

Progetto 20070703 (P.20070703) - DormoDaTe

Sebastiano Deodati

21 maggio 2024

Indice

1	Dati di interesse e funzionalità richieste	2
2	Diagramma ER	4
3	Dizionario dei dati	5
4	UML	7
5	Specifiche degli use-case	8
6	Scelta del DBMS	14
7	Ristrutturazione del diagramma ER	15
8	Transizione dei tipi di dati concettuali in tipi standard SQL	16

Dati di interesse e funzionalità richieste

1. Utente:

- 1.1. Nickname (univoco)
- 1.2. Nome
- 1.3. Cognome
- 1.4. Sesso
- 1.5. Data di nascita
- 1.6. Città di residenza (req. 2)
- 1.7. Ospitalità offerta (req. 3)

2. Città:

- 2.1. Nome
- 2.2. Provincia

3. Casa:

- 3.1. Indirizzo (req. 4)
- 3.2. Distanza (mt) dal centro città
- 3.3. Distanza (mt) dalla stazione autobus/metro/treno più vicina
- 3.4. Adulti residenti
- 3.5. Bambini residenti
- 3.6. Camere (req. 5)

4. Indirizzo:

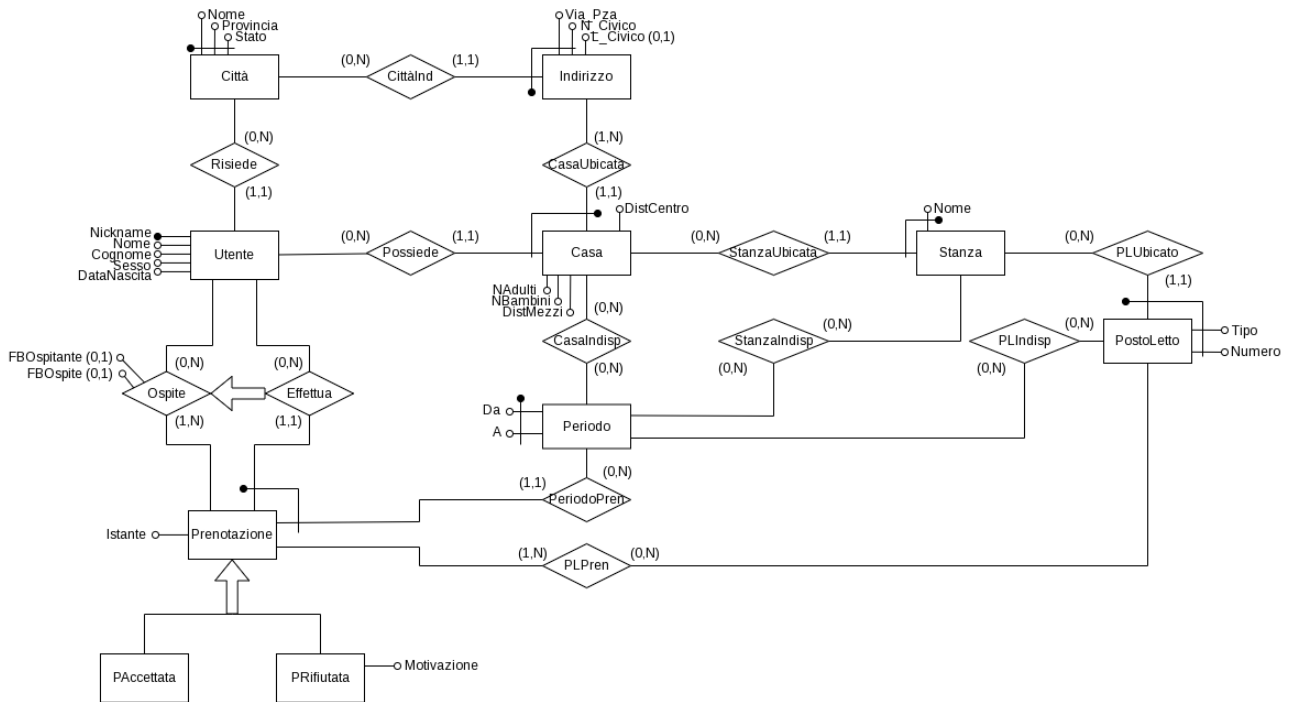
- 4.1. Via/Piazza
- 4.2. Numero civico
- 4.3. Città (req. 2)

