

2021/2022

Progetto laboratorio Basi di Dati

Lara Novelli 916272

Progetto di piattaforma di home booking

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 e stanze condivise¹.

Gli utenti si registrano al servizio fornendo indirizzo e-mail, 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% 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, una descrizione, il prezzo per notte per persona e i costi di pulizia, delle foto, i servizi, 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.

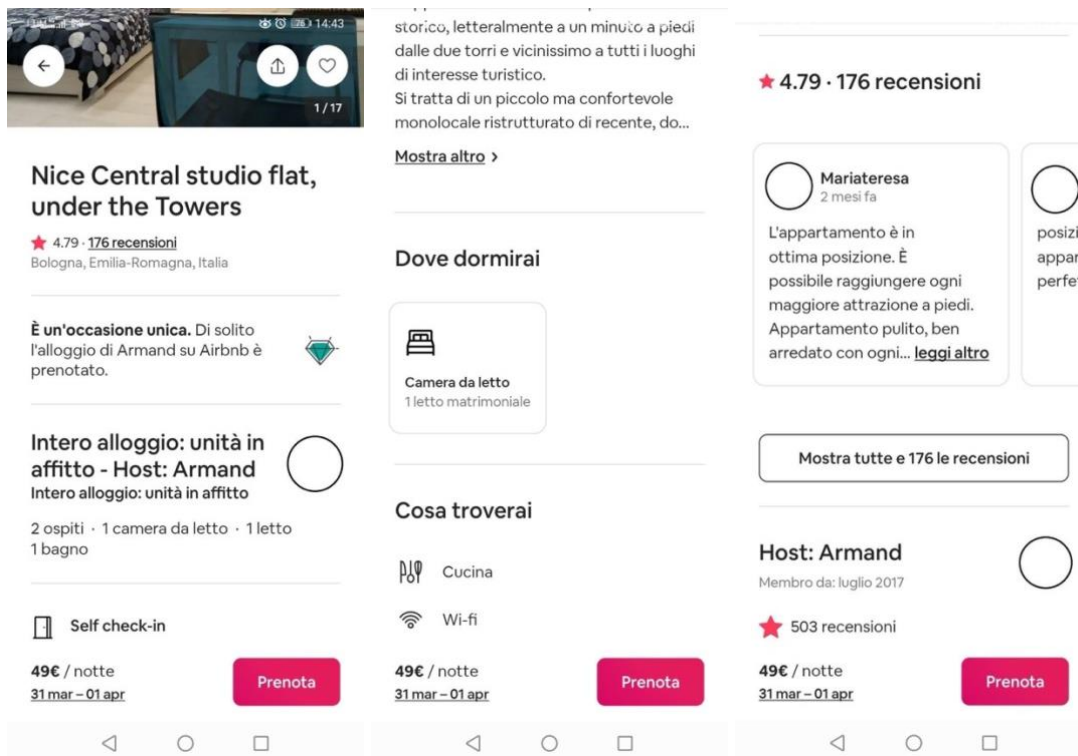


Fig. 1: Dettagli di un appartamento.

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. 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.

¹Il servizio descritto è liberamente ispirato a Airbnb (<https://www.airbnb.it/>) a cui è possibile fare riferimento per completare e disambiguare i requisiti.

1.2 Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	COLLEGAMENTI
Utente	Persona che usufruisce del servizio di affitto degli alloggi	Persona	Recensione, metodo_pagamento
Ospite	Persona che prenota l'alloggio	Utente	Utente, recensione
Host	Persona che affitta l'alloggio	Utente	Utente, recensione, alloggi
Alloggio	Spazio prenotabile	Appartamento, stanza privata, stanza condivisa	Utente, Recensione
Servizi	Servizi disponibili	-	Alloggi, recensione
Recensione	Serie di giudizi critici sotto forma di testo o punteggio	Valutazione	Utente, host, ospite, recensione, servizi
Prenotazione	Richiesta di soggiorno in un determinato lasso di tempo per un determinato numero di ospiti presso un alloggio	-	Utenti, alloggi, metodo_pagamento

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 permetta di affittare e prenotare alloggi di vario tipo

Frasi relative agli alloggi: sono di vario tipo, come: interi appartamenti, stanze private e stanze condivise.

Vengono descritti indicando:

- un nome.
- un indirizzo,
- descrizione,
- prezzo per notte a persona,
- costi di pulizia,
- foto,
- servizi (ad esempio cucina, wi-fi, lavatrice, ...)
- numero di letti,
- orario check-in e check-out,
- host di appartenenza,
- rating medio,
- numero di recensioni

Frase relative all'utente: gli utenti che si registrano al servizio forniscono:

- nome, cognome,
- indirizzo,
- email,
- password,
- numero o numeri di telefono.

L'utente deve indicare un metodo di pagamento per poter prenotare.

Gli utenti possono essere ospiti o "host".

Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti.

Frase relative agli ospiti: se gli ospiti sono a loro volta utenti del servizio, se ne possono indicare i nominativi.

L'ospite può cancellare la prenotazione.

L'ospite può valutare servizio, host e alloggio.

Gli ospiti possono commentare più volte le review in cui sono coinvolti creando un thread di discussione.

Frase relative agli host: gli host sono utenti.

Gli host possono diventare superhost (ricevono un badge) se soddisfano determinati requisiti

Gli host possono confermare o rifiutare la prenotazione

L'host può valutare l'ospite

Gli host possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione

Frase relative a prenotazione: l'utente deve indicare un metodo di pagamento per poter prenotare.

Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti.

La prenotazione deve essere confermata o rifiutata dall'host.

La prenotazione ha un costo totale e se confermata viene eseguito il pagamento.

La prenotazione può essere cancellata sia dall'ospite che dall'host.

Frase relative alle recensioni: la recensione fatta dagli ospiti comprende due testi e una serie di punteggi in una scala da 1 a 5 su dimensioni:

- pulizia,
- comunicazione,
- posizione,
- qualità/prezzo.

Le recensioni degli host comprendono solo un commento testuale.

