Laboratorio Basi di Dati 2021/22

Progetto di piattaforma di home booking

-----------------------------------------------------------------------------------------------------------------------------------------------

**Nome**: Mickael

**Cognome**: Ordine

**MATR**: 926927

**Turno**: T4

-----------------------------------------------------------------------------------------------------------------------------------------------

SOMMARIO

1. Progettazione concettuale

1.1. Requisiti iniziali.

1.2. Glossario dei termini.

1.3. Requisiti rivisti e strutturati in gruppi di frasi omogenee.

1.4. Schema E-R + business rules.

2. Progettazione logica

2.1. Tavola dei volumi (con motivazione delle scelte effettuate).

2.2. Tavola delle operazioni.

2.3. Ristrutturazione dello schema E-R:

2.3.1. Analisi delle ridondanze:

• Schema di navigazione relativo all’operazione in presenza e in assenza di

ridondanza.

• Tavola degli accessi in presenza e in assenza di ridondanza.

• Confronto in spazio e tempo tra presenza e assenza di ridondanza.

• Scelta se introdurre o non introdurre la ridondanza con motivazione.

2.3.2. Eliminazione delle generalizzazioni (con motivazione delle scelte effettuate).

2.3.3. Eventuale partizionamento/accorpamento di entità e associazioni (con motivazione

delle scelte effettuate).

2.3.4. Eventuale scelta degli identificatori principali (con motivazione delle scelte

effettuate).

2.4. Schema E-R ristrutturato + business rules.

2.5. Schema relazionale (con indicazione dei vincoli di integrità referenziale).

3. Implementazione

3.1. DDL di creazione del database;

3.2. DML di popolamento di tutte le tabelle del database;

3.3. Operazioni di cancellazione e modifica

PROGETTAZIONE CONCETTUALE

1.1. Requisiti iniziali.

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

alloggi di vario tipo ad esempio interi appartamenti, stanze private (camera privata e spazi

comuni) e stanze condivise (spazio in comune e camera condivisa).

Gli utenti si registrano al servizio fornendo indirizzo email, password, nome, cognome,

numero o numeri di telefono. Se l’utente fornisce la foto della carta d’identità, viene

riconosciuto come verificato. Inoltre, l’utente deve indicare un metodo di pagamento per

poter prenotare. Gli utenti possono essere ospiti o “host” ovvero possono a loro volta

ospitare altri utenti del servizio in uno o più alloggi di loro proprietà. Inoltre gli “host” possono

diventare “superhost” se soddisfano i seguenti requisiti:

● Devono aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.

● Devono aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni

100 prenotazioni) massimo.

● Devono aver mantenuto una valutazione complessiva di 4,8 considerando tutti i

soggiorni in tutte le case di sua proprietà.

Gli utenti superhost ricevono un badge sul loro profilo.

Gli alloggi sono descritti indicando un nome, l’indirizzo (visibile all’ospite solo quando la

prenotazione è confermata, altrimenti è visibile solo il comune), una descrizione, il prezzo per

notte per persona e i costi di pulizia, delle foto, i servizi (ad esempio, cucina, wi-fi, lavatrice,

ecc.), numero di letti e orario di check-in e check-out oltre all’host a cui appartiene, il rating

medio e il numero di recensioni

Gli utenti possono aggiungere alcune case tra i preferiti. Gli utenti possono avere diverse

liste, ad esempio in base al viaggio che vogliono compiere.

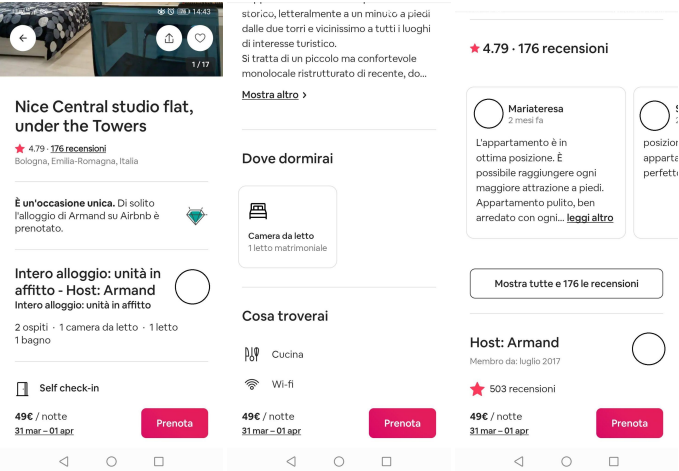
Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il

soggiorno e il numero degli ospiti. Se gli ospiti sono a loro volta utenti del servizio, se ne

possono indicare i nominativi. La prenotazione deve essere confermata o rifiutata dall’host.

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

prenotazione può essere cancellata sia dall’ospite che dall’host.



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

fatta dagli ospiti comprende due testi (uno per l’appartamento e uno per l’host) e una serie di

punteggi in una scala da 1 a 5 su dimensioni come pulizia, comunicazione, posizione,

qualità/prezzo. La valutazione complessiva del soggiorno è una media delle valutazioni

ricevute sulle singole dimensioni. Le recensioni degli host comprendono solo un commento

testuale. Le recensioni possono essere visibili o non visibili. Diventano visibili quando

entrambi hanno fatto la recensione oppure se uno dei due non ha fatto la recensione, l’altra

diventa visibile dopo 7 giorni dalla fine del soggiorno. Gli host e gli ospiti possono

commentare più volte le review in cui sono coinvolti, creando un thread di discussione.

Le recensioni sono visibili sui profili degli utenti suddivise in base a quelle ricevute come

ospite e come host.

La base di dati deve supportare le seguenti operazioni:

● Una volta a settimana viene effettuato un calcolo per aggiornare il tasso di

cancellazione di ciascun host.

● Una volta al giorno si controllano le condizioni per la qualifica di superhost e viene

aggiornato lo status degli host.

