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: .....

[Versione 2024-03-29]

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.

[Versione 2024-03-29]

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

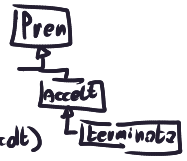
1 Analisi concettuale

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. Utente
   - 1.1 nome
   - 1.2 cognome
   - 1.3 sesso
   - 1.4 citta' di residenza
   - 1.5 ospitalita' da offrire/Casa
     - 1.5.1 distanza dal centro KM
     - 1.5.2 dist dalla stazione
     - 1.5.3 num membri famiglia(adulti/bambini)
     - 1.5.4 posti letto
       - 1.5.4.1 singolo/doppio
       - 1.5.4.2 camera sep/comune
       - 1.5.4.3 divano?
   - 1.6 date di non disponibilita' (no intersezioni)

2. Prenotazione
   - 2.1 richiedente
   - 2.2 alloggio
   - 2.3 posti (≤ posti tot alloggio)
   - 2.4 Accettata/Rifiutata
   - 2.4.1 se Accettata
     - 2.4.1.1 terminata?
     - 2.4.1.2 voto 0..5 ospitato
     - 2.4.1.3 voto 0..5 ospitante(Facdt)



   - 2.5 inizio(Data)
   - 2.6 Fine(Data)

[continua alla pagina seguente]

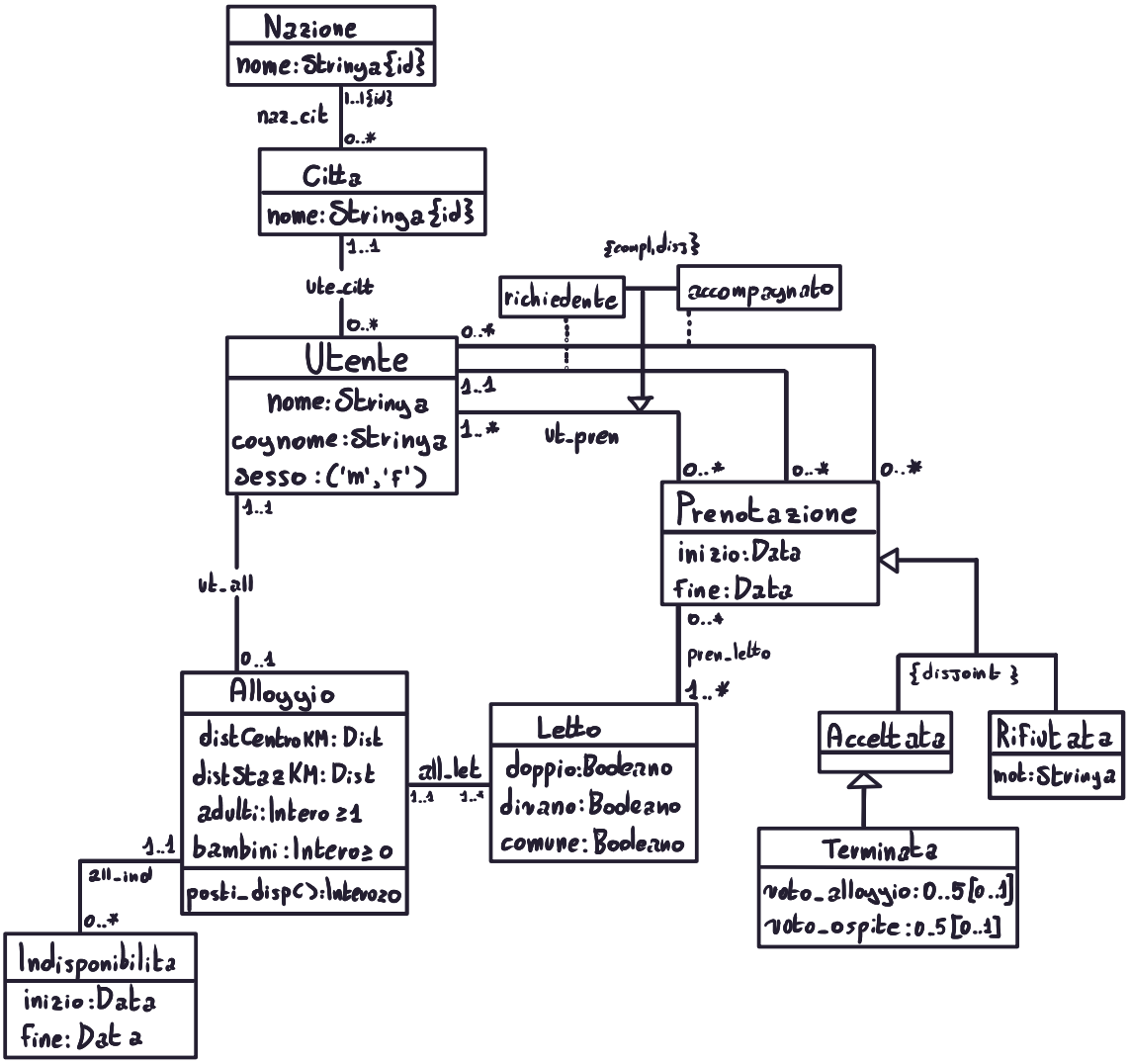
Risposta alla Domanda 1 (segue)