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

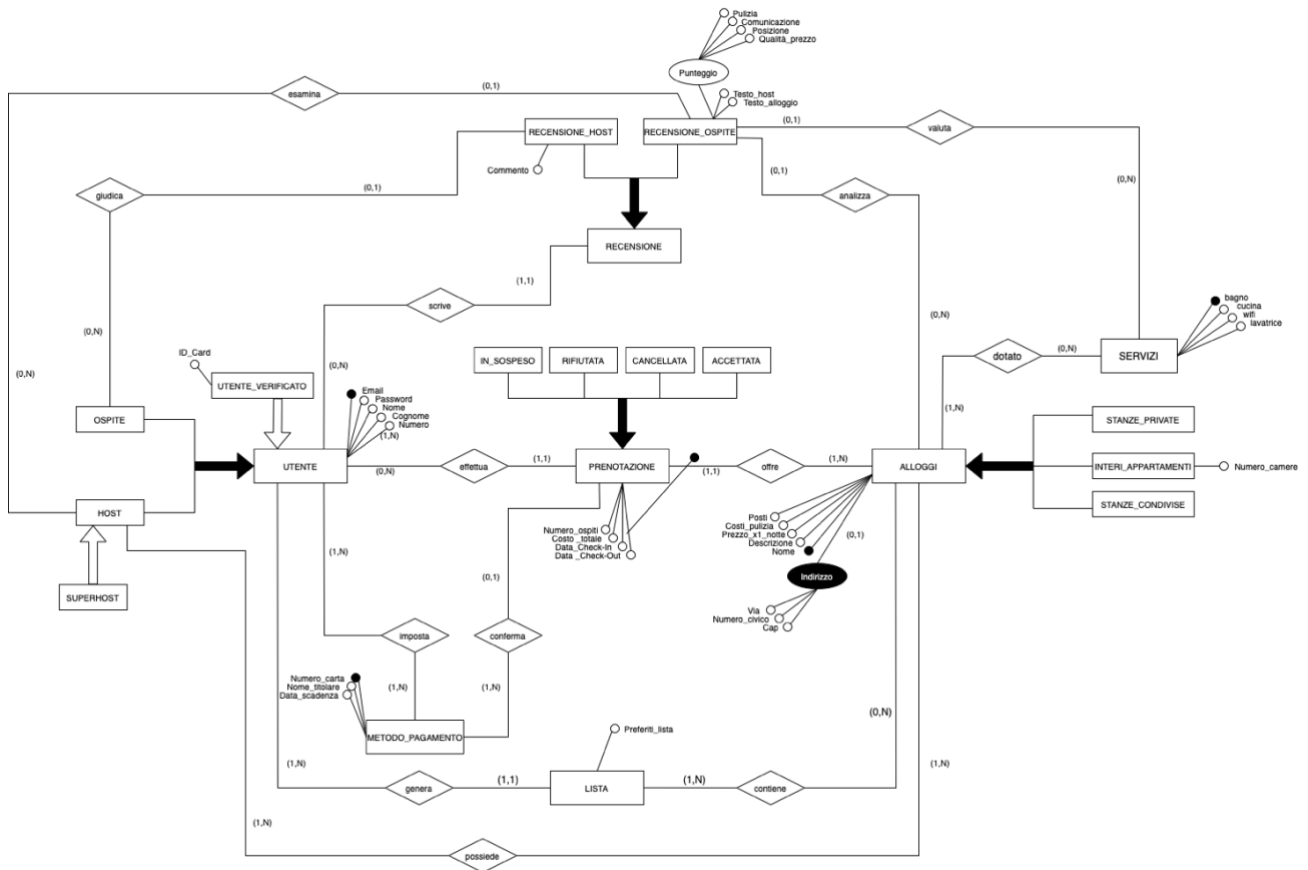
Frase relative alla valutazione: al termine del soggiorno gli ospiti e gli host si possono valutare a vicenda.

La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sulle singole dimensioni.

Frase relative al servizio: il servizio permette di affittare e prenotare alloggi di vario tipo.

Gli utenti si registrano al servizio

1.4. Schema E-R e business rules.



Dizionario dei dati (entità)

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICAZIONE
Utente	Persona che usufruisce del servizio di affitto	Email, password, nome, cognome, numero	Email
Utente_verificato	Utente che fornisce la foto della carta d'identità	ID_Card	Identificatore ereditato da [utente]: email
Superhost	Tipologia di host	-	Identificatore ereditato da [utente]: email
Ospite	Persona che prenota l'alloggio	-	Identificatore ereditato da [utente]: email
Host	Persona che affitta l'alloggio	-	Identificatore ereditato da [utente]: email
Prenotazione	Richiesta di pernottamento	Data_Check-In, data_Check-Out, numero_ospiti, costo_totale	Data_Check-In
In_sospeso	Fase di prenotazione	-	Identificatore ereditato da [prenotazione]: Data_Check-In

Rifiutata	Fase di prenotazione	-	Identificatore ereditato da [prenotazione]: Data_Check-In
Cancellata	Fase di prenotazione	-	Identificatore ereditato da [prenotazione]: Data_Check-In
Accettata	Fase di prenotazione	-	Identificatore ereditato da [prenotazione]: Data_Check-In
Recensione	Serie di giudizi critici sotto forma di testo o punteggio	-	-
Recensione_ospite	Serie di giudizi critici sotto forma di testo o punteggio	Punteggio, testo_host, testo_ospite	-
Recensione_host	Serie di giudizi critici sotto forma di testo o punteggio	Commento	-
Alloggi	Spazio prenotabile	Posti, costi_pulizia, prezzo_x1_notte, descrizione, nome, indirizzo	Nome, indirizzo, identificatore ereditato da [prenotazione]: Data_Check-In
Stanze_condivise	Tipologia di alloggio	-	Identificatori ereditato da [alloggi]: nome, indirizzo
Stanze_private	Tipologia di alloggio	-	Identificatori ereditato da [alloggi]: nome, indirizzo
Interi_appartamenti	Tipologia di alloggio	Numero_camere	Identificatori ereditato da [alloggi]: nome, indirizzo
Servizi	Servizi offerti con l'alloggio	Bagno, cucina, WIFI, lavatrice	Bagno
Metodo_pagamento	Metodo con cui si esegue il pagamento della prenotazione	Numero_carta, nome_titolare, data_scadenza	Numero_carta
Lista	Lista dei preferiti	Preferiti_lista	-

Dizionario dei dati (relazioni)

RELAZIONE	DESCRIZIONE	COMPONENTI	ATTRIBUTI
Effettua	Metodo tramite cui si effettua una prenotazione	Utente (0, N), prenotazione (1, 1)	-
Offre	Cosa comprende la prenotazione	Prenotazione (1, 1), alloggi (1, N)	-
Dotato	Cosa comprende l'alloggio	Alloggi (1, N), servizi (0, N)	-
Imposta	Metodo con cui si imposta il pagamento	Utente (1, N), metodo di pagamento (1, N)	-
Conferma	Metodo tramite cui si procede al pagamento	Metodo di pagamento (1, N), Prenotazione (0, 1)	-
Genera	L'utente ha la possibilità di avere una lista	Utente (1, N), lista (1, 1)	-
Contiene	La lista contiene gli alloggi	Lista (1, N), alloggi (0, N)	-
Possiede	L'host possiede degli alloggi	Host (1, N), alloggi (1, N)	-
Scrive	L'utente può scrivere una recensione	Utente (0, N), recensione (1, 1), conclusa (0, 1)	-
Giudica	L'host giudica l'ospite	Ospite (0, N), recensione host (0, 1)	-
Esamina	L'ospite giudica l'host	Host (0, N), recensione ospite (0, 1)	-
Valuta	L'ospite valuta i servizi dell'appartamento	Recensione ospite (0, 1), servizi (0, N)	-
Analizza	L'ospite valuta l'appartamento	Recensione ospite (0, 1), alloggi (0, N)	-

Vincoli d'integrità

Gli Utenti che vogliono usufruire del servizio devono fornire dati necessari

**Un account viene verificato se fornita una foto del documento d'identità in corso di validità
L'identificazione di un metodo di pagamento è necessaria affinché si possa richiedere una prenotazione**

Gli Host diventano Superhost se: hanno completato almeno 10 soggiorni, per un totale di almeno 100 notti; hanno conservato un tasso di cancellazione dell'1% massimo; hanno

mantenuto una valutazione complessiva del 4,8 considerando tutti i soggiorni in tutte le case di proprietà

Se si conferma la prenotazione viene eseguito il pagamento

La recensione fatta dagli ospiti comprende due testi e una serie di punteggi in una scala da 1 a 5

Le recensioni sono visibili quando entrambi hanno fatto la recensione oppure, se uno dei due non l'ha fatta, 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

Derivazioni

La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sulle singole dimensioni.