● Una volta al mese viene calcolata la classifica degli alloggi più graditi

1.2. Glossario dei termini.

|  |  |  |  |
| --- | --- | --- | --- |
| Termine | Descrizione | Sinonimi | Collegamenti |
| UTENTE | Colui che usufruisce della piattaforma come utilizzatore attivo della disponibilità degli alloggi presenti | Ospiti, persona | Lista preferiti, commento, recensione, prenotazione. |
| LISTA PREFERITI | Insieme di case che la persona aggiunge ai suoi favoriti | Preferiti | Alloggio, utente. |
| HOST | Colui che da la disponibilità di ospitare persone in alloggi di sua proprità. |  | Alloggio, prenotazione, recensione, commento. |
| COMMENTO | Elemento usato per comunicare nella recensione | Discussione | Recensione, utente, host. |
| RECENSIONE | Elemento usato per determinare i voti del tuo soggiorno, della disponibilità dell’host, e usato dall’host per determinare il comportamento dell’ospite | Valutare, review | Alloggi terminati, commento, utente, host. |
| PIATTAFORMA HOME BOOKING | Servizio su cui gli utenti possono affittare case fornite dagli host. | Servizio |  |
| OSPITI DELLA PRNEOTAZIONE | Attributo della prenotazione, determina quante persone oltre a te saranno presenti in un abitazione. |  |  |
| ALLOGGI | Case abitative di proprietà di un host, gli utenti le possono prenotare | Case, appartamenti | Host, prenotazione, lista preferiti. |
| PRENOTAZIONE | Atto che determina un utente specifico che vuole un abitazione specifica per un totale lasso di tempo | Affittare, Soggiorno | Utente, Host ,alloggi, soggiorni terminati. |
| ALLOGGI TERMINATI | Alloggi che hanno superato la data di fine degli alloggi prenotati. |  | Prenotazione, recensione. |

1.3. Requisiti rivisti e strutturati in gruppi di frasi omogenee.

Si vuole realizzare una base di dati per ~~un servizio~~ una piattaforma di HomeBooking che permette di ~~affittare e~~ prenotare alloggi come  ~~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 registrati hanno ~~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 hanno ~~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~~ hanno 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~~ utente, 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 al~~l’host a cui appartiene, il rating medio e il numero di recensioni ~~(vedere Fig. 1)~~.

Gli utenti possono aggiungere ~~alcune case~~ alloggi tra ~~i preferiti~~ le liste dei preferiti. Gli utenti possono avere diverse liste dei preferiti, ~~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~~ la prenotazione 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~~ dall’utente che dall’host.

Al termine ~~del soggiorno~~ della prenotazione, gli ~~ospiti~~ utenti e gli host si possono ~~valutare~~ recensire a vicenda. La recensione fatta dagli ~~ospiti~~ utenti comprende due testi (uno per ~~l’appartamento~~ l’alloggio 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~~ della prenotazione. Gli host e gli ~~ospiti~~ utenti possono commentare più volte le ~~review~~ recensioni in cui sono coinvolti, creando un thread di ~~discussione~~ commenti. ~~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

**Frasi relative agli utenti:**

Gli utenti registrati hanno indirizzo email, password, nome, cognome, numero o numeri di telefono. Se l’utente fornisce la foto della carta d’identità, viene riconosciuto come verificato. L’utente deve indicare un metodo di pagamento per poter prenotare.

Gli utenti possono aggiungere alloggi tra le liste dei preferiti. Gli utenti possono avere diverse liste dei preferiti. Gli utenti possono prenotare indicando un intervallo di date per la prenotazione e il numero degli ospiti.

Gli utenti possono recensire gli host e gli alloggi, e possono commentare le recensioni che gli host gli hanno fatto.

**Frasi relative agli Host:**

Gli “host” possono ospitare altri utenti in uno o più alloggi di loro proprietà. La prenotazione può essere cancellata sia dall’utente che dall’host.

Gli Host possono recensire gli utenti che hanno una prenotazione finita in un loro alloggio di proprità, gli host possono commentare recensioni che gli utenti gli hanno fatto.

**Frasi relative ai superHost:**

gli “host” possono diventare “superhost” se hanno:

● completato almeno 10 soggiorni, per un totale di almeno 100 notti.

● conservato un tasso di cancellazione dell'1% massimo.

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

~~I~~ superhost ricevono un badge sul loro profilo.

**Frasi relative ad alloggio:**

Si vuole realizzare una base di dati per una piattaforma di HomeBooking che permette di prenotare alloggi come appartamenti, stanze private e stanze condivise.

Gli alloggi hanno 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 utente, i costi di pulizia, delle foto, i servizi, numero di letti e orario di check-in e check-out, l’host a cui appartiene, il rating medio e il numero di recensioni.

**Frasi relative a prenotazione:**

Si vuole realizzare una base di dati per una piattaforma di HomeBooking che permette di prenotare alloggi.

**Frasi relative a soggiorni terminati:**

Al termine della prenotazione, gli utenti e gli host si possono recensire a vicenda.

**Frasi relative a liste preferiti:**

Gli utenti possono aggiungere alloggi tra le liste dei preferiti. Gli utenti possono avere diverse liste dei preferiti

**Frasi relative a recensione a U:**

Le recensioni degli host comprendono un commento testuale.

**Frasi relative a recensione a H:**

La recensione fatta dagli utenti comprende due testi (uno per l’alloggio e uno per l’host) e una serie di punteggi in una scala da 1 a 5 su pulizia, comunicazione, posizione, qualità/prezzo.

**Frasi relative a commentoHost:**

Gli host possono commentare più volte le recensioni in cui sono coinvolti, creando un thread di commenti.

**Frasi relative a commentoUtente:**

Gli utenti possono commentare più volte le recensioni in cui sono coinvolti, creando un thread di commenti.

1.4. Schema E-R + business rules.

**Dizionario dei dati (Entità):**

|  |  |  |  |
| --- | --- | --- | --- |
| **entità** | **descrizione** | **attributi** | **identificatore** |
| UTENTE | Entità che identifica colui che può prenotare un alloggio sulla piattaforma. | Email,psw,nome,cognome, numero di telefono, fotoC.I, metodo di pagamento. | Email |
| HOST | Entità che identifica l’utente che si registra dando dicendo di possedere degli alloggi. | EmailHost, psw, nome, cognome, numero di telefono, superHost, Badge | EmailHost |
| ALLOGGI | Entità che identifica il luogo in cui gli utenti potranno soggiornare, è di proprietà di un solo host. | nomeAlloggio, indirizzo, EmailHost, prezzoPerNotte, descrizione, costiPulizia, foto(1,N), servizi(1,N), numLetti(1,N), check-in, check-out, ratingmedio, comune | nomeAlloggio, indirizzo, EmailHost |
| LISTA PREFERITI | Entità che identifica la/e lista/e degli alloggi preferiti dagli utenti. | email, nomeAlloggio, indirizzo, EmailHost, nome lista | Email, nomeAlloggio, indirizzo, EmailHost. |
| PRENOTAZIONE | Entità che identifica un alloggio prenotato per un determinato periodo di tempo da uno specifico host che lo fornisce. | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,  ,costoTot, status.  Attributi composti:  ospite che ha nome, cognome, email. | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |
| SOGGIORNO TERMINATO | Entità che identifica che al giorno attuale diverse prenotazioni abbiamo superato la data di fine prenotazione. | NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza | NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |
| RECENSIONE A U | Entità che descrive la recensione che un Host fa ad un utente quando è finita la prenotazione. | nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza,  visibilità, testo | nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza |
| RECENSIONE A H | Entità che descrive la recensione che un utente fa ad un host è finita la prenotazione. | nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, testoApp, testoHost, pulizia, comunicazione, posizione, qualità/prezzo, visibilità | nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio |
| COMMENTO A U | Entità che descrive un commento che può essere aggiunto ad una recensione in cui un utente è stato interpellato. | nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, timestamp, testo | nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, timestamp |
| COMMENTO A H | Entità che descrive un commento che può essere aggiunto ad una recensione in cui è interpellato l’host | nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp, testo | nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp |
| SOGGIORNO CANCELLATO o RIFIUTATO | Entità che descrive gli alloggi che per 4 casistiche principali non sono andati a buon fine:  la prenotazione è stata cancellata dall’host o dall’utente,  la prenotazione è stata rifiutata dall’host,  i tempi per il pagamento sono scaduti | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,  cancellato | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |

**Dizionario dei dati (Relazioni):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazione** | **Descrizione** | **Entità Coinvolte** | **Attributi** |
| salva | Associa l’utente alla lista dei preferiti dando la possibilità di creare nuove liste | Utente, Lista dei preferiti | / |
| comprende | Associa lista dei preferiti e alloggio dando la possibilità di aggiungere alloggi alla/e lista/e | Lista dei preferiti, alloggio | / |
| possiede | Associa Host agli alloggi di sua proprietà. | Host, alloggio | / |
| assegnato | Associa l’alloggio ad una prenotazione che un utente ha effettuato. | Alloggio, Prenotazione | / |
| riserva | Associa l’utente alla/e sua/e prenotazione/i. | Utente, prenotazione | / |
| cancellaU | Associa l’utente alle prenotazione che ha cancellato. | Utente, prenotazione. | / |
| paga | Associa l’utente alle prenotazioni che ha pagato. | Utente, prenotazione | / |
| cancellaH | Associa l’host alle cancellazioni che ha effettuato su prenotazioni. | Host, prenotazioni | / |
| accetta | Associa l’host alle accettazioni che ha effettuato su prenotazioni. | Host, prenotazioni | / |
| rifiuta | Associa l’host alle prenotazioni che ha rifiutato. | Host, prenotazioni | / |
| scade | Associa lo scadere della data della prenotazione pagata ai soggiorni che sono terminati. | Prenotazione, Soggiorno cancellato o rifiutato | / |
| è valutato da U | Associa la recensione di un alloggio e l’host all’alloggio stesso | Soggiorno Terminato, recensione a H | / |
| è valutato da H | Associa le recensioni fatte da un utente al termine del soggiorno | Soggiorno terminato, recensione a U | / |
| effettuaU | Associa la possibilità che l’utente ha di effettuare una recensione sull’alloggio e sull’host al termine del soggiorno | Utente, recensione a H | / |
| effettuaH | Associa la possibilità che l’Host ha di effettuare una recensione sull’utente al termine del soggiorno | Host, recensione a U | / |
| scriveU | Associa la possibilità di un utente di commentare una recensione che gli hanno fatto | Utente, commento U | / |
| scriveH | Associa la possibilità di un host di commentare una recensione che un utente gli ha fatto. | Host, Commento H | / |
| possiedeU | Associazione tra recensione e commento che permette di stabilire quanti commenti ha la recensione. | Commento U, recensione a U | / |
| possiedeH | Associazione tra recensione e commento che permette di stabilire quanti commenti ha la recensione. | Commento H, recensione H | / |
| Risponde U | Relazione ricorsiva per l’entità commento, permette di avere nell’entità commento un susseguirsi di commenti | CommentoU, CommentoU | / |
| Risponde H | Relazione ricorsiva per l’entità commento, permette di avere nell’entità commento un susseguirsi di commenti | CommentoH, CommentoH | / |
| Ripensamento | Relazione che determina un ripensamento che c’è stato da parte di host o utente su una determinata prenotazione | Prenotazione, soggiorni non terminati | / |
| SCRIVE COMMENTO A H | Relazione che determina la possibilità da parte dell’utente di commentare un commento che l’host ha fatto | Utente, commento a H | / |
| SCRIVE COMMMENTO A U | Relazione che determina la possibilità da parte dell’host di commentare un commento che l’utente ha fatto | Host, commento a U | / |

Business Rule:

* Quando l’utente effettua una prenotazione, si verifica il metodo di pagamento se è diverso da NULL, se si si procede alla prenotazione, se no invece si fa richiesta di inserire i dati del metodo di pagamento che si vuole usare.
* l’indirizzo è visibile all’ospite solo quando la prenotazione è confermata, altrimenti è visibile solo il comune.
* Per diventare superHost bisogna:
  + 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.
* Le recensioni 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.
* Le prenotazioni vanno in soggiorno terminato quando la data attuale è > della data presente nell’attributo partenza della tabella prenotazione.

**2. Progettazione logica**

2.1. Tavola dei volumi

|  |  |  |  |
| --- | --- | --- | --- |
| **concetto** | **tipo** | **volume** | **motivazione** |
| utente | E | 3 milioni | Secondo il giornale “la repubblica” nel 2017 erano all’incirca 2 milioni di persone al mondo ad usare airBNB, ho stimato una crescita 50%. |
| host | E | 1,5 milioni | Secondo il giornale “La Repubblica” nel 2017 erano presenti 1 milione di Host, ho stimato una crescita del 50%. |
| alloggi | E | 2,7 milioni | Sappiamo che un Host ha sicuramente un alloggio, facendo una media si può pensare che ogni host ha 1,8 alloggi personali disponibili. |
| prenotazioni | E | 800 mila | Su 3 milioni di utenti, avremo che il 27% delle persone avrà una prenotazione in sospeso/conf./pagata |
| Soggiorni terminati | E | 3 milioni | Avremo bisogno di tenere almeno come media due alloggi negli storici per ogni Host così da poter conferire una media voto sensata |
| recensioni | E | 4,5 milioni | Contando che la metà tra utenti e host recensiscano i viaggi e i soggiorni che fanno e ospitano rispettivamente, con la possibilità che ne abbiano fatti 2. |
| commenti | E | 6 milioni | C’è maggiore probabilità che ci siano più commenti sotto una stessa recensione anche se non in tutte le recensioni. |
| Lista preferiti | E | 2 milioni | 1 utente su due in media si crea una lista degli alloggi preferiti e quasi sicuramente ne ha più di una. |
| Comprende | R | 15 | Ogni lista conterrà all’incirca 15 alloggi. |
| cancellaU | R | 60 mila | L’utente cancellerà una piccola parte delle prenotazioni che ha riservato |
| paga | R | 720 mila | Ne pagherà la maggior parte |
| accetta | R | 700 mila | L’host accetterà la maggior parte delle prenotazioni effettuate |
| cancellaH | R | 20 mila | L’host cancellerà poche prenotazioni |
| rifiuta | R | 20 mila | Ne rifiuterà poche |

