UNIVERSITÀ DEGLI STUDI DI TORINO

Corso di laurea in Informatica

Laboratorio Basi di Dati 2021/22

- Progetto piattaforma di home booking -

Relazione del progetto di:

Persico Loris - 949352

Rinaldi Flavio - 955750

Indice

1. Progettazione concettuale

1.1. Requisiti iniziali

Legenda:

= Entità

= Attributi

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.

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.

1.2. Glossario dei termini

Seguono i termini fondamentali identificati per modellare in maniera non ambigua il dominio

Termini	Descrizione	Sinonimi	Collegamenti
Utente	Persona che usufruisce del servizio Airbnb	Cliente	Lista, Prenotazione, Recensione alloggio
Alloggio	Luogo destinato a dimora temporanea	Casa	Prenotazione
Host	Persona che ospita altri utenti del servizio in uno o più alloggi di loro proprietà		Utente, Recensione
Superhost	Persona che va ben oltre i propri doveri di ospitalità ed è un esempio lampante di come dovrebbe		Host

	essere un host		
Lista	Elenco di una o più case aggiunte nei preferiti	Elenco	Utente
Recensione	Commento testuale e valutazione dell'alloggio o dell' ospitante	Commento	Host,Utente
Prenotazione	Riservare un posto in una struttura o alloggio, in una determinata data	Riservare	Alloggio, Utente
Thread di discussione	Sotto recensioni		Recensione

1.3. Requisiti rivisti e strutturati

Dominio: AIRBNB
Bisogna realizzare un servizio che permette di affittare e prenotare alloggi di vario tipo, deve contenere informazioni riguardo gli utenti, gli ospiti, gli host e i superhost, le recensioni lasciate agli alloggi e all'host.

Alloggi:

- Gli alloggi possono essere stanze private, stanze condivise, interi appartamenti e si caratterizzano in:
    - Nome
    - Indirizzo
    - Descrizione
    - Prezzo per notte per persona
    - Costi di pulizia
    - Foto
    - Servizi
    - Numero letti
    - Check in
    - Check out
    - Host a cui appartiene
    - Rating medio
    - Numero recensioni

Utenti:

- Gli utenti sono le persone che si iscrivono al servizio Airbnb e devono fornire:
    - Indirizzo email
    - Password
    - Nome
    - Cognome
    - Numero o numeri di telefono
    - Foto carta d'identità
    - Metodo di pagamento

- Gli utenti si suddividono in ospiti o "host" e possono aggiungere alcuni alloggi tra i preferiti e possono avere diverse liste in base al viaggio che vogliono compiere.
- Possono prenotare gli alloggi e devono indicare la data d'inizio soggiorno e la data di fine soggiorno e il numero di ospiti.
- Gli utenti una volta finito il soggiorno possono fare una recensione all'alloggio e all'host e una valutazione da 1 a 5 riguardo la pulizia, la comunicazione, la posizione, e il rapporto qualità prezzo.

