

PROGETTO DATA BASE MILANESIO – MUSSO

MATRICOLE 946397-964874

1 Progettazione concettuale

1.1 Requisiti iniziali

Si vuole realizzare una base di dati per un servizio che permette di affittare e prenotare alloggi di vario tipo ad esempio interi appartamenti, stanze private (camera privata e spazi comuni) e stanze condivise (spazio in comune e camera condivisa).

Gli utenti si registrano al servizio fornendo indirizzo email, password, nome, cognome, numero o numeri di telefono.

Se l'utente fornisce la foto della carta d'identità, viene riconosciuto come verificato. Inoltre, l'utente deve indicare un metodo di pagamento per poter prenotare. Gli utenti possono essere ospiti o "host" ovvero possono a loro volta ospitare altri utenti del servizio in uno o più alloggi di loro proprietà. Inoltre, gli "host" possono diventare "superhost" se soddisfano i seguenti requisiti:

- Devono aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.
- Devono aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.
- Devono aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.

Gli utenti superhost ricevono un badge sul loro profilo.

Gli alloggi sono descritti indicando un nome, l'indirizzo (visibile all'ospite solo quando la prenotazione è confermata, altrimenti è visibile solo il comune), una descrizione, il prezzo per notte per persona e i costi di pulizia, delle foto, i servizi (ad esempio, cucina, wi-fi, lavatrice, ecc.), numero di letti e orario di check-in e check-out oltre all'host a cui appartiene, il rating medio e il numero di recensioni.

Gli utenti possono aggiungere alcune case tra i preferiti. Gli utenti possono avere diverse liste, ad esempio in base al viaggio che vogliono compiere. Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti. Se gli ospiti sono a loro volta utenti del servizio, se ne possono indicare i nominativi.

La prenotazione deve essere confermata o rifiutata dall'host. La prenotazione ha un costo totale e se confermata viene eseguito il pagamento. Inoltre, la prenotazione può essere cancellata sia dall'ospite che dall'host.

Al termine del soggiorno, gli ospiti e gli host si possono valutare a vicenda. La recensione fatta dagli ospiti comprende due testi (uno per l'appartamento e uno per l'host) e una serie di punteggi in una scala da 1 a 5 su dimensioni come pulizia, comunicazione, posizione, qualità/prezzo. La valutazione complessiva del soggiorno è una media delle valutazioni ricevute

sulle singole dimensioni. Le recensioni degli host comprendono solo un commento testuale. Le recensioni possono essere visibili o non visibili.

Diventano visibili quando entrambi hanno fatto la recensione oppure se uno dei due non ha fatto la recensione, l'altra diventa visibile dopo 7 giorni dalla fine del soggiorno.

Gli host e gli ospiti possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione. Le recensioni sono visibili sui profili degli utenti suddivise in base a quelle ricevute come ospite e come host. La base di dati deve supportare le seguenti operazioni:

- Una volta a settimana viene effettuato un calcolo per aggiornare il tasso di cancellazione di ciascun host.
- Una volta al giorno si controllano le condizioni per la qualifica di superhost e viene aggiornato lo status degli host.
- Una volta al mese viene calcolata la classifica degli alloggi più graditi

1.2 Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Utente	Persone iscritte al sito, possono essere semplici ospiti oppure host	Ospite, host	Immobile, commento, telefono, prenotazione, metodo di pagamento, recensione host, recensione ospite
Immobile	Alloggi prenotabili dal sito	case, stanze condivise, interi appartamenti, stanze private, alloggi	Immagine, categorie di servizio, prenotazione, utente
Prenotazione	Alloggio prenotato dall'utente		Utente, immobile, recensione host, recensione ospite, metodo di pagamento
Recensione host	Valutazione dell'ospite da parte dell'host		Utente, prenotazione, thread
Recensione ospite	Valutazione del soggiorno da parte dell'ospite composto da punteggi da 0 a 5 per quattro parametri, un commento dell'immobile e un commento dell'host		Utente, prenotazione, thread

Thread	Discussione che è possibile aprire sotto le review in cui si è coinvolti	discussione	Commento, recensione host, recensione ospite
Commento	Messaggio lasciato sotto ad una recensione		Utente, thread
Metodo di pagamento	Modalità di pagamento inserito dall'utente per pagare le prenotazioni effettuate		Utente, prenotazione
Telefono	Numero telefonico		utente
Immagine	Foto descrittiva dell'immobile		immobile
Categoria di servizio	Categorie nelle quali l'host può raggruppare alcuni servizi		Servizio, immobile
Servizio	Servizio offerto durante il soggiorno in un immobile		Categoria di servizio
Lista	Immobili raccolti in gruppi dall'utente		Immobile, utente

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

Frasi di carattere generale

Si vuole realizzare una base di dati per un servizio che permette di affittare e prenotare alloggi di vario tipo

Frasi relative agli alloggi

Gli alloggi sono suddivisi per tipo:

- Interi appartamenti;
- Stanze private;
- Stanze condivise;

Sono indicati con un nome, indirizzo, prezzo per notte a persona, costi pulizia, foto, servizi (cucina, wi-fi, lavatrice...etc.), numero letti, orario check-in e check-out, host a cui appartiene, rating medio e numero di recensioni.