**Tavola delle operazioni**

|  |  |  |  |
| --- | --- | --- | --- |
| **Operazione** | **Tipo** | **Frequenza** | **Motivazione** |
| Tasso cancellazione Host | B | 1/settimana | Operazione importante per finalizzare correttamente la presenza di superhost sulla piattaforma |
| Qualifica status SuperHost | B | 1/giorno | Operazione che controlla la possibilità di diventare un SuperHost |
| Alloggi con avarage rating più alto | B | 1/mese | Operazione che permette di scegliere quali alloggi dovranno essere portati in altro a livello di ricerca. |
| Scadenza Prenotazioni | B | 1/giorno | Operazione che ci permette di stabilire quando l’utente e l’host possono fornirsi un parere a vicenda attraverso delle recensioni. |
| Controllo Voto Recensioni | B | 1/settimana | Operazione importante per finalizzare correttamente la presenza di superhost sulla piattaforma |
| Controllo del completamento di 10 soggiorni con 100 notti per gli Host | B | 1/settimana | Operazione importante per finalizzare correttamente la presenza di superhost sulla piattaforma |
| Prenotazioni effettuate | I | 3000/giorno | Operazione importante per determinare le nuove prenotazioni che sono state effettuate. |
| Registrazione utente/host | I | 2000/giorno | Operazione importante per determinare le nuove utenze del sistema. |
| Pagamenti | I | 2000/giorno | Operazione importante per determinare il pagamento degli alloggi |
| Cancellazioni | I | 100/giorno | Operazione importante per determinare il pagamento degli alloggi |

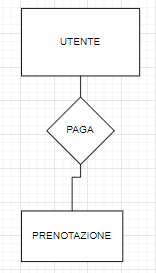
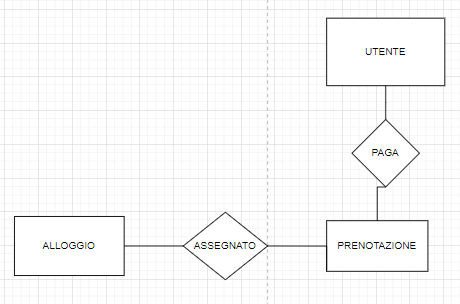
2.3. Ristrutturazione dello schema E-R:

**!!!!Analisi delle ridondanze:**

All’interno del mio schema E-R ho riscontrato una sola ridondanza, la presenza di costo totale all’interno di prenotazione è ridondante dal momento in cui con una semplice query di selezione siamo in grado di ottenere da alloggio il prezzo per notte e moltiplicarlo per il numero di giorni della durata del soggiorno.

È dunque possibile eliminare la presenza dell’attributo costo totale dalla tabella prenotazioni senza perdere integrità dei dati.

Con ridondanza: senza ridondanza:

Problema:

Pagamento da parte del nostro utente su una prenotazione già effettuata.

**Tavola degli accessi con ridondanza:**

|  |  |  |  |
| --- | --- | --- | --- |
| **concetto** | **costrutto** | **accessi** | **tipo** |
| utente | E | 1 | L |
| paga | R | 1 | L |
| prenotazione | E | 1 | L |

**Tavola degli accessi senza ridondanza:**

|  |  |  |  |
| --- | --- | --- | --- |
| **concetto** | **costrutto** | **accessi** | **tipo** |
| utente | E | 1 | L |
| paga | R | 1 | L |
| prenotazione | E | 1 | L |
| assegnato | R | 1 | L |
| alloggio | E | 1 | L |

**Operazione con ridondanza:**

ipotesi -> supponiamo di utilizzare 4 byte per salvare le informazioni di costo Totale.

2000 volte al giorno \* 3 tot degli accessi -> 6.000

Analizzando in seguito il volume totale delle prenotazioni, essere 800mila, possiamo calcolare cos’ il peso totale dell’attributo:

800000 \* 4 = 3.200.000 byte = 3,2 MB (costo trascurabile)

**Operazione senza ridondanza:**

Qui invece abbiamo 2000 volte al giorno \* 5 tot degli accessi -> 10.000

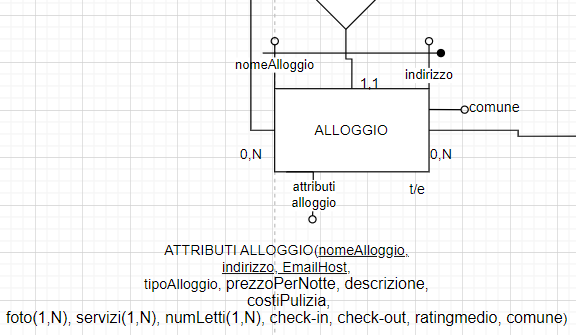
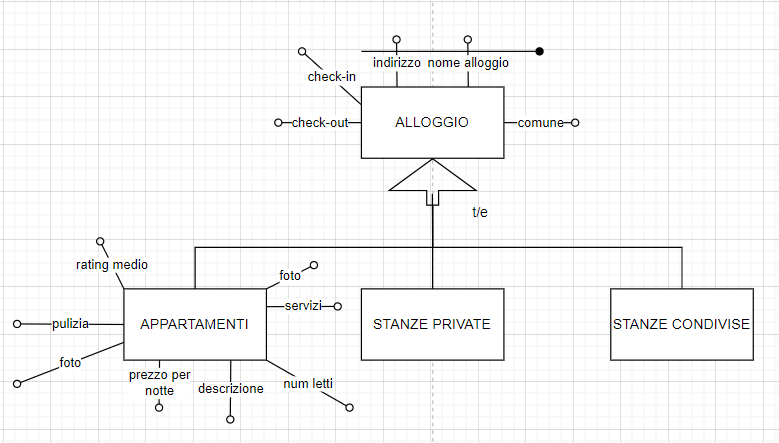
Quindi il numero totale degli accessi salirebbe a 10.000 che salirebbe quasi al doppio rispetto all’operazione contenente la ridondanza costo Totale.

**Conclusioni:**

Ho dunque deciso di tenere l’attributo CostoTot anche se ridondante poiché lo spazio occupato non è grande e permette una quantità di accessi dimezzati rispetto alla versione senza ridondanza.

**Eliminazione delle generalizzazioni**

Nel mio E-R model l’unico posto in cui mi era venuto in mente di inserirli era all’interno dell’entità padre alloggi con le entità figlie rispettivamente “Appartamenti, StanzePrivate, StanzeCondivise”.

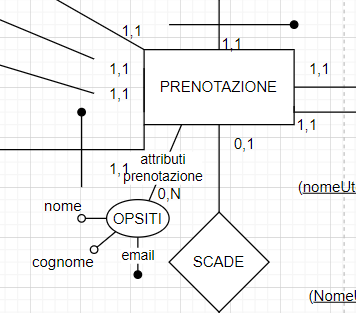
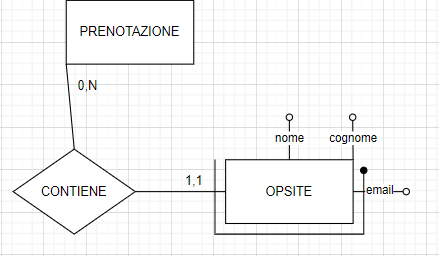




Nel mio caso però mi è subito venuto in mente di raggrupparli sotto un unico attributo che determinasse la tipologia di alloggio di cui si sta parlando creando l’attributo tipoAlloggio che può ottenere i valori di “appartamenti, StanzePrivate, StanzeCondivise”.

