

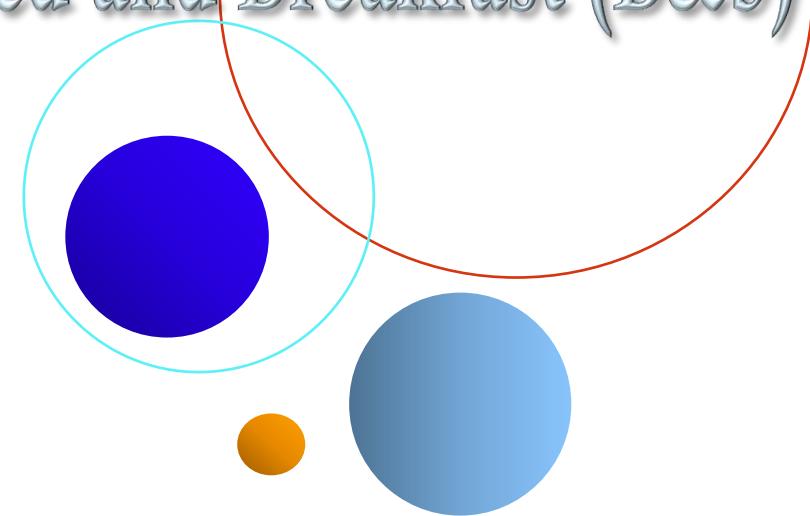
Basi di Dati

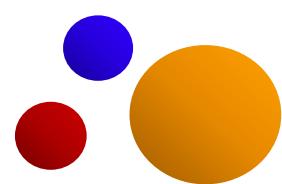


Relazione Progetto

A cura di *Maccarrone Alessia*, studentessa al C.d.L. di Informatica all'Università di Catania.

La gestione di un Bed and Breakfast (B&b)





Sommario

Premesse	3
Introduzione	4
1. Analisi della realtà: Specifica dei requisiti	5
Specifiche sui dati.....	5
Operazioni sui dati.....	6
Glossario dei termini	7
Decomposizione dei requisiti.....	9
2. Progettazione Concettuale: Schema E-R.....	10
Schema scheletro	10
Raffinamenti dello schema scheletro.....	11
Schema intermedio.....	19
Schema finale E-R per la fase concettuale	20
Vincoli non esprimibili dello schema E-R.....	21
Dizionario dei dati	22
Entità.....	22
Relazioni	24
3. Progettazione Logica: Schema Logico	26
Stime	26
Carico Applicativo	26
Tavola dei Volumi	27
Tavola delle Frequenze	28
Schemi delle operazioni.....	29
Analisi delle ridondanze	33
Eliminazione delle gerarchie	38
Accorpamento delle entità	40
Scelta degli identificatori principali	41
Schema E-R finale ristrutturato	42
Verso il modello relazionale	43
Traduzione delle associazioni molti a molti :.....	43
Traduzione delle associazioni uno a molti :.....	43
Traduzione delle associazioni uno a uno :.....	44
Schema logico	44
Diagramma UML.....	45
4. Progettazione Fisica	46
Premessa: traduzione linguistica.....	46
Definizione delle tabelle	48
Definizione delle operazioni	52
Trigger	56
Popolamento tabelle	60
Database esportato da phpMyAdmin	61
5. Interfaccia web	62
Demo di prova Laravel.....	62
Interfaccia illustrata.....	62
Repository github con corrispondente codice	63

Premesse

Questo rapporto si concretizza nella progettazione e nella creazione di un **database** relativo alla **gestione di un Bed and Breakfast**, circa le sue *camere, prenotazioni e della relativa fatturazione*.

In particolare, verrà effettuata una progettazione "**top-down**" sulla base delle specifiche relative al problema.

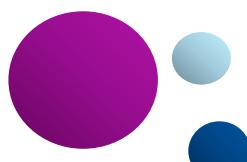
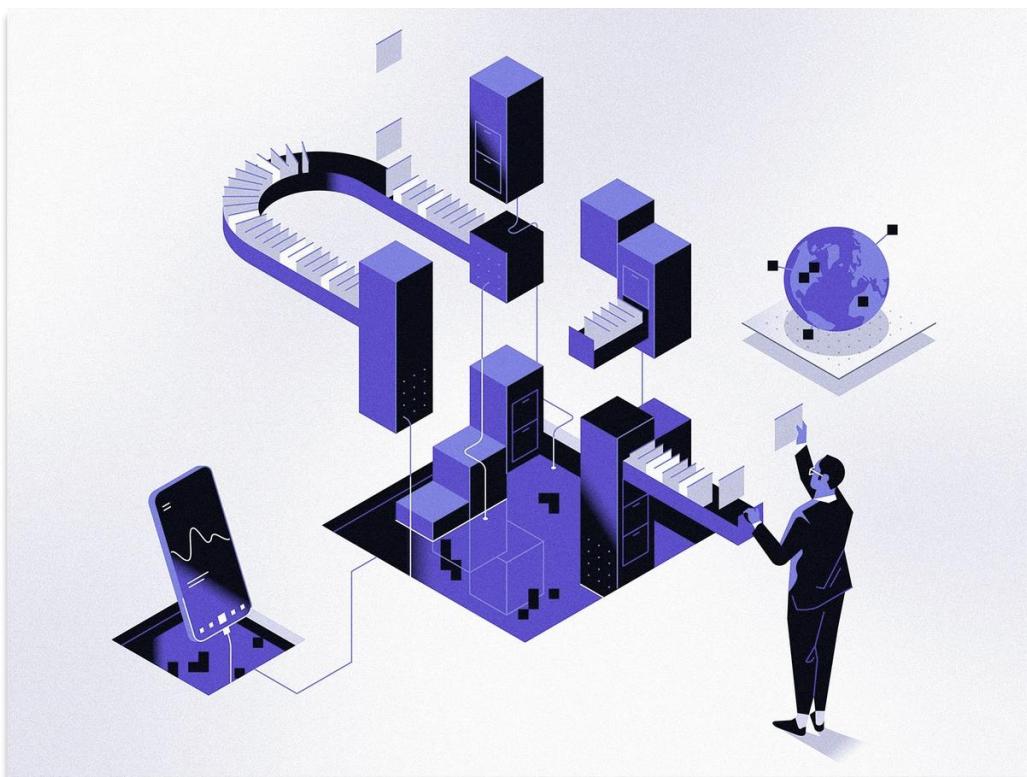
Ogni fase della progettazione verrà spiegata e documentata con diagrammi grafici, in modo da garantire un maggiore sul funzionamento logico e fisico del database in questione.

L'obiettivo principale del progetto è seguire i migliori design per creare un database funzionale caratterizzato da operazioni valide.

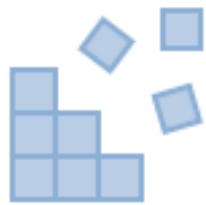
Pertanto, il progetto non rifletterà necessariamente l'implementazione di un sistema reale, ma farà riferimento a una **versione semplificata del problema**.

Tutti i file allegati verranno riportati nella sezione finale del PDF, ossia la Progettazione Fisica, ma anche sul relativo **repository su github**: <https://github.com/MaccarroneAlessia/database-project>.

Infine, nell'ultima sezione, sarà spiegata la **demo di un'interfaccia web** di esempio per i test e la visualizzazione grafica relativa a camere, prenotazioni e fatturazione.



Introduzione



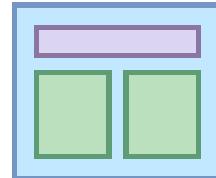
Si vuole realizzare il progetto relativo ad un database per la gestione di un Bed and Breakfast, circa le prenotazioni delle camere, la fatturazione e gestione del personale.

Da un set di specifiche, si procederà quindi all'analisi delle esigenze e alla progettazione concettuale, seguita dalla progettazione logica e, infine, dal layout fisico del database e le sue operazioni.

Di seguito si presenta la documentazione presente nel progetto:

1. **Analisi della realtà:** Specifica dei requisiti
 - o Analisi dei Requisiti
2. **Progettazione Concettuale:** Schema E-R
 - o Schema scheletro
 - o Dizionario dei dati: Entità e Relazioni
 - o Schema E-R e vincoli non esprimibili dallo schema E-R
 - o Dizionario dei dati
3. **Progettazione Logica:** Schema Logico
 - o Tavola dei Volumi e delle Frequenze
 - o Schemi delle operazioni
 - o Analisi delle ridondanze
 - o Eliminazione delle gerarchie
 - o Accorpamento delle entità
 - o Scelta degli identificatori
 - o Schema E-R finale
 - o Schema logico
 - o Diagramma UML
4. **Progettazione Fisica**
 - o Implementazione delle tabelle in SQL
5. **Interfaccia web**
 - o Demo di prova e test fatta con Laravel





1. Analisi della realtà: Specifica dei requisiti

La fase di analisi e definizione dei requisiti prevede innanzitutto lo svolgimento **un'indagine** preliminare in cui *si raccolgono le richieste degli utenti*: si analizza la situazione iniziale ed infine si producono indicazioni su come modellare la realtà di riferimento.

In base alle osservazioni e alle valutazioni si effettua poi una definizione maggiormente dettagliata, con l'obiettivo di identificare le proprietà e le funzionalità del sistema, fornendo così una **descrizione dei dati coinvolti e delle operazioni eseguibili** su di essi. Infine, deve essere prodotto un **glossario dei termini** caratterizzato dalle relative descrizioni, sinonimi e legami.

Specifiche sui dati

Si vuole realizzare il gestionale di un sistema di gestione relativo ad un bed and breakfast, un'attività che permette la prenotazione delle relative camere e richiesta di veri servizi, e la relativa fatturazione.

La figura principale che effettuerà le prenotazioni per sé ed eventualmente per altri è quella del **cliente**. Affinché il cliente venga correttamente registrato all'interno del sistema, è necessario che questo fornisca propri dati quali il nome, il cognome, l'e-mail e il numero di telefono.

I clienti prenoteranno le **camere** per un determinato **soggiorno** richiesto del cliente, che comprende la data di inizio e fine, indicando il numero di persone previste nella prenotazione, i servizi e/o i bisogni richiesti. Sulla base delle richieste del cliente viene assegnata una camera che rispecchia le richieste del cliente tra quelle **disponibili**.

Le camere saranno ubicate in varie **sedi** e di varie **tipologie** (camera singola, o doppia ad uso singola, camera doppia, matrimoniale, camera tripla, camera quadrupla) ognuna con **una propria tariffa**.

Si pensa ad un soggiorno con una o più stanze, con o senza bagno privato, che include di default il pernottamento e la prima colazione (senza la possibilità di eliminare questi servizi) ma con la possibilità per il cliente di selezionare altri servizi offerti (TV o altri apparecchi elettronici in stanza, parcheggio ecc.).

Quindi un servizio è legato alla camera o alla sede.

Quindi si presupporranno **varie fasce di prezzo** in base in base alla tipologia della camera, al **periodo** (inverno, estate, primavera, autunno) e ad altri **servizi** aggiuntivi richiesti dal cliente si calcolerà un costo complessivo.

Ogni utente può richiedere la propria **carta fedeltà** e ricevere punti, al fine di ottenere sconti ed entrare a far parte nel sistema di **clienti fidelizzati**; quindi, clienti che magari ritornano con le varie prenotazioni (storico, ci saranno tutte le spese e i pagamenti e così via).

Al termine del servizio, il cliente dovrà effettuare il **pagamento** per saldare la richiesta ed infine, si prevede un sistema di recensioni per le varie camere.

Altre figure fondamentali sono quelle relative al **personale**. Quest'ultimo composto dalle seguenti figure professionali:

- **Il direttore generale.** È manager dell'intera struttura ricettiva, gestisce l'intero staff amministrativo; quindi, si occupa dei compiti più importanti e genera un report degli incassi ogni mese.
- **L'addetto alla segreteria.** Si occupa dell'accoglienza degli ospiti ad inizio soggiorno e della presa e organizzazione delle prenotazioni che può avvenire anche telefonicamente.
- **L'addetto alla fornitura.** Si occupa della fornitura di tutto il necessario per la prima colazione e che sia garantita la fornitura di energia elettrica, acqua calda e fredda, riscaldamento.
- **Gli addetti al servizio pulizie.** Si occuperanno di pulire e disinfeccare ogni camera ad inizio e fine del soggiorno di ogni cliente ma anche quotidianamente negli orari comunicati, ma anche il cambio della biancheria, compresa quella da bagno, che deve avvenire due volte a settimana e a cambio dell'ospite.

Infine, si prevede la possibilità di **amministrare la fatturazione e i resoconti**.

Operazioni sui dati

Per la gestione delle prenotazioni delle camere di un Bed and Breakfast sono previste alcune operazioni che bisogna essere in grado di soddisfare. Di seguito si riportano tali operazioni prese in considerazione:

01	Cliente effettua prenotazione per una camera	200 volte al giorno
02	Cliente modifica prenotazione (orario o data)	30 volte al giorno
03	Cliente cancella la prenotazione	30 volte al giorno
04	Ricerca di una prenotazione	50 volte al giorno
05	Calcolo del costo della prenotazione	250 volte al giorno
06	Visualizzare lo storico delle prenotazioni per un cliente	10 volte al giorno
07	Visualizzare la descrizione di una richiesta	20 volte al giorno
08	Visualizzare i dettagli del soggiorno di un cliente	30 volte al giorno
09	Visualizzare disponibilità di una camera	600 volte al giorno
10	Aggiungere punti fedeltà nella tessera di un cliente	30 volte al giorno
11	Visualizzare punti nella tessera di un cliente	40 volte al giorno
12	Visualizzare recensione camera	500 volte al giorno
13	Gestore genera report	1 volta al mese
14	Aggiunta di una nuova sede	1 volta ogni 5 mesi

Glossario dei termini

TERMINE	DESCRIZIONE	SINONIMI	TERMINI COLLEGATI
CLIENTE (customer)	Cliente del B&B, persona che ha già effettuato una prenotazione e che può prenotare una o più camere e usufruirà dei servizi legati al soggiorno . Otterrà punti al termine del pagamento che dovrà essere effettuato al termine del soggiorno.	Utente, acquirente , compratore, consumatore, frequentatore più o meno abituale.	Camera, Pinti, Richiesta, Prenotazione, Soggiorno, Fattura.
CAMERA (room)	Stanza del Bed and Breakfast che accoglierà i clienti la notte, destinata ad essere oggetto di una prenotazione per un determinato soggiorno. Ha una tipologia, una fascia di prezzo ed è destinata ad essere usufruita da tot persone .	Locale, stanza , vano.	Sede, Disponibilità, Servizio, Prenotazione, Costo
SEDE (location)	Ogni luogo in quanto s'identifichi nell'ubicazione o localizzazione della residenza riconosciuta, del campo di svolgimento del Bed and Breakfast.	Alloggio, casa, dislocazione, domicilio, posizione , posto, collocazione, luogo.	Camera, Recapito, Servizio, Gestore, Segretario.
SERVIZIO (service)	Prestazione, complesso di mezzi costituiti e organizzati per soddisfare a un bisogno del cliente via una richiesta	Prestazione, commissione, articolo, set, trattamento.	Camera, Sede, Richiesta, Prenotazione, Costo.
DISPONIBILITÀ (availability)	La possibilità di libera utilizzazione della camera che può essere ancora prenotata .	Disposizione, libera.	Camera, Prenotazione.
RICHIESTA (request)	Domanda, per lo più motivata da una prassi o da una necessità; con riferimento al prezzo richiesto per una prenotazione di una camera e dei servizi.	Domanda.	Cliente, Camera, Servizio.
PRENOTAZIONE (booking)	Impegno a occupare e diritto ad aver riservato una camera o più, per sé o anche altri, tramite una prenotazione per un determinato periodo di tempo, con annessi i servizi pattuiti. Operazione effettuata dal cliente al termine del soggiorno con il metodo di pagamento concordato.	Impegno, riserva.	Cliente, Sede, Camera, Servizio.
COSTO (cost)	La spesa necessaria, ossia il prezzo in denaro che occorre versare, per ottenere la disponibilità della camera e usufruire dei servizi selezionati per tutto il soggiorno.	Prezzo , accordo, condizione, onere, tariffa, valore, spesa.	Camera, Servizio, Richiesta, Prenotazione.
STAGIONE (season)	Ciascuno dei quattro periodi in cui viene suddiviso l'anno. Importanti per l'attività turistica in quanto ogni stagione rappresenta un periodo con più affluenza turistica.	Periodo, tempo, momento.	Camera, Sede, Costo.

