

Laboratorio Basi di Dati 2021/22 Progetto di piattaforma di home booking

Lorenzo Lombardi – Emanuele Parusso – Federico Paschetta

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).¹

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 (vedere Fig. 1).

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.

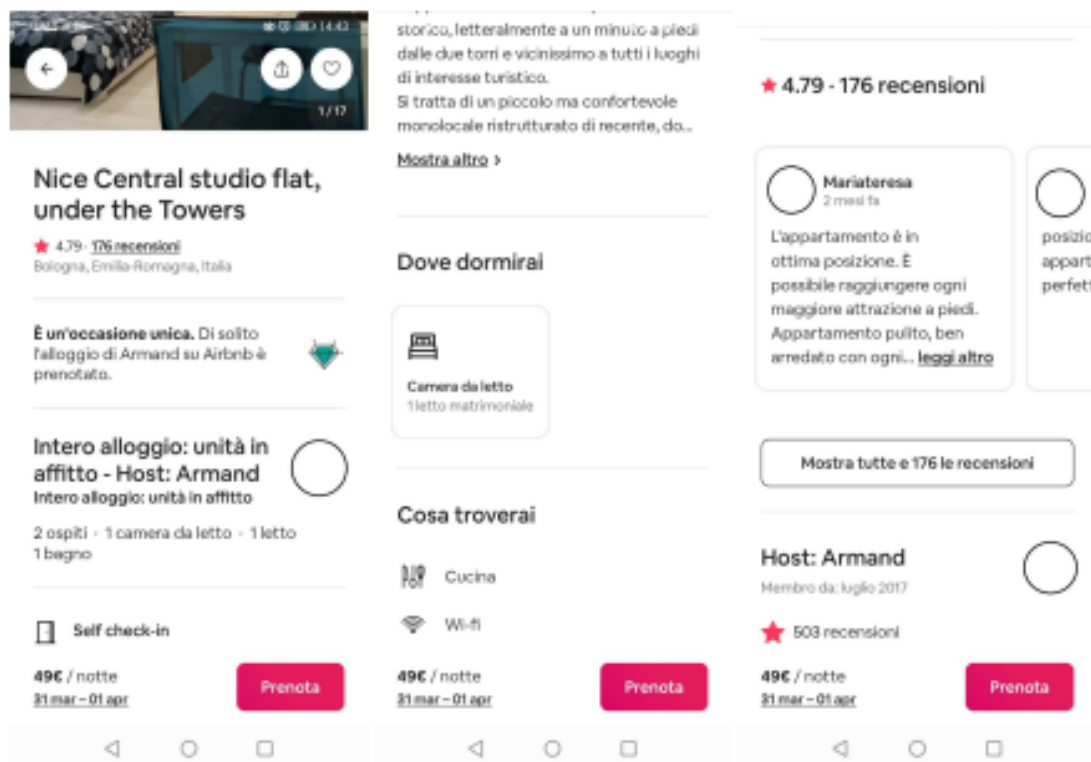


Fig. 1: Dettagli di un appartamento.

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

NOME	SIGNIFICATO	SINONIMI	RELAZIONI COINVOLTE
Utente	Persona fisica con account sulla piattaforma	User, nominativo	Seleziona
Utente verificato	Utente che usufruisce del servizio dal lato cliente e ha fornito una foto del documento d'identità		
Utente attivo	Utente verificato che fornisce un metodo di pagamento, abilitato alla prenotazione	Ospite, Utente abilitato alla prenotazione	Partecipazione al thread, U-P, Valuta ospite, Valuta alloggio, Valuta host, Selezione
Host	Utente attivo proprietario di 1 o più alloggi	Affittuario	Possiede, Partecipazione al thread (Lato host), host recensisce utente
Superhost	Host che soddisfa alcuni speciali requisiti e ha un badge sul profilo	Host verificato	
Alloggio	Spazio di proprietà dell'host in affitto all'ospite	Abitazione, Casa, struttura	Possiede, P-A, Valuta alloggio, Composta
Intero Appartamento	Intera unità abitativa dedicata ad un unico ospite	Intera casa, tutta la casa, chalet	
Stanza privata	Camera privata affittabile dall'ospite con spazi comuni	Camera singola in casa, camera d'hotel	
Stanza condivisa	Camera condivisa con altri ospiti	Ostello	
Prenotazione	Richiesta di soggiorno presso una determinata struttura in una determinata finestra tempo	Soggiorno, vacanza, permanenza, numero di notti	U-P, P-A, Utente recensisce host, Host recensisce utente
In sospeso	Stato di prenotazione: la prenotazione è sospesa		

Rifiutata	Stato di prenotazione: la prenotazione è rifiutata dall'host		
Annullata	Stato di prenotazione: la prenotazione è annullata dall'utente		
Confermata	Stato di prenotazione: la prenotazione è confermata dall'host		
Cancellata	Stato di prenotazione: la prenotazione è stata cancellata dopo essere stata confermata		
Terminata	Stato di prenotazione: la prenotazione è terminata dopo che il soggiorno si è concluso		
Recensione	Insieme di tutte le recensioni, sia da parte del cliente che da parte dell'host	Punteggio, Review, Valutazione	Afferenza(Thread)
Recensione scritta dal cliente	Valutazione da parte del cliente della qualità della struttura e del soggiorno e dell'host	Valutazione da parte del cliente, punteggio, review	Partecipazione al thread (Lato cliente), Utente recensisce host
Recensione della casa	Valutazione da parte del cliente riguardo all'abitazione nella quale ha soggiornato	Valutazione della casa, review	
Recensione dell'host	Valutazione da parte del cliente riguardo all'host proprietario dell'abitazione nella quale ha soggiornato	Valutazione dell'host	
Recensione scritta dall'host	Valutazione da parte dell'host del rispetto delle politiche del soggiorno e del comportamento dell'ospite	Valutazione dell'ospite, rating dell'ospite, punteggio, review	Host recensisce utente
Lista di preferiti	Insieme di alloggi, tutti tra i preferiti, tra cui scegliere per un futuro viaggi in una specifica destinazione		Selezione, Composta

Thread	Spazio dove l'utente e l'host coinvolti possono discutere riguardo a una recensione	Forum di discussione	Partecipazione al thread (Lato Utente), Partecipazione al thread (Lato Host), Afferenza
--------	---	----------------------	---

1.3 Requisiti rivisti

La piattaforma di home booking si basa sullo scambio tra due tipi di utenti: gli ospiti e gli host. Gli utenti possono registrarsi, verificare il loro profilo fornendo la carta d'identità e abilitarsi alla prenotazione inserendo un metodo di pagamento. Gli ospiti possono cercare un alloggio per un loro soggiorno, effettuare prenotazioni, aggiungere case tra i preferiti (e creare diverse liste di preferiti), lasciare recensioni della loro permanenza (una sull'host e una sull'alloggio, in particolare un voto da 1 a 5 su vari parametri). Gli host sono ospiti che, a loro volta, possono ospitare altri utenti del servizio in uno o più alloggi di loro proprietà, confermare o rifiutare le prenotazioni degli ospiti, commentare le recensioni degli ospiti (creando una discussione) o gli ospiti stessi. Una categoria speciale di host sono i superhost, i quali soddisfano alcuni requisiti, presenti nelle business rules, in cambio dei quali ottengono un badge sul loro profilo, che li rende riconoscibili.

Gli alloggi presenti sulla piattaforma possono essere di vario tipo: interi appartamenti, stanze private, stanze condivise e hanno diversi servizi ciascuno.

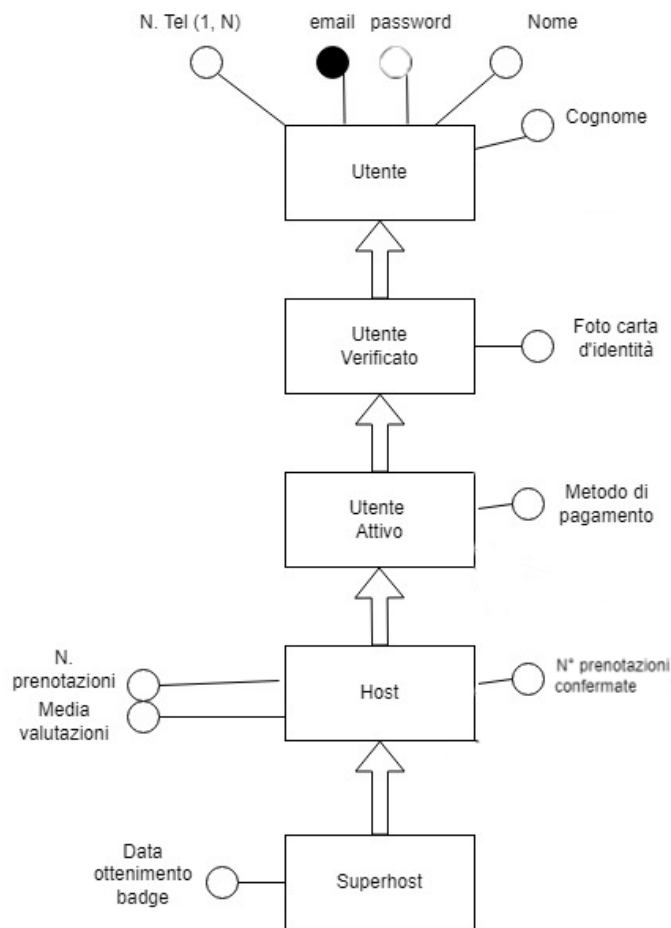
1.4. Schema E-R + business rules.