**Dizionario Dati (entità) ristrutturato:**

Effettuiamo la rimozione dell’attributo multi-valore Ospiti trasformandolo in un’entità a se in relazione con Prenotazione con la stessa cardinalità che aveva l’attributo multi-valore.

Con la prenotazione, appunto, che può comprendere da 0 a più ospiti per una prenotazione e invece l’ospite può appartenere esclusivamente ad una prenotazione soltanto.

Questo non vuol dire che un ospite “Guy Montag [guymontag451@gmail.com](mailto:guymontag451@gmail.com)” non possa appartenere ad altre prenotazioni, grazie al vincolo con prenotazione però vincoliamo la presenza di ospiti diversi per ogni prenotazione.

|  |  |  |  |
| --- | --- | --- | --- |
| **entità** | **descrizione** | **attributi** | **identificatore** |
| UTENTE | Entità che identifica colui che può prenotare un alloggio sulla piattaforma. | Email,psw,nome,cognome, numero di telefono, fotoC.I, metodo di pagamento. | Email |
| HOST | Entità che identifica l’utente che si registra dando dicendo di possedere degli alloggi. | EmailHost, psw, nome, cognome, numero di telefono, superHost, Badge | EmailHost |
| ALLOGGI | Entità che identifica il luogo in cui gli utenti potranno soggiornare, è di proprietà di un solo host. | nomeAlloggio, indirizzo, EmailHost, tipoAlloggio, prezzoPerNotte, descrizione, costiPulizia, foto(1,N), servizi(1,N), numLetti(1,N), check-in, check-out, ratingmedio, comune | nomeAlloggio, indirizzo, EmailHost |
| LISTA PREFERITI | Entità che identifica la/e lista/e degli alloggi preferiti dagli utenti. | email, nomeAlloggio, indirizzo, EmailHost, nome lista | Email, nomeAlloggio, indirizzo, EmailHost. |
| PRENOTAZIONE | Entità che identifica un alloggio prenotato per un determinato periodo di tempo da uno specifico host che lo fornisce. | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,  ,costoTot, status.  ~~Attributi composti:~~  ~~ospite che ha nome, cognome, email.~~ | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |
| SOGGIORNO TERMINATO | Entità che identifica che al giorno attuale diverse prenotazioni abbiamo superato la data di fine prenotazione. | NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza | NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |
| RECENSIONE A U | Entità che descrive la recensione che un Host fa ad un utente quando è finita la prenotazione. | nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza,  visibilità, testo | nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza |
| RECENSIONE A H | Entità che descrive la recensione che un utente fa ad un host è finita la prenotazione. | nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, testoApp, testoHost, pulizia, comunicazione, posizione, qualità/prezzo, visibilità | nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio |
| COMMENTO A U | Entità che descrive un commento che può essere aggiunto ad una recensione in cui un utente è stato interpellato. | nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, timestamp, testo | nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, timestamp |
| COMMENTO A H | Entità che descrive un commento che può essere aggiunto ad una recensione in cui è interpellato l’host | nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp, testo | nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp |
| OPSITE | Entità che descrive la presenza di ospiti all’interno di una prenotazione. | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza, nome, cognome, EmailOspite | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza, EmailOspite |
| SOGGIORNO CANCELLATO o RIFIUTATO | Entità che descrive gli alloggi che per 4 casistiche principali non sono andati a buon fine:  la prenotazione è stata cancellata dall’host o dall’utente,  la prenotazione è stata rifiutata dall’host,  i tempi per il pagamento sono scaduti | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,  cancellato | nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza |

**Dizionario Dati (relazioni) ristrutturato:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Relazione** | **Descrizione** | **Entità Coinvolte** | **Attributi** |
| salva | Associa l’utente alla lista dei preferiti dando la possibilità di creare nuove liste | Utente(0,N), Lista dei preferiti(1,1) | / |
| comprende | Associa lista dei preferiti e alloggio dando la possibilità di aggiungere alloggi alla/e lista/e | Lista dei preferiti(1,N), alloggio(0,N) | / |
| possiede | Associa Host agli alloggi di sua proprietà. | Host(1,N), alloggio(1,1) | / |
| assegnato | Associa l’alloggio ad una prenotazione che un utente ha effettuato. | Alloggio(0,N), Prenotazione(0,1) | / |
| riserva | Associa l’utente alla/e sua/e prenotazione/i. | Utente(0,N), Prenotazione(0,1) | / |
| cancellaU | Associa l’utente alle prenotazione che ha cancellato. | Utente(0,N), Prenotazione(0,1) | / |
| paga | Associa l’utente alle prenotazioni che ha pagato. | Utente(0,N), Prenotazione(0,1) | / |
| cancellaH | Associa l’host alle cancellazioni che ha effettuato su prenotazioni. | Host(0,N), Prenotazione(0,1) | / |
| accetta | Associa l’host alle accettazioni che ha effettuato su prenotazioni. | Host(0,N), Prenotazione(0,1) | / |
| rifiuta | Associa l’host alle prenotazioni che ha rifiutato. | Host(0,N), Prenotazione(0,1) | / |
| scade | Associa lo scadere della data della prenotazione pagata ai soggiorni che sono terminati. | Prenotazione(0,1), Soggiorno Terminato(1,1) | / |
| è valutato da U | Associa la recensione di un alloggio e l’host all’alloggio stesso | Soggiorno Terminato(0,1), recensione a H(1,1) | / |
| è valutato da H | Associa le recensioni fatte da un utente al termine del soggiorno | Soggiorno terminato(0,1), recensione a U(1,1) | / |
| effettuaU | Associa la possibilità che l’utente ha di effettuare una recensione sull’alloggio e sull’host al termine del soggiorno | Utente(0,N), recensione a H(1,1) | / |
| effettuaH | Associa la possibilità che l’Host ha di effettuare una recensione sull’utente al termine del soggiorno | Host(0,N), recensione a U(0,1) | / |
| scriveU | Associa la possibilità di un utente di commentare una recensione che gli hanno fatto | Utente(0,N), commento U(1,1) | / |
| scriveH | Associa la possibilità di un host di commentare una recensione che un utente gli ha fatto. | Host(0,N), commento H(1,1) | / |
| possiedeU | Associazione tra recensione e commento che permette di stabilire quanti commenti ha la recensione. | Commento U(1,1), recensione a U(0,N) | / |
| possiedeH | Associazione tra recensione e commento che permette di stabilire quanti commenti ha la recensione. | Commento H(1,1), recensione H(0,N) | / |
| contiene | Associa la prenotazione alla quantità di entità opsiti presenti all’interno della prenotazione. | Prenotazione(0,N), ospiti(1,1) | / |
| Ripensamento | Relazione che determina un ripensamento che c’è stato da parte di host o utente su una determinata prenotazione | Prenotazione(0,N), SOGGIORNI CANCELLATI O RIFIUTATI(1,1) | / |

