



UNIVERSITÀ
DEGLI STUDI
DI TORINO

Progetto di Piattaforma di Home Booking
Laboratorio Basi di Dati 2021/2022

Eduard Antonovic Occhipinti, Iman Solaih, Marco Molica

June 16, 2022

Contents

1	Progettazione Concettuale	2
1.1	Requisiti Iniziali	2
1.2	Glossario dei Termini	4
1.3	Requisiti rivisti e strutturati in gruppi di frasi omogenee	5
1.3.1	Requisiti rivisti	5
1.3.2	Requisiti strutturati in gruppi di frasi omogenee	6
1.4	Schema E-R + Business Rules	10
2	Progettazione Logica	11
2.1	Tavola dei Volumi	11
2.2	Tavola delle Operazioni	12
2.3	Ristrutturazione dello schema E-R	14
2.3.1	Analisi delle ridondanze	14
2.3.2	Eliminazione delle generalizzazioni	17
2.3.3	Partizionamento/accorpamento di entità e associazioni	18
2.3.4	Eliminazione degli attributi composti	19
2.4	Schema E-R ristrutturato + Business Rules	20
2.4.1	Business Rules	21
2.5	Schema Relazionale	23
3	Implementazione	24
3.1	DDL di Creazione dei Database	24
3.2	DML di Popolamento di Tutte le Tabelle del Database	29
3.3	Qualche Operazione di cancellazione e modifica	34
3.3.1	Qualche operazione di sistema	35

Chapter 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 (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.

The screenshot displays the Airbnb mobile app interface for a listing titled "Nice Central studio flat, under the Towers" in Bologna, Emilia-Romagna, Italia. The listing has a 4.79 star rating from 176 reviews. A banner image shows a modern interior. Below the title, it states "È un'occasione unica. Di solito l'alloggio di Armand su Airbnb è prenotato." and includes a "Self check-in" icon. The price is 49€ per notte for the dates 31 mar - 01 apr, with a "Prenota" button. To the right, a section titled "Dove dormirai" shows a "Camera da letto" with 1 letto matrimoniale. Below that, "Cosa troverai" lists amenities like Cucina, Wi-fi, and others. On the far right, a snippet of a review from Mariateresa is visible, mentioning the apartment's location and cleanliness. At the bottom, the host's profile "Host: Armand" is shown, a member since July 2017, with 503 reviews and a 49€ per notte price for the same dates, also featuring a "Prenota" button.

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	Individuo che usufruisce del servizio. Può essere ospitato da altri utenti. Può caricare una carta di identità per diventare un profilo verificato	Ospite	Host Prenotazione Recensione Liste Commento
Host	Utente che offre un servizio ovvero può ospitare altri utenti. Può diventare un superhost soddisfacendo determinati requisiti		Utente Alloggio Prenotazione Recensione Commento
Soggiorno	Utilizzo del servizio da parte di uno o più utenti, con durata variabile		Prenotazione Alloggio
Prenotazione	Richiesta di soggiorno		Alloggio Host Ospite
Alloggio	Proprietà posseduta da un utente. Può essere di diversi tipi	Casa Struttura	Prenotazione Liste Recensione
Recensione	Feedback lasciate tra gli utenti verso altri utenti o proprietà. Comprendono valutazioni e commenti testuali	Review	Utente Commento
Liste	Insiemi di alloggi preferiti da un utente		Alloggi Utente
Commento	Descrizione testuale appartenente a una recensione. Più commenti formano un thread		Recensione Utente
Servizio	Funzionalità messa a disposizione dall'alloggio		Alloggio

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

1.3.1 Requisiti rivisti

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.

Per il dato utente registriamo: indirizzo email, password, nome, cognome, numeri di telefono, carta di identità, metodo di pagamento e data registrazione.

Per il dato host registriamo: superhost.

Per il dato soggiorno registriamo: data inizio, data fine, idalloggio, idprenotazione. Ogni utente può avere 0 o più soggiorni.

Per il dato alloggio registriamo: nome, indirizzo, comune, descrizione, costo per notte per persona, costo pulizia, numero di letti, orario check-in e orario check-out. Ogni host può possedere uno o più alloggi. Ad ogni alloggio sono associate 0 o più foto. Ogni alloggio offre 0 o più servizi. Per ogni alloggio sono scritte 0 o più recensioni.

Gli host per diventare superhost devono soddisfare 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

Ogni utente può aggiungere 0 o più alloggi tra i preferiti e creare 0 più liste. Per il dato lista registriamo: descrizione, nome

Per il dato prenotazione registriamo: costo prenotazione, alloggio, stato, numero ospiti, data prenotazione, data inizio e data fine. Ogni prenotazione sarà associata ad un utente ed un host, potrà avere 0 o più ospiti associati. La prenotazione può essere cancellata sia dall'ospite che dall'host: verrà aggiornato lo stato.

Al termine del soggiorno, gli ospiti e gli host si possono valutare a vicenda. Per il dato recensione registriamo: visibilità, data, autore, corpo, valutazione pulizia, valutazione comunicazione, valutazione posizione, valutazione qualità-prezzo.

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

Per il dato commento registriamo: autore, testo, recensione. Gli host e gli ospiti possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione.

Il sistema 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

1.3.2 Requisiti strutturati in gruppi di frasi omogenee

Frasi relative a utenti:

- Ogni utente può essere sia ospite di altri utenti che host.
- Gli utenti sono rappresentati da un nome, cognome, email, password e uno o più numeri di telefono
- Gli utenti possono memorizzare la foto della carta di identità, necessaria per essere considerati utenti verificati
- Gli utenti possono memorizzare zero o più metodi di pagamento. Per poter effettuare delle prenotazioni è necessario avere almeno un metodo di pagamento
- Gli utenti possono scrivere recensioni con valutazioni verso gli host e gli alloggi
- Gli utenti possono ricevere recensioni dagli host dopo un soggiorno
- Ogni utente può creare delle liste, in cui aggiungere gli alloggi preferiti
- Ogni utente può effettuare una o più prenotazioni, purché abbia un metodo di pagamento. Può confermare o cancellare le proprie prenotazioni
- Ogni utente che ha effettuato una prenotazione, può aggiungere altri utenti ad essa, se registrati al servizio
- Ogni utente può scrivere commenti nei thread delle recensioni in cui è incluso

Frase relative a host:

- Un host è una specializzazione di utente, che può ospitare altri utenti.
- Un host può registrare uno o più alloggi
- Un host può scrivere recensioni testuali verso gli utenti che hanno effettuato un soggiorno presso i propri alloggi
- Un host può diventare un superhost se soddisfa determinati requisiti:
 - Ha completato almeno 10 soggiorni, per un totale di almeno 100 notti
 - Ha conservato un tasso di cancellazione dell'1
 - Ha mantenuto una valutazione complessiva di 4.8 tra tutti i soggiorni di tutti i suoi alloggi
- Un host può cancellare o confermare una prenotazione ricevuta

Frase relative a soggiorno:

- Ogni soggiorno viene effettuato al seguito di una prenotazione confermata.
- Ogni soggiorno è caratterizzato da una data inizio e una data fine
- Ogni soggiorno viene effettuato in un alloggio, da uno o più utenti
- Per ogni soggiorno è richiesta una prenotazione
- Ogni soggiorno può ricevere una recensione, con valutazione
- La valutazione complessiva di un soggiorno è la media delle sue valutazioni

Frase relative a prenotazione:

- Ogni prenotazione è associata ad un alloggio
- Ogni prenotazione è caratterizzata da un costo totale, una data di prenotazione, una data di inizio, una data di fine
- Ogni prenotazione deve essere confermata o rifiutata dagli host
- Può essere cancellata dagli utenti o dagli host
- Non è possibile effettuare una prenotazione se l'utente non ha caricato un metodo di pagamento
- Quando confermata viene eseguito il pagamento
- Ogni prenotazione confermata è associata ad un soggiorno

Frase relative a Alloggio:

