

# Progetto Laboratorio Base di Dati 21/22

Abourida Zakaria - 950120

## Progettazione Concettuale

### 1.1) Requisiti iniziali

#### Laboratorio Base di Dati 2021/22

##### Progetto di piattaforma di home booking

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).<sup>1</sup>

Gli utenti si registrano al servizio fornendo indirizzo email, password, nome, cognome, numero o numeri di telefono. Se l'utente fornisce la foto della carta d'identità, viene riconosciuto come verificato. Inoltre, l'utente deve indicare un metodo di pagamento per poter prenotare. Gli utenti possono essere ospiti o "host" ovvero possono a loro volta ospitare altri utenti del servizio in uno o più alloggi di loro proprietà. Inoltre gli "host" possono diventare "superhost" se soddisfano i seguenti requisiti:

- Devono aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.
- Devono aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.
- Devono aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.

Gli utenti superhost ricevono un badge sul loro profilo.

Gli alloggi sono descritti indicando un nome, l'indirizzo (visibile all'ospite solo quando la prenotazione è confermata, altrimenti è visibile solo il comune), una descrizione, il prezzo per notte per persona e i costi di pulizia, delle foto, i servizi (ad esempio, cucina, wi-fi, lavatrice, ecc.), numero di letti e orario di check-in e check-out oltre all'host a cui appartiene, il rating medio e il numero di recensioni (vedere Fig. 1).

Gli utenti possono aggiungere alcune case tra i preferiti. Gli utenti possono avere diverse liste, ad esempio in base al viaggio che vogliono compiere.

Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti. Se gli ospiti sono a loro volta utenti del servizio, se ne possono indicare i nominativi. La prenotazione deve essere confermata o rifiutata dall'host. La prenotazione ha un costo totale e se confermata viene eseguito il pagamento. Inoltre, la prenotazione può essere cancellata sia dall'ospite che dall'host.

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

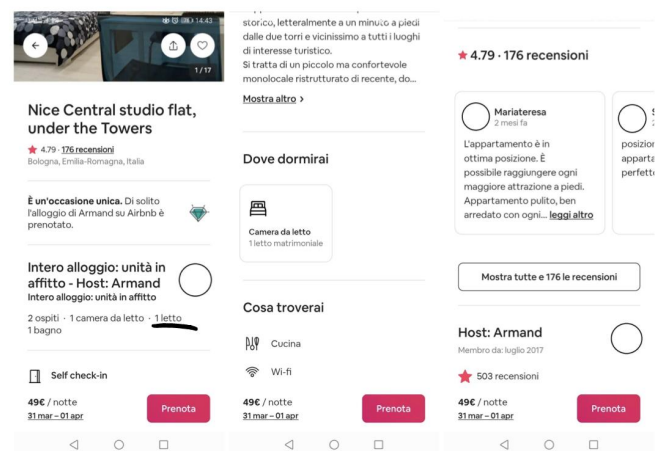


Fig. 1: Dettagli di un appartamento.

Al termine del soggiorno, gli ospiti e gli host si possono valutare a vicenda. La recensione fatta dagli ospiti comprende due testi (uno per l'appartamento e uno per l'host) e una serie di punteggi in una scala da 1 a 5 su dimensioni come pulizia, comunicazione, posizione, qualità/prezzo. La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sulle singole dimensioni. Le recensioni degli host comprendono solo un commento testuale. Le recensioni possono essere visibili o non visibili. Diventano visibili quando entrambi hanno fatto la recensione oppure se uno dei due non ha fatto la recensione, l'altra diventa visibile dopo 7 giorni dalla fine del soggiorno. Gli host e gli ospiti possono commentare più volte le review in cui sono coinvolti, creando un thread di discussione. Le recensioni sono visibili sui profili degli utenti suddivise in base a quelle ricevute come ospite e come host.

La base di dati deve supportare le seguenti operazioni:

- Una volta a settimana viene effettuato un calcolo per aggiornare il tasso di cancellazione di ciascun host.
- Una volta al giorno si controllano le condizioni per la qualifica di superhost e viene aggiornato lo status degli host.
- Una volta al mese viene calcolata la classifica degli alloggi più graditi.

## 1.2) Glossario dei termini

<b>TERMINE</b>	<b>DESCRIZIONE</b>	<b>SINONIMI</b>	<b>COLLEGAMENTI</b>
ALLOGGIO	Alloggi da prenotare	ABITAZIONE	PRENOTAZIONE, LISTA, HOST
UTENTE	Persone che possono prenotare e/o affittare	CLIENTE, CONSUMATORE	LISTA, PRENOTAZIONE
PRENOTAZIONE	Contiene i dati della prenotazione dell'utente X per l'alloggio Y	IMPEGNO, RISERVA	UTENTE, ALLOGGIO, RECENSIONE
RECENSIONE	Contiene le recensioni di host e ospite	REVISIONE, CRITICA	PRENOTAZIONE
LISTA	Contiene tutti gli alloggi preferiti e salvati	ELENCO, CATALOGO	UTENTE, ALLOGGIO
OSPITE	Utente che intende prenotare un alloggio	STRANIERO, FORESTIERO, TURISTA	UTENTE, RECENSIONE_OSPITE
HOST	Utente che mette in affitto il/i propri alloggi/o	PROPRIETARIO, AFFITTUARIO	UTENTE, ALLOGGIO, RECENSIONE_HOST
APPARTAMENTO	Tipo di alloggio	CASA	ALLOGGIO
STANZA_PRIVATA	Tipo di alloggio	CAMERA PRIVATA	ALLOGGIO
STANZA_CONDIVISA	Tipo di alloggio	CAMERA CONDIVISA	ALLOGGIO
RECENSIONE_OSPITE	Commento e valutazioni da parte dell'ospite		RECENSIONE, OSPITE
RECENSIONE_HOST	Commento da parte dell'host		RECENSIONE, HOST
P_IN_CORSO	Stato della prenotazione in corso di elaborazione		PRENOTAZIONE
P_CONFERMATA	Stato della prenotazione confermata		PRENOTAZIONE
P_CANCELLATA	Stato della prenotazione cancellata		PRENOTAZIONE

P_ARCHIVIATA	Stato della prenotazione archiviata		PRENOTAZIONE
--------------	-------------------------------------	--	--------------

### 1.3) Requisiti rivisti e strutturati in gruppi di frasi omogenee

#### **Frasi per “ALLOGGIO”:**

- ALLOGGI possono essere: APPARTAMENTI, STANZA PRIVATE o STANZE CONDIVISE.
- Gli ALLOGGI sono descritti indicando: nome, indirizzo, descrizione, il prezzo per notte per persona, i costi di pulizia, foto, servizi, numero di letti, orario di check-in e check-out, host a cui appartiene, il rating medio e il numero di recensioni.
- Valori aggiuntivi dall'immagine:
  - o Numero camere da letto
  - o Organizzazione letti nelle camere da letto
  - o Numero di ospiti massimo

#### **Frasi per “UTENTE”:**

- Gli UTENT\* si registrano al servizio fornendo: email, password, nome, cognome, numero o numeri di telefono.
- Se 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 essere OSPITI o HOST.
- Gli UTENTI possono prenotare degli ALLOGGI di qualsiasi tipo.
- Valori aggiunti ( per completezza):
  - o iban

#### **Frasi per “LISTA”:**

- Gli UTENTI possono aggiungere alcuni ALLOGGI tra i preferiti.
- Gli UTENTI possono avere diverse LISTE.

#### **Frasi per “HOST”:**

- Gli HOST possono a loro volta ospitare altri UTENTI del servizio in uno o più ALLOGGI di loro proprietà.
- Gli HOST possono diventare “superhost” se soddisfano dei requisiti.
- Gli UTENTI “superhost” ricevono un badge sul loro profilo.

#### **Frasi per “PRENOTAZIONE”:**

- 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 UTENTI del servizio, si possono indicare i nominativi.
- La PRENOTAZIONE deve essere CONFERMATA o rifiutata(CANCELLATA) dall'HOST.
- La PRENOTAZIONE ha un costo totale e se confermata viene eseguito il pagamento.
- La PRENOTAZIONE può essere CANCELLATA sia dall'OSPITE che dall'HOST.