Progettazione logica

2.1 Tavola dei volumi

CONCETTO	TIPO	VOLUME	SPIEGAZIONE
Utente	E	30 000	stima
Utente verificato	E	3 000	
Superhost	E	2 000	
Ospite	E	15 000	
Host	E	10 000	
Prenotazione	E	40 000	stima
In sospeso	E	14 000	
stimsRifiutata	E	9 000	
Cancellata	E	2 000	
Accettata	E	15 000	
Recensione	E	45 000	stima
Recensione ospite	E	30 000	
Recensione host	E	15 000	
Alloggi	E	80 000	stima
Stanze condivisa	E	15 000	
Stanze privata	E	20 000	
Interi appartamenti	E	5 000	
Servizi	E	100	
Metodo di pagamento	E	1	
Lista	E	35 000	stima
Effettua	R		
Offre	R		
Dotato	R	3	tipologie
Imposta	R		
Conferma	R		
Genera	R		
Contiene	R	20 000	stima
Possiede	R		
Scrive	R		
Giudica	R		
Esamina	R		
Valuta	R		
Analizza	R		

2.2 Tavola delle operazioni

OPERAZIONI	TIPO	FREQUENZA
1: Numero di prenotazioni richieste dall'utente	I	30/anno
2: Calcolo per aggiornare il tasso di cancellazione di ciascun host	B	1/settimana
3: Controllo delle condizioni per la qualifica di Superhost	B	1/giorno
4: Aggiornamento dello stato degli host	B	1/giorno
5: Calcolo della classificazione degli alloggi più graditi	B	1/mese
6: Calcolo della media del voto	B	1/giorno
7: Calcolo costo totale	I	
8: registrazione nuovo utente	I	10 000/anno

I: operazioni interattive

B: operazioni batch

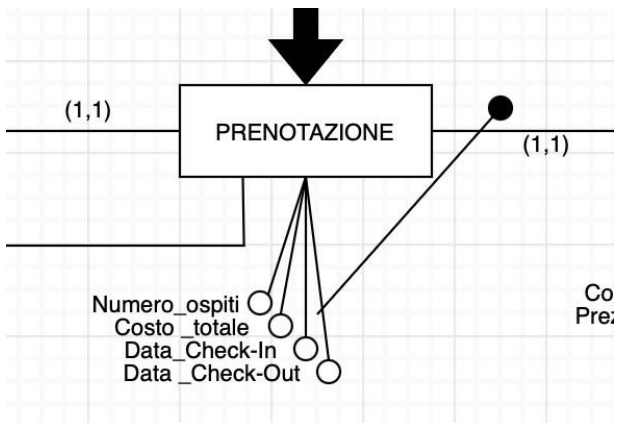
2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

Nello schema E-R è stata identificata una ridondanza: attributo 'Costo_totale' appartiene all'entità Prenotazione e attributi 'Costo_notte' e 'Costo_pulizia' appartengono all'entità Alloggi.

Operazione 7: calcolo del costo totale della prenotazione, con frequenza 30/giorno

Con ridondanza



CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	Entità	1	L

Accessi in lettura: 30

Senza ridondanza

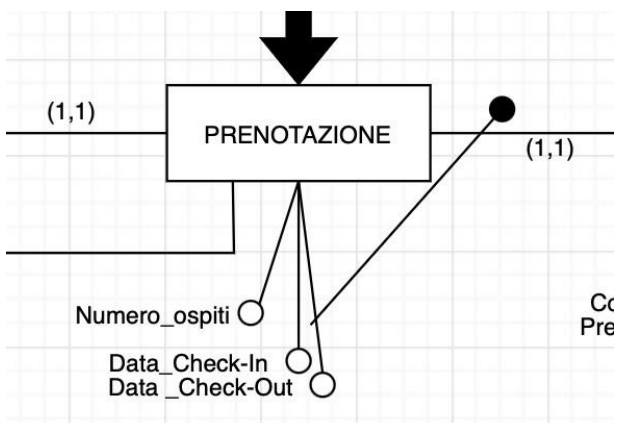


Tavola degli accessi

CONCETTO	COSTRUTTO	ACCESSI	TIPO
Prenotazione	Entità	1	L
Offre	Relazione	1	L
Alloggi	Entità	1	S
Alloggi	Entità	1	L

Accessi in lettura: 90

Accessi in scrittura: 60

Accessi totali: 150

CONCETTO	TIPO	VOLUME
Prenotazione	E	40 000

Costo aggiuntivo con ridondanza:

ipotesi: si utilizzano 4 byte per leggere il costo_totale in prenotazione

spazio necessario: $4 \times 40.000 = 160.000$ byte

Costo aggiuntivo senza ridondanza:

CONCETTO	TIPO	VOLUME
Prenotazione	E	40 000
	R	40 000
Alloggi	E	80 000
Alloggi	E	80 000