Business rules:

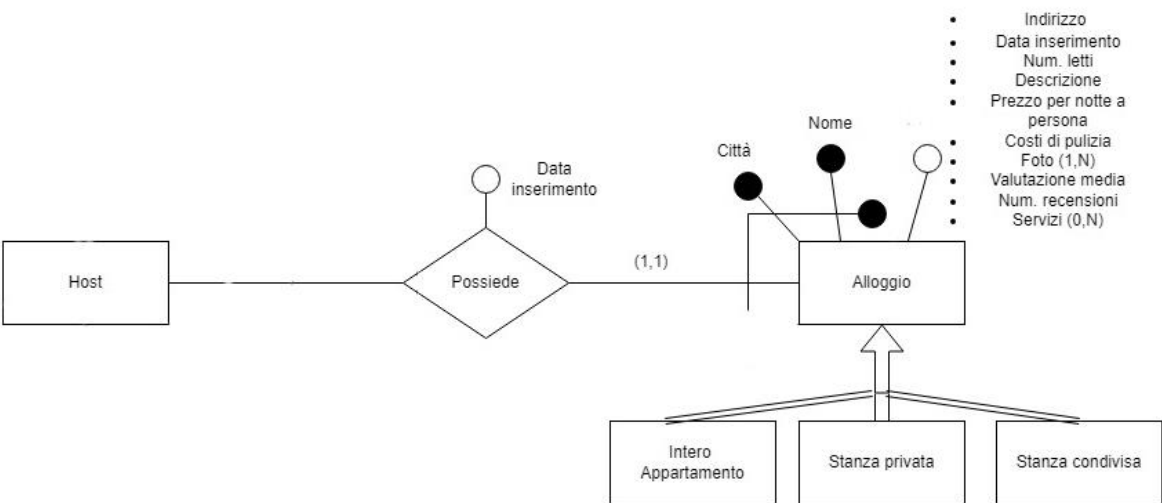
- Host è un utente verificato che ha aggiunto almeno un alloggio potenzialmente affittabile
- Superhost è un host dotato di badge per:
 - aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.
 - aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.
 - aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.
- Utente attivo potrà vedere l'indirizzo preciso dell'alloggio solo a prenotazione completata
- Testo della recensione visibile quando entrambi (utente e host) hanno fatto recensione, o a 30 giorni dalla fine del soggiorno
- se una prenotazione non viene confermata dall'host entro 24h questa viene rifiutata
- un alloggio non può essere collegato a più prenotazioni nelle stesse date

Schema E-R:

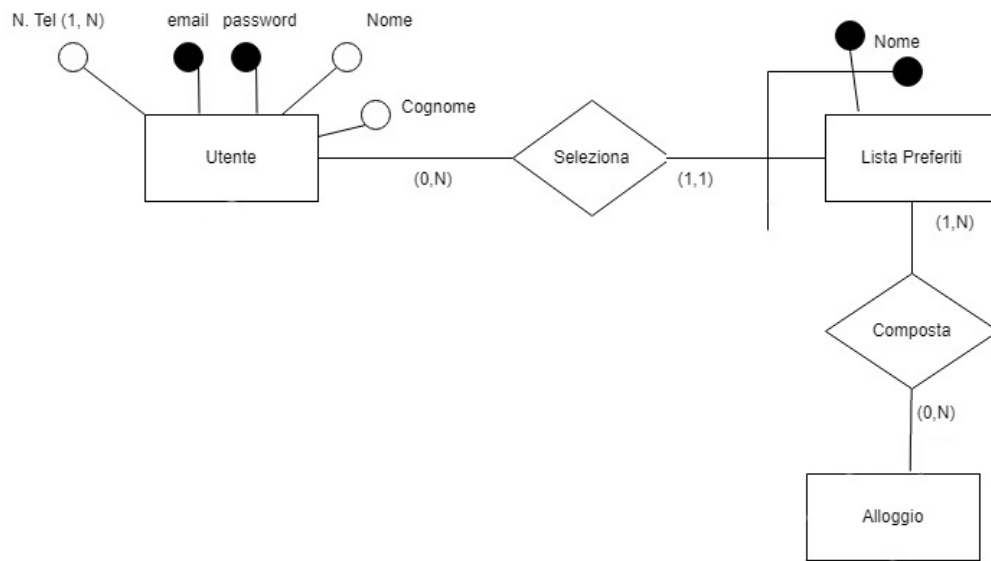
Utente:



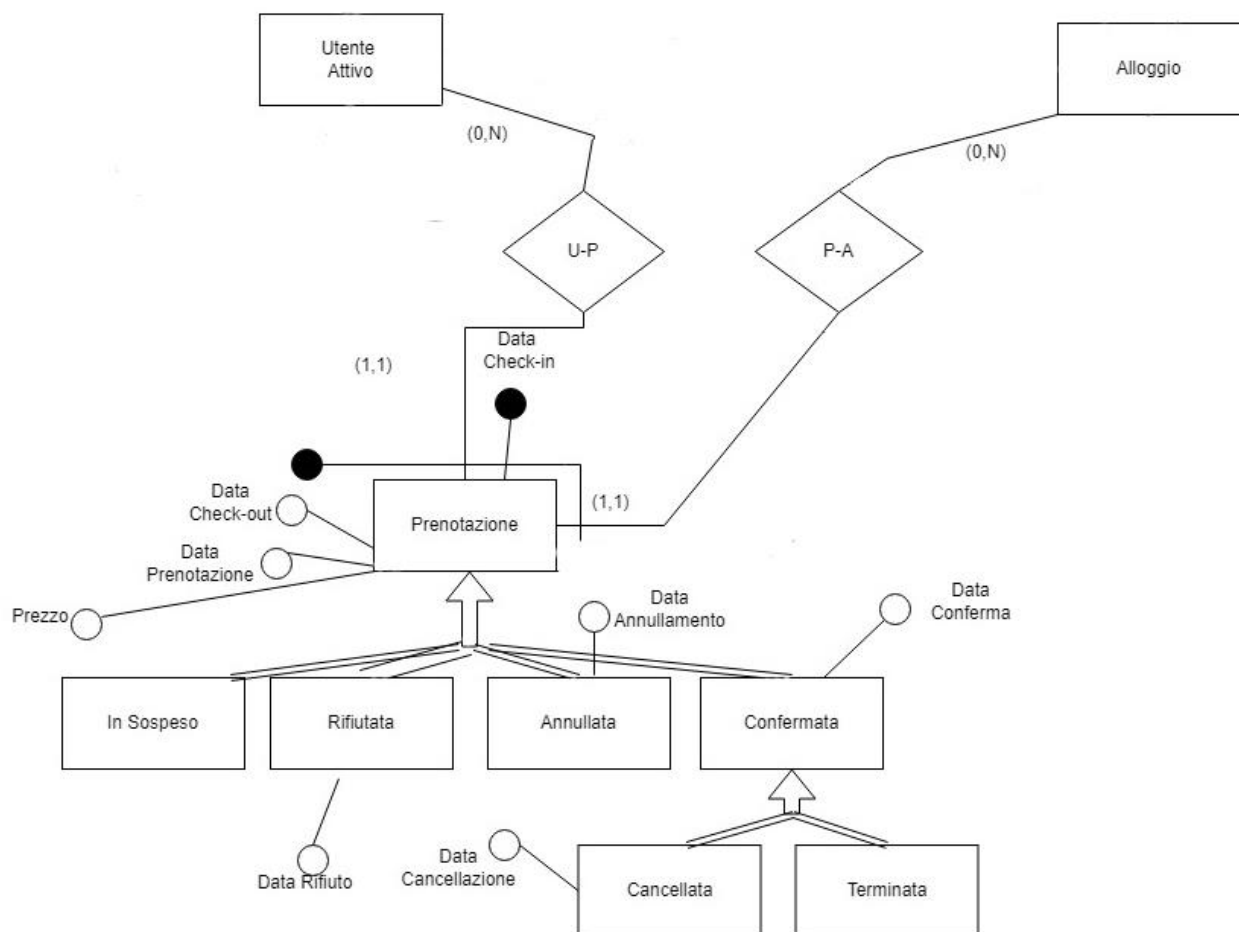
Alloggio:



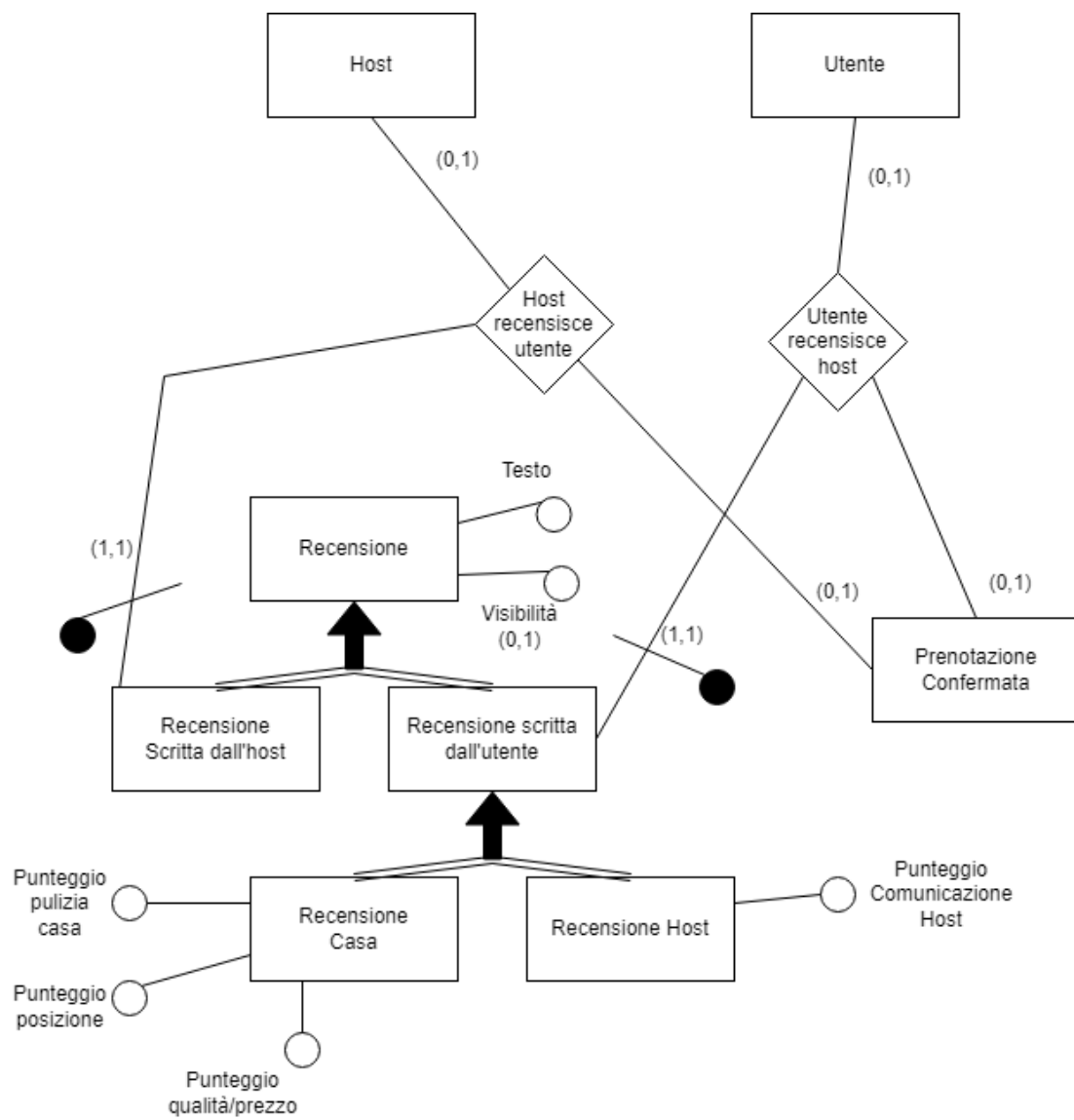
Lista preferiti:



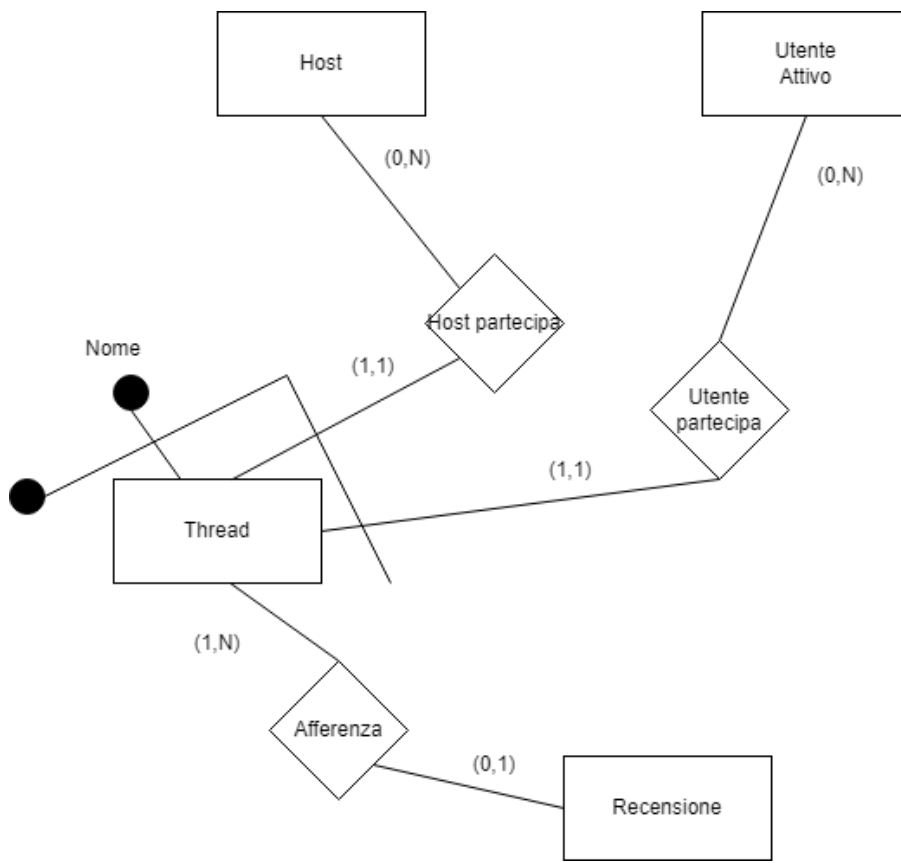
Prenotazione:



Recensione:



Thread:



2. Progettazione logica

2.1 Tavola dei volumi

Stimando che la piattaforma di home booking sia in funzione da poco tempo, essendo liberamente ispirata al leader di mercato Airbnb, per quanto riguarda il volume iniziale di utenti, abbiamo scelto di rappresentare l'1% del numero di utenti correntemente registrati presso Airbnb.

Questi sono i volumi annuali previsti dalla piattaforma:

CONCETTO	TIPO	VOLUME	MOTIVAZIONE
Utente	Entità	150.000	1% del volume di Airbnb
Utente Verificato	Entità	120.000	L'80% dei profili vuole effettuare una prenotazione
Utente Attivo	Entità	110.000	Il 6% degli utenti verifica il documento, ma non aggiunge il metodo di pagamento
Host	Entità	15.000	Il 10% degli utenti possiedono un alloggio
Superhost	Entità	3.000	Soltanto il 20% degli host rispetta gli stretti requisiti per ottenere il badge
Lista preferiti	Entità	150.000	2/3 degli utenti utilizza la funzione per le liste di preferiti, in media ogni utente che la usa salva 1.5 liste
Alloggio	Entità	20.000	Ogni host possiede in media 1.3 alloggi ciascuno
Intero appartamento	Entità	15.000	Il 75% circa degli alloggi è un appartamento (la maggior parte degli alloggi si trova in città)
Stanza privata	Entità	4.000	Il 20% degli alloggi sono stanze private (solitamente in hotel)
Stanza condivisa	Entità	1.000	Il 5% degli "alloggi" sono letti in stanze condivise (solitamente ostelli)
Prenotazione	Entità	313.100	Somma di tutte le specializzazioni
In sospeso	Entità	600	2 soggiorni all'anno * 110.000 utenti che possono effettuare prenotazioni / 365 giorni (contando che le prenotazioni devono essere accettate o rifiutate entro 24h)
Rifiutata	Entità	25.000	10% viene rifiutata

