

Esame Es.20230113 – Prova scritta del 13 gennaio 2023

Si vuole progettare *EasyToll*, un'applicazione per la gestione della rete autostradale nazionale. *EasyToll* deve permettere di rappresentare la struttura della rete autostradale italiana. Ogni **autostrada** nella rete è identificata da un **codice alfanumerico univoco**, da un **nome** (opzionale) e dalle due **località ai suoi estremi** (ad es., l'autostrada "A1" di nome "Autostrada del Sole" collega Milano a Napoli).

I **caselli** permettono **l'ingresso e l'uscita** dei veicoli in diversi punti della rete. Per ogni **casello** il sistema deve poter rappresentare **l'autostrada lungo la quale è collocato**, un **codice ed il nome** (univoci all'interno della rete, di solito il nome è quello della località in cui è ubicato), la sua **posizione in Km** da una delle due località estremi dell'autostrada (sempre la stessa per la stessa autostrada), il **volume di traffico massimo sostenibile** in entrata ed in uscita (in termini di veicoli/ora), le **modalità di pagamento ammesse** (in particolare: contanti, carte di credito, bancomat, ViaCard, Telepass), e l'eventuale **presenza di un servizio di assistenza ai clienti**.

Le **tariffe** applicate ai veicoli sono espresse in **Euro/Km**. Queste **possono essere diverse per autostrade diverse e per classi di veicoli diverse**.

Ogni qualvolta un **veicolo entra** nella rete autostradale attraverso un **casello**, il conducente è tenuto a **ritirare un tagliando** contrassegnato da un **codice identificativo**. Il sistema **associa il tagliando** al **casello di ingresso**, alla **classe del veicolo**, ed alla **data ed ora del passaggio**. (Tale tagliando è virtuale in caso il veicolo sia equipaggiato di sistema *Telepass*, v. seguito.) All'uscita dalla rete, il conducente è tenuto ad **inserire** il tagliando in suo possesso nell'apposita fessura dell'interfaccia hardware del casello oppure al personale di servizio che provvede a tale inserimento. Il sistema *EasyToll* deve poter **calcolare l'importo dovuto** applicando la **tariffa chilometrica** (relativa alla classe del veicolo) **per il numero di chilometri percorsi** e mostrarla all'automobilista per avviare la procedura di pagamento (per semplicità, in questa sede si ignori tale procedura).

In caso di utilizzo del sistema *Telepass*, le procedure di ingresso ed uscita da un casello avvengono senza l'arresto del veicolo. Ogni dispositivo *Telepass* è univocamente associato ad un **cliente** abbonato al servizio. All'ingresso, il casello identifica (mediante un segnale radio) il cliente e genera un **tagliando virtuale** (che viene memorizzato nel sistema) con le stesse informazioni dell'equivalente cartaceo. All'uscita, il casello identifica nuovamente il cliente e calcola l'importo dovuto per il tragitto. Di ogni **cliente** abbonato al servizio *Telepass* interessa **nome, cognome, codice fiscale e coordinate IBAN** per l'addebito del conto mensile.)

Il sistema di fatturazione (un sistema esterno a *EasyToll*), dato un **cliente** *Telepass*, un mese ed un anno, deve poter ottenere il **conto** di quel cliente nel dato mese del dato anno, contenente, per ogni singolo addebito del mese, il nome dei caselli e la data e l'ora di entrata ed uscita, oltre che l'importo addebitato.)

Per disincentivare gli eccessi di velocità, lungo la rete sono posizionati terminali del sistema **Tutor** equipaggiati da telecamere e di apposito software per il riconoscimento delle targhe dei veicoli. Il sistema **Tutor** permette di rilevare l'eccesso della **velocità media calcolata tenendo conto del tempo impiegato da un veicolo per viaggiare lungo un breve tratto delimitato da due terminali Tutor** (di solito posizionati tra i 10 e 25 Km tra loro). Per ogni coppia di terminali **Tutor** il sistema deve poter memorizzare **la sua ubicazione (autostrada, senso di marcia, posizione del terminale di inizio e di quello di fine, in modo analogo alle posizioni dei caselli)**.

Ogni volta che un veicolo attraversa il terminale primo di una coppia di terminali Tutor (quello di ingresso), EasyToll memorizza la targa e la classe del veicolo, la coppia di terminali che lo sta monitorando, la data e l'ora.) Quando un veicolo attraversa il terminale Tutor di uscita, il sistema calcola la sua velocità media nel breve tratto controllato. Se questa è minore o uguale alla velocità massima consentita (un parametro del sistema che può dipendere anche dalla classe del veicolo oggetto del controllo), il sistema elimina immediatamente i dati del passaggio, altrimenti li mantiene di modo che possano essere acceduti dal personale della Polizia Stradale.

targa
veicolo

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

Autstrada:

- codice: CadAd
- nome: Stringa [0..1]
- associato alle località:
agli istreni

Chitite:

- nome: Stringa
- Gagnone: Stringa
- CF: Codice fiscale
- IBAN: IBAN

Casello

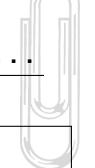
- associato all'Autstrada
- codice: CadCas
- nome: Stringe
- posizione: Resle ≥ 0
- associato ai pagamenti possibili
- gestenza: Baslema

Tutte:

- associato all'Autstrada
- tipo: Baslema
- PostInizio: Resle ≥ 0
- PostFine: Resle ≥ 0

