**DATI**

Stefan Alexandru Andries – 20018146 – [20018146@studenti.uniupo.it](mailto:20018146@studenti.uniupo.it)

Nicolò Benella – 20018152 – [20018152@studenti.uniupo.it](mailto:20018152@studenti.uniupo.it)

Florin Razvan Donie – 20018147 – 20018147@studenti.uniupo.it

**PROGETTAZIONE CONCETTUALE**

**1.1. Requisiti iniziali**

Si deve progettare una base di dati relativa a un sito di affitti di appartamenti. In particolare si possono inserire, cercare, prenotare e recensire interi appartamenti o stanze in appartamenti condivisi. Per potere usufruire dei servizi gli utenti devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, email.

Per prenotare una struttura i clienti devono selezionare un tipo di sistemazione e un intervallo di date.

Quando un cliente, dopo aver visionato qualche struttura, decide di prenotare, scrive al proprietario che risponde con un numero di prenotazione attraverso cui l’utente può fare modifiche alle sue scelte. La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno.

I clienti possono rilasciare un commento su una struttura solo se ne hanno realmente usufruito *(Implica un record degli hotel visitati)*.Gli utenti possono cambiare i dati delle strutture inserite di cui sono proprietari e rispondere sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio. Si tiene traccia nella base di dati sia delle risposte pubbliche (che vengono pubblicate sotto la recensione) sia di quelle private.

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

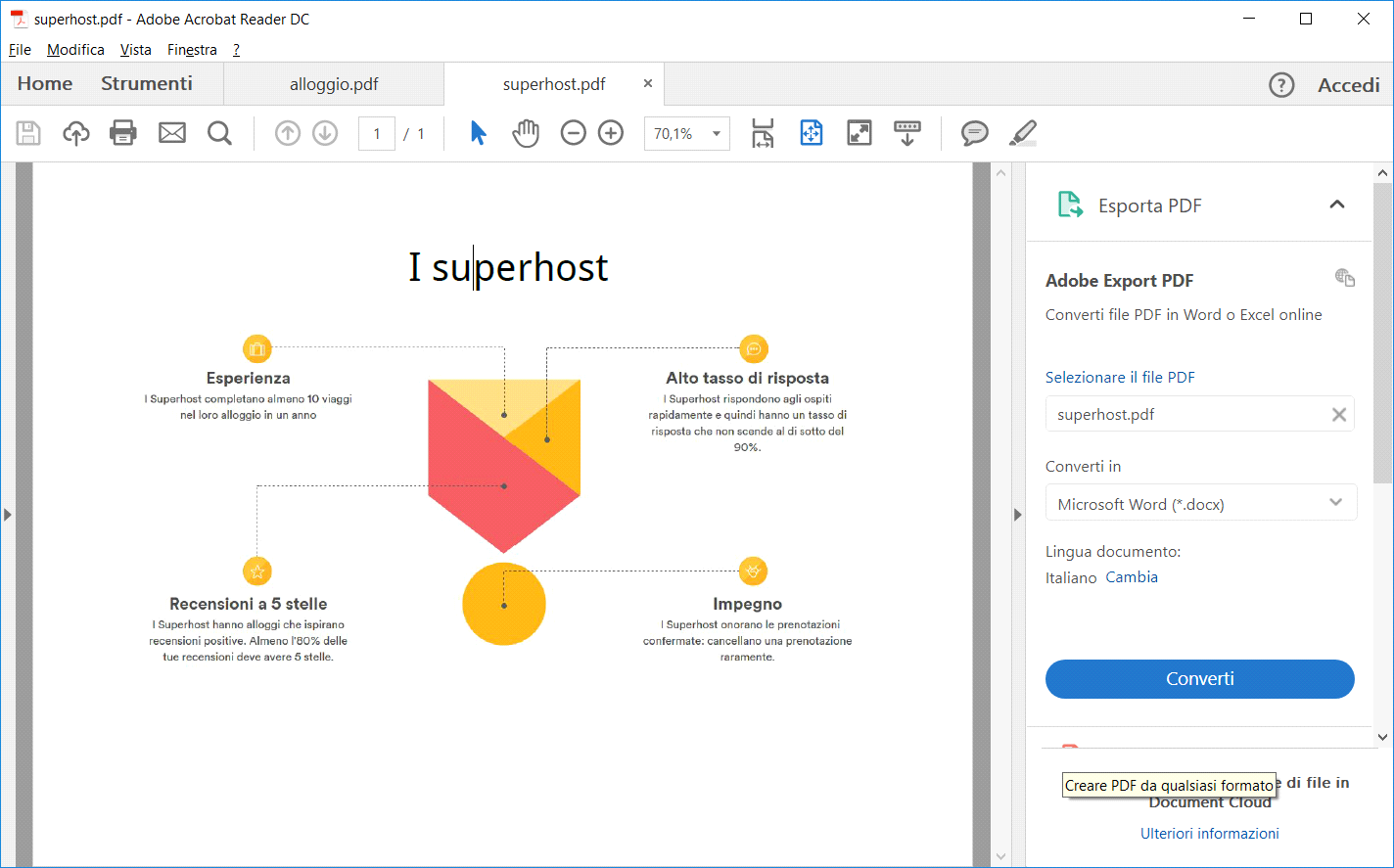
Alcuni proprietari possono essere classificati come superhost (si veda il file superhost.pdf).

Degli utenti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come host che come ospite, le eventuali proprietà e (se guadagnato) il simbolo di superhost.

Quando un cliente si cancella dal servizio le sue recensioni e le risposte ricevute relativamente ad esse rimangono memorizzate mentre i suoi dati personali vengono cancellati.

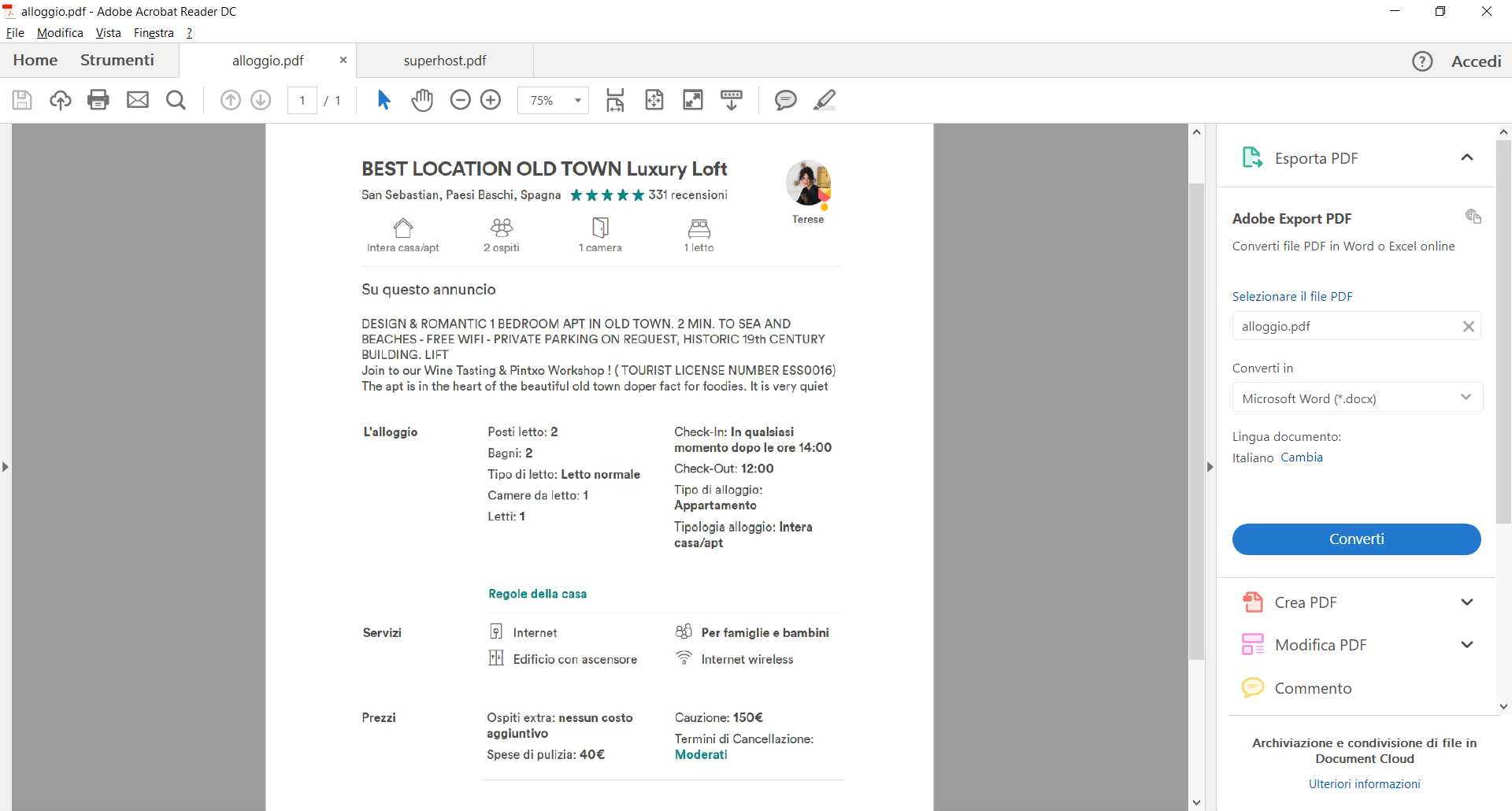
Le informazioni relative alle strutture sono reperibili nel documento alloggio.pdf.