Annullata	Entità	25.000	10% viene annullata
Confermata	Entità	262.500	Poco meno dell'80% delle prenotazioni vengono confermate
Cancellata	Entità	12.500	5% viene cancellata
Terminata	Entità	250.000	In media poco più di 2 soggiorni all'anno per ogni utente attivo
Thread	Entità	50.000	Meno del 10% di recensioni aprono un thread
Recensione	Entità	625.000	Somma delle specializzazioni
Recensione scritta dall'utente	Entità	400.000	L'80% degli utenti scrivono la recensione post-soggiorno * 2 recensioni
Recensione scritta dall'host	Entità	225.000	Il 90% degli host scrivono la recensione dell'utente
Recensione casa	Entità	200.000	L'80% degli utenti scrivono la recensione post-soggiorno
Recensione host	Entità	200.000	L'80% degli utenti scrivono la recensione post-soggiorno
Seleziona	Relazione	150.000	= cardinalità lista preferiti
Composta	Relazione	750.000	In media 5 alloggi * lista preferiti
Possiede	Relazione	5.000	= cardinalità alloggi
U-P	Relazione	313.100	= cardinalità prenotazioni
P-A	Relazione	313.100	= cardinalità prenotazioni
Utente recensisce host	Relazione	400.000	= cardinalità Recensione scritta dall'utente
Host recensisce utente	Relazione	225.000	= cardinalità Recensione scritta dall'host
Afferenza	Relazione	50.000	= cardinalità Thread
Utente partecipa	Relazione	50.000	= cardinalità Thread
Host partecipa	Relazione	50.000	= cardinalità Thread

2.2 Tavola delle operazioni

N°	NOME	TIPO	TIPOLOGIA	FREQUENZA
1	Iscrizione nuovo utente	Inserimento	I	400/giorno
2	Registrazione documento	Inserimento	I	330/giorno

3	Registrazione metodo di pagamento	Inserimento	I	300/giorno
4	Inserimento alloggio	Inserimento	I	50/giorno
5	Aggiunta nuova lista preferiti	Inserimento	I	400/giorno
6	Aggiunta alloggio a lista preferiti	Inserimento	I	2.000/giorno
7	Aggiornamento dati utente	Modifica	I	400/giorno
8	Aggiornamento dati host	Modifica	I	20/giorno
9	Aggiornamento dati alloggio	Modifica	I	20/giorno
10	Effettuazione prenotazione da utente	Inserimento	I	850/giorno
11	Conferma prenotazione host	Modifica	I	680/giorno
12	Aggiornamento prenotazione (rifiuto host)	Modifica	I	60/giorno
12b	Aggiornamento automatico prenotazione (rifiuto host->no conferma entro 24h)	Modifica	B	20/giorno
13	Aggiornamento prenotazione (annullamento utente)	Modifica	I	80/giorno
14	Aggiornamento prenotazione (cancellazione host)	Modifica	I	40/giorno
15	Aggiornamento prenotazione (terminazione)	Modifica	B	640/giorno
16	Aggiunta recensione host-utente	Inserimento	I	600/giorno
17	Aggiunta recensione utente-host	Inserimento	I	550/giorno
18	Aggiunta recensione utente-alloggio	Inserimento	I	550/giorno
19	Creazione thread	Inserimento	I	130/giorno
20	Aggiunta commenti al thread	Inserimento	I	250/giorno
21	Aggiornamento tasso di cancellazione host	Modifica	B	1/settimana
22	Aggiornamento status host/superhost	Modifica	B	1/giorno
23	Calcolo classifica alloggi più graditi	Interrogazione	B	1/mese
24	Lista alloggi prenotabili in una data città	Interrogazione	I	8.000/giorno
25	Visualizzazione dati Alloggio	Interrogazione	I	20.000/giorno
26	Visualizzazione dati host	Interrogazione	I	10.000/giorno
27	Visualizzazione dati utente	Interrogazione	I	700/giorno
28	Lista recensioni alloggio	Interrogazione	I	20.000/giorno
29	Lista recensioni utente	Interrogazione	I	700/giorno
30	Lista recensioni host	Interrogazione	I	10.000/giorno
31	Lista prenotazioni effettuate (utente attivo)	Interrogazione	I	4.000/giorno

32	Lista prenotazioni ricevute (host)	Interrogazione	I	20.000/giorno
33	Aggiornamento lista host	Modifica	B	50/giorno
34	Cancellazione alloggio	Cancellazione	I	10/giorno

2.3.1 Analisi delle ridondanze

Ridondanza su attributo “numero prenotazioni” di host, ricavabile anche attraverso l’analisi degli alloggi posseduti e le prenotazioni a questi relative

Operazione: Visualizzazione dati host

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	e	1	L

Con ridondanza

Al giorno = $1 \cdot 10000 = 10000$

Ipotizzando 4 byte per la memorizzazione dell’attributo in questione vengono occupati circa 40 kb

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	E	1	L
Possiede	R	1	L
Alloggio	E	1	L
P-A	R	16	L
Prenotazione	E	16	L

Senza ridondanza

Al giorno = $35 \cdot 10000 = 350000$

Non è richiesto spazio aggiuntivo

Ridondanza su attributo “numero prenotazioni confermate” di host, ricavabile anche attraverso l’analisi degli alloggi posseduti e le prenotazioni a questi relative

Operazione: Visualizzazione dati host

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	e	1	L

Con ridondanza

Al giorno = $1 \cdot 10000 = 10000$

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	E	1	L
Possiede	R	1	L
Alloggio	E	1	L
P-A	R	16	L
Prenotazione	E	16	L

Prenotazione confermata	E	13	L
-------------------------	---	----	---

Ipotizzando 4 byte per la memorizzazione dell'attributo in questione vengono

occupati circa 40 kb

Senza ridondanza

Al giorno = $48 * 10000 = 480000$

Non è richiesto spazio aggiuntivo

Operazione: Tasso cancellazione host

Per questa operazione si sfruttano le tabelle degli accessi precedenti, coi relativi costi. A cui si aggiunge una operazione tra i due attributi analizzati.

Operazione: effettuazione prenotazione

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	E	16	L
P-A	R	16	L
Alloggio	E	1	L
Possiede	R	1	L
Host	E	1	L
Host	E	1	S

Con ridondanza

*$38 * 850 / \text{giorno} = 32000$*

Senza ridondanza non si necessita dell'aggiornamento dell'attributo, per cui non vi son costi

Operazione: Conferma prenotazione host

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione confermata	E	13	L
Prenotazione	E	16	L
P-A	R	16	L
Alloggio	E	1	L
Possiede	R	1	L
Host	E	1	L
Host	E	1	S

Con ridondanza

*$50 * 680 / \text{giorno} = 34000$*

Senza ridondanza non si necessita dell'aggiornamento dell'attributo, per cui non vi son costi

In seguito alle precedenti valutazioni, si ritiene che sia opportuno rimuovere le ridondanze

Ridondanza su attributo "prezzo" di prenotazione, ricavabile anche attraverso l'analisi del prezzo per notte per persona dell'alloggio*numero di letto dell'alloggio*(data check-out - data check-in)

Operazione: Effettuazione prenotazione

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	E	1	L

Con ridondanza

Al giorno = $1 \cdot 850 = 850$

Ipotizzando 4 byte per la memorizzazione dell'attributo in questione vengono occupati circa 3,4kb

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Alloggio	E	1	L
P-A	R	16	L
Prenotazione	E	16	S

Senza ridondanza

Al giorno = $49 \cdot 850 = 41650$

Le altre operazioni non impattano in termini di tempo e spazio sulla ridondanza.

Si ritiene debba essere mantenuta la ridondanza

Ridondanza su attributo "media valutazioni" di Host, ricavabile anche attraverso l'analisi della somma di Recensioni Host/Numero prenotazioni Host

Operazione: Visualizzazione dati host

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	E	1	L

Con ridondanza

Costi: $1 \cdot 10000 / \text{giorno} = 10000$

Ipotizzando 4 byte per la memorizzazione dell'attributo in questione vengono occupati circa 40 kb

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Host	E	1	L
Possiede	R	1	L
Alloggio	E	1	L
P-A	R	16	L
Prenotazione	E	16	L
Prenotazione confermata	E	13	L
Utente recensisce Host	R	11	L
Recensione scritta dall'utente	E	11	L

Recensione Host	E	11	L
-----------------	---	----	---

Senza ridondanza

Costi: $81 \times 10000 / \text{giorno} = 810000$

Non si necessita spazio aggiuntivo

Operazione: Aggiunta recensione utente-host

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Recensione Host	E	1	L
Recensione scritta dall'utente	E	1	L
Utente recensisce Host	R	1	L
Prenotazione confermata	E	1	L
Prenotazione	E	1	L
P-A	R	1	L
Alloggio	E	1	L
Possiede	R	1	L
Host	E	1	L
Host	E	1	R

Senza ridondanza non si necessita dell'aggiornamento dell'attributo, per cui non vi son costi

Costo: $11 \times 550 / \text{giorno} = 6050$

Consideriamo il costo della ricerca di Numero prenotazioni Host trascurabile in quanto uguale sia con che senza ridondanza (per valutazione di un costo simile vedere il conteggio del numero recensioni alloggio)

Si ritiene debba essere mantenuta la ridondanza

Ridondanza su attributo "numero recensioni" di Alloggio, ricavabile anche attraverso l'analisi dell'indicativo di alloggio in Recensione casa