Tagliando:

- codice: CadTag
- MTPS: DatiPS
- associato al Casello
di ingresso



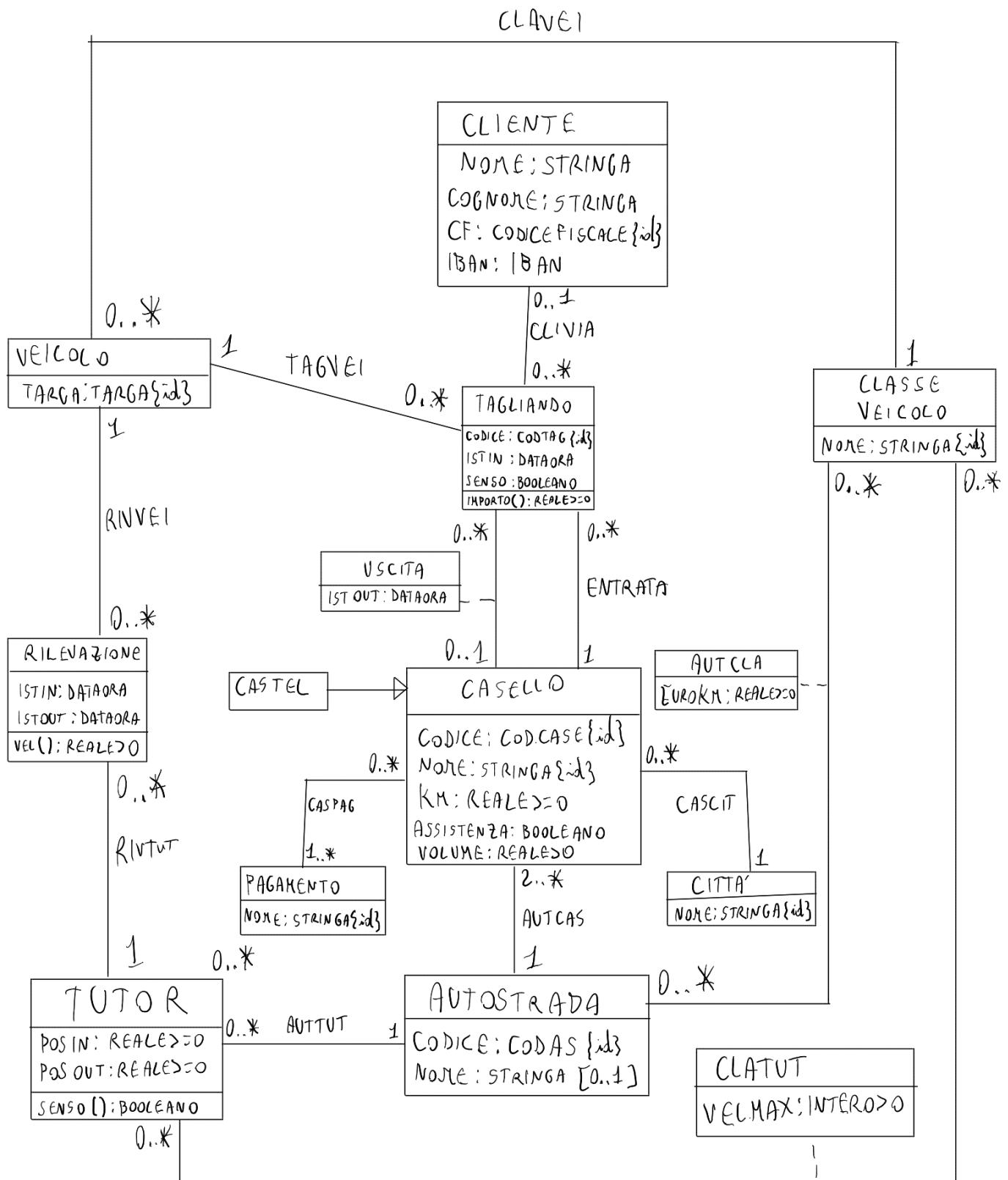
Risposta alla Domanda 1 (segue)

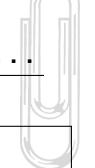
Domanda 2 (45 minuti; 75 minuti al massimo) Proseguire la fase di Analisi Concettuale dei requisiti, producendo un diagramma UML concettuale delle classi per l'applicazione, le specifiche di classi, associazioni, tipi di dato e vincoli esterni.

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'attrib. 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 $\text{op}(c, \text{arg}_1, \dots, \text{arg}_n, v)$: uno dei valori di ritorno di op , quando invocata sull'istanza c e con argomenti $\text{arg}_1, \dots, \text{arg}_n$ è v .
 - Il simbolo di $=/2$ (la cui interpretazione è la relazione che lega ogni elemento del dominio di interpretazione solo con se stesso) e opportuni simboli di predicato e di funzione, soggetti a semantica di modo reale, per relazioni e funzioni standard tra elementi dei tipi di dato, tra cui $\text{adesso}/0$, interpretato come il valore del dominio DataOra che rappresenta l'istante corrente.