**superhost.pdf**



*Gli utenti superhost per diventari tali devono aver completato almeno 10 viaggi in un anno in un alloggio di loro proprietà, devono avere almeno l'80% delle recensioni a 5 stelle, hanno un tasso di risposta agli ospiti del 90% e infine cancellano raramente le prenotazioni degli ospiti.*

**alloggio.pdf**



*Nella pagina di un determinato alloggio, l'utente deve poter vedere innazitutto nome del proprietario e relativo stato superhost se presente, in seguito il nome dell'alloggio, la location, il numero di recensioni con relativa media delle stelle, la descrizione dell'alloggio fatta dal proprietario, le informazioni principali dell'alloggio quali n° di camere, bagni, letti, e orari del check-in e check-out e infine i servizi offerti quali presenza di internet, permesso di poter portare animali, ecc.. e i prezzi dell'alloggio e dei servizi extra.*

**1.2. Glossario dei termini**

|  |  |  |  |
| --- | --- | --- | --- |
| **Termine** | **Descrizione** | **Sinonimi** | **Collegamenti** |
| Alloggio | Struttura che un cliente può prenotare, messa a disposizione dal proprietario. Può essere un intero appartamento o solo una stanza. | Appartamenti, Proprietà | Utente, Commento |
| Recensioni | Recensione scritta da un utente che ha usufruito dell’alloggio. |  | Utente, Commento, Alloggio |
| Commento | Commento lasciato da un utente. Può essere pubblico o privato. | Risposte | Utente, Alloggio |
| Utente | Può essere un cliente che affitta alloggi o un proprietario che li mette a disposizione. I proprietari possono anche essere superhost. | Ospiti, host | Commento, Alloggio, Recensioni |

**1.3. Requisiti riscritti**

Si deve progettare una base di dati relativa a un sito di affitti di ~~appartamenti~~ alloggi. In particolare si possono inserire, cercare, prenotare e recensire alloggi che possono essere interi appartamenti o stanze ~~in appartamenti condivisi~~.

Per potere usufruire dei servizi gli utenti devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, e-mail e possono essere clienti e proprietari.

~~Per prenotare una struttura i clienti devono selezionare un tipo di sistemazione e un intervallo di date.~~ Per prenotare, un cliente seleziona un tipo di alloggio e un intervallo di date.

Quando un cliente~~, dopo aver visionato qualche struttura~~, decide di prenotare, ~~scrive al proprietario che risponde con un numero di prenotazione attraverso cui l’utente può fare modifiche alle sue scelte.~~ contatta il proprietario che gli fornisce un numero di prenotazione. La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno.

I clienti possono rilasciare ~~un commento~~ una recensione su una struttura ~~solo se ne hanno realmente usufruito~~ usufruita. I ~~utenti~~ proprietari possono cambiare i dati delle proprie strutture inserite ~~di cui sono proprietari~~ e ~~rispondere~~ commentare sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio.

Si tiene traccia nella base di dati ~~sia delle risposte pubbliche (che vengono pubblicate sotto la recensione) sia di quelle private~~ dei commenti pubblici e privati.

~~Anche~~ I proprietari ~~degli appartamenti~~ possono recensire i clienti ~~ospiti~~ dopo il soggiorno.

Alcuni proprietari possono essere classificati come superhost ~~(si veda il file superhost.pdf).~~

Degli utenti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come ~~host~~ proprietario che come cliente ~~ospite~~, ~~le~~ gli eventuali alloggi ~~proprietà~~ e (se guadagnato) il simbolo di superhost.

Quando un ~~cliente~~ utente si cancella dal servizio le sue recensioni ~~e le risposte ricevute relativamente ad esse~~ e relativi commenti rimangono memorizzate mentre i suoi dati personali vengono cancellati.

**1.4. Requisiti strutturati in gruppi di frasi omogenee**

|  |
| --- |
| Frasi di carattere generale Si deve progettare una base di dati relativa a un sito di affitti di alloggi.  La gestione del pagamento tramite carta credito o altro è demandata a un ente esterno. |

|  |
| --- |
| Frasi relative a Alloggi Si possono inserire, cercare, prenotare e recensire alloggi che possono essere interi appartamenti o stanze. |