- Ogni alloggio è caratterizzato da un nome, un comune, un indirizzo, un costo per notte per persona, un costo pulizia, un numero di letti, orario del check-in, orario del check-out
- Ogni alloggio può avere 0 o più recensioni
- Ogni alloggio deve appartenere ad un host
- Più alloggi possono appartenere allo stesso host
- Un alloggio può essere di diversi tipi: intero appartamento, stanza privata o stanza condivisa
- Un alloggio di tipo stanza condivisa può ricevere prenotazioni per la stessa data fino al raggiungimento del numero di letti
- Un alloggio di tipo stanza privata o appartamento non può ricevere più prenotazioni per la stessa data
- Un alloggio può avere una o più foto
- Un alloggio può ricevere delle recensioni
- La valutazione complessiva di un alloggio è la media delle valutazioni ricevute
- Un alloggio può essere aggiunto tra i preferiti di zero o più utenti
- Un alloggio può offrire zero o più servizi

Frase relative a Recensione:

- Ogni recensione è caratterizzata da un corpo, una data e delle valutazioni
- Ogni recensione è associata ad un soggiorno
- Le recensioni sono di 3 tipi: fatte ad host, fatte ad utenti e fatte ad alloggio
- Una recensione su alloggio può contenere una valutazione della pulizia, una valutazione della posizione e una valutazione sul rapporto qualità-prezzo
- Una recensione su host può contenere una valutazione sulla comunicazione
- Una recensione su utenti non contiene valutazioni
- Ad ogni recensione può essere associato un thread di commenti
- Una recensione può essere lasciata da un host o da un utente
- Una recensione può essere visibile o non visibile

- Una recensione diventa visibile se sia l'autore che il recensito hanno pubblicato una recensione
- Una recensione diventa automaticamente visibile dopo 7 giorni dalla fine del soggiorno a cui è associata

Frase relative a Liste:

- Ogni lista è caratterizzata da un nome e una descrizione
- Una lista può contenere zero o più alloggi preferiti di un utente
- Ogni lista appartiene ad un solo utente

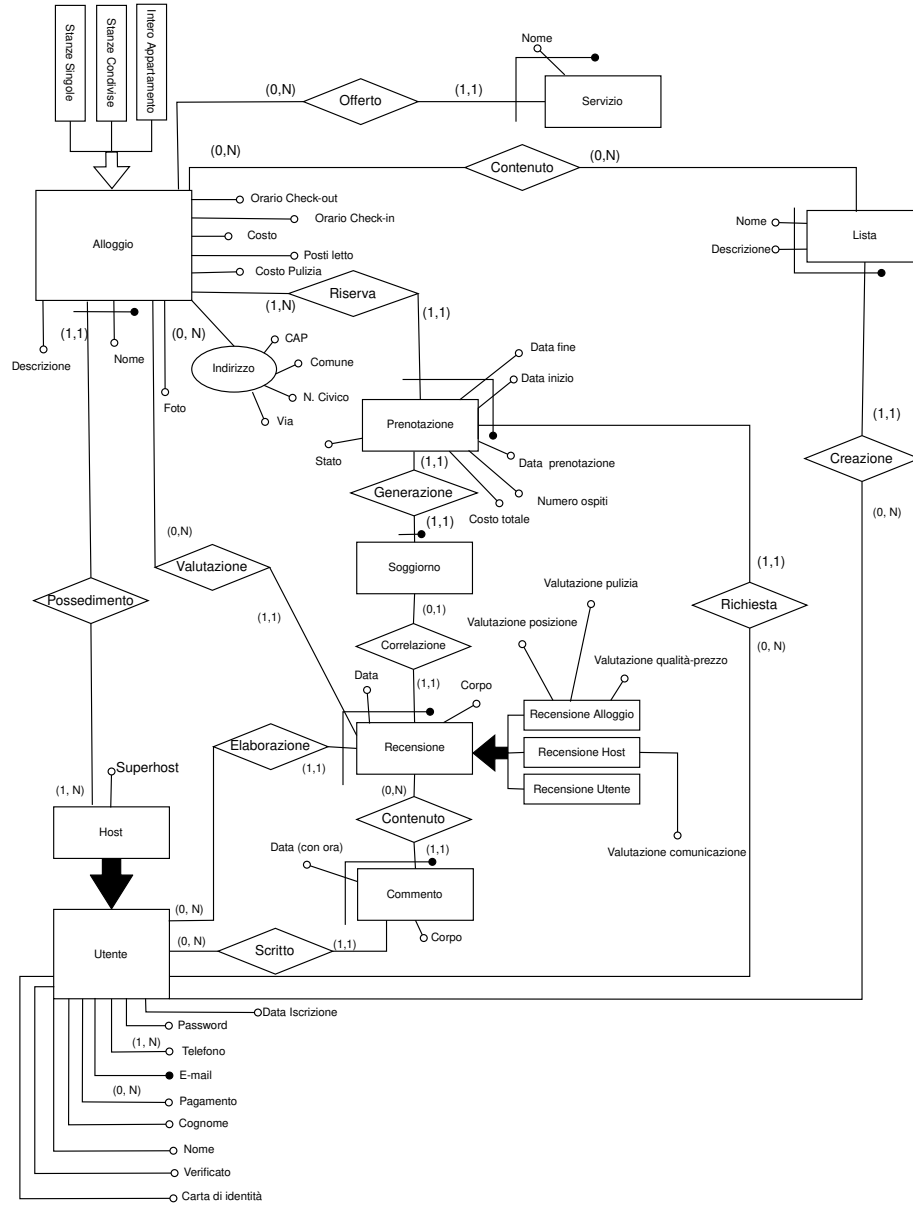
Frase relative a Commento:

- Ogni commento è caratterizzato da un testo
- Ogni commento può essere scritto nel thread di una recensione
- Ogni commento può essere scritto solo dall'utente o dall'host coinvolti nella recensione

Frase relative a Servizio:

- Un servizio può essere di più tipologie (tv, wifi, cucina, lavatrice, etc..)

1.4 Schema E-R + Business Rules



Chapter 2

Progettazione Logica

2.1 Tavola dei Volumi

Concetto	Tipo	Volume	Motivazione
Utente	E	30.000.000	Ipotizziamo una piattaforma in cui sono iscritte 30 milioni di utenti
Host	E	150.000	Ipotizziamo che sulla piattaforma si iscriveranno circa 150 mila host
Alloggio	E	169.000	Ipotizziamo che nella piattaforma verranno registrati circa 169 mila alloggi
Prenotazione	E	36.000.000	Ipotizziamo che sulla piattaforma siano state effettuate circa 36 milioni di prenotazioni
Soggiorno	E	34.920.000	Ipotizziamo che sulla piattaforma ci siano stati circa 35 milioni di soggiorni
Recensione	E	12.000.000	Ipotizziamo che sulla piattaforma vengano scritte circa 12 milioni di recensioni
Commento	E	16.000.000	Ipotizziamo che sulla piattaforma vengano scritti circa 16 milioni di commenti
Lista	E	45.000.000	Ipotizziamo che sulla piattaforma vengano create circa 45 milioni di liste di alloggi preferiti
Servizio	E	20	Ipotizziamo che sulla piattaforma vengano messi a disposizione circa 20 servizi differenti
Possedimento	R	169.000	Ipotizziamo che nella piattaforma ogni host abbia almeno un alloggio, e che 1 host su 8 abbia 2-3 alloggi
Continua all pagina successiva			

Concetto	Tipo	Volume	Motivazione
Richiesta	R	36.000.000	Ipotizziamo che nella piattaforma 4 utenti registrati su 5 abbiano fatto almeno una prenotazione, e 1 su 5 ne abbia fatto almeno 3
Generazione	R	34.920.000	Ipotizziamo che sul totale delle prenotazioni, circa il 2% vengano cancellate. Tutte le altre diventano soggiorni effettivi
Elaborazione	R	12.000.000	Ipotizziamo che 1 utente su 3 che ha effettuato una prenotazione poi scriva una recensione
Contenuto	R	16.000.000	Ipotizziamo che circa 1 recensione su 3 abbia un thread con almeno 3 commenti e 1 su 3 abbia un solo commento
Creazione	R	45.000.000	Ipotizziamo che circa 6 utenti su 10 creino delle liste, con una media di 2-3 liste per ciascuno di questi utenti
Scritto	R	16.000.000	Ipotizziamo che circa 1 utente su 5 abbia scritto un commento, e di questi uno ne abbia scritto circa 2-3
Correlazione	R	12.000.000	Ipotizziamo che circa 1 soggiorno su 6 riceva una recensione da parte dell'utente o dell'host, e che 1 su 6 la riceva da parte di entrambi
Riserva	R	36.000.000	Ipotizziamo che tutti gli alloggi vengano riservati circa 36 milioni di volte, una volta per ogni prenotazione
Offerto	R	250.000	Ipotizziamo che ogni alloggio offra più di una decina di servizi
Valutazione	R	2.000.000	Ipotizziamo che circa 1 recensione su 3 viene scritta verso un alloggio

