

# **Basi di dati e sistemi informativi - Progetto SQL –**

## **Prof. Manuel Striani – Anno 2019/2020**

### **Componenti del gruppo:**

Nome: Marco

Cognome: Ascia

Numero di matricola: 20028741

E-mail: [20028741@studenti.uniupo.it](mailto:20028741@studenti.uniupo.it)

Nome: Emilio

Cognome: Pizzuto

Numero di matricola: 20030421

E-mail: [20030421@studenti.uniupo.it](mailto:20030421@studenti.uniupo.it)

**Basi di dati e Sistemi Informativi: Sperimentazioni A.A. 2019-2020**

## **PROGETTAZIONE CONCETTUALE**

### **1.1. Requisiti iniziali**

Si deve progettare una base di dati relativa a un sito di affitti di appartamenti. In particolare, si possono inserire, cercare, prenotare e recensire interi appartamenti o stanze in appartamenti condivisi. Per potere usufruire dei servizi gli utenti (*proprietari e clienti*) devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, email. Per prenotare una struttura i clienti devono selezionare un tipo di sistemazione e un intervallo di date. Quando un cliente, dopo aver visionato qualche struttura, decide di prenotare, scrive al proprietario che risponde con un numero di prenotazione attraverso cui l'utente (*cliente*) può fare modifiche alle sue scelte. La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno. I clienti possono rilasciare un commento su una struttura solo se ne hanno realmente usufruito. Gli utenti (*proprietari*) possono cambiare i dati delle strutture inserite di cui sono proprietari e rispondere sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio. Si tiene traccia nella base di dati sia delle risposte pubbliche (che vengono pubblicate sotto la recensione) sia di quelle private. Anche i proprietari degli appartamenti possono recensire gli ospiti (*clienti*) dopo il soggiorno. Alcuni proprietari possono essere classificati come superhost (si veda il file superhost.pdf). Degli utenti (*proprietari e clienti*) vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come host (*proprietario*) che come ospite (*cliente*), le eventuali proprietà e (se guadagnato) il simbolo

di superhost. Quando un cliente si cancella dal servizio, le sue recensioni e le risposte ricevute relativamente ad esse rimangono memorizzate mentre i suoi dati personali vengono cancellati. Le informazioni relative alle strutture sono reperibili nel documento alloggio.pdf. Per maggiori dettagli: <https://www.airbnb.it/>.

## 1.2. Glossario dei termini

Termine	Descrizione	Sinonimi	Collegamenti
<b>Utente</b>	Una persona che può inserire, cercare, prenotare, recensire strutture e rispondere a recensioni.	Cliente, Ospite	Struttura, Cliente, Prenotazione
<b>Struttura</b>	Immobile fornito da un proprietario a disposizione dei clienti.	Appartamento, Stanza, Proprietà, Alloggio	Proprietario, Prenotazione
<b>Proprietario</b>	Una persona che possiede delle strutture, delle quali può gestire i dati, messe a disposizione dei clienti, dei quali può scrivere delle recensioni.	Host	Struttura
<b>Prenotazione</b>	Atto che definisce il contratto stipulato tra utente e proprietario da una data a un'altra in una struttura.	/	Struttura, Utente
<b>Recensione</b>	Descrizione della struttura o dell'utente al fine di dare una valutazione.	Commento	Struttura, Prenotazione
<b>Risposta</b>	Commento a una recensione.	/	Recensione

## 1.3. Requisiti riscritti

Si deve progettare una base di dati relativa a un sito di affitti di appartamenti. In particolare, si possono inserire, cercare, prenotare e recensire ~~interi~~ appartamenti o stanze ~~in appartamenti condivisi~~. Per potere usufruire dei servizi gli utenti devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, email. Per prenotare una struttura ~~i clienti~~ **un utente** ~~devono~~ **deve** selezionare ~~un tipo di sistemazione~~ **una struttura** e un intervallo di date. Quando ~~un cliente~~ **un utente**, dopo aver visionato qualche struttura, decide di prenotare,

~~serve al~~ **contatta il** proprietario che risponde con un numero di prenotazione attraverso cui l'utente può fare modifiche alle sue scelte. La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno. ~~I clienti~~ **Gli utenti** possono ~~rilasciare un commento su~~ **recensire** una struttura solo se ne hanno realmente usufruito. ~~Gli utenti~~ **I proprietari** possono cambiare i dati delle **proprie** strutture ~~inserite di cui sono proprietari~~ e rispondere sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio. Si tiene traccia nella base di dati sia delle risposte pubbliche (che vengono pubblicate sotto la recensione) sia di quelle private. Anche i proprietari degli appartamenti possono recensire ~~gli ospiti~~ **gli utenti** dopo il soggiorno. Alcuni proprietari possono essere classificati come superhost (si veda il file superhost.pdf). Degli utenti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come ~~host~~ **proprietario** che come ~~ospite~~ **cliente**, le eventuali proprietà e (se guadagnato) il simbolo di superhost. Quando ~~un cliente~~ **un utente** si cancella dal servizio, le sue recensioni e le risposte ricevute relativamente ad esse rimangono memorizzate mentre i suoi dati personali vengono cancellati. Le informazioni relative alle strutture sono reperibili nel documento alloggio.pdf. Per maggiori dettagli: <https://www.airbnb.it/>.

## 1.4. Requisiti strutturati in gruppi di frasi omogenee

### Frasi di carattere generale

Si deve progettare una base di dati relativa a un sito di affitti di appartamenti.

La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno.

### Frasi relative a Utente e Proprietario

Quando un utente, dopo aver visionato qualche struttura, decide di prenotare, contatta il proprietario che risponde con un numero di prenotazione attraverso cui l'utente può fare modifiche alle sue scelte.

Dei proprietari e dei clienti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come proprietario che come cliente, le eventuali proprietà e (se guadagnato) il simbolo di superhost.

### Frasi relative a Utente

Per potere usufruire dei servizi gli utenti devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, email. Per prenotare una struttura un utente deve selezionare una struttura e un intervallo di date.

Gli utenti possono recensire una struttura solo se ne hanno realmente usufruito.

Degli utenti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come proprietario che come cliente, le eventuali proprietà e (se guadagnato) il simbolo di superhost. Quando un utente si cancella dal servizio, le sue recensioni e le risposte ricevute relativamente ad esse rimangono memorizzate mentre i suoi dati personali vengono cancellati.

### Frasi relative a Struttura

In particolare, si possono inserire, cercare, prenotare e recensire appartamenti o stanze.

Le informazioni relative alle strutture sono reperibili nel documento alloggio.pdf. Per maggiori dettagli: <https://www.airbnb.it/>.

### Frasi relative a Proprietario

I proprietari possono cambiare i dati delle proprie strutture e rispondere sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio.

Anche i proprietari degli appartamenti possono recensire gli utenti dopo il soggiorno. Alcuni proprietari possono essere classificati come superhost (si veda il file superhost.pdf).

#### Frase relative a Prenotazione

Per prenotare una struttura un utente deve selezionare una struttura e un intervallo di date.

Quando un utente, dopo aver visionato qualche struttura, decide di prenotare, contatta il proprietario che risponde con un numero di prenotazione attraverso cui l'utente può fare modifiche alle sue scelte.

#### Frase relative a Recensione

Gli utenti possono recensire una struttura solo se ne hanno realmente usufruito.