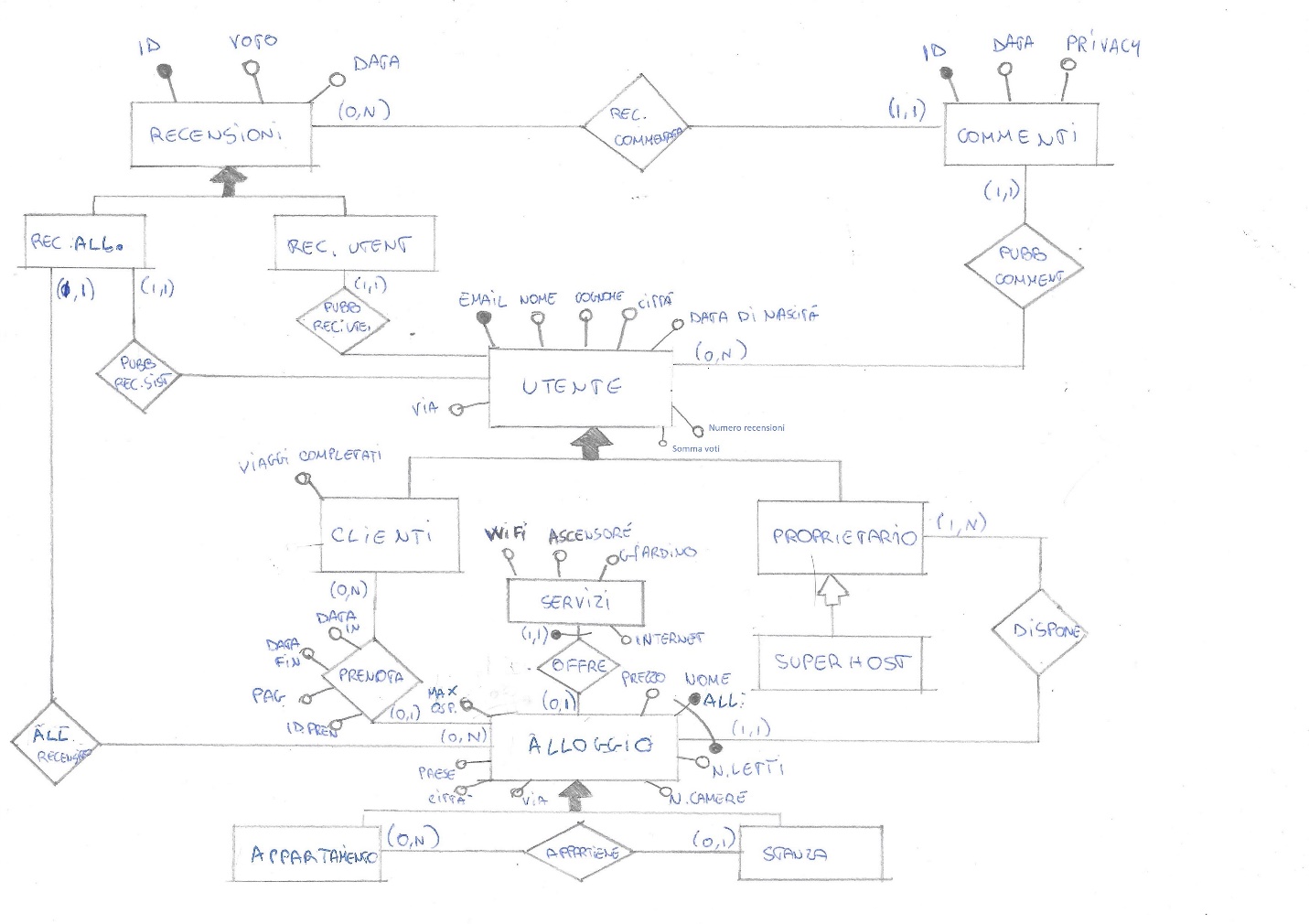
|  |
| --- |
| Frasi relative a Utenti Per potere usufruire dei servizi gli utenti devono registrarsi fornendo nome, cognome, indirizzo, data di nascita, e-mail e possono essere clienti e proprietari.  Degli utenti vengono mostrati il nome, una foto del profilo, le lingue parlate, le recensioni ricevute sia come proprietario che come cliente, gli eventuali alloggi e (se guadagnato) il simbolo di superhost.  Quando un utente si cancella dal servizio le sue recensioni e relativi commenti rimangono memorizzate mentre i suoi dati personali vengono cancellati. |

|  |
| --- |
| Frasi relative a Recensioni I clienti possono rilasciare una recensione su una struttura usufruita. |

|  |
| --- |
| Frasi relative a Commenti  I proprietari possono cambiare i dati delle proprie strutture inserite e commentare sia privatamente che pubblicamente alle recensioni pubblicate da chi ha usufruito del servizio.  Si tiene traccia nella base di dati dei commenti pubblici e privati. |

|  |
| --- |
| Frasi relative a tipi specifici di utenti Quando un cliente, decide di prenotare, contatta il proprietario che gli fornisce un numero di prenotazione.  I clienti possono rilasciare una recensione su una struttura usufruita. I proprietari possono cambiare i dati delle proprie strutture inserite e rispondere sia privatamente che pubblicamente alle recensioni pubblicate.  I proprietari possono recensire i clienti dopo il soggiorno.  Alcuni proprietari possono essere classificati come superhost. |

**1.5. Schema E-R + regole aziendali**



Essendo questo lo schema iniziale, sono ancora presenti tutti i problemi che derivano da uno schema ER non ristrutturato, quali per esempio ridondanze e generalizzazioni.

La strategia di progetto usata è stata quella della tecnica inside-out. Pur essendo in gruppo siamo partiti da una prima entità con i suoi attributi e da essa ci siamo estesi ad altre entità applicando le associazioni tra esse fino a rappresentare l’intero schema.

**Regole aziendali**

**Vincoli di Integrità:**

|  |  |
| --- | --- |
| RV1 | Un utente deve avere il codice prenotazione per modificare le sue scelte. |
| RV2 | Un utente deve aver usufruito di un alloggio per recensirlo. |
| RV3 | Un proprietario deve aver offerto un alloggio a un utente per poter essere recensito da tale. |
| RV4 | Un proprietario può recensire un utente solo dopo la fine del suo periodo di alloggio. |
| RV5 | Un proprietario può modificare solamente un suo alloggio. |

# 2.1. Tavola dei volumi

|  |  |  |
| --- | --- | --- |
| **Concetto** | **Tipo** | **Volume** |
| Utente | E | 1000 |
| Alloggio | E | 500 |
| Recensione | E | 1200 |
| Prenotazione | A | 600 |
| Dispone alloggio | A | 500 |
| Pubblica recensione | A | 1200 |
| Commenta recensione | A | 300 |
| Numero recensioni per utente | E | 5 |

Avendo messo come volume 1000 a l’entità Utente abbiamo dovuto ovviamente assegnare agli alloggi un volume minore, mentre invece le recensioni essendo un unione tra recensione utente e recensione alloggio abbiamo potuto assegnare loro un valore maggiore.

Essendo che non tutti gli Utenti prenotano anche l’associazione prenotazione ha un volume inferiore, stesso vale per dispone alloggio.

Commenta recensione è stato preso in maniera casuale.

# 2.2. Tavola delle operazioni

| Operazione | Descrizione | Tipo | Frequenza |
| --- | --- | --- | --- |
| Op1 | Creazione di un utente | I | 70 al giorno |
| Op2 | Prenotazione struttura. | I | 50 al giorno |
| Op3 | Creazione di annuncio per una struttura. | I | 20 al giorno |
| Op4 | Trova le strutture tramite determinati criteri, tipo di struttura, paese, prezzo ecc.. | I | 1000 al giorno |
| Op5 | Inserire recensione per un appartamento | I | 10 a settimana |
| Op6 | Inserire recensione per un utente | I | 5 volte al giorno |
| Op7 | Rispondere a una recensione con un commento pubblico o privato. | I | 4 a settimana |
| Op8 | Aggiornamento dati delle strutture di cui si è proprietari. | I | 10 al mese |
| Op9 | Cancellamento di un utente | I | 2 al giorno |
| Op10 | Stampo email degli utenti proprietari | B | 10 al giorno |
| Op11 | Trova recensioni di un alloggio | I | 500 al giorno |
| Op12 | Stampo media recensioni di un utente | I | 10 al giorno |
| Op13 | Stampo numero di viaggi completati | B | 10 al giorno |
| Op14 |  |  |  |
| Op15 | Stampo nome del proprietario di un alloggio | I | 20 al giorno |

# 2.3. Ristrutturazione dello schema E-R

## 2.3.1. Analisi delle ridondanze

### 2.3.1.1. RIDONDANZA 1 (somma voti recensioni, numero recensioni)

Utilizzando gli attributi somma voti e numero recensioni posso calcolare velocemente la media dei voti, mentre togliendoli dovrei andare ogni volta ad eseguire quelle operazioni.