Frase relative agli utenti:

Gli utenti si registrano al servizio fornendo indirizzo e-mail, password, nome, cognome, numero o numeri di telefono.

L'utente viene riconosciuto come verificato se fornisce la foto della carta d'identità.

Devono indicare un metodo di pagamento per poter prenotare.

Gli utenti si distinguono in:

- Ospiti
- Host (ovvero possono a loro volta ospitare altri utenti)

Gli host diventano superhost e ricevono un badge sul loro profilo se seguono determinate regole.

Gli utenti possono aggiungere alcune case tra i preferiti e possono creare diverse liste.

Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti. Quando gli ospiti sono a loro volta utenti del servizio, se ne indicano i nominativi.

Al termine del soggiorno, gli ospiti e gli host si possono valutare a vicenda.

Frase relative alle recensioni

Al termine del soggiorno, gli ospiti e gli utenti si possono valutare a vicenda.

L'ospite scrive due testi di valutazione, uno per l'host, uno per l'alloggio; una serie di punteggi in una scala da 1 a 5 su dimensioni come pulizia, comunicazione

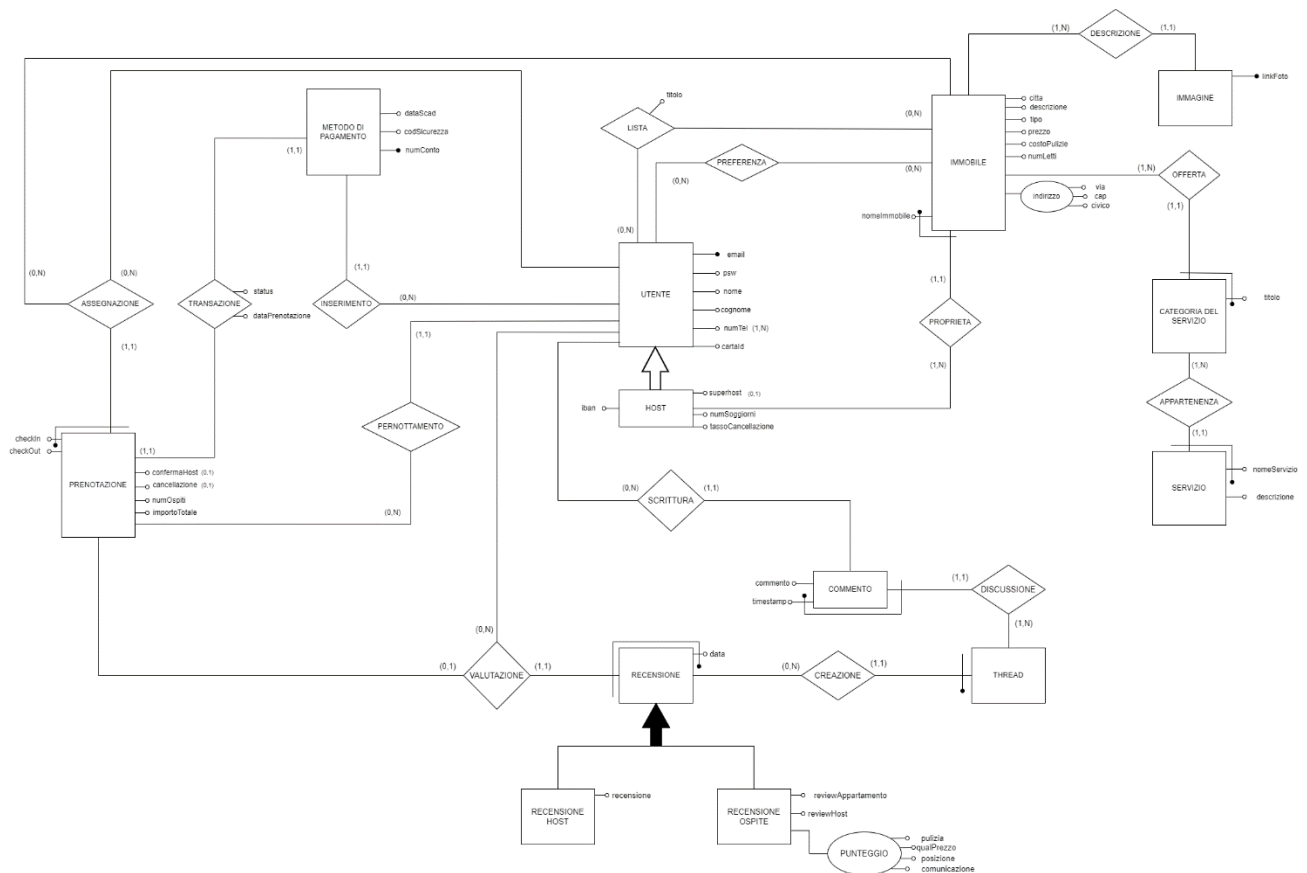
Frase relative alle prenotazioni

Gli utenti possono prenotare alloggi di qualsiasi tipo indicando l'intervallo di date per il soggiorno, il numero degli ospiti.