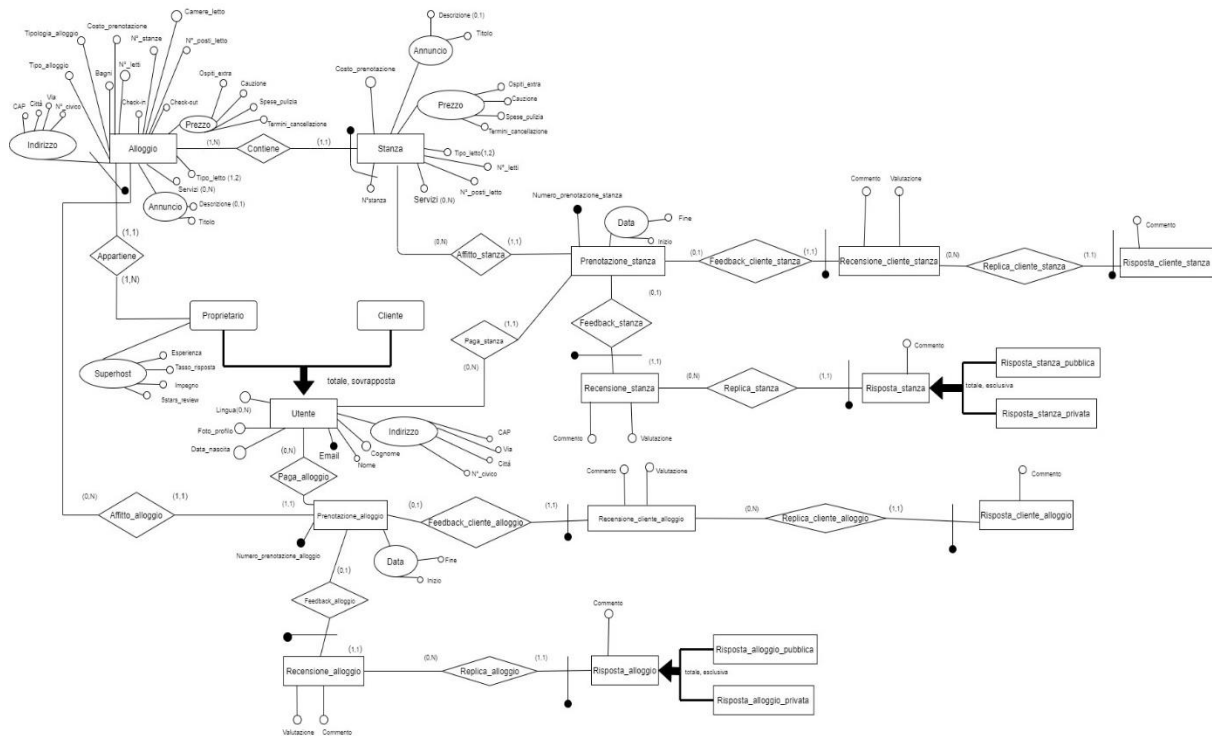
Anche i proprietari degli appartamenti possono recensire gli utenti dopo il soggiorno.

#### Frase relative a Risposta

I proprietari possono rispondere sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio.

Si tiene traccia nella base di dati sia delle risposte pubbliche (che vengono pubblicate sotto la recensione) sia di quelle private.

## 1.5. Schema E-R + regole aziendali



### Commenti sullo schema ER:

Per quanto concerne gli attributi composti, tra i quali “Indirizzo”, “Prezzo”, “Annuncio”, “Superhost” e “Data”, si deciderà nello schema ER finale di scomporli nei singoli attributi che li compongono.

Inoltre, abbiamo individuato dai requisiti iniziali il bisogno di attributi multivalore come le “Lingue parlate”, “Servizi” oppure i “Tipi di letto” di un alloggio o di una stanza, che verranno trasformati in entità (Reificazione di attributo in entità) nello schema ER finale.

### Pattern di progettazione:

- **Part-of:** Alloggio – Stanza dove l’esistenza di “Stanza” dipende dall’esistenza di ciò che la contiene, ossia “Alloggio”;
- **Evoluzione di concetto:** Proprietario – Utente dove in “Proprietario” vengono aggiunti degli attributi che, se soddisfatti, gli varranno il simbolo di “superhost”.

### Strategia di progetto utilizzata:

Per progettare il database abbiamo utilizzato la strategia di progettazione inside-out”, partendo inizialmente ad individuare i concetti principali del nostro DB (quelli citati nel

glossario) con i quali abbiamo realizzato uno schema “scheletro” e successivamente i loro legami e le loro relazioni.

In questo modo abbiamo individuato le entità come “Cliente”, “Proprietario, Prenotazione”, “Recensione” e “Risposta”.

Dopodiché abbiamo individuato quali di queste entità fossero riconducibili o legate a generalizzazioni, potendo quindi generalizzare, ad esempio, le “Risposte” in “Risposte pubbliche” e “Risposte private” oppure riassumere “Cliente” e “Proprietario” sotto un’unica generalizzazione “Utente”.

## Regole aziendali (business rules):

### Vincoli di Integrità:

RV	<concetto> deve/non deve <espressione>
1	Un alloggio non deve essere prenotato da un utente se dalla data di inizio della prenotazione fino alla data di fine prenotazione risulta già occupato.
2	Una stanza non deve essere prenotata da un utente se dalla data di inizio della prenotazione fino alla data di fine prenotazione risulta già occupata.
3	Una stanza deve appartenere a uno e un solo alloggio.
4	La data di inizio di una prenotazione non deve precedere con quella di fine prenotazione.
5	Ogni recensione deve essere sempre relativa a una prenotazione o un utente.
6	Un utente deve poter cancellare il proprio account dal sito; in questo caso i suoi dati personali verranno cancellati e rimarranno memorizzate solamente le recensioni e le risposte relative ad esse, tutto in forma anonima.
7	I proprietari devono poter cambiare i dati delle strutture inserite di cui sono proprietari.
8	Un utente deve poter modificare i dati di una prenotazione a lui associata.
9	La valutazione di ogni recensione deve avere un valore compreso tra [1,5].
10	Le risposte di tipo pubblico di un alloggio devono essere visibili sotto alla relativa recensione alloggio.
11	Le risposte di tipo pubblico di una stanza devono essere visibili sotto alla relativa recensione stanza.
12	Le risposte di tipo privato devono essere visibili esclusivamente dal mittente e dal destinatario.

### Derivazioni:

RD	<concetto> si ottiene <operazione>
1	Un utente ottiene lo status di proprietario possedendo un alloggio o una stanza.
2	Un proprietario ottiene lo status di superhost nel momento in cui soddisfa i seguenti requisiti: <ul style="list-style-type: none"><li>• Esperienza: completare almeno 10 viaggi in ogni loro alloggio in un anno</li><li>• Tasso_risposta: il tasso di risposta nei confronti dei clienti deve essere almeno del 90% (devono rispondere velocemente ad almeno 9 clienti su 10)</li><li>• 5stars_review: almeno l'80% delle recensioni che ricevono deve avere valutazione 5</li><li>• Impegno: devono cancellare delle prenotazioni solo in rari casi (effettua al massimo 15 cancellazione annue)</li></ul>



## AUTOVALUTAZIONE DELLO SCHEMA E-R:

- **CORRETTEZZA:** per definire lo schema non abbiamo considerato come verrà tradotto in un modello relazionale, evitando, per il momento, di inserire ID e codici laddove fosse possibile trovare un identificatore alternativo.

Inoltre, abbiamo indicato il tipo di ogni generalizzazione e non abbiamo dato lo stesso nome ad entità o associazioni. Le business rules presenti gestiscono concetti e informazioni non direttamente ricavabili dallo schema E-R.

- **COMPLETEZZA:** ogni informazione presente nei requisiti iniziali è stata gestita nello schema E-R e/o nelle business rules.
- **LEGGIBILITÀ:** lo schema ER risulta intuitivo poiché i nomi delle entità e delle associazioni rimandano immediatamente al concetto che esprimono e risultano di facile comprensione ad una persona esterna al progetto.
- **MINIMALITÀ:** sono presenti delle ridondanze relative ai servizi e ai tipi di letto di alloggi e stanze (attributi multivalore) che verranno gestite durante la ristrutturazione finale. Inoltre la numerosa presenza di identificatori esterni sarà scomposta in diversi identificatori surrogati interni alle varie entità.

Basi di dati e Sistemi Informativi: Sperimentazioni A.A. 2019-2020

## PROGETTAZIONE LOGICA

### 2.1. Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	10.000
Cliente	E	10.000
Proprietario	E	3.000
Alloggio	E	6.000
Prenotazione_alloggio	E	36.000
Recensione_alloggio	E	18.000
Risposta_alloggio	E	16.200
Risposta_alloggio_pubblica	E	12.960
Risposta_alloggio_privata	E	3.240
Recensione_cliente_alloggio	E	18.000

Risposta_cliente_alloggio	E	3.600
Stanza	E	18.000
Prenotazione_stanza	E	108.000
Recensione_stanza	E	54.000
Risposta_stanza	E	48.600
Risposta_stanza_pubblica	E	38.880
Risposta_stanza_privata	E	9.720
Recensione_cliente_stanza	E	54.000
Risposta_cliente_stanza	E	10.800
Appartiene	A	6.000
Affitto_alloggio	A	36.000
Paga_alloggio	A	36.000
Feedback_alloggio	A	18.000
Replica_alloggio	A	16.200
Feedback_cliente_alloggio	A	54.000
Replica_cliente_alloggio	A	3.600
Contiene	A	18.000
Affitto_stanza	A	108.000
Paga_stanza	A	108.000
Feedback_stanza	A	54.000
Replica_stanza	A	48.600
Feedback_cliente_stanza	A	54.000
Replica_cliente_stanza	A	10.800

Si suppone che la base di dati contenga una media di 10.000 utenti attivi, in particolare il numero di clienti è sempre leggermente superiore rispetto a quello dei proprietari poiché ci saranno più persone prenotanti rispetto alla disponibilità degli alloggi/stanze, in particolare il 30% degli utenti è anche un proprietario.