Risposta

<p>1 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...TAGLIANDO.....</p> <p>Operazioni, vincoli:</p> <p>V, USCITA DOPO ENTRATA</p> <p>$\forall t, co, in, out \ USCITA(t, co) \wedge ISENIN(in)$</p> <p>$\wedge ISENOUT(t, co, out) \rightarrow in < out$</p> <p>V, SENSO GIUSTO</p> <p>$\forall t, s, ci, co, kin, kont \ ENTRATA(t, ci) \wedge USCITA(t, co) \wedge$</p> <p>$SENSO(t, s) \wedge KM(ci, kin) \wedge KM(co, kont) \rightarrow$</p> <p>$[SENSO = true \wedge kin < kont] \vee [SENSO = false \wedge kin > kont]$</p> <p>V, STESSA AUTOSTRADA</p> <p>$\forall t, ci, co \ USCITA(t, co) \wedge ENTRATA(t, ci) \rightarrow$</p> <p>$\exists a \ AUTCAS(a, ci) \wedge AUTCLA(a, co) \wedge EUROKM(a, ci, t)$</p>	<p>2 Tipo: Classe Associazione (cerchiare)</p> <p>Nome: ...TAGLIANDO.....</p> <p>Operazioni, vincoli:</p> <p>$\text{IMPORTO}(): \text{REAL} \geq 0$</p> <p>- pre: $\exists c \ USCITA(t, ci, c)$</p> <p>- post: SIANO d, kin, kont, t $\exists v \ TAGUEI(t, v) \wedge CLAVEI(v, d)$ $\exists w, co \ USCITA(w, t) \wedge ENTRATA(w, ci) \wedge KM(ci, kin) \wedge KM(co, kont) \wedge$ $\exists a \ AUTCAS(a, ci) \wedge AUTCLA(a, co) \wedge EUROKM(a, ci, t)$</p> <p>RESULT = $kin - kont \cdot t$</p>
--	---

3	Tipo: Classe Associazione (cerchiare)
Nome:	CLIENTE
Operazioni, vincoli:	<p>$\forall t, \text{TELEPASSUSABILE}$</p> <p>$\forall c, t, \omega \text{ CLITAG}(c, t) \wedge$</p> <p>$[\text{USCITA}(t, \omega) \vee \text{ENTRATA}(t, \omega)] \rightarrow \text{CASTEL}(\omega)$</p>

6	Tipo: Classe Associazione (cerchiare)
Nome:	TUTOR
Operazioni, vincoli:	<p>$\text{SENSO}(): \text{BOOLEANO}$</p> <ul style="list-style-type: none"> - pre: - post: SIANO i, f TALI DA SODDISFARE: $\text{POSIN}(\text{this}, i) \wedge \text{POSOUT}(\text{this}, f)$ $(i < f \rightarrow \text{RESULT} = \text{TRUE}) \wedge (i > f \rightarrow \text{RESULT} = \text{FALSE})$ <p>$\forall . \text{DISGIUNTI}$</p> <p>$\forall a, t, t', p_i, p_i', p_o, p_o', s, s' \text{ AUTTUT}(a, t) \wedge \text{AUTTUT}(a, t') \wedge$</p> <p>$\text{POSIN}(t, p_i) \wedge \text{POSOUT}(t, p_o) \wedge \text{POSIN}(t', p_i') \wedge \text{POSIN}(t', p_o') \wedge$</p> <p>$\text{SENSO}(t, s) \wedge \text{SENSO}(t', s') \wedge s = s' \rightarrow (p_i < p_i' \wedge p_o < p_o') \vee (p_i' < p_o \wedge p_o' < p_i')$</p>

4	Tipo: Classe Associazione (cerchiare)
Nome:	RILEVAZIONE
Operazioni, vincoli:	<p>$\forall . \text{VEICOLO ESISTE}$</p> <p>$\forall m, i, \omega, v, t, p_i, p_o, s \text{ RILEVAZIONE}(m) \wedge$</p> <p>$\text{ISTIN}(i) \wedge \text{ISTOUT}(\omega) \wedge \text{RILVEI}(m, v) \wedge$</p> <p>$\text{RILTUT}(n, t) \wedge \text{POSIN}(t, p_i) \wedge \text{POSOUT}(p_o) \wedge$</p> <p>$\text{SENSO}(t, s) \rightarrow \exists \text{tag}, \text{it}, \text{ci}, \text{ki}, \text{st}$</p> <p>$\text{TAGVEI}(\text{tag}, v) \wedge \text{ISTIN}(\text{tag}, \text{it}) \wedge \text{KM}(\text{ci}, \text{ki}) \wedge$</p> <p>$\text{SENSO}(\text{tag}, \text{st}) \wedge \text{st} < \text{ii} \wedge \text{it} = \text{st} \wedge$</p> <p>$[\text{st} = \text{FALSE} \wedge \text{ki} > \text{pi} \wedge (\exists \text{co}, \text{at}, \text{ko} \text{ USCITA}(\text{co}, \text{tag}) \wedge \text{ISTOUT}(\text{tag}, \text{co}, \text{at}) \wedge$</p> <p>$\text{km}(\text{co}, \text{ko}) \rightarrow \text{at} > \text{io} \wedge \text{ko} < \text{po}) \vee$</p> <p>$[\text{st} = \text{TRUE} \wedge \text{ki} < \text{pi} \wedge (\exists \text{co}, \text{at}, \text{ko} \text{ USCITA}(\text{co}, \text{tag}) \wedge \text{ISTOUT}(\text{tag}, \text{co}, \text{at}) \wedge$</p> <p>$\text{km}(\text{co}, \text{ko}) \rightarrow \text{at} > \text{io} \wedge \text{ko} > \text{po}]$</p>