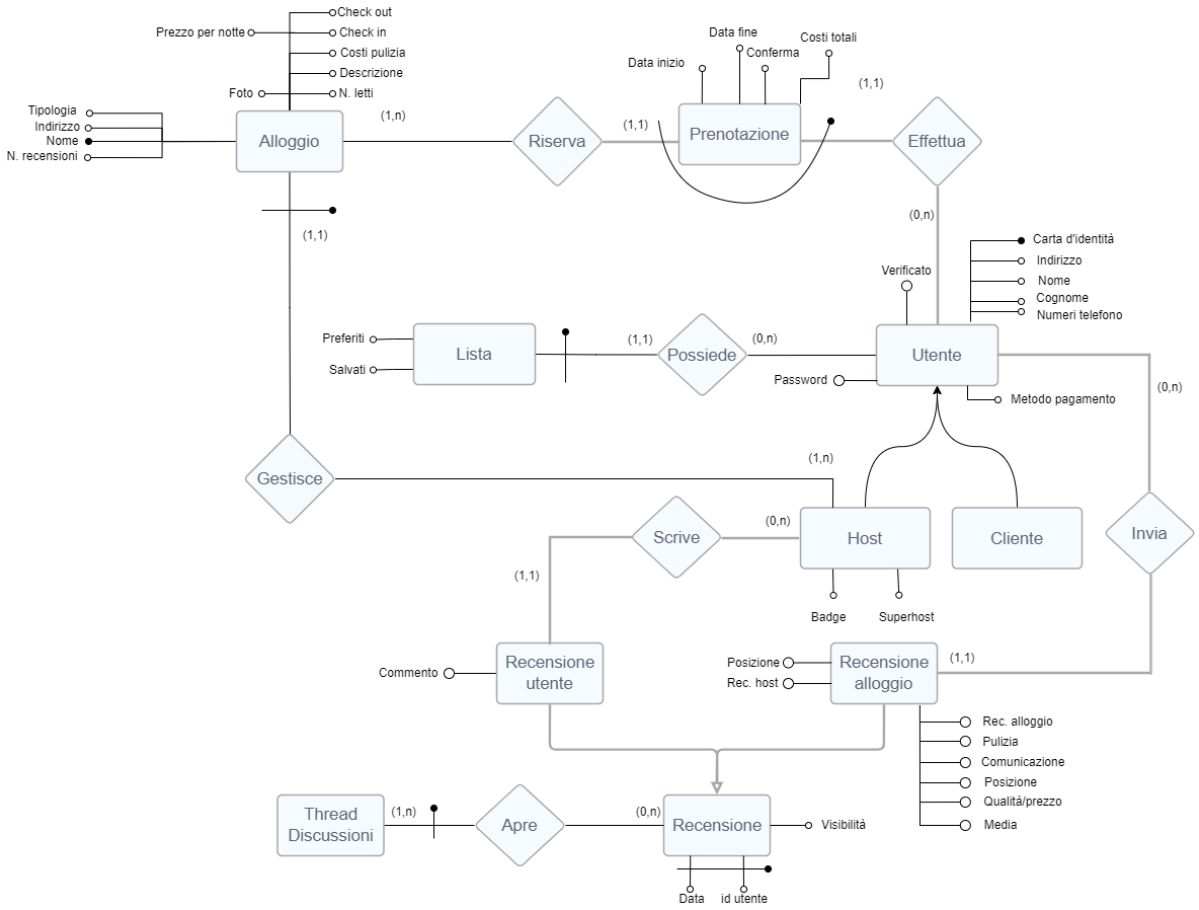
Host e Superhost:

- Gli host sono le persone che ospitano gli altri utenti in uno o più alloggi di loro proprietà.
- L'host può valutare l'utente che ha soggiornato nel suo alloggio e cio comprende solo un commento testuale.
- Un Host può diventare Superhost se rispetta 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.

Recensioni:

- Le recensioni vengono fatte sia dagli utenti sia dagli host
- Possono essere aperti thread di discussioni

1.4. Schema E-R + business rules



La progettazione concettuale è il primo step nel quale il database comincia a prendere forma.

Business rules:

1. Se l'utente fornisce la foto della carta d'identità, viene riconosciuto come verificato.
2. Il numero di ospiti in una prenotazione deve essere minore o uguale al numero di posti letto dell'alloggio prenotato.
3. Un host non può prenotare il suo stesso alloggio.
4. Un host non può recensire il proprio alloggio.
5. Per diventare superhost bisogna:
   5.1. Aver completato almeno 10 soggiorni, per un totale di almeno 100 notti.
   5.2. Devono aver conservato un tasso di cancellazione dell'1% (una cancellazione ogni 100 prenotazioni) massimo.

5.3. Devono aver mantenuto una valutazione complessiva di 4,8 considerando tutti i soggiorni in tutte le case di sua proprietà.
6. Una volta a settimana viene effettuato un calcolo per aggiornare il tasso di cancellazione di ciascun host.
7. Una volta al giorno si controllano le condizioni per la qualifica di superhost e viene aggiornato lo status degli host.
8. Una volta al mese viene calcolata la classifica degli alloggi più graditi
9. La data di inizio soggiorno non deve essere successiva alla data di fine soggiorno.
10. Un utente può usufruire dell'alloggio solo se la prenotazione è confermata
11. L'utente prima di poter scrivere una recensione deve aver soggiornato.
12. Se un host non riuscisse a rispettare l'impegno preso per una prenotazione a causa di una circostanza attenuante o di condizioni simili al di fuori del suo controllo, dovrà prendersi la responsabilità di cancellare la prenotazione il prima possibile per consentire al suo ospite di modificare i propri piani.

2. Progettazione logica

Lo scopo della progettazione logica è quello di raggiungere un modello relazionale. Preliminarmente alla progettazione è necessario avere però a disposizione la tabella dei volumi e delle frequenze necessarie.