ipotesi: si utilizzano 4 byte per leggere il costo totale in prenotazione

spazio necessario: $4 \times 40.000 = 160.000$ byte

ipotesi: si utilizzano 4 byte per leggere il costo totale in prenotazione

spazio necessario: $4 \times 40.000 = 160.000$ byte

ipotesi: si utilizzano 4 byte per leggere il costo totale in prenotazione

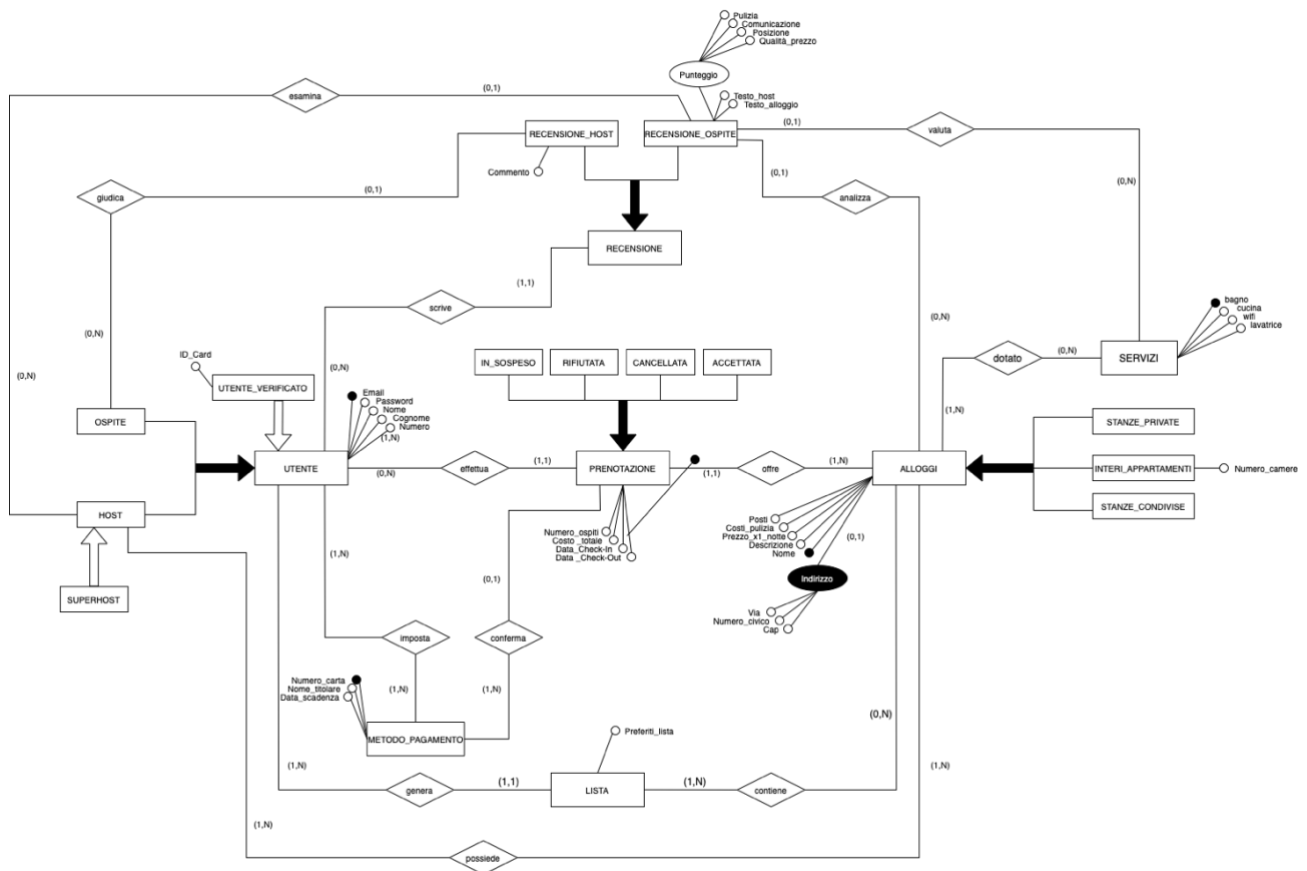
spazio necessario: $4 \times 80.000 = 320.000$ byte

ipotesi: si utilizzano 4 byte per leggere il costo totale in prenotazione

spazio necessario: $4 * 80.000 = 320.000$ byte.

LA RIDONDANZA È TRASCURABILE E QUINDI POSSIAMO MANTENERLA.

2.3.2 Eliminazione delle generalizzazioni



Accorpamento dell'entità figlia UTENTE_VERIFICATO con ID_card come attributo dell'entità UTENTE.

Accorpamento dell'entità figlia SUPERHOST come attributo nell'entità padre HOST.

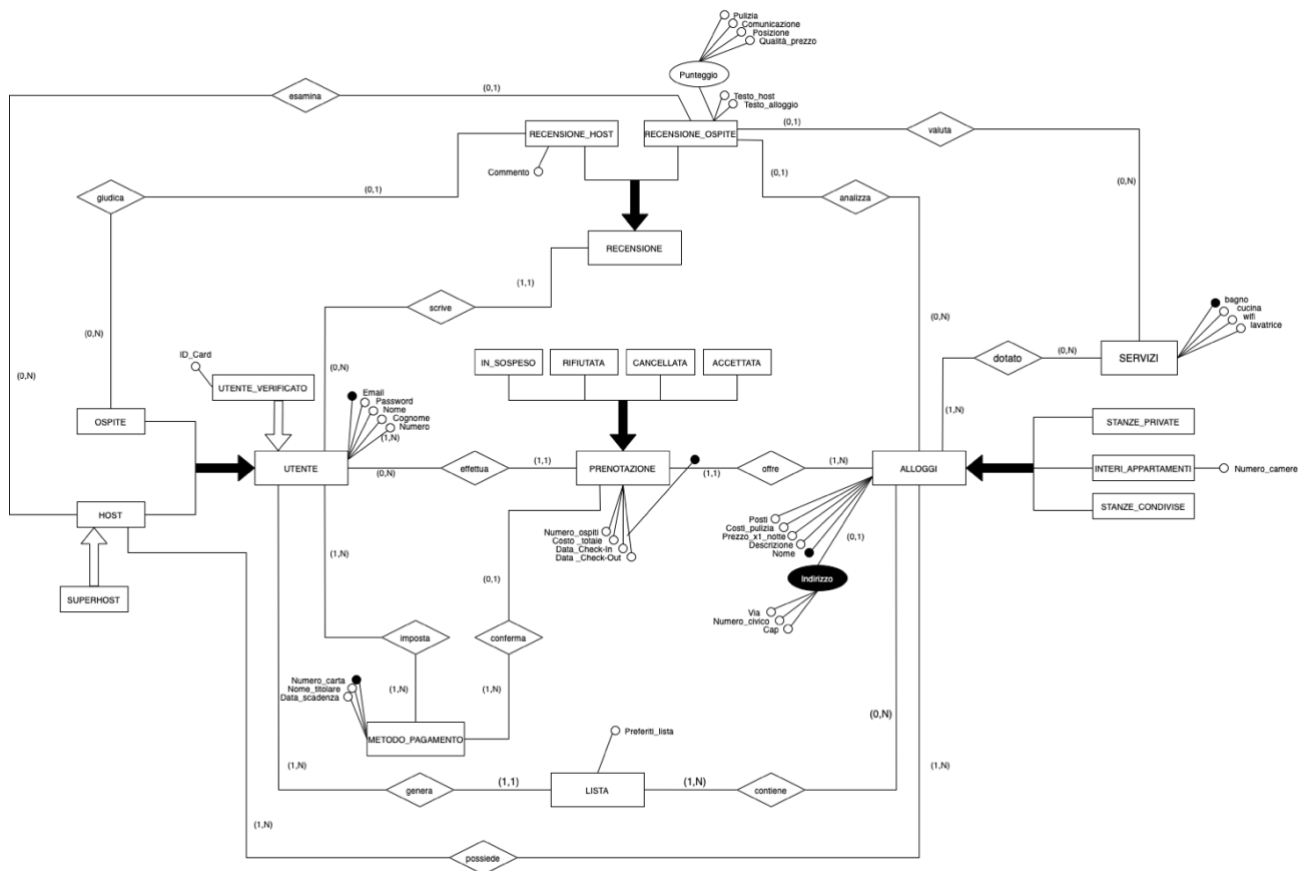
Accorpamento delle entità figlie ACCETTATA, IN_SOSPESA, RIFIUTATA, CANCELLATA con STATO_PRENOTAZIONE come attributo nell'entità padre PRENOTAZIONE.

Accorpamento delle entità figlie APPARTAMENTO, STANZA_PRIVATA, STANZA_CONDIVISA con tipo_alloggio come attributo nell'entità padre ALLOGGIO.

Aggiunta di due relazioni per mantenere separate le entità HOST e OSPITE poiché sono collegate a loro volta in maniera diversa con altre entità.