Siccome un proprietario può possedere più alloggi, si prevedono più alloggi rispetto al numero di proprietari, in particolare un proprietario possiede mediamente 2 alloggi.

Ogni alloggio viene prenotato circa 6 volte l'anno.

La recensione di ogni alloggio viene fatta da 1 persona su 2 che ne ha usufruito.

Un proprietario risponde almeno il 90% delle volte che riceve una recensione, motivato dal fatto di ottenere lo status di “superhost”.

Le risposte pubbliche sono privilegiate rispetto a quelle private perché il proprietario, nel bene o nel male della recensione, deve avere il buon senso di rendere pubblica la sua risposta affinché futuri clienti siano informati su eventuali eventi passati (80% pubbliche – 20% private)

Le recensioni dei clienti degli alloggi vengono fatte da 1 proprietario su 2, ovvero una recensione del cliente dell'alloggio ogni 2 prenotazioni alloggi, mentre un cliente risponde il 20% delle volte che viene recensito.

Ci saranno più stanze che alloggi. In particolare, un alloggio contiene mediamente 2 stanze (il bagno viene considerata una stanza a parte, infatti è attribuito di alloggio).

Una stanza viene solitamente prenotata per un breve arco di tempo (weekend), quindi le prenotazioni di stanze sono molto più frequenti rispetto alle prenotazioni degli alloggi.

Per quanto riguarda recensioni e risposte relativi alle stanze, si è deciso di mantenere la stessa proporzione che si è data agli alloggi, quindi:

- 50% recensioni stanze sulle prenotazioni delle stanze;
- 90% risposte stanze sulle recensioni delle stanze;
- 80% risposte pubbliche e 20% private sulle risposte stanze;
- 50% recensioni clienti sul numero totale di prenotazioni stanze;
- 20% risposte clienti sul numero delle recensioni del cliente.

La relazione “Appartiene” avrà una partecipazione tale al numero totale di alloggi presenti nel DB.

“Affitto alloggio” avrà la stessa partecipazione di “prenotazione alloggio”.

“Paga alloggio” avrà la stessa partecipazione di “affitto alloggio”, di conseguenza anche di “prenotazione alloggio”.

“Feedback alloggio” avrà la stessa partecipazione di “recensioni alloggio”.

“Replica alloggio” avrà la stessa partecipazione di “risposte alloggio”.

“Feedback cliente alloggio” avrà la stessa partecipazione di “recensione cliente alloggio”.

“Replica cliente alloggio” avrà la stessa partecipazione di “risposta cliente alloggio”.

“Contiene” avrà la stessa partecipazione di “stanze”.

Per quanto riguarda le relazioni di stanza, si ha la stessa partecipazione che si ha in alloggi, quindi:

- “Affitto stanza” avrà la stessa partecipazione di “prenotazione stanza”;

- “Paga stanza” avrà la stessa partecipazione di “affitto stanza”, di conseguenza anche di “prenotazione stanza”;
- “Feedback stanza” avrà la stessa partecipazione di “recensioni stanza”;
- “Replica stanza” avrà la stessa partecipazione di “risposte stanza”;
- “Feedback cliente stanza” avrà la stessa partecipazione di “recensione cliente stanza”;
- “Replica cliente stanza” avrà la stessa partecipazione di “risposta cliente stanza”;

#### **Osservazioni:**

- Il numero di “Clienti” equivale al numero di “Utenti” perché essere l’essere “Cliente” è uno status ottenuto in fase di registrazione;
- Per quanto riguarda il numero di “Prenotazione\_alloggio”, si è tenuto conto di 1 anno solare di prenotazioni nel complesso di tutti gli alloggi. Si considera che si prenotino tutte le stanze appartenenti a quell’alloggio;
- Si considera che il numero di stanze di un alloggio equivalga al numero di camere da letto dell’alloggio stesso;
- Per quanto riguarda il numero di “Prenotazione\_stanza”, si è tenuto conto di 1 anno solare di prenotazioni nel complesso di tutte le stanze. Si considera che una stanza venga prenotata 3 volte di più rispetto ad un alloggio. Ad esempio, un alloggio viene prenotato una sola volta per 7gg, mentre una stanza viene prenotata 3 volte, ma per 2gg.

## **2.2. Tavola delle operazioni**

<b>Operazione</b>	<b>Descrizione</b>	<b>Tipo</b>	<b>Frequenza</b>
1	Un utente prenota un alloggio	I	99 al giorno
2	Un utente scrive una recensione relativa ad una stanza	I	148 al giorno
3	Memorizza un nuovo alloggio con il nome del relativo proprietario	I	2 al giorno
4	Stampa tutti i dati di un proprietario	I	2 al giorno
5	Stampare tutte le recensioni di una stanza con relativa data di inizio e data di fine prenotazione	I	3 a settimana

6	Stampare tutte le prenotazioni di una stanza	I	1 a settimana
---	--	---	---------------

Solitamente è sufficiente considerare le operazioni principali per la regola dell' "80-20": l'80% del carico viene generato dal 20% delle operazioni, difatti sono state considerate solo parte delle operazioni possibili nel DB.

Tipo di operazione:

- I – interattiva: richiede dei parametri
- B – batch: è definita a priori

## Osservazioni:

- Operazione 1: la frequenza è stata determinata facendo "Numero\_prenotazioni\_alloggio"/365;
- Operazione 2: la frequenza è stata determinata facendo "Recensione\_stanza"/365;
- Operazione 3: ogni anno gli inserimenti degli alloggi aumentano del 10%, quindi se nel primo anno di esistenza nella BD ci sono 6.000 alloggi, dal secondo anno ce ne saranno 6.600, dal terzo 7.220 e così via. In questo modo, nel primo anno, verranno inseriti circa 2 alloggi al giorno, ciò è stato dedotto facendo il numero di alloggi aggiunti dopo un anno (10% di 6.000, ossia 600)/365;
- Operazione 4: definiamo di avere una BD che stampi tutti i dati di un proprietario con una frequenza di 2 volte al giorno;
- Operazione 5: definiamo di avere una BD che stampi tutti i dati delle recensioni di una stanza e relative data di inizio e di fine della sua prenotazione con una frequenza di 3 a settimana;
- Operazione 6: definiamo di avere una BD che stampi tutte le prenotazioni di una stanza con una frequenza di 1 a settimana.

## 2.3. Ristrutturazione dello schema E-R

### 2.3.1. Analisi delle ridondanze

#### 2.3.1.1. RIDONDANZA 1 (Alloggio, Proprietario, Utente, Appartiene)

##### OPERAZIONI COINVOLTE

- Operazione 3: memorizza un nuovo alloggio con il relativo proprietario
- Operazione 4: stampa tutti i dati di un proprietario

##### PRESENZA DI RIDONDANZA

Op3

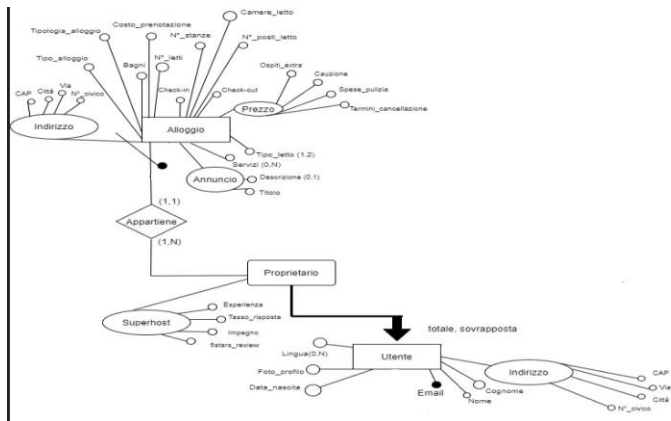


Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Alloggio	Entità	1	S	Memorizzo il nuovo alloggio
Appartiene	Associazione	1	S	Memorizzo la nuova coppia alloggio-proprietario
Proprietario	Entità	1	S	Memorizzo il suo proprietario
Utente	Entità	1	L	Cerco l'email (per attribuire la proprietà dell'alloggio al relativo utente, ossia il suo proprietario)

Un accesso in scrittura ha un costo maggiore di un accesso in lettura, perciò  $1S = 2L$  (1 scrittura vale 2 letture);

Costo:

- S:  $3(*2)*2 = 12$  accessi/giorno
- L:  $1*2 = 2$  accessi/giorno
- TOT: 14 accessi/giorno

## Op4

È stato aggiunto l'attributo "N°\_alloggi" in modo provvisorio per verificare la ridondanza, mentre nella nostra BD il numero di alloggi di un proprietario potrà essere calcolato tramite una query nel DML di modifica.

