

LABORATORIO BASI DI DATI: Progetto di piattaforma di home booking

Progettazione concettuale

1.1 Requisiti iniziali

Si vuole realizzare una base di dati per un servizio che permette di affittare e prenotare alloggi di vario tipo ad esempio interi appartamenti, stanze private (camera privata e spazi comuni) e stanze condivise (spazio in comune e camera condivisa). * Gli utenti si registrano al servizio fornendo indirizzo email, password, nome, cognome, numero o numeri di telefono. Se l'utente fornisce la foto della carta d'identità, viene riconosciuto come verificato. Inoltre, l'utente deve indicare un metodo di pagamento per poter prenotare. Gli utenti possono essere ospiti o "host" ovvero possono a loro volta ospitare altri utenti del servizio in uno o più alloggi di loro proprietà. Inoltre gli "host" possono diventare "superhost" se soddisfano i seguenti requisiti:

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

Gli utenti superhost ricevono un badge sul loro profilo. Gli alloggi sono descritti indicando un nome, l'indirizzo (visibile all'ospite solo quando la prenotazione è confermata, altrimenti è visibile solo il comune), una descrizione, il prezzo per notte per persona e i costi di pulizia, delle foto, i servizi (ad esempio, cucina, wi-fi, lavatrice, ecc.), numero di letti e orario di check-in e check-out oltre all'host a cui appartiene, il rating medio e il numero di recensioni (vedere Fig. 1). Gli utenti possono aggiungere alcune case tra i preferiti. Gli utenti possono avere diverse liste, ad esempio in base al viaggio che vogliono compiere. Gli utenti possono prenotare degli alloggi di qualsiasi tipo indicando un intervallo di date per il soggiorno e il numero degli ospiti. Se gli ospiti sono a loro volta utenti del servizio, se ne possono indicare i nominativi. La prenotazione deve essere confermata o rifiutata dall'host. La prenotazione ha un costo totale e se confermata viene eseguito il pagamento. Inoltre, la prenotazione può essere cancellata sia dall'ospite che dall'host.

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

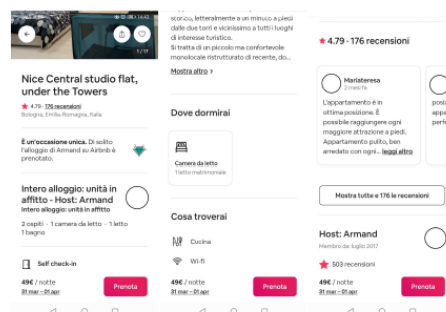


Fig. 1: Dettagli di un appartamento.

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

Termine	Descrizione	Sinonimi	Collegamenti
Alloggio	Struttura scelta in fase di prenotazione	Casa, appartamento	Host, prenotazione
Intero appartamento	Tipologia di alloggio		Alloggio
Stanza condivisa	Tipologia di alloggio		Alloggio
Stanza privata	Tipologia di alloggio		Alloggio
Utente	Persona che utilizza il servizio		Cancellata, liste preferiti, ospite, host, prenotazione
Ospite	Affittuario di un alloggio		Recensione ospite, utenti
Host	Proprietario di un alloggio		Recensione host, utenti, alloggio
Superhost	Host che soddisfa particolari requisiti		Host
Verificato	Utente che ha fornito la carta d'identità		Utenti
Liste preferiti	Liste degli alloggi preferiti dall'utente		Utenti
Recensione	Valutazione e voto del soggiorno	Review, valutazione complessiva, valutazioni	Commento, visibile, non visibile, recensione ospite, recensione host
Recensione ospite	Recensione scritta un ospite		Ospite, recensione
Recensione host	Recensione scritta da un host		Host, recensione
Visibile	Recensione visibile a ospiti e host		Recensione
Non visibile	Recensione non visibile a ospiti e host		Recensione
Commento	Risposta ad una recensione o ad un altro commento		Commento, recensione
Prenotazione	Richiesta di un alloggio da parte di un utente		Confermata, rifiutata, utenti, alloggio
Confermata	Prenotazione accettata dall'host		Cancellata, terminata
Rifiutata	Prenotazione non accettata dall'host		Prenotazione
Cancellata	Prenotazione cancellata dall'ospite o dall'host		Confermata, utenti
Terminata	Prenotazione conclusa		Confermata

1.3 Requisiti rivisti e strutturati in gruppi di frasi omogenee

Alloggi

Gli alloggi sono oggetto di prenotazione e si dividono in interi appartamenti, stanze private (camera privata e spazi comuni) e stanze condivise (spazio in comune e camera condivisa).

Per ogni alloggio si memorizza: 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.

Ogni alloggio deve essere di proprietà di un host.

Utenti

Per ogni utente si memorizza: 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 avere diverse liste preferiti, ad esempio in base al viaggio che vogliono compiere.

Ospiti e Host

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.

Superhost

Gli "host" possono diventare "superhost", ricevendo un'etichetta (badge) sul loro profilo, se soddisfano i seguenti requisiti:

- Devono aver terminato almeno 10 prenotazioni, 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 tutte le prenotazioni terminate in tutte gli alloggi di sua proprietà.

Prenotazione

Una prenotazione può essere effettuata dagli utenti per alloggi di qualsiasi tipo, indicando un intervallo di date per il soggiorno e il numero degli ospiti. Se questi ultimi sono utenti del servizio, ne vengono indicati anche i nominativi.

La prenotazione deve essere confermata o rifiutata dall'host e se è confermata può essere cancellata (sia dall'ospite sia dall'host) o terminata.

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

Recensione

Al termine della prenotazione, gli ospiti e gli host possono recensirsi a vicenda.

La recensione fatta dagli ospiti comprende due testi (uno per l'alloggio e uno per l'host) e una serie di punteggi in una scala da 1 a 5 sui servizi dell'alloggio come pulizia, comunicazione, posizione, qualità/prezzo.

La valutazione complessiva del soggiorno è una media dei punteggi ottenuti sui singoli servizi.

Le recensioni degli host comprendono solo un commento.