SOGGIORNO (sojourn/stay)	Permanenza temporanea in una camera del Bed and Breakfast.	Fermata, sosta, dimora, vacanza, permanenza .	Cliente, Camera, Sede.
FATTURA (ticket)	Documento con le indicazioni sulla camera fornita e i servizi annessi, dell'ammontare dell'importo e delle relative condizioni di pagamento, che il Bed and Breakfast trasmette al cliente per la durata del soggiorno prenotata.	Bolletta, scontrino, resoconto .	Cliente, Gestore, Sede.
SCONTO (discount)	Riduzione del prezzo, ottenuto dal cliente accumulando punti sulla propria tessera.	Abbuono, facilitazione, tara, saldo.	Cliente, Sede, Richiesta.
PERSONALE (staff)	Complesso degli impiegati del Bed and Breakfast, ossia di tutte le persone che ricoprono un ruolo e lavorano all'interno per garantire il giusto funzionamento della struttura.	equipe, squadra, gruppo, team, troupe, redazione.	Gestore, Segretario, Addetto.
GESTORE (manager)	Amministratore di beni o affari per conto del proprietario del Bed and Breakfast. Chi prende le decisioni importanti per il Bed and Breakfast.	Amministratore, direttore, responsabile.	Personale, Sede.
SEGRETARIO (secretary)	Incaricato di mansioni varie e riservate, quali il disbrigo della corrispondenza, lo svolgimento di pratiche amministrative e burocratiche, il coordinamento del Bed and Breakfast. In particolare la presa delle prenotazioni e accoglienza del cliente.	Assistente, collaboratore, aiutante, responsabile, amministratore, coordinatore, organizzatore.	Personale.
ADDETTO (officer)	Persona inquadrata in un ambiente di lavoro, destinato, assegnato a un determinato compito quali la fornitura del necessario per l'attività.	Incaricato, delegato , incaricato, specialista, funzionario, impiegato.	Personale.
RECAPITO (contact information)	Gli indirizzi e i contatti necessari del Bed and Breakfast e tutto lo staff incaricato ad interagire col pubblico, per il cliente, per l'assistenza, prenotazioni, e così via.	Consegna, luogo, posto, domicilio, indirizzo, contatti.	Sede, Personale.

Decomposizione dei requisiti

Tra le attività preliminari alla progettazione concettuale c'è la decomposizione, per cui di seguito verranno riportati i dati generali relativi agli elementi principali della base di dati in questione.

Dati di carattere generale

- Si vuole progettare un gestionale per un Bed and Breakfast, circa le prenotazioni delle camere e tutte le altre attività annesse.
- Bisogna far sì che il cliente abbia la possibilità di prenotare camere disponibili presso le varie sedi.
- All'interno del sistema si vuole tenere traccia di dati intrinseci relativi alle singole camere, come le richieste, i pagamenti e i servizi richiesti.
- Ogni sede ha il proprio personale, che si occupa di prendere le prenotazioni e preparare il soggiorno.
- Infine, i gestori devono avere la possibilità di generare report mensilmente al fine di controllare il fatturato.

Dati sul CLIENTE

- Per quanto riguarda clienti è prevista la rappresentazione di alcuni **dati anagrafici**, quali il nome, il cognome, l'email, il numero di telefono ma anche **dati bancari**.
- Ogni cliente registrato all'interno della base di dati verrà identificato tramite un id numero.

Dati sulla CAMERA

- Ogni Camera è identificata da un proprio id.
- Si vuole tener traccia del nome identificativo, della tipologia e del numero di letti, così da identificare quante persone può accogliere.

Dati sulla SEDE

- La sede è l'ubicazione dell'attività che contiene varie camere e il proprio personale.
- Si rende utile tenere traccia dell'indirizzo e dei recapiti legati alla residenza.

Dati sulla RICHIESTA

- La richiesta è effettuata dal cliente nel momento della prenotazione.
- Contiene tutte le informazioni utili per cercare la camera adatta al cliente.

Dati sul PERSONALE

- Il personale di ogni sede è suddivisa in gestore, segretario e i vari addetti.
- I segretari sono la figura professionale più vicina al cliente, mentre gli addetti si occuperanno di tutte quelle attività che rendono possibile l'attività, quali la manutenzione e fornitura, successivamente troviamo il gestore che genera report mensili.
- Il CF è l'identificativo di ogni componente del personale, inoltre, si vogliono rappresentare altri dati come il telefono, l'email e lo stipendio.

Dati sulla FATTURA

- Importante per i report e lo storico delle transizioni del cliente.
- Utile conservare i movimenti monetari e tenere traccia dei documenti finanziari dell'attività



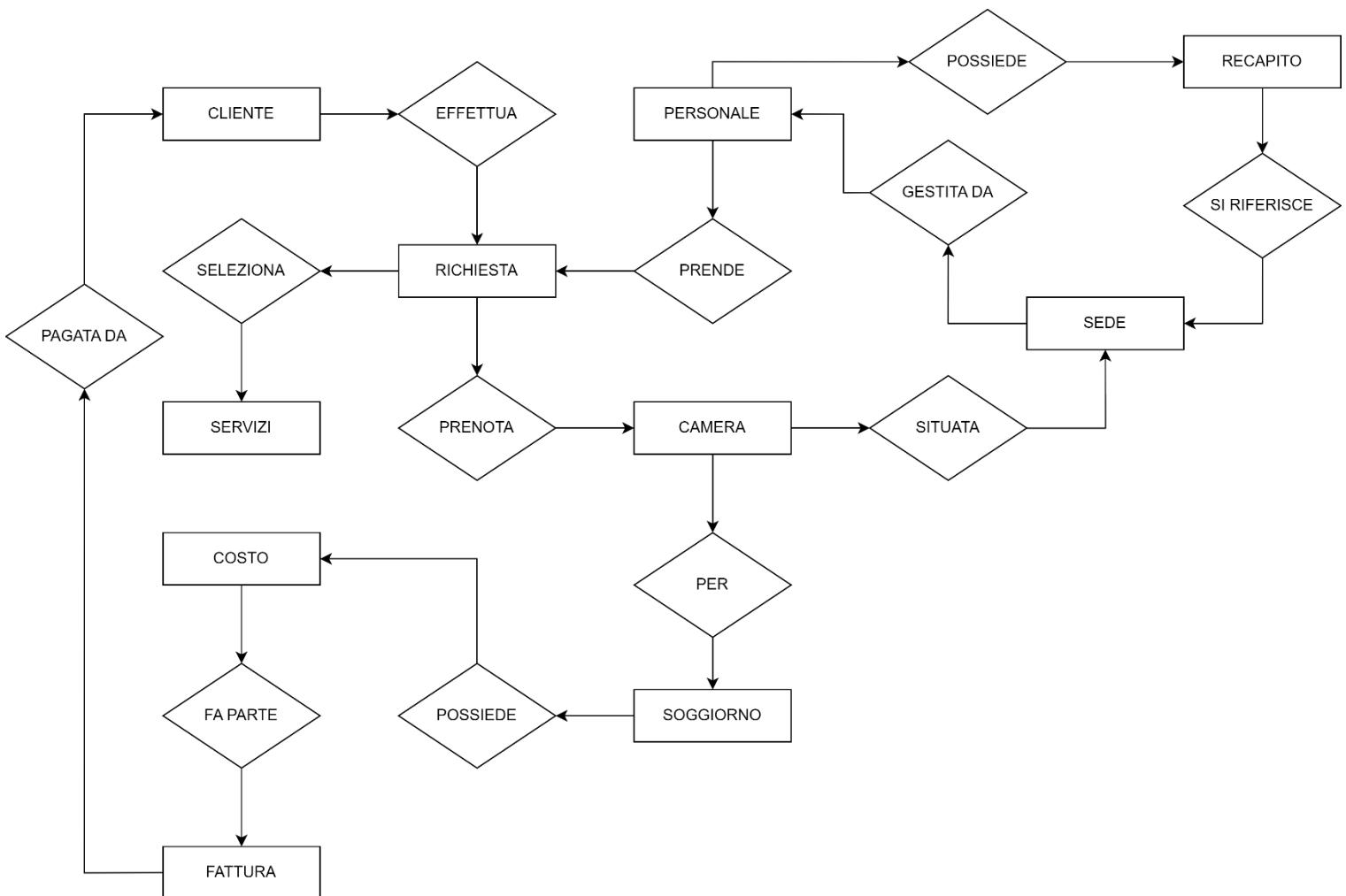
2. Progettazione Concettuale: Schema E-R

La progettazione concettuale ha come compito quello di prendere in **input le specifiche informali** della realtà di riferimento, per **produrre una descrizione formale** e completa che sia però indipendente dalla rappresentazione utilizzata nel DBMS. In altre parole, la progettazione concettuale vuole rappresentare il contenuto informativo della base di dati *senza preoccuparsi dell'effettiva implementazione*. L'output di questa fase è il cosiddetto **schema concettuale o diagramma Entità/Relazioni**.

Verrà attuata la strategia **top-down**: Partendo da un insieme di requisiti e specifiche, si procede alla stesura di uno **schema di base**. Successivamente, dopo aver apportato modifiche sulla base di vari raffinamenti, si ottiene uno **schema intermedio**. Infine, applicando ulteriori raffinamenti, si procede con la stesura dello **schema finale**.

Schema scheletro

Il seguente **diagramma iniziale** viene tracciato sulla base delle specifiche e dei principali termini ottenuti. I termini di base sono tradotti in **entità**, le quali collegano tra loro secondo le informazioni ottenute dal testo analizzato nella fase precedente.



Raffinamenti dello schema scheletro

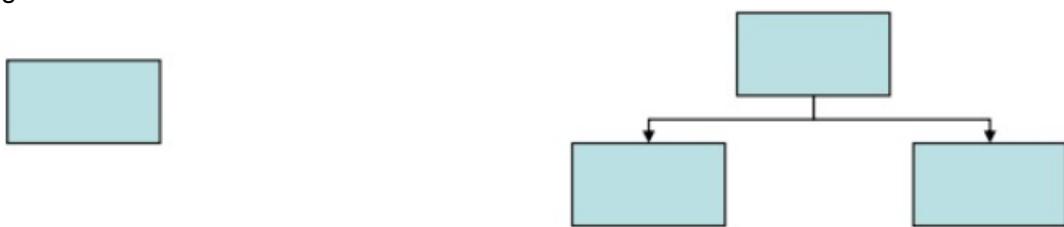
Si procede all'applicazione di trasformazioni elementari, dette primitive della trasformazione TOP-DOWN, al fine di modificare e affinare lo schema.

Le trasformazioni che useremo sono le seguenti:

- **T1:** applicata quando un'entità descrive due concetti diversi legati fra di loro, trasforma l'entità in due entità e una relazione.



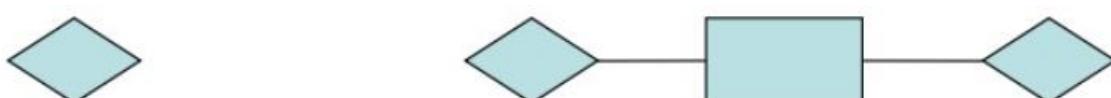
- **T2 Generalizzazione:** quando un'entità è composta da sotto-entità distinte allora diventa una gerarchia di entità correlate.



- **T3:** quando una relazione in realtà descrive due relazioni diverse tra le stesse entità viene trasformata in due relazioni.



- **T4:** quando una relazione descrive un concetto con esistenza autonoma. In questo caso si sostituisce con un'entità e le relative relazioni.



- **T5:** si applica per aggiungere attributi ad entità.

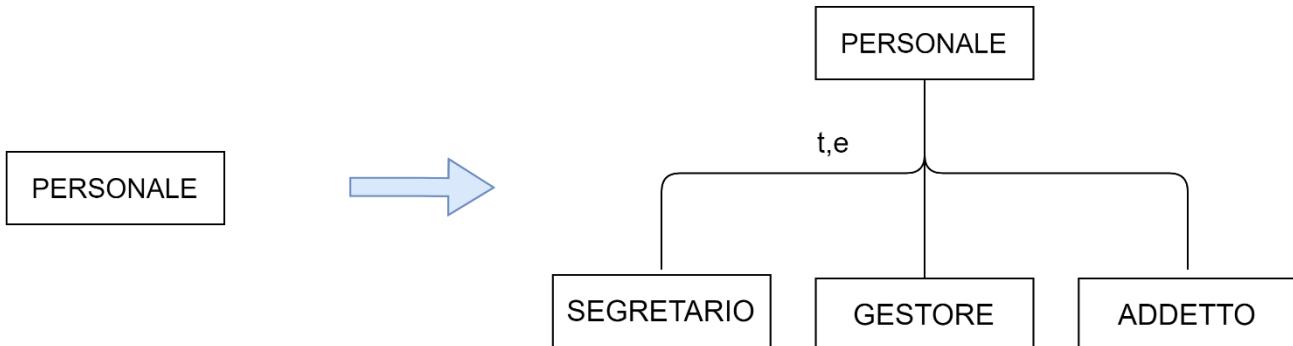


- **T6:** si applica per aggiungere attributi a relazioni.



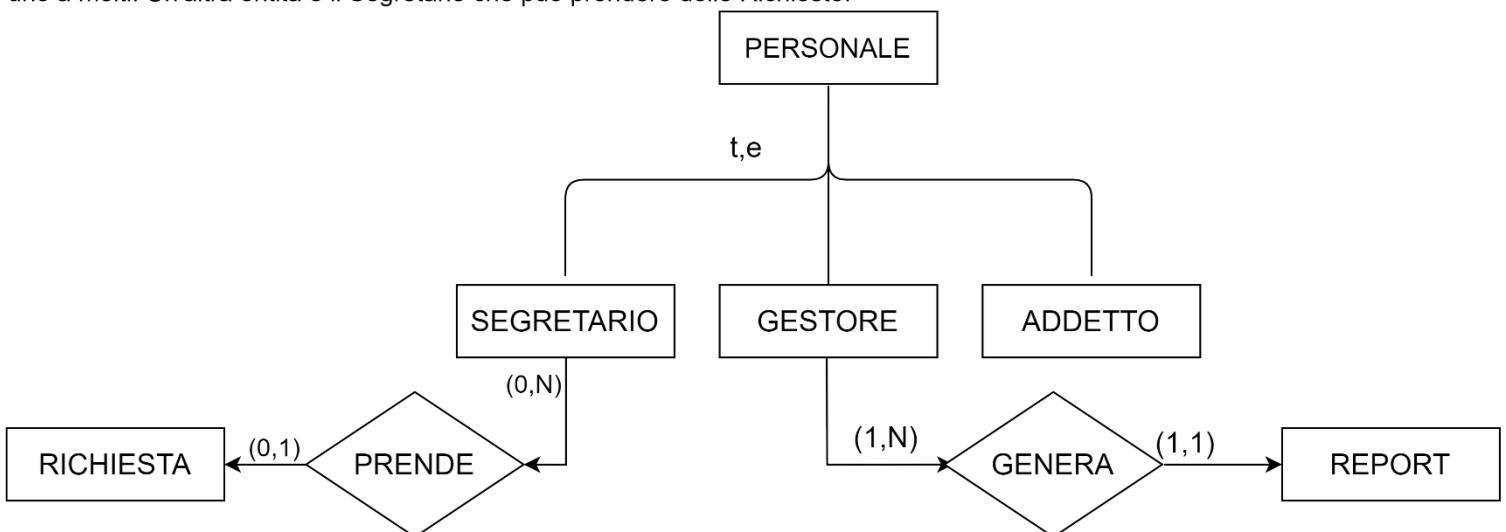
Trasformazione n.1 : Generalizzazione (Regola T2)

L'entità personale è chiaramente una generalizzazione a da cui si possono estrarre termini più specifici. In particolare, si ricavano i termini Gestore, Segretario, Addetto. Ogni membro del personale svolge determinati ruoli all'interno del ristorante. Poiché ogni ruolo è distinto e nessuno può svolgere più ruoli, si tratta di una generalizzazione totale ed esclusiva.