7	Tipo: Classe Associazione (cerchiare)
Nome:	RILEVAZIONE
Operazioni, vincoli:	<p>$\text{VELOCITA}(); \text{REALE} > 0$</p> <ul style="list-style-type: none"> - pre: - post: SIANO i, ω, p_i, p_o, s TALI DA SODDISFARE: $\text{ISTIN}(\text{this}, i) \wedge \text{ISTOUT}(\text{this}, \omega) \wedge$ $\exists t \text{ RILTUT}(\text{this}, t) \wedge \text{POSIN}(t, p_i) \wedge \text{POSOUT}(t, p_o) \wedge$ $\text{ORE}(\omega - i, s)$ <p>$\text{RESULT} = \frac{ p_i - p_o }{s}$</p> <p>$\forall . \text{FINE DOPOLIZIO}$</p> <p>$\forall m, i, f \text{ RILEVAZIONE}(m) \wedge \text{ISTOUT}(m, f) \wedge \text{ISTIN}(m, i) \rightarrow f > i$</p>

5	Tipo: Classe Associazione (cerchiare)
Nome:	VEICOLO
Operazioni, vincoli:	<p>$\forall . \text{TAGLIANDI DISGIUNTI}$</p> <p>$\forall v \text{ VEICOLO}(v) \rightarrow \exists t, t', i, i', d \text{ DATAORA}(d)$</p> <p>$\text{TAGVEI}(t) \wedge \text{TAGVEI}(t') \wedge \text{ISTIN}(i, t) \wedge \text{ISTIN}(i', t') \wedge$</p> <p>$d > i \wedge d > i' \wedge (\forall c, f \text{ ISTFINE}(c, t, f) \rightarrow f > d) \wedge t \neq t' \wedge$</p> <p>$(\forall c, f \text{ ISTFINE}(c, t', f) \rightarrow f > d)$</p> <p>$\forall . \text{RILEVAZIONI DISGIUNTE}$</p> <p>$\forall v \text{ VEICOLO}(v) \rightarrow \exists n, m, i, i', f, f', d \text{ DATAORA}(d)$</p> <p>$\text{RILVEI}(n, v) \wedge \text{RILVEI}(m, v) \wedge \text{ISTIN}(i, n) \wedge \text{ISTIN}(i', m) \wedge$</p> <p>$\text{ISTFINE}(f, n) \wedge \text{ISTFINE}(f', m) \wedge d > i \wedge d > i' \wedge d < f \wedge d < f' \wedge$</p> <p>$m \neq n$</p>

8	Tipo: Classe Associazione (cerchiare)
Nome:	
Operazioni, vincoli:	

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

CODICEFISCALE : (STRINGA SECONDO STANDARD)

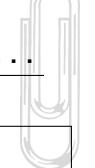
IBAN : (STRINGA SECONDO STANDARD)

TARGA : ([A-Z]{2}{2}[0-9]{3}[A-Z]{2})

CODTAG : (STRINGA SECONDO STANDARD)

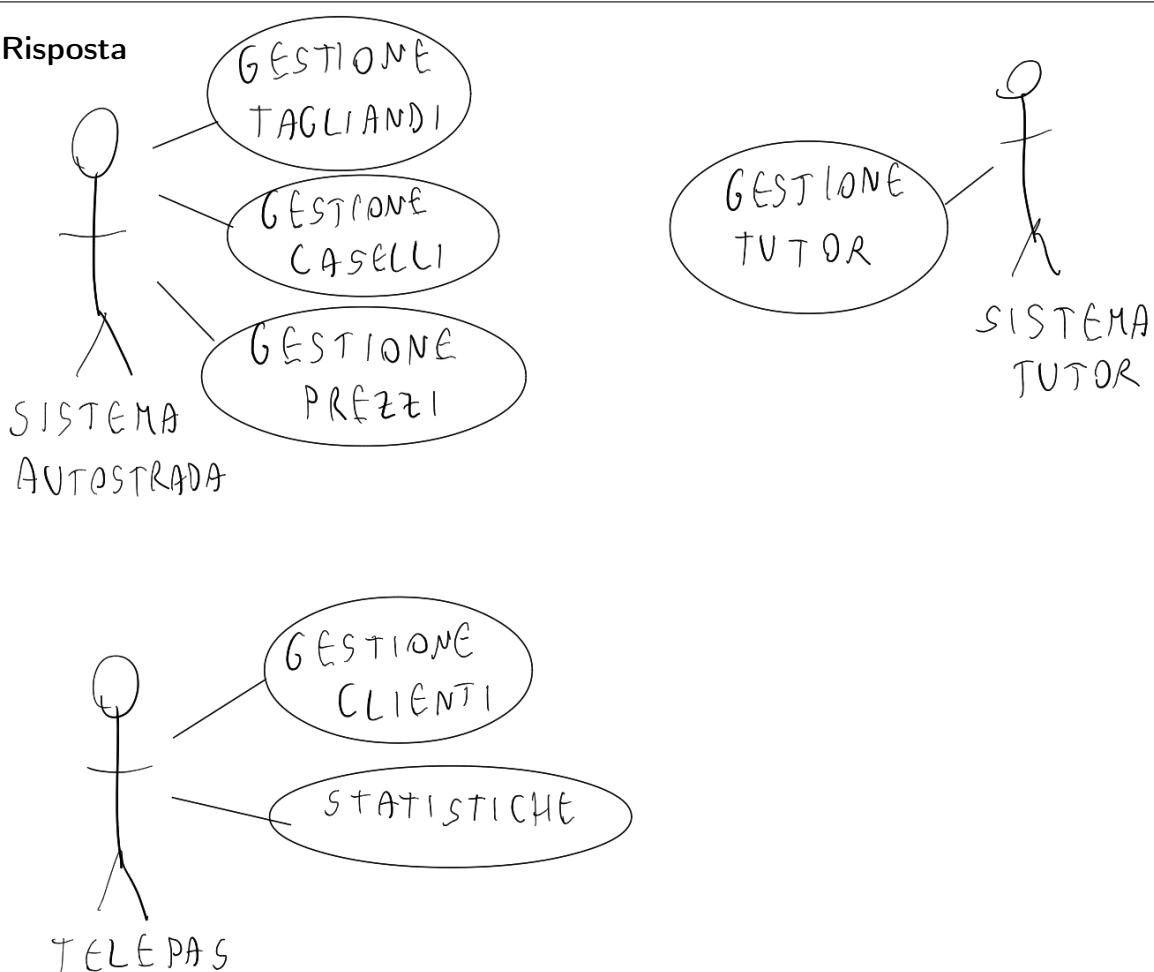
CODAS : (STRINGA SECONDO STANDARD)