La prenotazione deve essere confermata o rifiutata dall'host.

La prenotazione può essere cancellata sia dall'ospite che dall'host entro le 24 ore antecedenti al check-in.

1.4 Schema E-R + business rules



Business rules

- Un host per diventare superhost deve soddisfare tre requisiti:
 - Deve aver completato almeno 10 soggiorni, per un totale di almeno 100 notti
 - Deve aver conservato un tasso di cancellazione dell'1% massimo
 - Deve aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà
- L'indirizzo degli alloggi diventa visibile solo quando la prenotazione è conservata, altrimenti è visibile solo il comune
- La prenotazione, che può essere cancellata sia dall'ospite che dall'host, ha un costo totale ed il pagamento viene eseguito solo se confermata
- Le recensioni diventano visibili quando sia host che ospite hanno fatto la recensione oppure se uno dei due non ha fatto la recensione l'altra diventa visibile dopo 7 giorni dalla fine del soggiorno
- Gli host e gli ospiti possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione

2 Progettazione logica

2.1 *Tavola dei volumi*

CONCETTO	TIPO	VOLUME
Utente	E	500.000
Host	E	50.000
Immobile	E	150.000
Immagine	E	450.000
Servizio	E	1.500.000
Categoria del servizio	E	600.000
Prenotazione	E	2.000.000
Recensione	E	3.000.000
Recensione host	E	1.500.000
Recensione ospite	E	1.500.000
Thread	E	2.500.000
Commento	E	5.000.000
Metodo di pagamento	E	1.000.000
Assegnazione	R	2.000.000
Transazione	R	1.500.000
Inserimento	R	1.000.000
Pernottamento	R	3.500.000
Scrittura	R	5.000.000
Valutazione	R	3.000.000
Creazione	R	2.500.000
Discussione	R	5.000.000
Preferenza	R	1.500.000
Lista	R	1.500.000
Descrizione	R	450.000
Offerta	R	750.000
Appartenenza	R	1.500.000
Proprietà	R	150.000

Partendo dall'ipotesi che la nostra piattaforma sia appena stata sviluppata abbiamo stimato un bacino di **utenza** pari a 500.000, di cui **host** 50.000.

Tenendo in considerazione che un host possieda una media di 3 **immobili** abbiamo considerato un volume di 150.000 sia per l'entità Immobile sia per la relazione Proprietà.

Per ogni immobile verranno caricate una media di 3 **immagini** descrittive arrivando ad un volume di 450.000.

I possibili **servizi** offerti sono di base circa 45, di questi 45 servizi per ogni immobile l'host ne sceglie in media 10, arrivando così ad un volume stimato di 1.500.000.

Per quanto riguarda le **categorie del servizio**, si presume una media di 4 categorie per immobile, per un volume totale di 600.000.

Le **prenotazioni** effettuate sono mediamente 4 per utente, quindi 2.000.000 di cui solo 1.500.000 a buon fine, di conseguenza avremo un numero di **recensioni host e ospite** pari al numero di prenotazioni andate a buon fine.

Partendo dal fatto che non da tutte le recensioni partirà un **thread** con relativi **commenti**, il volume dei thread sarà pari a 500.000 in meno delle recensioni mentre i commenti saranno 5.000.000 immaginando due commenti in media sotto i thread creati.

2.2 Tavola delle operazioni

OPERAZIONE	TIPO	FREQUENZA
Controllo delle condizioni per la qualifica di superhost e aggiornamento dello status degli host	B	1/giorno
Aggiornamento del tasso di cancellazione di ciascun host	B	1/settimana
Aggiornamento del numero di soggiorni di ciascun host	B	1/settimana
Calcolo della classifica degli alloggi più graditi	B	1/mese
Iscrizione di un utente	I	1.000/settimana
Prenotazione immobile	I	1.000/settimana
Aggiunta di un immobile	I	50/mese
Aggiunta di una recensione	I	1.200/settimana
Inserimento metodo di pagamento da parte di un utente	I	1.000/settimana
Inserimento di un commento di risposta a una recensione	I	1.200/settimana

Abbiamo stilato le operazioni interattive e di batch più rilevanti effettuate dal nostro servizio che permette di prenotare e affittare un immobile.

Considerando la piattaforma nascente si prospetta un incremento di iscrizioni pari a 1.000 utenti alla settimana.

La stima di prenotazioni si aggira intorno ai 1.000 immobili settimanali, con un'aggiunta di 50 immobili mensili.

Considerando che le recensioni possono essere scritte sia dagli host che dagli ospiti stimiamo che su un totale di 1.000 prenotazioni settimanali (2.000 utenti coinvolti) verranno caricate 1.200 recensioni.