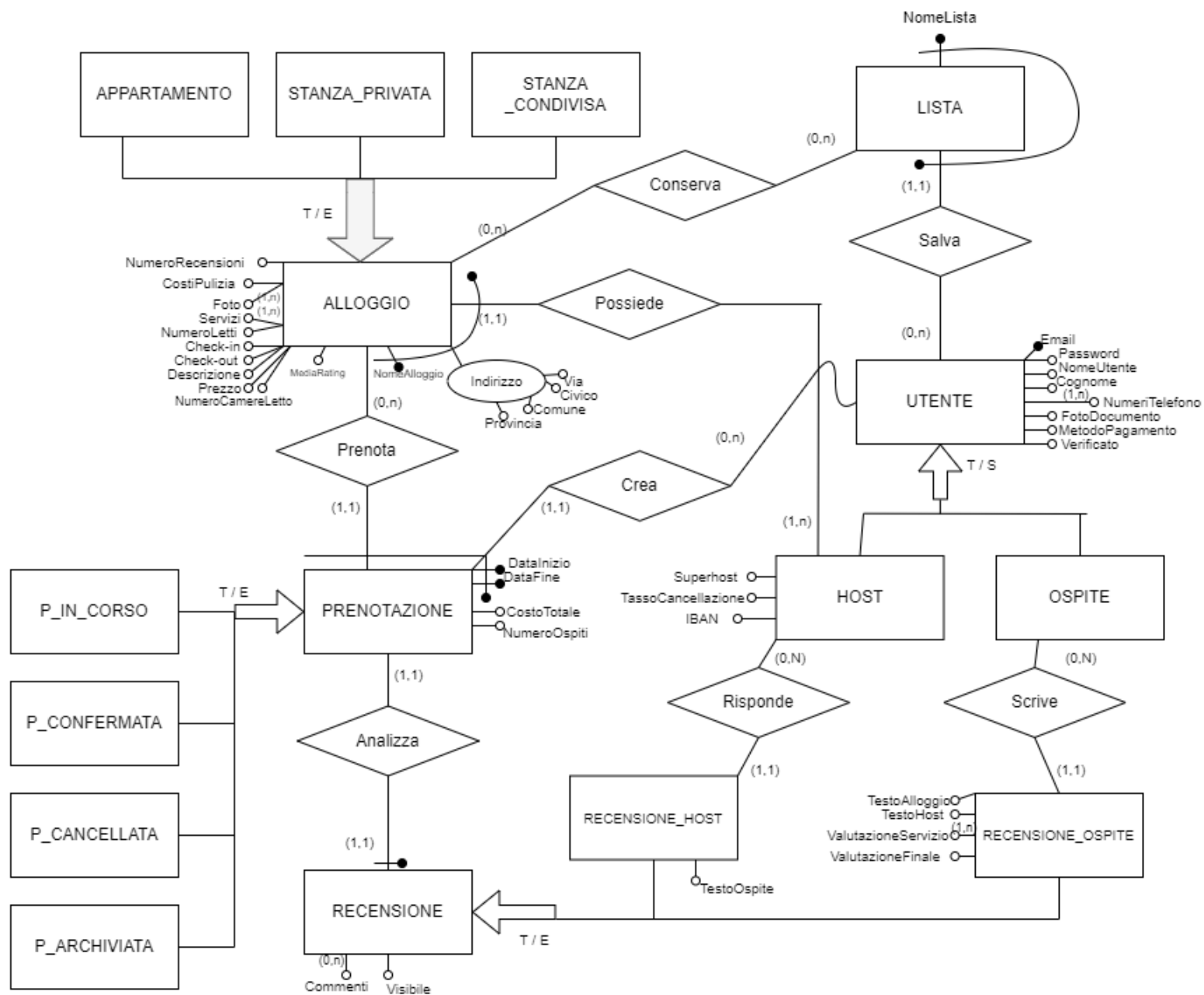
#### **Fraasi per "RECENSIONE":**

- Al termine del soggiorno gli OSPITI e gli HOST possono valutarsi a vicenda.
- La RECENSIONE fatta dagli OSPITI comprende due testi, uno per l'ALLOGGIO e uno per l'HOST (con una serie di punteggi in scala da 1 a 5 su varie dimensioni).
- La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sulle singole dimensioni.
- Le RECENSIONI degli HOST comprendono solo un commento testuale.
- Le RECENSIONI possono essere visibili o non visibili.
- Le RECENSIONI diventano visibili quando entrambi hanno fatto la RECENSIONE, altrimenti se ne manca una essa 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.

## 1.4) Schema E/R principale + Business Rules

### **BUSINESS RULES:**

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



- L'utente deve indicare un intervallo di date ed il numero di ospiti per poter prenotare.
- Un utente per essere un Superhost deve:
  - aver completato almeno 10 soggiorni (100 notti in totale)
  - aver conservato un tasso di cancellazione dell'1% al massimo (1 cancellazione ogni 100 soggiorni)
  - la valutazione complessiva di almeno 4,8 per tutte le proprietà.
- La prenotazione dev'essere confermata o cancellata dall'host.
- La prenotazione può essere cancellata anche dall'ospite.

- Se la prenotazione viene accettata allora viene eseguito il pagamento.
- Una volta che la prenotazione viene confermata o cancellata, alla fine del soggiorno viene archiviata.
- L'indirizzo diventa visibile solo se la prenotazione viene confermata, altrimenti è visibile solo il comune della proprietà.
- Se gli ospiti nella prenotazione sono utenti del servizio se ne possono indicare i nominativi.
- Solo al termine del soggiorno l'utente e l'host possono fare le rispettive recensioni.
- La recensione diventa visibile solo se l'ospite e l'host hanno fatto entrambi la corrispettiva recensione.
- La valutazione complessiva è una media delle valutazioni ricevute sulle singole dimensioni.
- Nella prenotazione la data d'inizio non può essere antecedente o coincidere con la data di fine.
- Negli alloggi l'ora di check-out deve precedere l'orario di check-in.

## **REGOLE DI DERIVAZIONE:**

- Il tasso di cancellazione si calcola facendo il quoziente tra il numero di prenotazioni archiviate di un host e le sue prenotazioni cancellate ( $\text{n. prenotazioni archiviate} / \text{n. prenotazioni cancellate}$ )
- La verifica avviene se l'utente carica una foto della carta d'identità.
- La media del rating si calcola facendo la media con le valutazioni delle recensioni degli ospiti per alloggio.
- Il numero di recensioni si calcola dalle prenotazioni archiviate.
- Il costo totale della prenotazione si calcola sommando i costi di pulizia e il prezzo del soggiorno a notte per ciascun ospite.
- La valutazione finale si calcola facendo la media tra tutte le valutazioni dell'ospite date sulle dimensioni di un determinato alloggio (servizi...).