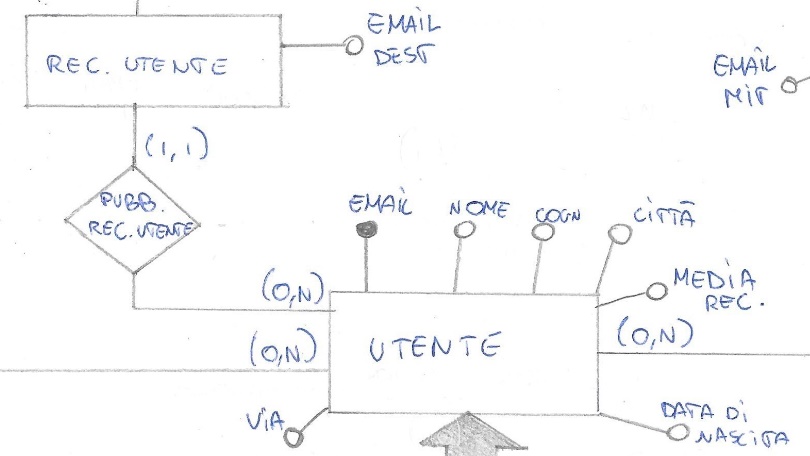
**OPERAZIONI COINVOLTE**

* Op6 - Inserire recensione per un utente
* Op12 - Stampo media recensioni di un utente

**PRESENZA DI RIDONDANZA**

**Op6**

Tavola degli accessi:

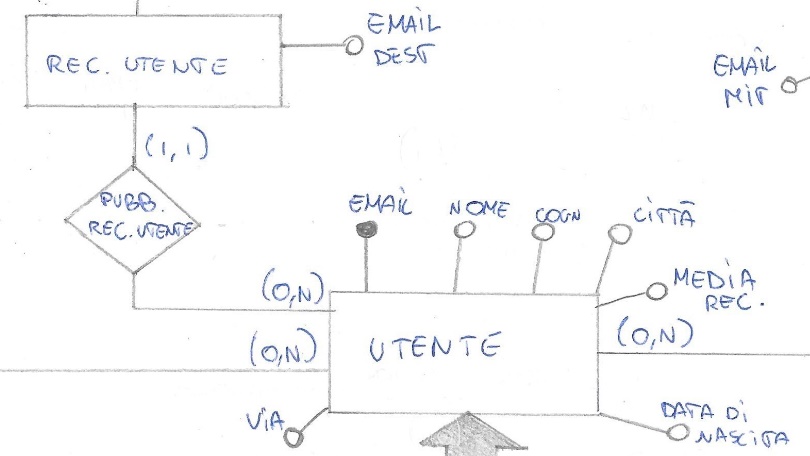


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Rec.Utente | Entità | 1 | S | Memorizzo la recensione |
| Pubb.Rec.Utente | Assoc. | 1 | S | Cerco l’associazione rec.utente - utente |
| Utente | Entità | 1 | L | Cerco la persona destinatario della rec. |
| Utente | Entità | 1 | S | Aggiorno somma voti e numero rec |
|  |  |  |  |  |

Costo: S: 2(\*4) \* 5 = 40 L: 1\*5 = 5 TOT: 45

**Op12**

Tavola degli accessi:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Utente | Entità | 1 | L | Calcolo la media usando i due attributi |
|  |  |  |  |  |

Costo: S: 0 L: 1\*10 TOT: 10

**ASSENZA DI RIDONDANZA**

**Op6**

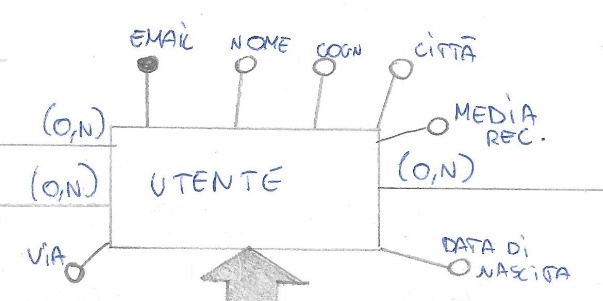
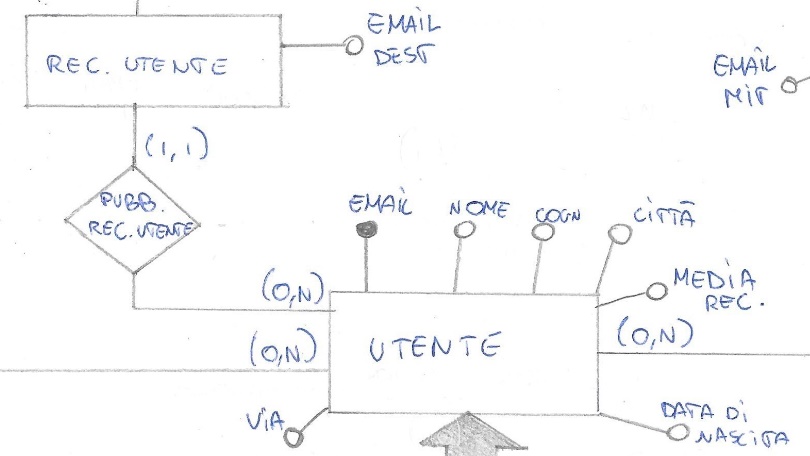


Tavola degli accessi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Rec.Utente | Entità | 1 | S | Memorizzo la recensione |
| Utente | Entità | 1 | S | Memorizzo l’associazione con utente |

Costo: S:1(\*2) \*5 = 10 L: 0 TOT: 10

**Op12**

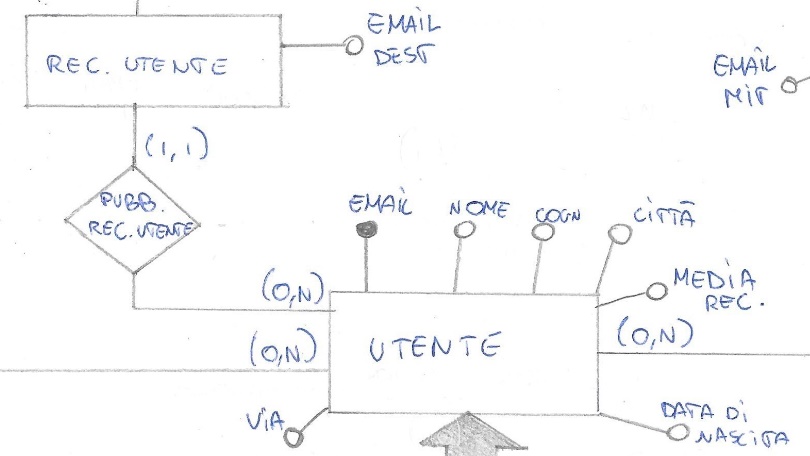


Tavola degli accessi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Utente | Entità | 1 | L | Cerco l’utente |
| Pub.Rec.Utente | Assoc | 1 | L | Cerco le associazioni con rec. utente |
| Rec.Utente | Entità | 3 | L | Conto il numero di recensioni |
| Rec.Utente | Entità | 1 | L | Conto la somma dei voti e stampo la media |

Costo: S: 0 L: 4\*10 TOT: 40

**TOTALI PER RIDONDANZA 1**

**Presenza di ridondanza**

**Spazio:** 2\*(4\*1200) = 9600

**Tempo:** 55

**Assenza di ridondanza**

**Spazio:** 0

**Tempo:** 50

**Decisione:** Rimozione dell’attributo in quanto non occupo spazio per mantenere i due attributi anche se ci vuole più tempo per calcolare la media.

### 2.3.1.2. RIDONDANZA 2 (numero viaggi completati)

Utilizzando l’attributo numero viaggi completati non ho bisogno di andare ogni volta a calcolarlo con l’associazione prenotazione.