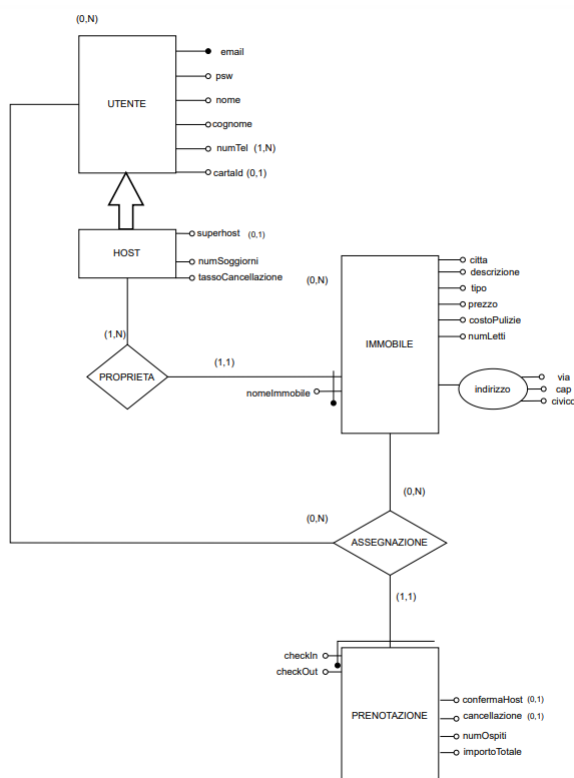
2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

Gli attributi **tassoCancellazione** e **numSoggiorni** dell'entità HOST sono derivabili dal conteggio dell'attributo **cancellazione** presente nella tabella PRENOTAZIONE.

L'attributo **importoTotale** dell'entità PRENOTAZIONE è derivabile calcolando il prezzo dell'immobile per la differenza di giorni tra la data di check-out e di check-in, sommato al costo delle pulizie.

Prendiamo in considerazione la ridondanza per il conteggio del tasso di cancellazione nell'entità HOST.



CONCETTO	TIPO	VOLUME
Utente	E	500.000
Host	E	50.000
Immobile	E	150.000
Prenotazione	E	2.000.000
Proprietà	R	150.000
Assegnazione	R	2.000.000

OPERAZIONE 1: Aggiornamento del tasso di cancellazione di ciascun host (1/settimana).

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PRENOTAZIONE	E	1	L
HOST	E	1	S

CON RIDONDANZA

1 accesso in lettura e
1 accesso in scrittura

CONCETTO	COSTRUTTO	ACCESSI	TIPO
PRENOTAZIONE	E	1	L

SENZA RIDONDANZA

1 accesso in lettura

➤ Con ridondanza

- 1 accesso in scrittura vale come 2 accessi e 1 accesso in lettura, per cui in totale 3 accessi.

➤ Senza ridondanza

- 1 accesso in lettura.

Ipotizzando un volume di host pari a 50.000 e un utilizzo di 4 byte per memorizzare il tasso di cancellazione, lo spazio totale necessario sarà $4 * 50.000$ ovvero 200 Kbyte.

Con ridondanza	Senza ridondanza
3 accessi	1 accesso
200 Kbyte di spazio aggiuntivo	0 byte di spazio aggiuntivo

Avendo un inferiore numero di accessi e un maggior numero di spazio libero è conveniente eliminare la ridondanza.

2.3.2 Eliminazione delle generalizzazioni

➤ **Specializzazione Host da Utente**

La strategia utilizzata è stata l'accorpamento della specializzazione (host) nel genitore (utente) in quanto gli accessi a utente e host sono contestuali. Accorpendo le due entità abbiamo modificato la cardinalità dell'associazione PROPRIETÀ da (1, N) a (0, N) in quanto non tutti gli utenti possiedono un immobile.

➤ **Generalizzazione Recensione**

Essendo una generalizzazione totale abbiamo optato per l'accorpamento del genitore (Recensione) nella generalizzazione delle figlie (Recensione host e Recensione ospite) in quanto gli accessi a quest'ultime sono distinti.

2.3.3 Eventuale partizionamento/accorpamento di entità e associazioni

Non è stato necessario partizionare o accorpare entità nello schema E-R ma abbiamo eliminato gli attributi multi-valore introducendo nuove relazioni.

L'attributo numTel di utente con cardinalità (1, N) è stato modificato aggiungendo una nuova relazione RECAPITO collegata a una nuova entità TELEFONO con PK numTel (numero telefonico).

Per quanto riguarda l'attributo composto punteggio presente nell'entità RECENSIONE OSPITE è stato partizionato aggiungendo i suoi attributi all'entità coinvolta.