Aggiunta di due relazioni per mantenere separate le entità RECENSIONE_HOST e RECENSIONE_OSPITE poiché sono collegate a loro volta in maniera diversa con altre entità.

2.3.3. Partizionamento/accorpamento di entità e associazioni

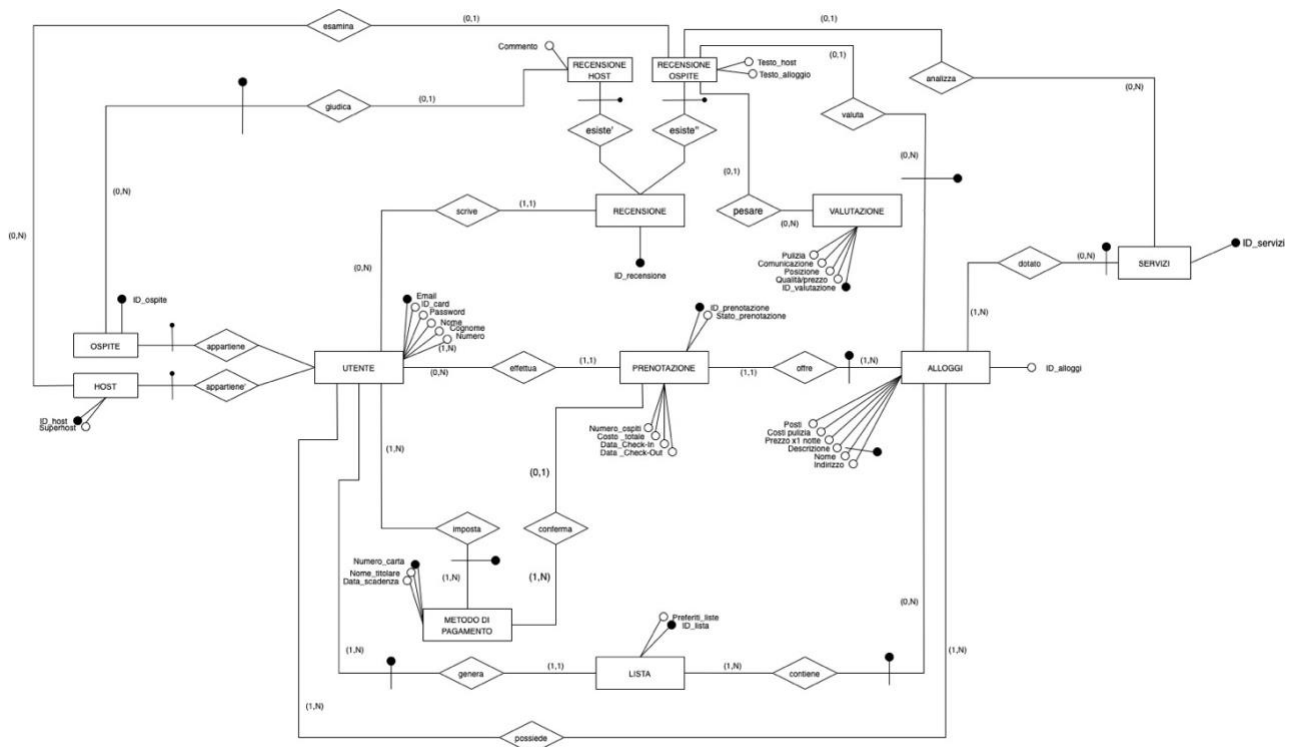


Partizionamento dell'attributo composto Punteggio e dell'attributo composto Indirizzo.

2.3.4. Scelta degli identificatori principali

ENTITA'	IDENTIFICATORE
Utente	Email
Prenotazione	Data_Check-In
Alloggi	Nome, Indirizzo
Metodo di pagamento	Numero_carta
Lista	ID_lista
Servizi	ID_servizi
Recensione	ID_recensione
Valutazione	ID_valutazione

2.4. Schema E-R ristrutturato e business rules



Dizionario dei dati (entità)

ENTITA'	DESCRIZIONE	ATTRIBUTI	IDENTIFICAZIONE
Utente	Persona che usufruisce del servizio di affitto degli alloggi	Email, ID_card, password, nome, cognome, numero	Email
Ospite	Persona che prenota l'alloggio	ID_ospite, email	ID_ospite, email
Host	Persona che affitta l'alloggio	ID_host, superhost, email	ID_host, email
Prenotazione	Richiesta di pernottamento	ID_prenotazione, ID_alloggi, Stato_prenotazione, Data_Check-In, data_Check-Out, numero_ospiti, costo_totale	ID_prenotazione, ID_alloggi
Recensione	Serie di giudizi critici sotto forma di testo o punteggio	ID_recensione, email	ID_recensione, email

Recensione_ospite	Serie di giudizi critici sotto forma di testo o punteggio	Testo_host, testo_ospite	ID_recensione, ID_ospite,
Recensione_host	Serie di giudizi critici sotto forma di testo o punteggio	Commento	ID_recensione, ID_host
Valutazione	Serie di punteggio riferiti all'alloggio	Pulizia, comunicazione, posizione, qualità-prezzo, ID_valutazione	ID_valutazione
Alloggi	Spazio prenotabile	Posti, costi_pulizia, prezzo_x1_notte, descrizione, nome, indirizzo, ID_alloggi	ID_alloggi
Servizi	Servizi offerti con l'alloggio	ID_servizi	ID_servizi, ID_alloggi
Metodo_pagamento	Metodo con cui si esegue il pagamento della prenotazione	Numero_carta, nome_titolare, data_scadenza, email	Numero_carta, email
Lista	Lista dei preferiti	Preferiti_lista, ID_alloggi, ID_lista	ID_lista, ID_alloggi

Dizionario di dati (relazioni)

RELAZIONE	DESCRIZIONE	COMPONENTI	ATTRIBUTI
Effettua	Metodo tramite cui si effettua una prenotazione	Utente (0, N), prenotazione (1, 1)	-
Offre	Cosa comprende la prenotazione	Prenotazione (1, 1), alloggi (1, N)	-
Dotato	Cosa comprende l'alloggio	Alloggi (1, N), servizi (0, N)	-
Imposta	Metodo con cui si imposta il pagamento	Utente (1, N), metodo di pagamento (1, N)	-
Conferma	Metodo tramite cui si procede al pagamento	Metodo di pagamento (1, N), Prenotazione (0, 1)	-
Genera	L'utente ha la possibilità di avere una lista	Utente (1, N), lista (1, 1)	-
Contiene	La lista contiene gli alloggi	Lista (1, N), alloggi (0, N)	-
Possiede	L'host possiede degli alloggi	Host (1, N), alloggi (1, N)	-
Scrive	L'utente può scrivere una recensione	Utente (0, N), recensione (1, 1), conclusa (0, 1)	-
Giudica	L'host giudica l'ospite	Ospite (0, N), recensione host (0, 1)	-
Esamina	L'ospite giudica l'host	Host (0, N), recensione ospite (0, 1)	-
Valuta	L'ospite valuta i servizi dell'appartamento	Recensione ospite (0, 1), servizi (0, N)	-
Analizza	L'ospite valuta l'appartamento	Recensione ospite (0, 1), alloggi (0, N)	-
Appartiene	Associa tipi di utente ad utente	Utente (0,1), Host (1,1)	
Appartiene'	Associa tipi di utente ad utente	Utente (0,1), Ospite (1,1)	
Esiste'	Associa tipi di recensione a recensione	Prenotazione(1,1), recensione host(0,1)	
Esiste''	Associa tipi di recensione a recensione	Prenotazione(1,1), recensione ospitef(0,1)	