# Progettazione Logica

## 2.1) Tavola dei volumi



<b>CONCETTO</b>	<b>TIPO</b>	<b>VOLUM E</b>	<b>DESCRIZIONE</b>
UTENTE	E	10'000'000	Stima degli utenti registrati tra ospiti e host
LISTA	E	6'000'000	Stima delle liste create dagli utenti
ALLOGGIO	E	4'500'000	Si stima che ogni host abbia almeno 1/2 alloggio
PRENOTAZIONE	E	14'000'000	Si stima circa 2 soggiorni per ogni ospite
RECENSIONE	E	12'000'000	Si stima circa 1,5 recensioni per ogni soggiorno
OSPITE	E	7'000'000	Si stima che il 70% degli utenti siano ospiti
HOST	E	3'000'000	Si stima che il 30% degli utenti siano host
APPARTAMENTO	E	1'800'000	Si stima che il 40% degli alloggi siano appartamenti
STANZA_ PRIVATA	E	1'575'000	Si stima che il 35% degli alloggi siano stanze private
STANZA_ CONDIVISA	E	1'125'000	Si stima che il 25% degli alloggi siano stanze private
P_IN_CORSO	E	76'712	Si stima circa 2 prenotazioni all'anno per ogni utente(in 24h)
P_ CONFERMATA	E	11'200'000	Si stima che l'80% delle prenotazioni vengono confermate
P_ CANCELLATA	E	1'400'000	Si stima che il 10% delle prenotazioni vengono cancellate
P_ ARCHIVIATA	E	14'000'000	Si stima che in media ci siano 2 soggiorni per ospite all'anno
RECENSIONE_ OSPITE	E	6'600'000	Si stima che il 55% delle recensioni siano da parte degli ospiti
RECENSIONE_ HOST	E	5'400'000	Si stima che il 45% delle recensioni siano da parte degli host
Salva	A	6'000'000	Stima della creazione delle liste
Conserva	A	12'000'000	Si stima che ogni lista contenga almeno 2 alloggi
Possiede	A	4'500'000	Si stima che ogni host abbia almeno 1/2 alloggio
Crea	A	14'000'000	Si stima circa 2 soggiorni per ogni ospite

Prenota	A	14'000'000	Si stima circa 2 soggiorni per ogni ospite
Analizza	A	12'000'000	Si stima circa 1,5 recensioni per ogni soggiorno
Scrive	A	6'600'000	Si stima che il 55% delle recensioni siano da parte degli ospiti
Risponde	A	5'400'000	Si stima che il 45% delle recensioni siano da parte degli host

## 2.2)Tavola delle operazioni

<b>OPERAZIONE</b>	<b>DESCRIZIONE</b>	<b>TIPO</b>	<b>FREQUENZA</b>
1	Inserimento nuovo utente	<b>I</b>	27'397 al giorno
2	Inserimento nuovo alloggio	<b>I</b>	12'328 al giorno
3	Inserimento nuovo Host	<b>I</b>	8'219 al giorno
4	Aggiungo un alloggio ad	<b>I</b>	16'438 al giorno
5	Inserimento prenotazione	<b>I</b>	38'356 al giorno
6	Aggiornamento dati utente	<b>I</b>	3'000 al giorno
7	Aggiornamento dati alloggio	<b>I</b>	3'000 al giorno
8	Aggiornamento dati	<b>I</b>	5'000 al giorno
9	Inserimento recensione da	<b>I</b>	18'082 al giorno
10	Inserimento recensione da	<b>I</b>	14'794 al giorno
11	Inserimento commento	<b>I</b>	2'000 al giorno
12	Conferma prenotazione	<b>I</b>	30'684 al giorno
13	Cancella prenotazione	<b>I</b>	3'835 al giorno
14	Calcolo del Tasso di	<b>B</b>	1 alla settimana
15	Controllo condizioni	<b>B</b>	1 al giorno
16	Calcolo dell'alloggio più	<b>B</b>	1 al mese
17	Controllo utente verificato	<b>B</b>	3'000 al giorno
18	Verifica presenza metodo di	<b>B</b>	3'000 al giorno
19	Calcolo della Media del	<b>B</b>	3'000 al giorno
20	Calcolo del numero di	<b>B</b>	3'000 al giorno

### **Motivazioni scelte dei dati:**

Sono state inserite le operazioni principali quali di inserimento, aggiornamento e calcolo di eventuali attributi. La frequenza è stata calcolata basandosi sui numeri presenti nella tavola dei volumi e preso spunto da vari siti di report presi su internet.

## 2.3) Ristrutturazione dello schema E/R

### 2.3.1) Analisi delle ridondanze

Le ridondanze trovate sono:

1. Verificato in UTENTE;
2. TassoDiCancellazione in HOST;
3. MediaRating in ALLOGGIO;
4. NumeroRecensioni in ALLOGGIO;
5. CostoTotale in PRENOTAZIONE;
6. ValutazioneFinale in RECENSIONE\_OSPITE.

Ho deciso di tenerle tutte perché sono esplicitamente richieste dal testo e ritengo che in questo modo si possa essere in grado di accedere alle informazioni direttamente e più velocemente.

**Ridondanza analizzata:** Verificato in UTENTE.

#### VERIFICATO:

Tavola Accessi Operazione n°1: Inserimento nuovo utente

<b>CONCETTO</b>	<b>COSTRUTTO</b>	<b>ACCESSI</b>	<b>TIPO</b>
UTENTE	ENTITA'	1	S

Tavola Accessi Operazione n°6: Aggiornamento dati utente

<b>CONCETTO</b>	<b>COSTRUTTO</b>	<b>ACCESSI</b>	<b>TIPO</b>
UTENTE	ENTITA'	1	S

Tavola Accessi Operazione n°17: Controllo utente verificato

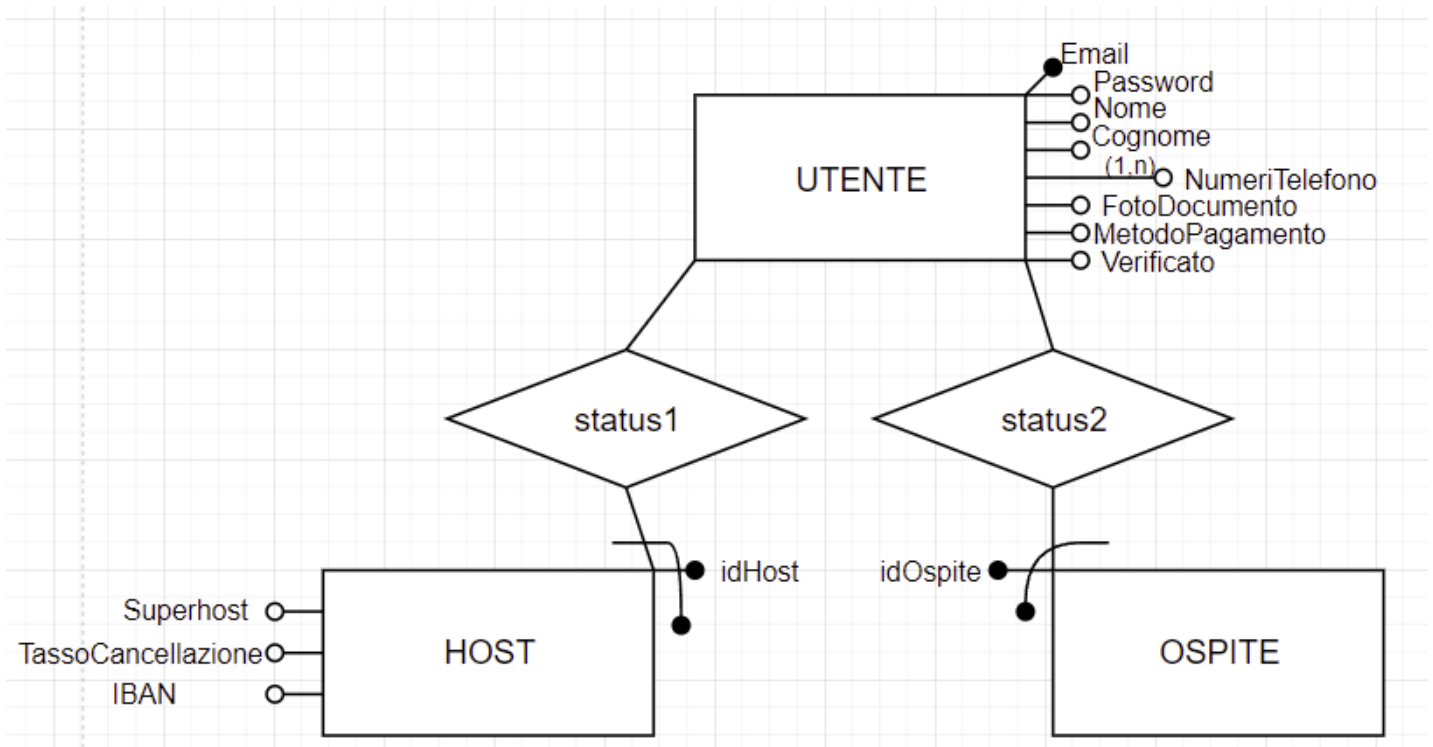
<b>CONCETTO</b>	<b>COSTRUTTO</b>	<b>ACCESSI</b>	<b>TIPO</b>
UTENTE	ENTITA'	1	S