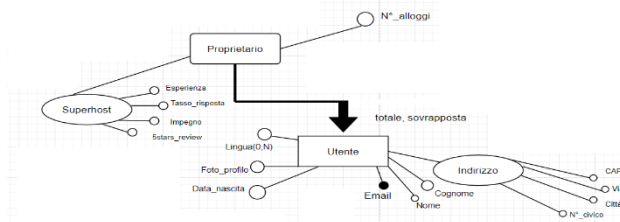


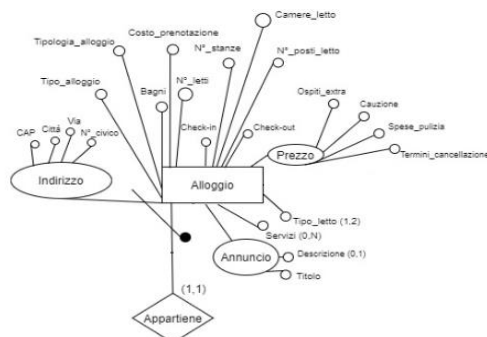
Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Proprietario	Entità	1	L	Cerco il proprietario d'interesse
Utente	Entità	1	S	Memorizzo tutti i dati di un proprietario

Costo:

- S:  $1(2) \cdot 2 = 4$  accessi/giorno
- L:  $1 \cdot 2 = 2$  accessi/giorno
- TOT: 6 accessi/giorno

## ASSENZA DI RIDONDANZA



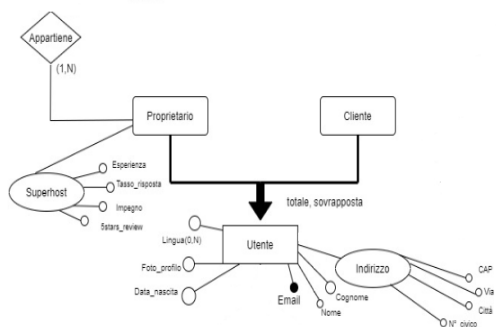
## Op3

Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Alloggio	Entità	1	S	Memorizzo il nuovo alloggio
Appartiene	Associazione	1	S	Memorizzo la nuova coppia alloggio-proprietario

Costo:

- S:  $2(*2)*2 = 8$  accessi/giorno
- L: 0 accessi/giorno
- TOT: 8 accessi/giorno



#### Op4

Concetto	Costrutto	Accessi	Tipo	Descrizione
Proprietario	Entità	1	L	Cerco il proprietario d'interesse
Utente	Entità	1	L	Memorizzo tutti i dati di un proprietario
Appartiene	Associazione	6000	L	Leggo le partecipazioni del proprietario nell'associazione

Costo:

- S: 0 accessi/giorno
- L:  $3*6000 = 18.000$  accessi/giorno
- TOT: 18.000 accessi/giorno
- **TOTALI PER RIDONDANZA 1**
  - **Presenza di ridondanza**



- **Spazio:** assumendo di usare 4 byte per memorizzare il numero di alloggi di ogni proprietario,  $4 \cdot 3000 \cdot 2 = 24.000$  byte aggiuntivi
- **Tempo:**
  - Operazione 3: 14 accessi/giorno
  - Operazione 4: 6 accessi/giorno
  - Totale di 20 accessi/giorno
- **Assenza di ridondanza**
  - **Spazio:** 0 byte aggiuntivi
  - **Tempo:**
    - Operazione 3: trascurabile (8 accessi/giorno)
    - Operazione 4: 18.000 accessi/giorno
    - Totale di 18.000 accessi/giorno
- **Decisione:** eliminando la ridondanza risparmieremmo 24.000 byte, ossia circa 24 KB, ma avremmo 17.800 accessi al giorno in più al DB. In questo caso, mantenere aggiornato il dato ridondante costa molto meno che ricalcolarlo, se assumiamo che preferiamo effettuare meno accessi e darci una limitazione spaziale nel database che ci consenta di aggiungere 24KB.

#### **2.3.1.1. RIDONDANZA 2 (Recensione\_stanza, Feedback\_stanza, Prenotazione\_stanza, Affitto\_stanza, Stanza, Contiene, Alloggio, Appartiene, Proprietario, Utente)**

##### **OPERAZIONI COINVOLTE**

- Operazione 5: stampare tutte le recensioni di una stanza con relativa data di inizio e data di fine prenotazione
- Operazione 6: definiamo di avere una BD che stampi tutte le prenotazioni di una stanza con una frequenza di 1 a settimana.

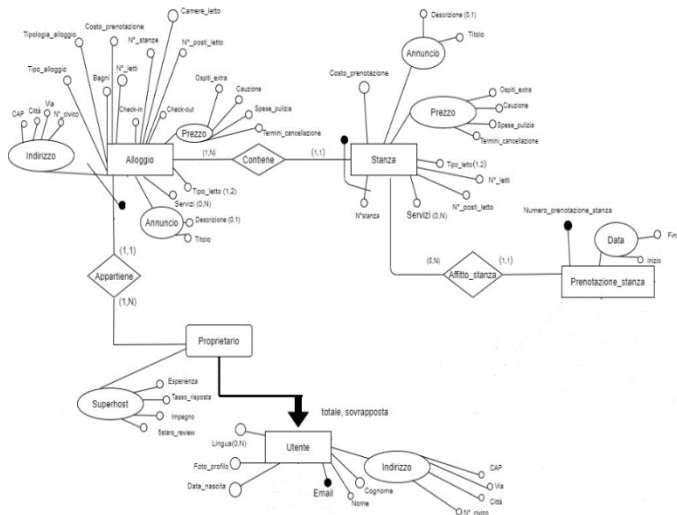
##### **PRESENZA DI RIDONDANZA**

Concetto	Costrutto	Accessi	Tipo	Descrizione
Recensione_stanza	Entità	1	S	Stampo tutte le recensioni della stanza d'interesse
Feedback_stanza	Associazione	1	L	Relazione tra la recensione e la prenotazione della stanza
Prenotazione_stanza	Entità	1	L	Cerco la data di inizio e la data di fine della relativa prenotazione
Prenotazione_stanza	Entità	1	S	Stampo la data di inizio e la data di fine della relativa prenotazione
Affitto_stanza	Associazione	2	L	Relazione tra la prenotazione e la stanza stessa
Stanza	Entità	1	L	Cerco la stanza d'interesse
Contiene	Associazione	1	L	Relazione tra l'alloggio e la stanza ad esso appartenente
Alloggio	Entità	1	L	Cerco l'alloggio a cui è associata la stanza
Appartiene	Associazione	1	L	Relazione tra l'alloggio e il relativo proprietario
Proprietario	Entità	1	L	Cerco il proprietario a cui è associato l'alloggio
Utente	Entità	1	L	Cerco l'email del proprietario a cui è associato l'alloggio, a cui a sua volta è associata la stanza e di conseguenza la prenotazione e la recensione

Si tiene conto del fatto che una stanza viene recensita il 50% delle volte in cui viene prenotata.