5. Camera:

- 5.1. Ubicazione (req. 3)
- 5.2. Posti letto (req. 6)
- 5.3. Eventuali periodi di indisponibilità

6. Posto letto:
 - 6.1. Ubicazione (req. 5)
 - 6.2. Tipo (letto singolo, letto matrimoniale, divano letto)
 - 6.3. Eventuali periodi di indisponibilità
7. Prenotazione:
 - 7.1. Richiedente (req. 1)
 - 7.2. Eventuali accompagnatori (req. 1)
 - 7.3. Scelta dei posti letto (req. 6)
 - 7.4. Periodo di prenotazione
 - 7.5. Stato (in attesa, accettata, rifiutata)
 - 7.5.1. Motivazione del rifiuto (se rifiutata)
 - 7.6. Istante di prenotazione (per controllo su feedback obbligatori)
8. Feedback:
 - 8.1. Prenotazione di riferimento (req. 7)
 - 8.2. Feedback di tutti gli ospiti (voto da 0 a 5)
 - 8.3. Feedback dell'ospitante sugli ospiti (facoltativo)
9. Funzionalità richieste:
 - 9.1. Il sistema deve consentire a tutti gli iscritti di aggiornare in qualunque momento le informazioni circa la sua disponibilità ad ospitare persone, per ogni sistemazione, specificando date in cui una data sistemazione non è disponibile
 - 9.2. I viaggiatori che desiderano organizzare un viaggio usufruendo del servizio devono poter interrogare il sistema per ottenere l'insieme di tutti gli iscritti della città desiderata che sono disposti a ricevere ospiti nel periodo richiesto e hanno, per le date richieste, posti sufficienti disponibili
 - 9.3. Data una richiesta di prenotazione, il relativo ospitante deve poter liberamente accettare o rifiutare la richiesta, in quest'ultimo caso specificando il motivo
 - 9.4. Al termine di un periodo di ospitazione, l'ospitante deve poter esprimere un feedback (voto da 0 a 5) sui suoi ospiti, mentre gli ospiti devono essere costretti ad esprimere un feedback sull'ospitante, impedendogli di effettuare un'altra prenotazione finché non avranno espresso il loro feedback

Diagramma ER



Dizionario dei dati

Entità Utente:

Attributo	Tipo	Note
Nickname	stringa	
Nome	Stringa	
Cognome	Stringa	
Sesso	{ 'M', 'F' }	
Nascita	dataora	

Entità Città:

Attributo	Tipo	Note
Nome	stringa	
Provincia	stringa	
Stato	stringa	

Entità Indirizzo:

Attributo	Tipo	Note
Via_Pza	stringa	
N_Civico	intero ≥ 0	0 indica SNC
L_Civico(0,1)	lettera	Lettera del numero civico (opzionale)

Entità Casa:

Attributo	Tipo	Note
DistMezzi	intero > 0	Distanza in metri dalla stazione dei trasporti pubblici più vicina
DistCentro	intero > 0	Distanza in metri dal centro città
NAdulti	intero ≥ 0	Adulti residenti
NBambini	intero ≥ 0	Bambini residenti

Entità Periodo:

Attributo	Tipo	Note
Da	dataora	Inizio del periodo di tempo
A	dataora	Fine del periodo di tempo

Entità Stanza:

Attributo	Tipo	Note
Nome	stringa	

Entità PostoLetto:

Attributo	Tipo	Note
Numero	intero ≥ 0	Un numero incrementale per identificare i posti letto dello stesso tipo in una stanza
Tipo	{'Letto singolo', 'Letto matrimoniale', 'Divano letto'}	Il tipo identifica anche il numero di posti

Entità Prenotazione:

Attributo	Tipo	Note
Istante	dataora	Istante della prenotazione, utilizzato per impedire agli utenti di effettuare prenotazioni in mancanza di feedback relativi a vecchie prenotazioni