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.



[continua alla pagina seguente]

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 $r(x_1, \ldots, x_n)$: $x_1, \ldots, x_n$ 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 $a(x_1, \ldots, x_n, v)$: uno dei valori dell'attr. a dell'istanza $(x_1, \ldots, x_n)$ 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: Letto

attributo	dominio	moltepl. (*)

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

Vincoli:

[V. no_2_prenotazioni_accettate_intersecano]

$\forall l, p1, p2, i1, i2, f1, f2$
$[Letto(l) \wedge pren\_letto(l, p1) \wedge pren\_letto(l, p2) \wedge Accettata(p1)$
$\wedge Accettata(p2)$
$\wedge inizio(p1, i1) \wedge inizio(p2, i2) \wedge Fine(p1, f1) \wedge$
$Fine(p2, f2)] \rightarrow [f2 < i2 \vee f2 < i1]$

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

Nome: Prenotazione

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.inizio_prima_fine]

$\forall p, i, f\ [Prenotazione(p) \wedge inizio(p, i) \wedge fine(p, f)] \rightarrow i < f$

[V.istante_terminazione]

$\forall p, f, now, d\ [Terminata(p) \wedge Fine(p, f) \wedge Adesso(now)$
$\wedge Data(now, d)] \rightarrow f \leq d$

[Versione 2024-03-29]

3 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Prenotazione

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.utente_no_pren_che_si_intersecano]

$\forall u, p1, p2, i1, i2, F1, F2 \ \neg Rifiutata(p1) \wedge \neg Rifiutata(p2)$
$[Utente(u) \wedge ute\_pren(u, p1) \wedge ute\_pren(u, p2) \wedge$
$inizio(p1, i1) \wedge inizio(p2, i2) \wedge Fine(p1, F1) \wedge Fine(p2, f2)]$
$\rightarrow [f1 < i2] \vee [F2 < i2]$

[V.no_prenotazioni_casa_propria]

$\forall u, a, l [Utente(a) \wedge letto(l) \wedge ut\_all(u, a) \wedge$
$all\_let(a, l)] \rightarrow [\neg \exists p \ Prenotazione(p) \wedge pren\_letto(p, l)$
$\wedge ute\_pren(u, p)]$

5 Tipo: ~~Entità~~ | **Relationship** (cerchiare)

Nome: Indisponibilita

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.no_pren_durante_indisp]