### 2.3.2) Eliminazione delle generalizzazioni

Le generalizzazioni sono 4:

#### 1. UTENTE → HOST, OSPITE

(3<sup>a</sup> eliminazione)



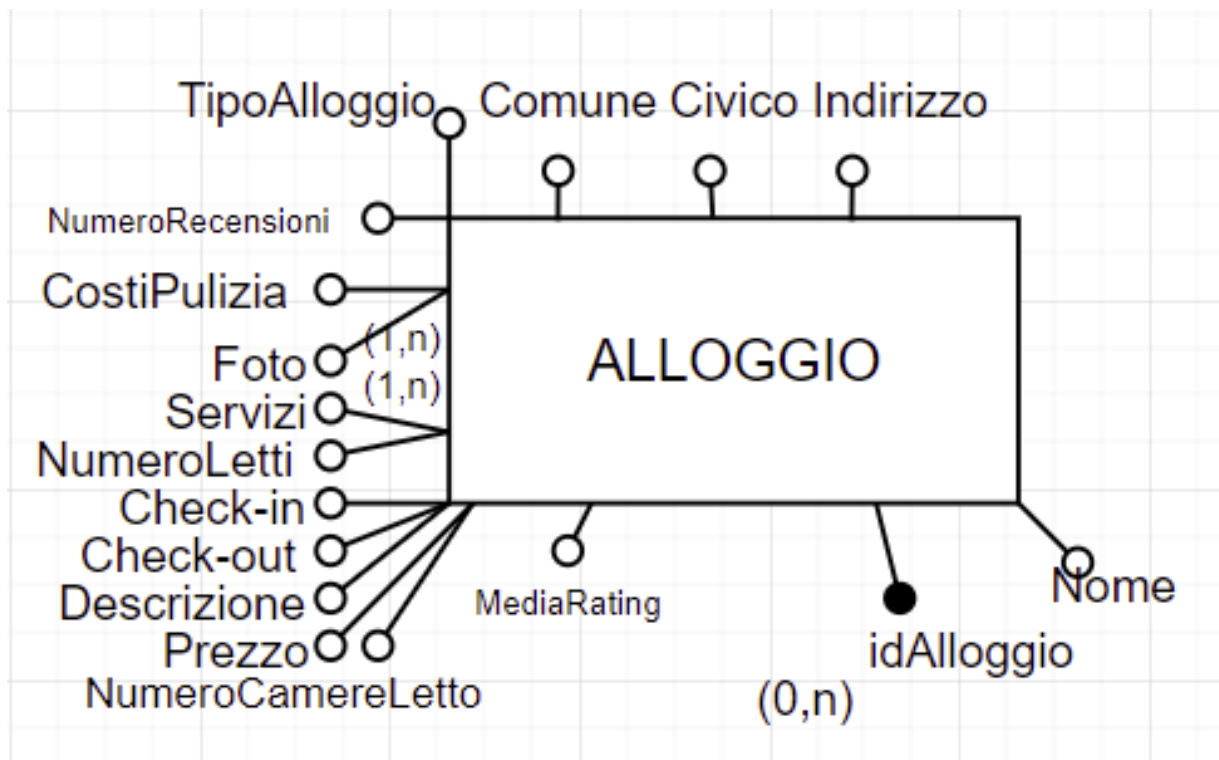
Aggiunte Business Rules:

- Ogni occorrenza di UTENTE deve partecipare a HOST oppure OSPITE;
- Ogni occorrenza di UTENTE può partecipare contemporaneamente a HOST e OSPITE.

**~Motivazione:** Ho eliminato in questo modo la generalizzazione perché secondo me si risparmia spazio e s'ottiene un accesso più semplice e generale a host e ospite. In questo modo non ci saranno duplicazioni di dati in caso un host sia anche un ospite.

#### 2. ALLOGGIO → APPARTAMENTO, STANZA\_PRIVATA, STANZA\_CONDIVISA

(1<sup>a</sup> eliminazione)



Aggiunte Business Rules:

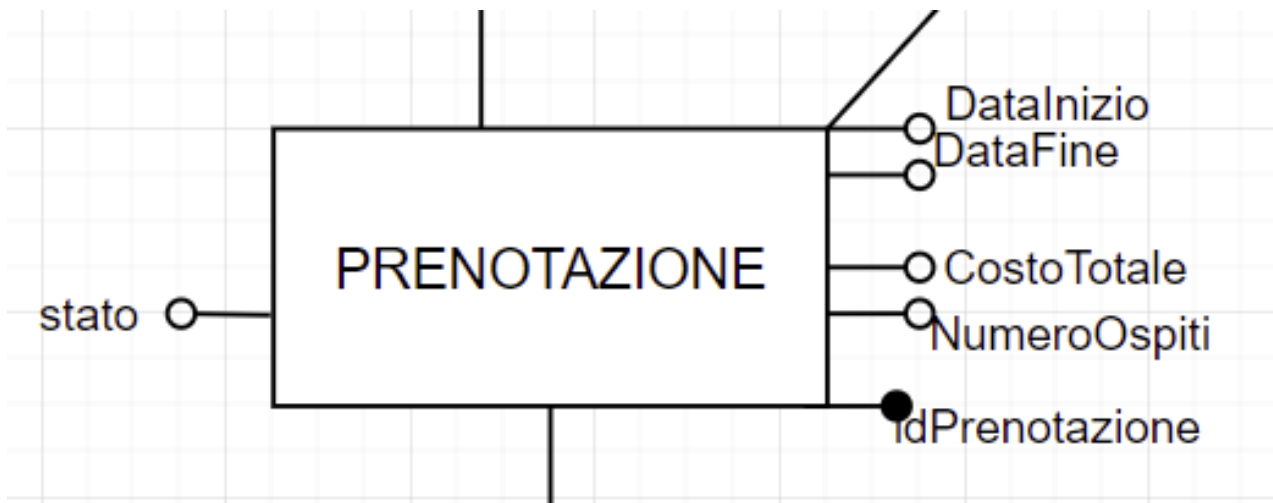
- Il tipo di ALLOGGIO può essere solo tra appartamento, stanza privata e stanza condivisa.

**~Motivazione:** Mantenere le entità figlie sarebbe stato solo uno spreco di spazio di dati con valori nulli dato che le operazioni non facevano un granché distinzioni fra le entità figlie. In questo modo s'otterrà una sola tabella relazionale.

3. PRENOTAZIONE → P\_IN\_CORSO, CONFERMATA, CANCELLATA, ARCHIVIATA  
(1^ eliminazione)

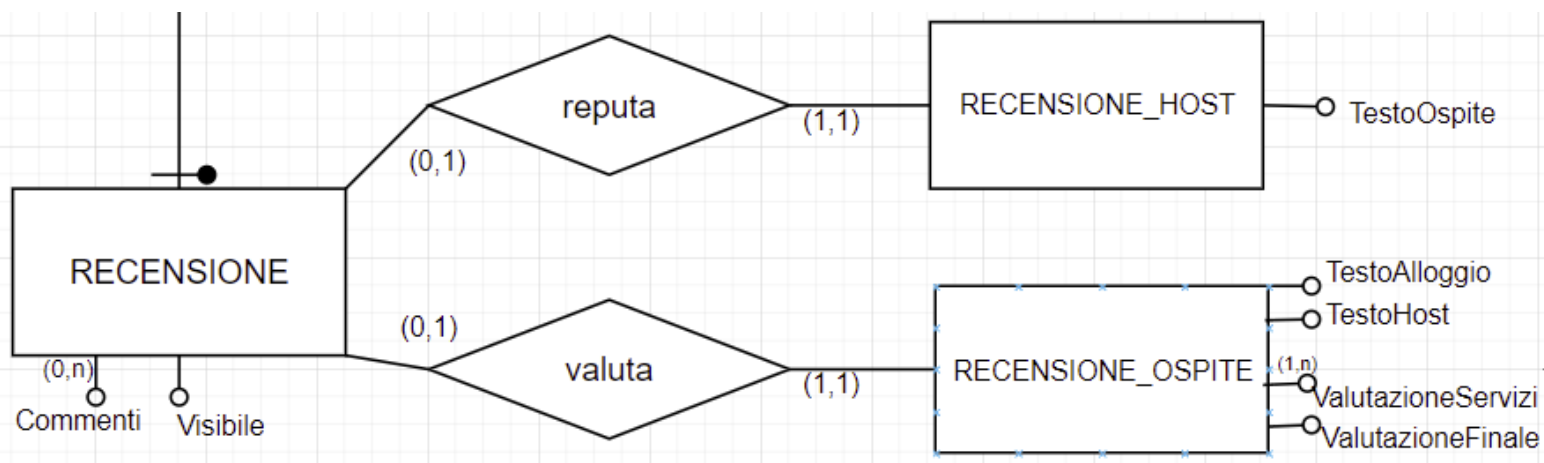
Aggiunte Business Rules:

- Lo stato della PRENOTAZIONE può essere solo inCorso, Confermata, Cancellata o Archiviata.



**~Motivazione:** Mantenendo le entità separate comporterebbe un costo notevole perché si andrebbe a visualizzare ogni entità se la prenotazione sia in corso, cancellata, confermata o archiviata. Accorpendo tutte le entità figlie nel padre s'aggiunge un attributo "Stato" che indica una delle entità figlie. Vengono anche aggiunti vincoli sulla correttezza delle prenotazioni.

#### 4. RECENSIONE → RECENSIONE\_OSPITE, RECENSIONE\_HOST (3<sup>a</sup> eliminazione)



Aggiunte Business Rules:

- Ogni occorrenza di RECENSIONE deve partecipare a RECENSIONE\_OSPITE e RECENSIONE\_HOST.
- Ogni occorrenza di RECENSIONE può essere di RECENSIONE\_HOST e/o RECENSIONE\_OSPITE.

**~Motivazione:** Ho deciso di mantenere le due entità figlie perché trovo sia più comodo ed efficiente poter visualizzare le Recensioni partendo dall'entità padre. In questo modo all'interno dell'entità padre ci saranno presenti le recensioni sia degli Host che degli Ospiti anche nell'eventualità che uno dei due, o nessuno, non abbia fatto la recensione (oltre al fatto che è meglio tenere i "commenti" nell'entità padre e non sdoppiarli nelle entità figlie).

### 2.3.3) Eventuale partizionamento/accorpamento di entità o associazioni

Non ho ritenuto che sia necessario nessun partizionamento o accorpamento di entità o associazioni.

### 2.3.5) Attributi multivalore e composti

#### **Attributi multivalore:**

- OSPITE(numeriTelefono);
- ALLOGGIO(Foto);
- ALLOGGIO(Servizio);
- RECENSIONE(Commenti);
- RECENSIONE\_OSPITE(ValutazioneServizio);

#### **Attributi composti:**

- ALLOGGIO(Indirizzo);

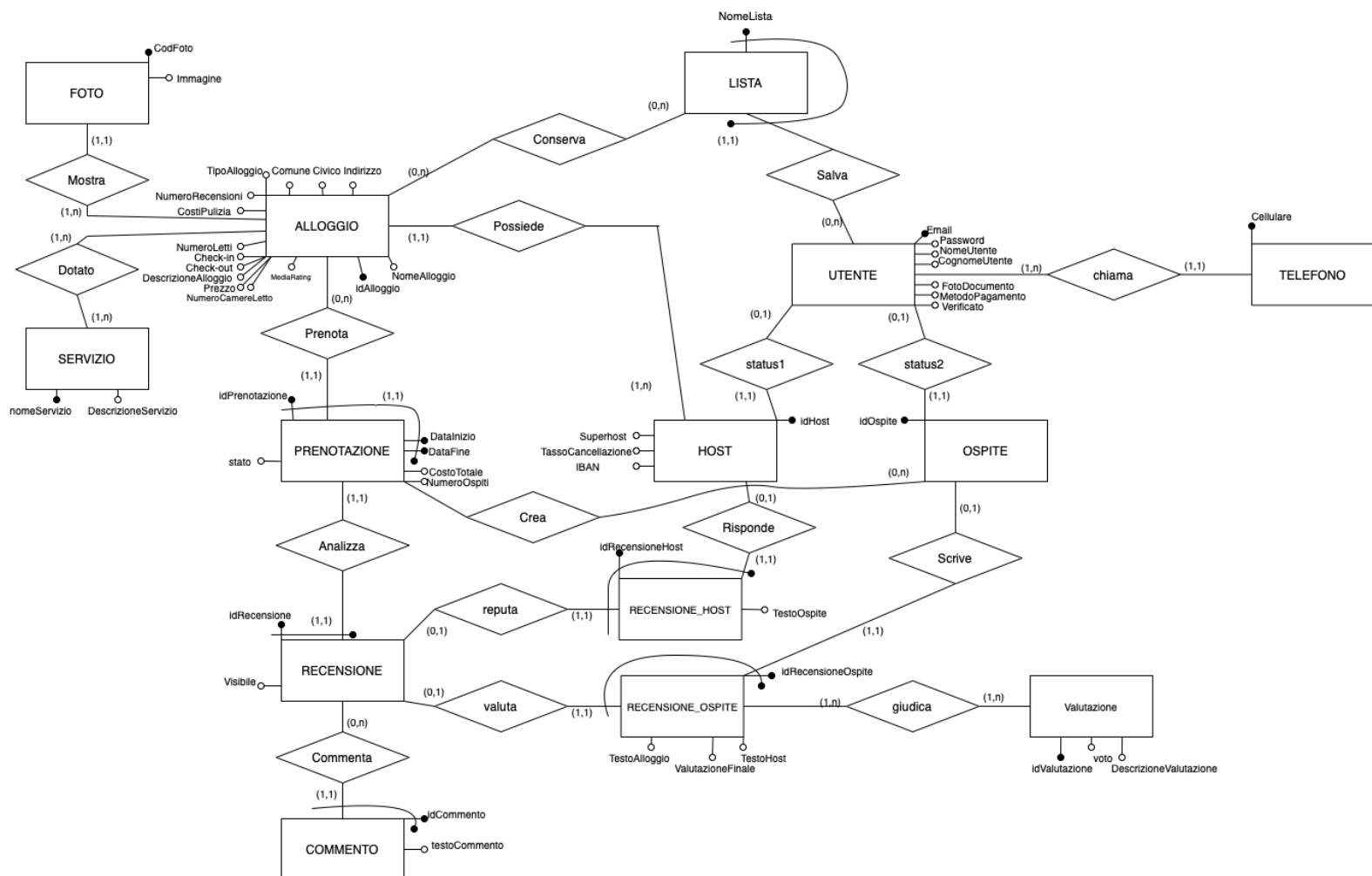
(Per quest'attributo composto ho deciso di sostituirlo con l'aggiunta di \*Comune, \*Civico e \*Indirizzo)

### 2.3.4) Eventuale scelta degli identificatori principali

- ★ **UTENTE:** email
- ★ **TELEFONO:** Cellulare
- ★ **HOST:** idHost
- ★ **OSPITE:** idOspite
- ★ **LISTA:** nomeLista, UTENTE(email)
- ★ **ALLOGGIO:** idAlloggio
- ★ **FOTO:** CodFoto
- ★ **SERVIZIO:** nomeServizio

- ★ **PRENOTAZIONE:** idPrenotazione, ALLOGGIO(idAlloggio), DataInizio, DataFine
- ★ **RECENSIONE:** idRecensione, PRENOTAZIONE(idPrenotazione)
- ★ **COMMENTO:** idCommento, RECENSIONE(idRecensione)
- ★ **RECENSIONE\_HOST:** idRecensioneHost, RECENSIONE(idRecensione), HOST(idHost)
- ★ **RECENSIONE\_OSPITE:** idRecensioneOspite, RECENSIONE(idRecensione), OSPITE(idOspite)
- ★ **VALUTAZIONE:** idValutazione

**~Motivazioni:** Ho deciso di cambiare gli identificatori di ciascuna entità in modo da rendere più semplice ed accessibile il database. In questo modo le relazioni sono chiare e semplici da gestire.