2.2 Tavola delle Operazioni

#	Operazione	Tipo	Frequenza	Motivazione
1	Registrazione al servizio	I	50.000/day	Operazione iniziale della piattaforma che permette ad una persona di utilizzare il servizio
2	Prenotazione di un alloggio	I	25.000/day	Operazione principale che permette di effettuare un soggiorno
Continua alla pagina successiva				

#	Operazione	Tipo	Frequenza	Motivazione
3	Conferma di una prenotazione	I	24.500/day	Operazione che permette di finalizzare la richiesta di una prenotazione, effettuando il pagamento
4	Declino di una prenotazione	I	500/day	Operazione che permette di rifiutare una prenotazione richiesta
5	Scrittura di una recensione	I	8000/day	Operazione che permette di lasciare un feedback testuale da parte di un host e/o di un utente, e una valutazione da parte di un utente
6	Scrittura di un commento	I	13000/day	Operazione che permette ad un host e/o un utente di lasciare ulteriori feedback testuali ad una recensione
7	Creazione di una lista di alloggi preferiti	I	7000/day	Operazione che permette di creare dei raggruppamenti di alloggi preferiti
8	Registrazione di un nuovo alloggio	I	300/day	Operazione che permette di inserire un nuovo alloggio nella piattaforma
9	Aggiornamento del tasso di cancellazione di ciascun host	B	1/week	Operazione che una volta a settimana aggiorna il numero di prenotazioni cancellate da parte dell'host
10	Calcolo del numero di soggiorni e tempo di soggiorno complessivo per ciascun host	B	1/day	Operazione che permette di aggiornare il tempo e il numero di soggiorni complessivi per ogni host, in modo da poterne calcolare lo status di superhost
11	Calcolo della valutazione complessiva di ogni host per i soggiorni totali ai propri alloggi	B	1/day	Operazione che permette di aggiornare la media delle valutazioni ricevute per ogni host, in modo da poterne calcolare lo status di superhost
12	Aggiornamento dello status di ciascun host	B	1/day	Operazione che tenendo conto di altri calcoli effettua un aggiornamento per definire quali host sono considerati superhost
13	Aggiornamento della classifica degli alloggi più graditi	B	1/month	Operazione che permette di ottenere gli alloggi con una valutazione media più alta
Continua alla pagina successiva				

#	Operazione	Tipo	Frequenza	Motivazione
14	Aggiornamento della visibilità delle recensioni	B	1/day	Operazione che permette di aggiornare lo status di visibilità delle recensioni
15	Aggiornamento dello stato soggiorno delle prenotazioni	B	1/day	Operazione che permette di aggiornare di aggiornare l'attributo soggiorno delle prenotazioni confermate

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

Relazione VALUTAZIONE

All'interno dello schema ER è stata identificata 1 ridondanza: la relazione valutazione tra le entità alloggio e recensione.

Questa ridondanza ci permette di ottenere le recensioni effettuate su un alloggio utilizzando solamente le entità alloggio e recensione. La sezione della tavola dei volumi di interesse è:

Concetto	Tipo	Volume	Motivazione
Alloggio	E	169.000	Ipotizziamo che nella piattaforma verranno registrati circa 169 mila alloggi
Prenotazione	E	36.000.000	Ipotizziamo che sulla piattaforma siano state effettuate circa 36 milioni di prenotazioni
Soggiorno	E	34.920.000	Ipotizziamo che sulla piattaforma ci siano stati circa 35 milioni di soggiorni
Recensione	E	12.000.000	Ipotizziamo che sulla piattaforma vengano scritte circa 12 milioni di recensioni
Generazione	R	34.920.000	Ipotizziamo che sul totale delle prenotazioni, circa il 2% vengano cancellate. Tutte le altre diventano soggiorni effettivi
Correlazione	R	12.000.000	Ipotizziamo che circa 1 soggiorno su 6 riceva una recensione da parte dell'utente o dell'host, e che 1 su 6 la riceva da parte di entrambi
Riserva	R	36.000.000	Ipotizziamo che tutti gli alloggi vengano riservati circa 36 milioni di volte, una volta per ogni prenotazione
Valutazione	R	2.000.000	Ipotizziamo che circa 1 recensione su 3 viene scritta verso un alloggio

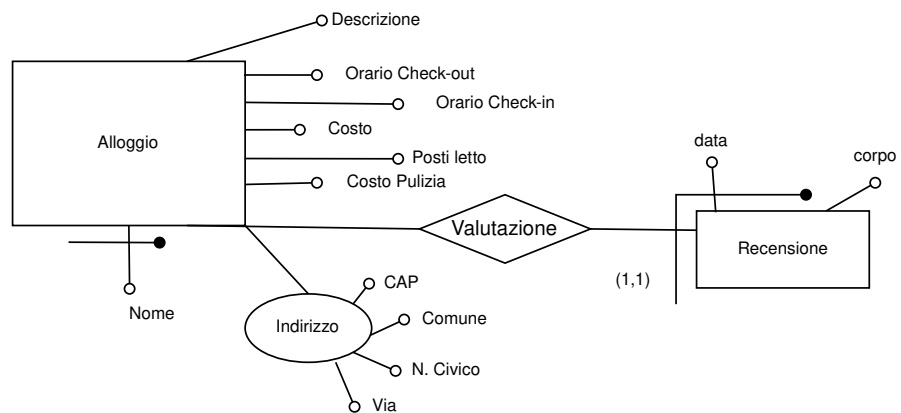


Figure 2.1: In presenza di ridondanza

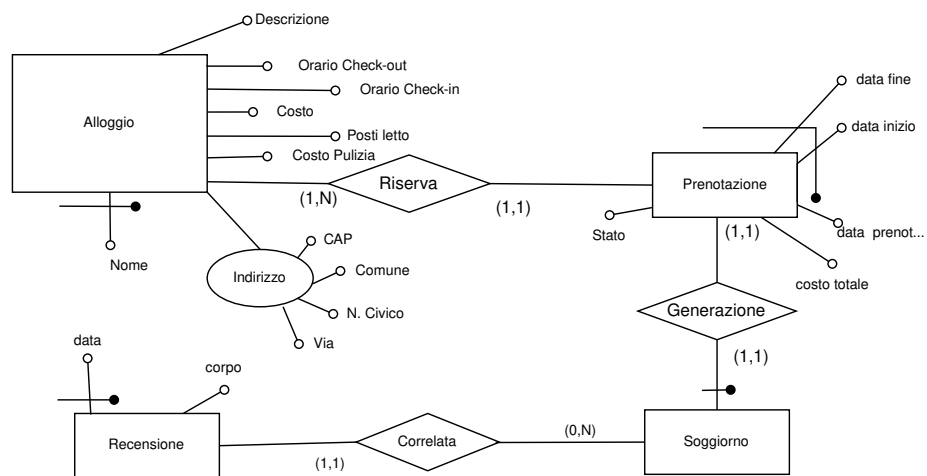


Figure 2.2: Senza la ridondanza

Tavola degli accessi

Analizziamo l'operazione "Scrittura di una recensione su un alloggio (3000 volte al giorno)"

In presenza di ridondanza

Concetto	Costrutto	Accessi	Tipo
Alloggio	E	1	S
Alloggio	E	1	L
Recensione	E	1	S
Valutazione	R	1	S

Senza la ridondanza

Concetto	Costrutto	Accessi	Tipo
Alloggio	E	1	L
Riserva	R	1	L
Prenotazione	E	1	L
Generazione	R	1	L
Soggiorno	E	1	S
Soggiorno	E	1	L
Correlazione	R	1	S
Recensione	E	1	S

Analisi di complessità in presenza di ridondanza:

- In termini di tempo, vengono effettuati un accesso in lettura e tre accessi in scrittura, quindi $3000 + 3000 * 3 * 2$ (contiamo doppi gli accessi in scrittura), per un totale di 21 mila accessi.
- In termini di spazio, viene aggiunta una relazione in cui si memorizzano i dati chiave dell'alloggio all'interno dell'entità recensione: ipotizziamo quindi 200 byte per ogni recensione scritta. Considerando 200 byte per 2.000.000 di recensioni totali, il costo totale in termini di spazio risulta essere $200 * 2.000.000$ (381.47Mb).