Vincoli d'integrità

Gli Utenti che vogliono usufruire del servizio devono fornire dati necessari

Un account viene verificato se fornita una foto del documento d'identità in corso di validità

L'identificazione di un metodo di pagamento è necessaria affinché si possa richiedere una prenotazione

Gli Host diventano Superhost se: hanno completato almeno 10 soggiorni, per un totale di almeno 100 notti; hanno conservato un tasso di cancellazione dell'1% massimo; hanno mantenuto una valutazione complessiva del 4,8 considerando tutti i soggiorni in tutte le case di proprietà

Se si conferma la prenotazione viene eseguito il pagamento

La recensione fatta dagli ospiti comprende due testi e una serie di punteggi in una scala da 1 a 5

Le recensioni sono visibili quando entrambi hanno fatto la recensione oppure, se uno dei due non l'ha fatta, 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

Derivazioni:

La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sulle singole dimensioni.

2.5. Schema relazionale

UTENTE(Email, ID_Card, Password, Nome, Cognome, Numero)

OSPITE(ID_ospite, Email)

HOST(ID_host, Email, Superhost)

METODO_PAGAMENTO(Numero carta, Email, Nome titolare, Data scadenza)

ALLOGGI(ID_alloggi, Indirizzo, Nome, Descrizione, Prezzo x1 notte, Costi pulizia, Posti)

LISTA(ID_lista, Email, ID_alloggi, Preferiti_lista)

SERVIZI(ID_servizi, ID_alloggi, Bagno, Cucina, WIFI, Lavatrice)

PRENOTAZIONE(ID_prenotazione, ID_alloggi, Stato_prenotazione, Numero ospiti, Costo totale, Data Check-Out, Data Check-In)

RECENSIONE(ID_recensione, Email)

RECENSIONE_HOST(ID_recensione-host, ID_recensione, ID_ospite, Commento)

RECENSIONE_OSPITE(ID_recensione-ospite, ID_recensione, ID_alloggi, Testo_host, Testo_alloggio)

VALUTAZIONE(ID_valutazione, ID_recensione-ospite, Pulizia, Comunicazione, Posizione, Qualità/prezzo)

Vincoli di integrità referenziale:

Email in OSPITE referencia la chiave di UTENTE

Email in HOST referencia la chiave di UTENTE

Email in METODO_PAGAMENTO referencia la chiave di UTENTE

Email in LISTA referencia la chiave di UTENTE

Email in RECENSIONE referencia la chiave di UTENTE

ID_alloggi in LISTA referencia la chiave di ALLOGGI

ID_alloggi in SERVIZI referencia la chiave di ALLOGGI

ID_alloggi in PRENOTAZIONE referencia la chiave di ALLOGGI

ID_alloggi in RECENSIONE OSPITE referencia la chiave di ALLOGGI

ID_recensione in RECENSIONE_HOST referencia la chiave di RECENSIONE

ID_recensione in RECENSIONE_OSPITE referencia la chiave di RECENSIONE

ID_recensione_ospite in VALUTAZIONE referencia la chiave di RECENSIONE_OSPITE

ID_ospite in RECENSIONE_HOST referencia la chiave di OSPITE

Implementazione

3.1. DDL di creazione del database

```
CREATE TABLE public.utente
```

```
(  
    email character varying COLLATE pg_catalog."default" NOT NULL,  
    id_card character varying COLLATE pg_catalog."default" NOT NULL,  
    password text COLLATE pg_catalog."default" NOT NULL,  
    nome character varying COLLATE pg_catalog."default" NOT NULL,  
    cognome character varying COLLATE pg_catalog."default" NOT NULL,  
    numero character varying COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT utente_pkey PRIMARY KEY (email)  
)
```

```
CREATE TABLE public.ospite
```

```
(  
    id_ospite character varying COLLATE pg_catalog."default" NOT NULL,  
    email character varying COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT ospite_pkey PRIMARY KEY (email),  
    CONSTRAINT ospite_id_ospite_key UNIQUE (id_ospite)  
,  
    CONSTRAINT fk_ospite_utente FOREIGN KEY (email)  
        REFERENCES public.utente (email) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)
```

```
CREATE TABLE public.host
```

```
(  
    id_host character varying COLLATE pg_catalog."default" NOT NULL,  
    email character varying COLLATE pg_catalog."default" NOT NULL,  
    superhost character varying COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT host_pkey PRIMARY KEY (id_host, email),  
    CONSTRAINT fk_host FOREIGN KEY (email)  
        REFERENCES public.utente (email) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)
```

```
CREATE TABLE public.metodo_pagamento
(
    numero_carta character varying(16) COLLATE pg_catalog."default" NOT NULL,
    email character varying COLLATE pg_catalog."default" NOT NULL,
    nome_titolare character varying COLLATE pg_catalog."default" NOT NULL,
    data_scadenza character varying(4) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT numero_carta_pkey PRIMARY KEY (numero_carta, email),
    CONSTRAINT fk_metodo_pagamento FOREIGN KEY (email)
        REFERENCES public.utente (email) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

```
CREATE TABLE public.alloggi
(
    id_alloggi character varying COLLATE pg_catalog."default" NOT NULL,
    indirizzo character varying COLLATE pg_catalog."default" NOT NULL,
    nome character varying COLLATE pg_catalog."default" NOT NULL,
    descrizione character varying COLLATE pg_catalog."default" NOT NULL,
    prezzo_xl_notte character varying COLLATE pg_catalog."default" NOT NULL,
    costi_pulizie character varying COLLATE pg_catalog."default" NOT NULL,
    posti character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT alloggi_pkey PRIMARY KEY (id_alloggi)
)
```

```
CREATE TABLE public.lista
(
    id_lista character varying COLLATE pg_catalog."default" NOT NULL,
    email character varying COLLATE pg_catalog."default" NOT NULL,
    id_alloggi character varying COLLATE pg_catalog."default" NOT NULL,
    preferiti_lista character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT lista_key PRIMARY KEY (id_lista, email, id_alloggi),
    CONSTRAINT fk_lista FOREIGN KEY (email)
        REFERENCES public.utente (email) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT fk_lista_alloggi FOREIGN KEY (id_alloggi)
        REFERENCES public.alloggi (id_alloggi) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)
```

```
CREATE TABLE public.servizi
```

```
(  
    id_servizi character varying COLLATE pg_catalog."default" NOT NULL,  
    id_alloggi character varying COLLATE pg_catalog."default" NOT NULL,  
    preferiti_lista character varying COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT servizi_pkey PRIMARY KEY (id_servizi, id_alloggi),  
    CONSTRAINT fk_servizi FOREIGN KEY (id_alloggi)  
        REFERENCES public.alloggi (id_alloggi) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)
```