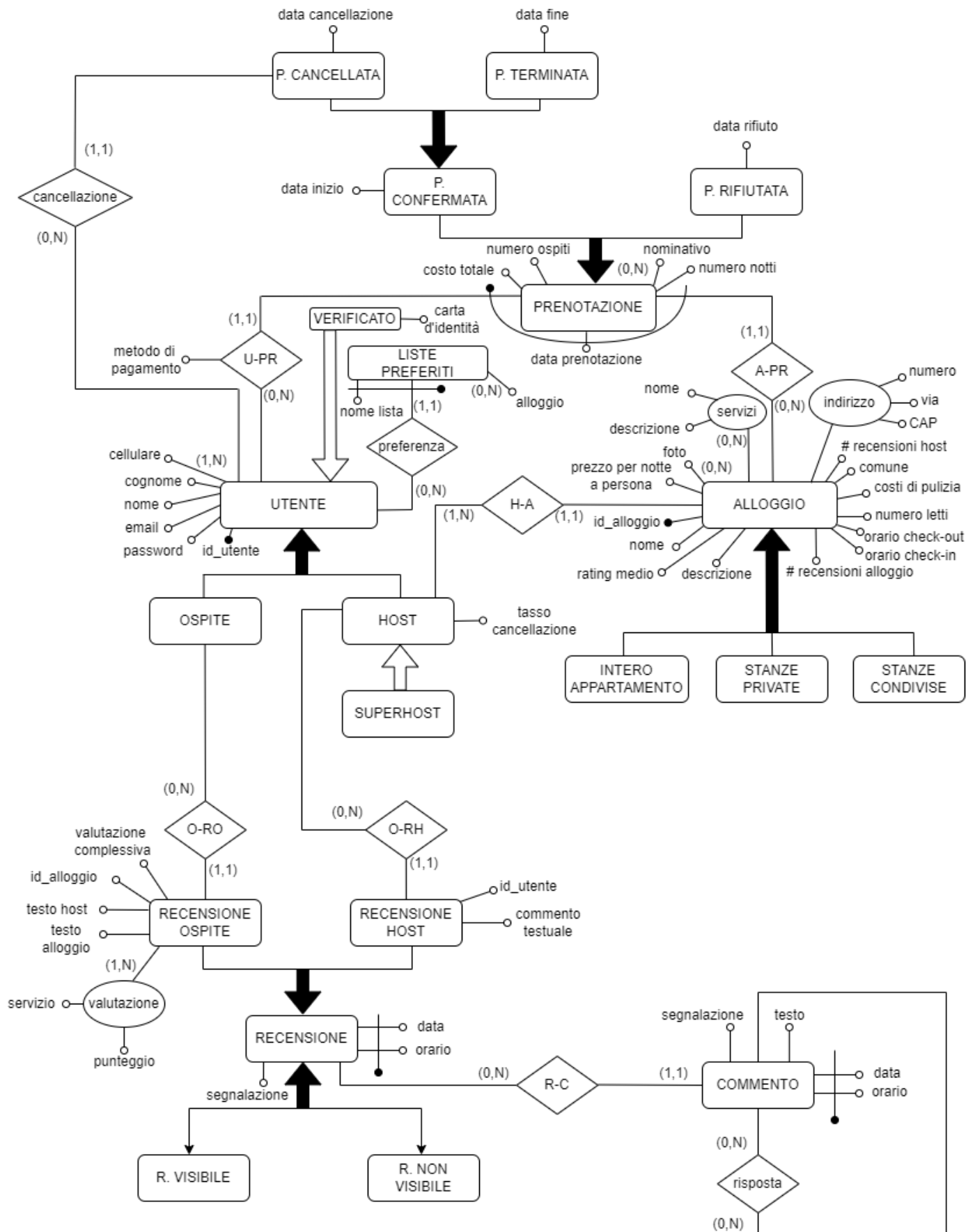
Le recensioni possono essere visibili o non visibili: diventano visibili quando ospite e host hanno fatto la recensione, oppure se uno dei due non ha fatto la recensione l'altra diventa visibile dopo 7 giorni dalla fine della prenotazione.

Le recensioni sono visibili sui profili degli utenti suddivise in base a quelle ricevute come ospite e come host.

Commento

Gli host e gli ospiti possono commentare più volte le recensioni in cui sono coinvolti, creando così una catena di commenti (thread).

1.4.1 Schema E-R



1.4.2 Business rules

1. Gli "host" possono diventare "superhost", ricevendo un'etichetta (badge) sul loro profilo, se soddisfano i seguenti requisiti:
 - Devono aver terminato almeno 10 prenotazioni, 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 tutte le prenotazioni terminate in tutte gli alloggi di sua proprietà.
2. Le recensioni diventano visibili quando:
 - ospite e host hanno fatto la recensione;
 - se uno dei due non ha fatto la recensione, l'altra diventa visibile dopo 7 giorni dalla fine della prenotazione.
3. Un utente non può effettuare prenotazioni per alloggi di cui è host.
4. Un alloggio non deve essere eliminato definitivamente dal database se è stata effettuata almeno una prenotazione su di esso.
5. Non ci possono essere più prenotazioni per uno stesso alloggio che si sovrappongano.
6. Il numero degli ospiti indicato in fase di prenotazione non può superare il numero di posti letto dell'alloggio.
7. Data prenotazione < data inizio, data fine, data cancellazione, data rifiuto.
8. Data inizio + numero notti = data fine && data inizio > data cancellazione, data rifiuto.
9. I nominativi possono essere aggiunti alla prenotazione solo se riguardano utenti già iscritti al servizio.
10. L'attributo 'segnalazione' di commento e recensione vale true quando un utente lo ha segnalato al servizio in quanto ritenuto offensivo, false altrimenti.

1.4.3 Regole di derivazione

1. La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sui singoli servizi.
2. Il rating medio di un alloggio è la media delle valutazioni complessive date dagli utenti.
3. Il numero di recensioni di un alloggio è la somma del numero di recensioni scritte dagli ospiti per l'alloggio con campo 'testo alloggio' diverso da NULL.
4. Il numero di recensioni dell'host è la somma del numero di recensioni scritte dagli utenti per l'host con campo 'testo host' diverso da NULL.
5. Il tasso di cancellazione di un host si ottiene come (numero di prenotazioni cancellate/numero totale di prenotazioni confermate) %, in cui le prenotazioni considerate riguardano alloggi di proprietà dell'host.
6. Il costo totale di una prenotazione si calcola come: 'numero notti' x 'prezzo per notte a persona' x 'numero ospiti'.

Progettazione logica

2.1.1 Tavola dei volumi: entità

Concetto	Tipo	Volume	Descrizione
Utente	E	150.000.000	Circa 150milioni di utenti all'anno
Ospite	E	146.000.000	Differenza tra numero di utenti e numero di host
Host	E	4.000.000	Circa 4milioni di host all'anno
Superhost	E	400.000	Circa il 10% degli host sono superhost
Verificato	E	138.700.000	Circa il 95% degli ospiti sono verificati
Alloggio	E	5.600.000	Più di 5milioni di alloggi disponibili all'anno
Intero appartamento	E	3.360.000	Il 60% degli alloggi sono interi appartamenti
Stanze condivise	E	560.000	Il 10% degli alloggi sono stanze condivise
Stanze private	E	1.680.000	Il 30% degli alloggi sono stanze private
Recensione	E	116.800.000	L'80% degli utenti effettua una recensione
Recensione ospite	E	70.080.000	Il 60% delle recensioni sono scritte dagli ospiti
Recensione host	E	46.720.000	Il 40% delle recensioni sono scritte dagli host
Recensione visibile	E	99.280.000	L'85% delle recensioni è visibile
Recensione non visibile	E	17.520.000	Il 15% delle recensioni non è visibile
Commento	E	233.600.000	In media consideriamo 2 commenti a recensione
Prenotazione	E	450.000.000	In media 3 prenotazioni all'anno per utente
Prenotazione confermata	E	427.500.000	Il 95% delle prenotazioni sono confermate
Prenotazione rifiutata	E	22.500.000	Il 5% delle prenotazioni sono rifiutate
Prenotazione cancellata	E	21.375.000	Il 5% delle prenotazioni confermate sono cancellate
Prenotazione terminata	E	406.125.000	Il 95% delle prenotazioni confermate sono terminate
Liste preferiti	E	450.000.000	In media 3 liste preferiti a utente