Operazione: Visualizzazione dati Alloggio

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Alloggio	E	1	L

Con ridondanza

Costo = $1 \times 20000 / \text{giorno} = 20000$

Ipotizzando 4 byte per la memorizzazione dell'attributo in questione vengono occupati circa 80 kb

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Alloggio	E	1	L
P-A	R	1	L
Prenotazione	E	16	L
Prenotazione Confermata	E	13	L
Utente Recensisce Host	R	11	L
Recensione Scritta Dall'utente	R	11	L
Recensione Casa	E	11	L

Senza ridondanza

Costo = $64 * 20000 / \text{giorno} = 1280000$

Operazione: Aggiunta recensione utente-casa

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Recensione Casa	E	1	L
Recensione scritta dall'utente	E	1	L
Utente recensisce Host	R	1	L
Prenotazione confermata	E	1	L
Prenotazione	E	1	L
P-A	R	1	L
Alloggio	E	1	L
Alloggio	E	1	S

Senza ridondanza non si necessita dell'aggiornamento dell'attributo, per cui non vi son costi

Costo: $9 * 550 / \text{giorno} = 4950$

Si ritiene debba essere mantenuta la ridondanza

Ridondanza su attributo "media valutazioni" di Alloggio, ricavabile anche attraverso l'analisi della somma di Recensioni casa / Numero prenotazioni di Alloggio

Operazione: Visualizzazione dati Alloggio

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Alloggio	E	1	L

Con ridondanza

Costo = $1 * 20000 / \text{giorno} = 20000$

Ipotizzando 4 byte per la memorizzazione dell'attributo in questione vengono occupati circa 80 kb

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Alloggio	E	1	L
P-A	R	16	L
Prenotazione	E	16	L
Prenotazione confermata	E	13	L
Utente recensisce Host	R	11	L
Recensione scritta dall'utente	E	11	L
Recensione casa	E	11	L

Senza ridondanza

Costi: $79 \times 20000 / \text{giorno} = 1580000$

Non si necessita spazio aggiuntivo

Operazione: Aggiunta recensione utente-casa

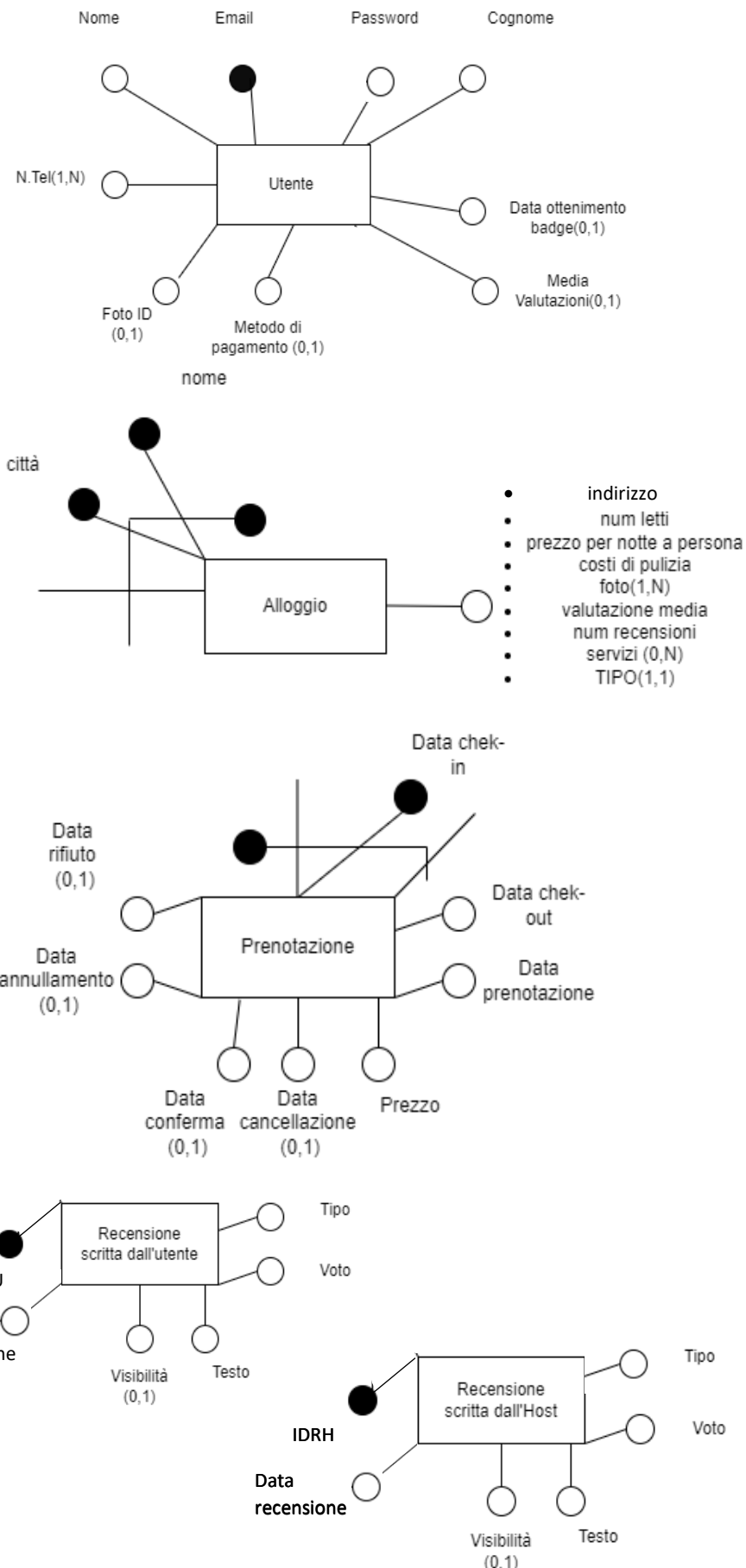
CONCETTO	COSTRUTTO	ACCESSI	TIPO
Recensione casa	E	1	L
Recensione scritta dall'utente	E	1	L
Utente recensisce Host	R	1	L
Prenotazione confermata	E	1	L
Prenotazione	E	1	L
P-A	R	1	L
Alloggio	E	1	L
Alloggio	E	1	R

Senza ridondanza non si necessita dell'aggiornamento dell'attributo, per cui non vi son costi

Costo: $9 \times 550 / \text{giorno} = 4950$

Si ritiene debba essere mantenuta la ridondanza

2.3.2 Eliminazione delle generalizzazioni



GENERALIZZAZIONE UTENTE:

abbiamo deciso di accorpare i vari livelli di generalizzazioni figlie nel genitore, riducendo il numero di accessi

GENERALIZZAZIONE ALLOGGIO:

abbiamo deciso di accorpare le generalizzazioni figlie nel genitore, riducendo il numero di accessi

GENERALIZZAZIONE PRENOTAZIONE:

abbiamo deciso di accorpare i vari livelli di generalizzazioni figlie nel genitore, riducendo il numero di accessi

GENERALIZZAZIONE RECENSIONE:

abbiamo deciso di dividere la generalizzazione in due entità a se stanti, essendo quella di partenza una generalizzazione totale, e non potendo fare diversamente, essendo le chiavi frutto di collegamento a relazioni con entità esterne diverse

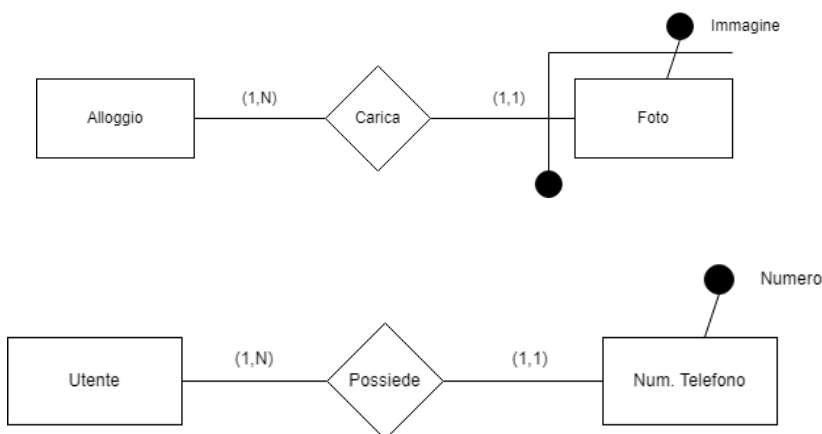
2.3.3. Eventuale partizionamento/accorpamento di entità e associazioni

In seguito alla eliminazione della generalizzazione su recensione si è dovuto dividere la relazione “afferenza” in “afferenza host” e “afferenza utente”.

2.3.4. Eventuale scelta degli identificatori principali

Abbiamo scelto di utilizzare degli ID per Thread, Prenotazione, Recensione Scritta Dall’utente e Recensione Scritta Dall’host in modo da rendere più snelle le relazioni e minimizzare il numero di attributi chiave per relazione.