```
CREATE TABLE public.prenotazione
```

```
(  
    id_prenotazione character varying COLLATE pg_catalog."default" NOT NULL,  
    id_alloggi character varying COLLATE pg_catalog."default" NOT NULL,  
    stato_prenotazione character varying COLLATE pg_catalog."default" NOT NULL,  
    numero_ospiti integer NOT NULL,  
    costo_totale character varying(30) COLLATE pg_catalog."default" NOT NULL,  
    data_check_out character varying(8) COLLATE pg_catalog."default" NOT NULL,  
    data_check_in character varying(8) COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT prenotazione_pkey PRIMARY KEY (id_prenotazione, id_alloggi),  
    CONSTRAINT fk_prenotazione_alloggi FOREIGN KEY (id_alloggi)  
        REFERENCES public.alloggi (id_alloggi) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)
```

```
CREATE TABLE public.recensione
```

```
(  
    id_recensione character varying COLLATE pg_catalog."default" NOT NULL,  
    email character varying COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT recensione_pkey PRIMARY KEY (id_recensione),  
    CONSTRAINT fk_recensione FOREIGN KEY (email)  
        REFERENCES public.utente (email) MATCH SIMPLE  
        ON UPDATE CASCADE  
        ON DELETE CASCADE  
)
```



```

CREATE TABLE public.recensione_host
(
    id_recensione_host character varying COLLATE pg_catalog."default" NOT NULL,
    id_recensione character varying COLLATE pg_catalog."default" NOT NULL,
    id_ospite character varying COLLATE pg_catalog."default" NOT NULL,
    commento character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT recensione_host_pkey PRIMARY KEY (id_recensione_host, id_recensione, id_ospite),
    CONSTRAINT fk_recensione_host FOREIGN KEY (id_recensione)
        REFERENCES public.recensione (id_recensione) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT fk_recensione_ospite FOREIGN KEY (id_ospite)
        REFERENCES public.ospite (id_ospite) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

CREATE TABLE public.recensione_ospite
(
    id_recensione_ospite character varying COLLATE pg_catalog."default" NOT NULL,
    id_recensione character varying COLLATE pg_catalog."default" NOT NULL,
    id_host character varying COLLATE pg_catalog."default" NOT NULL,
    id_alloggi character varying COLLATE pg_catalog."default" NOT NULL,
    testo_host character varying COLLATE pg_catalog."default" NOT NULL,
    testo_alloggio character varying COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT recensione_ospiti_pkey PRIMARY KEY (id_recensione_ospite, id_recensione, id_host, id_alloggi),
    CONSTRAINT recensione_ospite_id_recensione_ospite_key UNIQUE (id_recensione_ospite)
,
    CONSTRAINT fk_alloggi FOREIGN KEY (id_alloggi)
        REFERENCES public.alloggi (id_alloggi) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE,
    CONSTRAINT fk_recensione_ospite FOREIGN KEY (id_recensione)
        REFERENCES public.recensione (id_recensione) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

CREATE TABLE public.punteggio
(
    id_valutazione character varying COLLATE pg_catalog."default" NOT NULL,
    id_recensione_ospite character varying COLLATE pg_catalog."default" NOT NULL,
    pulizia character varying(5) COLLATE pg_catalog."default" NOT NULL,
    comunicazione character varying(5) COLLATE pg_catalog."default" NOT NULL,
    posizione character varying(5) COLLATE pg_catalog."default" NOT NULL,
    "qualità_prezzo" character varying(5) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT punteggio_pkey PRIMARY KEY (id_valutazione, id_recensione_ospite),
    CONSTRAINT fk_punteggio FOREIGN KEY (id_recensione_ospite)
        REFERENCES public.recensione_ospite (id_recensione_ospite) MATCH SIMPLE
        ON UPDATE CASCADE
        ON DELETE CASCADE
)

```

3.2. DML di popolamento di tutte le tabelle del database

```
INSERT INTO public.utente(  
    email, id_card, password, nome, cognome, numero)  
VALUES ('nicola.trentino@rhyta.com', '2345458905733834', 'nicolatr', 'Nicola', 'Trentino', '3453513892'),  
('LeilaPagnotto@gmail.com', '2649626947284925', 'virgola-01', 'Leila', 'Pagnotto', '3396757948'),  
('BenedettaBuccho@gmail.com', '2346803689375810', 'Casa', 'Benedetta', 'Buccho', '3903394171925'),  
('fiorenzoLombardi@gmail.com', '6388149404863129', 'uf$ksq', 'Fiorenzo', 'Lombardi', '3852757663'),  
('CaioIadanza@rhyta.com', '3249471729490696', 'uZ-ns1%', 'Caio', 'Iadanza', '5972475411'),  
('lucio.endrizzi@teleworm.us', '2737732000172164', 'dc5oHFNV', 'Lucio', 'Endrizzi', '92808701286'),  
('AdrianaCapon@armyspy.com', '1149399132887964', 'nwl#Vdjn', 'Adriana', 'Capon', '9539916937'),  
('AntoninaConti@teleworm.us', '1244593711587356', 'ggC42MNt', 'Antonina', 'Conti', '2808701286'),  
('GregorioSal@armyspy.com', '3410993420506038', '8kkn^kI', 'Gregorio', 'Sal', '4601590606'),  
('SandraBoni@armyspy.com', '1275044245716026', 'lC!s-Vkq', 'Sandra', 'Boni', '55755558005');
```

```
INSERT INTO public.ospite(  
    id_ospite, email)  
VALUES ('nico-trentino', 'nicola.trentino@rhyta.com'),  
('antonica-67', 'AntoninaConti@teleworm.us'),  
('Sandra', 'SandraBoni@armyspy.com');
```

```
INSERT INTO public.host(  
    id_host, email, superhost)  
VALUES ('leila', 'LeilaPagnotto@gmail.com', 'si'),  
('benedetta-buccho', 'BenedettaBuccho@gmail.com', 'no'),  
('fiorenzo', 'fiorenzoLombardi@gmail.com', 'no'),  
('iadanza', 'CaioIadanza@rhyta.com', 'si'),  
('lucio-56', 'lucio.endrizzi@teleworm.us', 'si'),  
('Andry_62', 'AdrianaCapon@armyspy.com', 'no'),  
('Greg_Sal', 'GregorioSal@armyspy.com', 'no');
```

```
INSERT INTO public.metodo_pagamento(  
    numero_carta, email, nome_titolare, data_scadenza)  
VALUES ('2345458905733834', 'nicola.trentino@rhyta.com', 'Nicola Trentino', '0427'),  
('2649626947284925', 'LeilaPagnotto@gmail.com', 'Leila Pagnotto', '1224'),  
('2346803689375810', 'BenedettaBuccho@gmail.com', 'Benedetta Pagnotto', '1026'),  
('6388149404863129', 'fiorenzoLombardi@gmail.com', 'Fiorenzo Lombardi', '1230'),  
('3249471729490696', 'CaioIadanza@rhyta.com', 'Caio Iadanza', '0529'),  
('2737732000172164', 'lucio.endrizzi@teleworm.us', 'Lucio Endrizzi', '1122'),  
('1149399132887964', 'AdrianaCapon@armyspy.com', 'Adriano Capon', '0632'),  
('1244593711587356', 'AntoninaConti@teleworm.us', 'Antonina Conti', '0128'),  
('3410993420506038', 'GregorioSal@armyspy.com', 'Gregorio Sal', '0825'),  
('1275044245716026', 'SandraBoni@armyspy.com', 'Sandra Boni', '0329');
```