2.1. Tavola dei volumi

Tabella volumi			
Concetto	**Tipo**	**Volumi**	**Descrizione**
Utente	E	350.000	Coloro che hanno un account airbnb
Alloggio	E	30.000	Alcuni host hanno più di un alloggio
Prenotazione	E	600.000	Prenotazione utente alloggio
Lista	E	90.000	Liste di utenti sugli alloggi preferiti
Host	E	15.000	Host + superhost

Cliente	E	300.000	Totale clienti che hanno usato Airbnb
Recensioni utente	E	100.000	Host che recensiscono gli ospiti + ospiti che recensiscono gli host
Recensione alloggio	E	300.000	Recensioni sugli alloggi da parte degli ospiti
Recensione	E	400.000	Recensioni alloggio + recensioni persona
Thread discussioni	E	200.000	Sotto recensioni
Scrivere	R	100.000	Host *scrive* recensione all'utente
Aprire	R	200.000	Ogni recensione puo *aprire* un thread di discussioni
Inviare	R	300.000	Utente *invia* una recensione all'alloggio
Effettuare	R	600.000	Prenotazione *effettuate* da Utenti
Gestire	R	3000	Il 10% degli ospiti *gestisce* un alloggio
Possedere	R	90.000	Utenti che *posseggono* una lista
Riservare	R	600.000	Alloggio viene *riservato* per prenotazione

Specifica il numero stimato di istanze per ogni entità (E) e relazione (R) dello schema
I valori sono necessariamente approssimati, ma indicativi

2.2. Tavola delle operazioni

Operazioni	Tipo	Frequenza	Descrizione
Registrazione utente	I	500/giorno	
Effettuare prenotazione	I	1000/giorno	
Inserimento nuova recensione	I	1000/giorno	Recensioni a alloggi/ospiti/host
Calcolo costo prenotazione	I	1000/giorno	
Recensione alloggio da parte di un utente	I	700/giorno	
Recensioni tra Host e Ospiti	I	300/giorno	
Commento di una recensione da parte dell'host/utente	I	2000/giorno	
Caricamento nuovo alloggio sul sito	I	150/giorno	
Aggiornamento tasso di cancellazione di Host	B	1 Settimana	
Controllo qualifica Superhost e aggiornato lo stato degli host	B	1/Giorno	
Calcolo media delle valutazioni ricevute	B	1/Mese	

Operazioni	Tipo	Frequenza	Descrizione
Registrazione utente	I	500/giorno	
Effettuare prenotazione	I	1000/giorno	
Inserimento nuova recensione	I	1000/giorno	Recensioni a alloggi/ospiti/host
Calcolo classifica alloggi più graditi	B	1/Mese	
Aggiornamento dati di un alloggio	I	5/giorno	Aggiorno tutti i dati relativi agli alloggi che sono stati riservati da una prenotazione

2.3. Ristrutturazione dello schema E-R

2.3.1. Analisi delle ridondanze

- Una ridondanza in uno schema E-R è una informazione significativa ma derivabile da altre
- In questa fase si decide se eliminare le ridondanze eventualmente presenti o mantenerle
- Se si mantiene una ridondanza
  – si semplificano alcune interrogazioni, ma
  – si appesantiscono gli aggiornamenti
  – si occupa maggior spazio
- Le possibili ridondanza riguardano
  – Attributi derivabili da altri attributi
  – Relazioni derivabili dalla composizione di altre relazioni (presenza di cicli)

1. **Analisi della ridondanza dell'attributo "costo totale" in prenotazione derivabile dal prodotto tra "prezzo per notte" in alloggio e il conteggio delle notti(tra data inizio e data fine) in prenotazione.**

operazione 1 memorizza il costo totale di una nuova prenotazione di un alloggio (1000/giorno)
operazione 2 stampa tutti i dati di una prenotazione relativa ad un alloggio(5/giorno)

TABELLA VOLUMI

Analisi dello schema con ridondanza:

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	S
Riserva	E	1	S
Alloggio	R	1	L
Alloggio	E	1	S

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	L
Riserva	R	20	L
Alloggio	E	20	L

Per un totale di 3000 accessi in scrittura e 1000 accessi in lettura al giorno nella tabella 1 e 205 accessi in lettura per la tabella 2 = 7205 accessi al giorno

Analisi schema senza ridondanza

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	S
Riserva	R	1	S

Concetto	Costrutto	Accessi	Tipo
Prenotazione	E	1	L
Riserva	R	20	L

Per un totale di 2000 accessi in scrittura al giorno per la tabella 1 e 105 accessi in lettura per la tabella 2 = 4105 accessi al giorno L'assenza di ridondanza permette di effettuare meno accessi.