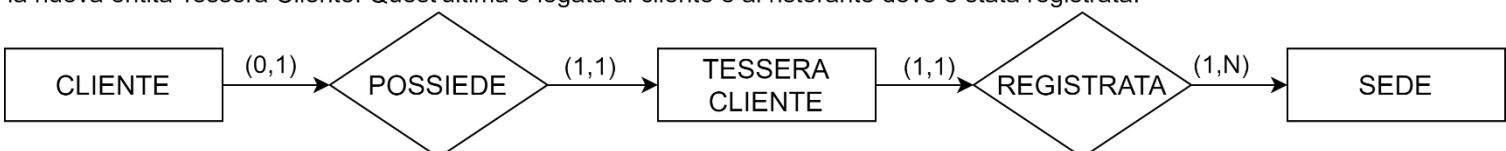
Trasformazione n.2 : Aggiunta di una nuova entità con relazione (e cardinalità)

Si considera l'entità Gestore. Questo ha la facoltà di generare un report mensile, nel quale mette in evidenza il numero di clienti mensili e il fatturato. Viene quindi aggiunta la nuova entità Report con la rispettiva relazione generata. Si tratta di una relazione di tipo uno a molti. Un'altra entità è il Segretario che può prendere delle Richieste.



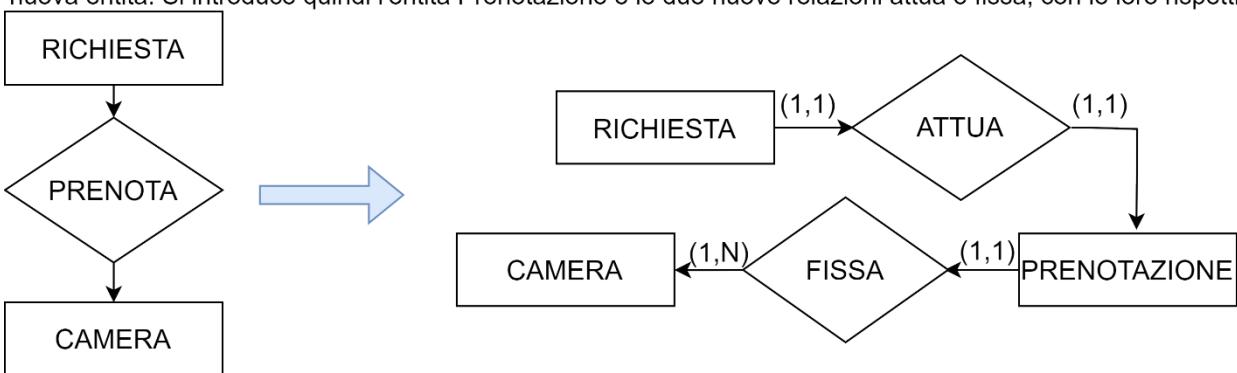
Trasformazione n.3 : Definizione di una nuova entità (TesseraCliente)

Si considera l'entità Cliente. Per una maggiore coerenza dello schema, risulta necessario effettuare un partizionamento, ottenendo la nuova entità Tessera Cliente. Quest'ultima è legata al cliente e al ristorante dove è stata registrata.

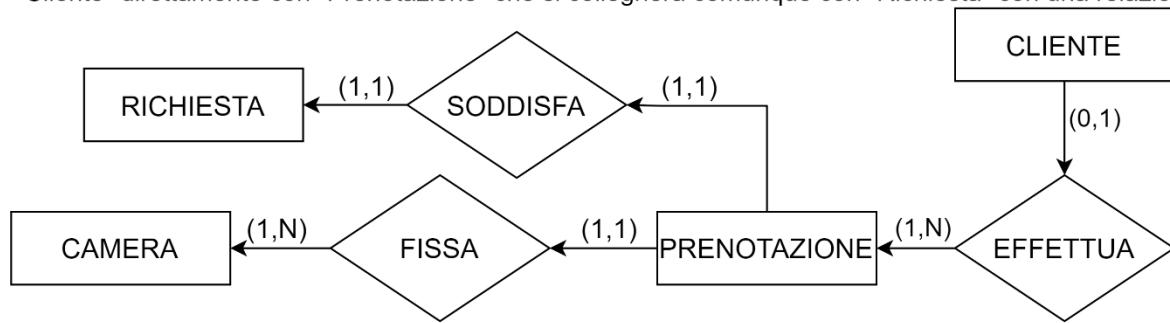


Trasformazione n.4 : Relazione che esprime un concetto con esistenza autonoma (Reg. T4 applicata a PRENOTA)

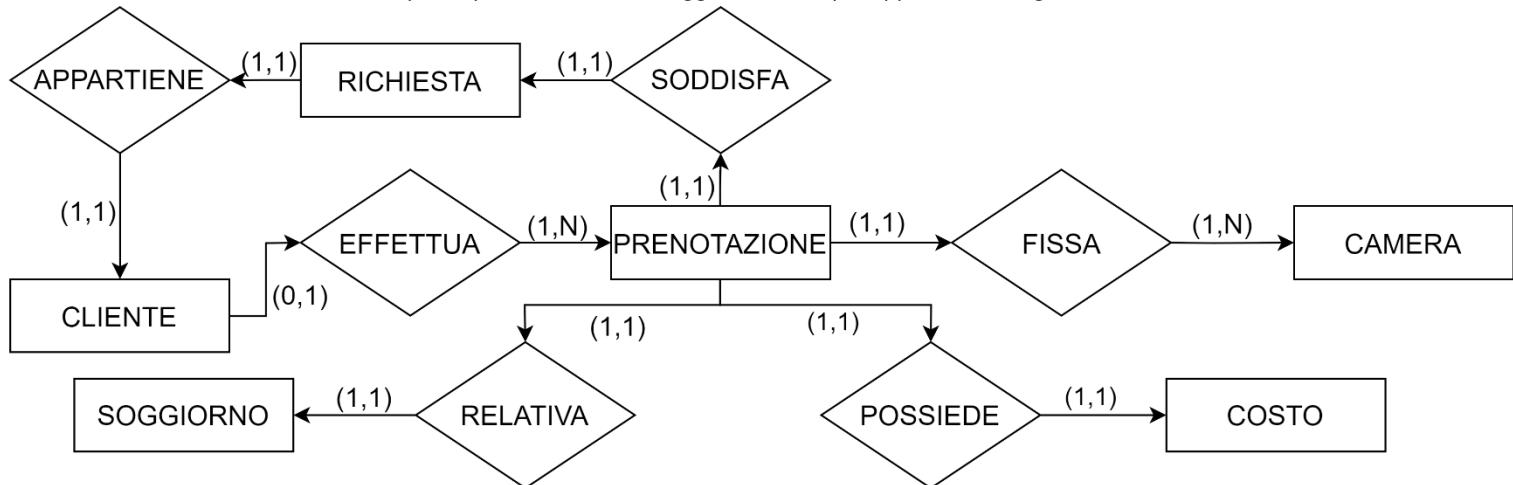
Si considera la relazione prenota, posta tra le entità Richiesta e Camera. Risulta evidente che nella base di dati in questione, il concetto di "Prenotazione" sia fondamentale ed è necessario, quindi, apportare una sostituzione della relazione Prenota con una nuova entità. Si introduce quindi l'entità Prenotazione e le due nuove relazioni attua e fissa, con le loro rispettive cardinalità.



Guardando però la relazione tra "Cliente" e "Richiesta" attraverso la relazione "Effettua" adesso sembra più adatto a collegare "Cliente" direttamente con "Prenotazione" che si collegherà comunque con "Richiesta" con una relazione diversa, ossia "Tramite".

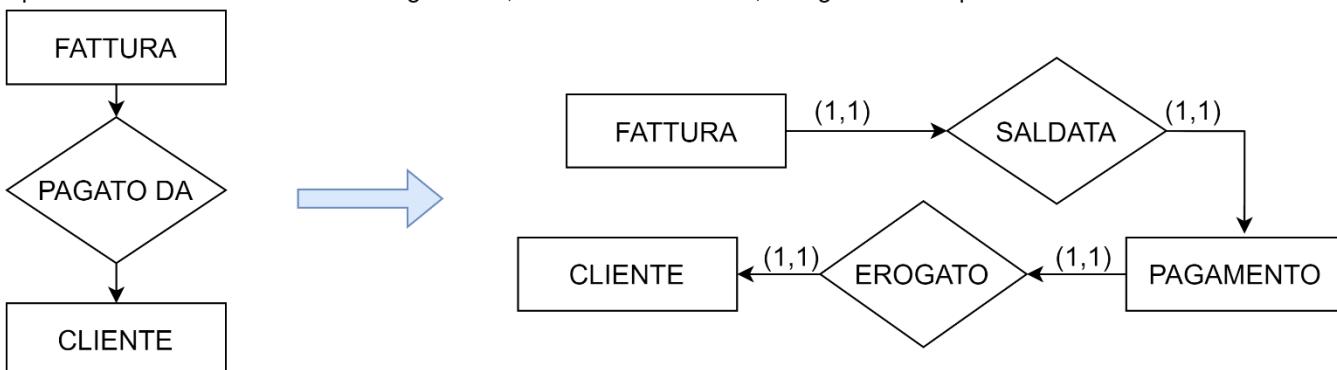


A questo punto risulta opportuno collegare l'entità "Soggiorno" con l'entità "Prenotazione", tramite la relazione "Relativa", e "Richiesta" a "Cliente". Una relazione da sistemare poi è quella tra costo e Soggiorno che è più opportuna collegarla a Prenotazione.



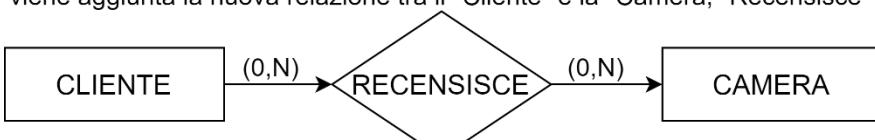
Trasformazione n.5: Relazione che esprime un concetto con esistenza autonoma (Regola T4 applicata a PAGATO DA)

Si considera la relazione "Pagato da" tra le entità "Fattura" e "Cliente", poiché la relazione "pagata dal" esprime, in realtà, un concetto con esistenza autonoma, ovvero quello dei pagamenti, risulta necessario effettuare una sostituzione con un'entità. Viene quindi introdotta la nuova entità Pagamento, le relazioni "Saldato", "Erogato" e le rispettive cardinalità.



Trasformazione n.6: Definizione nuova relazione che definisce l'azione di Recensione della camera

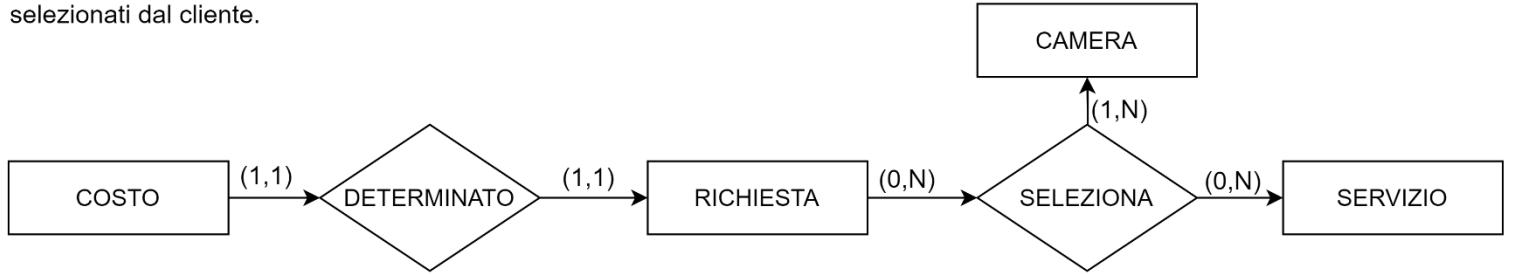
Si vuole aggiungere un sistema che possa tener traccia delle recensioni che i clienti effettuano nei confronti della camera. Pertanto, viene aggiunta la nuova relazione tra il "Cliente" e la "Camera", "Recensisce" che rappresenta l'azione di commentare la camera.



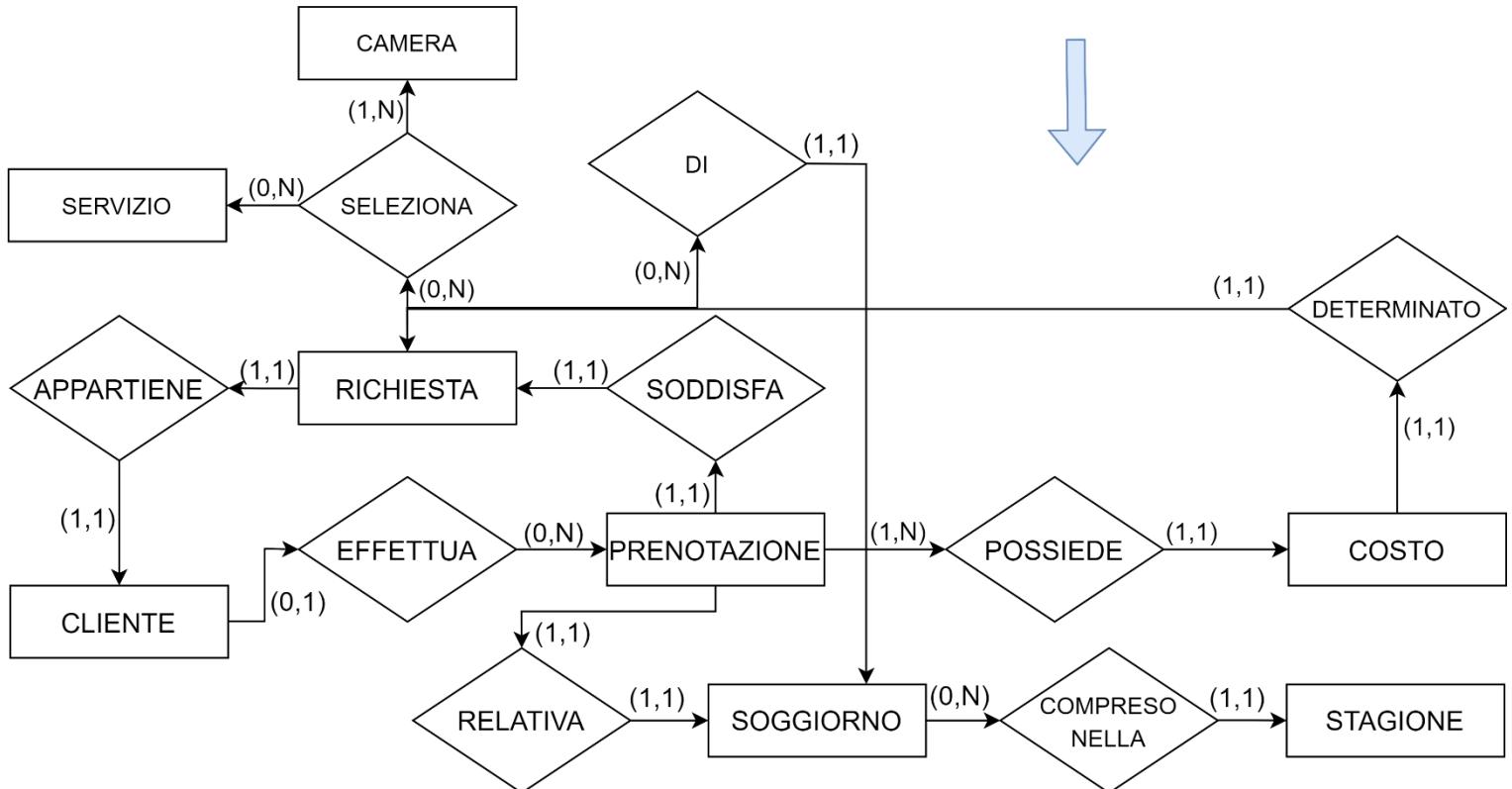
Si tratta di una relazione di tipo molti a molti, infatti, più clienti possono recensire più camere e la stessa camera può essere recensita da più clienti, quindi contenere più recensioni.

Trasformazione n.7: Definizione di una nuova entità (Stagione)

Un'altra entità da sistemare è quella del Costo, determinato dal Prezzo della Camera in base alla tipologia e dei relativi servizi selezionati dal cliente.