Entità PRifiutata:

Attributo	Tipo	Note
Motivazione	stringa	Motivo del rifiuto

Relazione Ospite:

Attributo	Tipo	Note
FBOspite(0,1)	intero[0,5]	Feedback espresso dall'ospitante sull'ospite
FBOspitante(0,1)	intero[0,5]	Feedback espresso dall'ospite sull'ospitante

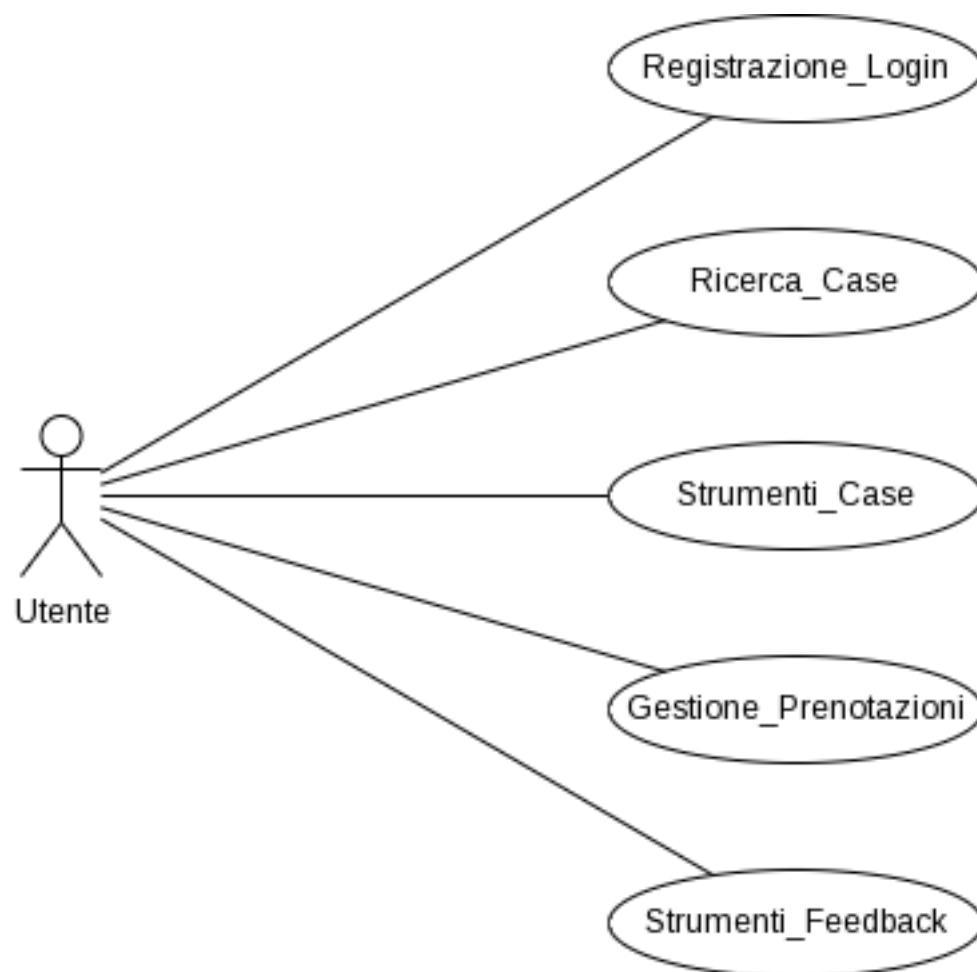
Vincoli esterni:

V.Indirizzo.lettera: $\forall i \text{ Indirizzo}(i) \wedge \text{NCivico}(i, 0) \rightarrow \neg \exists l \text{ lettera}(l) \wedge \text{LCivico}(i, l)$

V.Periodo.nonneg: $\forall p, i, f \text{ Periodo}(p) \wedge \text{Da}(p, i) \wedge \text{A}(p, f) \rightarrow i \leq f$