CODCASE : (STRINGA SECONDO STANDARD)



Risposta alla Domanda 2 (segue)

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

Risposta

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

Risposta

GESTIONE TAGLIANDI:

NEW TAGLIANDO (v: VEICOLO, c: CASELLO, s: DOLCANO, cod: CODTAG) : TAGLIANDO

USCITA TAGLIANDO (t: TAGLIANDO, c: CASELLO)

GESTIONE CASELLI:

NEW CASELLO (a: AUTOSTRADA, km: REALE>=0, cod: CODCASE, n: STRINGA, a: BOOLEAN, val: REALE>=0) : CASELLO

GESTIONE PREZZI:

IMPOSTAPREZZO (c: CLASSEVEICOLO, a: AUTOSTRADA, p: REALE>=0)

GESTIONE CLIENTI:

NEW CLIENTE (n: STRINGA, CF: CODICEFISCALE, i: IBAN) : CLIENTE

STATISTICHE:

IMPORTOMESE (a: INTERO>0, m: 1..12, c: CLIENTE) : (CASELLO, CASELLO, DATAORA, DATAORA, REALE>=0)

GESTIONE TUTORI:

NEW TUTOR (a: AUTOSTRADA, km: REALE>=0, kont: REALE>=0) : TUTOR

CONTROLLARILEVAZIONE (m: RILEVAZIONE)

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

$\text{IMPORTO MESE}(\alpha: \text{INTERO} \geq 0, m: 1..12, C: \text{CLIENTE}): (\text{CASELLO}, \text{CASELLO}, \text{DATAORA}, \text{DATAORA}, \text{REALE} = 0)[0..*]$

- pre:
- post:

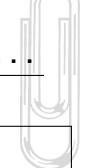
$$\text{RESULT} = \left\{ (c_m, cont, \hat{i}_m, i_m, p) \mid \begin{array}{l} \exists t \text{ CLITAG}(t, c) \wedge \text{ENTRATA}(t, c_m) \wedge \text{USCITA}(t, cont) \\ \wedge \text{ISTOUT}(t, cont, i_m) \wedge \text{ANNO}(\hat{i}_m, a) \wedge \text{MESE}(\hat{i}_m, m) \wedge \text{ISTIN}(t, \hat{i}_m) \\ \wedge \text{IMPORTO}(t, p) \end{array} \right\}$$

CONTROLLA TUTOR ($m: \text{RILEVAZIONE}$)

- pre:
- post: SI AVM, V TALE DA SODDISFARE

$$\exists t, v_m, d \text{ RICTUT}(m, t) \wedge \text{RILVEI}(m, v_m) \wedge \text{CLAVEI}(d, v_m) \wedge \text{VELOCITAMAX}(t, d, v_m) \\ \wedge \text{VELOCITA}(m, v_m)$$

$$v <= v_{max} \rightarrow \text{REMOVE}(m)$$



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.....