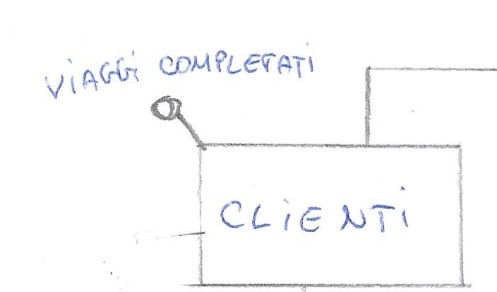
**OPERAZIONI COINVOLTE**

* Op13 – Stampo numero viaggi completati
* Op14 -

**PRESENZA DI RIDONDANZA**

**Op13**

Tavola degli accessi:

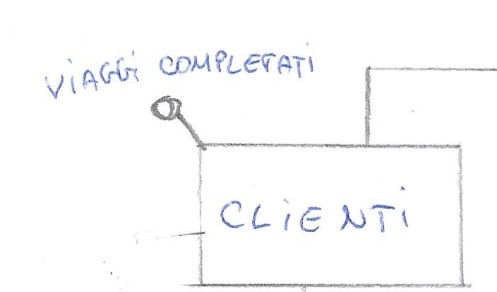


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Clienti | Entità | 1 | L | Stampo numero viaggi completati |

Costo:0 L: 1\*10 = 10 TOT: 10

**Op14**

Tavola degli accessi:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
|  |  |  |  | . |
|  |  |  |  |  |

Costo: S: L: TOT:

**ASSENZA DI RIDONDANZA**

**Op13**

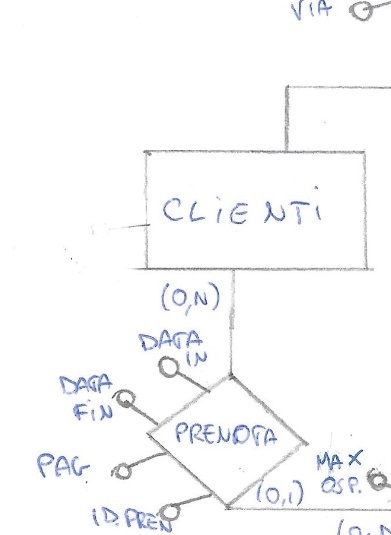


Tavola degli accessi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
| Utente | Entità | 1 | L | Seleziono l’utente |
| Prenota | Associazione | 1 | L | Seleziono le prenotazioni con data fin passata. |
| Prenota | Associazione | 1 | L | Calcolo il numero di viaggi completati |

Costo: S: 0 L: 3 \* 10 TOT: 30

**Op14**

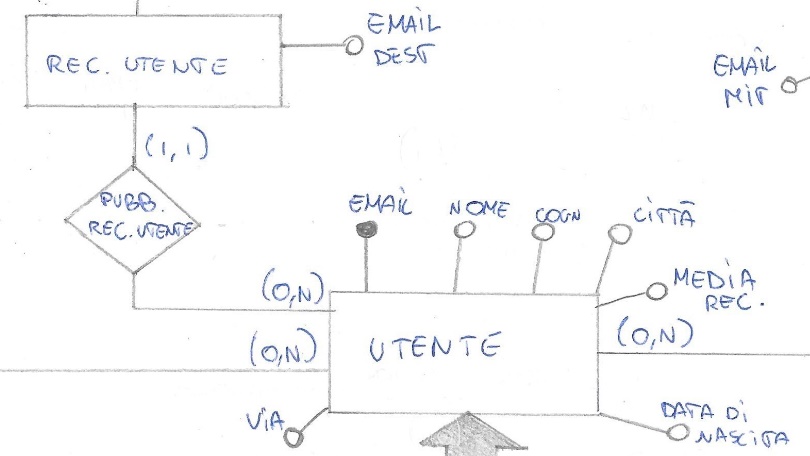
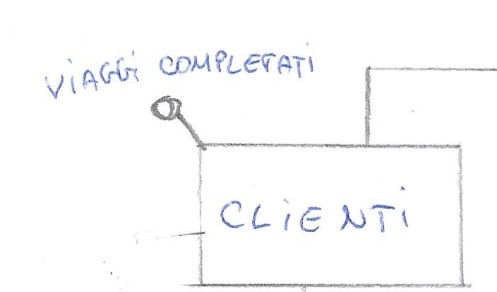


Tavola degli accessi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concetto | Costrutto | Accessi | Tipo | Descrizione |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Costo: S: L: TOT:

**TOTALI PER RIDONDANZA 2**

**Presenza di ridondanza**

**Spazio:**

**Tempo:**

**Assenza di ridondanza**

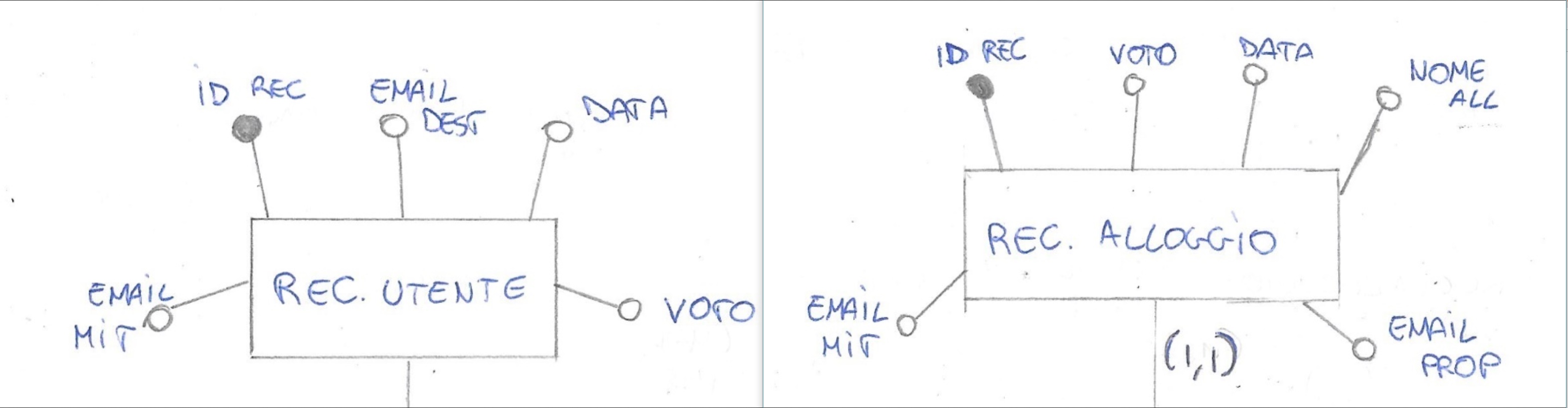
**Spazio:** 0

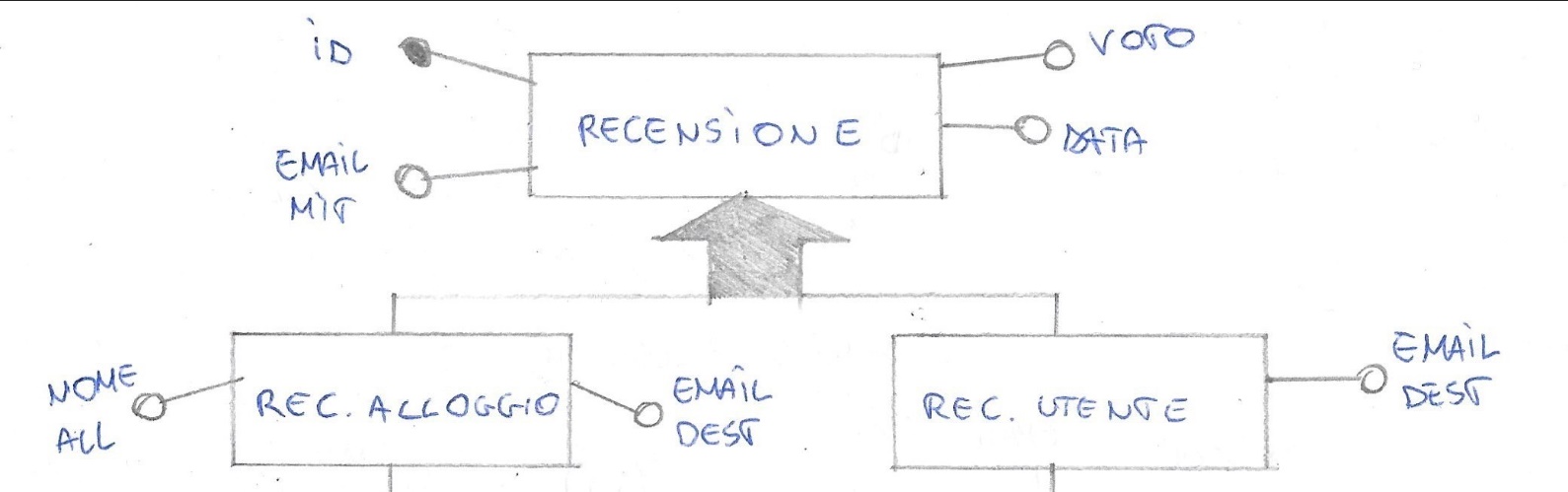
**Tempo:**

**Decisione:** Rimozione dell’attributo

## 2.3.2. Eliminazione delle generalizzazioni

### 2.3.2.1. Generalizzazione 1 <Recensioni>

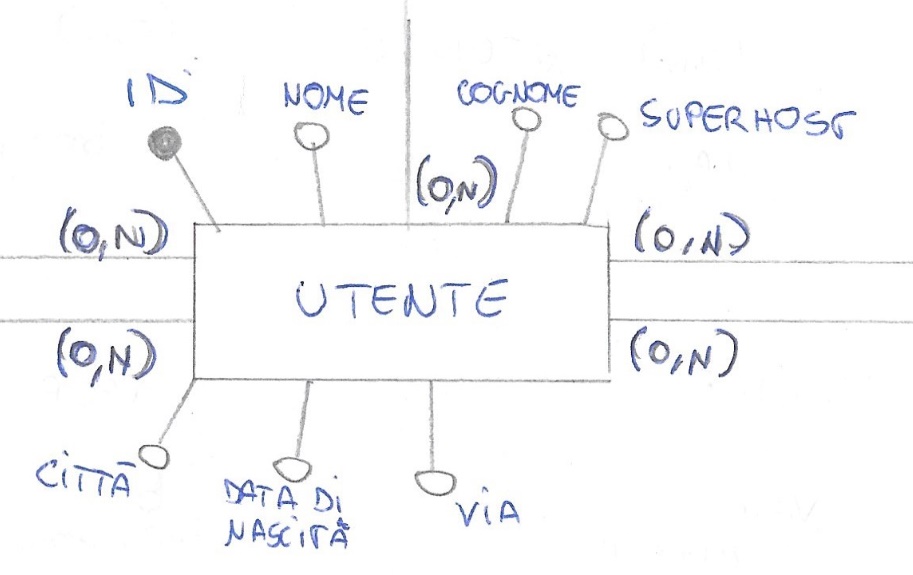
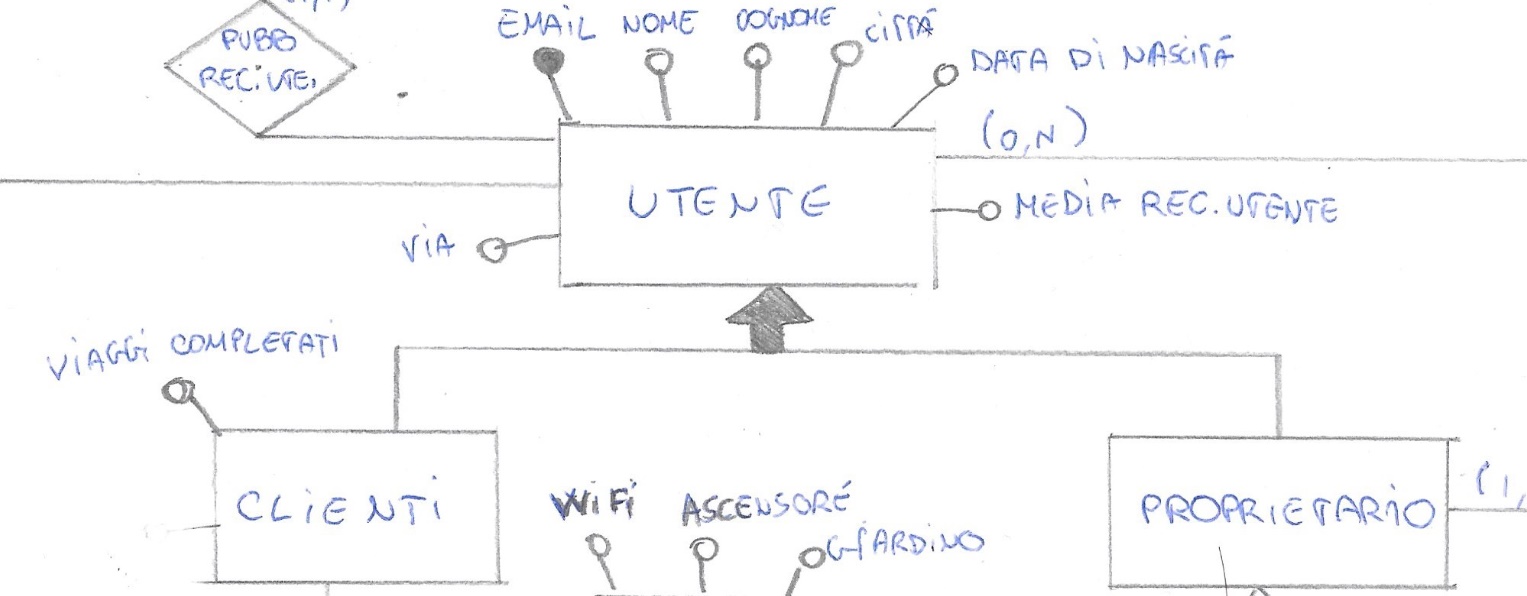


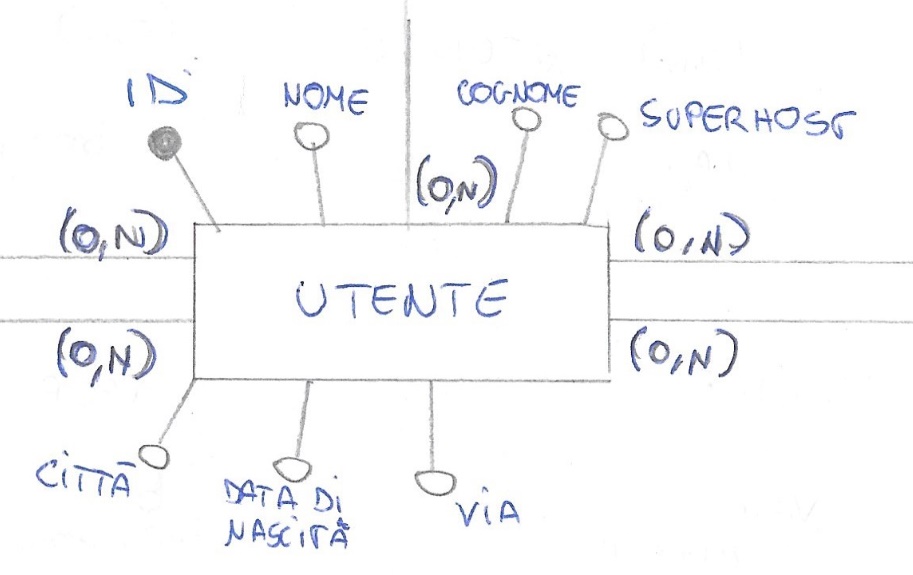


PORZIONE DI SCHEMA PRIMA DELL’ELIMINAZIONE

Rimozione della generalizzazione e manteniamo solamente l’entità utente anziche entrambi i figli. Tutti gli attributi dei figli sono stati incorporati in utente.