Analisi di complessità in assenza di ridondanza:

- In In termini di spazio, il costo totale è 0 byte.
- In termini di tempo, vengono effettuati tre accessi in scrittura e cinque accessi in lettura, quindi $3000 * 5 + 3000 * 3 * 2$ (contiamo doppi gli accessi in scrittura), per un totale di 33 mila accessi.

Dall'analisi effettuata, con l'assenza di ridondanza, risulta un peggioramento nei tempi di accesso (circa il 35% di tempo in più) ma un risparmio notevole in termini di spreco di memoria: decidiamo per cui di rimuovere la ridondanza.

2.3.2 Eliminazione delle generalizzazioni

Entità RECENSIONE

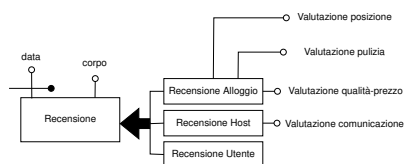


Figure 2.3: Prima

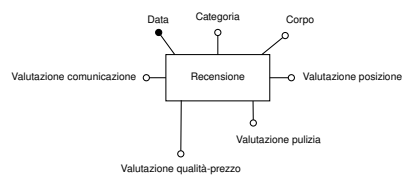


Figure 2.4: Dopo

La generalizzazione è di tipo **totale ed esclusiva**. La decisione consiste nel raggruppamento delle entità figlie nell'attributo categoria. Gli attributi specifici delle entità figlie (valutazioni) vengono spostate nell'entità padre, diventando annullabili. L'attributo categoria sarà un valore not null.

Entità ALLOGGIO

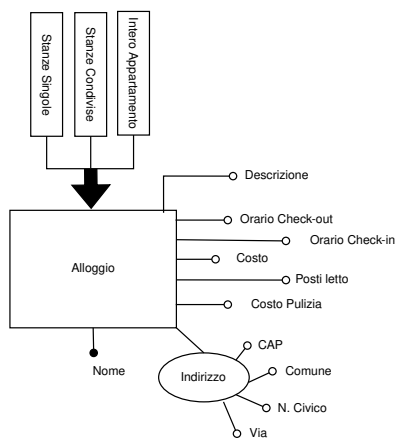


Figure 2.5: Prima

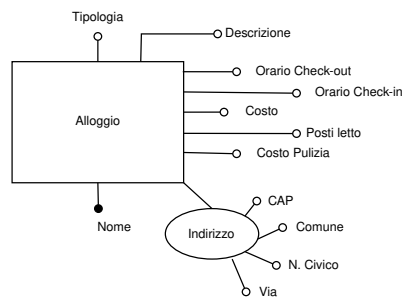


Figure 2.6: Dopo

La generalizzazione è di tipo **totale ed esclusiva**. La decisione consiste nel raggruppamento delle entità figlie nell'attributo tipologia, con valore not null.

Entità UTENTE

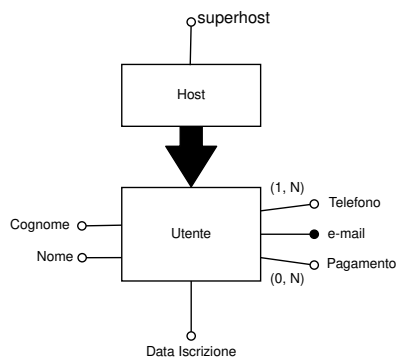


Figure 2.7: Prima

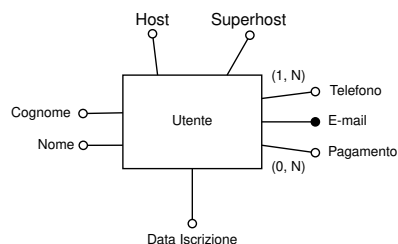


Figure 2.8: Dopo

La generalizzazione è di tipo **totale ed esclusiva**. La decisione consiste nel raggruppamento dell'entità figlia nell'attributo host, con valore not null. L'attributo superhost dell'entità figlia viene trasferito al padre.

2.3.3 Partizionamento/accorpamento di entità e associazioni

Accorpamento Entità PRENOTAZIONE e SOGGIORNO

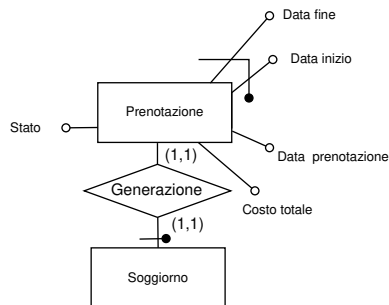


Figure 2.9: Prima

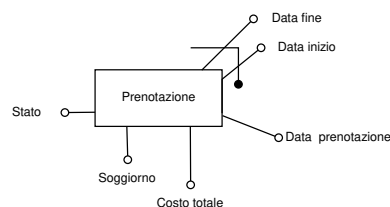


Figure 2.10: Dopo

La decisione di accorpare le entità Prenotazione e Soggiorno in un'unica entità con attributo soggiorno (di tipo booleano) deriva dal fatto che l'entità Soggiorno viene generata dalle prenotazioni con stato "confermata".

Partizionamento Entità UTENTE

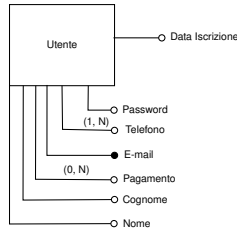


Figure 2.11: Prima

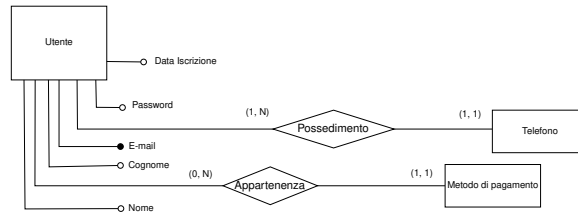


Figure 2.12: Dopo

Decidiamo di partizionare l'entità Utente estraendo gli attributi telefono e pagamento, facendoli diventare rispettivamente una nuova entità Telefono, associata a Utente tramite la relazione Possedimento, e una nuova entità Pagamento, associata a Utente tramite la relazione Appartenenza.