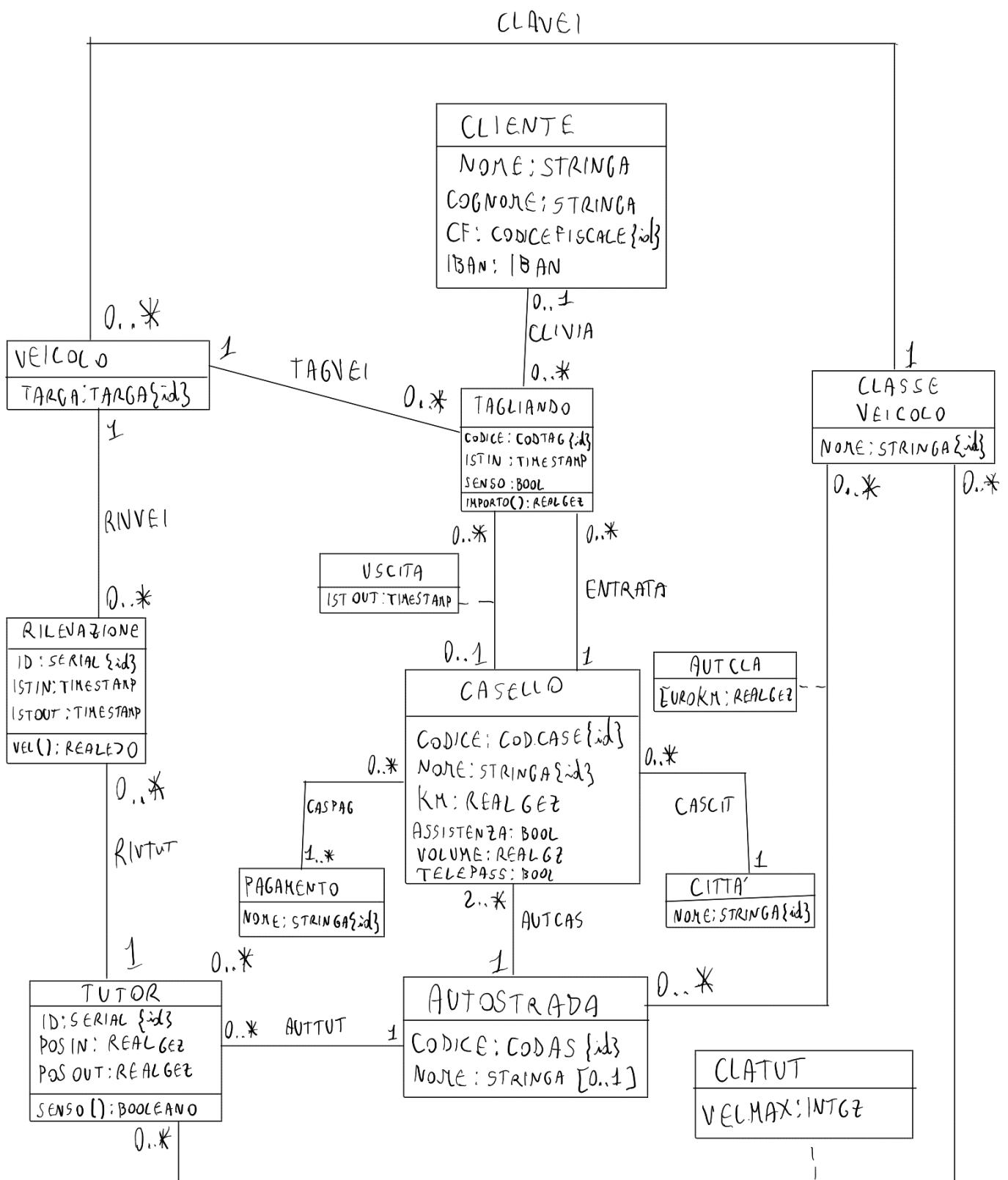
Corrispondenza tra tipi di dato concettuali e domini supportati dal DBMS

```

CREATE DOMAIN STRINGA AS VARCHAR NOT NULL;
CREATE DOMAIN IBAN AS VARCHAR;
CREATE DOMAIN CODICEFISCALE AS VARCHAR;
CREATE DOMAIN CODTAG AS VARCHAR;
CREATE DOMAIN CODCASE AS VARCHAR;
CREATE DOMAIN COD AS VARCHAR;
CREATE DOMAIN TARGA AS VARCHAR CHECK(VALUE ILIKE '[A-Z]{2}[0-9]{3}[A-Z]{2}');
CREATE DOMAIN INTGZ AS INTEGER CHECK(VALUE>0);
CREATE DOMAIN INTGEZ AS INTEGER CHECK(VALUE>=0);
CREATE DOMAIN REALGZ AS REAL CHECK(REAL>0);
CREATE DOMAIN REALGEZ AS REAL CHECK(REAL>=0);
CREATE DOMAIN NMESCE AS INTEGER CHECK(VALUE>=1 AND VALUE<=12);

```

Diagramma UML delle classi ristrutturato



Breve descrizione delle scelte effettuate durante la ristrutturazione

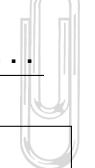
FUSIONE SU CASELLO

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)

$\forall c, t, cos \text{ CLIENTE. TELEPASS USABILE}$

$$\forall c, t, cos \quad CLITAG(c, t) \wedge [USCITA(t, cos) \vee ENTRATA(t, cos)] \rightarrow TELEPASS(cos, \text{true})$$



Risposta alla Domanda 6 (segue)

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

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

1 Relazione	<u>CLIENTE</u> ... (nome)	Derivante da: classe associazione (cerchiare)
--------------------	---------------------------	---

Attributi	<u>NOME</u>	<u>COGNOME</u>	<u>CF</u>	<u>IBAN</u>				
Domini	<u>STRINGA</u>	<u>STRINGA</u>	<u>CODICE FISCALE</u>	<u>IBAN</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:

2 Relazione	<u>TAGLIANDO</u> ... (nome)	Derivante da: classe associazione (cerchiare)
--------------------	-----------------------------	---

Attributi	<u>CODICE</u>	<u>ISTIN</u>	<u>ISENSO</u>	<u>CLIENTE*</u>	<u>VEICOLO</u>	<u>ENTRATA</u>	<u>USCITA*</u>	<u>ISTOUT*</u>
Domini	<u>CODCASE</u>	<u>TIMESTAMP</u>	<u>BOOL</u>	<u>CODICEFISCALE</u>	<u>TARGA</u>	<u>CODCASE</u>	<u>CODCASE</u>	<u>TIMESTAMP</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):