$\forall i, a, l, ii, Fi, p, ip, Fp$
$[Indisponibilita(i) \wedge Alloggio(a) \wedge all\_ind(a, i) \wedge inizio(i, ii)$
$\wedge Fine(i, Fi) \wedge Letto(l) \wedge all\_let(a, l) \wedge$
$pren\_letto(p, l) \wedge inizio(p, ip) \wedge Fine(p, fp)]$
$\rightarrow [Fp < ii \ \vee \ fi < ip]$

[V.inizio_ind_maggiore_Fine]

$\forall in, i, f \ [Indisponibilita'(in) \wedge inizio(in, i) \wedge Fine(in, F)]$
$\rightarrow i < F$

4 Tipo: **Entità** | ~~Relationship~~ (cerchiare)

Nome: Prenotazione

attributo	dominio	moltepl. (*)

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

Vincoli:

[V.pren_stesso_alloggio]

$\forall p, l1, l2, a$
$[Prenotazione(p) \wedge Alloggio(a) \wedge pren\_letto(p, l1) \wedge$
$pren\_letto(p, l2) \wedge all\_let(a, l1)] \rightarrow all\_let(a, l2)$

[V.veduta_tutte]

$\forall p, u$
$[Utente(p) \wedge \neg Terminata(p) \wedge \neg Rifiutata(p) \wedge ute\_pren(u, p)]$
$\rightarrow [\forall \ p1 \ Terminata(p1) \wedge ute\_pren(u, p1) \wedge$
$\exists k \ vede\_alloggio(p1, k)]$

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

Nome: .....

attributo	dominio	moltepl. (*)

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

Vincoli:

[Versione 2024-03-29]

Ulteriori vincoli esterni, specifica di eventuali operazioni ausiliarie invocate da tali vincoli, e specifica dei domini concettuali non di tipo base

Alloggio

posti_disp( ): Intero ≥ 0

•pre-condizioni: nessuna

•pre-condizioni: $PD = \{ l \mid \text{all\_let}(\text{this}, l) \wedge \neg\exists p \ \text{Accettata}(p) \wedge \neg\text{Terminata}(p) \wedge \text{pren\_letto}(p, l) \wedge \text{doppio}(l, \text{TRUE}) \}$

$P = \{ l \mid \text{all\_let}(\text{this}, l) \wedge \neg\exists p \ \text{Accettata}(p) \wedge \neg\text{Terminata}(p) \wedge \text{pren\_letto}(p, l) \wedge \text{doppio}(l, \text{FALSE}) \}$