2.3.5 Eliminazione degli attributi multivalore

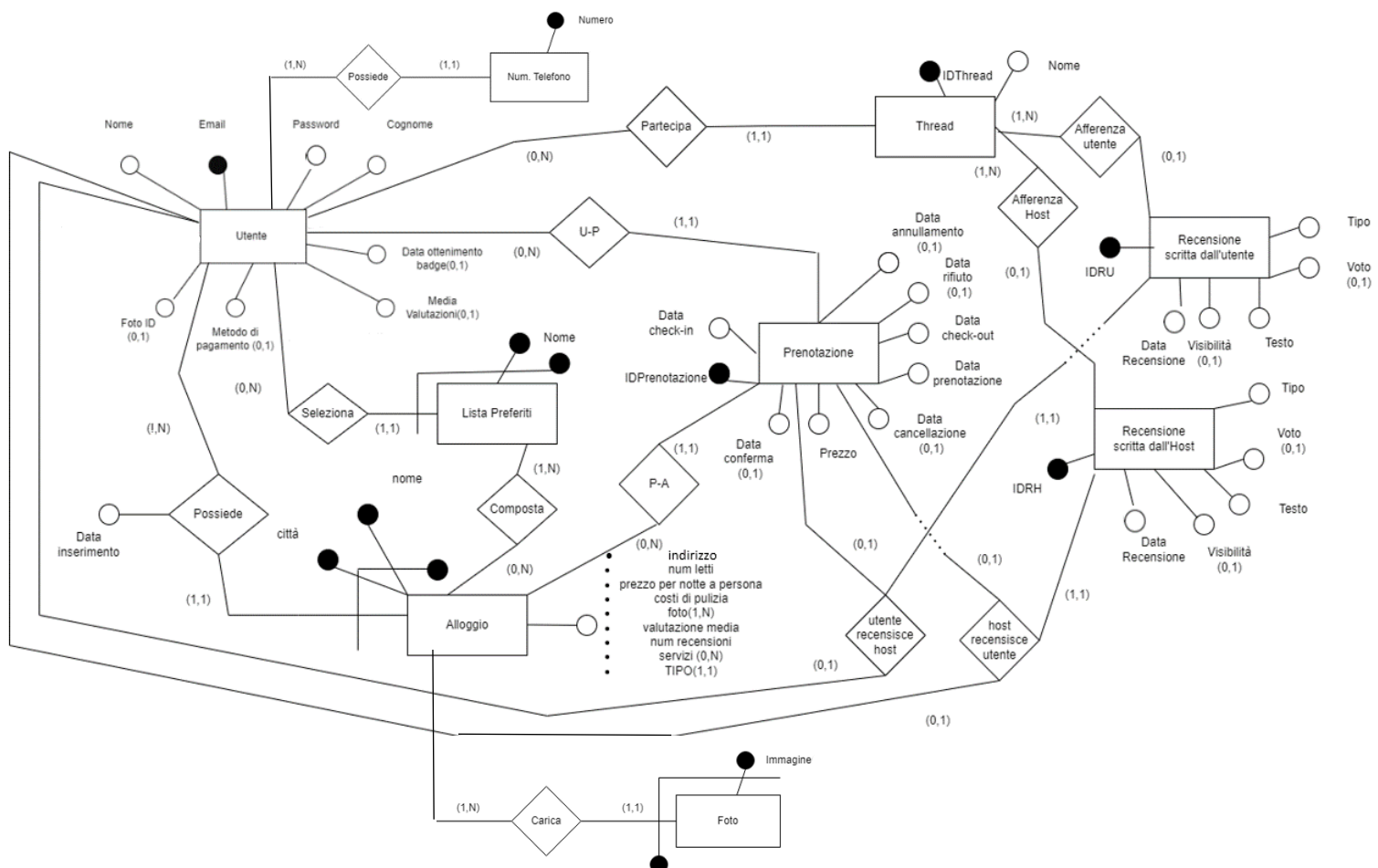


Nello schema E-R originario erano presenti due attributi multivalore, N.Tel (1,N), attributo dell’entità Utente, e Foto (1,N), attributo dell’entità Alloggio.

Poiché nel modello relazionale non è possibile rappresentare attributi multivalore, entrambi gli attributi sono stati sostituiti da un’entità rappresentante l’attributo stesso (con a sua volta un attributo che

rappresenta il contenuto) e una relazione che unisce la nuova entità e l’entità che possedeva precedentemente.

2.4. Schema E-R ristrutturato + business rules.



- Utente potrà vedere l'indirizzo preciso dell'alloggio solo a prenotazione completata
- Testo della recensione visibile quando entrambi (utente e host) hanno fatto recensione, o a 30 giorni dalla fine del soggiorno
- se una prenotazione non viene confermata dall'host entro 24h questa viene rifiutata (prenotazione in sospeso)
- un alloggio non può essere collegato a più prenotazioni nelle stesse date
- il metodo di pagamento si può aggiungere solo se si è già aggiunto il documento di identità (ex-utente verificato ->utente attivo)
- media valutazioni ammette valori solo se utente è in relazione tramite Possiede con alloggio (se quinti è un host)
- data ottenimento badge ha valore solo se Utente è un host (è in relazione tramite Possiede con alloggio) e rispetta i seguenti criteri:
 - aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.
 - aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.
 - aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.
- Tipo in Alloggio assume valori in {"intero appartamento", "stanza privata", "stanza condivisa"}
- data Conferma in prenotazione ha valore in seguito alla conferma da parte dell'host, mentre prende in alternativa valore data Rifiuto se l'host la rifiuta. Data annullamento prende valore se annullata dall'utente prima di data conferma mentre data cancellazione prende valore se viene cancellata dall'utente dopo data conferma.(se ha valore solo data conferma dopo data check-out vuol dire che il soggiorno si è concluso correttamente)
- tipo in recensione scritta dall'utente assume valori in {"recensione host", "recensione casa", "commento"}
- tipo in recensione scritta dall'host assume valori in {"recensione utente", "commento"}

2.5. Schema relazionale

UTENTE(Email, Password, Nome, Cognome, Foto ID*, Metodo di Pagamento*, MediaValutazioni*, DataOttenimentoBadge*)

NUM.TELEFONO(Numero, Utente)

ALLOGGIO(Città, Nome, Utente_email, NumLetti, indirizzo, PrezzoNottePersona, CostiPulizia, ValutazioneMedia, NRecensioni, Servizi*, Tipo, DataInserimento)

PRENOTAZIONE(IDPrenotazione, Alloggio_Città, Alloggio_Nome, Host_email, Utente_email, DPrenotazione, DCheckIn, DCheckOut, Prezzo, DConferma*, DCancellazione*, DRifiuto*, DAnnullamento*)

FOTO(Immagine, Alloggio_Città, Alloggio_Nome, Alloggio_Utente_Email)

RECENSIONESCRITTADAUTENTE(IDRU, Data Recensione ,Prenotazione,Utente_email, ,Tipo, Voto*, Testo, Visibilità*,NomeThread*)

RECENSIONESCITTADAHOST(IDRH,Data Recensione ,Prenotazione, Host_email, ,Tipo,Voto*, Testo, Visibilità*, NomeThread*)

THREAD(IDthread,Nome, Utente_email)

LISTAPREFERITI(Nome, Utente_email)

COMPOSTA(Lista Preferiti Nome, Utente_email, Alloggio Città, Alloggio Nome, Host_email)

UTENTERECENSISCEHOST(Prenotazione, Utente_email, IDRU)

HOSTRECENSISCEUTENTE(Prenotazione, Host_email, IDRH)

Con vincolo di integrità referenziale fra:

- l'attributo "Utente" in Num.Telefono e la chiave della relazione Utente
- l'attributo "Utente_email" in Alloggio e la chiave della relazione Utente
- gli attributi "Alloggio_Città, Alloggio_Nome, Host_email" in Foto e la chiave della relazione Alloggio
- gli attributi "Alloggio_Città, Alloggio_Nome, Host_email" in Prenotazione e le chiavi della relazione Alloggio
- l'attributo "Utente_email" in Prenotazione e la chiave della relazione Utente
- l'attributo "Utente_email" in RecensioneScrittaDall'Utente e la chiave della relazione Utente
- l'attributo "Prenotazione" in RecensioneScrittaDall'Utente e la chiave di Prenotazione
- l'attributo "Host_email" in RecensioneScrittaDall'Host e la chiave della relazione Utente
- l'attributo "Prenotazione" in RecensioneScrittaDall'Host e la chiave di Prenotazione
- l'attributo "Utente_email" in Thread e la chiave della relazione Utente
- l'attributo "Utente_email" in Lista Preferiti e la chiave della relazione Utente
- gli attributi "Lista Preferiti_Nome, Utente_email," in Composta e le chiavi della relazione Lista Preferiti
- gli attributi "Alloggio_Città, Alloggio_Nome, Host_email" in Composta e le chiavi della relazione Alloggio
- l'attributo "Prenotazione" in UtenteRecensisceHost e la chiave di Prenotazione
- l'attributo "Prenotazione" in HostRecensisceUtente e la chiave di Prenotazione
- l'attributo "Utente_email" in UtenteRecensisceHost e la chiave della relazione Utente
- l'attributo "Host_email" in HostRecensisceUtente e la chiave della relazione Utente
- l'attributo "IDRU" in UtenteRecensisceHost e la chiave della relazione RecensioneScrittaDall'Utente
- l'attributo "IDRH" in HostRecensisceUtente e la chiave della relazione RecensioneScrittaDall'Host