$\text{FK}(\text{CLIENTE}) \text{ REF } \text{CLIENTE}(\text{CF})$ $\text{FK}(\text{ENTRATA}) \text{ REF } \text{CASELLO}(\text{CODICE})$ $\text{CHECK}((\text{USCITA IS NULL}) = (\text{ISTOUT IS NULL}))$
 $\text{FK}(\text{VEICOLO}) \text{ REF } \text{VEICOLO}(\text{TARGA})$ $\text{FK}(\text{USCITA}) \text{ REF } \text{CASELLO}(\text{CODICE})$ $\text{CHECK}(\text{ISTIN} \leq \text{ISTOUT})$

La relazione accorda le relazioni che implementano le seguenti associazioni: CLITAG..ENTRATA, USCITA..TAGVEI

3 Relazione	<u>CASELLO</u> ... (nome)	Derivante da: classe associazione (cerchiare)
--------------------	---------------------------	---

Attributi	<u>CODICE</u>	<u>NOME</u>	<u>KM</u>	<u>ASSISTENZA</u>	<u>TELEPASS</u>	<u>VOLUME</u>	<u>CITTÀ</u>	<u>AS</u>
Domini	<u>CODCASE</u>	<u>STRINGA</u>	<u>REALIZZ</u>	<u>BOOL</u>	<u>BOOL</u>	<u>REALIZZ</u>	<u>STRINGA</u>	<u>CODAS</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 (none) $\text{FK}(\text{AS}) \text{ REF } \text{AUTOSTRADA}(\text{CODICE})$ $\text{ID} \subseteq \text{CASPA}G(\text{CASELLO})$
 $\text{FK}(\text{CITTÀ}) \text{ REF } \text{CITTÀ}(\text{NOME})$

La relazione accorda le relazioni che implementano le seguenti associazioni:

4 Relazione	<u>CITTÀ</u> (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:

5 Relazione	<u>PAGAMENTO</u> ... (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:

6 Relazione ...CAS.PAG.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>CASELLO</u> <u>PAGAMENTO</u>	
Domini CODCASE 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 (CASELLO) REF CASELLO (CODICE)

FK (PAGAMENTO) REF PAGAMENTO (NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

7 Relazione ...T.TUT.QR.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>ID</u> POSIN POSOUT AS	
Domini SERIAL REALGEZ REALGEZ CODAS	

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 (AS) REF AUTOSTRADA (CODICE)

La relazione accorda le relazioni che implementano le seguenti associazioni:

8 Relazione ...CLASSE.VEIColo.... (nome)	Derivante da: classe associazione (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 ennupla, di dominio):

La relazione accorda le relazioni che implementano le seguenti associazioni:

9 Relazione ...AUT.CLA.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>AS</u> <u>CLASSE</u> TARIFFE	
Domini CODAS STRINGA REALGEZ	

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 (AS) REF AUTOSTRA DA (CODICE)

FK (CLASSE) REF CLASSE.VEIColo (NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

10 Relazione ...CLAT.VT.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi <u>TUTOR</u> <u>CLASSE</u> VELMAX	
Domini INTEGER STRINGA INT6Z	

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 (TUTOR) REF TUTOR (ID)

FK (CLASSE) REF CLASSE.VEIColo (NOME)

La relazione accorda le relazioni che implementano le seguenti associazioni:

11 Relazione ...AUTOSTRADA... (nome)	Derivante da: classe associazione (cerchiare)
Attributi CODICE NOME *	
Domini CODAS 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):

La relazione accorda le relazioni che implementano le seguenti associazioni:

12 Relazione ...VEICOLO.... (nome)	Derivante da: classe associazione (cerchiare)
Attributi TARGA CLASSE	
Domini TARGA 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(CLASSE) REF CLASSEVEICOLO(Nome)

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

13 Relazione ...RILEVAZIONE (nome)	Derivante da: classe associazione (cerchiare)
Attributi ID ISTIN ISTOUT TUTOR VEICOLO	
Domini SERIAL TIMESTAMP TIMESTAMP INTEGER TARGA	

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(TUTOR) REF TUTOR(ID)

CHECK(ISTIN < ISTOUT)

Fk(VEICOLO) REF VEICOLO(TARGA)

La relazione accorda le relazioni che implementano le seguenti associazioni: ...RILTUT, RILVEI.....

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

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

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

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

La relazione accorda le relazioni che implementano le seguenti associazioni:

Ulteriori vincoli esterni

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

- INSERT OR UPDATE TAGLIANDO

- ERROR= EXIST (SELECT * FROM TAGLIANDO t, CASELLO cont, CASELLO cin
WHERE CODICE=NEW.CODICE AND t.ENTRATA=cin.CODICE AND t.USCITA=cont.CODICE
AND (t.SENSO=TRUE AND cin.KM > cont.KM) OR (SENTO=FALSE AND cin.KM < cont.KM))

```
IF ERROR: ERROR
ELSE: COMMIT
```

T. TAGLIANDO. STESSA AUTOSTRADA

- INSERT OR UPDATE TAGLIANDO

- ERROR= EXIST (SELECT * FROM TAGLIANDO t, CASELLO cont, CASELLO cin
WHERE CODICE=NEW.CODICE AND t.ENTRATA=cin.CODICE AND t.USCITA=cont.CODICE
AND cin.AS <> cont.AS)

```
IF ERROR: ERROR
ELSE: COMMIT
```