## 2.4) Schema E/R ristrutturato + Business Rules

### **BUSINESS RULES:**

- L'utente deve aggiungere un metodo di pagamento per poter prenotare.
- L'utente deve indicare un intervallo di date ed il numero di ospiti per poter prenotare.
- Un utente per essere un Superhost deve: aver completato almeno 10 soggiorni(100 notti in totale), aver conservato un tasso di cancellazione dell'1% al massimo (1 cancellazione ogni 100 soggiorni) ed infine la valutazione complessiva di almeno 4,8 per tutte le proprietà.
- La prenotazione deve essere confermata o cancellata dall'host.
- La prenotazione può essere cancellata anche dall'ospite.
- Se la prenotazione viene accettata viene eseguito il pagamento
- Una volta che la prenotazione viene confermata o cancellata alla fine del soggiorno viene archiviata.
- L'indirizzo diventa visibile solo se la prenotazione viene confermata, altrimenti è visibile solo il comune.
- Se gli ospiti nella prenotazione sono utenti del servizio se ne possono indicare i nominativi.
- Solo al termine del soggiorno l'utente e l'host possono fare la recensione.
- La recensione diventa visibile solo se ospite ed host hanno fatto la recensione entrambi.
- La valutazione complessiva è una media delle valutazioni ricevute sulle singole dimensioni.
- Nella prenotazione la data d'inizio non può essere antecedente o coincidere con la data di fine.
- Negli alloggi l'ora di check-out deve precedere l'orario di check-in.

### **NUOVE BUSINESS RULES AGGIUNTE:**

- Ogni occorrenza di UTENTE deve partecipare a HOST oppure OSPITE;
- Ogni occorrenza di UTENTE può partecipare contemporaneamente a HOST e OSPITE.
- Il tipo di ALLOGGIO può essere solo tra appartamento, stanza privata e stanza condivisa.

- Lo stato della PRENOTAZIONE può essere solo inCorso, Confermata, Cancellata e Archiviata.
- Ogni occorrenza di RECENSIONE deve partecipare a RECENSIONE\_OSPITE e RECENSIONE\_HOST.
- Ogni occorrenza di RECENSIONE può essere di RECENSIONE\_HOST e/o RECENSIONE\_OSPITE.

## REGOLE DI DERIVAZIONE:

- Il tasso di cancellazione si calcola facendo il quoziente tra il numero di prenotazioni archiviate di un host e le sue prenotazioni cancellate
- La verifica avviene se l'utente carica una foto della carta d'identità.  
(n. prenotazioni archiviate / n. prenotazioni cancellate)
- La media del rating si calcola facendo la media con le valutazioni delle recensioni degli ospiti per alloggio.
- Il numero di recensioni si calcola dalle prenotazioni archiviate.
- Il costo totale della prenotazione si calcola sommando i costi di pulizia e il prezzo del soggiorno a notte per ciascun ospite.
- La valutazione finale si calcola facendo la media tra tutte le valutazioni dell'ospite sulle dimensioni date di un determinato alloggio.

## 2.5)Schema Relazionale

- UTENTE(email, password, NomeUtente, CognomeUtente, FotoDocumento, MetodoPagamento, Verificato)
- LISTA(NomeLista, UTENTE)
  - LISTA(UTENTE) referencia UTENTE(email)
- TELEFONO(Cellulare, UTENTE)
  - TELEFONO(UTENTE) referencia UTENTE(email)
- OSPITE(idOspite, UTENTE)
  - OSPITE(UTENTE) referencia UTENTE(email)
  - OSPITE(UTENTE) unique
- HOST(idHost, SuperHost, TassoCancellazione, Iban, UTENTE)
  - HOST(UTENTE) referencia UTENTE(email)
  - HOST(UTENTE) unique

- ALLOGGIO(idAlloggio, HOST, NomeAlloggio, Indirizzo, Civico, Comune, CostiPulizie, Prezzo, NumeroLetti, NumeroCamereDaLetto, Check-in, Check-out, DescrizioneAlloggio, TipoAlloggio, MediaRating)
  - ALLOGGIO(HOST) referencia HOST(idHost)
- CONSERVA(LISTA, ALLOGGIO)
  - CONSERVA(LISTA) referencia LISTA(NomeLista)
  - CONSERVA(ALLOGGIO) referencia ALLOGGIO(idAlloggio)
- FOTO(CodFoto, Immagine, ALLOGGIO)
  - FOTO(ALLOGGIO) referencia ALLOGGIO(idAlloggio)
- DOTATO(SERVIZIO, ALLOGGIO)
  - DOTATO(SERVIZIO) referencia SERVIZIO(nomeServizio)
  - DOTATO(ALLOGGIO) referencia ALLOGGIO(idAlloggio)
- SERVIZIO(NomeServizio, DescrizioneServizio, ALLOGGIO, )
  - SERVIZIO(ALLOGGIO) referencia ALLOGGIO(idAlloggio)
- PRENOTAZIONE(idPrenotazione, OSPITE, ALLOGGIO, DataInizio, DataFine, CostoTotale, NumeroOspiti, Stato)
  - PRENOTAZIONE(OSPITE) referencia OSPITE(idOspite)
  - PRENOTAZIONE(ALLOGGIO) referencia ALLOGGIO(idAlloggio)
- RECENSIONE(idRecensione, PRENOTAZIONE, Visibile)
  - RECENSIONE(PRENOTAZIONE) referencia PRENOTAZIONE(idPrenotazione)
  - RECENSIONE(PRENOTAZIONE) unique
- COMMENTO(idCommento, TestoCommento, RECENSIONE)
  - COMMENTO(RECENSIONE) referencia RECENSIONE(idRecensione)
- RECENSIONE\_HOST(id RecensioneHost, HOST, TestoOspite, RECENSIONE)
  - RECENSIONE\_HOST(RECENSIONE) referencia RECENSIONE(idRecensione)
  - RECENSIONE\_HOST(RECENSIONE) unique
  - RECENSIONE\_HOST(HOST) referencia HOST(idHost)
  - RECENSIONE\_HOST(HOST) unique
- RECENSIONE\_OSPITE(idRecensioneOspite, OSPITE, TestoAlloggio, TestoHost, ValutazioneFinale, RECENSIONE)
  - RECENSIONE\_OSPITE(RECENSIONE) referencia RECENSIONE(idRecensione)

- RECENSIONE\_OSPITE(RECENSIONE) unique
- RECENSIONE\_OSPITE(OSPITE) referenzia OSPITE(idOspite)
- RECENSIONE\_OSPITE(OSPITE) unique
- VALUTAZIONE(idValutazione, voto, descrizioneVoto, RECENSIONE\_OSPITE)
  - VALUTAZIONE(RECENSIONE\_OSPITE) referenzia RECENSIONE\_OSPITE(IdRecensioneOspite)
- GIUDICA(VALUTAZIONE, RECENSIONE\_OSPITE)
  - GIUDICA(VALUTAZIONE) referenzia VALUTAZIONE(idValutazione)
  - GIUDICA(RECENSIONE\_OSPITE) referenzia RECENSIONE\_OSPITE(idRecensioneOspite)

### ***Vincoli d'integrità referenziale:***

- Un utente non può effettuare una prenotazione su un alloggio di cui è proprietario.
- Quando un alloggio viene prenotato non dev'essere eliminato dal database.
- Un alloggio non dev'essere incluso nei risultati della ricerca se è stato già prenotato nelle medesime date selezionate o se già stato prenotato nell'arco di esse.
- Un utente può non avere alcuna lista.
- Solo gli attributi Verificato, MetodoPagamento e FotoDocumento possono essere nulli.
- Le Recensioni\_ospite e le Recensioni\_host possono essere nulle.
- Recensione può contenere la Recensione\_Ospite e la Recensione\_Host o una o nessuna delle due.