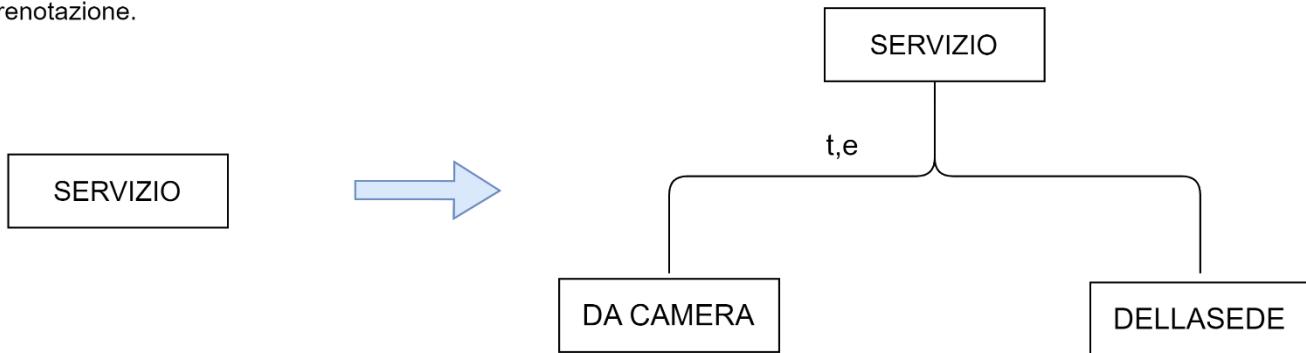


Guardando però il quadro generale vediamo che se la Richiesta seleziona la Camera, è superfluo dire che la prenotazione fissa la camera se la Prenotazione soddisfa la Richiesta

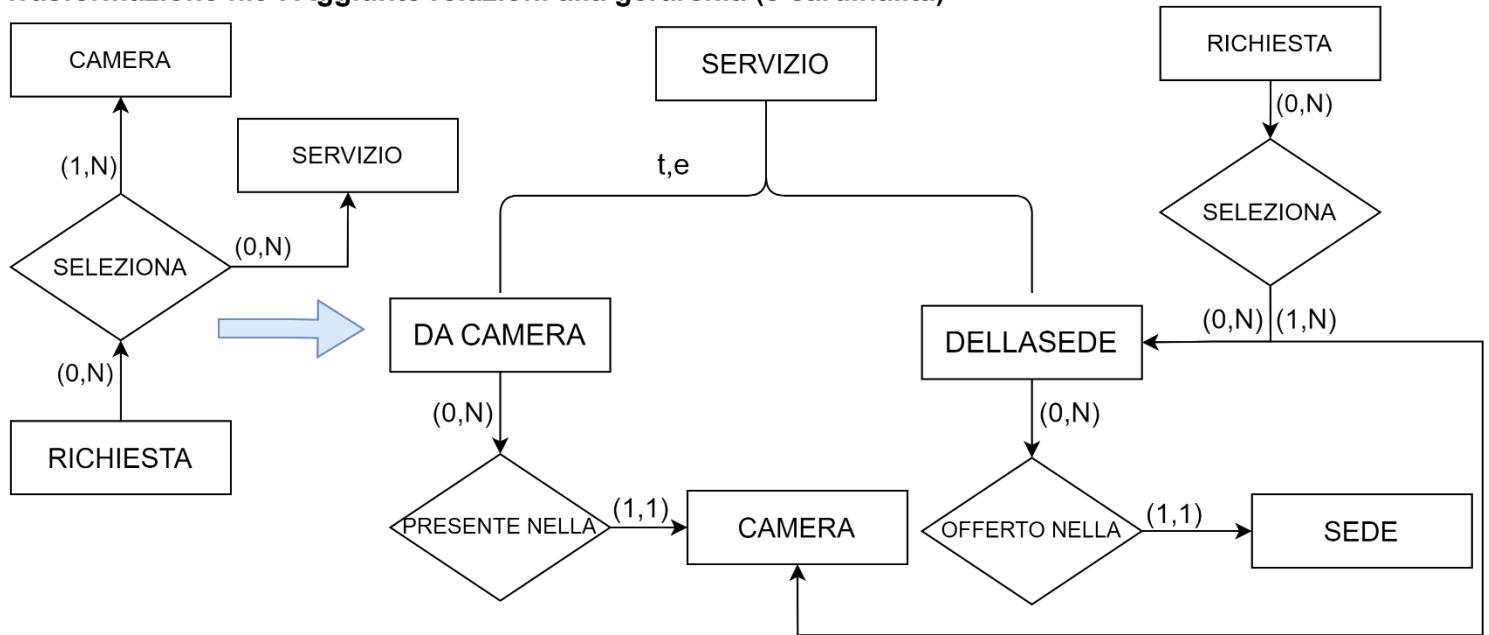


Trasformazione n.8 : Generalizzazione (Regola T2)

L'entità Servizi è chiaramente una generalizzazione a cui si possono estrarre termini più specifici. In particolare, si ricavano i termini Servizi da camera e Servizi della sede. I servizi dipendono dalla loro appartenenza, i primi sono compresi nella camera mentre i secondi sono servizi che offre la sede, entrambi sono servizi selezionabili dal cliente e che aggiungeranno un costo alla prenotazione.



Trasformazione n.9 : Aggiunte relazioni alla gerarchia (e cardinalità)

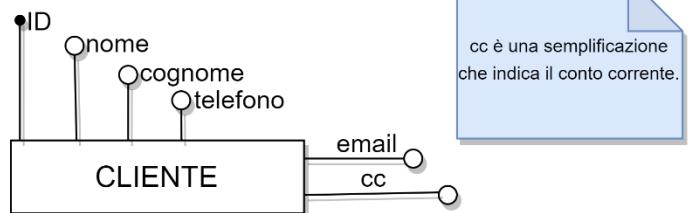


Trasformazione n.10: Aggiunta degli attributi a entità (reg. T5) e relazioni (reg. T6)

In questa trasformazione si procede con l'inserimento dei principali attributi sulle entità e sulle relazioni definite finora, supponendo, in una prima fase, quali chiave esterne potrebbero contenere le varie entità.

- Attributi per l'entità Cliente:

Entità collegata ad altre entità quali Richiesta, Prenotazione, Camera attraverso l'associazione molti a molti Recensisce, Pagamento e Tessera. Tra queste prende, come chiave esterna solo l'id della TesseraCliente.



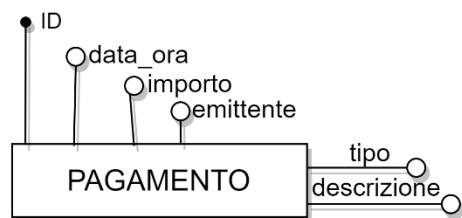
- Attributi per l'entità TesseraCliente:

Entità in relazione con Cliente e Sede.



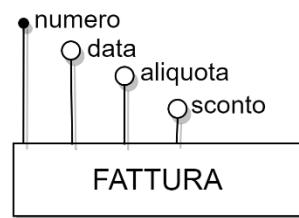
- Attributi per l'entità Pagamento:

Entità in relazione con Cliente e Fattura.
Tra questi eredita l'id del cliente come chiave esterna.



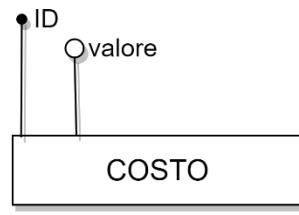
- Attributi per l'entità Fattura:

Entità in relazione con Pagamento, Costo e Report.
Di tutte e tre prende il loro id come chiave esterna.



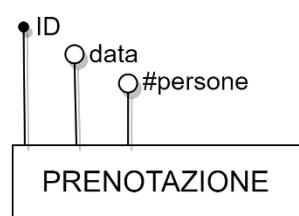
- Attributi per l'entità Costo:

Entità in relazione con Richiesta, Prenotazione e Fattura.
Tra questi eredita le chiavi delle entità Richiesta e Fattura.



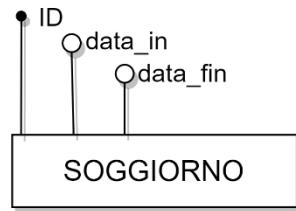
- Attributi per l'entità Prenotazione:

Entità in relazione con le entità Cliente, Costo, Soggiorno e Richiesta.
Ed di tutte e tre ne eredita la chiave come chiave esterna.



- Attributi per l'entità Soggiorno:

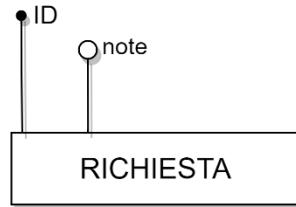
Entità in relazione con solo con l'entità Prenotazione.



- Attributi per l'entità Richiesta:

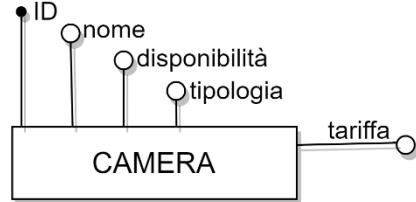
Entità in relazione con Camera, Servizio e Stagione, Segretario, Cliente, Prenotazione e Costo.

Di questi prende in eredità le chiavi delle entità Cliente, Camera, Servizio, Segretario e Stagione.



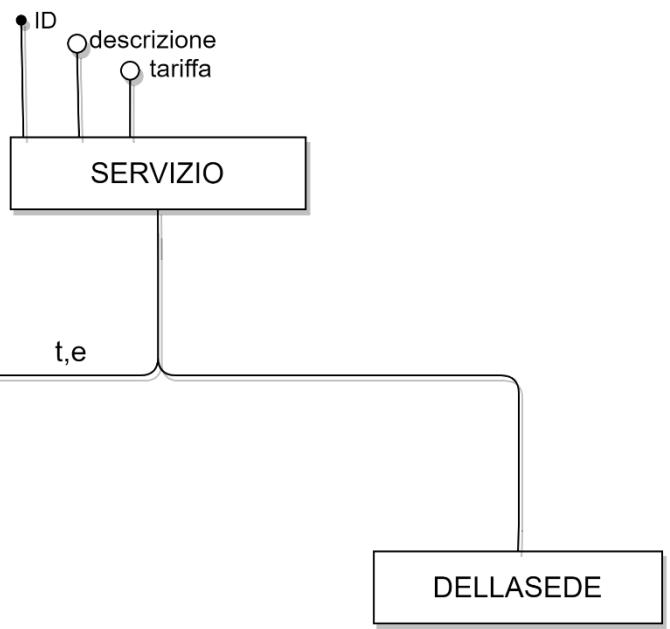
- Attributi per l'entità Camera:

Entità in relazione con Richiesta, Sede e Cliente attraverso l'associazione molti a molti Recensisce. Tra queste eredita la chiave della Sede e della Richiesta soltanto come chiavi esterne.



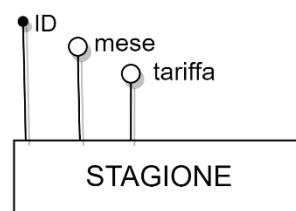
- Attributi per l'entità Servizio:

Entità in relazione con Richiesta e ne eredita la chiave primaria. Inoltre, essendo una generalizzazione, le due classi specifiche hanno rispettivamente una relazione in più, una con la camera e una con la sede in base a quale servizio si riferiscono e a cosa sia legato.



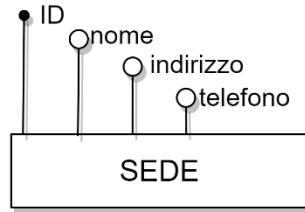
- Attributi per l'entità Stagione:

Entità in relazione con Richiesta e ne eredita la chiave primaria.



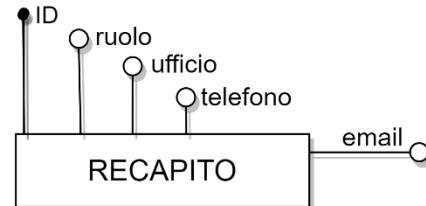
- Attributi per l'entità Sede:

Entità in relazione con Camera, Personale, Recapito e Tessera. Tra queste entità prende le chiavi della Tessera e Camera.



- Attributi per l'entità Recapito:

Entità in relazione con Personale e Sede. Tra queste prende la Chiave della Sede come chiave esterna.

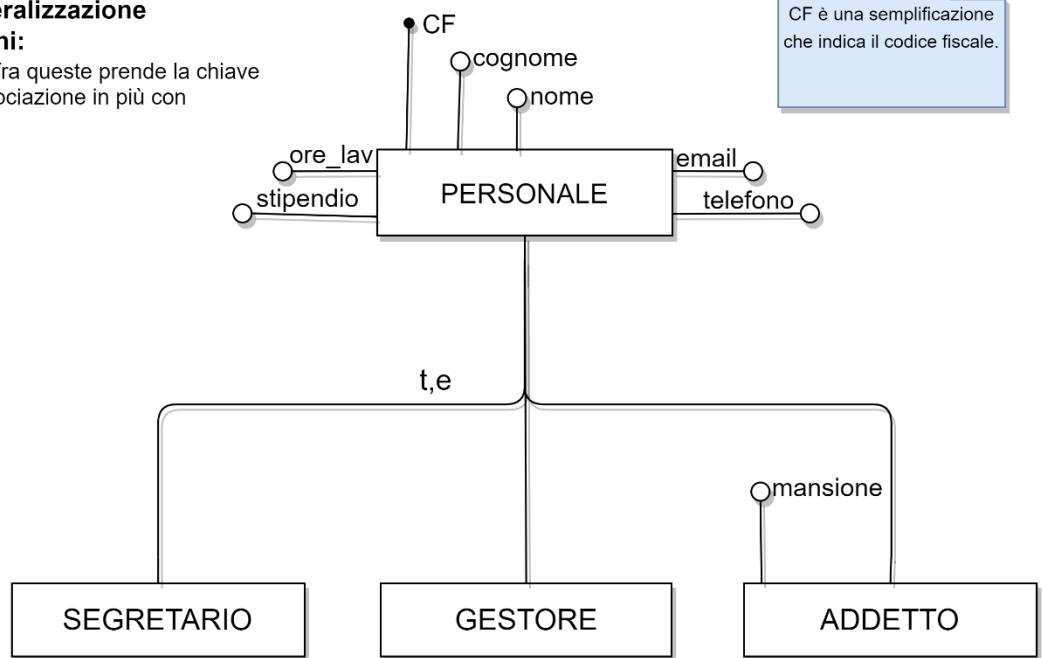


- Attributi per la gerarchia di generalizzazione

Personale e le sue specializzazioni:

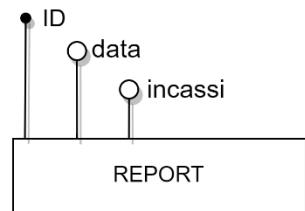
Entità in relazione con Sede e Recapito. Tra queste prende la chiave della Sede soltanto. Segretario ha un associazione in più con Richiesta e Gestore con Report.

CF è una semplificazione che indica il codice fiscale.



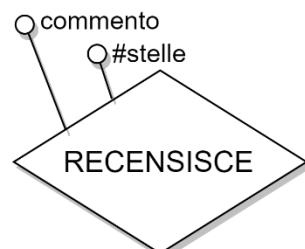
- Attributi per l'entità Report:

Entità in relazione con Gestore e Fattura. Tra queste eredita la chiave del Gestore soltanto.