1. **Analisi della ridondanza “numero recensioni” dell’attributo alloggio derivabile dal conteggio dell’associazione “contiene “**
    a. memorizza una nuova recensione relativa ad un alloggio (700/giorno)
    b. stampa tutti i dati di un alloggio compreso il numero delle sue recensioni (5/giorno)

TABELLA VOLUMI

Analisi dello schema con ridondanza:

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

Concetto	Costrutto	Accessi	Tipo
Alloggio	E	1	L

Per un totale di 2100 accessi in scrittura e 700 in lettura per la tabella 1 e 5 accessi in lettura nella tabella 2 = 4905 accessi al giorno

Analisi dello schema senza ridondanza:

Concetto	Costrutto	Accessi	Tipo
Recensione	E	1	S
Inviare	R	1	S

Concetto	Costrutto	Accessi	Tipo
Recensione	E	12	L
Inviare	R	12	L
Alloggio	E	1	L

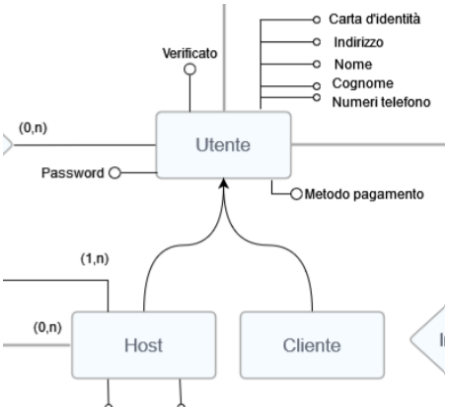
Per un totale di 1400 accessi in scrittura per la tabella 1 e 25 accessi in lettura nella tabella 2 = 1425 accessi al giorno L'assenza di ridondanza permette di effettuare meno accessi.

2.3.2. Eliminazione delle generalizzazioni

Obiettivo :
Sostituire le generalizzazioni nello schema ER con strutture alternative, in quanto non direttamente rappresentabili nel modello relazionale.
Tre possibilità:
1. accorpamento delle figlie della generalizzazione nel genitore
2. accorpamento del genitore della generalizzazione nelle figlie
3. sostituzione della generalizzazione con relazioni

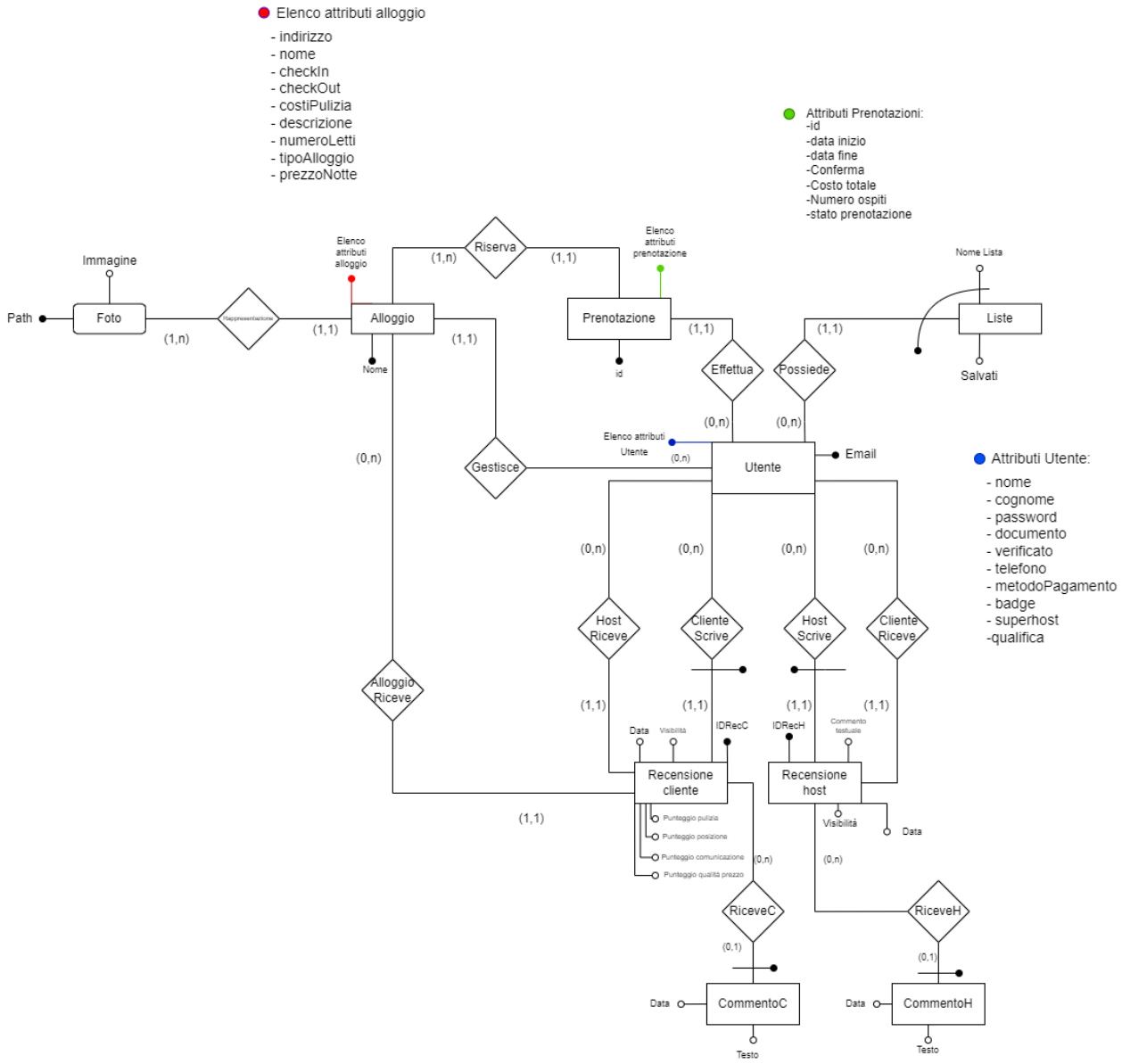
- Per la generalizzazione Utente entità genitore e Host e Cliente entità figlie abbiamo optato per la soluzione numero 1.
- Passi:
    - Eliminazione delle entità figlie.
    - Accorpamento nel padre di tutti gli attributi specifici delle entità figlie come attributi opzionali.
    - Accorpamento nel padre delle relazioni che coinvolgono le entità figlie assegnando cardinalità minima uguale a zero.