V.Utente.fb_obbligatorio: $\forall u, p, i \text{ Utente}(u) \wedge \text{Prenotazione}(p) \wedge \text{Istante}(p, i) \wedge \text{Ospite}(u, p) \rightarrow \forall p', i' [\text{Prenotazione}(p') \wedge \text{Istante}(p', i') \wedge i' < i] \rightarrow \exists f \text{ "intero}[0, 5]"(f) \wedge \text{FBOspitante}(u, p', f)$

UML



Specifiche degli use-case

Specifiche use-case Registrazione_Login

Registra(nickname: stringa, nome: stringa, cognome: stringa, sesso: {'M', 'F'}, nascita: data, residenza: Città) : Utente

precondizioni: $\neg \exists u \text{ Utente}(u) \wedge \text{Nickname}(u, \text{nickname})$

postcondizioni:

modifica al livello estensionale dei dati:

nuovi elementi del dominio di interpretazione: u

nuove ennuple di predicati:

Utente(u)

Nickname(u, nickname)

Nome(u, nome)

Cognome(u, cognome)

Sesso(u, sesso)

DataNascita(u, nascita)

Risiede(u, residenza)

valore di ritorno: $result = u$

Login(nickname: Stringa) : Utente

precondizioni: $\exists u \text{ Utente}(u) \wedge \text{Nickname}(u, \text{nickname})$

postcondizioni:

modifica al livello estensionale dei dati: nessuna

valore di ritorno:

Sia u t.c. $\text{Utente}(u) \wedge \text{Nickname}(u, \text{nickname})$

$result = u$

Specifiche use-case Ricerca_Case

Cerca(citta: Citta, p: Periodo, nposti: intero > 0) : Casa(0,N)

precondizioni: nessuna

postcondizioni:

modifica al livello estensionale dei dati: nessuna

valore di ritorno:

Sia $C_{citta} = \{c \mid Casa(c) \wedge [\exists i Indirizzo(i) \wedge CasaUbicata(c, i) \wedge CittaInd(citta, i)]\}$

sia $C_{ok} = \{c \in C_{citta} \mid "Operazioni_Case".Posti_disponibili(c, p) > nposti\}$

$result = C_{ok}$

Specifiche use-case Strumenti_Case

PL_disponibile(pl: PostoLetto, p: Periodo) : booleano

precondizioni: nessuna

postcondizioni:

modifica al livello estensionale dei dati: nessuna

valore di ritorno:

$result = [\exists c, s PLUbicato(p, s) \wedge StanzaUbicata(s, c)]$

$\wedge [\neg \exists pr, p' Prenotazione(pr) \wedge PLPren(pr, p') \wedge p \cap p' \neq \emptyset]$

$\wedge [\neg \exists i, j, k PLIndisp(p, i) \wedge StanzaIndisp(s, j) \wedge CasaIndisp(c, k)$

$\wedge (i \cup j \cup k) \cap p \neq \emptyset]$

Posti_disponibili(c: Casa, p: Periodo) : intero ≥ 0

precondizioni: nessuna

postcondizioni:

modifica al livello estensionale dei dati: nessuna

valore di ritorno:

Sia $P_{L1} = \{pl \mid PostoLetto(pl) \wedge Tipo(pl, 'Lettosingolo')$

$\wedge "Strumenti_Case".PL_disponibile(pl, p) = vero\}$

Sia $P_{L2} = \{pl \mid PostoLetto(pl) \wedge \neg(Tipo(pl, 'Lettosingolo'))$

$\wedge "Strumenti_Case".PL_disponibile(pl, p) = vero\}$

$result = |P_{L1}| + 2 * |P_{L2}|$

Registra_Casa(u: Utente, indirizzo: Indirizzo, distCentro: intero ≥ 0 , distMezzi: intero ≥ 0 , nAdulti: intero ≥ 0 , nBambini: intero ≥ 0) : Casa

precondizioni: nessuna

postcondizioni:

modifica al livello estensionale dei dati:

nuovi elementi del dominio di interpretazione: c

nuove ennuple di predicati:

Casa(c)

DistCentro(c, distCentro)

DistMezzi(c, distMezzi)

NAdulti(c, nAdulti)

NBambini(c, nBambini)

Possiede(u, c)