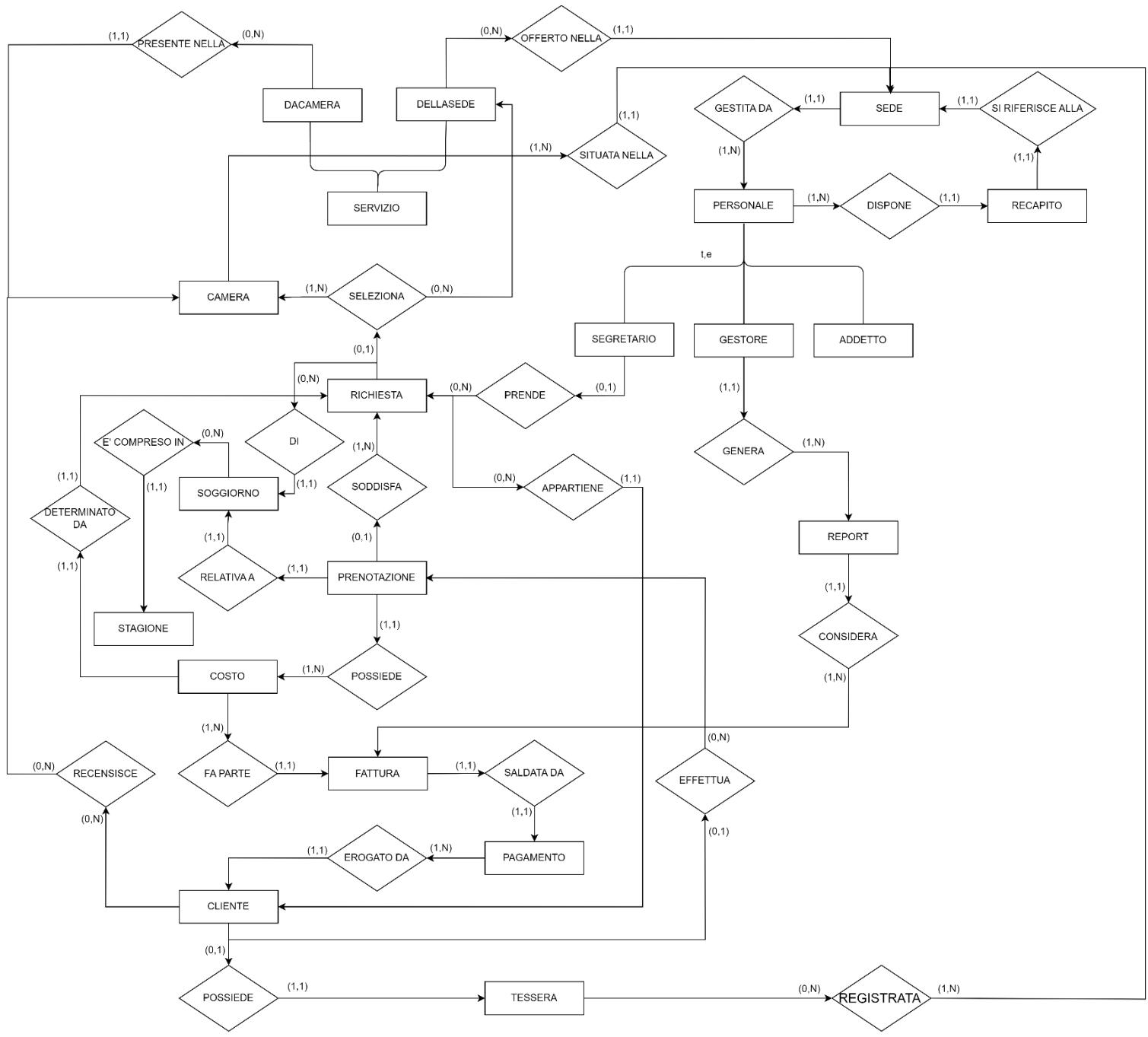
- Attributi per la relazione Recensisce:

Associazione molti a molti tra le entità Cliente e Camera. Dalle quali erediterà sicuramente le chiavi come chiavi esterne e primarie insieme al numero delle stelle della recensione.



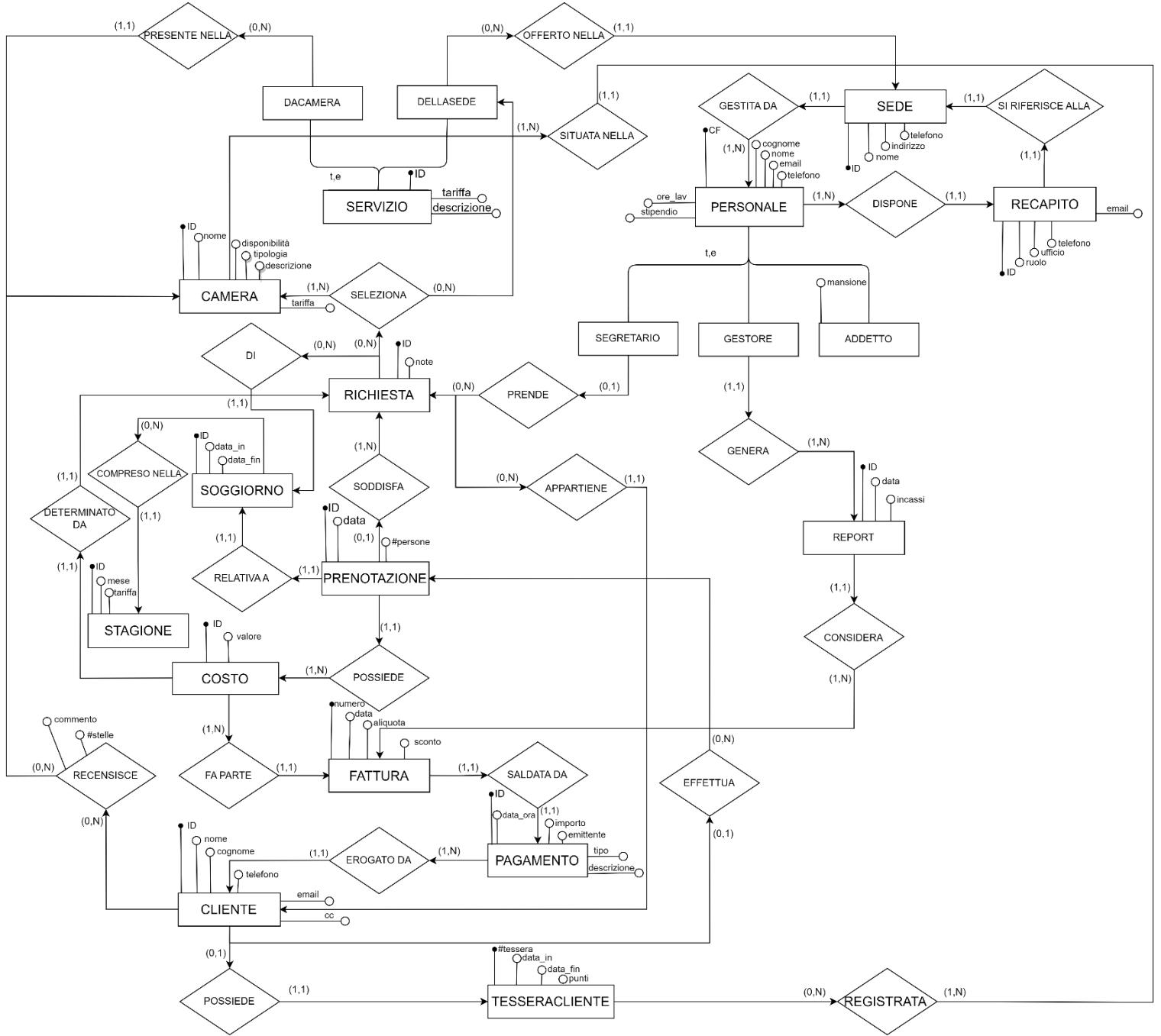
Schema intermedio

Ecco come si presenta lo schema dopo l'applicazione delle precedenti trasformazioni:



Schema finale E-R per la fase concettuale

Invece, il seguente schema rappresenta come si presenta lo schema una volta aggiunti gli attributi. Questo schema è lo schema finale per la fase concettuale ma potrebbe essere rivisitato o subire ulteriori modifiche nelle fasi successive.



Vincoli non esprimibili dello schema E-R

Di seguito si riportano alcuni vincoli che non possono trasparire dallo schema Entità-Relazione:

Prenotazioni dei clienti:

- Un cliente può effettuare una prenotazione per una certa data e ora in una determinata sede solo se vi è disponibilità delle camere.
- Sarà il personale, al momento della prenotazione, ad occuparsi di verificare se vi sono camere disponibili

Soggiorno

- Un soggiorno può contenere più prenotazioni o richieste ma di camere diverse, o se della stessa camera, di date diverse.

Tessera fedeltà dei clienti:

- Un cliente può registrare una e una sola tessera.
- Inoltre, gli sconti sono applicabili solo durante il periodo di attivazione di quest'ultima.

Stipendio Segretari o Addetti:

- Un segretario o un addetto non può avere uno stipendio maggiore di quello del gestore.

Pagamenti dei Clienti:

- Un cliente può effettuare un pagamento solo dopo aver concluso la prenotazione, quindi non potrà modificare la richiesta.

Dizionario dei dati

Di seguito sono riportati i dizionari delle Entità e delle Relazioni presenti nello schema E-R.

Entità

ENTITA'	Descrizione	Attributi	Identificatore
Cliente	Persona che usufruisce delle camere e i servizi delle sedi del bed and breakfast.	Id, nome, cognome, telefono, mail, cc.	Id
TesseraCliente	Tessera intestata al cliente che permette di ottenere sconti.	Id_tessera, inizio, fine, punti.	Id_tessera
Pagamento	Transizione effettuata dal cliente per usufruire del servizio.	Id_pagamento, tipo, importototale, data_ora, descrizione.	Id_pagamento
Fattura	Elenco delle contabilità, o meglio transizioni economiche.	Numero, data, aliquota, sconto.	numero
Costo	Valore calcolato per una richiesta.	Id_costo, valore.	Id_costo
Prenotazione	È l'atto del cliente di riservare una camera al fine di poi usufruirla per il suo soggiorno.	Id_prenotazione, data, num_persone.	Id_prenotazione
Soggiorno	È il lasso di tempo per cui il cliente prenota una camera.	Id_soggiorno, data_in, data_fine.	Id_soggiorno
Richiesta	La richiesta è la selezione della camera e servizi richiesti dal cliente per il suo soggiorno.	Id_richiesta, note.	Id_richiesta
Camera	Elenco delle camere prenotabili dai clienti quindi messe a disposizione da una sede.	Id_camera, nome, disponibilità, tipologia, tariffa, descrizione.	Id_camera
Servizio	Elenco dei servizi prenotabili dai clienti quindi messe a disposizione da una sede o dalla camera.	Id_servizio, descrizione, tariffa.	Id_servizio
DaCamera	Servizi presenti in una camera quindi inseriti in una prenotazione a prescindere se selezionati o meno singolarmente, basti selezionare la camera. (bagno privato, tv...)	Nessun attributo in più rispetto alla generalizzazione. (vedi servizio)	vedi servizio
DellaSede	Servizi che il cliente può decidere di selezionare o meno che sono presenti nella sede. (posto auto...)	Nessun attributo in più rispetto alla generalizzazione. (vedi servizio)	vedi servizio
Stagione	Elenco dei periodi nei quali i clienti prenoteranno, utili per sconti o aggiunte di prezzo sull'ammontare del costo della prenotazione.	Id_stagione, mese, tariffa.	Id_stagione
Sede	È il luogo allibito all'attività turistica, quindi una casa con delle camere allibita ad essere prestata a delle prenotazioni.	Id_sede, nome, indirizzo, telefono.	Id_sede
Recapito	Sono gli indirizzi di una sede collegati al personale ai quali i clienti possono rivolgersi per qualsiasi interazione o aiuto.	Id_recapito, ruolo, ufficio, telefono, email.	Id_recapito

Personale	Segretario, Gestore e addetti di una determinata sede dell'attività.	CF, cognome, nome, email, telefono, ore_lav, stipendio.	CF
Segretario	E' la persona incaricata a prendere le prenotazioni e in tutte quelle attività che riguardano il contatto con il pubblico.	Nessun attributo in più rispetto alla generalizzazione. (vedi personale)	vedi personale
Gestore	Colui che gestisce una sede e genera i report.	Nessun attributo in più rispetto alla generalizzazione. (vedi personale)	vedi personale
Addetto	È una terza figura professionale presente nella gestione dell'attività che si occuperà delle forniture o lavori più manuali che il servizio potrebbe richiedere, come la pulizia della sede.	Mansione e tutti gli altri attributi della generalizzazione.	vedi personale
Report	Generato dal gestore per visualizzare gli incassi mensili.	Id_report, data, incassi	Id_report

Relazioni

RELAZIONE	Entità partecipanti	Molteplicità	Descrizione	Attributi
Effettua	Cliente-Prenotazione	1:N	Un Cliente può effettuare più Prenotazioni e una Prenotazione è relativa a un solo Cliente.	-
Possiede	Cliente-TesseraCliente	1:1	Un Cliente può possedere una Tessera Cliente e una Tessera Cliente è posseduta da un solo Cliente	-
Registrata	TesseraCliente-Sede	N:N	Una tessera Cliente può essere registrata a più Sedi e in una Sede possono essere registrate più Tessere Cliente	
Erogato da	Pagamento-Cliente	N:1	Un Pagamento è erogato da un Cliente e un Cliente può erogare più Pagamenti.	-
Recensisce	Cliente-Camera	N:N	Un Cliente può recensire più Camere e una Camera può essere recensita da più Clienti	num_stelle, commento
Saldata da	Fattura-Pagamento	1:1	Una Fattura è saldata da un solo Pagamento e un Pagamento può saldare una sola Fattura.	-
Fa parte	Costo-Fattura	N:1	Un Costo fa parte di una Fattura e una Fattura può contenere più Costi.	-
Possiede	Prenotazione-Costo	1:N	Una Prenotazione può possiede più Costi e un Costo può essere posseduto da più Prenotazioni.	-
Determinato da	Costo-Richiesta	1:1	Un Costo è determinato da una Richiesta e una Richiesta ha un Costo.	-
Relativa a	Prenotazione-Soggiorno	1:1	Una Prenotazione è relativa a un Soggiorno e un Soggiorno è relativo a una Prenotazione.	-
E' compreso in	Soggiorno-Stagione	N:1	Un Soggiorno è compreso in una Stagione e in una Stagione possono essere compresi più Soggiorni.	-
Soddisfa	Prenotazione-Richiesta	1:N	Una Prenotazione può soddisfare più Richieste e una Richiesta può essere soddisfatta da una Prenotazione.	-
Appartiene	Richiesta-Cliente	N:1	Una Richiesta appartiene a un Cliente e un Cliente può fare più Richieste.	-

Seleziona	Richiesta-Camera Richiesta-Servizio della Sede	N:N N:N	Una Richiesta seleziona più camere e una Camera è selezionata da più Richieste. Una Richiesta può selezionare più Servizi della sede e un Servizio può essere selezionato da più Richieste.	
Di	Richiesta-Soggiorno	N:1	Una richiesta di un Soggiorno e un Soggiorno può essere di più Richieste	-
Presente nella	ServizioDaCamera-Camera	N:1	Un servizio da camera è presente in una camera e in una camera possono essere presenti più servizi.	-
Offerto nella	ServizioDellaSede-Sede	N:1	Un Servizio della Sede è offerto da una Sede e una Sede può offrire più Servizi.	-
Situata nella	Camera-Sede	N:1	Una Camera è situata in una Sede e in una Sede sono situate più Camere.	-
Gestita da	Sede-Personale	1:N	Una Sede è gestita da più Personale e un Personale gestisce solo una Sede.	-
Dispone	Personale-Recapito	N:1	Il Personale dispone di un Recapito ma un Recapito può essere disposto da più Personale.	-
Si riferisce alla	Recapito-Sede	1:1	Un Recapito si riferisce alla Sede e una Sede è il riferimento di un Recapito.	-
Prende	Segretario-Richiesta	1:N	Un Segretario prende più Richieste e una Richiesta è presa da un Segretario.	-
Genera	Gestore-Report	1:N	Un gestore può generare più Report e un Report è generato da un gestore.	-
Considera	Report-Fattura	1:N	Un report considera più Fatture e una Fattura è considerata in un Report	-



3. Progettazione Logica: Schema Logico

La fase successiva alla progettazione concettuale è quella della progettazione logica. In questa fase viene riorganizzato lo schema concettuale e si effettuano le opportune analisi e considerazioni sul carico applicativo e sulle eventuali ridondanze della base di dati in questione.

La progettazione logica prende come input lo schema concettuale prodotto nella fase precedente e lo traduce in uno **schema logico** che fa riferimento alla “famiglia” di database utilizzato, in questo caso si userà un **modello relazionale**.

Stime

Si riportano le seguenti stime:

- In media, si registrano circa 15 clienti per sede;
- Ogni sede prende circa 5 prenotazioni.
- In media, ogni richiesta contiene 1 camera e 1 servizio.

Carico Applicativo

Si procede a descrivere il carico applicativo tramite la **Tavola dei volumi** e la **Tavola delle frequenze**.