L'utente in questo caso si suddivide in host, superhost e cliente. Quindi è l'utente che può gestire 1 o più alloggi da host, può effettuare 0 o più prenotazioni da cliente o host, può possedere 0 o più 'liste preferiti' da cliente. In fine è dall'utente che partono tutte le recensioni.



- Un’altra generalizzazione che è stata rimossa riguarda le **recensioni**, il precedente schema ER aveva recensione come entità padre e recensione utente e alloggio come entità figlie mentre lo schema ER ristrutturato non esiste più l’entità padre Recensione.
  Abbiamo quindi adottato la Soluzione numero 2 e accorpato entità genitore della generalizzazione delle figlie, così da risparmiare sugli accessi. Sono rimaste le entità Recensione Host e Recensione Cliente.
  Recensione Host riguarda il caso in cui il cliente riceva una recensione dall’host e l’host che scrive la recensione stessa.
  Recensione Cliente invece riguarda il caso in cui l’host riceve la recensione cliente e il cliente che scrive la recensione stessa. Inoltre Recensione cliente è pure collegata ad alloggio, poiché un cliente può scrivere 0 o più recensioni ad un alloggio, con i seguenti attributi: La pulizia, il costo, la comunicazione, la posizione ecc…

2.4. Schema E-R ristrutturato + business rules



Vincoli e Regole Aziendali

Utente:

L’entità utente contiene “email” come chiave e altri attributi:
L’attributo verifica è un booleano che ritorna true se la carta d'identità è stata verificata, false altrimenti. Ha poi il nome, il cognome, il numero di telefono, le foto della carta d'identità ed un metodo di pagamento che deve selezionare obbligatoriamente. Può avere anche una lista degli alloggi preferiti.Un attributo qualifica che può essere Host o cliente o superhost.

- Un utente non deve effettuare una prenotazione su un alloggio di cui è proprietario.
- Un utente non può effettuare una recensione di una struttura se non vi ha alloggiato
- Un utente può usufruire dell’alloggio solo se la prenotazione è confermata
- Un host non può prenotare il suo stesso alloggio
- Un host non può recensire il proprio alloggio

Prenotazione:

Una prenotazione può essere eseguita da un ospite e possiede una chiave “id” (chiave sostituita a causa dell’eliminazione delle chiavi esterne nello schema precedente) e diversi attributi:
Data inizio e data fine che indicano i giorni della prenotazione, stato prenotazione che indica se la prenotazione è confermata o meno, un numero di ospiti e un costo totale della prenotazione.