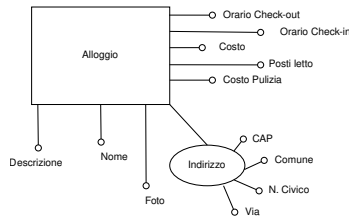


Figure 2.13: Prima

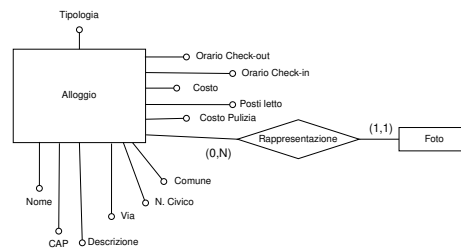


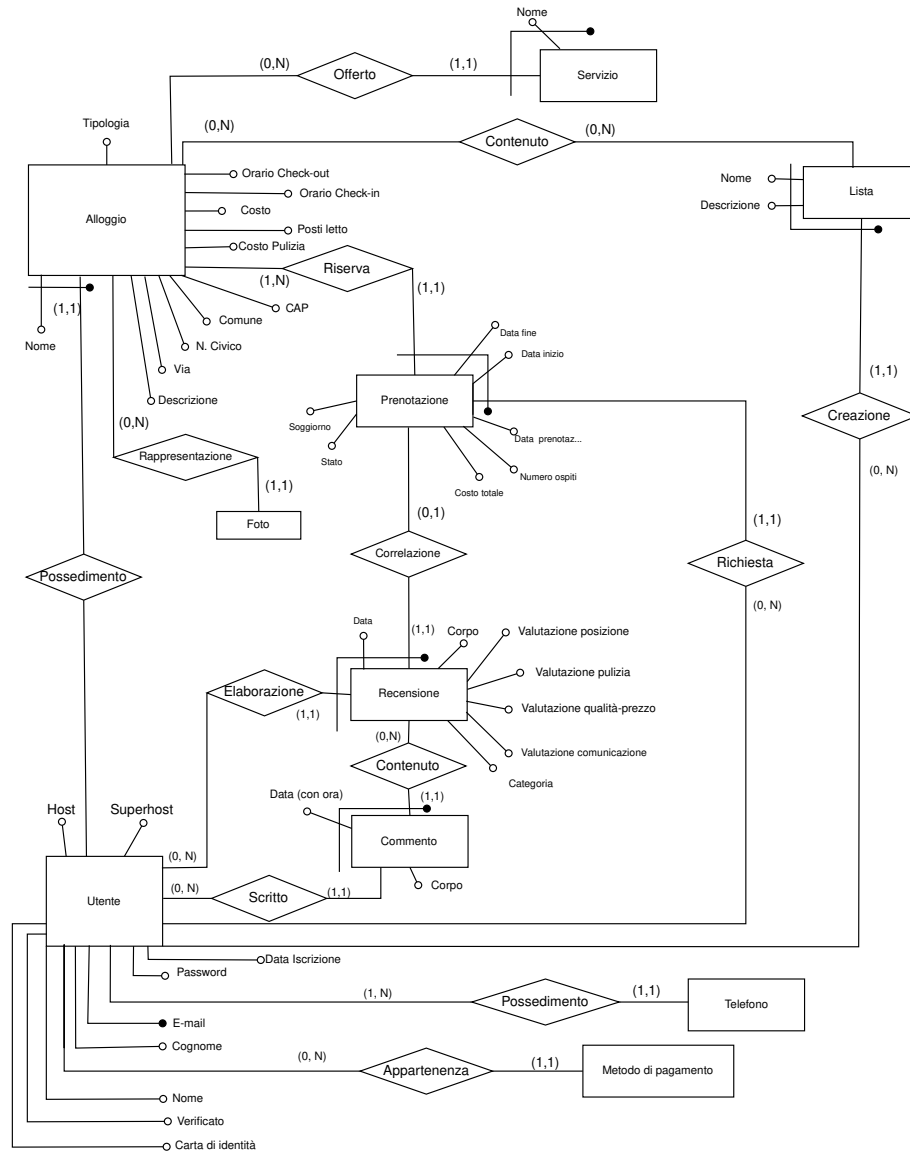
Figure 2.14: Dopo

Decidiamo di partizionare ulteriormente l'entità Utente estraendo l'attributo foto, facendolo diventare una nuova entità Foto, associata a Utente tramite la relazione Rappresentazione.

2.3.4 Eliminazione degli attributi composti

L'attributo composto “indirizzo” viene eliminato, considerando i suoi componenti come attributi semplici. Nel caso di indirizzo abbiamo: via, numero civico, cap e comune. Tale eliminazione viene effettuato nell'entità Alloggio

2.4 Schema E-R ristrutturato + Business Rules



2.4.1 Business Rules

Derivate dal testo

- Un utente non può prenotare se non ha inserito un metodo di pagamento valido
- Un host per diventare superhost deve aver completato almeno 10 soggiorni per un totale di almeno 100 notti, aver conservato un tasso di cancellazione dell'1% e avere una valutazione complessiva di 4.8
- Un utente può visualizzare l'indirizzo di un alloggio solamente dopo aver effettuato una prenotazione con avvenuta conferma da parte dell'host
- Un utente non può indicare in una prenotazione nomi di altri ospiti, se questi non sono iscritti alla piattaforma
- Un utente può valutare un host, e viceversa, solamente al termine del soggiorno soggetto alla valutazione
- Un utente non può cancellare una prenotazione dopo l'inizio di un soggiorno
- Un host non può cancellare una prenotazione dopo l'inizio di un soggiorno
- Una recensione da parte di un host risulta visibile solamente quando il soggiorno a cui fa riferimento riceve la recensione da parte dell'utente, e viceversa
- Una recensione risulta visibile trascorsi 7 giorni dal termine del soggiorno a cui fa riferimento
- Le recensioni sono visibili sui profili degli utenti suddivise in base a quelle ricevute come ospite e come host

Introdotte

- L'indirizzo e-mail di un utente che effettua la registrazione deve corrispondere al formato e-mail (nomemail@dominio.TLD)
- Un utente viene ritenuto host se possiede almeno un alloggio
- Un utente può cancellare solamente le proprie prenotazioni effettuate
- Un host può confermare o rifiutare solamente le prenotazioni effettuate ai propri alloggi
- Le cancellazioni delle prenotazioni da parte degli utenti non influisce sul tasso di cancellazione di un host
- Un utente non può indicare, in una prenotazione, un numero di ospiti maggiore dei posti disponibili nell'alloggio prenotato

- Un utente non può effettuare una prenotazione presso un alloggio di tipo stanza condivisa se il numero di posti letto ancora disponibili non è sufficiente
- Un utente non può valutare soggiorni non prenotati da lui
- Un host non può valutare ospiti che non hanno effettuato soggiorni presso i proprio alloggi
- Un utente non può scrivere commenti sotto recensioni altrui
- Un host non può scrivere commenti sotto recensioni altrui
- Un utente non può effettuare due recensioni per lo stesso soggiorno
- Una prenotazione non può essere effettuata presso un alloggio di tipo stanza singola o appartamento se la data di inizio o di fine intercorre tra la data di inizio e di fine di un'altra prenotazione presso lo stesso alloggio
- Una prenotazione non può essere cancellata completamente dal sistema: quando una prenotazione viene cancellata, viene aggiornato il suo stato a "cancellata"
- Una prenotazione non può essere cancellata se la data di inizio è antecedente alla data odierna di oggi
- Quando un alloggio viene rimosso, lo stato delle sue prenotazioni con soggiorno false viene aggiornato a "cancellata"
- Quando un host viene cancellato, lo stato delle prenotazioni in corso associate ai suoi alloggi viene aggiornato a "cancellata"
- Quando un utente viene cancellato, le recensioni effettuate risulteranno scritte da "utenti cancellato"

2.5 Schema Relazionale

- Utente (Email, Nome, Cognome, Password, Host, Superhost, Verificato, Carta di identità)
- Telefono (Utente, Numero, Prefisso)
- Pagamento (Utente, Numero, Circuito)
- Utente (Email, Nome, Cognome, Password, Host, Superhost)
- Alloggio (ID Alloggio, Nome, Host, Descrizione, Tipologia, Orario check-in, Orario check-out, Costo, Costo pulizia, Posti letto, CAP, Comune, Civico, Via) Alloggio (Host) referencia Utente (Email)
- Foto (Path, Alloggio) Alloggio (Foto) referencia Alloggio (ID Alloggio)
- Prenotazione (ID Prenotazione, Richiedente, ID Alloggio, Data inizio, Data fine, Soggiorno, Stato, Numero ospiti) Prenotazione (ID Alloggio) referencia Alloggio (ID Alloggio) Prenotazione (Richiedente) referencia Utente (Email)
- Recensione (ID Recensione, Autore, ID Prenotazione, Data, Categoria, Valutazione posizione, Valutazione pulizia, Valutazione qualità-prezzo, Valutazione comunicazione, Corpo) Recensione (Autore) referencia Utente (Email) Recensione (ID Prenotazione) referencia Prenotazione (ID Prenotazione)
- Commento (ID Commento, Data, ID Recensione, Autore, Corpo) Commento (Autore) referencia Utente (Email) Commento (ID Recensione) referencia Recensione (ID Recensione)
- Lista (ID Lista, Nome, Descrizione, Autore) Lista (Autore) referencia Utente (Email)
- Contenuto(ID Lista, ID Alloggio) Contenuto (ID Lista) referencia Lista (ID Lista) Contenuto (ID Alloggio) referencia Alloggio (ID Alloggio)