2.1.2 Tavola dei volumi: relazioni

Concetto	Tipo	Volume
U-PR	R	450.000.000
A-PR	R	450.000.000
H-A	R	5.600.000
O-RO	R	146.000.000
O-RH	R	4.000.000
R-C	R	116.800.000
Cancellazione	R	21.375.000
Preferenza	R	450.000.000
Risposta	R	233.600.000

I volumi inseriti in tabella sono in parte ipotizzati e in parte ricavati dal sito:

<https://www.thezebra.com/resources/home/airbnb-statistics/#infographic>

2.2 Tavola delle operazioni

Operazione	Tipo	Frequenza
Aggiornamento del tasso di cancellazione di ciascun host	Batch	Una volta a settimana
Controllo e aggiornamento della condizione di superhost	Batch	Una volta al giorno
Calcolo della classifica degli alloggi più graditi	Batch	Una volta al mese
Calcolo del rating medio di un alloggio	Batch	Una volta a settimana
Controllo sull'inserimento della C.I. di un utente	Batch	Una volta al giorno
Controllo sulla segnalazione di un commento o una recensione	Batch	Una volta al giorno
Prenotazione di un alloggio da parte di un utente	Interattivo	Un milione al giorno
Iscrizione di un utente al servizio	Interattivo	400 mila al giorno
Aggiunta di un alloggio al servizio	Interattivo	15 mila al giorno
Inserimento di una recensione	Interattivo	330 mila al giorno
Aggiunta di una nuova lista preferiti	Interattivo	Un milione al giorno
Cancellazione di una prenotazione	Interattivo	50 mila al giorno
Disiscrizione di un utente dal servizio	Interattivo	1000 al giorno
Rimozione di un alloggio dal servizio	Interattivo	500 al giorno
Inserimento di un commento	Interattivo	350 mila al giorno
Rimozione di un commento o di una recensione segnalata	Interattivo	10 mila al giorno

2.3 Ristrutturazione dello schema E-R

2.3.1 Analisi delle ridondanze

1. 'Costo totale' in prenotazione.
2. 'Numero di recensioni' in alloggio.
3. 'Rating medio' in alloggio.
4. 'Valutazione complessiva' in recensione ospite.
5. 'Tasso di cancellazione' in host.

Il numero di recensioni di un alloggio si può calcolare mettendo in join alloggio e recensione su id_alloggio, tuttavia risulterebbe un'operazione molto costosa in quanto per un numero di prenotazioni molto elevato le tuple mosse dal join sarebbero troppe.

Il tasso di cancellazione di un host si può calcolare trovando il numero di prenotazioni cancellate che riguardano un alloggio di proprietà dell'host, tuttavia come nel caso precedente l'operazione risulterebbe molto costosa, pertanto è utile mantenere l'attributo.

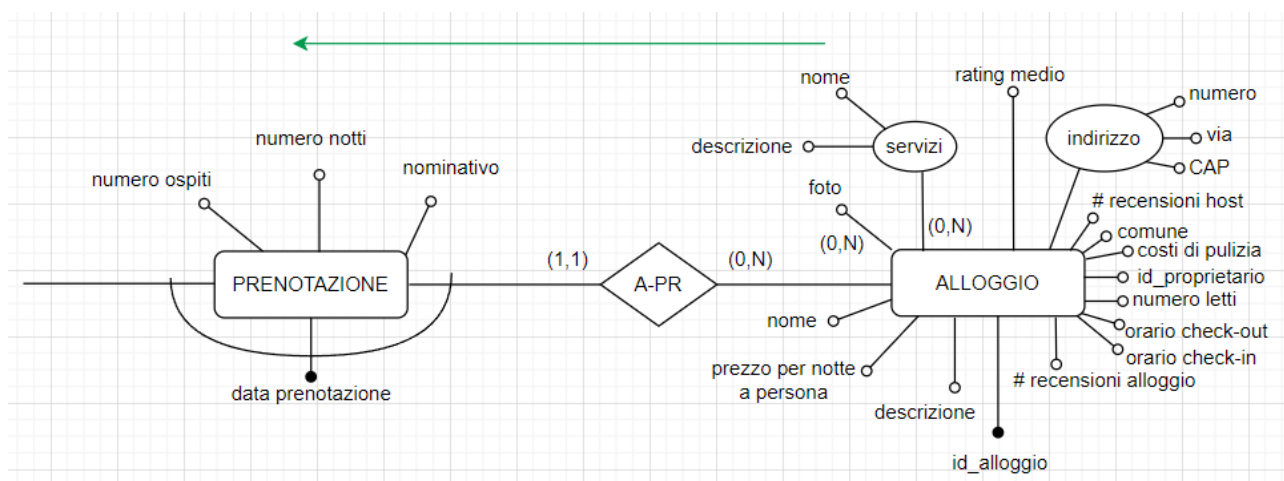
Il rating medio e la valutazione complessiva di un alloggio si possono eliminare in quanto non sono costose da calcolare:

- la valutazione complessiva di un alloggio si può calcolare facendo la media delle valutazioni di un alloggio sulle singole recensioni;
- il rating medio di un alloggio è derivabile trovando tutte le recensioni sull'alloggio e facendo la media delle valutazioni complessive.

Analisi specifica: costo totale

Il costo totale di una prenotazione è molto facile da calcolare: è sufficiente moltiplicare 'numero notti' per 'numero ospiti' (presenti in prenotazione) per 'prezzo per notte a persona' (presente in alloggio).

Schema di navigazione



Tavole e analisi degli accessi

Per calcolare il costo totale con ridondanza occorre:

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	L

(tabella 1)

Per calcolare il costo totale senza ridondanza occorre:

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	2	L
Alloggio	E	1	L
Prenotazione	E	1	S

(tabella 2)

Supponendo 100 accessi al giorno, stimiamo gli accessi in lettura e in scrittura giornalieri per un singolo alloggio:

- Con ridondanza:
 - Accessi in lettura: $100 * 1 = 100$;
 - Accessi in scrittura: 0;
 - Totale: 100 accessi al giorno.
- Senza ridondanza:
 - Accessi in lettura: $100 * 3 = 300$;
 - Accessi in scrittura: $(100 * 1) * 2 = 200$ (gli accessi in scrittura costano il doppio);
 - Totale: 500 accessi al giorno.

Confronto in spazio e tempo tra presenza e assenza di ridondanza

Per memorizzare il costo totale supponiamo di utilizzare 4 byte (utilizzando il tipo float).

Con ridondanza	Senza ridondanza
100 accessi	500 accessi
4 byte * (5.600.000 alloggi) = 22.400.000 byte	0 byte

(tabella 3)

Conclusioni

Le tabelle 1 e 2 mostrano come rimuovere la ridondanza provochi un incremento del numero di accessi: si passa infatti da 100 a 500 accessi al giorno.

I benefici del rimuovere l'attributo 'costo totale' si possono riscontrare nella tabella 3:

essa mostra infatti come rimuovere la ridondanza permetta di ottenere un miglioramento significativo nella gestione dello spazio, passando da più di 20milioni di byte richiesti a 0 byte necessari.