CasaUbicata(c, indirizzo)

valore di ritorno: *result* = c

Registra_Stanza(u: Utente, c: Casa, nome: stringa) : Stanza

precondizioni: nessuna

postcondizioni:

modifica al livello estensionale dei dati:

nuovi elementi del dominio di interpretazione: s

nuove ennuple di predicati:

Stanza(s)

Nome(s, nome)

StanzaUbicata(s, c)

valore di ritorno: *result* = s

Registra_PostoLetto(u: Utente, s: Stanza, tipo: {'Letto singolo', 'Letto matrimoniale', 'Divano letto'}, n: intero ≥ 0) : PostoLetto

precondizioni: $\neg \exists p \text{ PostoLetto}(p) \wedge \text{PLUbicato}(p, s) \wedge \text{Tipo}(p, \text{tipo}) \wedge \text{Numero}(p, n)$

postcondizioni:

modifica al livello estensionale dei dati:

nuovi elementi del dominio di interpretazione: p

nuove ennuple di predicati:

PostoLetto(p)

Tipo(p, tipo)

Numero(p, n)

PLUbicato(p, s)

valore di ritorno: *result* = p

Indisponibilita_Casa(c: Casa, p: Periodo)precondizioni: $\neg CasaIndisp(c, p)$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati: CasaIndisp(c, p)

valore di ritorno: nessuno

Indisponibilita_Stanza(s: Stanza, p: Periodo)precondizioni: $\neg StanzaIndisp(s, p)$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati: StanzaIndisp(s, p)

valore di ritorno: nessuno

Indisponibilita_PL(pl: PostoLetto, p: Periodo)precondizioni: $\neg PLIndisp(pl, p)$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati: PLIndisp(pl, p)

valore di ritorno: nessuno

Specifiche use-case Gestione_Prenotazioni**Prenota**(u: Utente, A: Utente(0,N), PL: PostoLetto(1,N), p: Periodo) : Prenotazione

precondizioni:

 $\exists c Casa(c) \wedge [\forall pl \in PL \exists s PLUbicato(pl, s) \wedge StanzaUbicata(s, c)]$ $"Strumenti_Case".PL_disponibile(pl, p) \forall pl \in PL$

postcondizioni:

modifica al livello estensionale dei dati:

nuovi elementi del dominio di interpretazione: pr

nuove ennuple di predicati:

Prenotazione(pr)

Istante(pr, *adesso*)

Effettua(u, pr)

Ospite(u, pr)

Ospite(a, pr) $\forall a \in A$

PeriodoPren(p, pr)

PLPren(pl, pr) $\forall pl \in PL$ valore di ritorno: *result = pr*

Accetta_Prenotazione(h: Utente, p: Prenotazione)

precondizioni: $\exists pl, s, c \text{ } PLPRen(pl, p) \wedge PLUbicato(pl, s) \wedge StanzaUbicata(s, c) \wedge Possiede(h, c)$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati: PAccettata(p)

valore di ritorno: nessuno

Rifiuta_Prenotazione(h: Utente, p: Prenotazione, motivo: stringa)

precondizioni: $\exists pl, s, c \text{ } PLPRen(pl, p) \wedge PLUbicato(pl, s) \wedge StanzaUbicata(s, c) \wedge Possiede(h, c)$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati:

PRifiutata(p)

Motivazione(p, motivo)

valore di ritorno: nessuno

Annulla_Prenotazione(u: Utente, p: Prenotazione)

precondizioni: $Effettua(u, p) \wedge \neg PRifiutata(p)$

$\wedge \neg \exists per, i [PeriodoPren(per, p) \wedge Da(per, i) \wedge i \geq adesso]$

postcondizioni:

modifica al livello estensionale dei dati:

elementi del dominio di interpretazione che non esistono più: p

ennuple di predicati che non valgono più:

Prenotazione(p)