Costo:

- S:  $2(*2)*3 = 12$  accessi/settimana
- L:  $10*3 = 30$  accessi/settimana
- TOT: 42 accessi/settimana



## Op6

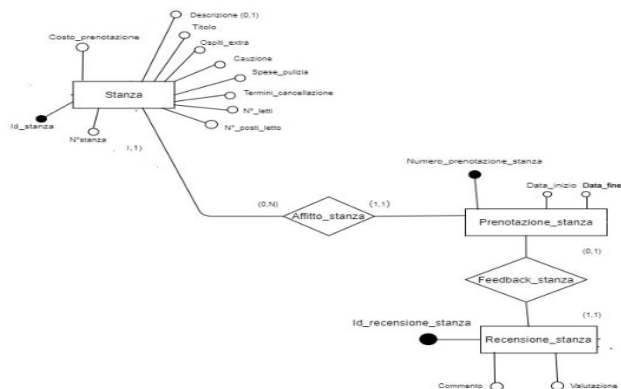
Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Prenotazione_stanza	Entità	1	S	Stampo tutte le prenotazioni relative alla stanza
Affitto_stanza	Associazione	1	L	Relazione tra la prenotazione e la stanza stessa
Stanza	Entità	1	L	Cerco la stanza d'interesse
Contiene	Associazione	1	L	Relazione tra l'alloggio e la stanza ad esso appartenente
Alloggio	Entità	1	L	Cerco l'alloggio a cui è associata la stanza
Appartiene	Associazione	1	L	Relazione tra l'alloggio e il relativo proprietario
Proprietario	Entità	1	L	Cerco il proprietario a cui è associata la stanza
Utente	Entità	1	L	Cerco l'email del proprietario a cui è associato l'alloggio, a cui a sua volta è associata la stanza e di conseguenza la prenotazione

Costo:

- S:  $1(*2)*1 = 2$  accessi/settimana
- L:  $7*1 = 7$  accessi/settimana
- TOT: 9 accessi/settimana

## ASSENZA DI RIDONDANZA



## Op5

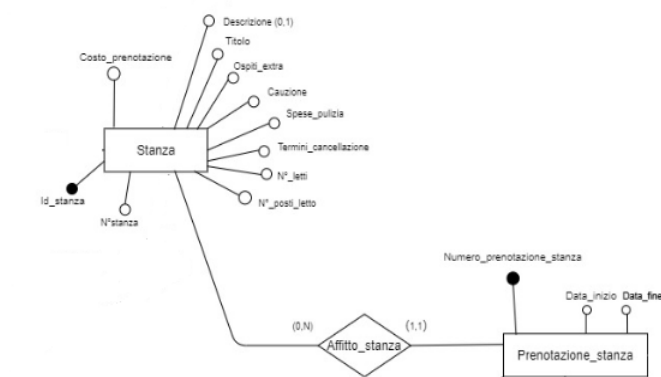
Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Recensione_stanza	Entità	1	S	Stampo tutte le recensioni della stanza d'interesse
Feedback_stanza	Associazione	54.000	L	Relazione tra la recensione e la prenotazione della stanza
Prenotazione_stanza	Entità	1	L	Cerco la data di inizio e la data di fine della relativa prenotazione
Prenotazione_stanza	Entità	1	S	Stampo la data di inizio e la data di fine della relativa prenotazione
Affitto_stanza	Associazione	108.000	L	Relazione tra la prenotazione e la stanza stessa
Stanza	Entità	1	L	Cerco la stanza d'interesse

Costo:

- S:  $2(*2)*3 =$  (trascurabili) 12 accessi/settimana
- L:  $(54.000+108.000)*2*3 = 972.000$  accessi/settimana

- TOT: 972.000 accessi/settimana



## Op6

Tavola degli accessi:

Concetto	Costrutto	Accessi	Tipo	Descrizione
Prenotazione_stanza	Entità	1	S	Stampo tutte le prenotazioni relative alla stanza
Affitto_stanza	Associazione	108.000	L	Relazione tra la prenotazione e la stanza stessa
Stanza	Entità	1	L	Cerco la stanza d'interesse

Costo:

- S:  $1(2) \cdot 1 =$  (trascurabili) 2 accessi/settimana
- L:  $108.000 \cdot 2 \cdot 1 = 216.000$  accessi/settimana
- TOT: 216.000 accessi/settimana

## • TOTALI PER RIDONDANZA 2

### • Presenza di ridondanza

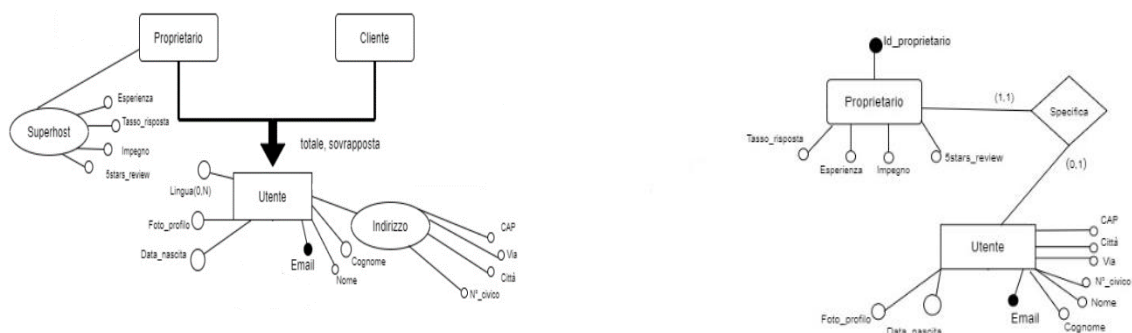
- **Spazio:** assumendo di usare 4 byte per memorizzare il numero di recensioni e prenotazioni di una stanza,  $4 \cdot 54.000 \cdot 108.000 = 23.328.000.000$  byte aggiuntivi
- **Tempo:**
  - Operazione 5: 42 accessi/settimana
  - Operazione 6: 9 accessi/settimana

- Totale di 51 accessi/settimana
- **Assenza di ridondanza**
  - **Spazio:** 0 byte aggiuntivi
  - **Tempo:**
    - Operazione 5: 972.000 accessi/settimana
    - Operazione 6: 216.000 accessi/settimana
    - Totale di 1.188.000 accessi/settimana
- **Decisione:** eliminando la ridondanza risparmieremmo 23.328.000.000 byte, ossia circa 23 GB e avremmo 1.137.000 accessi alla settimana in più al DB. In questo caso, eliminare il dato ridondante costa molto meno che ricalcolarlo, se assumiamo che preferiamo effettuare più accessi e darci una limitazione spaziale del nostro database, che in questo caso non aggiungerebbe un'ulteriore spesa in termini di spazio.

### 2.3.2. Eliminazione delle generalizzazioni

1. Utente → Proprietario & Cliente (totale, sovrapposta)
2. Risposta\_alloggio → Risposta\_alloggio\_pubblica & Risposta\_alloggio\_privata (totale, esclusiva)
3. Risposta\_stanza → Risposta\_stanza\_pubblica & Risposta\_stanza\_privata (totale, esclusiva)

#### 2.3.2.1. Generalizzazione (Utente -> Proprietario & Cliente)

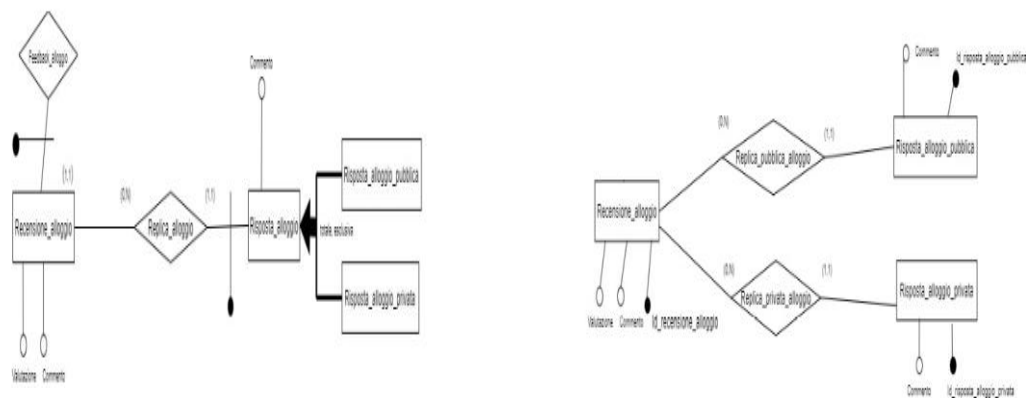


**Tecnica usata:** abbiamo deciso di decomporre la generalizzazione di tipo totale e sovrapposta accorpando l'entità figlia "Cliente" nel genitore poiché non possiede informazioni aggiuntive (specificazioni) rispetto all'entità padre "Utente", infatti ne è un sinonimo.