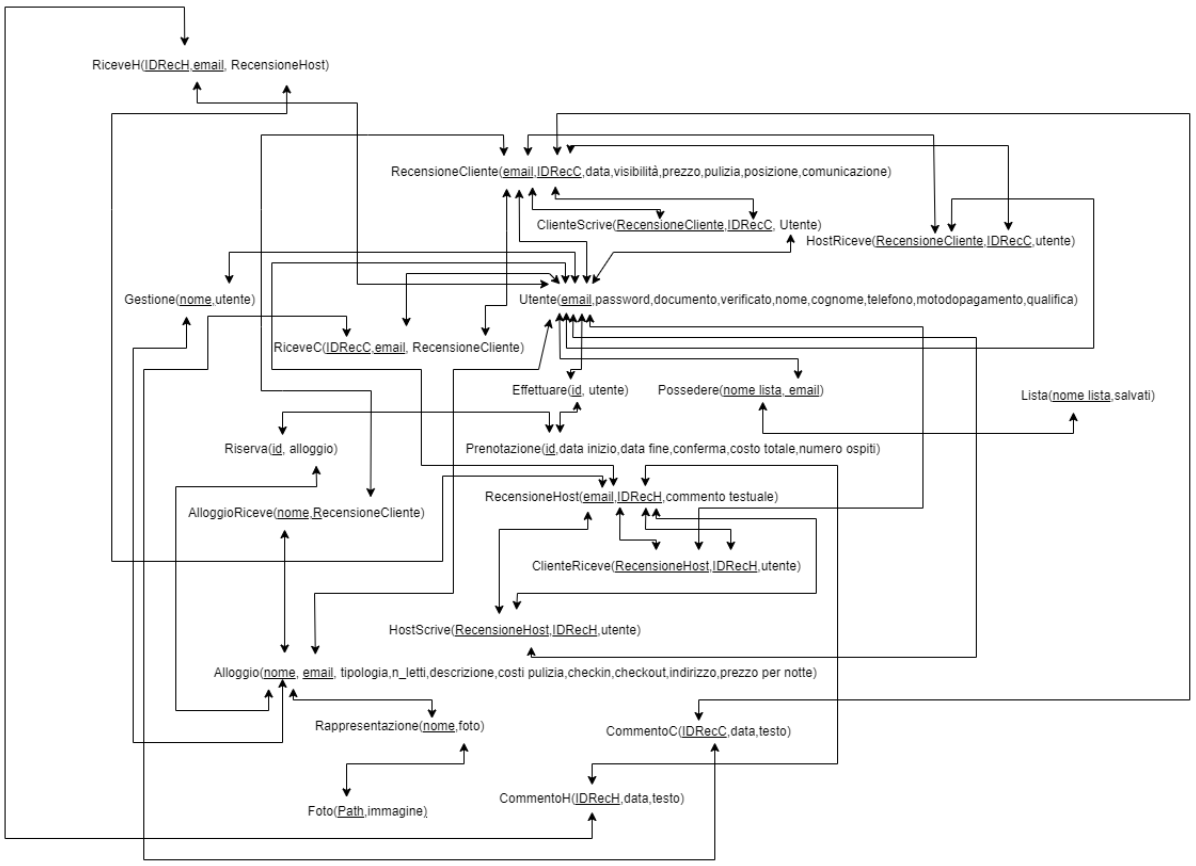
Alloggio:

Un alloggio può essere posseduto da un host o prenotato da un ospite, l’entità alloggio possiede come chiave un nome (NON SI POSSONO DARE NOMI UGUALI AD ALTRI ALLOGGI ALL’INTERNO DEL SITO) e diversi attributi:
Un orario per il check-in e uno per il check-out, una descrizione, un rating medio (da 1 a 5 stelle), la tipologia a cui appartiene (appartamento, soffitta, camera condivisa, casa condivisa, loft, ecc.…), delle foto, i costi di pulizia a fine soggiorno e il numero di posti letto.
Un alloggio non deve essere eliminato definitivamente dal database se sono state effettuate prenotazioni su di esso.

Commenti:

Un commento deve essere fatto ad una e una sola recensione e vengono identificati dalla Recensione Host/Cliente che a sua volta è identificata da un ID Recensione e dall’Host/Cliente che l’ha scritta. Host e ospiti possono commentarsi tra loro, rispondendo alle recensioni di uno e dell’altro creando una discussione.