Pertanto, poiché la dimensione dello spazio richiesto è un aspetto importante nella gestione di una base dati, scegliamo di rimuovere l'attributo considerato dall'entità prenotazione.

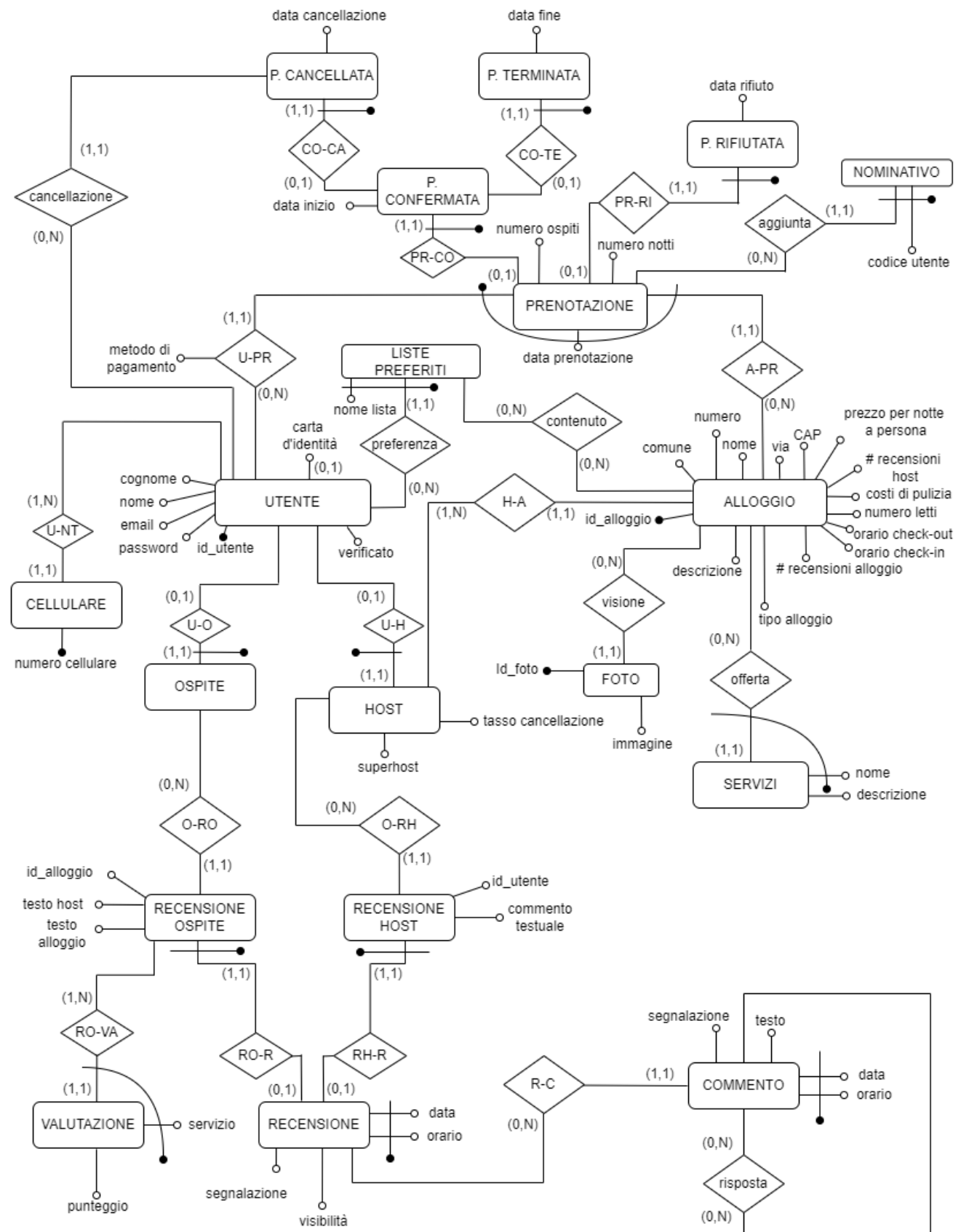
2.3.2 Eliminazione delle generalizzazioni

- **Alloggio:** poiché gli accessi al padre e alle figlie sono contestuali (il tipo di alloggio non influenza sulle modalità di prenotazione e sulle caratteristiche dell'alloggio), abbiamo eliminato la generalizzazione accorpando le entità figlie nel genitore;
- **Utente verificato, Superhost, Recensione visibile e non visibile:** poiché l'accesso all'entità figlia è contestuale al padre, abbiamo eliminato le generalizzazioni come al punto precedente;
- **Prenotazione:** poiché ci sono delle entità figlie con pochi attributi i cui accessi sono separati all'entità padre, abbiamo eliminato la generalizzazione introducendo delle relazioni tra il padre e i figli;
- **Utenti (ospite e host), Recensione ospite e recensione host:** come nel caso precedente.

2.3.4 Scelta degli identificatori principali

- **Prenotazione:** identificata dall'utente e dall'alloggio che la riguardano e dalla data di inizio prenotazione;
- **Prenotazione confermata e rifiutata:** identificate mediante vincoli di chiave esterna da prenotazione;
- **Prenotazione cancellata e terminata:** identificate mediante vincoli di chiave esterna da prenotazione confermata;
- **Nominativo:** identificato dal codice utente (tramite vincolo di integrità referenziale con 'id_utente' in utente), e dalla prenotazione in cui è inserito;
- **Liste preferiti:** identificate dal nome della lista e dall'utente a cui appartengono (utenti diversi possono avere liste con lo stesso nome);
- **Utenti:** identificati mediante un identificativo univoco associato ad ogni iscritto al servizio;
- **Alloggio:** identificato mediante un identificativo univoco associato ad ogni alloggio registrato nel servizio;
- **Cellulare:** identificato dal numero di cellulare, presupponendo che non possano esistere due numeri di telefono uguali;
- **Ospite e host:** identificati mediante vincolo di chiave esterna da utente;
- **Servizi:** identificati dal nome del servizio e, mediante vincolo di chiave esterna, dall'alloggio a cui appartengono;
- **Foto:** identificata da un id_foto univoco associato ad ogni foto;
- **Recensione ospite e recensione host:** identificati mediante vincoli di chiave esterna da recensione;
- **Valutazione:** identificata mediante vincolo di chiave esterna dalla recensione a cui appartiene e dal nome del servizio a cui si riferisce;
- **Recensione:** identificata da data e orario in cui è stata scritta;
- **Commento:** identificato da data e orario in cui è stato scritto.