**Business rules:** tramite questa scomposizione si può dire che un utente ottiene lo status di proprietario possedendo un alloggio o una stanza (derivazione), mentre un proprietario ottiene lo status di superhost nel momento in cui soddisfa i seguenti requisiti:

- Esperienza: completare almeno 10 viaggi in ogni loro alloggio in un anno
- Tasso\_risposta: il tasso di risposta nei confronti dei clienti deve essere almeno del 90% (devono rispondere velocemente ad almeno 9 clienti su 10)
- 5stars\_review: almeno l'80% delle recensioni che ricevono deve avere valutazione 5
- Impegno: devono cancellare delle prenotazioni solo in rari casi

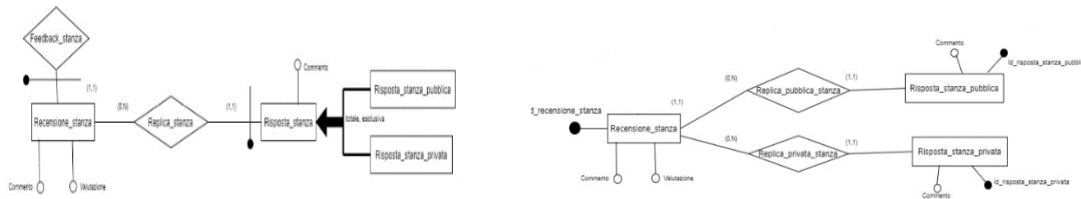
### 2.3.2.2. Generalizzazione (Risposta\_alloggio → Risposta\_alloggio\_pubblica & Risposta\_alloggio\_privata)



**Tecnica usata:** si è deciso di eliminare l'entità padre "Risposta\_alloggio" dalla generalizzazione totale ed esclusiva poiché risultava ridondante mantenerla memorizzata nella base di dati dato che era richiesto di fare distinzione tra risposte "pubbliche" e "private". In questo caso le entità figlie ereditano le relazioni e gli attributi dell'entità padre. Ciò è stato possibile perché la generalizzazione era a partecipazione totale.

**Business rules:** tramite questa decomposizione si ottiene un vincolo di integrità per cui le risposte di tipo pubblico di un alloggio devono essere visibili sotto alla relativa recensione alloggio.

### 2.3.2.3. Generalizzazione (Risposta\_stanza → Risposta\_stanza\_pubblica & Risposta\_stanza\_privata)



**Tecnica usata:** si è deciso di eliminare l’entità padre “Risposta\_stanza” dalla generalizzazione totale ed esclusiva poiché risultava ridondante mantenerla memorizzata nella base di dati dato che era richiesto di fare distinzione tra risposte “pubbliche” e “private”. In questo caso le entità figlie ereditano le relazioni e gli attributi dell’entità padre. Ciò è stato possibile perché la generalizzazione era a partecipazione totale.

**Business rules:** tramite questa decomposizione si ottiene un vincolo di integrità per cui Le risposte di tipo pubblico di una stanza devono essere visibili sotto alla relativa recensione stanza.

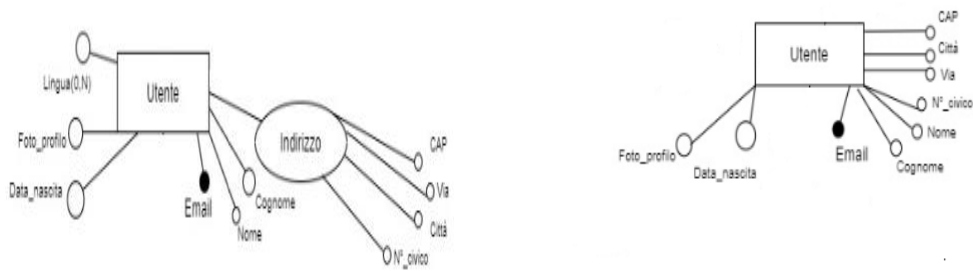
### 2.3.3. Partizionamento/accorpamento di entità e associazioni

1. L’attributo multivalore *lingua* di “Utente” diventa un’entità con un proprio identificatore e un attributo “Nome\_lingua”;
2. L’attributo multivalore *servizi* di “Alloggio” e di “Stanza” diventa un’entità con un proprio identificatore e con un attributo “Nome\_Servizio”;
3. L’attributo multivalore *tipo letto* di “Alloggio” e di “Stanza” diventa un’entità con un proprio identificatore e con un attributo “Nome”;
4. L’attributo composto *indirizzo* di “Alloggio” e di “Utente” viene scomposto in “Via”, “N°\_civico”, “CAP” e “Città”;
5. L’attributo composto *data* di “Prenotazione\_alloggio” e “Prenotazione\_stanza” viene scomposto in “Data\_inizio” e “Data\_fine”;
6. L’attributo composto *prezzo* di “Alloggio” e di “Stanza” viene scomposto in “Ospiti\_extra”, “Cauzione”, “Spese\_pulizia” e “Termini\_cancellazione”;
7. L’attributo composto *superhost* di “Proprietario” viene scomposto in “Esperienza”, “Tasso\_risposta”, “Impegno” e “5stars\_review”;





### 2.3.3.4. Eliminazione di attributi composti (Indirizzo)

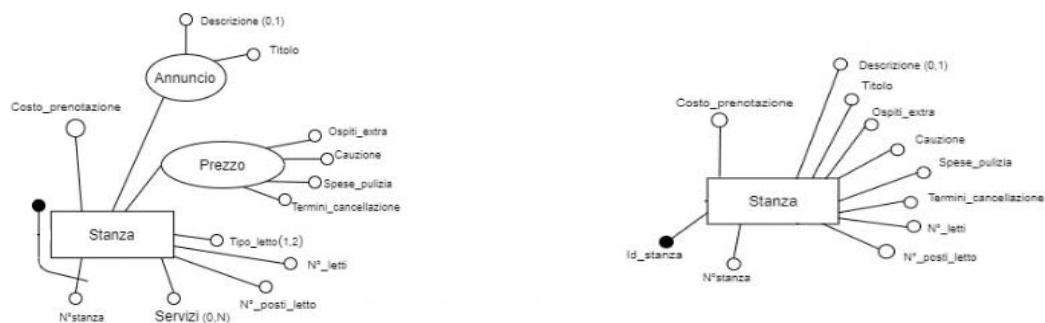


Questo tipo di partizionamento è necessario perché gli attributi composti non sono direttamente rappresentabili nel modello relazionale. Per risolvere il problema, vengono scomposti negli attributi che li compongono e restano collegati alla stessa entità a cui erano collegati in precedenza.

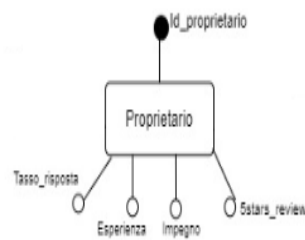
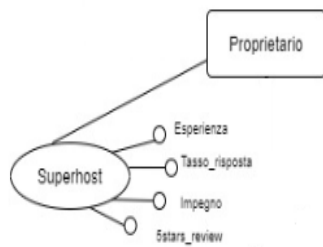
### 2.3.3.5. Eliminazione di attributi composti (Data)



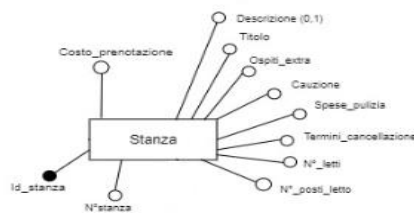
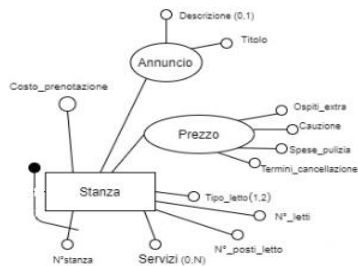
### 2.3.3.6. Eliminazione di attributi composti (Prezzo)



### 2.3.3.7. Eliminazione di attributi composti (Superhost)



### 2.3.3.8. Eliminazione di attributi composti (Annuncio)



**Osservazioni:** per quanto riguarda gli attributi opzionali, ad esempio “Descrizione” nella nostra BD, verranno rappresentati con un “\*” che segue il loro nome nel modello relazionale, perciò non serve gestirli in qualche modo particolare.