3.1. Implementazione DDL

```
begin;  
  
create table UTENTE (  
    email varchar(50) not null,  
    UPassword varchar(50) not null,  
    nome varchar(50) not null,
```

```

    cognome varchar(50) not null,
    fotoID boolean,
    metodoDiPagamento varchar(50),
    mediaValutazioni float(4),
    DataOtttenimentoBadge date,
    constraint UTENTE_PK primary key(email)
);

create table NUMTELEFONO (
    numero decimal(12) not null,
    Utente_email varchar(50) not null,
    constraint NUMTELEFONO_PK primary key(numero),
    constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade
);

create table ALLOGGIO (
    email varchar(50) not null,
    Citta varchar(50) not null,
    indirizzo varchar(50) not null,
    Nome varchar(50) not null,
    numletti decimal(5) not null,
    PrezzoNottePersona float(5) not null,
    CostiPulizia float(5) not null,
    ValutazioneMedia decimal(5),
    NRecensioni decimal(5) not null,
    Servizi varchar(200),
    Tipo varchar(50) not null,
    DataInserimento date,

    constraint ALLOGGIO_PK primary key(Citta,Nome,email),
    constraint email_FK foreign key(email) references UTENTE(email) on delete cascade
);

create table FOTO(
    Immagine varchar(50) not null,
    Alloggio_Città varchar(50) not null,
    Alloggio_Nome varchar(50) not null,
    Host_email varchar(50) not null,
    constraint FOTO_PK primary key(Immagine, Alloggio_Città, Alloggio_Nome,
Host_email),
    constraint Alloggio_FK foreign key(Alloggio_Città, Alloggio_Nome, Host_email)
references ALLOGGIO(Citta,Nome,email) on delete cascade
);

create table PRENOTAZIONE(
    IDPrenotazione varchar(50) not null,
    Alloggio_Città varchar(50) not null,
    Alloggio_Nome varchar(50) not null,
    Host_email varchar(50) not null,
    Utente_email varchar(50) not null,

```

```

DPrenotazione date not null,
DCheckIn date not null,
DCheckOut date not null,
Prezzo float(5) not null,
DConferma date,
DCancellazione date,
DRifiuto date,
DAnnullamento date,

constraint PRENOTAZIONE_PK primary key(IDPrenotazione),
constraint Alloggio_FK foreign key(Alloggio_Città, Alloggio_Nome, Host_email)
references ALLOGGIO(Città,Nome,email) on delete cascade,
constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade
);

create table RECENSIONESCRITTADAUTENTE(
IDRU varchar(50) not null,
Data_Recensione date not null,
Prenotazione varchar(50) not null,
Utente_email varchar(50) not null,
Tipo varchar(50) not null,
Voto decimal(2),
Testo varchar(50) not null,
Visibilità boolean,
IdThread varchar(50),
constraint RECENSIONESCRITTADAUTENTE_PK primary key(IDRU),
constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade,
constraint thread_FK foreign key(IdThread) references THREAD(IDthread) on delete
cascade ,
constraint Prenotazione_FK foreign key(Prenotazione) references
PRENOTAZIONE(IDPrenotazione) on delete cascade
);

create table RECENSIONESCRITTADAHOST(
IDRH varchar(50) not null,
Data_Recensione date not null,
Prenotazione varchar(50) not null,
Host_email varchar(50) not null,
Tipo varchar(50) not null,
Voto decimal(2),
Testo varchar(50) not null,
Visibilità boolean,
IdThread varchar(50),
constraint RECENSIONESCRITTADAHOST_PK primary key(IDRH),
constraint Host_email_FK foreign key(Host_email) references UTENTE(email) on
delete cascade,
constraint Prenotazione_FK foreign key(Prenotazione) references
PRENOTAZIONE(IDPrenotazione) on delete cascade,

```



```

    constraint thread_FK foreign key(IdThread) references THREAD(IDthread) on delete
cascade
);
create table THREAD(
    IDthread varchar(50) not null,
    Nome varchar(50) not null,
    Utente_email varchar(50) not null,
    constraint THREAD_PK primary key(IDthread),
    constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade

);
create table LISTAPREFERITI(
    Nome varchar(50) not null,
    Utente_email varchar(50) not null,
    constraint LISTAPREFERITI_PK primary key(Nome,Utente_email),
    constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade

);
create table COMPOSTA(
    Lista_Preferiti_Nome varchar(50) not null,
    Utente_email varchar(50) not null,
    Alloggio_Città varchar(50) not null,
    Alloggio_Nome varchar(50) not null,
    Host_email varchar(50) not null,
    constraint COMPOSTA_PK primary key(Lista_Preferiti_Nome, Utente_email,
Alloggio_Città, Alloggio_Nome, Host_email),
    constraint Alloggio_FK foreign key(Alloggio_Città, Alloggio_Nome, Host_email)
references ALLOGGIO(Città,Nome,email) on delete cascade,
    constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade,
    constraint Listapreferiti_FK foreign key(Lista_Preferiti_Nome,Utente_email)
references LISTAPREFERITI(Nome ,Utente_email) on delete cascade

);

create table UTENTERECENSISCEHOST(
Prenotazione varchar(50) not null,
Utente_email varchar(50) not null,
IDRU varchar(50) not null,
constraint UTENTERECENSISCEHOST_PK primary key(IDRU, Utente_email, Prenotazione),
constraint Prenotazione_FK foreign key(Prenotazione) references
PRENOTAZIONE(IDPrenotazione) on delete cascade,
constraint Utente_email_FK foreign key(Utente_email) references UTENTE(email) on
delete cascade

);
create table HOSTRECENSISCEUTENTE(
Prenotazione varchar(50) not null,

```

```

Host_email varchar(50) not null,
IDRH varchar(50) not null,
constraint HOSTRECENSISCEUTENTE_PK primary key(IDRH, Host_email, Prenotazione),
constraint Prenotazione_FK foreign key(Prenotazione) references
PRENOTAZIONE(IDPrenotazione) on delete cascade,
constraint Utente_Host_FK foreign key(Host_email) references UTENTE(email) on
delete cascade

);
commit;

```

3.2 DML di popolamento

```

insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('emanuele.parusso@gmail.com','ema1234','Emanuele','Parusso',true,'carta',0,null);
insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('federico.paschetta@gmail.com','fede1234','Federico','Paschetta',true,'PayPal',0,
null);
insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('lorenzo.lombardi@gmail.com','lollo1234','Lorenzo','Lombardi',true,'carta',0,null)
;
insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('luca.parusso@gmail.com','luca1234','Luca','Parusso',true,'carta',0,null);
insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('anna.paschetta@gmail.com','anna1234','Anna','Paschetta',true,'PayPal',0,null);
insert into UTENTE(email,
UPassword,nome,cognome,fotoID,metodoDiPagamento,mediaValutazioni,DataOttenimentoBad
ge) values
('fabio.lombardi@gmail.com','fabio1234','Fabio','Lombardi',true,'bancomat',0,null);

insert into ALLOGGIO
(email,Citta,indirizzo,Nome,numletti,PrezzoNottePersona,CostiPulizia,ValutazioneMed
ia,NRecensioni,
Servizi,Tipo,DataInserimento)values('emanuele.parusso@gmail.com','Alba','via verdi
15','Casa in Langa',4,30,50,0,0,'wifi','Intero alloggio','2022-06-02');
insert into ALLOGGIO
(email,Citta,indirizzo,Nome,numletti,PrezzoNottePersona,CostiPulizia,ValutazioneMed