```

INSERT INTO public.alloggi(
    id_alloggi, indirizzo, nome, descrizione, prezzo_x1_notte, costi_pulizie, posti)
VALUES ('0101', 'Via Vittorio Veneto 24, Torino', 'Casa Lucia', 'Soggiorno gestito con cura e passione in un open space ristrutturato. Spazio
& Luce per sentirvi a casa, lavorare e visitare la città. Thè, dolcezze per la colazione, libri e video su Torino e un morso al padre della
Nutella: il Gianduio!t!', '47€', '28€', '1 letto matrimoniale'),
('0102', 'Via Vittorio Gioberti 13, Roma', 'Borghetto', 'Camera molto luminosa e confortevole con due letti singoli, perfetta per gli
studenti. Presenti un armadio, connessione internet ad alta velocità, tavolo, una cucina ben attrezzata e un bagno confortevole!', '115€x1
notte', '12€x1 notte', '2 letti singoli'),
('0103', 'Via Andrea Boggio 57, Milano', 'Casa Vittoriana', 'La casa si trova vicino agli ospedali Sant Andrea e Villa San pietro . All uscita
3 del gra , vicino al golf Olgiata', '33€x1 notte', '7€', '1 letto singolo'),
('0104', ' ', 'Via Michele Prado 42, Venezia', 'Sunrise', 'Sunrise offre sistemazione in pieno centro storico a pochi minuti dal ponte di Rialto e
in una zona molto tranquilla di Venezia. Ideale per i viaggiatori amanti di storia e musei e per trascorrere un romantico soggiorno.', '90€x1
notte', '19€', 'Alloggio intero'),
('0105', 'Piazza Ombroso 12b, Firenze', 'Cesare', 'Piccola matrimoniale di un piccolo albergo, tranquilla e lontana dai rumori del traffico.
La camera è dotata di aria condizionata. Bagno dotato di doccia.', '41€x1 notte', '4€', '1 letto matrimoniale'),
('0106', 'Via dei Vecchi 13, Napoli', 'Casa Matilda', 'Rilassati in questo spazio tranquillo in posizione centrale.', '15€x1 notte', '7€', '3
stanze codivise'),
('0107', 'Via delle Rosine 67, Siracusa', 'Casa Paradiso', 'Mini appartamento in zona strategica a 2 passi dal mare e dal centro storico con
parcheggio in struttura', '54€x1 notte', '11€', 'Appartamento intero'),
('0108', 'Via Leopardi 3, Bologna', 'Narciso', 'Ti diamo il benvenuto nella camera vicina al centro città.', '20€x1 notte', '5€', 'Stanza
condivisa');

```

```

INSERT INTO public.lista(
    id_lista, email, id_alloggi, preferiti_lista)
VALUES ('0106', 'nicola.trentino@rhyta.com', '01010', 'Alloggi-Torino'),
('0107', 'GregorioSal@armyspy.com', '0105', 'Alloggi preferiti'),
('0109', 'SandraBoni@armyspy.com', '0103', 'Alloggi Preferito');

```

```

INSERT INTO public.servizi(
    id_servizi, id_alloggi, preferiti_lista)
VALUES ('Bagno, Cucina, WIFI', '0101', 'Alloggi-Torino'),
('Bagno, WIFI', '0102', 'Alloggio Preferito'),
('Bagno, Cucina, WIFI, Piscina', '0103', 'Alloggi-Prefetiti'),
('Bagno, Cucina, WIFI', '0104', 'Alloggi-Torino'),
('Bagno, Cucina', '0105', 'Alloggi-Torino'),
('Bagno, Cucina, Piscina', '0106', 'Alloggi-Torino'),
('Bagno, Cucina, WIFI', '0107', 'Alloggi-Prefetiti'),
('Bagno, WIFI', '0108', 'Alloggi-Prefetiti');

```

```

INSERT INTO public.prenotazione(
    id_prenotazione, id_alloggi, stato_prenotazione, numero_ospiti, costo_totale, data_check_out, data_check_in)
VALUES ('3425', '0101', 'Accettata', '2', '122€', '18/04/23', '16/04/23');
('4567', '0104', 'Rifiutata', '4', '919€', '14/05/23', '24/05/23'),
('9461', '0107', 'In sospeso', '3', '335€', '27/11/22', '2/12/22');

```

```

INSERT INTO public.recensione(
    id_recensione, email)
VALUES ('0105', 'nicola.trentino@rhyta.com')
('0104', 'AntoninaConti@teleworm.us'),
('0103', 'SandraBoni@armyspy.com'),
('0102', 'lucio.endrizzi@teleworm.us'),
('0101', 'fiorenzoLombardi@gmail.com');

```

```

INSERT INTO public.recensione_host(
    id_recensione_host, id_recensione, id_ospite, commento)
VALUES ('0104', '0105', 'antonica-67', 'Ospite cordiale e educata.');
```

```

('0105', '0104', 'Sandra', 'Ospite di tutto rispetto, cura nel mantenere il suo spazio e nessun rumore');
```

```

INSERT INTO public.recensione_ospite(
    id_recensione_ospite, id_recensione, id_host, id_alloggi, testo_host, testo_alloggio)
VALUES ( '0103', '0101', 'lucio-56', '0108', 'Host simpatico, disponibile nelle indicazioni e in consigli.', 'Alloggio vicino al
centro e accogliente.');
```

```

('0106', '0102', 'fiorenzo', '0106', 'Host cordiale e a modo.', 'Alloggi pulito e ordinato, leggermente lontano dal centro ma comodo
per i mezzi.');
```

```

('0105', '0103', 'Andry_62', '0102', 'Host super cordiale e disponibile, senza nessun problema ci ha fornito il doppione delle chiavi.',
'Appartamento elegante e lussuoso, comodo per la stazione.');
```

```

INSERT INTO public.punteggio(
    id_valutazione, id_recensione_ospite, pulizia, comunicazione, posizione, "qualità_prezzo")
VALUES ('05', '0106', '5', '3', '3', '4'),
('04', '0105', '2', '5', '3', '3');
```

3.3 Operazioni di cancellazione e modifica

DELETE FROM utente WHERE email = 'Matty_99@gmail.com'; → **elimina la riga sia dalla tabella email sia dalla tabella ospite**

INSERT INTO public.ospite(id_ospite, email) VALUES ('003', 'emanuela.sardo@gmail.com');

→ **ERROR: insert or update on table "ospite" violates foreign key constraint "fk_ospite_utente"**

DETAIL: Key (email)=(emanuela.sardo@gmail.com) is not present in table "utente". SQL state: 23503

UPDATE public.host SET id_host= 'fiorenzo', email= 'fiorenzoLombardi@gmail.com', superhost='no' WHERE email = 'fiorenzo.lombardi@gmail.com'; → **aggiorna l'attributo sia dalla tabella host sia dalla tabella utente**