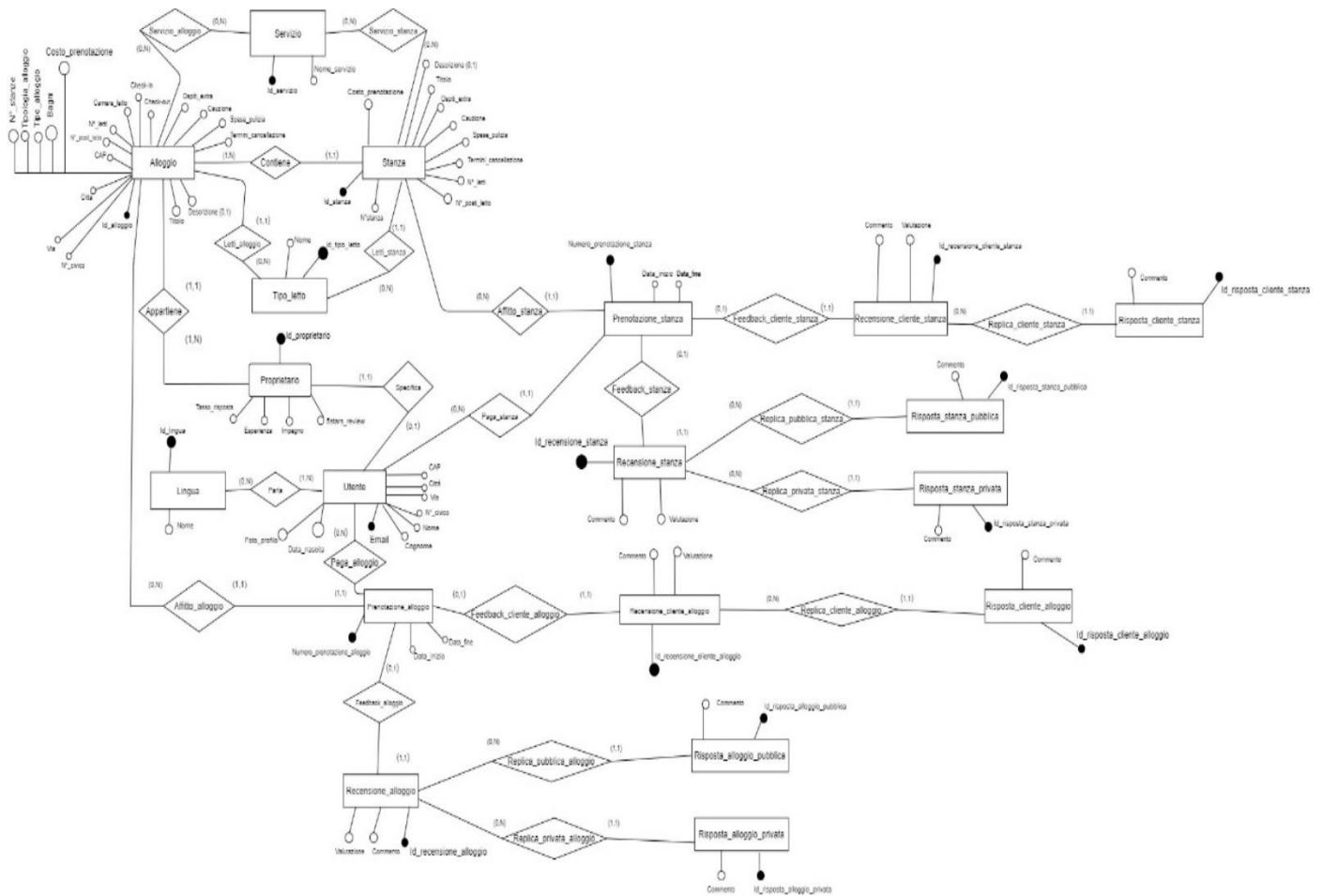
### 2.3.4. Scelta degli identificatori principali

Entità	Identificatore principale
Alloggio	Id_alloggio
Stanza	Id_stanza
Proprietario	Id_proprietario
Utente	Email
Servizio	Id_servizio
Lingue parlate	Id_lingua
Tipo letto	Id_tipo_letto
Prenotazione stanza	Numero_prenotazione_stanza
Prenotazione alloggio	Numero_prenotazione_alloggio

Recensione stanza	Id_recensione_stanza
Recensione alloggio	Id_recensione_alloggio
Recensione cliente stanza	Id_recensione_cliente_stanza
Recensione cliente alloggio	Id_recensione_cliente_alloggio
Risposta cliente stanza	Id_risposta_cliente_stanza
Risposta cliente alloggio	Id_risposta_cliente_alloggio
Risposta pubblica stanza	Id_risposta_stanza_pubblica
Risposta privata stanza	Id_risposta_stanza_privata
Risposta pubblica alloggio	Id_risposta_alloggio_pubblica
Risposta privata alloggio	Id_risposta_alloggio_privata

Le entità “Utente”, “Prenotazione stanza” e “Prenotazione alloggio” possiedono una loro chiave (superchiave minimale) data dai requisiti iniziali, mentre per tutte le altre entità abbiamo deciso di utilizzare identificatori “surrogati”, ipotizzandoli popolati da valori univoci, per semplificare la gestione del database e non avere una ridondanza dei medesimi dati su più tabelle. In particolare, avremmo avuto numerosi identificatori esterni che sarebbero stati composti da più attributi.

## 2.4 Schema E-R ristrutturato + regole aziendali



## Regole aziendali (business rules):

### Vincoli di Integrità:

RV1	<concetto> deve/non deve <espressione>
1	Un alloggio non deve essere prenotato da un utente se dalla data di inizio della prenotazione fino alla data di fine prenotazione risulta già occupato.
2	Una stanza non deve essere prenotata da un utente se dalla data di inizio della prenotazione fino alla data di fine prenotazione risulta già occupata.
3	Una stanza deve appartenere a uno e un solo alloggio.
4	La data di inizio di una prenotazione non deve precedere con quella di fine prenotazione.
5	Ogni recensione deve essere sempre relativa a una prenotazione o un utente.
6	Un utente deve poter cancellare il proprio account dal sito; in questo caso i suoi dati personali verranno cancellati e rimarranno memorizzate solamente le recensioni e le risposte relative ad esse, tutto in forma anonima.
7	I proprietari devono poter cambiare i dati delle strutture inserite di cui sono proprietari.
8	Un utente deve poter modificare i dati di una prenotazione a lui associata.
9	La valutazione di ogni recensione deve avere un valore compreso tra [1,5].
10	Le risposte di tipo pubblico di un alloggio devono essere visibili sotto alla relativa recensione alloggio.
11	Le risposte di tipo pubblico di una stanza devono essere visibili sotto alla relativa recensione stanza.
12	Le risposte di tipo privato devono essere visibili esclusivamente dal mittente e dal destinatario.

### Derivazioni:

RD1	<concetto> si ottiene <operazione>
1	Un utente ottiene lo status di proprietario possedendo un alloggio o una stanza.
2	Un proprietario ottiene lo status di superhost nel momento in cui soddisfa i seguenti requisiti: <ul style="list-style-type: none"><li>• Esperienza: completare almeno 10 viaggi in ogni loro alloggio in un anno</li><li>• Tasso_risposta: il tasso di risposta nei confronti dei clienti deve essere almeno del 90% (devono rispondere velocemente ad almeno 9 clienti su 10)</li><li>• 5stars_review: almeno l'80% delle recensioni che ricevono deve avere valutazione 5</li><li>• Impegno: devono cancellare delle prenotazioni solo in rari casi (effettua al massimo 15 cancellazione annue)</li></ul>

## 2.5 Schema relazionale con vincoli di integrità referenziale

<Utente(Email, Data\_nascita, Nome, Cognome, Foto\_profilo, Città, CAP, N°\_civico, Via)>

<Lingua(Id\_lingua, Nome)>

<Proprietario(Id\_proprietario, Esperienza, Tasso\_risposta, 5stars\_review, Impegno, Email)

Proprietario(Email) referencia Utente(Email)>

<Alloggio(Id\_alloggio, Titolo, Descrizione\*, Camere\_letto, N°\_stanze, N°\_posti\_letto, N°\_letti, Costo\_prenotazione, Ospiti\_extra, Spese\_pulizia, Cauzione, Termini\_cancellazione, Check-in, Check-out, Tipo\_alloggio, Tipologia\_alloggio, Bagni, CAP, Via, N°\_civico, Città, Tipo\_letto, Proprietario)

Alloggio(Proprietario) referencia Proprietario(Id\_proprietario)