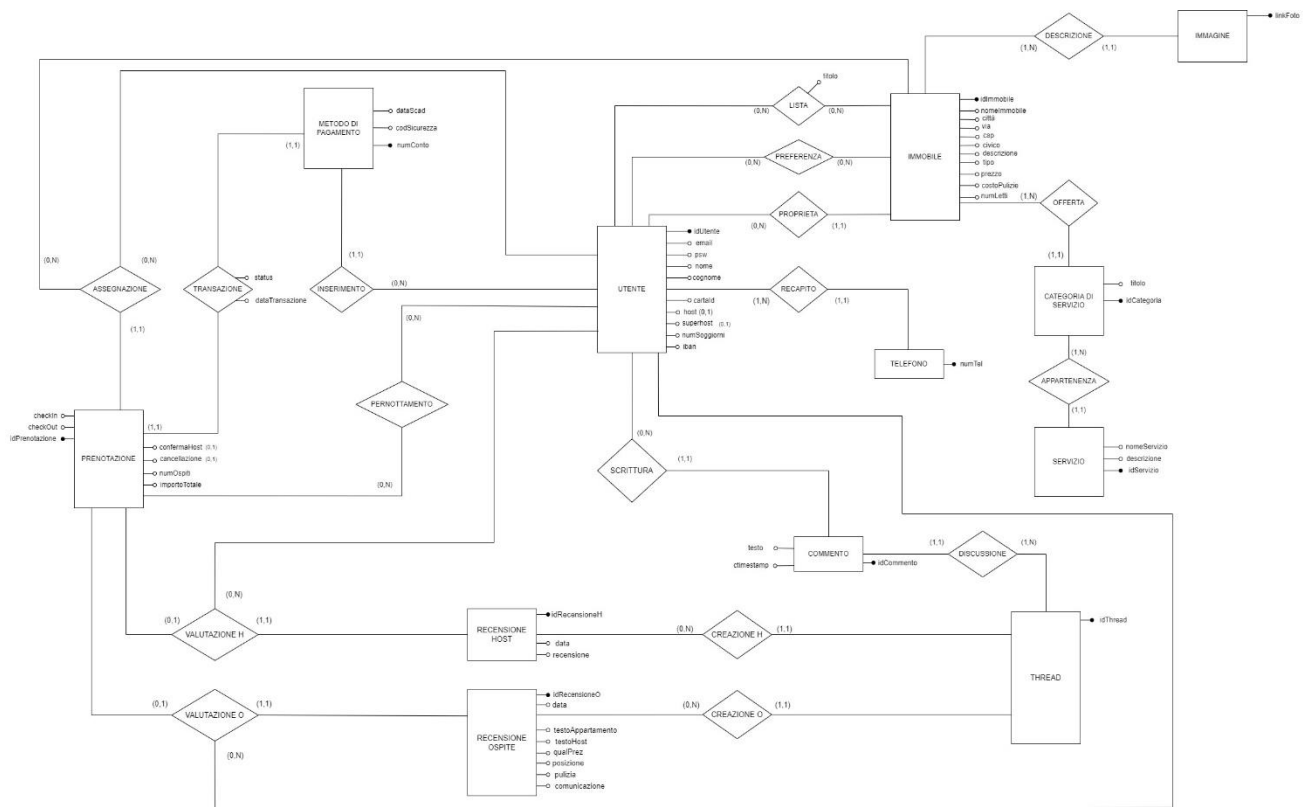
2.3.4 Eventuale scelta degli identificatori principali

La PK di utente è stata variata dall'email ad un codice in quanto essendo questa entità utilizzata nella quasi totalità delle operazioni il confronto di una stringa risulta più costoso di quello di un numero intero.

Per la stessa ragione abbiamo introdotto un Id per le entità immobile, prenotazione, recensione host e recensione ospite.

Ulteriori id sono stati introdotti per le entità thread, commento, categoria di servizio e servizio in quanto nella gamma dei loro attributi non era possibile effettuare una scelta semplice ed efficace.

2.4 Schema ER ristrutturato + business rules



Business rules

- Un host per diventare superhost deve soddisfare tre requisiti:
 - Deve aver completato almeno 10 soggiorni, per un totale di almeno 100 notti
 - Deve aver conservato un tasso di cancellazione dell'1% massimo
 - Deve aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà
- L'indirizzo degli alloggi diventa visibile solo quando la prenotazione è conservata, altrimenti è visibile solo il comune
- La prenotazione, che può essere cancellata sia dall'ospite che dall'host, ha un costo totale ed il pagamento viene eseguito solo se confermata
- Le recensioni diventano visibili quando sia host che ospite hanno fatto la recensione oppure se uno dei due non ha fatto la recensione l'altra diventa visibile dopo 7 giorni dalla fine del soggiorno
- Gli host e gli ospiti possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione
- Un utente non può prenotare un immobile di sua proprietà
- Non è possibile inserire nella prenotazione un numero di ospiti superiore al numero di letti disponibili nell'immobile

- Il nome indicato nella registrazione di un immobile deve essere univoco
- Ogni utente registrato deve avere almeno un recapito telefonico associato