2.4.1 Schema E-R ristrutturato



2.4.2 Business rules

1. L'attributo 'superhost' in host vale true se sono soddisfatti i seguenti requisiti:
 - Devono aver terminato almeno 10 prenotazioni, 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 tutte le prenotazioni in tutte gli alloggi di sua proprietà.
2. L'attributo 'visibilità' in recensione vale true quando:
 - Ospite e host hanno fatto la recensione;
 - Se uno dei due non ha fatto la recensione, l'altra diventa visibile dopo 7 giorni dalla fine della prenotazione.
3. Un utente non può effettuare prenotazioni per alloggi di cui è host.
4. Un alloggio non deve essere eliminato definitivamente dal database se è stata effettuata almeno una prenotazione su di esso.
5. Non ci possono essere più prenotazioni per uno stesso che si sovrappongano.
6. Il numero degli ospiti indicato in fase di prenotazione non può superare il numero di posti letto dell'alloggio.
7. Data prenotazione < data inizio, data fine, data cancellazione, data rifiuto.
8. Data inizio + numero notti = data fine && data inizio > data cancellazione, data rifiuto.
9. I nominativi possono essere aggiunti alla prenotazione solo se riguardano utenti già iscritti al servizio.
10. L'attributo 'segnalazione' di commento e recensione vale true quando un utente lo ha segnalato al servizio in quanto ritenuto offensivo, false altrimenti.
11. L'attributo 'verificato' in utente vale true solo se l'attributo 'carta d'identità' non è NULL, false altrimenti.

2.4.3 Regole di derivazione

1. La valutazione complessiva del soggiorno è una media delle valutazioni ricevute sui singoli servizi.
2. Il rating medio di un alloggio è la media delle valutazioni complessive date dagli utenti.
3. Il numero di recensioni di un alloggio è la somma del numero di recensioni scritte dagli ospiti per l'alloggio con campo 'testo alloggio' diverso da NULL.
4. Il numero di recensioni dell'host è la somma del numero di recensioni scritte dagli utenti per l'host con campo 'testo host' diverso da NULL.
5. Il tasso di cancellazione di un host si ottiene come (numero di prenotazioni cancellate/numero totale di prenotazioni confermate) %, in cui le prenotazioni considerate riguardano alloggi di proprietà dell'host.
6. Il costo totale di una prenotazione si calcola come: 'numero notti' x 'prezzo per notte a persona' x 'numero ospiti'.

2.5 Schema relazionale

COMMENTO (data, orario, segnalazione, testo, dataRecensione, orarioRecensione)

Risposta (data1, orario1, data2, orario2)

RECENSIONE (data, orario, visibilità, segnalazione)

VALUTAZIONE (dataRecensione, orarioRecensione, servizio, punteggio)

RECENSIONE OSPITE (dataRecensione, orarioRecensione, Id_alloggio, testoHost, testoAlloggio)

RECENSIONE HOST (dataRecensione, orarioRecensione, Id_utente, commentoTestuale)

O-Ro (dataRecensione, orarioRecensione, Id_ospite)

O-Rh (dataRecensione, orarioRecensione, Id_host)

OSPITE (Id_utente)

HOST (Id_host, tassoCancellazione, superhost)

UTENTE (Id_utente, nome, cognome, email, password, verificato, cartaD'Identità*)

CELLULARE (Id_utente, numeroCellulare)

LISTE PREFERITI (NomeLista, Id_utente)

Contenuto (NomeLista, Id_utente, Id_alloggio)

ALLOGGIO (Id_alloggio, nome, descrizione, numero, via, CAP, comune, numRecensioniHost, costiPulizia, numeroLetti, orarioCheckIn, orarioCheckOut, numRecensioniAlloggio, tipoAlloggio, Id_host, PrezzoPerNotteAPersona)

FOTO(Id_foto, Id_alloggio, immagine)

P. CANCELLATA (dataPrCancellata, Id_utente, Id_alloggio, dataCancellazione)

U-Pr (dataPrenotazione, Id_utente, Id_alloggio, metodoPagamento)

PRENOTAZIONE (dataPrenotazione, Id_utente, Id_alloggio, numeroOspiti, NumeroNotti)

P. CONFERMATA (dataPrConfermata, Id_utente, Id_alloggio, dataInizio)

P. TERMINATA (dataPrTerminata, Id_utente, Id_alloggio, dataFine)

P. RIFIUTATA (dataPrRifiutata, Id_utente, Id_alloggio, dataRifiuto)

NOMINATIVO (dataPrenotazione, Id_utente, Id_alloggio, codiceUtente)

SERVIZI (nome, Id_alloggio, descrizione)

Con vincoli di integrità referenziale:

- dataRecensione e orarioRecensione in COMMENTO con la chiave di RECENSIONE;
- data1 e data2, orario1 e orario2 in Risposta con la chiave di COMMENTO;
- dataRecensione e orarioRecensione in VALUTAZIONE con la chiave di RECENSIONE OSPITE;
- dataRecensione e orarioRecensione in RECENSIONE OSPITE con la chiave di RECENSIONE, Id_alloggio in RECENSIONE OSPITE con la chiave di ALLOGGIO;
- dataRecensione e orarioRecensione in RECENSIONE HOST con la chiave di RECENSIONE, Id_utente in RECENSIONE HOST con la chiave di OSPITE;
- Id_ospite in O-Ro con la chiave di OSPITE, dataRecensione e orarioRecensione in O-Ro con la chiave di RECENSIONE OSPITE;
- Id_host in O-Rh con la chiave di HOST, dataRecensione e orarioRecensione in O-Rh con la chiave di RECENSIONE HOST;
- Id_utente in OSPITE con la chiave di UTENTE;
- Id_host in HOST con la chiave di UTENTE;
- Id_utente in CELLULARE con Id_utente in UTENTE;
- Id_utente in LISTE PREFERITI con Id_utente in UTENTE;
- NomeLista, Id_utente in Contenuto con la chiave di LISTE PREFERITI, Id_alloggio in Contenuto con la chiave di ALLOGGIO;
- Id_host in ALLOGGIO con la chiave di HOST;
- DataPrCancellata, Id_utente e Id_alloggio in P. CANCELLATA con la chiave di P. CONFERMATA;
- DataPrenotazione, Id_utente e Id_alloggio in U-Pr con la chiave di PRENOTAZIONE;
- Id_utente in PRENOTAZIONE con la chiave di UTENTE, Id_alloggio in PRENOTAZIONE con la chiave di ALLOGGIO;
- DataPrConfermata, Id_utente e Id_alloggio in P. CONFERMATA con la chiave di PRENOTAZIONE;
- DataPrTerminata, Id_utente e Id_alloggio in P. TERMINATA con la chiave di P. CONFERMATA;
- DataPrRifiutata, Id_utente e Id_alloggio in P. RIFIUTATA con la chiave di PRENOTAZIONE;
- DataPrenotazione, Id_utente e Id_alloggio in NOMINATIVO con la chiave di PRENOTAZIONE, codiceUtente in NOMINATIVO con la chiave di UTENTE;
- Id_alloggio in SERVIZI con la chiave di ALLOGGIO.
- Id_alloggio in FOTO con la chiave di ALLOGGIO.