PAccettata(p)

valore di ritorno: nessuno

Specifiche use-case Strumenti_Feedback

Valuta_Ospitante(u: Utente, p: PAccettata, voto: intero [0,5])

precondizioni: $Ospite(u, p) \wedge \exists per, f [PeriodoPren(per, p) \wedge A(per, f) \wedge f \leq adesso]$

postcondizioni:

modifica al livello estensionale dei dati:

dominio di interpretazione: invariato

nuove ennuple di predicati: FBOspitante(u, p, voto)

valore di ritorno: nessuno

Valuta_Ospite(h: Utente, o: Utente, p: PAccettata, voto: intero [0,5])
 precondizioni: $Ospite(o, p) \wedge \exists pl, s, c$
 $[PLPren(pl, p) \wedge PLUbicato(pl, s) \wedge StanzaUbicata(s, c) \wedge Possiede(h, c)]$
 $\wedge \exists per, f [PeriodoPren(per, p) \wedge A(per, f) \wedge f \leq adesso]$
 postcondizioni:
 modifica al livello estensionale dei dati:
 dominio di interpretazione: invariato
 nuove ennuple di predicati: FBOspite(o, p, voto)
 valore di ritorno: nessuno

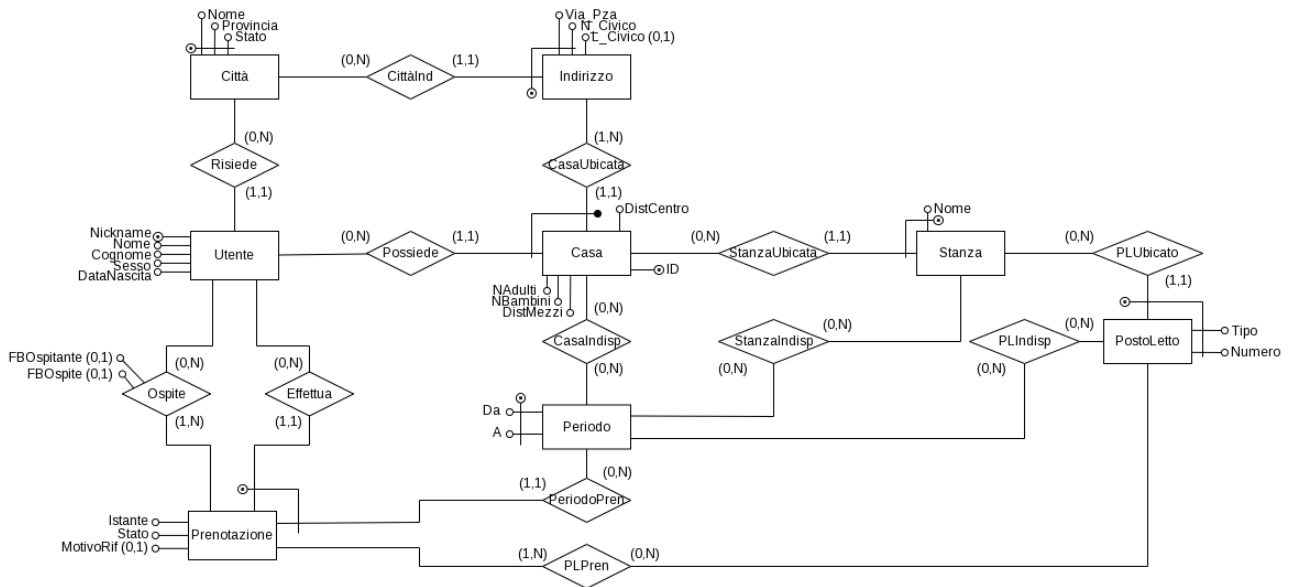
Feedback_Ospitante(h: Utente) : intero[0,5](0,N)
 precondizioni: nessuna
 postcondizioni:
 modifica al livello estensionale dei dati: nessuna
 valore di ritorno:
 Sia $H = \{(p, v) \mid \exists o [Effettua(o, p) \wedge \exists pl, s, c$
 $[PLPren(pl, p) \wedge PLUbicato(pl, s) \wedge StanzaUbicata(s, c)$
 $\wedge Possiede(h, c)] \wedge FBOspitante(o, p, v)\}$
 $result = \{v \mid \forall (p, v) \in H\}$

Feedback_Ospite(o: Utente) : intero[0,5](0,N)
 precondizioni: nessuna
 postcondizioni:
 modifica al livello estensionale dei dati: nessuna
 valore di ritorno:
 Sia $O = \{(p, v) \mid Ospite(o, p) \wedge FBOspite(o, p, v)\}$
 $result = \{v \mid \forall (p, v) \in O\}$

Scelta del DBMS

Il database verrà implementato sul DBMS PostgreSQL.