2.5. Schema relazionale



UTENTE (email, qualifica, numero di telefono, password, metodo di pagamento, cognome, nome, verificato)

ALLOGGIO (nome ,email, orario check-in, orario check-out, descrizione, rating medio, numero recensioni, via, tipologia, foto, costi pulizia, posti letto)

PRENOTAZIONE(id, data inizio, data fine, stato prenotazione, numero ospiti,Conferma,costo totale)

LISTE(Nome lista, Salvati)

RECENSIONECLIENTE(email, IDRecC, data,visibilità,prezzo,pulizia,posizione,comunicazione)

RECENSIONEHOST(email, IDRecH, commento testuale)

FOTO(Path,immagine)

RAPPRESENTAZIONE(nome,foto)

HOSTSCRIVE(RecensioneHost, IDRecH, utente)

ALLOGGIORICEVE(nome,RecensioneCliente)

EFFETTUARE(id, utente)

POSSEDERE(nome lista, email, salvati)

GESTIONE(nome,utente)

CLIENTESCRIVE(RecensioneCliente, IDRecC, Utente)

HOSTRICEVE(RecensioneCliente, IDRecC, utente)

CLIENTERICEVE(RecensioneHost,IDRecH, utente)

RISERVA(id, alloggio)

POSSEDERE(nome lista, email, salvati)

COMMENTOC(RecensioneC, Data, Testo)

COMMENTOH(RecensioneH, Data, Testo)

RICEVEH(IDRecH, email, RecensioneHost)

RICEVEC(IDRecC, email, RecensioneCliente)

Vincoli di integrità referenziale:

- nomelista(possedere) referenzia nomelista(lista)
- utente(possedere) referenzia email(utente)

- id(effettuare) referenzia id(prenotazione)
- utente(effettuare)referenzia email(utente)
- nome-alloggio(rappresentazione) referenzia nome(alloggio)
- foto(rappresentazione) referenzia Path(foto)
- nome(gestione) referenzia nome(alloggio)
- utente(gestione) referenzia email(utente)
- email(RecensioneCliente) referenzia email(utente)
- email(RecensioneHost) referenzia email(utente)
- RecensioneCliente(clientescrive) referenzia email(RecensioneCliente)
- utente(clientescrive) referenzia email (utente)
- RecensioneHost(hostscrive) referenzia email(RecensioneHost)
- utente(hostscrive) refrenzia email(utente)
- RecensioneCliente(hostriceve) refrenzia email(RecensioneCliente)
- utente(hostriceve) referenzia email(utente)
- RecensioneHost(clientericeve) referenzia email (RecensioneHost)
- utente(clientericeve) referenzia email(utente)
- id prenotazione(riserva) referenzia id(prenotazione)
- alloggio(riserva) referenzia nome(alloggio)
- nome(alloggioriceve) referenzia nome(alloggio)
- RecensioneCliente(alloggioriceve) referenzia email (RecensioneCliente)
- RecensioneH(commentoH) referenzia IDRecH(RecensioneHost)
- RecensioneC(commentoC) referenzia IDRecC(RecensioneCliente)
- IDRecH(riceveH) referenzia RecensioneH(commentoH)
- utente(riceveH) referenzia email(utente)
- RecensioneHost(riceveH) referenzia email(RecensioneHost)
- IDRecC(riceveC) referenzia RecensioneC(commentoC)
- utente(riceveC) referenzia email(utente)

- RecensioneCliente(riceveC) referenzia email(RecensioneCliente)