3. Implementazione

3.1 DDL di creazione del database

```
begin;  
  
drop table if exists UTENTE cascade;  
  
drop table if exists ALLOGGIO cascade;  
  
drop table if exists PRENOTAZIONE cascade;  
  
drop table if exists OSPITE cascade;  
  
drop table if exists HOST cascade;  
  
drop table if exists COMMENTO cascade;  
  
drop table if exists LISTA_PREFERITI cascade;  
  
drop table if exists CELLULARE cascade;  
  
drop table if exists U_PR cascade;  
  
drop table if exists NOMINATIVO cascade;  
  
drop table if exists P_CONFERMATA cascade;  
  
drop table if exists P_RIFIUTATA cascade;  
  
drop table if exists P_TERMINATA cascade;  
  
drop table if exists P_CANCELLATA cascade;  
  
drop table if exists CONTENUTO cascade;  
  
drop table if exists SERVIZI cascade;  
  
drop table if exists OFFERTA cascade;  
  
drop table if exists FOTO cascade;  
  
drop table if exists RECENSIONE cascade;  
  
drop table if exists RECENSIONE_OSPITE cascade;  
  
drop table if exists RECENSIONE_HOST cascade;  
  
drop table if exists O_RO cascade;  
  
drop table if exists O_RH cascade;  
  
drop table if exists RISPOSTA cascade;  
  
drop table if exists VALUTAZIONE cascade;
```

```
create table UTENTE (  
  IdUtente varchar(8),  
  Nome varchar(8) not null,  
  Cognome varchar(8) not null,  
  Email varchar(40) not null,  
  Password varchar(12) not null,  
  Verificato boolean default false not null,  
  CartaDIdentità bytea,  
  primary key(IdUtente),  
  check(Email like ' _%@_%._%'),  
  check>Password like ' _____%')  
);  
  
create table OSPITE (  
  IdOspite varchar(8),  
  primary key(IdOspite),  
  foreign key (IdOspite) references UTENTE(IdUtente) on update cascade on delete cascade  
);  
  
create table HOST (  
  IdHost varchar(8),  
  TassoCancellazione real default 0.00 not null,  
  Superhost boolean default false not null,  
  primary key(IdHost),  
  foreign key (IdHost) references UTENTE(IdUtente) on update cascade on delete cascade,  
  check(TassoCancellazione>=0.00 AND TassoCancellazione<=1.00)  
);
```



```
create table ALLOGGIO (  
    IdAlloggio varchar(8),  
    Nome varchar(20) not null,  
    Descrizione text not null,  
    Numero smallint,  
    Via varchar(20) not null,  
    Cap int not null,  
    Comune varchar(15) not null,  
    NumeroRecensioniHost int default 0 not null,  
    NumeroRecensioniAlloggio int default 0 not null,  
    CostiPulizia real default 0 not null,  
    PrezzoPerNotteAPersona real default 0 not null,  
    NumeroLetti smallint default 0 not null,  
    OrarioCheckIn time(2) not null,  
    OrarioCheckOut time(2) not null,  
    TipoAlloggio varchar(25) not null,  
    IdHost varchar(8) not null,  
    primary key(IdAlloggio),  
    foreign key(IdHost) references HOST(IdHost) on update cascade on delete cascade,  
    check( TipoAlloggio like 'Intero Appartamento' OR TipoAlloggio like 'Stanza Condivisa' OR TipoAlloggio like 'Stanza Privata'),  
    check( CAP>=0 AND CAP <=99999),  
    check( NumeroLetti>0),  
    check( PrezzoPerNotteAPersona>0)  
);  
  
create table PRENOTAZIONE (  
    DataPrenotazione date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    NumeroOspiti smallint default 0 not null,  
    NumeroNotti smallint not null,  
    primary key(DataPrenotazione, IdUtente, IdAlloggio),  
    foreign key(IdUtente) references UTENTE(IdUtente) on update cascade on delete set null,  
    foreign key(IdAlloggio) references ALLOGGIO(IdAlloggio) on update cascade on delete cascade,  
    check(NumeroOspiti>0),  
    check(NumeroNotti>0)  
);
```

```
create table LISTA_PREFERITI (  
    NomeLista varchar(8),  
    IdUtente varchar(8),  
    primary key(IdUtente, NomeLista),  
    foreign key(IdUtente) references UTENTE(IdUtente) on update cascade on delete cascade  
);
```

```
create table CELLULARE (  
    IdUtente varchar(8),  
    NumeroCellulare char(12),  
    primary key(IdUtente, NumeroCellulare),  
    foreign key(IdUtente) references UTENTE(IdUtente) on update cascade on delete cascade,  
    check(NumeroCellulare not like '%[^0-9]%')  
);
```

```
create table U_PR (  
    DataPrenotazione date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    MetodoDiPagamento varchar(32) not null,  
    primary key(DataPrenotazione, IdUtente, IdAlloggio),  
    foreign key(DataPrenotazione, IdUtente, IdAlloggio) references PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio) on update cascade on delete cascade  
);
```

```
create table NOMINATIVO (  
    DataPrenotazione date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    CodiceUtente varchar(8),  
    primary key(DataPrenotazione, IdUtente, IdAlloggio, CodiceUtente),  
    foreign key(DataPrenotazione, IdUtente, IdAlloggio) references PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio) on update cascade on delete cascade,  
    foreign key(CodiceUtente) references UTENTE(IdUtente) on update cascade on delete cascade  
);
```

```
create table P_RIFIUTATA (  
    DataPrRifiutata date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    DataRifiuto date not null,  
    primary key(DataPrRifiutata, IdUtente, IdAlloggio),  
    foreign key(DataPrRifiutata, IdUtente, IdAlloggio) references PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio) on update cascade on delete cascade,  
    check(DataPrRifiutata<=DataRifiuto)  
);  
  
create table P_CONFERMATA (  
    DataPrConfermata date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    DataInizio date not null,  
    primary key(DataPrConfermata, IdUtente, IdAlloggio),  
    foreign key(DataPrConfermata, IdUtente, IdAlloggio) references PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio) on update cascade on delete cascade,  
    check(DataPrConfermata<DataInizio)  
);  
  
create table P_CANCELLATA (  
    DataPrCancellata date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    DataCancellazione date not null,  
    primary key(DataPrCancellata, IdUtente, IdAlloggio),  
    foreign key(DataPrCancellata, IdUtente, IdAlloggio) references P_CONFERMATA(DataPrConfermata, IdUtente, IdAlloggio) on update cascade on delete cascade,  
    check(DataPrCancellata<=DataCancellazione)  
);
```

```
create table P_TERMINATA (  
    DataPrTerminata date,  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    DataFine date not null,  
    primary key(DataPrTerminata, IdUtente, IdAlloggio),  
    foreign key(DataPrTerminata, IdUtente, IdAlloggio) references P_CONFERMATA(DataPrConfermata, IdUtente, IdAlloggio) on update cascade on delete cascade,  
    check(DataPrTerminata<DataFine)  
);  
  
create table CONTENUTO (  
    NomeLista varchar(8),  
    IdUtente varchar(8),  
    IdAlloggio varchar(8),  
    primary key(NomeLista, IdUtente, IdAlloggio),  
    foreign key(NomeLista, IdUtente) references LISTA_PREFERITI(NomeLista, IdUtente) on update cascade on delete cascade,  
    foreign key(IdAlloggio) references ALLOGGIO(IdAlloggio) on update cascade on delete cascade  
);  
  
create table SERVIZI (  
    NomeServizio varchar(20),  
    IdAlloggio varchar(8),  
    Descrizione text not null,  
    primary key (NomeServizio, IdAlloggio),  
    foreign key(IdAlloggio) references ALLOGGIO(IdAlloggio) on update cascade on delete cascade  
);  
  
create table RECENSIONE (  
    DataRecensione date,  
    Orario time(2),  
    Visibilità boolean default false not null,  
    Segnalazione boolean default false not null,  
    primary key(DataRecensione, Orario)  
);
```

```
create table COMMENTO (  
    DataRecensione date not null,  
    OrarioRecensione time(2) not null,  
    Testo text,  
    DataCom date,  
    Orario time(2),  
    Segnalazione boolean default false not null,  
    primary key(DataCom, Orario),  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE(DataRecensione, Orario) on update no action on delete no action,  
    check(DataCom>=DataRecensione OR ((DataCom=DataRecensione AND Orario>=OrarioRecensione)))  
);  
  
create table FOTO (  
    IdFoto varchar(8),  
    Immagine bytea,  
    IdAlloggio varchar(8),  
    primary key(IdFoto),  
    foreign key(IdAlloggio) references ALLOGGIO(IdAlloggio) on update cascade on delete cascade  
);  
  
create table RECENSIONE_OSPITE (  
    DataRecensione date,  
    OrarioRecensione time(2),  
    IdAlloggio varchar(8) not null,  
    TestoHost text,  
    TestoAlloggio text,  
    primary key(DataRecensione, OrarioRecensione),  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE(DataRecensione, Orario) on update no action on delete no action,  
    foreign key (IdAlloggio) references ALLOGGIO(IdAlloggio) on update cascade on delete cascade  
);
```

```
create table RECENSIONE_HOST (  
    DataRecensione date,  
    OrarioRecensione time(2),  
    IdUtente varchar(8) not null,  
    CommentoTestuale text,  
    primary key(DataRecensione, OrarioRecensione),  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE(DataRecensione, Orario) on update no action on delete no action,  
    foreign key (IdUtente) references OSPITE(IdOspite) on update cascade on delete cascade  
);  
  
create table O_RO (  
    IdOspite varchar(8),  
    DataRecensione date,  
    OrarioRecensione time(2),  
    primary key(DataRecensione, OrarioRecensione),  
    foreign key (IdOspite) references OSPITE(IdOspite) on update cascade on delete set null,  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE_OSPITE(DataRecensione, OrarioRecensione) on update no action on delete cascade  
);  
  
create table O_RH (  
    IdHost varchar(8),  
    DataRecensione date,  
    OrarioRecensione time(2),  
    primary key(DataRecensione, OrarioRecensione),  
    foreign key (IdHost) references HOST(IdHost) on update cascade on delete cascade,  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE_HOST(DataRecensione, OrarioRecensione) on update no action on delete cascade  
);
```

```
create table VALUTAZIONE (  
    DataRecensione date,  
    OrarioRecensione time(2),  
    Servizio varchar(15),  
    Punteggio smallint not null,  
    primary key(DataRecensione, OrarioRecensione, Servizio),  
    foreign key (DataRecensione, OrarioRecensione) references RECENSIONE_OSPITE(DataRecensione, OrarioRecensione) on update no action on delete cascade,  
    check(Punteggio>=1 AND Punteggio<=5)  
);
```

```
create table RISPOSTA (  
    Data1 date,  
    Data2 date,  
    Orario1 time(2),  
    Orario2 time(2),  
    primary key(Data1, Data2, Orario1, Orario2),  
    foreign key (Data1, Orario1) references COMMENTO(DataCom, Orario) on update no action on delete no action,  
    foreign key (Data2, Orario2) references COMMENTO(DataCom, Orario) on update no action on delete no action  
);  
commit;
```