Business Rule:

* Quando l’utente effettua una prenotazione, si verifica il metodo di pagamento se è diverso da NULL, se si, si procede alla prenotazione, se no invece si fa richiesta di inserire i dati del metodo che si vuole usare.
* l’indirizzo è visibile all’ospite solo quando la prenotazione è confermata, altrimenti è visibile solo il comune.
* Per diventare superHost bisogna:
  + 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.
* Le recensioni 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.
* Le prenotazioni vanno in soggiorno terminato quando la data attuale è > della data presente nell’attributo partenza della tabella prenotazione e lo status di prenotazione = 5

**2.5. Schema relazionale**

* UTENTE (Email,psw,nome,cognome, numero di telefono (1,N), fotoC.I\*,metodo di pagamento\*)
* LISTA (email, nomeAlloggio, indirizzo, EmailHost, nome lista)

Con vincolo tra email e email della tabella Utente,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella Alloggio,

con vincolo tra indirizzo e indirizzo della tabella Alloggio,

con vincolo tra EmailHost e EmailHost della tabella Host.

* ALLOGGIO (nomeAlloggio, indirizzo, EmailHost, tipoAlloggio, prezzoPerNotte, descrizione, costiPulizia, foto(1,N), servizi(1,N), numLetti(1,N), check-in, check-out, ratingmedio, comune)

Con vincolo tra EmailHost e EmailHost della tabella Host.

TipoAlloggio può avere i seguenti valori:

“APPARTAMENTO”

“STANZA PRIVATA”

“STANZA CONDIVISA”

* HOST (EmailHost, psw, nome, cognome, numero di telefono (1,N), superHost, Badge)
* PRENOTAZIONE(nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,

costoTot, status)

Con vincolo tra nomeUtente e email della tabella Utente,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella Alloggio,

con vincolo tra indirizzo e indirizzo della tabella alloggio,

con vincolo tra nomeHost e EmailHost della tabella Alloggio

Status può assumere i seguenti valori:

0. pending

1. accettato

2.rifiutato

3. cancellatoHost

4.cancellatoUtente

5. pagato

Con check(può diventare 5 solo se lo status == 1);

* OSPITE(nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza, EmailOspite, nome, cognome)

Con vincolo tra nomeUtente e nomeUtente della tabella prenotazione.

Con vincolo tra nomeAlloggio e nomeAlloggio della tabella prenotazione.

Con vincolo tra indirizzo e indirizzo della tabella prenotazione.

Con vincolo tra nomeHost e nomeHost della tabella prenotazione.

Con vincolo tra arrivo e arrivo della tabella prenotazione.

Con vincolo tra partenza e partenza della tabella prenotazione.

* SOGGIORNOTERMINATO (NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza)

Con vincolo tra NomeUtente e email della tabella prenotazione,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella prenotazione,

con vincolo tra indirizzo e indirizzo della tabella prenotazione,

con vincolo tra nomeHost e nomeHost della tabella prenotazione,

con vincolo tra arrivo e arrivo della tabella Prenotazione,

con vincolo tra partenza e partenza della tabella Prenotazione

* RECENSIONEAU(nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza,

visibilità, testo)

con vincolo tra nome e NomeHost della tabella SoggiornoTerminato,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella SoggiornoTerminato,

con vincolo tra indirizzo e indirizzo della tabella SoggiornoTerminato,

Con vincolo tra NomeUtente e NomeUtente della tabella SoggiornoTerminato,

con vincolo tra arrivo e arrivo della tabella SoggiornoTerminato,

con vincolo tra partenza e partenza della tabella SoggiornoTerminato

* RECENSIONEAH(nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, testoApp, testoHost, pulizia, comunicazione, posizione, qualità/prezzo, visibilità)

con vincolo tra nome e nomeUtente della tabella SoggiornoTerminato,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella SoggiornoTerminato,

con vincolo tra indirizzo e indirizzo della tabella SoggiornoTerminato,

Con vincolo tra nomeHost e nomeHost della tabella SoggiornoTerminato,

con vincolo tra arrivo e arrivo della tabella SoggiornoTerminato,

con vincolo tra partenza e partenza della tabella SoggiornoTerminato

* COMMENTOUTENTE(nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, timestamp, testo)

con vincolo tra nomeHost e Nome della tabella recensioneau,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella recensioneau,

con vincolo tra indirizzo e indirizzo della tabella recensioneau,

Con vincolo tra NomeUtente e NomeUtente della tabella recensioneau,

con vincolo tra arrivo e arrivo della tabella recensioneau,

con vincolo tra partenza e partenza della tabella recensioneau

* COMMENTOHOST(nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp, testo)

Con vincolo tra nomeUtente e nome della tabella recensioneah,

con vincolo tra nomeAlloggio e nomeAlloggio della tabella recensioneah,

con vincolo tra indirizzo e indirizzo della tabella recensioneah,

Con vincolo tra NomeHost e NomeHost della tabella recensioneah,

con vincolo tra arrivo e arrivo della tabella recensioneah,

con vincolo tra partenza e partenza della tabella recensioneah

* SOGGIORNICANCELLATIERIFIUTATI (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza, cancellato)

Con vincolo tra nomeUtente e nomeUtente della tabella prenotazione,

Con vincolo tra nomeAlloggio e nomeAlloggio della tabella prenotazione,

Con vincolo tra indirizzo e indirizzo della tabella prenotazione,

Con vincolo tra nomeHost e nomeHost della tabella prenotazione,

Con vincolo tra arrivo e arrivo della tabella prenotazione,

Con vincolo tra partenza e partenza della tabella prenotazione.