2.5 Schema relazionale

Utente (idUtente, email, psw, nome, cognome, cartald, host, superhost, numSoggiorni)

MetodoDiPagamento (numConto, dataScad, codSicurezza, idUtente)

Immagine (linkFoto, idImmobile)

Telefono (numTel, idUtente)

Immobile (idImmobile, nomeImmobile, citta, via, cap, civico, descrizione, tipo, prezzo, costoPulizie, numLetti, idUtente)

CategoriaDiServizio (idCategoria, titolo)

Servizio (idServizio, descrizione, nomeServizio, idCategoria)

Offerta (idImmobile, idCategoria)

Prenotazione (idPrenotazione, checkIn, checkOut, confermaHost, cancellazione, numOspiti, importoTotale, status, dataTransazione, idUtente, idImmobile)

RecensioneHost (idRecensioneH, data, recensione, idPrenotazione, idUtente)

RecensioneOspite (idRecensioneO, data, reviewAppartamento, reviewHost, qualitaPrezzo, posizione, pulizia, comunicazione, idPrenotazione, idUtente)

Thread (idThread, idRecensioneH, idRecensioneO)

Commento (idCommento, testo, timestamp, idThread, idUtente)

Pernottamento (idPrenotazione, idUtente)

Lista (idImmobile, idUtente, titolo)

Preferenza (idImmobile, idUtente)

Assegnazione (idImmobile, idUtente)

3 Implementazione

3.1 Creazione del database

```
CREATE TABLE utente (  
    idUtente SERIAL primary key,  
    email varchar (319) NOT NULL,  
    psw varchar (8) NOT NULL,  
    nome varchar (20) NOT NULL,  
    cognome varchar (20) NOT NULL,  
    iban varchar(27);  
    cartaID varchar (255),  
    host boolean default false,  
    superhost boolean default false,  
    numSoggiorni integer default 0  
);
```



```
CREATE TABLE metododipagamento (  
    numConto varchar (16) primary key,  
    dataScad char (5) NOT NULL,  
    codSicurezza char (3),  
    idUtente SERIAL references utente(idUtente)  
);
```



```
CREATE TABLE telefono (  
    numTel varchar (15) primary key,  
    idUtente SERIAL references utente(idUtente)  
);
```



```
CREATE TABLE immobile (  
    idImmobile SERIAL primary key,  
    nomeImmobile varchar (255) unique NOT NULL,  
    citta varchar (255) NOT NULL,  
    via varchar (255) NOT NULL,  
    civico varchar (255) NOT NULL,  
    cap char (5) NOT NULL,  
    descrizione varchar (500),  
    tipo varchar (255) NOT NULL,  
    prezzo float NOT NULL,  
    costoPulizie float NOT NULL,  
    numLetti integer NOT NULL,  
    idUtente SERIAL references utente(idUtente)  
);
```

```
CREATE TABLE Immagini (  
    linkFoto varchar (255) primary key,  
    idImmobile SERIAL references immobile(idImmobile)  
);
```

```
CREATE TABLE servizio (  
    idServizio SERIAL primary key,  
    descrizione varchar (500),  
    nomeServizio varchar (255) NOT NULL,  
    idCategoria SERIAL references categoriadiservizio(idCategoria)  
);
```

```
CREATE TABLE categoriaDiServizio (  
    idCategoria SERIAL primary key,  
    titolo varchar (255) NOT NULL,  
    idImmobile SERIAL references immobile(idImmobile)  
);
```

```
create domain StatoPagamento  
as varchar (9) default 'in attesa'  
check (value = 'in attesa' OR value = 'pendente' OR value = 'eseguito');
```

```
CREATE TABLE prenotazione (  
    idPrenotazione SERIAL primary key,  
    checkIn varchar (16) NOT NULL,  
    checkOut varchar (16) NOT NULL,  
    confermaHost boolean default false,  
    cancellazione boolean default false,  
    numOspiti integer default 0,  
    importoTotale float NOT NULL,  
    status StatoPagamento,  
    dataTransazione date,  
    idUtente SERIAL references utente(idUtente),  
    idImmobile SERIAL references immobile(idImmobile)  
);
```

```
CREATE TABLE recensioneHost (  
    idRecensioneH SERIAL primary key,  
    data timestamp default current_timestamp NOT NULL,  
    recensione varchar (255) NOT NULL,  
    idUtente SERIAL references utente(idUtente),  
    idPrenotazione SERIAL references prenotazione(idPrenotazione) unique  
);
```

create domain Voto

as smallint default NULL

check (value >=0 AND value <= 5);

CREATE TABLE recensioneOspite (

idRecensioneO SERIAL primary key,

data timestamp default current_timestamp,

reviewAppartamento varchar (500),

reviewHost varchar (500),

qualitaPrezzo Voto NOT NULL,

posizione Voto NOT NULL,

pulizia Voto NOT NULL,

comunicazione Voto NOT NULL,

idUtente SERIAL references utente(idUtente),

idPrenotazione SERIAL references prenotazione(idPrenotazione) unique

);