3.2 DML di popolamento di tutte le tabelle del database

```
insert into UTENTE(IdUtente, Nome, Cognome, Email, Password, Verificato, CartaDIdentità)
values('1', 'Paolo', 'Rossi', 'paolorossi@gmail.com', 'PaoloRos', false, NULL);
```

```
insert into UTENTE(IdUtente, Nome, Cognome, Email, Password, Verificato, CartaDIdentità)
values('2', 'Dennis', 'Montesi', 'dennismontesi@gmail.com', 'DenniMon', false, NULL);
```

```
insert into UTENTE(IdUtente, Nome, Cognome, Email, Password, Verificato, CartaDIdentità)
values('3', 'Michele', 'Merico', 'michelemerico@gmail.com', 'MicheMer', false, NULL);
```

```
insert into OSPITE(IdOspite) values ('1');
```

```
insert into HOST(IdHost, TassoCancellazione, Superhost) values ('2', 0, false);
```

```
insert into HOST(IdHost, TassoCancellazione, Superhost) values ('3', 0, false);
```

```
insert into ALLOGGIO(IdAlloggio, Nome, Descrizione, Numero, Via, Cap, Comune, NumeroRecensioniHost,
NumeroRecensioniAlloggio, CostiPulizia, PrezzoPerNotteAPersona, NumeroLetti, OrarioCheckIn, OrarioCheckOut,
TipoAlloggio, IdHost)
values ('1', 'Trullo', 'Meraviglioso trullo in zone rurali', NULL, 'Contrada Bella', 10567, 'Fantasia', 0, 0, 100, 50, 4,
'10:00:00', '11:00:00', 'Intero Appartamento', '2');
```

```
insert into ALLOGGIO(IdAlloggio, Nome, Descrizione, Numero, Via, Cap, Comune, NumeroRecensioniHost,
NumeroRecensioniAlloggio, CostiPulizia, PrezzoPerNotteAPersona, NumeroLetti, OrarioCheckIn, OrarioCheckOut,
TipoAlloggio, IdHost)
values ('2', 'Casa', 'Meravigliosa casa in città', 3, 'Via Alan Turing', 72018, 'San Michele', 0, 0, 85, 60, 2, '09:00:00',
'12:00:00', 'Stanza Privata', '3');
```

```
insert into PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio, NumeroOspiti, NumeroNotti)
values ('01/01/2020', '1', '2', 3, 3);
```

```
insert into PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio, NumeroOspiti, NumeroNotti)
values ('03/03/2020', '2', '2', 4, 4);
```

```
insert into PRENOTAZIONE(DataPrenotazione, IdUtente, IdAlloggio, NumeroOspiti, NumeroNotti)
values ('02/02/2021', '1', '1', 2, 5);
```

```
insert into LISTA_PREFERITI(NomeLista, IdUtente) values ('Inverno', '1');
```

```
insert into CELLULARE(IdUtente, NumeroCellulare) values ('1', '393324567899');
```

```
insert into CELLULARE(IdUtente, NumeroCellulare) values ('2', '613455566778');
```

```
insert into CELLULARE(IdUtente, NumeroCellulare) values ('3', '011112341753');
```

```
insert into U_PR(DataPrenotazione, IdUtente, IdAlloggio, MetodoDiPagamento)
values ('01/01/2020', '1', '2', 'IT3214151718');
```

```
insert into U_PR(DataPrenotazione, IdUtente, IdAlloggio, MetodoDiPagamento)
values ('02/02/2021', '1', '1', 'IT3981231718');
```

```
insert into U_PR(DataPrenotazione, IdUtente, IdAlloggio, MetodoDiPagamento)
values ('03/03/2020', '2', '2', 'IT12345678');
```



```
insert into NOMINATIVO(DataPrenotazione, IdUtente, IdAlloggio, CodiceUtente)
values ('01/01/2020', '1', '2', '2');
```

```
insert into NOMINATIVO(DataPrenotazione, IdUtente, IdAlloggio, CodiceUtente)
values ('02/02/2021', '1', '1', '3');
```

```
insert into P_RIFIUTATA(DataPrRifiutata, IdUtente, IdAlloggio, DataRifiuto)
values ('02/02/2021', '1', '1', '12/02/2021');
```

```
insert into P_CONFERMATA(DataPrConfermata, IdUtente, IdAlloggio, DataInizio)
values ('01/01/2020', '1', '2', '01/20/2020');
```

```
insert into P_CONFERMATA(DataPrConfermata, IdUtente, IdAlloggio, DataInizio)
values ('03/03/2020', '2', '2', '03/24/2020');
```

```
insert into P_CANCELLATA(DataPrCancellata, IdUtente, IdAlloggio, DataCancellazione)
values ('01/01/2020', '1', '2', '01/20/2020');
```

```
insert into P_TERMINATA(DataPrTerminata, IdUtente, IdAlloggio, DataFine)
values ('03/03/2020', '2', '2', '03/28/2020');
```

```
insert into CONTENUTO(NomeLista, IdUtente, IdAlloggio)
values ('Inverno', '1', '1');
```

```
insert into CONTENUTO(NomeLista, IdUtente, IdAlloggio)
values ('Inverno', '1', '2');
```

```
insert into SERVIZI(NomeServizio, IdAlloggio, Descrizione)
values ('Cucina', '1', 'Ampia cucina');
```

```
insert into SERVIZI(NomeServizio, IdAlloggio, Descrizione)
values ('Bagno', '1', 'Bagno con box doccia');
```

```
insert into SERVIZI(NomeServizio, IdAlloggio, Descrizione)
values ('Alexa', '2', 'Alexa presente in loco');
```

```
insert into RECENSIONE(DataRecensione, Orario, Visibilità, Segnalazione)
values ('03/29/2020', '10:09:27', false, false);
```

```
insert into RECENSIONE(DataRecensione, Orario, Visibilità, Segnalazione)
values ('03/29/2020', '22:14:11', false, false);
```

```
insert into FOTO(IdFoto, Immagine, IdAlloggio)
values ('123', NULL, '1');
```

```
insert into FOTO(IdFoto, Immagine, IdAlloggio)
values ('4463', NULL, '2');
```

```
insert into FOTO(IdFoto, Immagine, IdAlloggio)
values ('4464', NULL, '2');
```

```
insert into RECENSIONE_OSPITE(DataRecensione, OrarioRecensione, IdAlloggio, TestoHost, TestoAlloggio)
values ('03/29/2020', '10:09:27', '2', 'Gentile e cordiale', 'Pulito e accogliente');
```

```
insert into RECENSIONE_HOST(DataRecensione, OrarioRecensione, IdUtente, CommentoTestuale)
values ('03/29/2020', '22:14:11', '1', 'Molto puliti e gentili');
```

```
insert into O_RO(IdOspite, DataRecensione, OrarioRecensione)
values ('1', '03/29/2020', '10:09:27');
```

```
insert into O_RH(IdHost, DataRecensione, OrarioRecensione)
values ('3', '03/29/2020', '22:14:11');
```

```
insert into VALUTAZIONE(DataRecensione, OrarioRecensione, Servizio, Punteggio)
values ('03/29/2020', '10:09:27', 'Pulizie', 4);
```

```
insert into VALUTAZIONE(DataRecensione, OrarioRecensione, Servizio, Punteggio)
values ('03/29/2020', '10:09:27', 'Posizione', 3);
```

```
insert into VALUTAZIONE(DataRecensione, OrarioRecensione, Servizio, Punteggio)
values ('03/29/2020', '10:09:27', 'Qualità/Prezzo', 5);
```

```
insert into COMMENTO(DataRecensione, OrarioRecensione, Testo, DataCom, Orario, Segnalazione)
values('03/29/2020', '22:14:11', 'Grazie mille', '03/29/2020', '22:52:34', false);
```

```
insert into COMMENTO(DataRecensione, OrarioRecensione, Testo, DataCom, Orario, Segnalazione)
values('03/29/2020', '22:14:11', 'Prego', '03/30/2020', '08:32:10', false);
```

```
insert into RISPOSTA(Data1, Data2, Orario1, Orario2)
values ('03/29/2020', '03/30/2020', '22:52:34', '08:32:10');
```