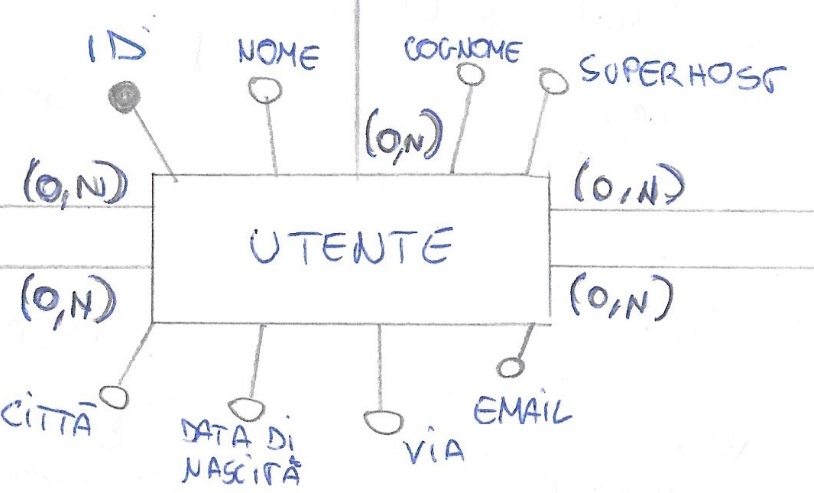
### 2.3.2.2. Generalizzazione 2 <Utente>

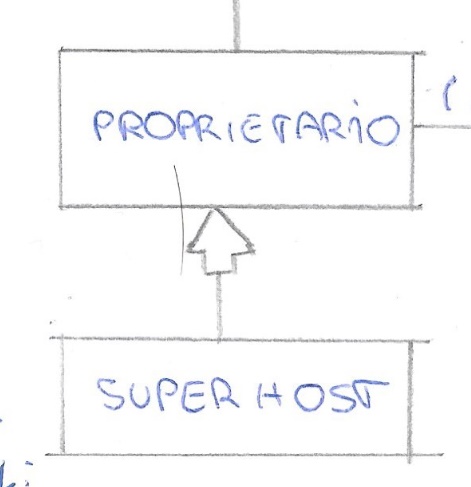




Rimozione della generalizzazione e manteniamo solamente l’entità utente anziche entrambi i figli. Tutti gli attributi dei figli sono stati incorporati in utente.

### 2.3.2.3. Generalizzazione 2 <Superhost>

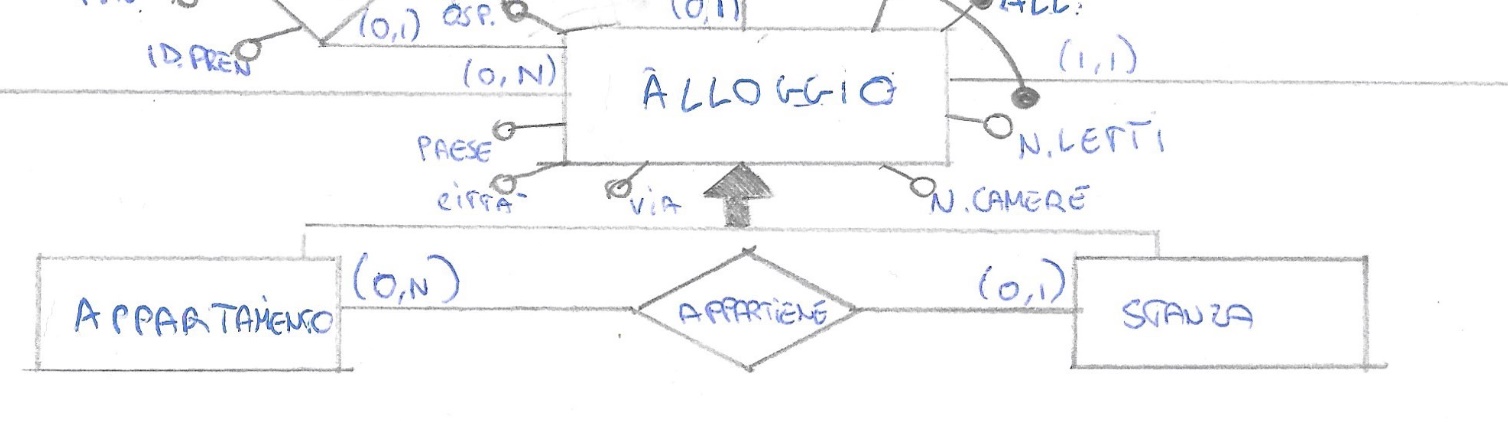




L’entità superhost è stato eliminata ed è ora diventata un attributo di utente.

### 2.3.2.3. Generalizzazione 2 <Alloggio>





Le entità stanza e appartamento sono state eliminate mantenendo soltanto l’entità padre alloggio, e l’attributo tipo che ne specifica appunto il tipo.

## 2.3.4.scelta degli identificatori principali

|  |  |
| --- | --- |
| **Entità** | **Identificatore principale** |
| Utente | ID |
| Alloggio | ID |
| Recensione utente | ID Recensione |
| Recensione alloggio | ID Recensione |
| Servizi | ID |
| Commento | ID |

Abbiamo optato per un id per tutte le entità.