## Implementazione

### 3.1)DDL di creazione del database

```
create table UTENTE(
    Email          varchar(30) primary key,
    PasswordUtente varchar(20) not null,
```

```
NomeUtente    varchar(20) not null,  
CognomeUtente varchar(20) not null,  
FotoDocumento boolean,  
MetodoPagamento boolean,  
Verificato    boolean  
);
```

```
create table LISTA(  
    NomeLista    varchar(30) primary key,  
    EmailUtente  varchar(30) references UTENTE(email) ON DELETE  
                                     CASCADE ON UPDATE CASCADE  
);
```

```
create table TELEFONO(  
    Cellulare    numeric(10) primary key CHECK(Cellulare>9999999999),  
    Proprietario varchar(20) references UTENTE(email) ON DELETE  
                                     CASCADE ON UPDATE CASCADE  
);
```

```
create table OSPITE(  
    idOspite     varchar(10) primary key,  
    emailUtente  varchar(30) unique references UTENTE(email) ON DELETE  
                                     NO ACTION ON UPDATE CASCADE  
);
```

```
create table HOST(  
    idHost       varchar(10) primary key,  
    TassoCancellazione float(3),  
    Iban         varchar(27) not null,  
    emailUtente  varchar(30) unique references UTENTE(email) ON DELETE  
                                     NO ACTION ON UPDATE CASCADE  
);
```

```
create domain TipoAll as varchar(20) CHECK (VALUE ='Appartamento' OR VALUE  
='Stanza Condivisa' OR VALUE ='Stanza Privata' );
```

```

create table ALLOGGIO(
    idAlloggio    varchar(10) primary key,
    Host          varchar(10) references HOST(idHost) ON DELETE NO ACTION
                                     ON UPDATE CASCADE,

    NomeAlloggio  varchar(30) not null,
    Indirizzo     varchar(40) not null,
    Civico        numeric(3),
    Comune        varchar(30) not null,
    CostiPulizia  numeric(4) not null check(CostiPulizia > 0),
    Prezzo        numeric(7) not null check(Prezzo > 0),
    NumeroLetti   numeric(2) not null check(NumeroLetti > 0),
    NumeroCamereDaLetto numeric(2) check(NumeroCamereDaLetto > 0),
    CheckIn       time not null check (Checkin > CheckOut),
    CheckOut      time not null check (CheckOut < CheckIn),
    DescrizioneAlloggio varchar(250),
    TipoAlloggio  TipoAll not null,
    MediaRating   float(5)
);

```

```

create table CONSERVA(
    Lista         varchar(20) references LISTA(NomeLista) ON DELETE
                                     CASCADE ON UPDATE CASCADE,
    Alloggio      varchar(10) references ALLOGGIO(idAlloggio) ON DELETE
                                     CASCADE ON UPDATE CASCADE
);

```

```

create table FOTO(
    CodFoto       varchar(10) primary key,
    Immagine      boolean,
    Alloggio      varchar(10) references ALLOGGIO(idAlloggio) ON DELETE
                                     CASCADE ON UPDATE CASCADE
);

```

```

create table SERVIZIO(
    nomeServizio  varchar(25) primary key,
    DescrizioneServizio varchar(250)
);

```

```
create domain statoprenotazione AS VARCHAR(12) CHECK(VALUE ='Archiviata'  
OR VALUE ='Confermata' OR VALUE ='Cancellata' OR VALUE='In Corso');
```

```
create table RECENSIONE(
    idRecensione  varchar(10) primary key,
    Prenotazione  varchar(10) unique references
    PRENOTAZIONE(idPrenotazione) ON DELETE CASCADE ON UPDATE
    CASCADE,
    Visibile      boolean
);
```

[illegible]

```

create table RECENSIONE_HOST(
    idRecensioneHost varchar(10) primary key,
    Host varchar(10) unique references HOST(idHost) ON DELETE NO
                                ACTION ON UPDATE CASCADE,
    TestoOspite varchar(150),
    Recensione varchar(10) unique references
RECENSIONE(idRecensione) ON DELETE NO ACTION ON UPDATE CASCADE
);

```

```

create table RECENSIONE_OSPITE(
    idRecensioneOspite varchar(10) primary key,
    Ospite varchar(10) unique references OSPITE(idOspite) ON
                                DELETE NO ACTION ON UPDATE CASCADE,
    TestoAlloggio varchar(150),
    TestoHost varchar(150),
    ValutazioneFinale numeric(5),
    Recensione varchar(10) unique references
RECENSIONE(idRecensione) ON DELETE NO ACTION ON UPDATE CASCADE
);

```

```

create table VALUTAZIONE(
    NomeValutazione varchar(25) primary key,
    Voto numeric(2) not null check(Voto > 0 AND Voto <= 5),
    DescrizioneVoto varchar(150)
);

```

```

create table GIUDICA(
    ValutazioneServizio varchar(25) references
VALUTAZIONE(NomeValutazione),
    RecensioneOspite varchar(10) references
RECENSIONE_OSPITE(idRecensioneOspite)
);

```

### 3.2)DML di popolamento di tutte le tabelle del database

```

insert into UTENTE(email, passwordutente, nomeutente, cognomeutente,
fotodocumento, metodopagamento, verificato) VALUES('abourida@unito.it','arancia','
Zakaria', 'Abourida', true, true, true);