Ricordiamo che le operazioni considerate in precedenza sono le seguenti:

01	Cliente effettua prenotazione per una camera	200 volte al giorno
02	Cliente modifica prenotazione (orario)	30 volte al giorno
03	Cliente cancella la prenotazione	30 volte al giorno
04	Ricerca di una prenotazione	50 volte al giorno
05	Calcolo del costo della prenotazione	250 volte al giorno
06	Visualizzare lo storico delle prenotazioni per un cliente	10 volte al giorno
07	Visualizzare la descrizione di una richiesta	20 volte al giorno
08	Visualizzare i dettagli del soggiorno di un cliente	30 volte al giorno
09	Visualizzare disponibilità di una camera	600 volte al giorno
10	Aggiungere punti fedeltà nella tessera di un cliente	30 volte al giorno
11	Visualizzare punti nella tessera di un cliente	40 volte al giorno
12	Visualizzare recensione camera	500 volte al giorno
13	Gestore genera report	1 volta al mese
14	Aggiunta di una nuova sede	1 volta ogni 5 mesi

Tavola dei Volumi

Consideriamo il numero (medio) di occorrenze di ogni entità ed associazione.

Concetto	Tipo	Volume
Cliente	E	10.000
TesseraCliente	E	10.000
Pagamento	E	20.000
Fattura	E	20.000
Costo	E	25.000
Prenotazione	E	20.000
Soggiorno	E	20.000
Richiesta	E	40.000
Camera	E	40
Servizio	E	15
DaCamera	E	10
DellaSede	E	5
Stagione	E	12
Sede	E	5
Recapito	E	5
Personale	E	20
Segretario	E	5
Gestore	E	5
Addetto	E	10
Report	E	50
Effettua	R	20.000
Possiede	R	10.000
Registrata	R	10.000
Erogato da	R	20.000
Recensisce	R	15.000
Saldata da	R	20.000
Fa parte	R	25.000
Possiede	R	25.000
Determinato da	R	40.000
Relativa a	R	20.000
Compreso nella	R	20.000
Soddisfa	R	40.000
Appartiene	R	40.000
Seleziona	R	40.000
Di	R	40.000
Presente nella	R	40
Offerto nella	R	5
Situata nella	R	20
Gestita da	R	25
Dispone	R	25
Si riferisce alla	R	10
Prende	R	40.000
Genera	R	50
Considera	R	20.500

Tavola delle Frequenze

Nella tabella seguente considereremo il tipo di operazione (**interattiva** indicata semplicemente con I, sta per interattivo dove l'utente finale interagisce inserendo dei dati all'interno dei parametri, o **batch** indicata con B, il quale indica un tipo di operazione che non ha bisogno di alcun dato da inserire da parte dell'utente finale) e la **frequenza** (esecuzioni/tempo)

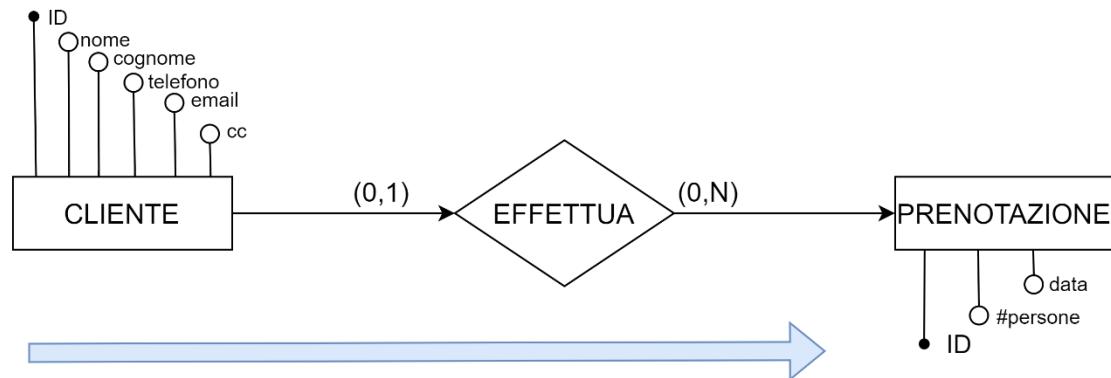
Operazione	Descrizione	Tipo	Frequenza
01	Cliente effettua prenotazione per una camera	I	200 volte al giorno
02	Cliente modifica prenotazione (data del soggiorno)	I	30 volte al giorno
03	Cliente cancella la prenotazione	I	30 volte al giorno
04	Ricerca di una prenotazione	I	50 volte al giorno
05	Calcolo del costo della prenotazione	B	250 volte al giorno
06	Visualizzare lo storico delle prenotazioni per un cliente	B	10 volte al giorno
07	Visualizzare la descrizione di una richiesta	B	20 volte al giorno
08	Visualizzare i dettagli del soggiorno di un cliente	B	30 volte al giorno
09	Visualizzare disponibilità di una camera	B	600 volte al giorno
10	Aggiungere punti fedeltà nella tessera di un cliente	B	30 volte al giorno
11	Visualizzare punti nella tessera di un cliente	B	40 volte al giorno
12	Visualizzare recensione camera	B	500 volte al giorno
13	Gestore genera report	I	1 volta al mese
14	Aggiunta di una nuova sede	I	1 volta ogni 5 mesi

Schemi delle operazioni

Schema operazioni O3-O6

Operazione O3 : Cliente cancella la prenotazione

Operazione O6 : Visualizzare lo storico delle prenotazioni per un cliente

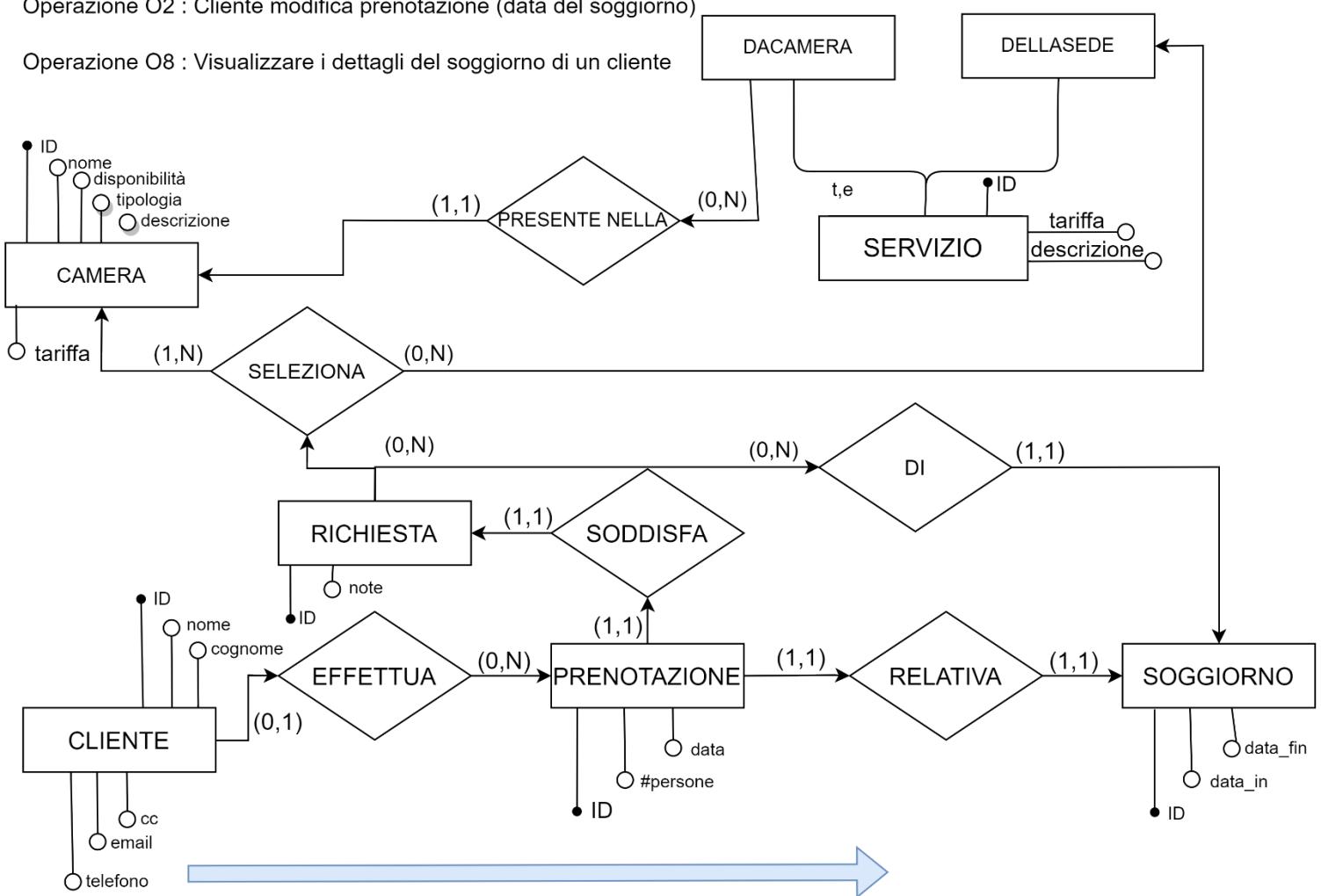


Schema operazioni O1-O2-O8

Operazione O1 : Cliente effettua prenotazione per una camera

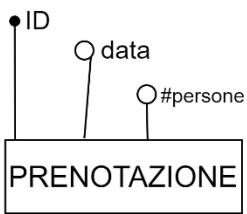
Operazione O2 : Cliente modifica prenotazione (data del soggiorno)

Operazione O8 : Visualizzare i dettagli del soggiorno di un cliente



Schema operazioni O4

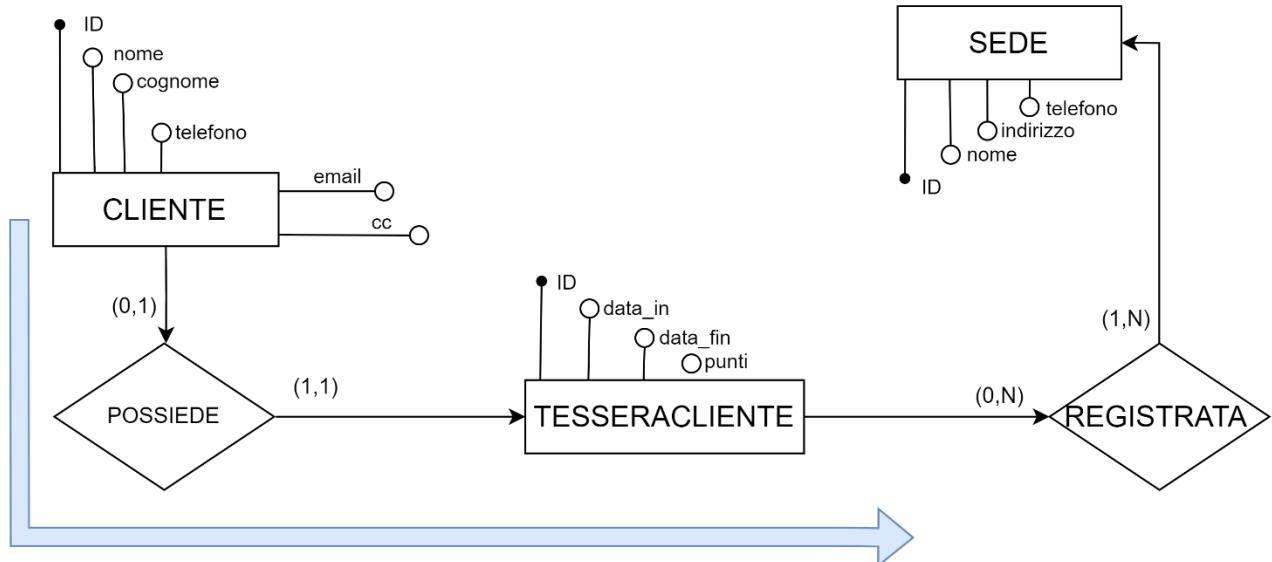
Ricerca di una prenotazione



Schema operazioni O10-O11

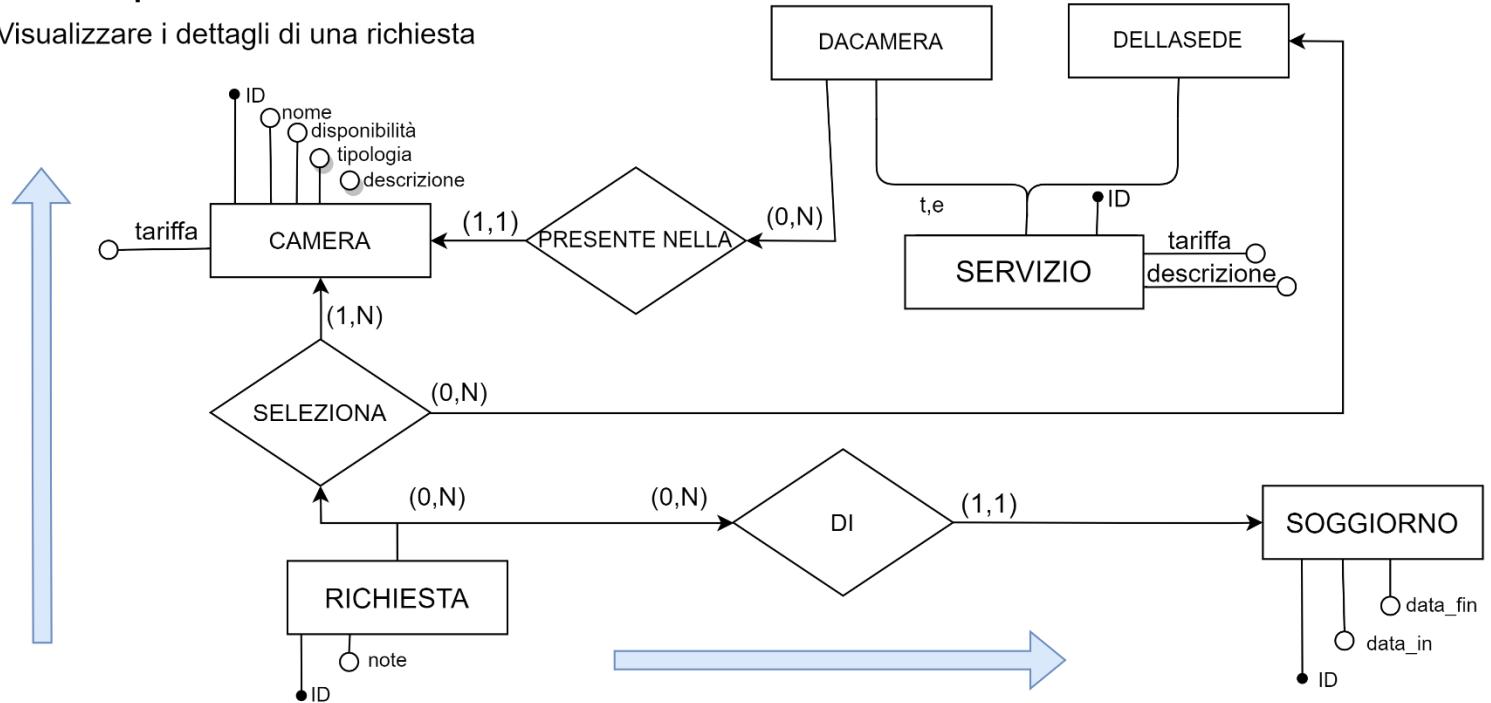
Operazione O10 : Aggiungere punti fedeltà nella tessera di un cliente