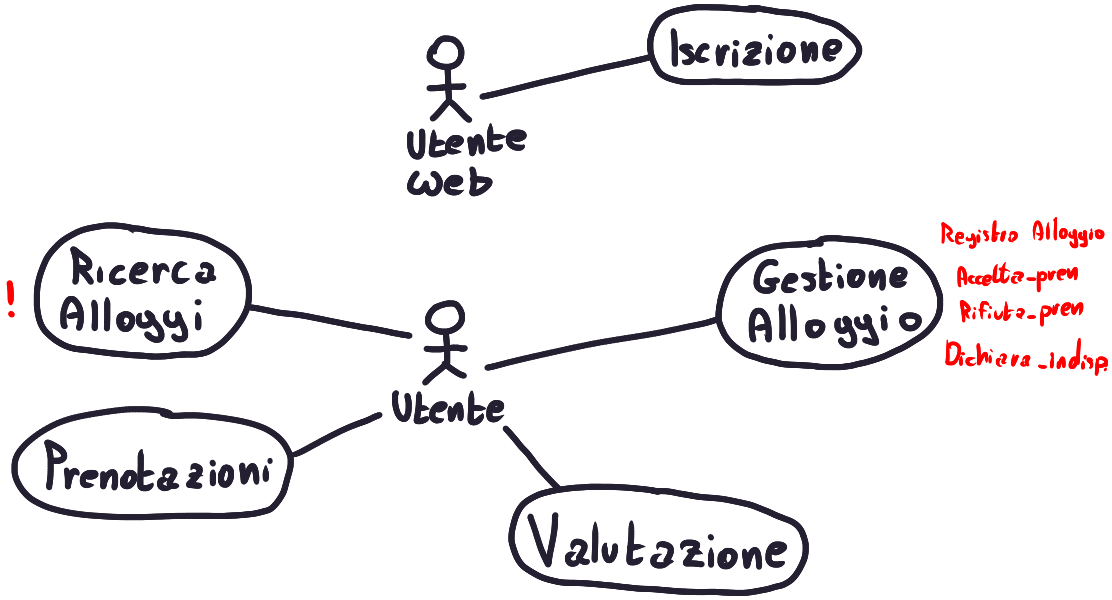
Result: $|P| + 2 \cdot |PD|$

[Versione 2024-03-29]

[continua alla pagina seguente]

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.

Mancano le segnature degli usecase

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

Operazioni dello use-case:

2 Specifica use-case: . . . . . (nome use-case)

Operazioni dello use-case:

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

Operazioni dello use-case:

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:

[Versione 2024-03-29]

[continua a pagina .....delle minute]

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

Ricerca Alloggi

cerca_ospitanti(c:Citta, dal:Data, al:Data, n:Intero): Alloggio[0..*]

- pre-cond: al > dal
- post-cond: $A = \{ a \mid \text{Alloggio}(a) \wedge [\exists u\ \text{ut\_all}(u,a) \wedge \text{ute\_citt}(u,c)] \wedge \exists k\ \text{posti\_disp}(a,k) \wedge k \geq n \wedge [\forall in, i, f\ [\text{Indisponibilita}(in) \wedge \text{inizio}(in,i) \wedge \text{fine}(in,f) \wedge \text{all\_ind}(a,in)] \rightarrow [f < dal \vee al < i]] \}$

Result = A

[Versione 2024-03-29]

[continua alla pagina seguente]

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

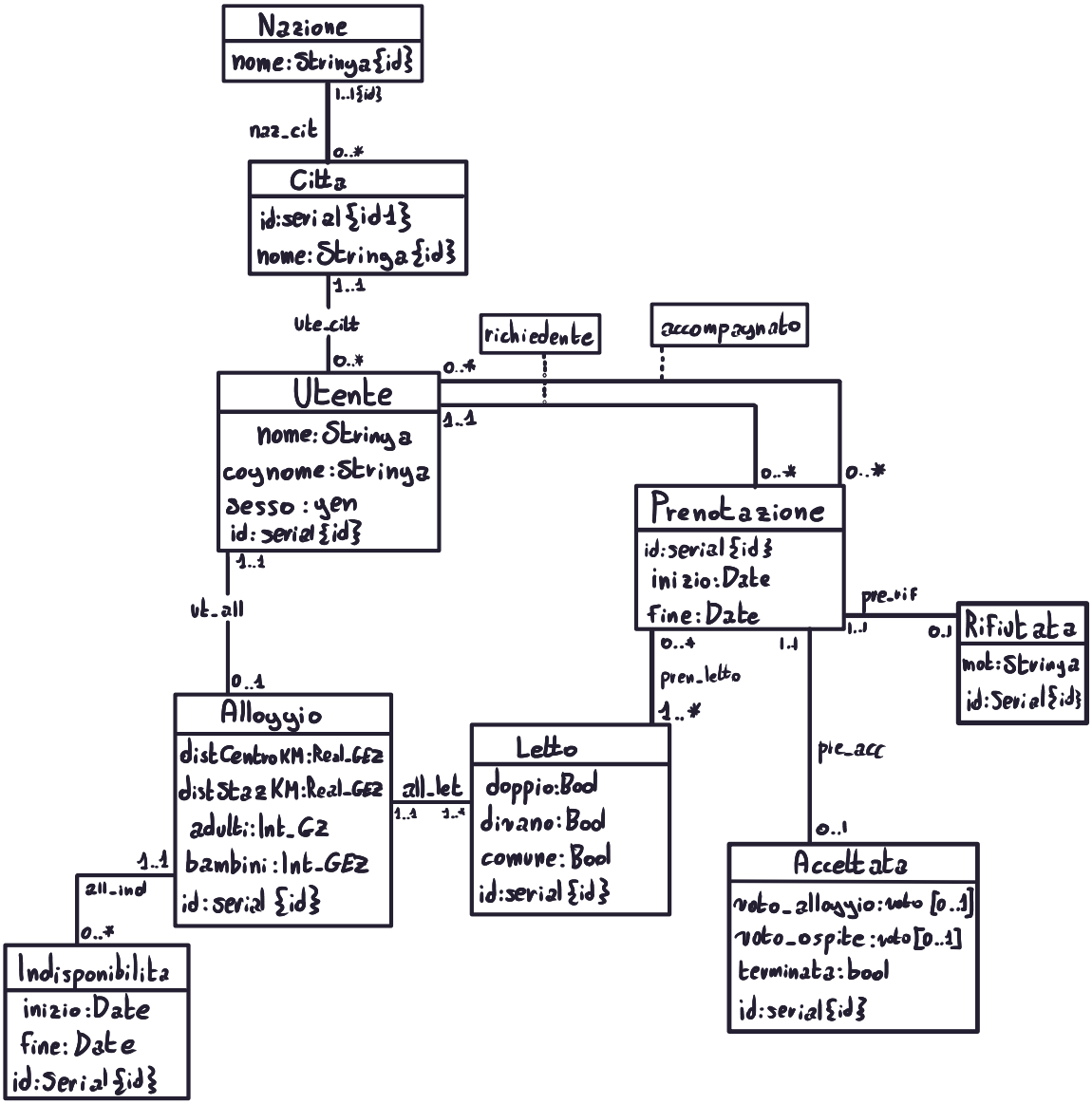
Corrispondenza tra domini concettuali e domini supportati dal DBMS

```
Create domain Stringa as varchar NOT NULL;
Create domain voto as Integer check(value>=0 and value<6);
create type gen as enum('M','F');

create domain Int-GZ as Integer check(value > 0);
create domain Int-GEZ as Integer check(value >= 0);
create domain Real-GEZ as Real check(value >= 0);
```