Chapter 3

Implementazione

Riportiamo in seguito alcune query significative per il nostro database

3.1 DDL di Creazione dei Database

```
1  -- @block initialize the database
2  -- CREATE DATABASE home_booking;
3  DROP TABLE IF EXISTS Utente,
4      Telefono,
5      Pagamento,
6      Prenotazione,
7      Alloggio,
8      Recensione,
9      Commento,
10     Lista,
11     Contenuto,
12     Servizio,
13     Foto;
14  DROP TYPE IF EXISTS StatoPrenotazione,
15     CategoriaRecensione,
16     TipologiaAlloggio;
17
18
19  -- @block initialize the enums
20  CREATE TYPE StatoPrenotazione AS ENUM (
21      'Prenotato',
22      'Confermato',
23      'Rifiutato',
24      'Cancellato',
25      'Cancellato dall" host'
26  );
27  CREATE TYPE TipologiaAlloggio AS ENUM (
28      'Appartamento',
29      'Stanza singola',
30      'Stanza condivisa'
31  );
32  CREATE TYPE CategoriaRecensione AS ENUM ('Host', 'Utente', 'Alloggio');
```

```

1  -- @block initialize Utente
2  -- Superhost: Non si può essere superhost se non si è host
3  -- Verificato: Non si può essere verificati se non si ha la carta di identità
4  CREATE TABLE Utente (
5      Nome VARCHAR(45) NOT NULL,
6      Cognome VARCHAR(45) NOT NULL,
7      Password VARCHAR(40) NOT NULL,
8      Email VARCHAR(45),
9      DataRegistrazione TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
10     Host BOOLEAN DEFAULT FALSE NOT NULL,
11     Superhost BOOLEAN DEFAULT FALSE NOT NULL,
12     Verificato BOOLEAN DEFAULT FALSE NOT NULL,
13     CartaIdentita VARCHAR(30),
14     PRIMARY KEY (Email),
15     Constraint user_valid_host CHECK (
16         (
17             Superhost = TRUE
18             AND Host = TRUE
19         )
20         OR Superhost = FALSE
21     ),
22     Constraint user_valid_verified CHECK (
23         (
24             Verificato = TRUE
25             AND CartaIdentita IS NOT NULL
26         )
27         OR Verificato = FALSE
28     ),
29     Constraint user_valid_email CHECK (
30         email ~* '^[A-Za-z0-9._+%-]+@[A-Za-z0-9.-]+[.][A-Za-z]+$'
31     )
32 );

1  -- @block initialize Telefono
2  CREATE TABLE Telefono (
3      Utente VARCHAR(45) NOT NULL,
4      Numero VARCHAR(15) NOT NULL,
5      Prefisso VARCHAR(5) NOT NULL,
6      FOREIGN KEY (Utente) REFERENCES Utente(Email) ON DELETE CASCADE ON
          UPDATE CASCADE,
7      PRIMARY KEY (Utente, Numero)
8  );

1  -- @block initialize Pagamento
2  CREATE TABLE Pagamento (
3      Utente VARCHAR(45) NOT NULL,
4      Numero VARCHAR(16) NOT NULL CHECK (char_length(Numero) = 16),
5      Prefisso VARCHAR(30) NOT NULL,
6      FOREIGN KEY (Utente) REFERENCES Utente(Email) ON DELETE CASCADE ON
          UPDATE CASCADE,
7      PRIMARY KEY (Utente, Numero)
8  );

```

```

1
2 -- @block initialize Alloggio
3 -- Nome: Si potrebbe creare un indice su Nome per velocizzare le operazioni
4 -- di ricerca sull'alloggio
5 -- Host: Può essere NULL per supportare le casistiche in cui un host è stato
6 -- cancellato dal sistema
7
8 CREATE TABLE Alloggio (
9     IDAlloggio UUID NOT NULL DEFAULT gen_random_uuid(),
10     Nome VARCHAR(45) NOT NULL,
11     Host VARCHAR(45),
12     Descrizione TEXT,
13     Tipologia TipologiaAlloggio NOT NULL,
14     OrarioCheckIn TIME NOT NULL,
15     OrarioCheckOut TIME NOT NULL,
16     Costo NUMERIC NOT NULL CHECK (Costo > 0),
17     CostoPulizia NUMERIC DEFAULT 0 CHECK (CostoPulizia >= 0),
18     CAP INT NOT NULL CHECK (CAP > 0),
19     Comune VARCHAR(45) NOT NULL,
20     Civico INT NOT NULL CHECK (Civico > 0),
21     Via VARCHAR(45) NOT NULL,
22     FOREIGN KEY (Host) REFERENCES Utente(Email) ON DELETE
23     SET NULL ON UPDATE CASCADE,
24     PRIMARY KEY (IDAlloggio)
25 );

```

```

1
2 -- @block initialize Prenotazione
3 -- IDAlloggio: Può essere NULL per supportare le casistiche in cui un
4 -- alloggio è stato cancellato dal sistema
5 -- Il check su DataInizio > CURRENT_TIMESTAMP potrebbe non permettere
6 -- inserimenti massivi futuri da parte di un DBA,
7 -- in caso di manutenzione: meglio evitare questo tipo di controllo
8
9 CREATE TABLE Prenotazione (
10     IDPrenotazione UUID NOT NULL DEFAULT gen_random_uuid(),
11     IDAlloggio UUID,
12     Richiedente VARCHAR(45) NOT NULL,
13     DataInizio TIMESTAMP NOT NULL,
14     DataFine TIMESTAMP NOT NULL,
15     Soggiorno BOOLEAN DEFAULT FALSE NOT NULL,
16     NumeroOspiti INT NOT NULL CHECK (NumeroOspiti >= 0),
17     Stato StatoPrenotazione NOT NULL DEFAULT 'Prenotato',
18     Visibile BOOLEAN DEFAULT FALSE NOT NULL,
19     FOREIGN KEY (IDAlloggio) REFERENCES Alloggio(IDAlloggio) ON DELETE
20     SET NULL,
21     FOREIGN KEY (Richiedente) REFERENCES Utente(Email) ON DELETE CASCADE
22     ON UPDATE CASCADE,
23     PRIMARY KEY (IDPrenotazione),
24     Constraint prenotazione_start_before_end CHECK (DataInizio <
25     DataFine) - - CHECK (DataInizio > CURRENT_TIMESTAMP)
26 );

```

```

1
2 -- @block initialize Recensione
3 -- IDPrenotazione: Una prenotazione non può essere cancellata dal sistema, a
  meno di attività manuali sul db
4 CREATE TABLE Recensione (
5     IDRecensione UUID NOT NULL DEFAULT gen_random_uuid(),
6     IDPrenotazione UUID NOT NULL,
7     Autore VARCHAR(45) NOT NULL,
8     DataRecensione DATE NOT NULL,
9     ValutazionePosizione INT CHECK(
10         ValutazionePosizione >= 0
11         AND ValutazionePosizione <= 5
12     ),
13     ValutazionePulizia INT CHECK(
14         ValutazionePulizia >= 0
15         AND ValutazionePulizia <= 5
16     ),
17     ValutazioneQualitaPrezzo INT CHECK(
18         ValutazioneQualitaPrezzo >= 0
19         AND ValutazioneQualitaPrezzo <= 5
20     ),
21     ValutazioneComunicazione INT CHECK(
22         ValutazioneComunicazione >= 0
23         AND ValutazioneComunicazione <= 5
24     ),
25     Corpo TEXT,
26     Categoria TEXT NOT NULL,
27     FOREIGN KEY (IDPrenotazione) REFERENCES Prenotazione(IDPrenotazione) ON
        DELETE CASCADE,
28     FOREIGN KEY (Autore) REFERENCES Utente(Email) ON DELETE
29     SET NULL ON UPDATE CASCADE,
30     PRIMARY KEY (IDRecensione),
31     Constraint recensione_valid_category CHECK (
32         (
33             Categoria = 'Host'
34             AND ValutazionePosizione IS NULL
35             AND ValutazionePulizia IS NULL
36             AND ValutazioneQualitaPrezzo IS NULL
37         )
38         OR (
39             Categoria = 'Utente'
40             AND ValutazionePosizione IS NULL
41             AND ValutazionePulizia IS NULL
42             AND ValutazioneQualitaPrezzo IS NULL
43             AND ValutazioneComunicazione IS NULL
44         )
45         OR (
46             Categoria = 'Alloggio'
47             AND ValutazioneComunicazione IS NULL
48         )
49     )
50 );

```

```

1
2 -- @block initialize Commento
3 CREATE TABLE Commento (
4     IDCommento UUID NOT NULL DEFAULT gen_random_uuid(),
5     IDRecensione UUID NOT NULL,
6     DataCommento TIMESTAMP NOT NULL,
7     Autore VARCHAR(45) NOT NULL,
8     Corpo TEXT,
9     FOREIGN KEY (IDRecensione) REFERENCES Recensione(IDRecensione) ON DELETE
10     CASCADE,
11     FOREIGN KEY (Autore) REFERENCES Utente(Email) ON DELETE
12     SET NULL ON UPDATE CASCADE,
13     PRIMARY KEY (IDCommento)
14 );