Operazione O11 : Visualizzare punti nella tessera di un cliente



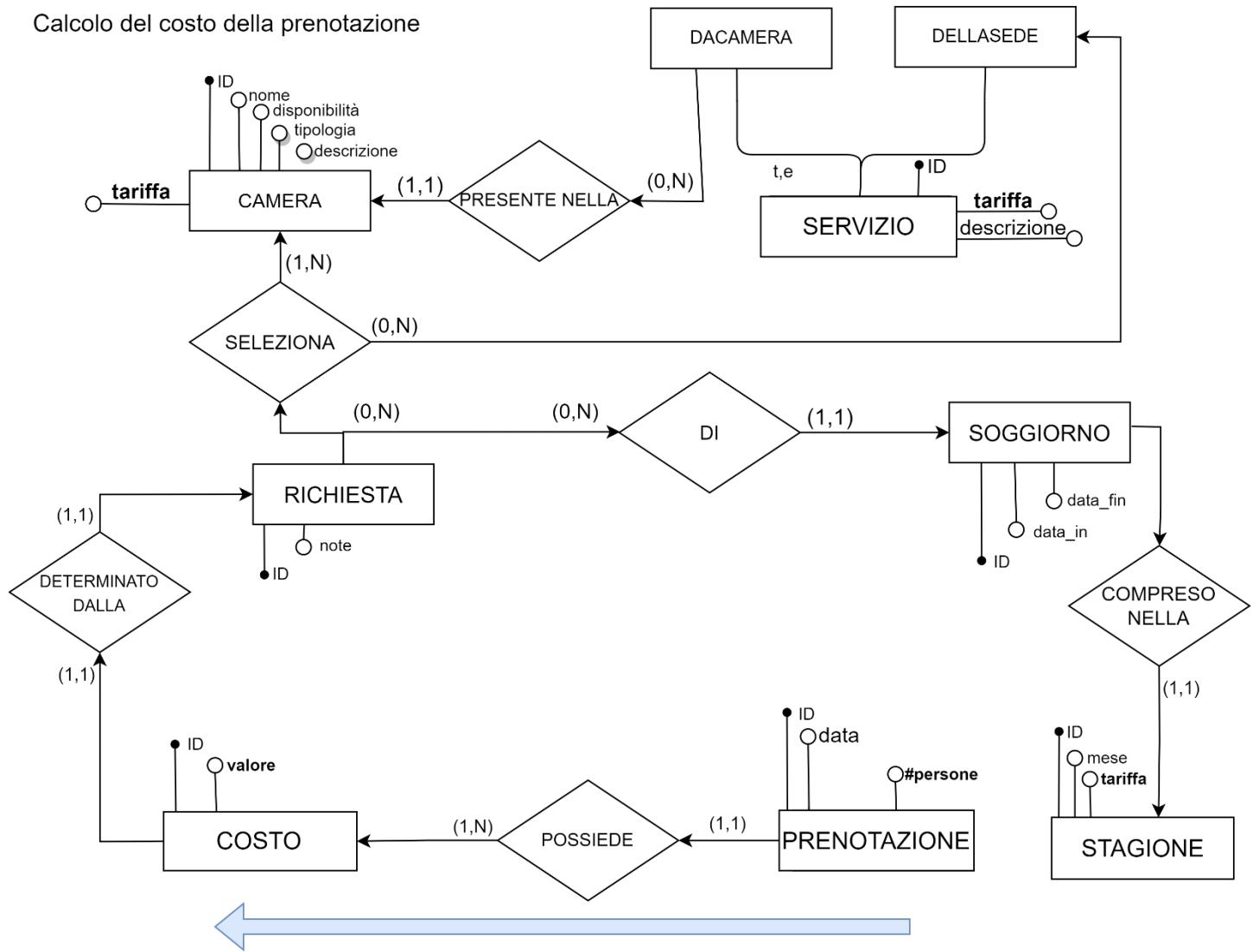
Schema operazioni O7

Visualizzare i dettagli di una richiesta



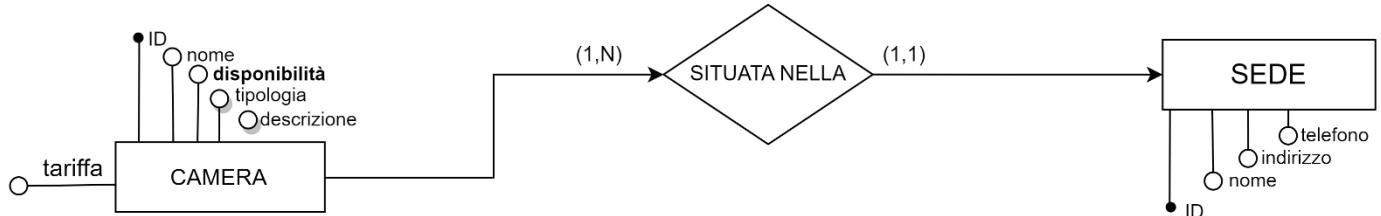
Schema operazioni O5

Calcolo del costo della prenotazione



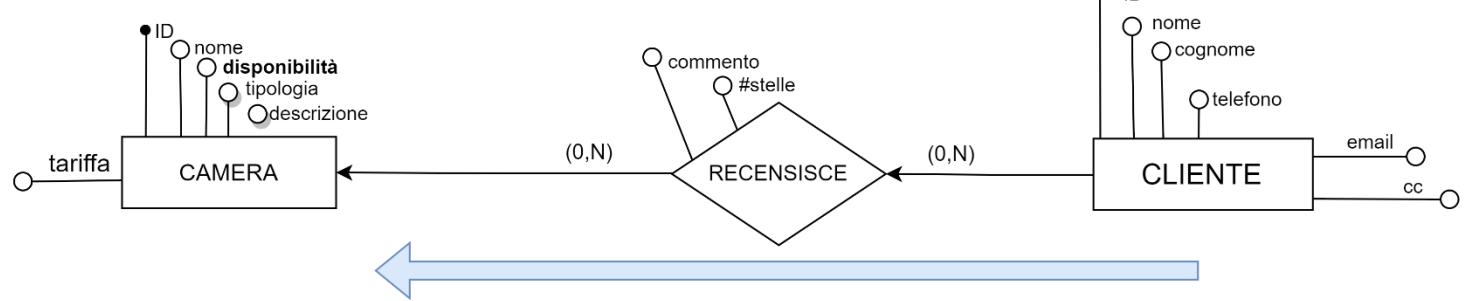
Schema operazioni O9

Visualizzare disponibilità di una camera



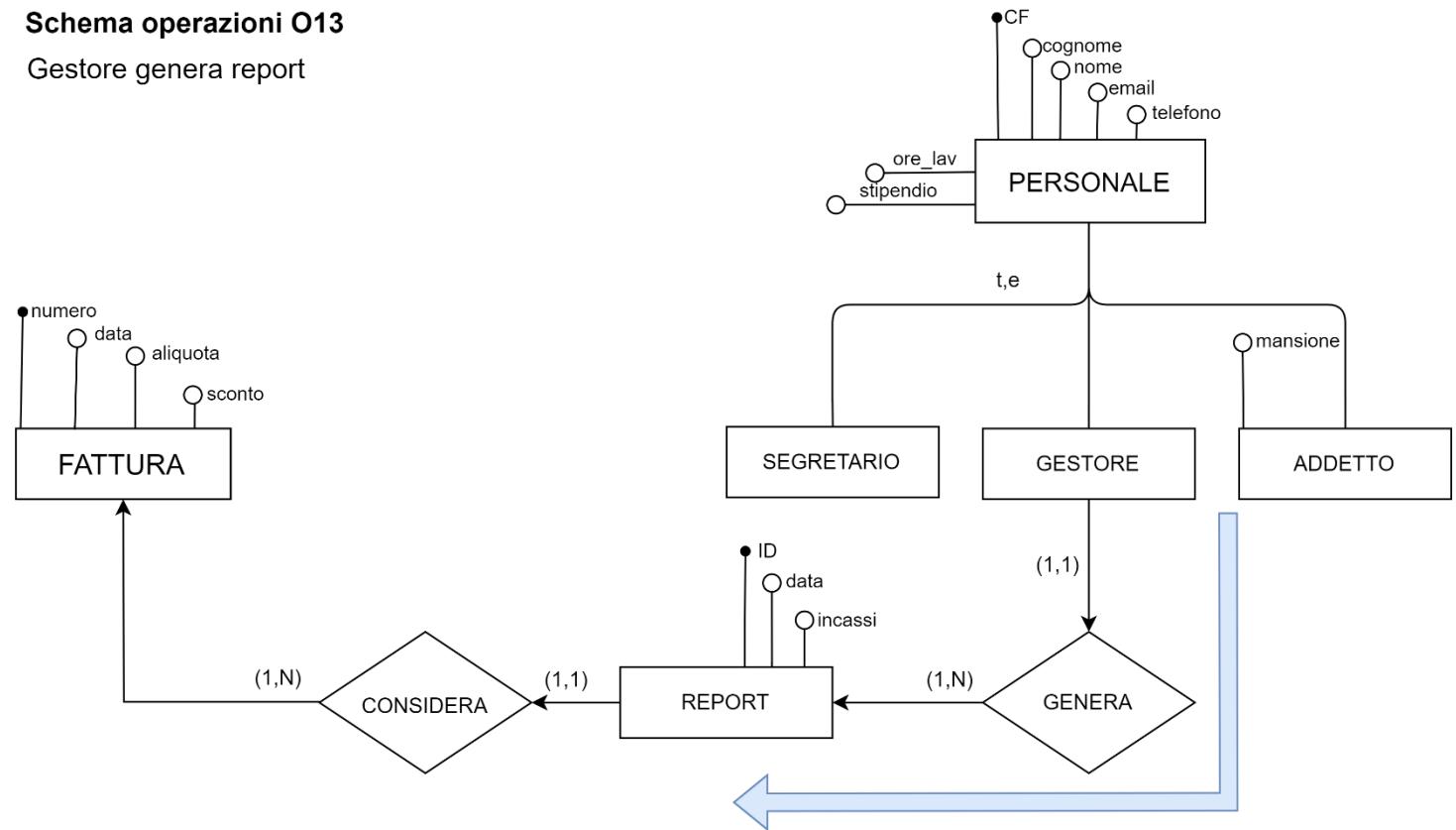
Schema operazioni O12

Visualizzare recensione camera



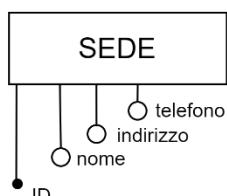
Schema operazioni O13

Gestore genera report



Schema operazioni O14

Aggiunta di una nuova sede



Analisi delle ridondanze

In questa fase viene valutata l'introduzione le ridondanze all'interno dello schema.

Le **ridondanze** presenti sono:

- In CAMERA:
 - La **disponibilità** che può essere ottenuta da **soggiorno**, verificando se ci sono **prenotazioni** per quella camera e quei giorni. (presente nell'operazione 9 considerata precedentemente svolta, come ipotizzato, 600 volte al giorno)
- In COSTO:
 - Il **valore** può essere determinato dalla tariffa della **camera**, **servizi**, **stagione** moltiplicata per i giorni del **soggiorno** e il numero di persone contenuto nella **prenotazione**. (presente nell'operazione 5 considerata precedentemente svolta, come ipotizzato, 250 volte al giorno)
- In FATTURA:
 - Lo **sconto** può essere ricavato ogni volta, se necessario, in base ai punti della **Tessera** sul valore del Costo o semplicemente dal **Pagamento** sottraendo al costo il valore pagato.
- In REPORT:
 - Gli **incassi** possono essere ricavati ogni volta dal valore del Costo meno l'aliquota e lo sconto applicato nella Fattura o dagli importi dei **Pagamenti**. (presente nell'operazione 13 considerata precedentemente svolta, come ipotizzato, 1 volta al mese)

Essendo noto dall'analisi della realtà che abbiamo ipotizzato:

- Le camere sono presenti in quantità non superiore a 40,
- I costi in quantità 25.000,
- Le Fatture in 20.000 quantità,
- I Report in 50 quantità.

Lo spreco di memoria quindi sarà:

tabella	Spreco di memoria
Camera	$40 \times 1\text{Bytes} = 40\text{Bytes} = 0,04\text{ KB}$
Costo	$25.000 \times 4\text{Bytes} = 100.000\text{Bytes} = 100\text{ KB}$
Fattura	$20.000 \times 4\text{Bytes} = 80.000\text{Bytes} = 80\text{ KB}$
Report	$50 \times 4\text{Bytes} = 20.000\text{Bytes} = 20\text{KB}$
totale	$200,04\text{ KB}$

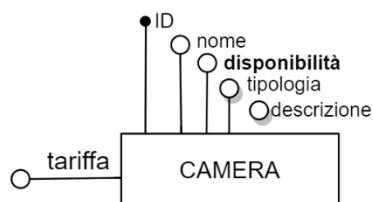
Inoltre, sono anche presenti **associazioni ridondanti**, cioè associazioni derivabili dalla composizione di altre associazioni (presenza di cicli):

- “Appartiene” tra Richiesta e Cliente quando Cliente è già collegato a Prenotazioni tramite l'associazione “Effettua” e Prenotazione con Richiesta tramite l'associazione “Soddisfa”;
- “Possiede”, associazione tra Prenotazione e Costo quando è già presente la relazione “Determinato da” tra Costo e Richiesta;
- L'associazione “Si riferisce alla” tra Recapito e Sede quando sia e Sede che Recapito fanno riferimento allo stesso Personale collegati rispettivamente con le associazioni “Dispone” e “Gestita da”.

Schema operazioni O9

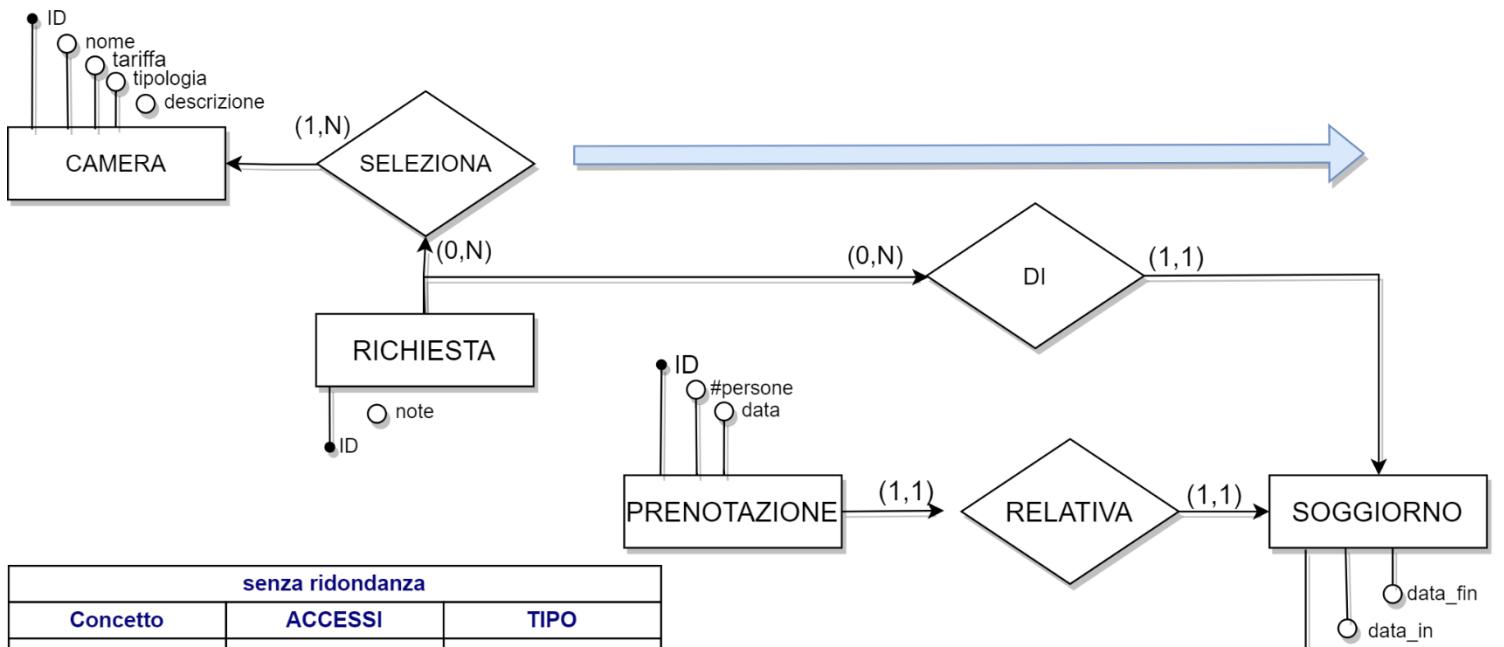
Visualizzare disponibilità di una camera : l'attributo disponibilità è una ridondanza poichè è un valore derivabile da altri, in particolare verificando se ci sono prenotazioni per quella camera e quei giorni. (600 volte al giorno)