CREATE TABLE thread (

idThread SERIAL primary key,

idRecensioneH SERIAL references recensionehost(idRecensioneH),

idRecensioneO SERIAL references recensioneospite(idRecensioneO)

);

CREATE TABLE commento (

idCommento SERIAL primary key,

testo varchar (255) NOT NULL,

ctimestamp timestamp default current_timestamp,

idUtente SERIAL references utente (idUtente),

idThread SERIAL references thread (idThread)

);

CREATE TABLE pernottamento (

idPrenotazione SERIAL references prenotazione(idPrenotazione),

idUtente SERIAL references utente(idUtente),

primary key (idPrenotazione, idUtente)

);

CREATE TABLE lista (

idImmobile SERIAL references immobile(idImmobile),

idUtente SERIAL references utente(idUtente),

titolo varchar (255) NOT NULL,

primary key (idImmobile, idUtente)

);

```
CREATE TABLE preferenza (  
idImmobile SERIAL references immobile(idImmobile),  
idUtente SERIAL references utente(idUtente),  
primary key (idImmobile, idUtente)  
);
```

3.2 DML di popolamento di tutte le tabelle del database

Iscrizione degli utenti, con i relativi numeri di telefono e metodo di pagamento.

```
insert into utente (email, psw, nome, cognome, host, iban) values  
('robertorossi@gmail.com',12345678, 'Roberto', 'Rossi', true, 'ITXXXXPROVA');
```

```
insert into utente (email, psw, nome, cognome) values ('annaverdi@gmail.com', 33345678, 'Anna',  
'Verdi');
```

```
insert into utente (email, psw, nome, cognome) values  
('sabinasole@gmail.com',14445678,'Sabrina','Sole');
```

```
insert into utente (email, psw, nome, cognome) values  
('mariofabbri@gmail.com',12355678,'Mario','Fabbri');
```

```
insert into utente (email, psw, nome, cognome) values  
('valeristella@gmail.com',12346678,'Valeria','Stella');
```

```
insert into telefono (numtel, idUtente) values ('3333047684', 1);
```

```
insert into telefono (numtel, idUtente) values ('3933047684', 2);
```

```
insert into telefono (numtel, idUtente) values ('3383047774', 3);
```

```
insert into telefono (numtel, idUtente) values ('3933047564', 4);
```

```
insert into telefono (numtel, idUtente) values ('3939847684', 5);
```

```
insert into metonodipagamento (numconto, datascad, codsicurezza, idUtente) values  
(1233434367869987, 05/23, 334,1);
```

```
insert into metododipagamento (numconto, datascad, codsicurezza, idUtente) values  
(1233435557869987, 06/23, 334,2);
```

```
insert into metododipagamento values (3323435557869987, 06/24, 334,3);
```

```
insert into metododipagamento (numconto, datascad, codsicurezza, idUtente) values  
(1233436457869987, 10/23, 334,4);
```

```
insert into metododipagamento (numconto, datascad, codsicurezza, idUtente) values  
(1233436457866657, 04/23, 534,5);
```


insert into immobile (nomeimmobile, citta, via, civico, cap, descrizione, tipo, prezzo, costopulizie, numletti, idUtente) **values** ('casa Margherita', 'Firenze', 'via Masaccio',3, 10140, 'ampio salotto e vista panoramica', 'bilocale', 150, 40, 3,2);

insert into immagine (linkFoto, idImmobile) **values** ('C:\Users\rache\OneDrive\Immagini\Catture di schermata/adjajdsdf.png',1)

insert into immobile (nomeimmobile, citta, via, civico, cap, descrizione, tipo, prezzo, costopulizie, numletti, idUtente) **values** ('villa Regina', 'Torino', 'via Garibaldi',10, 10165, 'villa pittoresca nel centro storico', 'villa', 300, 50, 4,2);

insert into immagine (linkFoto, idImmobile) **values** ('C:\Users\rache\OneDrive\Immagini\salotto',2);

insert into immobile (nomeimmobile, citta, via, civico, cap, descrizione, tipo, prezzo, costopulizie, numletti, idUtente) **values** ('villa Aurora', 'Torino', 'via Roma',72, 10185, 'situata nel centro storico', 'villa', 350, 50, 4,2);

insert into immagine (linkFoto, idImmobile) **values** ('C:\Users\rache\OneDrive\Immagini\foto.png',3);

insert into immobile (nomeimmobile, citta, via, civico, cap, descrizione, tipo, prezzo, costopulizie, numletti, idUtente) **values** ('Appartamento studente', 'Torino', 'via Pozzo Strada',7, 10135, 'vicino alla metropolitana', 'monolocale', 100, 20, 1,1);

insert into immagine (linkFoto, idImmobile) **values** ('C:\Users\rache\OneDrive\Immagini\descrizioneP123391.png',4)

insert into categoriadiservizio (titolo, idImmobile) **values** ('giardino',2);

