCO F

GROUP BY F.ID

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.]

- NUOVA - CONSEGNA CO: Integ_{v:Index} (v:Index):

MEMORIA IN IS VALIDA LA QUGLI

$\exists \text{ SELECT } F.\text{MAX_EL} \geq (\text{CALCOLA_IN_ORDINE}(V.ID) + \text{CALCOLA_VOLUME}(Q.ID))$
 FROM FATTORIAC F, VIA GLO 1, ORD_VIA6 OV, ACCESSIONE A
 WHERE F.ID = V.FATTORIAC AND V.ID = V AND OV.VIA_GLO = V.ID AND
 Q.V.ORDINE = Q.ID AND Q.ID = 0 AND A.ORDINE = Q.ID

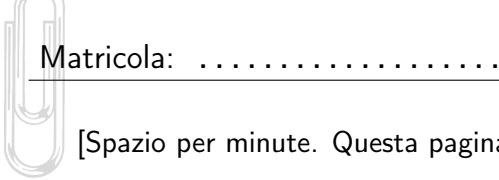
IF IS VARIO:

INSERIRE Ogni VIA GLO (ORDINE, VIA GLO, IS1 VIA GLO)

VALUES (PAR_1, PAR_2, NOW())

Dopo AVER CAMBIATO PAR_1 E PAR_2 DIVERSAMENTE CON O E V

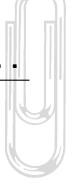
[Spazio per minute. Questa pagina non sarà valutata a meno che non sia puntata da pagine precedenti.]



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.]