Ristrutturazione del diagramma ER



Transizione dei tipi di dati concettuali in tipi standard SQL

Entità Utente:

Attributo	Tipo	Note
Nickname	varchar(50)	
Nome	varchar(30)	
Cognome	varchar(30)	
Sesso	gender_t	
Nascita	datetime	

Entità Città:

Attributo	Tipo	Note
Nome	varchar(30)	
Provincia	varchar(20)	
Stato	varchar(20)	

Entità Indirizzo:

Attributo	Tipo	Note
Via_Pza	varchar(50)	
N_Civico	int_nonneg	0 indica SNC
L_Civico(0,1)	char(1)	Lettera del numero civico (opzionale)

Entità Casa:

Attributo	Tipo	Note
ID	int_pos	ID incrementale univoco introdotto per motivi di performance
DistMezzi	int_pos	Distanza in metri dalla stazione dei trasporti pubblici più vicina
DistCentro	int_pos	Distanza in metri dal centro città
NAdulti	int_nonneg	Adulti residenti
NBambini	int_nonneg	Bambini residenti

Entità Periodo:

Attributo	Tipo	Note
Da	datetime	Inizio del periodo di tempo
A	datetime	Fine del periodo di tempo

Entità Stanza:

Attributo	Tipo	Note
Nome	varchar(20)	

Entità PostoLetto:

Attributo	Tipo	Note
Numero	int_nonneg	Un numero incrementale per identificare i posti letto dello stesso tipo in una stanza
Tipo	sist_t	Il tipo identifica anche il numero di posti

Entità Prenotazione:

Attributo	Tipo	Note
Istante	datetime	Istante della prenotazione, utilizzato per impedire agli utenti di effettuare prenotazioni in mancanza di feedback relativi a vecchie prenotazioni
Stato	pstatus	Stato della prenotazione (accettata, rifiutata, in attesa)
MotivoRif(0,1)	varchar(100)	Motivo del rifiuto (se la prenotazione è stata rifiutata)

Relazione Ospite:

Attributo	Tipo	Note
FBOspite(0,1)	rate_t	Feedback espresso dall'ospitante sull'ospite
FBOspitante(0,1)	rate_t	Feedback espresso dall'ospite sull'ospitante

Vincoli esterni:

V.Indirizzo.lettera: $\forall i \text{ Indirizzo}(i) \wedge NCivico(i,0) \rightarrow \neg \exists l \text{ lettera}(l) \wedge LCivico(i,l)$

V.Periodo.nonneg: $\forall p, i, f \text{ Periodo}(p) \wedge Da(p,i) \wedge A(p,f) \rightarrow i \leq f$

V.Utente.fb_obbligatorio: $\forall u, p, i \text{ Utente}(u) \wedge Prenotazione(p) \wedge Istante(p,i) \wedge Ospite(u,p) \rightarrow \forall p', i' [\text{Prenotazione}(p') \wedge Istante(p', i') \wedge i' < i] \rightarrow \exists f \text{ "intero}[0,5]"(f) \wedge FBOspitante(u, p', f)$

V.Effettua.ospite: $\forall u, p \text{ Utente}(u) \wedge Effettua(u, p) \rightarrow Ospite(u, p)$

V.Prenotazione.motivazione_obbligatoria: $\forall p, m \text{ Prenotazione}(p) \wedge Stato(p, 'r') \wedge MotivoRif(p, m) \rightarrow m \neq NULL$

Tipi di dati personalizzati

- gender_t: enum('m', 'f')
- int_nonneg: integer check value >= 0
- int_pos: integer check value > 0
- sist_t: enum('Letto singolo', 'Letto matrimoniale', 'Divano letto')