```

```

ia,NRecensioni,
Servizi,Tipo,DataInserimento)values('luca.parusso@gmail.com','Genova','via rossi
115','Casa al Mare',4,30,50,0,0,'wifi','stanza privata','2022-06-02');
insert into ALLOGGIO
(email,Citta,indirizzo,Nome,numletti,PrezzoNottePersona,CostiPulizia,ValutazioneMed
ia,NRecensioni,
Servizi,Tipo,DataInserimento)values('federico.paschetta@gmail.com','Carmagnola','vi
a gialli 215','Casa in periferia',4,30,50,0,0,'wifi','Intero alloggio','2022-06-
02');
insert into ALLOGGIO
(email,Citta,indirizzo,Nome,numletti,PrezzoNottePersona,CostiPulizia,ValutazioneMed
ia,NRecensioni,
Servizi,Tipo,DataInserimento)values('fabio.lombardi@gmail.com','Alba','via
garibaldi 15','Casa in città',4,30,50,0,0,'wifi','stanza condivisa','2022-06-02');

insert into
NUMTELEFONO(Numero,Utente_email)values(3332221111,'emanuele.parusso@gmail.com');
insert into
NUMTELEFONO(Numero,Utente_email)values(3332222222,'federico.paschetta@gmail.com');
insert into
NUMTELEFONO(Numero,Utente_email)values(3332223333,'lorenzo.lombardi@gmail.com');
insert into
NUMTELEFONO(Numero,Utente_email)values(3332224444,'luca.parusso@gmail.com');
insert into
NUMTELEFONO(Numero,Utente_email)values(3332225555,'anna.paschetta@gmail.com');
insert into
NUMTELEFONO(Numero,Utente_email)values(3332226666,'fabio.lombardi@gmail.com');

insert into
FOTO(Immagine,Alloggio_Città,Alloggio_Nome,Host_Email)values('casa.jpg','Alba','Cas
a in Langa','emanuele.parusso@gmail.com');
insert into
FOTO(Immagine,Alloggio_Città,Alloggio_Nome,Host_Email)values('casetta.jpg','Carmagn
ola','Casa in periferia','federico.paschetta@gmail.com');
insert into
FOTO(Immagine,Alloggio_Città,Alloggio_Nome,Host_Email)values('casotta.png','Alba','
Casa in città','fabio.lombardi@gmail.com' );

insert into THREAD(IDthread,Nome,Utente_email)values('T01','bello sto
alloggio','lorenzo.lombardi@gmail.com');
insert into THREAD(IDthread,Nome,Utente_email)values('T02','bella
vacanza','federico.paschetta@gmail.com');
insert into THREAD(IDthread,Nome,Utente_email)values('T03','host
simpatico','federico.paschetta@gmail.com');
insert into THREAD(IDthread,Nome,Utente_email)values('T04','è stato un ospite
delizioso','emanuele.parusso@gmail.com');

```

```

insert into LISTAPREFERITI(Nome,Utente_email)values('vacanza in
campagna','federico.paschetta@gmail.com');
insert into LISTAPREFERITI(Nome,Utente_email)values('vacanza al
mare','emanuele.parusso@gmail.com');
insert into LISTAPREFERITI(Nome,Utente_email)values('per quando avrò
soldi','anna.paschetta@gmail.com');
insert into LISTAPREFERITI(Nome,Utente_email)values('per
capodanno','lorenzo.lombardi@gmail.com');

insert into
COMPOSTA(Lista_Preferiti_Nome,Utente_email,Alloggio_Città,Alloggio_Nome,Host_email)
values('per capodanno','lorenzo.lombardi@gmail.com','Alba','Casa in
Langa','emanuele.parusso@gmail.com');
insert into
COMPOSTA(Lista_Preferiti_Nome,Utente_email,Alloggio_Città,Alloggio_Nome,Host_email)
values('vacanza in campagna','federico.paschetta@gmail.com','Alba','Casa in
città','fabio.lombardi@gmail.com');
insert into
COMPOSTA(Lista_Preferiti_Nome,Utente_email,Alloggio_Città,Alloggio_Nome,Host_email)
values('vacanza al mare','emanuele.parusso@gmail.com','Genova','Casa al
Mare','luca.parusso@gmail.com');
insert into
COMPOSTA(Lista_Preferiti_Nome,Utente_email,Alloggio_Città,Alloggio_Nome,Host_email)
values('per quando avrò soldi','anna.paschetta@gmail.com','Genova','Casa al
Mare','luca.parusso@gmail.com');

insert into PRENOTAZIONE
(IDPrenotazione,Alloggio_Città,Alloggio_Nome,Host_email,Utente_email,DPrenotazione,
DCheckIn,DCheckOut,Prezzo,DConferma)values('A01','Alba','Casa in
Langa','emanuele.parusso@gmail.com','lorenzo.lombardi@gmail.com','2022-04-
10','2022-05-20','2022-05-27',1680,'2022-04-10');
insert into PRENOTAZIONE
(IDPrenotazione,Alloggio_Città,Alloggio_Nome,Host_email,Utente_email,DPrenotazione,
DCheckIn,DCheckOut,Prezzo,DConferma)values('A02','Alba','Casa in
Langa','emanuele.parusso@gmail.com','federico.paschetta@gmail.com','2022-05-
11','2022-06-10','2022-06-17',1680,'2022-05-12');
insert into PRENOTAZIONE
(IDPrenotazione,Alloggio_Città,Alloggio_Nome,Host_email,Utente_email,DPrenotazione,
DCheckIn,DCheckOut,Prezzo,DRifiuto)values('B01','Alba','Casa in
città','fabio.lombardi@gmail.com','anna.paschetta@gmail.com','2022-02-05','2022-03-
14','2022-03-28',1680,'2022-02-05');
insert into PRENOTAZIONE
(IDPrenotazione,Alloggio_Città,Alloggio_Nome,Host_email,Utente_email,DPrenotazione,
DCheckIn,DCheckOut,Prezzo,DConferma,DAnnullamento)values('B02','Carmagnola','Casa
in periferia','federico.paschetta@gmail.com','luca.parusso@gmail.com','2022-02-
10','2022-03-19','2022-03-26',1680,'2022-02-20','2022-03-10');

```

```

insert into PRENOTAZIONE
(IDPrenotazione,Alloggio_Città,Alloggio_Nome,Host_email,Utente_email,DPrenotazione,
DCheckIn,DCheckOut,Prezzo,DConferma)values('B03','Genova','Casa al
Mare','luca.parusso@gmail.com','lorenzo.lombardi@gmail.com','2022-01-22','2022-04-
03','2022-04-10',1680,'2022-01-23');

insert into
UTENTERECENSISCEHOST(Prenotazione,Utente_email,IDRU)values('A01','lorenzo.lombardi@
gmail.com','R01');
insert into
UTENTERECENSISCEHOST(Prenotazione,Utente_email,IDRU)values('A02','federico.paschett
a@gmail.com','R02');
insert into
UTENTERECENSISCEHOST(Prenotazione,Utente_email,IDRU)values('A02','federico.paschett
a@gmail.com','R03');
insert into
UTENTERECENSISCEHOST(Prenotazione,Utente_email,IDRU)values('B03','lorenzo.lombardi@
gmail.com','R04');
insert into
UTENTERECENSISCEHOST(Prenotazione,Utente_email,IDRU)values('A01','lorenzo.lombardi@
gmail.com','R05');

insert into
HOSTRECENSISCEUTENTE(Prenotazione,Host_email,IDRH)values('A01','emanuele.parusso@gm
ail.com','RH01');
insert into
HOSTRECENSISCEUTENTE(Prenotazione,Host_email,IDRH)values('A02','emanuele.parusso@gm
ail.com','RH02');
insert into
RECENSIONESCRITTADAUTENTE(IDRU,Data_Recensione,Prenotazione,Utente_email,Tipo,Voto,
Testo,Visibilità,IdThread)values('R01','2022-05-
28','A01','lorenzo.lombardi@gmail.com','Casa',9,'Casa perfetta, ottima
location','true','T01');
insert into
RECENSIONESCRITTADAUTENTE(IDRU,Data_Recensione,Prenotazione,Utente_email,Tipo,Voto,
Testo,Visibilità,IdThread)values('R02','2022-06-
20','A02','federico.paschetta@gmail.com','Casa',8,'Casa buona','false','T02');
insert into
RECENSIONESCRITTADAUTENTE(IDRU,Data_Recensione,Prenotazione,Utente_email,Tipo,Voto,
Testo,Visibilità,IdThread)values('R03','2022-06-
20','A02','federico.paschetta@gmail.com','Host',8,'Host simpatico','false','T03');
insert into
RECENSIONESCRITTADAUTENTE(IDRU,Data_Recensione,Prenotazione,Utente_email,Tipo,Voto,
Testo,Visibilità)values('R04','2022-04-
12','B03','lorenzo.lombardi@gmail.com','Casa',2,'Casa pessima e posizione
scomoda','false');
insert into
RECENSIONESCRITTADAUTENTE(IDRU,Data_Recensione,Prenotazione,Utente_email,Tipo,Testo

```

```
,Visibilità,IdThread)values('R05','2022-06-02','A01','lorenzo.lombardi@gmail.com','Commento','Grazie','true','T04');

insert into
RECENSIONESCRITTADAHOST(IDRH,Data_Recensione,Prenotazione,Host_email,Tipo,Voto,Testo,Visibilità,IdThread) values('RH01','2022-05-24','A01','emanuele.parusso@gmail.com','Utente',10,'Ospite molto cordiale e rispettoso','true','T04');
insert into
RECENSIONESCRITTADAHOST(IDRH,Data_Recensione,Prenotazione,Host_email,Tipo,Voto,Testo,Visibilità) values('RH02','2022-06-20','A02','emanuele.parusso@gmail.com','Utente',8,'Ospite educato','true');
```