```

```

1
2 -- @block initialize Lista
3 CREATE TABLE Lista (
4     IDLista UUID NOT NULL DEFAULT gen_random_uuid(),
5     Nome VARCHAR(45) NOT NULL,
6     Descrizione TEXT,
7     Autore VARCHAR(45) NOT NULL,
8     FOREIGN KEY (Autore) REFERENCES Utente(Email) ON DELETE CASCADE ON
9     UPDATE CASCADE,
10     PRIMARY KEY (IDLista)
11 );

```

```

1 -- @block initialize Contenuto
2 CREATE TABLE Contenuto (
3     IDLista UUID NOT NULL,
4     IDAlloggio UUID NOT NULL,
5     FOREIGN KEY (IDLista) REFERENCES Lista(IDLista) ON DELETE CASCADE,
6     FOREIGN KEY (IDAlloggio) REFERENCES Alloggio(IDAlloggio) ON DELETE
7     CASCADE,
8     PRIMARY KEY (IDLista, IDAlloggio)
9 );

```

```

1 -- @block initialize Foto
2 CREATE TABLE Foto (
3     Path TEXT,
4     IDAlloggio UUID NOT NULL,
5     FOREIGN KEY (IDAlloggio) REFERENCES Alloggio(IDAlloggio) ON DELETE
6     CASCADE,
7     PRIMARY KEY (Path, IDAlloggio)
8 );

```

```

1
2 -- @block initialize Servizio
3 CREATE TABLE Servizio (
4     TipoServizio TEXT,
5     IDAlloggio UUID NOT NULL,
6     FOREIGN KEY (IDAlloggio) REFERENCES Alloggio(IDAlloggio) ON DELETE
7     CASCADE,
8     PRIMARY KEY (TipoServizio, IDAlloggio)
9 );

```

3.2 DML di Popolamento di Tutte le Tabelle del Database

```
1  -- @block Dump dei dati per la tabella Utente --
2  INSERT INTO Utente (Nome, Cognome, Email, Password, Host)
3  VALUES (
4      'Marco',
5      'Molica',
6      'marco.molica@edu.unito.to',
7      'a',
8      TRUE
9  );
10 INSERT INTO Utente (Nome, Cognome, Email, Password)
11 VALUES (
12     'Iman',
13     'Solaih',
14     'iman.solaih@edu.unito.to',
15     'a'
16 );
17 INSERT INTO Utente (Nome, Cognome, Email, Password, Host)
18 VALUES (
19     'Eduard',
20     'Occhipinti',
21     'eduard.occhipinti@edu.unito.to',
22     'a',
23     TRUE
24 );
25 INSERT INTO Utente (Nome, Cognome, Email, Password)
26 VALUES (
27     'Host',
28     'Incredibile',
29     'host.incredibile@edu.unito.to',
30     'a'
31 );
32 INSERT INTO Utente (Nome, Cognome, Email, Password)
33 VALUES (
34     'Host',
35     'Sensazionale',
36     'host.sensazionale@edu.unito.to',
37     'a'
38 );
39 INSERT INTO Utente (Nome, Cognome, Email, Password)
40 VALUES (
41     'Super',
42     'Hostone',
43     'superhostone@edu.unito.to',
44     'a'
45 );
46 -- @block Dump dei dati per la tabella Telefono --
47 INSERT INTO Telefono (Utente, Numero, Prefisso)
48 VALUES ('marco.molica@edu.unito.to', '3465432121', '+39');
49 INSERT INTO Telefono (Utente, Numero, Prefisso)
50 VALUES ('marco.molica@edu.unito.to', '3392516621', '+39');
51 -- @block Dump dei dati per la tabella Pagamento --
52 INSERT INTO Pagamento (Utente, Numero, Prefisso)
53 VALUES (
54     'marco.molica@edu.unito.to',
55     '0000012345678901',
56     'Visa'
57 );
58 INSERT INTO Pagamento (Utente, Numero, Prefisso)
59 VALUES (
60     'iman.solaih@edu.unito.to',
61     '2141512512231231',
62     'American Express'
63 );
```

```

64 -- @block Dump dei dati per la tabella Alloggio --
65 INSERT INTO Alloggio (
66     IDAlloggio,
67     Nome,
68     Host,
69     Descrizione,
70     Tipologia,
71     OrarioCheckIn,
72     OrarioCheckOut,
73     Costo,
74     CostoPulizia,
75     CAP,
76     Comune,
77     Civico,
78     Via
79 )
80 VALUES (
81     '97c645ea-e19c-411c-8539-5d21def970cc',
82     'Casa di Emme',
83     'marco.molica@edu.unito.to',
84     'Un alloggio sempre disponibile per lo studio in compagnia e
      la realizzazione del progetto di db',
85     'Appartamento',
86     '18:00:00',
87     '23:59:00',
88     1,
89     0,
90     10129,
91     'Torino',
92     3,
93     'Via Peano'
94 );
95 INSERT INTO Alloggio (
96     IDAlloggio,
97     Nome,
98     Host,
99     Descrizione,
100    Tipologia,
101    OrarioCheckIn,
102    OrarioCheckOut,
103    Costo,
104    CostoPulizia,
105    CAP,
106    Comune,
107    Civico,
108    Via
109 )
110 VALUES (
111     'e3084a88-a6fc-42bb-8a00-4c24f07229b6',
112     'Casa di Eduard',
113     'eduard.occhipinti@edu.unito.to',
114     'Stanza molto tranquilla e confortevole, ottima per studenti
      del dipartimento di informatica',
115     'Stanza condivisa',
116     '09:00:00',
117     '22:00:00',
118     10,
119     5,
120     10129,
121     'Torino',
122     12,
123     'Via non ricordo indirizzo'
124 );
125 -- @block Dump dei dati per la tabella Prenotazione --
126 INSERT INTO Prenotazione (
127     IDPrenotazione,
128     Richiedente,
129     IDAlloggio,

```

```

130         DataInizio,
131         DataFine,
132         NumeroOspiti,
133         Stato
134     );
135 VALUES (
136     '9a0d4e3f-0e1e-49b5-838a-b9521c794e6f',
137     'marco.mollica@edu.unito.to',
138     'e3084a88-a6fc-42bb-8a00-4c24f07229b6',
139     '2022-01-05',
140     '2023-01-05',
141     0,
142     'Confermato'
143 );
144 INSERT INTO Prenotazione (
145     IDPrenotazione,
146     Richiedente,
147     IDAlloggio,
148     DataInizio,
149     DataFine,
150     NumeroOspiti,
151     Stato,
152     Soggiorno
153 )
154 VALUES (
155     '12af2073-3418-4a7a-8b2c-e9f91f2234ed',
156     'marco.mollica@edu.unito.to',
157     'e3084a88-a6fc-42bb-8a00-4c24f07229b6',
158     '2021-01-01',
159     '2022-01-01',
160     0,
161     'Confermato',
162     TRUE
163 );
164 INSERT INTO Prenotazione (
165     IDPrenotazione,
166     Richiedente,
167     IDAlloggio,
168     DataInizio,
169     DataFine,
170     NumeroOspiti
171 )
172 VALUES (
173     '99acbb6b-9f4e-4234-a49d-a27acd0f0173',
174     'iman.solai@edu.unito.to',
175     '97c645ea-e19c-411c-8539-5d21def970cc',
176     '2022-10-06',
177     '2023-10-13',
178     0
179 );
180 -- @block Dump dei dati per la tabella Recensione --
181 INSERT INTO Recensione (
182     IDRecensione,
183     IDPrenotazione,
184     Autore,
185     DataRecensione,
186     ValutazionePosizione,
187     ValutazionePulizia,
188     Corpo,
189     Categoria
190 )
191 VALUES (
192     'e6ff4eb7-c7cb-4f3e-b59e-267544c984a2',
193     '12af2073-3418-4a7a-8b2c-e9f91f2234ed',
194     'marco.mollica@edu.unito.to',
195     '2022-01-05',
196     4,
197     5,