Alloggio(Tipo\_letto) referencia Tipo\_letto(Id\_tipo\_letto)>

<Stanza(Id\_stanza, N°\_stanza, N°\_letti, Costo\_prenotazione, N°\_posti\_letto, Ospiti\_extra, Spese\_pulizia, Cauzione, Termini\_cancellazione, Descrizione\*, Titolo, Alloggio, Tipo\_letto)

Stanza(Alloggio) referencia Alloggio(Id\_alloggio)

Stanza(Tipo\_letto) referencia Tipo\_letto(Id\_tipo\_letto)>

<Tipo\_letto(Id\_tipo\_letto, Nome)>

<Servizio(Id\_servizio, Nome)>

<Prenotazione\_stanza(Numero\_prenotazione\_stanza, Data\_inizio, Data\_fine, Stanza, Utente)

Prenotazione\_stanza(Stanza) referencia Stanza(Id\_stanza)

Prenotazione\_stanza(Utente) referencia Utente(Email)>

<Prenotazione\_alloggio(Numero\_prenotazione\_alloggio, Data\_inizio, Data\_fine, Alloggio, Utente)

Prenotazione\_alloggio(Alloggio) referencia Alloggio(Id\_alloggio)

Prenotazione\_alloggio(Utente) referencia Utente(Email)>

<Recensione\_cliente\_stanza(Id\_recensione\_cliente\_stanza, Valutazione, Commento, Prenotazione\_stanza)

Recensione\_cliente\_stanza(Prenotazione\_stanza) referencia  
Prenotazione\_stanza(Numero\_prenotazione\_stanza)>

<Risposta\_cliente\_stanza(Id\_risposta\_cliente\_stanza, Commento, Recensione\_cliente\_stanza)

Risposta\_cliente\_stanza(Recensione\_cliente\_stanza) referencia  
Recensione\_cliente\_stanza(Id\_recensione\_cliente\_stanza)>

<Recensione\_stanza(Id\_recensione\_stanza, Valutazione, Commento, Prenotazione\_stanza)

Recensione\_stanza(Prenotazione\_stanza) referencia  
 Prenotazione\_stanza(Id\_prenotazione\_stanza)>

<Risposta\_stanza\_pubblica(Id\_risposta\_stanza\_pubblica, Commento, Recensione\_stanza)

Risposta\_stanza\_pubblica (Recensione\_stanza) referencia  
 Recensione\_stanza(Id\_recensione\_stanza)>

<Risposta\_stanza\_privata(Id\_risposta\_stanza\_privata, Commento, Recensione\_stanza)

Risposta\_stanza\_privata (Recensione\_stanza) referencia  
 Recensione\_stanza(Id\_recensione\_stanza)>

<Recensione\_cliente\_alloggio(Id\_recensione\_cliente\_alloggio, Valutazione, Commento, Prenotazione\_alloggio)

Recensione\_cliente\_alloggio(Prenotazione\_alloggio) referencia  
 Prenotazione\_alloggio(Numero\_prenotazione\_alloggio)>

<Risposta\_cliente\_alloggio(Id\_risposta\_cliente\_alloggio, Commento, Recensione\_cliente\_alloggio)

Risposta\_cliente\_alloggio(Recensione\_cliente\_alloggio) referencia  
 Recensione\_cliente\_alloggio(Id\_recensione\_cliente\_alloggio)>

<Recensione\_alloggio(Id\_recensione\_alloggio, Valutazione, Commento, Prenotazione\_alloggio)

Recensione\_alloggio(Prenotazione\_alloggio) referencia  
 Prenotazione\_alloggio(Id\_prenotazione\_alloggio)>

<Risposta\_alloggio\_pubblica(Id\_risposta\_alloggio\_pubblica, Commento, Recensione\_alloggio)

Risposta\_alloggio\_pubblica(Recensione\_alloggio) referencia  
 Recensione\_alloggio(Id\_recensione\_alloggio)>

<Risposta\_alloggio\_privata(Id\_risposta\_alloggio\_privata, Commento, Recensione\_alloggio)

Risposta\_alloggio\_privata(Recensione\_alloggio) referencia  
 Recensione\_alloggio(Id\_recensione\_alloggio)>

<Parla(Lingua, Utente)

Parla(Lingua) referencia Lingua(Id\_lingua)

<Servizio\_stanza(Servizio, Stanza)

Servizio\_stanza(Servizio) referencia Servizio(Id\_servizio)

Servizio\_stanza(Stanza) referencia Stanza(Id\_stanza)>

<Servizio\_alloggio(Servizio, Alloggio)

Servizio\_alloggio(Servizio) referencia Servizio(Id\_servizio)

Servizio\_alloggio(Alloggio) referencia Alloggio(Id\_alloggio)>



## **DDL E DML**

### **3 DDL di creazione del database**

Nella creazione del database abbiamo utilizzato i seguenti costrutti:

- CREATE: per creare il database e le tabelle;
- PRIMARY KEY(PK): per le chiavi primarie;
- SERIAL: utilizzato per definire un attributo incrementale (intero) in modo automatico; in particolare l'abbiamo utilizzato per gli identificatori surrogati. Purtroppo, a causa di un problema tecnico di PgAdmin4, ci siamo trovati degli identificatori che, anziché partire da 1, partivano da un altro valore, il che causa diversi problemi. La soluzione che abbiamo deciso di adottare è stata quella di utilizzare le sequenze. In particolare:
  - CREATE SEQUENCE NomeSequenza
  - INCREMENT BY 1
  - START WITH 1;
  - CREATE TABLE NomeTabella (
    - Attributo1 INTEGER DEFAULT nextval('NomeSequenza') PRIMARY KEY (include già il "NOT NULL");
- REFERENCES: serve per linkare le chiavi esterne (FK), ossia i vincoli di integrità referenziale;
- ON DELETE CASCADE: elimina a cascata;
- ON UPDATE CASCADE: aggiorna a cascata;
- CHECK: esegue controlli sui valori degli attributi;
- ON DELETE NO ACTION: in caso di eliminazione l'attributo non esegue nessuna azione;
- ON DELETE SET NULL: in caso di eliminazione l'attributo viene settato a valore nullo;
- Attributo "Foto\_profilo" di "Utente": abbiamo preferito inserirlo come tipo booleano poiché utilizzare tipi come "BYTEA" o altri dei quali non abbiamo approfondito durante il corso non ci sembrava una buona idea;

### **4 DML di popolamento di tutte le tabelle del database**

I dati da noi inseriti per il popolamento coprono tutti i casi possibili, in particolare quelli limite. Per quanto concerne i dati inseriti ai proprietari, sono stati inseriti in modo strategico affinché qualcuno di essi soddisfi i requisiti per essere superhost.

### **5 DML di modifica**

Operazioni della tabella delle operazioni:

1. Un utente prenota un alloggio;
2. Un utente scrive una recensione relativa ad una stanza;
3. Memorizza un nuovo alloggio con il nome del relativo proprietario;
4. Stampa tutti i dati di un proprietario;
5. Stampare tutte le recensioni di una stanza con relativa data di inizio e data di fine prenotazione;
6. Stampare tutte le prenotazioni di una stanza.

Operazioni svolte nel file "Ascia\_Pizzuto\_DMLUPD.sql":

- Creazione della vista "superhost" per visualizzare quali sono i proprietari che soddisfano i requisiti per essere superhost;
- Elimina la lingua 'Cinese';
- Operazione di verifica dei requisiti iniziali e relativi vincoli di integrità referenziale: quando un cliente si cancella dal servizio, le sue recensioni e le risposte ricevute relativamente ad esse rimangono memorizzate mentre i suoi dati personali vengono cancellati;
- Operazione di aggiornamento della chiave primaria ('Email') di una ennupla di un "Utente" verificando la sua modifica anche nella tabella di "Proprietario" grazie ai vincoli di integrità referenziale presenti tra "Proprietario" e "Utente" (in questo specifico è stato scelto un utente che era anche proprietario);
- Operazione di prenotazione per verificare che la data di fine prenotazione deve essere  $\geq$  (successiva temporalmente) della data di inizio prenotazione;
- Operazione di eliminazione di un proprietario (da utente, così verifico anche che si cancelli sia da utente sia da proprietario), in questo modo, secondo i vincoli di integrità referenziale da noi posti, verranno cancellate tutte le stanze e tutti gli alloggi da lui posseduti nella base di dati. Verranno cancellati l'alloggio con id "1", di cui è proprietario e, di conseguenza, le stanze con id "5" e "6" appartenenti al medesimo alloggio.