```



```
insert into UTENTE(email, passwordutente, nomeutente, cognomeutente,
fotodocumento, metodopagamento, verificato) VALUES('Lavric@unito.it',
'mandarino', 'Cosmin', 'Lavric', false, true, false);
insert into UTENTE(email, passwordutente, nomeutente, cognomeutente,
fotodocumento, metodopagamento, verificato) VALUES('eusebio@unito.it', 'pesca',
'Stefano', 'Eusebio', true, true, true);
insert into UTENTE(email, passwordutente, nomeutente, cognomeutente,
fotodocumento, metodopagamento, verificato) VALUES('calcutta@unito.it', 'ananas',
'Marco', 'Calcutta', false, true, false);
insert into UTENTE(email, passwordutente, nomeutente, cognomeutente,
fotodocumento, metodopagamento, verificato) VALUES('diparma@unito.it', 'pera',
'Francesco', 'Diparma', true, true, true);
```

```
insert into TELEFONO(cellulare, proprietario) VALUES('3265165410',
'abourida@unito.it');
insert into TELEFONO(cellulare, proprietario) VALUES('5165121561',
'Lavric@unito.it');
insert into TELEFONO(cellulare, proprietario) VALUES('5198131065',
'eusebio@unito.it');
insert into TELEFONO(cellulare, proprietario) VALUES('1654116516',
'calcutta@unito.it');
insert into TELEFONO(cellulare, proprietario) VALUES('1616411351',
'diparma@unito.it');
```

```
insert into HOST(idhost, tassocancellazione, iban, emailUtente)
VALUES('host000001',0.5,'sf086dasfg95','abourida@unito.it');
insert into HOST(idhost, tassocancellazione, iban, emailUtente)
VALUES('host000002',1,'fo3b304g0333', 'Lavric@unito.it');
```

```
insert into OSPITE(idospite, emailUtente) VALUES('ospite0001', 'eusebio@unito.it');
insert into OSPITE(idospite, emailUtente) VALUES('ospite0002', 'calcutta@unito.it');
insert into OSPITE(idospite, emailUtente) VALUES('ospite0003', 'diparma@unito.it');
```

```
insert into LISTA(nomelista, emailutente) VALUES('estate2k22', 'abourida@unito.it');
insert into LISTA(nomelista, emailutente) VALUES('primavera2k23',
'diparma@unito.it');
```

```
insert into ALLOGGIO(idalloggio, host, nomealloggio, indirizzo, civico, comune,
costipulizia, prezzo, numeroletti, numerocameredaletto, checkin, checkout,
descrizionealloggio, tipoalloggio, mediarating)
VALUES('alloggio01', 'host000001', 'casaStregata', 'via garibaldi', 31, 'Torino', 30, 50,
3, 2, '15:00', '12:00', 'bella casa', 'Appartamento', 4.8);
```

```
insert into ALLOGGIO(idalloggio, host, nomealloggio, indirizzo, civico, comune,
costipulizia, prezzo, numeroletti, numerocamera, daletto, checkin, checkout,
descrizionealloggio, tipoalloggio, mediarating)
VALUES('alloggio02', 'host000001', 'casa in campagna', 'via cavour', 12, 'Milano', 45,
70,1, 1, '18:00', '9:00', 'accogliente e carina', 'Stanza Privata', 3.0);
insert into ALLOGGIO(idalloggio, host, nomealloggio, indirizzo, civico, comune,
costipulizia, prezzo, numeroletti, numerocamera, daletto, checkin, checkout,
descrizionealloggio, tipoalloggio, mediarating)
VALUES('alloggio03', 'host000002', 'Giuseppina', 'via ernesto', 5, 'Trieste', 80, 200, 6,
4, '16:00', '10:00', 'lusso sfrenato per tanti amici', 'Appartamento', 5.0);
```

```
insert into CONSERVA(lista, alloggio) VALUES('estate2k22', 'alloggio01');
insert into CONSERVA(lista, alloggio) VALUES('estate2k22', 'alloggio03');
insert into CONSERVA(lista, alloggio) VALUES('primavera2k23', 'alloggio02');
```

```
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000001', true,
'alloggio01');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000002', true,
'alloggio01');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000003', true,
'alloggio01');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000004', true,
'alloggio02');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000005', true,
'alloggio02');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000006', true,
'alloggio03');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000007', true,
'alloggio03');
insert into FOTO(codfoto, immagine, alloggio) VALUES('foto000008', true,
'alloggio03');
```

```
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Bagno', 'bagno
per bisogni');
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Cucina', 'grande
cucina');
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Soggiorno',
'soggiorno con 2 divani');
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Tv', 'Televisione');
insert into SERVIZIO(nomeservizio, descrizoneservizio)
VALUES('Lavanderia', 'stanza con elettrodomestici per lavaggio panni...');
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Terrazza', 'ampia
terrazza');
```

```
insert into SERVIZIO(nomeservizio, descrizoneservizio) VALUES('Giardino', '30mq di giardino');
```

```
insert into DOTATO(servizio, alloggio) VALUES('Bagno', 'alloggio01');
insert into DOTATO(servizio, alloggio) VALUES('Bagno', 'alloggio02');
insert into DOTATO(servizio, alloggio) VALUES('Bagno','alloggio03');
insert into DOTATO(servizio, alloggio) VALUES('Cucina', 'alloggio01');
insert into DOTATO(servizio, alloggio) VALUES('Cucina', 'alloggio02');
insert into DOTATO(servizio, alloggio) VALUES('Cucina','alloggio03');
insert into DOTATO(servizio, alloggio) VALUES('Soggiorno', 'alloggio01');
insert into DOTATO(servizio, alloggio) VALUES('Soggiorno', 'alloggio02');
insert into DOTATO(servizio, alloggio) VALUES('Soggiorno','alloggio03');
insert into DOTATO(servizio, alloggio) VALUES('Tv', 'alloggio01');
insert into DOTATO(servizio, alloggio) VALUES('Tv', 'alloggio02');
insert into DOTATO(servizio, alloggio) VALUES('Tv','alloggio03');
insert into DOTATO(servizio, alloggio) VALUES('Terrazza', 'alloggio02');
insert into DOTATO(servizio, alloggio) VALUES('Terrazza','alloggio03');
insert into DOTATO(servizio, alloggio) VALUES('Giardino', 'alloggio03');
```

```
insert into PRENOTAZIONE(idprenotazione, ospite, alloggio, datainizio, datafine,
costototale, numeroospiti, stato)
VALUES('prenot0001', 'ospite0001', 'alloggio01', '2022-07-22', '2022-07-30', 730, 2,
'In Corso');
insert into PRENOTAZIONE(idprenotazione, ospite, alloggio, datainizio, datafine,
costototale, numeroospiti, stato)
VALUES('prenot0002', 'ospite0002', 'alloggio02', '2022-07-22', '2022-08-30',535, 1,
'Archiviata');
insert into PRENOTAZIONE(idprenotazione, ospite, alloggio, datainizio, datafine,
costototale, numeroospiti,stato)
VALUES('prenot0003', 'ospite0003', 'alloggio03', '2022-07-22', '2022-07-30', 4280, 3,
'Confermata');
```

```
insert into RECENSIONE(idrecensione, prenotazione, visibile)
VALUES('rec0000001', 'prenot0001', true);
```

```
insert into RECENSIONE_HOST(idrecensionehost, host, testoospite, recensione)
VALUES('rechost001', 'host000001', 'buon ospite', 'rec0000001');
```

```
insert into RECENSIONE_OSPITE(idrecensioneospite, ospite, testoalloggio,
testohost, valutazionefinale, recensione)
VALUES('recosp0001', 'host000002', 'posto carino', 'host molto gentile', 4,
'rec0000001');
```

```
insert into GIUDICA(valutazioneservizio, recensioneospite)
VALUES('Bagno', 'recosp0001');
insert into GIUDICA(valutazioneservizio, recensioneospite)
VALUES('Cucina', 'recosp0001');
insert into GIUDICA(valutazioneservizio, recensioneospite)
VALUES('Soggiorno', 'recosp0001');
insert into GIUDICA(valutazioneservizio, recensioneospite)
VALUES('Tv', 'recosp0001');
```

## 1. MODIFICA/AGGIORNAMENTO DATI DELLE LISTE

////////////////////////////////////

////////////////////////////////////

### 3. MODIFICA DATI DI ALCUNI OSPITI

```
UPDATE OSPITE SET idOspite = 'ospite0042' WHERE idOspite = 'ospite0002';
UPDATE OSPITE SET idOspite = 'ospite0032' WHERE idOspite = 'ospite0042';
UPDATE OSPITE SET idOspite = 'ospite0222' WHERE idOspite = 'ospite0032';
```

////////////////////////////////////

#### 4. MODIFICA DATI DEGLI UTENTI(VARIE CONDIZIONI)

```
insert into UTENTE(email, passwordutente, nomeutente, cognomeutente, fotodocumento,
metodopagamento, verificato)
VALUES ('digiorio@gmail.com', 'banana', 'mario', 'digiorio', false, false, false);
```

```
UPDATE UTENTE SET fotoDocumento = true WHERE email = 'digiorio@gmail.com';
```

```
UPDATE UTENTE SET verificato = true WHERE fotodocumento = true;
```

////////////////////////////////////

## 5. AGGIUNTA DI NUOVI ALLOGGI ED ELIMINAZIONE DI UN ALLOGGIO PER VERIFICARE LO STATO DELLE PRENOTAZIONI

```
insert into ALLOGGIO(idalloggio, host, nomealloggio, indirizzo, civico, comune, costipulizia,
prezzo, numeroletti, numerocameredaletto, checkin, checkout, descrizionealloggio,
tipoalloggio, mediarating)
VALUES('alloggio04', 'host000002', 'casa in montagna', 'via marta', 1, 'Roma', 30, 50, 3, 2,
'15:00', '12:00', 'bella casa', 'Appartamento', 2.5);
```

```
insert into ALLOGGIO(idalloggio, host, nomealloggio, indirizzo, civico, comune, costipulizia,
prezzo, numeroletti, numerocameradaletto, checkin, checkout, descrizionealloggio,
tipoalloggio, mediarating)
VALUES('alloggio05', 'host000002', 'casa in collina', 'via giuseppe', 80, 'Ivrea', 30, 50, 1, 1,
'15:00', '12:00', 'bella casa', 'Stanza Condivisa', 1.0);
```

```
DELETE FROM ALLOGGIO WHERE idAlloggio = 'alloggio03';
```