T. CLIENTE. TELEPASS USABILE

- INSERT OR UPDATE TAGLIANDO

- ERROR= EXIST (SELECT * FROM TAGLIANDO t, CASELLO cont, CASELLO cin
WHERE CODICE=NEW.CODICE AND t.ENTRATA=cin.CODICE AND t.USCITA=cont.CODICE AND t.CLIENTE IS NOT NULL
AND (cin.TELEPASS=FALSE OR cont.TELEPASS=FALSE))

```
IF ERROR: ERROR
ELSE: COMMIT
```

T. RILEVAZIONE. VEICOLO ESISTE

- INSERT OR UPDATE RILEVAZIONE

- VALID= EXIST (WITH T AS (SELECT * FROM TUTOR WHERE ID=NEW.TUTOR)
SELECT * FROM TAGLIANDO t, CASELLO cin LEFT OUTER JOIN CASELLO cont
ON t.USCITA=cont.CODICE
WHERE t.ENTRATA=cin.CODICE AND t.SENSO=SENTO(T.ID) AND
VEICOLO=NEW.VEICOLO AND LISTIN<NEW.LISTIN AND (t.ISTOUT IS NULL OR t.ISTOUT>NEW.ISTOUT)
AND (t.SENSO=TRUE AND cin.KM < t.POSIN AND (cont.KM > t.POSOUT OR cont.KM IS NULL)) OR
(t.SENSO=FALSE AND cin.KM > t.POSIN AND (cont.KM < t.POSOUT OR cont.KM IS NULL)))

```
IF VALID: COMMIT
ELSE: ERROR
```

Risposta alla Domanda 7 (segue)

TUTOR.DISGIVNTJ

- INSERT OR UPDATE TUTOR

- ERROR := EXIST (SELECT * FROM TUTOR
 WHERE ID <> NEW.ID AND AS = NEW.AS AND SENSO(NEW.ID) = SENSO(ID)
 AND (NEW.POSIN BETWEEN POSIN AND POSOUT OR NEW.POSOUT BETWEEN POSIN AND POSOUT))

IF ERROR; ERROR

ELSE; COMMIT

VEICOLO.RILEVAZIONI DISGIUNTE

- INSERT OR UPDATE RILEVAZIONI

- ERROR := EXIST (SELECT * FROM RILEVAZIONE
 WHERE ID <> NEW.ID AND (NEW.ISTIN, NEW.ISTOUT) OVERLAPS (ISTIN, ISTOUT)
 AND VEICOLO = NEW.VEICOLO)

IF ERROR; ERROR

ELSE; COMMIT

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

IMPORTO MESE(*a*: INTGE, *m*: NNESE, *C*: CODICEFISCALE); (*CODCASE*, *CODCASE*, *TIMESTAMP*, *TIMESTAMP*, *REALGEZ*) [0..*]

RESELECT ENTRATA, USCITA, ISTIN, ISTOUT, IMPORTO(CODTAC) FROM TAGLIANDO
WHERE EXTRACT(MONTH FROM ISTOUT)=*m* AND EXTRACT(YEAR FROM ISTOUT)=*a* AND CLIENTE IS NOT NULL

RETURN R

CONTROLLA TUTOR(*M*: INTEGER):

R:=WITH VMAX AS (SELECT VELMAX FROM RILEVAZIONE m1, TUTOR t, VEICOLO v, CLATUT ct
WHERE m1.TUTOR=t.ID AND m1.TARGA=v.TARGA AND ct.TUTOR=t.ID AND ct.CLASSE=v.CLASSE)
SELECT VMAX.VELMAX<VELOCITA(*M*) t FROM VMAX

IF R.t=FALSE;

DELETE FROM RILEVAZIONE WHERE ID=*M*

Risposta alla Domanda 8 (segue)

```
CREATE FUNCTION IMPORTO(t: CODTAG): REALGEZ
ERROR=EXIST(SELECT * FROM TAGLIANDO WHERE CODICE=t AND USCITA IS NULL)
```

$R = \text{SELECT ac.TARIFFE * ABS(cin.km - cout.km)}$
 FROM TAGLIANDO tag, VEICOLO v, CASELLO cin, CASELLO cout, AUT CLA ac
 WHERE tag.ID=t AND tag.USCITA=cout.CODICE AND tag.ENTRATA=cin.CODICE AND
 $tag.VEICOLO=v.TARGA \text{ AND } v.CLASSE=ac.CLASSE \text{ AND } ac.AS=cin.AS$

RESULT=R

```
CREATE FUNCTION SENSO(t: INTEGER): BOOL
```

$R = \text{WITH T AS (SELECT * FROM TUTOR WHERE ID=t)}$
 $\text{SELECT T.POSIN < T.POSOUT FROM T}$

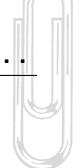
RESULT=R

```
CREATE FUNCTION VELOCITA(m: INTEGER): REALGEZ
```

$R = \text{SELECT ABS(t.POSIN - t.POSOUT) / (EXTRACT(EPOCH FROM (ml.ISTOUT - ml.ISTIN))) / 3600}$
 FROM RILEVAZIONE ml, TUTOR t
 WHERE ml.TUTOR=t.ID

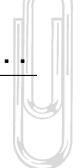
RETURN=R

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.]