3. Implementazione

**3.1. DDL di creazione del database.**

begin;

drop table if exists UTENTE;

drop table if exists HOST;

drop table if exists LISTAPREFERITI;

drop table if exists ALLOGGIO;

drop table if exists PRENOTAZIONE;

drop table if exists SOGGIORNOTERMINATO;

drop table if exists RECENSIONEAU;

drop table if exists RECENSIONEAH;

drop table if exists COMMENTOU;

drop table if exists COMMENTOH;

drop table if exists NUMEROTELEFONO;

drop table if exists FOTOALLOGGIO;

drop table if exists SERVIZIALLGGIO;

create domain paymentmethodtype

as varchar(30) default NULL

check (value like 'paypal' or value like 'creditCard' or value like 'prepaidCard' or value like 'satispay' );

create domain boolean

as varchar(5) default 'false'

check (value like 'true' or value like 'false' );

create domain alloggioType

as varchar(20)

check (value like 'Appartamento' or value like 'stanza privata' or value like 'stanza condivisa' );

create table UTENTE (

email varchar(30) not null,

psw varchar(15),

nome varchar(15),

cognome varchar(15),

fotoCI varchar(50) default NULL, --URL immagine contuenuta in un altro DB

paymentMethod paymentmethodtype,

Constraint Utente\_PK primary key(Email)

);

Create table NUMEROTELEFONO(

email varchar(30),

telnum bigint,

Constraint NumeroTel\_PK primary key (Telnum)

);

create table HOST (

emailHost varchar(30) not null,

psw varchar(15),

nome varchar(15),

cognome varchar(15),

superHost boolean,

badge boolean,

Constraint Host\_PK primary key (emailHost)

);

create table ALLOGGIO (

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

emailHost varchar(30) not null,

tipoAlloggio alloggioType,

prezzoPerNotte decimal(5),

descrizione varchar(500),

costiPulizia decimal(3),

numeroLetti smallint,

checkIn smallint,

checkOut smallint,

ratingMedio smallint,

comune varchar(15),

Constraint Alloggio\_PK primary key (nomeAlloggio,indirizzo,emailHost),

Constraint Alloggio\_FK foreign key(emailHost) references HOST(emailHost) on delete Cascade

);

Create table FOTOALLOGGIO(

nomeAlloggio varchar(30),

Foto varchar(30) --URL DELLA FOTO

);

Create table SERVIZIALLGGIO(

nomeAlloggio varchar(30),

servizio varchar(15)

);

Create table LISTA(

email varchar(30) not null,

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

emailHost varchar(30) not null,

nomeLista varchar(20),

Constraint Lista\_PK primary key(email,nomeAlloggio,indirizzo,emailHost,nomeLista),

Constraint Lita\_FKU foreign key(email) references UTENTE(email) on delete cascade

);

create domain statusVal as smallint default 0

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

create table PRENOTAZIONE(

nomeUtente varchar(30),

nomeAlloggio varchar(20),

indirizzo varchar(20),

nomeHost varchar(30),

arrivo date,

partenza date,

costoTot decimal(5),

status statusVal,

constraint Prenotazione\_PK primary key (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza),

constraint Prenotazione\_FKU foreign key (nomeUtente) references UTENTE(email) on delete restrict,

constraint Prenotazione\_FKA0 foreign key (nomeAlloggio,indirizzo,nomehost) references ALLOGGIO(nomeAlloggio,indirizzo,emailhost) on delete restrict

);

create table OSPITE(

nomeUtente varchar(30) not null,

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

nomehost varchar(30) not null,

arrivo date not null,

partenza date not null,

emailospite varchar(30) not null,

nome varchar(15),

cognome varchar(15),

constraint Ospite\_PK primary key (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza, emailospite),

constraint Opsite\_FKU foreign key (nomeUtente,nomeAlloggio,indirizzo,nomeHost,arrivo,partenza) references PRENOTAZIONE(nomeUtente,nomeAlloggio,indirizzo,nomeHost,arrivo,partenza) on delete cascade

);

SOGGIORNOTERMINATO (NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza)

create table SOGGIORNOTERMINATO(

nomeUtente varchar(30) not null,

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

nomeHost varchar(30) not null,

arrivo date not null,

partenza date not null,

constraint SoggiornoTerminato\_PK primary key(NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza),

constraint SoggTerm\_FKP0 foreign key(nomeutente,nomeAlloggio,indirizzo,nomehost,arrivo,partenza) references PRENOTAZIONE(nomeutente,nomeAlloggio,indirizzo,nomehost,arrivo,partenza) on delete cascade

);

create table SOGGIORNINONTERMINATI(

nomeUtente varchar(30) not null,

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

nomeHost varchar(30) not null,

arrivo date not null,

partenza date not null,

status statusVal,

constraint SoggNonTerminato\_PK primary key (NomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza),

constraint SoggNonTerminato\_KF foreign key(nomeutente,nomeAlloggio,indirizzo,nomehost,arrivo,partenza) references PRENOTAZIONE(nomeutente,nomeAlloggio,indirizzo,nomehost,arrivo,partenza) on delete cascade

);

        RECENSIONEAU(nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, visibilità, testo)

create table RECENSIONEAU(

nome varchar(30) not null,

nomeAlloggio varchar(20) not null,

indirizzo varchar(20) not null,

nomeUtente varchar(30) not null,

arrivo date not null,

partenza date not null,

visibilità boolean,

testo varchar(400),

constraint RecensioneaU\_PK primary key (nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza),

constraint RecensioneaU\_FKST0 foreign key (nome,nomeAlloggio,indirizzo,nomeUtente,arrivo,partenza) references SOGGIORNOTERMINATO(nomeHost,nomeAlloggio,indirizzo,nomeUtente,arrivo,partenza) on delete cascade

);

RECENSIONEAH(nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, testoApp, testoHost,

pulizia, comunicazione, posizione, qualità/prezzo, visibilità)

create table RECENSIONEAH(

nome varchar(30) not null,

nomeHost varchar(30),

indirizzo varchar(20),

arrivo date,

partenza date,

nomeAlloggio varchar(20),

testoApp varchar(400),

testoHost varchar(400),

pulizia statusVal,

comunicazione statusVal,

posizione statusVal,

qualitàprezzo statusVal,

visibilità boolean,

constraint RecensioneAH\_PK primary key (nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio),

constraint RecensioneAH\_FKST0 foreign key (nome,nomeHost,nomeAlloggio,indirizzo,arrivo,partenza) references SOGGIORNOTERMINATO(nomeUtente,nomeHost,nomeAlloggio,indirizzo,arrivo,partenza) on delete cascade

);

create table COMMENTOUTENTE(

nomeHost varchar(30),

nomeAlloggio varchar(20),

indirizzo varchar(20),

nomeUtente varchar(30),

arrivo date,

partenza date,

tempo timestamp,

testo varchar(400),

constraint CommentoUtente\_PK primary key (nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza, tempo),

constraint CommentoUtente\_FK foreign key (nomeHost, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza) references Recensioneau(nome, nomeAlloggio, indirizzo, nomeUtente, arrivo, partenza)

);

        COMMENTOHOST(nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, timestamp, testo)

create table COMMENTOHOST(

nomeUtente varchar(30),

nomeHost varchar(30),

indirizzo varchar(20),

arrivo date,

partenza date,

nomeAlloggio varchar(20),

tempo timestamp,

testo varchar(400),

constraint CommentoHost\_PK primary key (nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio, tempo),

constraint CommentoHost\_FK foreign key (nomeUtente, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio) references RecensioneaH(nome, nomeHost, indirizzo, arrivo, partenza, nomeAlloggio)