[Versione 2024-03-29]

Diagramma ER ristrutturato



[Versione 2024-03-29]

Breve descrizione delle scelte effettuate durante la ristrutturazione

Fusione su accettata

Sostituzione con ass. su Prenotazion

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.utente_no_pren_che_si_intersecano]
$\forall u, p1, p2, i1, i2, f1, f2 \ \neg Rifiutata(p1) \wedge \neg Rifiutata(p2)$
$[Utente(u)$
$\wedge [richiedente(u, p1) \vee accompagnato(u, p1)]$
$\wedge [richiedente(u, p2) \vee accompagnato(u, p2)]$
$inizio(p1, i1) \wedge inizio(p2, i2) \wedge Fine(p1, f1) \wedge Fine(p2, f2)]$
$\rightarrow [f1 < i2] \vee [f2 < i2]$

Ne mancano altri da riscrivere

[V.no_prenotazioni_casa_propria]
$\forall u, a, l [Utente(a) \wedge Letto(l) \wedge ut\_all(u, a) \wedge$
$all\_let(a, l)] \rightarrow [\neg \exists p \ Prenotazione(p) \wedge pren\_letto(p, l)$
$\wedge [richiedente(u, p) \vee accompagnato(u, p)]$

[V.valuta_tutte]
$\forall p, u$
$[Utente(p) \wedge \neg Terminata(p) \wedge \neg Rifiutata(p) \wedge [richiedente(u, p) \vee accompagnato(u, p)]$
$\rightarrow [\forall p1 \ Terminata(p1) \wedge [richiedente(u, p1) \vee accompagnato(u, p1)] \wedge$
$\exists k \ voto\_alloggio(p1, k)]$

[Versione 2024-03-29]

[continua alla pagina seguente]

Risposta alla Domanda 6 (segue)

[V. terminazione_pren]