insert into categoriadiservizio (titolo, idImmobile) **values** ('stoviglie e cucina',3);

insert into servizio (descrizione, nomeservizio, idCategoria) **values** ('6 coperti con lavaggio rapido', 'lavastoviglie',1);

insert into servizio (descrizione, nomeservizio, idCategoria) **values** ('idromassaggio', 'piscina',2);

insert into servizio (descrizione, nomeservizio, idCategoria) **values** ('peperoncino e aglio', 'erbe aromatiche',1);

insert into servizio (descrizione, nomeservizio, idCategoria) **values** ('pentola a pressione', 'stoviglie',1);

insert into prenotazione (checkin, checkout, numospiti, status, importototale, idImmobile, idUtente) **values** ('15/12/2022-10.30', '17/12/2022-19.00', 2, 'eseguito', 340, 1, 1);

insert into pernottamento (idPrenotazione, idUtente) **values** (12,4);

insert into recensionehost (recensione, idPrenotazione, idUtente) **values** ('un ospite perfetto',8,2);

insert into recensioneospite (reviewappartamento, reviewhost, qualitaprezzo, posizione, pulizia, comunicazione, idPrenotazione, idUtente) **values** ('alloggio favoloso', 'accoglienza impeccabile', 4, 4, 5, 5, 8, 1);

insert into thread (idRecensioneH, idRecensioneO) **values** (6,5);

insert into commento (testo, idThread, idUtente) **values** ('grazie per la calorosa accoglienza',4,1);

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) **values** ('16/06/2022-11.30','19/06/2022-19.00',2,1250,2,1);

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) **values** ('17/07/2022-11.30','18/07/2022-19.00',1, 400,3,1);

update prenotazione **set** status= 'pendente' **where** (idUtente=1 AND checkin='17/07/2022-11.30');

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) **values** ('07/07/2024-11.30','08/07/2024-21.00',1, 400,3,1);

update prenotazione **set** cancellazione=true **where** (idUtente=1 AND checkin='07/07/2024-11.30');

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) **values** ('18/08/2022-12.30','19/08/2022-18.00',0, 120,4,2);

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) **values** ('16/10/2022-11.30','17/10/2022-19.00',2, 450,2,3);

insert into recensioneospite (reviewappartamento, reviewhost, qualitaprezzo, posizione, pulizia, comunicazione, idPrenotazione, idUtente) **values** ('alloggio favoloso', 'accoglienza impeccabile', 4, 4, 5, 5, 12, 3);

insert into recensionehost (recensione, idPrenotazione, idUtente) **values** ('ospite in perfetto orario', 12, 2);

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) values ('20/11/2022-11.30', '21/11/2022-20.00', 0, 120, 4, 4);

insert into prenotazione (checkin, checkout, numospiti, importototale, idImmobile, idUtente) values ('01/06/2022-11.30', '02/06/2022-19.00', 0, 450, 2, 5);

insert into lista (idImmobile, idUtente, titolo) values (1,1, 'viaggio a Parigi');

insert into preferenza (idImmobile, idUtente) values (1,1);

insert into preferenza (idImmobile, idUtente) values (4,3);

insert into preferenza (idImmobile, idUtente) values (4,4);

insert into preferenza (idImmobile, idUtente) values (2,5);

3.3 operazioni di prova

Un utente diventa host.

update utente set host = true and iban = 'MOZZAR13321ELLA' where idUtente = 2;

Aggiornamento pagamento eseguito a seguito di una prenotazione.

update prenotazione set status= 'eseguito' where (idUtente=1 AND checkin='15/12/2022-10.30');

update prenotazione set status= 'eseguito' where (idUtente=1 AND checkin='16/06/2022-11.30');

update prenotazione set status= 'eseguito' where (idUtente=5 AND checkin='01/06/2022-11.30');

update prenotazione set status= 'eseguito' where (idUtente = 4 AND checkin = '20/11/2022-11.30');

update prenotazione set status = 'eseguito' where (idUtente=2 AND checkin='18/08/2022-12.30');

update prenotazione set status= 'eseguito' where (idUtente = 3 AND checkin = '16/10/2022-11.30');

Aggiornamento numSoggiorni a seguito di una prenotazione andata a buon fine.

update utente **set** numSoggiorni = numSoggiorni + 1 **where** idUtente=1;

update utente **set** numSoggiorni = numSoggiorni + 1 **where** idUtente=2;

Eliminazione di una recensione avente un thread aperto.

delete from recensionhost **where** idPrenotazione = 8;

Questa query restituisce **giustamente** un errore SQL state 2350 in quanto l'eliminazione di questa recensione viola il vincolo di chiave esterna thread_idrecensione_fkey sulla tabella Thread. Essendo che un utente che ha scritto la recensione può decidere di eliminarla bisogna eliminare e ricreare il vincolo aggiungendo ON DELETE CASCADE.

Eliminazione di un numero di telefono.

delete from telefono **where** numtel='3333047684';