- `pstatus`: `enum('a', 'r', 'p')`
- `rate_t`: integer check value between 0 and 5

Schema concettuale

Utente(Nickname: varchar(50), Nome: varchar(30), Cognome: varchar(30), Sesso: gender_t,
Nascita: datetime, Citta: varchar(30), Prov: varchar(20), Stato: varchar(20))
Vincolo foreign key: (Citta, Prov, Stato) references Citta(Nome, Provincia, Stato)
Vincolo ennupla: (CURRENT_DATE - Nascita) \leq interval('18 years')

Citta(Nome: varchar(30), Prov: varchar(20), Stato: varchar(20))

Casa(ID: int_pos, Proprietario: varchar(50), Via_Pza: varchar(50), N_Civico: int_nonneg, Citta: varchar(30),
Prov: varchar(20), Stato: varchar(20), NAdulti: int_nonneg, NBambini: int_nonneg,
DistCentro: int_pos, Distmezzi: int_pos)
Vincolo foreign key: Proprietario references Utente(Nickname)
Vincolo foreign key: (Via_Pza, N_Civico, Citta, Prov, Stato) references Indirizzo(Via_Pza, N_Civico, Citta, Prov, Stato)
Vincolo chiave: (Proprietario, Via_Pza, N_Civico, Citta, Prov, Stato)

Indirizzo(Via_Pza: varchar(50), N_Civico: int_nonneg, Citta: varchar(30), Prov: varchar(20),
Stato: varchar(20), L_Civico*: char(1))
Vincolo foreign key: (Citta, Prov, Stato) references Citta(Nome, Provincia, Stato)

Stanza(Casa: int_pos, Nome: varchar(20))
Vincolo foreign key: Casa references Casa(ID)

PostoLetto(Casa: int_pos, Stanza: int_pos, Numero: int_nonneg, Tipo: sist_t)
Vincolo foreign key: (Casa, Stanza) references Stanza(Casa, Nome)

Periodo(Da: datetime, A: datetime)
Vincolo ennupla: Da \leq A

CasaIndisp(Casa: int_pos, Da: datetime, A: datetime)
Vincolo foreign key: Casa references Casa(ID)
Vincolo foreign key: (Da, A) references Periodo(Da, A)

StanzaIndisp(Casa: int_pos, Stanza: varchar(20), Da: datetime, A: datetime)
 Vincolo foreign key: (Casa, Stanza) references Stanza(Casa, Nome)
 Vincolo foreign key: (Da, A) references Periodo(Da, A)

PLIndisp(Casa: int_pos, Stanza: varchar(20), NPL: int_nonneg, TPL: sist_t,
Da: datetime, A: datetime)
 Vincolo foreign key: (Casa, Stanza, NPL, TPL) references
 PostoLetto(Casa, Stanza, Numero, Tipo)
 Vincolo foreign key: (Da, A) references Periodo(Da, A)

Prenotazione(Resp: varchar(50), Da: datetime, A: datetime, Istante: datetime, Stato: pstatus,
 MotivoRif*: varchar(100))
 Vincolo foreign key: Resp references Utente(Nickname)
 Vincolo foreign key: (Da, A) references Periodo(Da, A)
 Vincolo inclusione: (Resp, Da, A) \subseteq Ospite(Resp, Da, A)
 Vincolo ennupla: istante \leq CURRENT_TIMESTAMP

Ospite(Resp: varchar(50), Da: datetime, A: datetime, Ospite: varchar(50), FBOspite*: rate_t,
 FBOspitante*: rate_t)
 Vincolo foreign key: (Resp, Da, A) references Prenotazione (Resp, Da, A)
 Vincolo foreign key: Ospite references Utente(Nickname)

PLPren(Casa: int_pos, Stanza: varchar(20), NPL: int_nonneg, TPL: sist_t, Resp: varchar(50),
Da: datetime, A: datetime)
 Vincolo foreign key: (Casa, Stanza, NPL, TPL) references
 PostoLetto(Casa, Stanza, Numero, Tipo)
 Vincolo foreign key: (Resp, Da, A) references Prenotazione (Resp, Da, A)