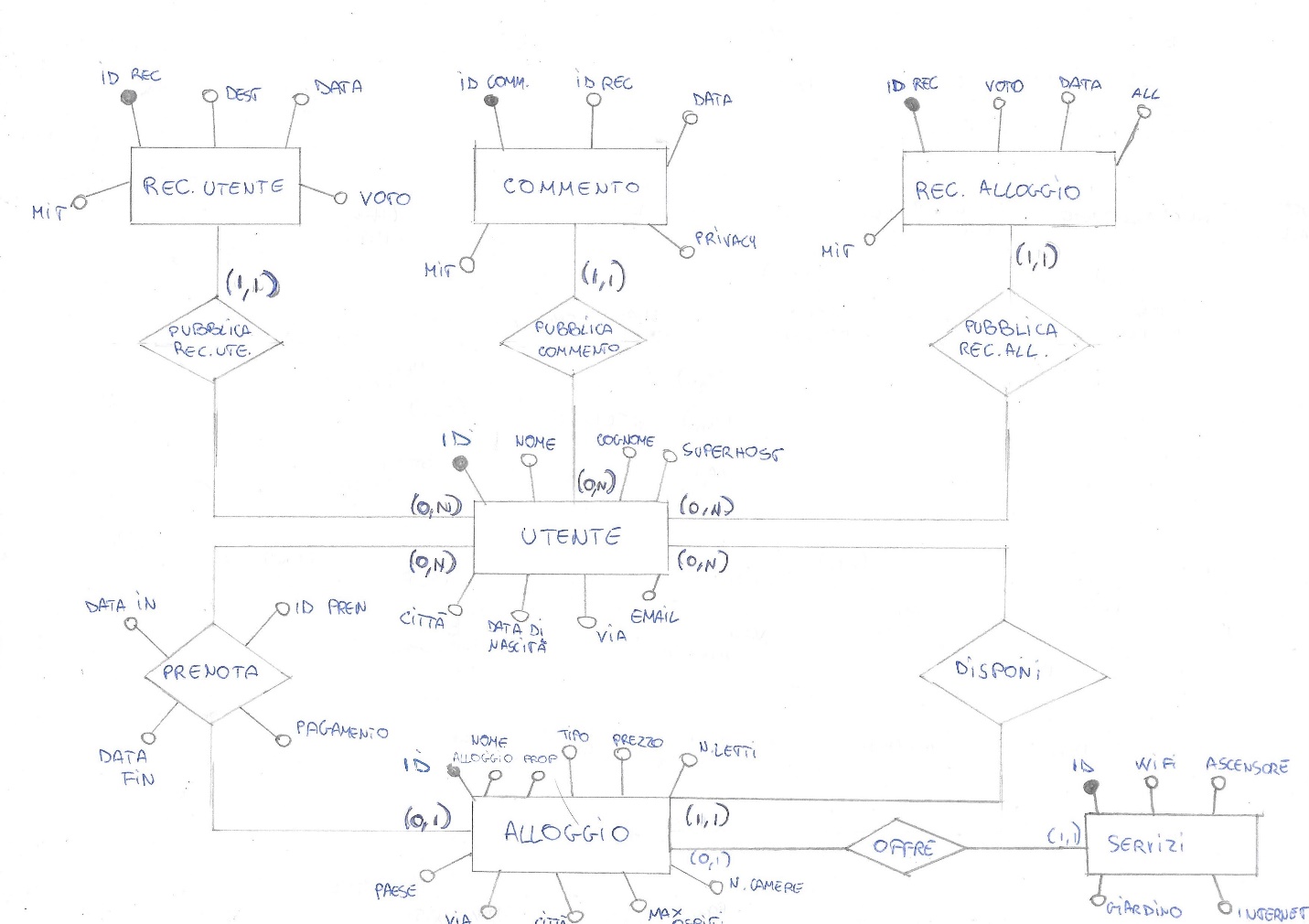
Per le recensioni abbiamo scelto di optare per un id surrogato in quanto essendo necessario mantenere le recensioni quando un utente si elimina dal sito, non ci è stato possibile usare come chiave l’email di quest’ultimo insieme ad altri attributi, in quanto alla sua eliminazione l’email non sarebbe più stata reperibile. Con l’ID surrogato invece la recensione sarà per sempre identificata e solamente l’attributo riguardante l’email del utente ( che sia destinatario o mittente ) verrà settata a vuoto.

Stesso motivo per i commenti.

Per gli alloggi abbiamo usato un id in quanto utilizzare sia il nome che l’email proprietario poteva diventare allungare di molto le istanze.

I servizi invece usare un id è l’unico modo che abbiamo trovato per identificarlo.

# 2.4 Schema E-R ristrutturato + regole aziendali



## Regole aziendali

**Vincoli di Integrità:**

|  |  |
| --- | --- |
| RV1 | Un utente deve avere il codice prenotazione per modificare le sue scelte. |
| RV2 | Un utente deve aver usufruito di un alloggio per recensirlo. |
| RV3 | Un proprietario deve aver offerto un alloggio a un utente per poter essere recensito da tale. |
| RV4 | Un proprietario può recensire un utente solo dopo la fine del suo periodo di alloggio. |
| RV5 | Un proprietario può modificare solamente un suo alloggio. |

# 2.5 Schema relazionale con vincoli di integrità referenziale

<Utente (ID, Email, Nome, Cognome, Data nascità, Città, Via, Superhost)  
<Servizi (ID, Wifi, Ascensore, Giardino, Internet)>

<Alloggio (ID, Nome, Proprietario, Servizi, Tipo, Prezzo, N.Letti, N.Camere, Max ospiti, Città, Via, Paese)>

<Alloggio(Proprietario) referenzia Utente(ID)>

<Alloggio(Servizi) referenzia Servizi(ID)>

<Recensione alloggio (ID, Mittente, Alloggio, Voto, Data)>

<Recensione alloggio(Mittente) referenzia Utente(ID)>

<Recensione alloggio(Alloggio) referenzia Alloggio(ID)>

<Recensione utente (ID, Mittente, Destinatario, Voto, Data)>

<Recensione utente(Mittente) referenzia Utente(ID)>

<Recensione utente(Destinatario) referenzia Utente(ID)>

<Commento (ID, Mittente, ID recensione, Data e ora, Privacy)>

<Commento(Mittente) referenzia Utente(ID)>

<Commento(ID recensione) referenzia Recensione alloggio(ID)>

<Prenotazione alloggio (Utente, Alloggio, ID prenotazione, Data Inizio, Data fine, Tipo pagamento)>..

< Prenotazione alloggio (Utente) referenzia Utente(ID)>

< Prenotazione alloggio (Alloggio) referenzia Alloggio(ID)>

<Pubblicazione annunci (Proprietario, Alloggio)>

< Pubblicazione annunci (Proprietario) referenzia Utente(ID)>

< Pubblicazione annunci (Alloggio) referenzia Alloggio(ID)>

<Pubblicazione commenti (Mittente, ID recensione, Data e ora)>

< Pubblicazione commenti (Mittente) referenzia Utente(ID)>

< Pubblicazione commenti (ID recensione) referenzia Recensione alloggio(ID)>

< Pubblicazione commenti (Data e ora) referenzia Commento(Data e ora)>

<Pubblicazione recensione alloggio (Mittente, ID recensione)>

< Pubblicazione recensione alloggio (Mittente) referenzia Utente(ID)>

< Pubblicazione recensione alloggio (ID recensione) referenzia Recensione alloggio(ID)>

< Pubblicazione recensione utente(Mittente, ID recensione)>

< Pubblicazione recensione utente (Mittente) referenzia Utente(ID)>

< Pubblicazione recensione utente (ID recensione) referenzia Recensione utente(ID)>

# 3 DDL di creazione del database

Le tabelle sono abbastanza standard, ognuna ha la propria primary key ed escludendo Utente hanno tutte foreign key che fanno riferimento ad altre tabelle.

C’è stato bisogno di creare un dominio adatto per l’attributo voto, in quanto esso può andare solamente da 1 a 5.

Dovendo mantenere le recensioni e i commenti, nelle tabelle dopo aver fatto il vincolo referenziale con i dati dell’utente abbiamo impostato gli on delete con reazione set null, in modo tale da mantenere la recensione se l’utente si elimina, perdendo solamente il nome di quest’ultimo e eventuale alloggio se era un proprietario.

# 4 DML di popolamento di tutte le tabelle del database

Abbiamo inserito i dati testando che le varie default null, not null, e altre opzioni funzionassero. Abbiamo aggiunto anche in tutte le tabelle un entry extra, per testare il funzionamento corretto dell’eliminazione.

Abbiamo testato che nel caso di valori null, le tabelle mostrassero comunque correttamente il dato.

# 5 DML di modifica

Abbiamo aggiunto un paio di operazioni basilari per il database e che ne dimostrano anche il corretto funzionamento , come per esempio nel caso dell’eliminazione di un utente.

Corrispondono tutte a le operazioni che abbiamo in precedenza nella tavola delle operazioni, quali per esempio l’op1, op3, op4, op11.