```



```

198         'Ottima casa, ci tornerò',
199         'Alloggio'
200     );
201 INSERT INTO Recensione (
202     IDRecensione,
203     IDPrenotazione,
204     Autore,
205     DataRecensione,
206     ValutazioneComunicazione,
207     Corpo,
208     Categoria
209 )
210 VALUES (
211     '14155146-d619-41dd-8aca-fd789262287d',
212     '12af2073-3418-4a7a-8b2c-e9f91f2234ed',
213     'marco.mollica@edu.unito.to',
214     '2022-01-05',
215     2,
216     'Ho avuto molte difficoltà a comunicare con Eduard perchè
        parla spesso russo',
217     'Host'
218 );
219 INSERT INTO Recensione (
220     IDRecensione,
221     IDPrenotazione,
222     Autore,
223     DataRecensione,
224     Corpo,
225     Categoria
226 )
227 VALUES (
228     '16ffe4b2-b4e6-45cb-86e1-5402b28d372d',
229     '12af2073-3418-4a7a-8b2c-e9f91f2234ed',
230     'eduard.occhipinti@edu.unito.to',
231     '2022-01-05',
232     'Marco è stato un grande ospite, mi ha aiutato in tutti i
        progetti',
233     'Utente'
234 );
235 -- @block Dump dei dati per la tabella Commento --
236 INSERT INTO Commento (IDrecensione, DataCommento, Autore, Corpo)
237 VALUES (
238     'e6ff4eb7-c7cb-4f3e-b59e-267544c984a2',
239     '2022-01-06 12:04:21',
240     'eduard.occhipinti@edu.unito.to',
241     'Sono molto contento che sia stato tutto di tuo gradimento'
242 );
243 INSERT INTO Commento (IDrecensione, DataCommento, Autore, Corpo)
244 VALUES (
245     'e6ff4eb7-c7cb-4f3e-b59e-267544c984a2',
246     '2022-01-06 13:01:18',
247     'marco.mollica@edu.unito.to',
248     'Grazie, lo sono anche io'
249 );
250 -- @block Dump dei dati per la tabella Commento --
251 INSERT INTO Lista (IDLista, Nome, Descrizione, Autore)
252 VALUES (
253     '14e9d1b4-7144-4e9f-9ec2-571432ef2068',
254     'Lista1',
255     'Lista di prova',
256     'marco.mollica@edu.unito.to'
257 );
258 -- @block Dump dei dati per la tabella Contenuto --
259 INSERT INTO Contenuto (IDLista, IDAlloggio)
260 VALUES (
261     '14e9d1b4-7144-4e9f-9ec2-571432ef2068',
262     '97c645ea-e19c-411c-8539-5d21def970cc'
263 );

```

```

264 INSERT INTO Contenuto (IDLista, IDAlloggio)
265 VALUES (
266         '14e9d1b4-7144-4e9f-9ec2-571432ef2068',
267         'e3084a88-a6fc-42bb-8a00-4c24f07229b6'
268     );
269 -- @block Dump dei dati per la tabella Foto --
270 INSERT INTO Foto (IDAlloggio, Path)
271 VALUES (
272         '97c645ea-e19c-411c-8539-5d21def970cc',
273         'http://www.unito.it/wp-content/uploads/2017/01/unito-logo.png'
274     );
275 -- @block Dump dei dati per la tabella Foto --
276 INSERT INTO Servizio (IDAlloggio, TipoServizio)
277 VALUES ('97c645ea-e19c-411c-8539-5d21def970cc', 'Wifi');
278 INSERT INTO Servizio (IDAlloggio, TipoServizio)
279 VALUES (
280         '97c645ea-e19c-411c-8539-5d21def970cc',
281         'Animali ammessi'
282     );

```

3.3 Qualche Operazione di cancellazione e modifica

```
1 INSERT INTO Utente (Nome, Cognome, Email, Password, Host, SuperHost)
2 VALUES (
3     'Marco',
4     'Molica',
5     'marco.molica@ehostone.it',
6     'a',
7     FALSE,
8     TRUE
9 );
10 -- L'utente non viene creato perchè questa query di insert viola il vincolo
    aggiunto in fase di DDL per cui un utente non può essere un superhost
    se non è un host
11 INSERT INTO Utente (Nome, Cognome, Email, Password, Verificato)
12 VALUES (
13     'Marco',
14     'Molica',
15     'marco.molica@verificatone.it',
16     'a',
17     TRUE
18 );
19 -- L'utente non viene creato perchè questa query di insert viola il vincolo
    aggiunto in fase di DDL per cui un utente può risultare verificato
    solamente se ha una carta di identità valida
20 INSERT INTO Prenotazione (
21     Richiedente,
22     IDAlloggio,
23     DataInizio,
24     DataFine,
25     NumeroUspiti
26 )
27 VALUES (
28     'iman.solaih@edu.unito.to',
29     '97c645ea-e19c-411c-8539-5d21def970cc',
30     '2022-10-06',
31     '2021-10-13',
32     0
33 );
34 -- La prenotazione non viene creata perchè viene violato il vincolo per cui
    la prenotazione deve avere una data di inizio coerente con la data di
    fine
35 INSERT INTO Recensione (
36     IDPrenotazione,
37     Autore,
38     DataRecensione,
39     Corpo,
40     ValutazionePosizione,
41     Categoria
42 )
43 VALUES (
44     '12af2073-3418-4a7a-8b2c-e9f91f2234ed',
45     'eduard.occhipinti@edu.unito.to',
46     '2022-01-05',
47     'Marco è stato un grande ospite, mi ha aiutato in tutti i progetti',
48     5,
49     'Utente'
50 );
51 -- La recensione non viene creata perchè viene violato il vincolo di
    categoria per cui una recensione fatta ad utente non deve avere
    valutazioni
52 DELETE FROM Alloggio
53 WHERE IDAlloggio = '97c645ea-e19c-411c-8539-5d21def970cc';
54 -- In questo il DBMS ci permetterà di effettuare la cancellazione, però
    eliminerà tutti i record nella tabella Contenuto che fanno riferimento
```

```

a questo alloggio.
55 -- Tutte le prenotazioni che fanno riferimento a questo alloggio riceveranno
    un update all'attributo IDAlloggio con NULL, per il vincolo ON DELETE
    SET NULL aggiunto in fase di DDL.
56 UPDATE Utente
57 SET email = 'mmolica@edu.unito.to'
58 WHERE Email = 'marco.molica@edu.unito.to' -- In questo caso l'aggiornamento
    dell'attributo email comporterà l'aggiornamento a cascata di tutte le
    tabelle dove è presente una relazione con l'attributo email, per via
    della clausola ON UPDATE CASCADE.

```

3.3.1 Qualche operazione di sistema

```

1 -- @block Calcolo per aggiornare il tasso di cancellazione di ciascun host
    che abbia effettuato almeno un soggiorno
2 WITH ElencoPrenotazioni as (
3     SELECT Prenotazione.Stato as Stato,
4           Utente.Email as Host
5     FROM prenotazione
6          INNER JOIN Alloggio on prenotazione.IDAlloggio = alloggio.idalloggio
7          INNER JOIN Utente on Alloggio.Host = Utente.Email
8     WHERE Utente.Host = TRUE
9 )
10 SELECT Host,
11        Count(*) as Totale,
12        Count(*) FILTER (
13            WHERE Stato = 'Cancellato dall" host'
14        ) as Cancellato,
15        ROUND(
16            (
17                Count(*) FILTER (
18                    WHERE Stato = 'Cancellato dall" host'
19                ) / Count(*)::numeric
20            ) * 100,
21            2
22        ) as PercentualeCancellati
23 FROM ElencoPrenotazioni
24 GROUP BY Host;
25
26
27 -- @block Calcolo del numero di soggiorni di ogni host che abbia effettuato
    almeno un soggiorno
28 WITH ElencoPrenotazioni as (
29     SELECT Prenotazione.DataInizio as DStart,
30           Prenotazione.DataFine as DEnd,
31           Prenotazione.Soggiorno,
32           Prenotazione.Stato as Stato,
33           Utente.Email as Host
34     FROM prenotazione
35          INNER JOIN Alloggio on prenotazione.IDAlloggio = alloggio.idalloggio
36          INNER JOIN Utente on Alloggio.Host = Utente.Email
37     WHERE Utente.Host = TRUE
38           AND Soggiorno = TRUE
39 )
40 SELECT Host,
41        SUM (DEnd::Date - DStart::Date) AS GiorniTotali,
42        Count(*) As Soggiorni
43 FROM ElencoPrenotazioni
44 GROUP BY Host;

```