3. Implementazione

3.1. DDL di creazione del database

```
CREATE TABLE utente(
email VARCHAR(100) NOT NULL,
password VARCHAR(100) NOT NULL,
qualifica VARCHAR(32) NOT NULL,
documento VARCHAR(100),
verificato BOOLEAN,
badge BOOLEAN,
nome VARCHAR(40) NOT NULL,
cognome VARCHAR(40) NOT NULL,
telefono NUMERIC(10,0) NOT NULL,
metodoPagamento VARCHAR(32),
PRIMARY KEY (email),
UNIQUE (email)
);

CREATE TABLE alloggio(
indirizzo VARCHAR(40) NOT NULL,
nome VARCHAR(50) NOT NULL,
checkIn TIME NOT NULL,
checkOut TIME NOT NULL,
costiPulizia FLOAT NOT NULL,
descrizione VARCHAR(255) NOT NULL,
numeroLetti TINYINT NOT NULL,
prezzoNotte INT NOT NULL,
tipoalloggio VARCHAR(20) NOT NULL,
email VARCHAR(100) NOT NULL,
PRIMARY KEY (nome) UNIQUE,
FOREIGN KEY (email) REFERENCES utente(email)
);
```

```sql
CREATE TABLE recensione_cliente(
idRecC INT NOT NULL SERIAL,
email VARCHAR(100) NOT NULL,
data DATE NOT NULL,
pulizia TINYINT NOT NULL,
comunicazione TINYINT NOT NULL,
posizione TINYINT NOT NULL,
qualitàPrezzo TINYINT NOT NULL,
recensioneHost VARCHAR(255),
visibilità BOOLEAN NOT NULL,
PRIMARY KEY (idRecC),
FOREIGN KEY (email) REFERENCES utente(email)
)

CREATE TABLE recensione_host(
idRecH INT NOT NULL SERIAL,
commento VARCHAR(256) NOT NULL,
email VARCHAR(100) NOT NULL,
visibilità BOOLEAN NOT NULL,
PRIMARY KEY (idRecH),
FOREIGN KEY (email) REFERENCES utente(email)
)

CREATE TABLE foto(
path VARCHAR(128) NOT NULL,
PRIMARY KEY(path),
nome VARCHAR(50) NOT NULL,
FOREIGN KEY (nome) REFERENCES alloggio(nome)
);

CREATE TABLE prenotazione(
id INT PRIMARY KEY,
dataInizio DATE NOT NULL,
dataFine DATE NOT NULL,
statoPrenotazione VARCHAR(100),
nOspiti INT NOT NULL,
conferma BOOLEAN,
email VARCHAR(100) NOT NULL,
nome VARCHAR(50) NOT NULL,
FOREIGN KEY (nome) REFERENCES alloggio(nome),
FOREIGN KEY (email) REFERENCES utente(email)
);
```

```
CREATE TABLE commentoC(
data DATE NOT NULL,
commento VARCHAR(256) NOT NULL,
idRecC INT NOT NULL,
FOREIGN KEY (idRecC) REFERENCES recensione_cliente(idRecC)
)

CREATE TABLE lista(
nomeLista VARCHAR(100) NOT NULL,
salvati VARCHAR(8192),
email VARCHAR(100) NOT NULL,
FOREIGN KEY (email) REFERENCES utente(email)
)

CREATE TABLE commentoH(
data DATE NOT NULL,
commento VARCHAR(256) NOT NULL,
idRecH INT NOT NULL,
FOREIGN KEY (idRecH) REFERENCES recensione_host(idRecH)
)
```

3.2. DML di popolamento di tutte le tabelle di database

Inserire un utente

INSERT INTO utente (email, password, qualifica ,nome, cognome, telefono)
VALUES ('utente@email', 'password', ‘host’ ,'nome', 'cognome', 3333333333)

Inserire un nuovo alloggi

INSERT INTO alloggio(email, indirizzo, nome, checkIn, checkOut, costiPulizia, descrizione, numeroLetti, tipoalloggio, prezzoNotte)
VALUES (‘host@email’, ‘Via prova 3’, ‘Casa vacanze prova’, 15:30, 10:30 , ‘50’, ‘testo’, 3, ‘Intero appartamento’, 120);

Inserire una nuova recensione cliente

INSERT INTO recensione_cliente(email, data, pulizia, comunicazione, posizione, qualitàprezzo, recensioneHost, visibilità)

VALUES (‘host@email’,07-07-2022, 3, 4, 2, 4, TRUE);

Inserire una nuova recensione host

INSERT INTO recensione_host(commento, email, visibilità)
VALUES (‘Marco è stato un ottimo cliente’, ‘utente@email’, TRUE);

Inserire una nuova prenotazione

INSERT INTO prenotazione(dataInizio, dataFine, nOspiti, email , nome)
VALUES (07-07-2022, 14-07-2022, 5, ‘utente@mail’, ‘Alloggio x’);

Inserire un nuovo commento ad una recensione

INSERT INTO commentoC(data, commento, idRecC)
VALUES (07-07-2022, ‘testo commento….’, 4);

Inserire lista

INSERT INTO lista(nomeLista, salvati, email)
VALUES (‘Preferiti di Torino’, ‘Appartamento 1, Appartamento 2’, ‘utente@mail’);

3.3. Operazioni di cancellazione e modifica

DELETE FROM utente
WHERE email = ‘utenteDaEliminare@mail’;

DELETE FROM alloggio
WHERE nome = ‘Alloggio da eliminare’ ;

UPDATE utente
SET password = ‘nuovaPassword’
WHERE email = ‘utente@mail’ ;

UPDATE prenotazione
SET conferma = TRUE
WHERE id = 4;

UPDATE alloggio
SET prezzoNotte = 150
WHERE nome = ‘Alloggio da modificare’ ;