$\forall p \; [\text{Accettata}(p) \wedge [\exists k \; \text{voto\_ospite}(p,k) \vee \exists k' \; \text{voto\_alloggio}(p,k')]] \rightarrow [\text{terminata}(p, \text{TRUE})]$

[V. compl_disj_pren]

$\forall p \; [\text{Prenotazione}(p) \wedge \exists r \; \text{pre\_rif}(p,r)] \rightarrow [\neg \exists a \; \text{pre\_acc}(p,a)]$

[V. Utente_richiede_non_accompagna]

$\forall u,p \; [\text{Utente}(u) \wedge \text{richiedente}(u,p)] \rightarrow [\neg \; \text{accompagnato}(u,p)]$

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** Nazione ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	nome						
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 accorpa le relazioni che implementano le seguenti relationship: .....

2 **Relazione** Citta ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	id	nome	nazione				
Domini	serial	Stringa	Stringa				

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

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

Fk nazione ref Nazione(nome);

Unique(nome, nazione);

La relazione accorpa le relazioni che implementano le seguenti relationship: naz_cit

3 **Relazione** Utente ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	nome	cognome	sesso	id	citta		
Domini	Stringa	Stringa	gen	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 ennupla, di dominio):

Fk citta ref Citta(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: ute_citt

4 **Relazione** Alloggio ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	distCentroKM	distStazKM	adulti	bambini	id	utente	
Domini	Real_GEZ	Real_GEZ	Int_GZ	Int_GZ	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 ennupla, di dominio):

Fk utente ref Utente(id);

Unique(utente);

La relazione accorpa le relazioni che implementano le seguenti relationship: ut_all

5 **Relazione** Indisponibilita ..... (nome) Derivante da: **entità** | **relationship** (cerchiare)

Attributi	inizio	Fine	id	alloggio			
Domini	Date	Date	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 ennupla, di dominio):

check(inizio <= Fine);

Fk alloggio ref Alloggio(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: all_ind

[Versione 2024-03-29]

6 **Relazione** Letto (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	id	doppio	divano	comune	alloggio			
Domini	serial	bool	bool	bool	Integer			

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 alloggio ref Alloggio(id); alloggio ⊆ Alloggio(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: all_let

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

Attributi	id	inizio	Fine	richiedente				
Domini	serial	Date	Date	id				

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 richiedente ref Utente(id); check(inizio <= Fine);

La relazione accorpa le relazioni che implementano le seguenti relationship: richiedente

8 **Relazione** accompagnato (nome) Derivante da: **entità** | ~~relationship~~ (cerchiare)

Attributi	utente	prenotazione						
Domini	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 ennupla, di dominio):

Fk utente ref Utente(id);

Fk prenotazione ref Prenotazione(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: .....

9 **Relazione** Rifiutata (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	mot	id	prenotazione					
Domini	Stringa	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 ennupla, di dominio):

unique(prenotazione);

Fk prenotazione ref Prenotazione(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: pre_rif

10 **Relazione** Accettata (nome) Derivante da: ~~entità~~ | **relationship** (cerchiare)

Attributi	voto_alloggio*	voto_ospite*	terminata	id	preno-zione			
Domini	voto	voto	bool	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 ennupla, di dominio):

unique(prenotazione); check(((voto_alloggio<>NULL OR voto_ospite<>NULL) AND terminata=TRUE) OR terminata='False');

Fk prenotazione ref Prenotazione(id); check(terminata='True' AND fine<cast(NOW() as Date)) OR terminata='False');

La relazione accorpa le relazioni che implementano le seguenti relationship: pre_acc

[Versione 2024-03-29]

6 **Relazione** pren_letto ..... (nome) Derivante da: **entità** | ~~**relationship**~~ (cerchiare)

Attributi	pren	letto						
Domini	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 ennupla, di dominio):

FK pren ref Prenota e(ou(id)
FK letto ref Letto(id);

La relazione accorpa le relazioni che implementano le seguenti relationship: .....