);

commit;

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('edo@gmail.com','123','Edoardo','Di Leone',null,null);

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('mick@gmail.com','123','Mickael','ordine',null,'satispay');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('erika@gmail.com','123','Erika','Thedy',null,'paypal');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('guy@gmail.com','451','Guy','Montag',null,null);

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Abba@gmail.com','123','Abba','Musica',null,'creditCard');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Sony@gmail.com','123','Sony','Soft',null,'prepaidCard');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Muse@gmail.com','123','Muse','Musing',null,null);

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Asus@gmail.com','123','Asus','Game',null,'creditCard');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Steel@gmail.com','123','Steel','Series',null,'prepaidCard');

insert into utente(Email,psw,nome,cognome, fotoCI,paymentmethod) values ('Valmora@gmail.com','123','Valmora','Water',null,'creditCard');

insert into host(EmailHost, psw, nome, cognome, superHost, Badge) values ('Moggi@gmail.com','123','Luciano','Moggi',false,false);

insert into host(EmailHost, psw, nome, cognome, superHost, Badge) values ('Jacopo@gmail.com','123','Jacopo','Luoni',false,false);

insert into host(EmailHost, psw, nome, cognome, superHost, Badge) values ('Giacomo@gmail.com','123','Giacomo','Rossi',false,false);

insert into alloggio (nomeAlloggio, indirizzo, EmailHost, tipoAlloggio, prezzoPerNotte, descrizione, costiPulizia,

numeroLetti, checkin, checkout, ratingmedio, comune) values

('RostaResidence', 'Via Buttigliera 11', 'Moggi@gmail.com', 'Appartamento', 110,

'Stanza molto pulita con ottima vista sulla città di rosta, consigliamo vivamente a chi si voglia rilassare di soggiornare pacificamante nel nostro alloggio', 25, 2, 10, 12, null, 'Rosta');

insert into alloggio (nomeAlloggio, indirizzo, EmailHost, tipoAlloggio, prezzoPerNotte, descrizione, costiPulizia,

numeroLetti, checkin, checkout, ratingmedio, comune) values

('Rivoli Home Guest', 'Via gabriele 17', 'Jacopo@gmail.com', 'stanza privata', 180,

'Stanza con cui si può tranquillamente soggiornare a rivoli, molto tranquilla ed ospitale.',

20, 1, 8, 12, null, 'Rivoli');

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('mick@gmail.com', 'RostaResidence', 'Via Buttigliera 11', 'Moggi@gmail.com', '2022-10-11', '2022-10-29', 1980, 0);

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('erika@gmail.com', 'RostaResidence', 'Via Buttigliera 11', 'Moggi@gmail.com', '2022-10-30', '2022-11-2', 440, 0);

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('Asus@gmail.com', 'Rivoli Home Guest', 'Via gabriele 17', 'Jacopo@gmail.com', '2022-1-4', '2022-1-29', 4500, 0);

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('Abba@gmail.com', 'Rivoli Home Guest', 'Via gabriele 17', 'Jacopo@gmail.com', '2022-2-1', '2022-2-5', 540, 0);

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('Sony@gmail.com', 'Torino restroom', 'Via Umberto 45', 'Jacopo@gmail.com', '2022-10-11', '2022-10-29', 1620, 0);

insert into prenotazione (nomeUtente, nomeAlloggio, indirizzo, nomeHost, arrivo, partenza,costoTot, status) values

('Steel@gmail.com', 'Extra Lusso Roma', 'Via Torino 12', 'Moggi@gmail.com', '2022-12-25', '2023-1-2', 2400, 0);

/\*dopo aver effettuato l'update di tutti gli appartamenti terminati prima della data odierna, ho aggiunto il risultato delle query ai soggiorni terminati\*/

insert into soggiornoterminato values ('Asus@gmail.com','Rivoli Home Guest', 'Via gabriele 17', 'Jacopo@gmail.com', '2022-01-04', '2022-01-29');

insert into soggiornoterminato values ('Abba@gmail.com', 'Rivoli Home Guest', 'Via gabriele 17','Jacopo@gmail.com', '2022-02-01', '2022-02-05');

insert into recensioneau values ('Jacopo@gmail.com', 'Rivoli Home Guest', 'Via gabriele 17',

'Abba@gmail.com', '2022-02-01', '2022-02-05', 'false', 'Ospite molto cordiale, puntuale con gli orari di check-in e check-out,

la stanza era particolarmente pulita, quando vuoi tornare a rivoli, sai che il mio alloggio è disponibile, Cordiali Saluti');

/\*grazie al vincolo tra soggiorno terminato e recensione a h, saremo sicuri che la recensione sarà effettuata dall'utente che potrà avere i diritti di farlo\*/

insert into recensioneah values ('Abba@gmail.com', 'Jacopo@gmail.com','Via gabriele 17' ,'2022-02-01', '2022-02-05', 'Rivoli Home Guest','niente da dire, alloggio favoloso',

'mi è piaciuta soprattutto la disponibilità dell host, che era praticamente immediata, ci tornerò sicuramente',

4,5,4,4,'false');

insert into recensioneah values ('Asus@gmail.com', 'Jacopo@gmail.com','Via gabriele 17' ,'2022-01-04', '2022-01-29', 'Rivoli Home Guest','alloggio buono',

'forse era un periodo sbagliato, ma l host era irraggiungibile',

4,1,4,4,'false');

insert into commentohost values ('Asus@gmail.com', 'Jacopo@gmail.com', 'Via gabriele 17', '2022-01-04', '2022-01-29',

'Rivoli Home Guest', CURRENT\_TIMESTAMP, 'chiedo formalmente scusa per la mia assenza

durante il tuo pernottamento, se la prossima volta vuoi tornare, potrai usufruire di un 20% di costo che è valido

solo per te e per i tuoi ospiti, ancora scusa, cordialmente, Jacopo')

Operazioni di interrogazione

**Per diventare superhost:**

**Devono aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.**

select emailhost

from host H join prenotazione P ON (P.nomehost like H.emailhost) join soggiornoterminato ST ON(ST.nomehost = p.nomehost)

group by emailhost

having count(st.nomehost) > 10 and SUM(st.partenza::timestamp - st.arrivo::timestamp ) > interval '100' day

**Devono aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.**

select emailhost

from host H join prenotazione P ON (P.nomehost like H.emailhost) join soggiorninonterminati SNT ON(SNT.nomehost = p.nomehost)

join soggiornoterminato st on(p.nomehost = st.nomehost)

group by emailhost

having (count(snt.nomehost) \* count(st.nomehost))/100 < 0.1

**Devono aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.**

select H.emailhost

from host H join alloggio A on(h.emailhost = a.emailhost)

where ratingmedio > 4.8

group by H.emailhost

TUTTO IL MODELLO E-R

Immagine che contiene freccia

Descrizione generata automaticamente