3.3 Operazioni di modifica e cancellazione

UPDATE UTENTE SET IdUtente='5' WHERE IdUtente like '1';

Aggiorniamo l'Id_utente associato ad un utente: la modifica si ripercuote su tutte le tabelle che dipendono da UTENTE.

Prima:

	idutente [PK] character varying (8)	nome character varying (8)	cognome character varying (8)	email character varying (40)	password character varying (12)	verificato boolean	cartadidentità bytea
1	1	Paolo	Rossi	paolorossi@gmail.com	PaoloRos	false	[null]
2	2	Dennis	Montesi	dennismontesi@gmail.com	DenniMon	false	[null]
3	3	Michele	Merico	michelemerico@gmail.com	MicheMer	false	[null]

Dopo:

	idutente [PK] character varying (8)	nome character varying (8)	cognome character varying (8)	email character varying (40)	password character varying (12)	verificato boolean	cartadidentità bytea
1	2	Dennis	Montesi	dennismontesi@gmail.com	DenniMon	false	[null]
2	3	Michele	Merico	michelemerico@gmail.com	MicheMer	false	[null]
3	5	Paolo	Rossi	paolorossi@gmail.com	PaoloRos	false	[null]

UPDATE RECENSIONE SET DataRecensione='03/30/2020' WHERE DataRecensione= '03/29/2020';

Proviamo a modificare la data di scrittura di una recensione: l'operazione non è consentita in quanto si genererebbero inconsistenze nelle date di scrittura di eventuali altri commenti (ordine temporale non più rispettato).

ERROR: update or delete on table "recensione" violates foreign key constraint "recensione_ospite_datarecensione_fkey" on table "recensione_ospite"
DETAIL: Key (datarecensione, orario)=(2020-03-29, 10:09:27) is still referenced from table "recensione_ospite".
SQL state: 23503

DELETE FROM FOTO WHERE IdFoto like '4463' AND IdAlloggio like '2';

Eliminiamo una foto associata ad un alloggio: l'operazione è sempre consentita.

Prima:

	idfoto [PK] character varying (8)	immagine bytea	idalloggio character varying (8)
1	123	[null]	1
2	4463	[null]	2
3	4464	[null]	2

Dopo:

	idfoto [PK] character varying (8)	immagine bytea	idalloggio character varying (8)
1	123	[null]	1
2	4464	[null]	2

DELETE FROM COMMENTO WHERE DataCom='03/29/2020' AND Orario='22:52:34';

Proviamo ad eliminare un commento dalla relazione COMMENTO: l'operazione non è consentita a causa della presenza della clausola "on delete no action" e dei vincoli di chiave esterna delle tabelle che dipendono da COMMENTO.

ERROR: update or delete on table "commento" violates foreign key constraint "risposta_data1_fkey" on table "risposta"
DETAIL: Key (datacom, orario)=(2020-03-29, 22:52:34) is still referenced from table "risposta".
SQL state: 23503