7 **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 accorpa le relazioni che implementano le seguenti relationship: .....

8 **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 accorpa le relazioni che implementano le seguenti relationship: .....

9 **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 accorpa le relazioni che implementano le seguenti relationship: .....

10 **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 accorpa le relazioni che implementano le seguenti relationship: .....

[V. no_inters_prev_accettate]
Trigger: Insert o Update Accettata

```
Error = EXISTS( SELECT *
                FROM Accompagnato ac, Utente u, Prenotazione pr
                WHERE ac.utente = u.id AND
                (u.id = new.richiedente OR
                 ac.prenotazione = new.id) AND
                 ac.prenotazione = pr.id AND
                 (pr.inizio, pr.fine) INTERSET (new.inizio, new.fine) = 'True');
```

Se Error = True, segnala errore,
altrimenti permetti operazione.

[V. no_2_~~accettate~~_sullo_stesso_letto]
Trigger: Insert o Update Accettata

```
FROM  pren_letto pl, Accettata a
WHERE pl.pren = new.id
AND
```

Ulteriori vincoli esterni

Per ogni ulteriore vincolo esterno (non ancora espresso perché non definibile mediante vincoli di chiave, foreign key, ennupla, dominio, inclusione), progettare un trigger che lo implementi, definendo: (a) gli eventi da intercettare (inserimento, modifica, eliminazione di ennuple); (b) quando intercettare tali eventi (appena prima o subito dopo l'evento intercettato); (c) la relativa funzione in pseudo-codice con SQL immerso che implementa il controllo del vincolo.

[continua alla pagina seguente]

Risposta alla Domanda 7 (segue)

Domanda 8 (30 minuti; 45 minuti al massimo) Proseguire la fase di progettazione dell'applicazione producendo le specifiche realizzative delle operazioni di 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

```
cerca_ospitanti(c:Integer, dal:Data, al:Data, n:Int-Gz):Insieme(<Integer>) //id degli alloggi

• pre-condizioni: dal < al

• post-condizioni: Q = (SELECT al.id
                        FROM Alloggio al, Utente u, Citta cit, Indisponibilita in
                        WHERE cit.id = c
                        AND u.citta = cit.nome
                        AND al.utente = u.id
                        AND posti_disp(al.id) >= n
                        AND in.alloggio = al.id
                        AND (in.inizio,in.fine) INTERSET (dal, al) = 'FALSE' );

                    Result = Q
```


[continua alla pagina seguente]

Risposta alla Domanda 8 (segue)

```
CREATE FUNCTION posti_disp(a:Integer): Int_GEz

begin:

   S = SELECT count(*)
       FROM
      ((SELECT pr.id
        FROM Letto l, pren_letto, Prenotazione pr
        WHERE l.alloggio = a
        AND l.doppio: 'False'
        AND pr.id = pren_letto.pren
        AND pren_letto.letto = l.id) EXCEPT

       (SELECT pr.id
        FROM Letto l, pren_letto, Prenotazione pr, Accettata ac
        WHERE l.alloggio = a
        AND l.doppio: 'False'
        AND pr.id = pren_letto.pren
        AND pren_letto.letto = l.id
        AND ac.prenotazione = pr.id
        AND ac.terminata = 'False'))

   D = SELECT count(*)*2
       FROM
      ((SELECT pr.id
        FROM Letto l, pren_letto, Prenotazione pr
        WHERE l.alloggio = a
        AND l.doppio: 'True'
        AND pr.id = pren_letto.pren
        AND pren_letto.letto = l.id) EXCEPT

       (SELECT pr.id
        FROM Letto l, pren_letto, Prenotazione pr, Accettata ac
        WHERE l.alloggio = a
        AND l.doppio: 'True'
        AND pr.id = pren_letto.pren
        AND pren_letto.letto = l.id
        AND ac.prenotazione = pr.id
        AND ac.terminata = 'False'))

    result = S+D
```

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.]