In presenza di ridondanza



In presenza di Ridondanza		
Concetto	ACCESSI	TIPO
Camera	1	L

Senza ridondanza



senza ridondanza

Concetto	ACCESSI	TIPO
Camera	1	L
Selezione	1	L
Richiesta	1	L
DI	1	L
Soggiorno	1	L
Prenotazione	1	L

In presenza di ridondanza ridondanza :

$$1L \times 600 = 600L \text{ al giorno}$$

Senza ridondanza :

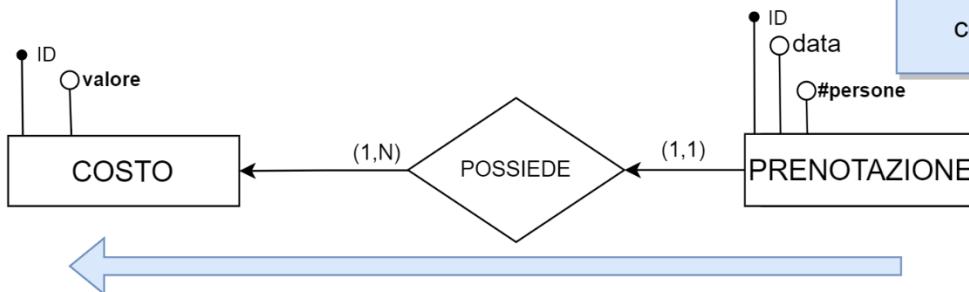
$$6L \times 600 = 3.600L \text{ al giorno}$$

conviene lasciare la ridondanza

Schema operazioni O5

Calcolo del costo della prenotazione. (250 volte al giorno)

In presenza di ridondanza



In presenza di ridondanza ridondanza :

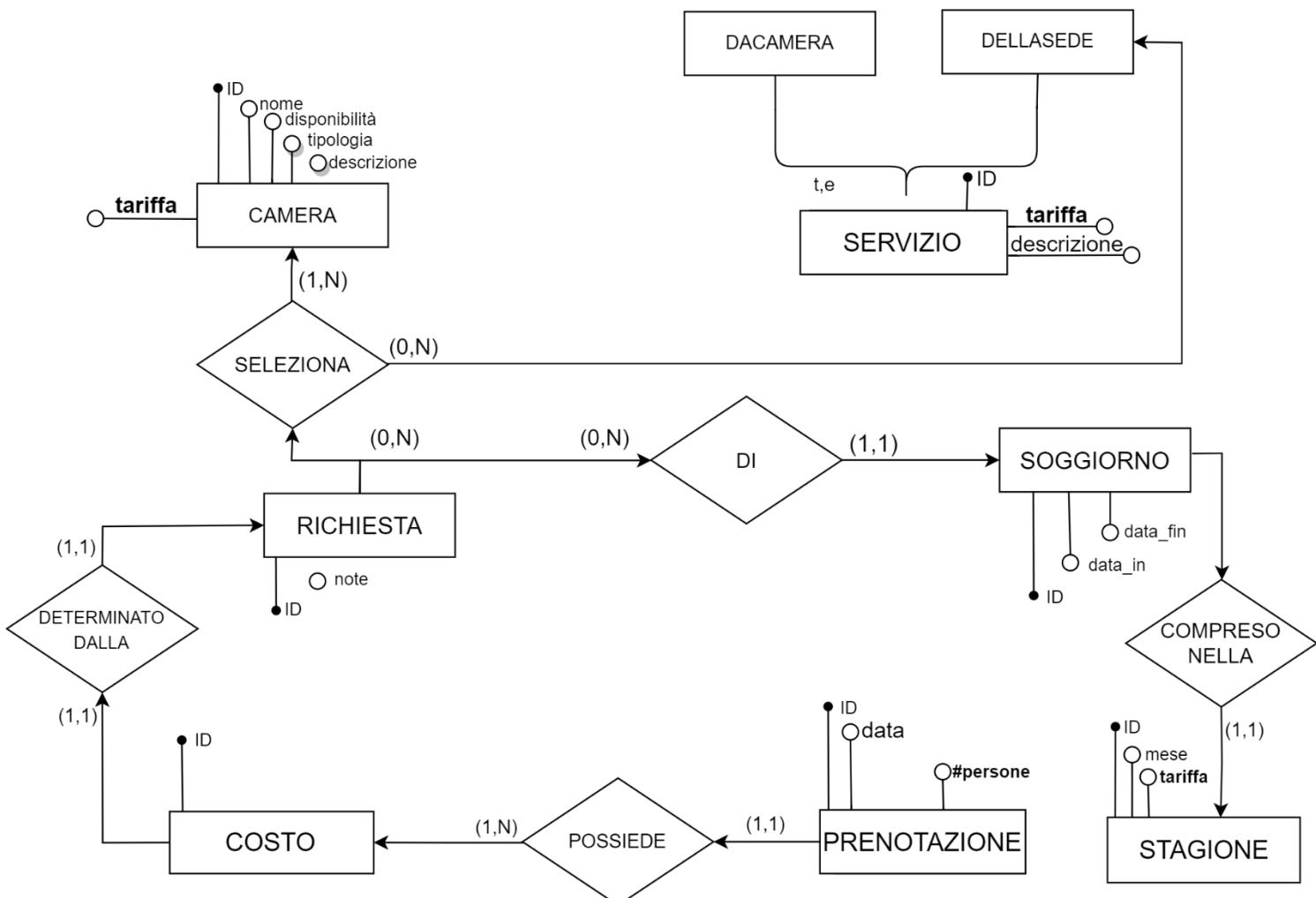
$3L \times 250 = 750L$ al giorno

Senza ridondanza :

$14L \times 250 = 3.500L$ al giorno

conviene lasciare la ridondanza

Senza ridondanza

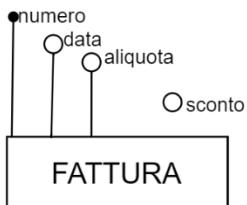


senza ridondanza

Concetto	ACCESSI	TIPO
In presenza di Ridondanza		
Concetto	ACCESSI	TIPO
Prenotazione	1	L
Possiede	1	L
Costo	1	L
senza ridondanza		
Concetto	ACCESSI	TIPO
Prenotazione	1	L
Possiede	1	L
Costo	1	L
Determinato dalla	1	L
Richesta	1	L
Selezione	3	L
Camera	1	L
Servizio Della Sede	1	L
Di	1	L
Soggiorno	1	L
Compreso nella	1	L
Stagione	1	L

Attributo ridondante sconto nell'entità Fattura

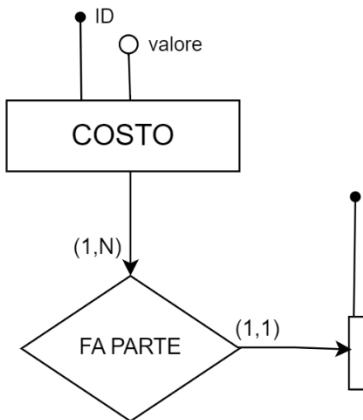
In presenza di ridondanza



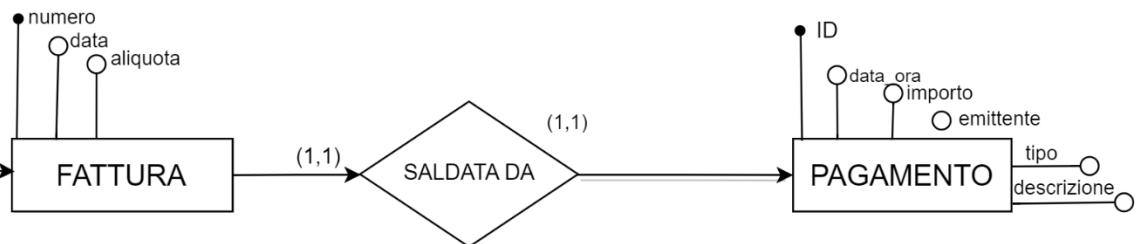
In presenza di Ridondanza

Concetto	ACCESSI	TIPO
Fattura	1	L

Senza ridondanza



senza ridondanza		
ACCESSI	TIPO	Concetto
Fattura	2	L
Fa parte	2	L
Costo	1	L
Saldata da	1	L
Pagamento	1	L



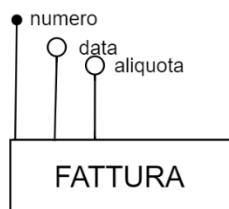
In presenza di ridondanza ridondanza :

1L

Senza ridondanza :

7L

conviene eliminare la ridondanza

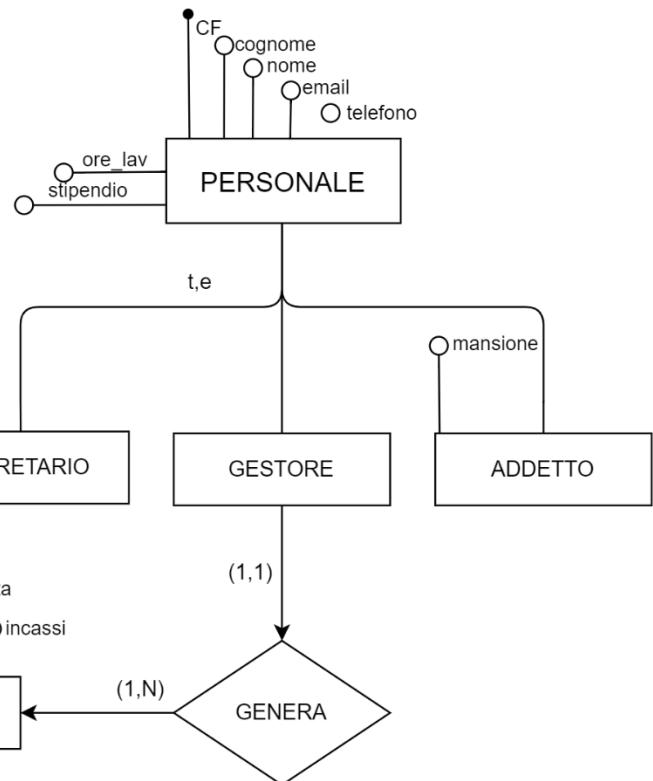


Schema operazioni O13

Gestore genera report.(1 volta al mese)

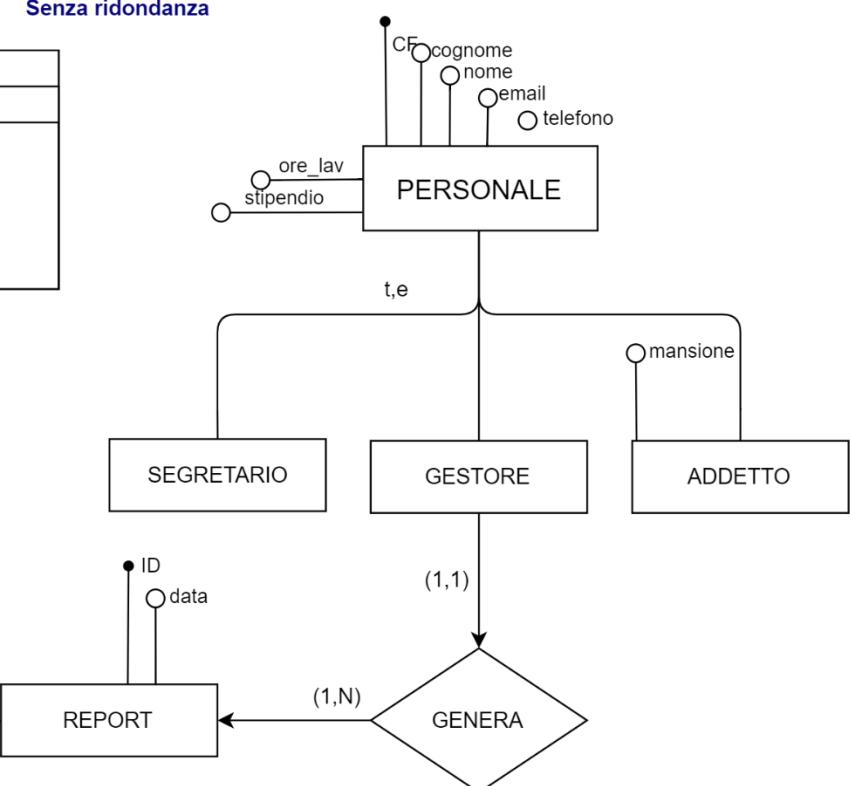
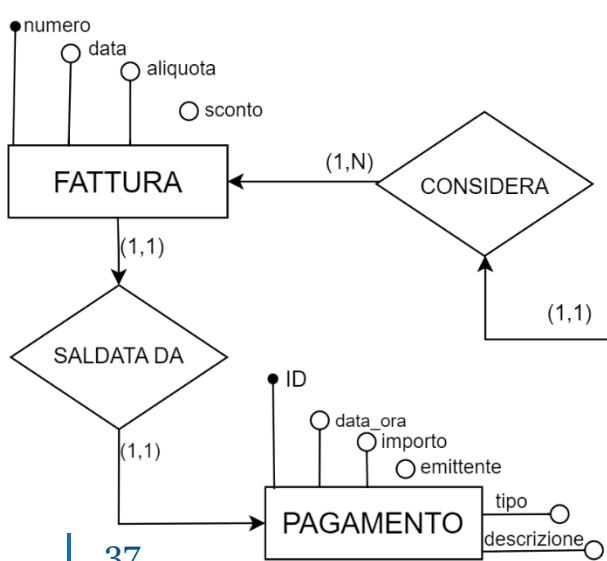
In presenza di ridondanza

In presenza di Ridondanza		
Concetto	ACCESSI	TIPO
Gestore Genera Report	1 1 1	L L L



Senza ridondanza

senza ridondanza		
Concetto	ACCESSI	TIPO
Gestore Genera Report	1 1 1	L L L
Fattura	30	L
Saldata da Pagamento	30 30	L L

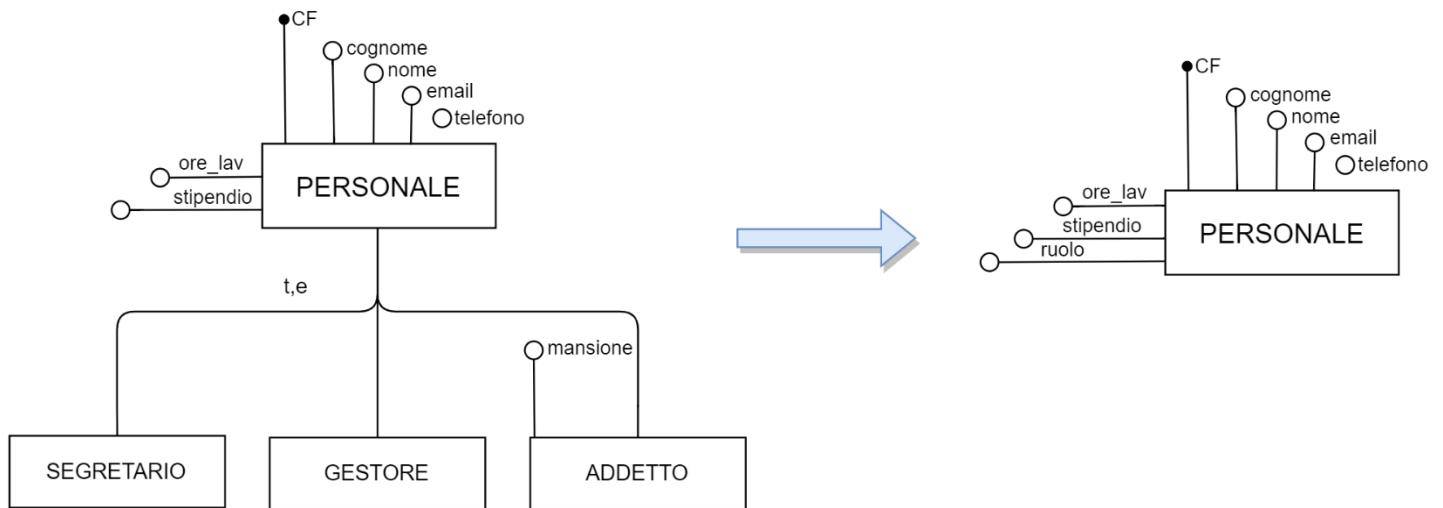


Eliminazione delle gerarchie

All'interno dello schema E-R sono presenti due gerarchie totali ed esclusive.

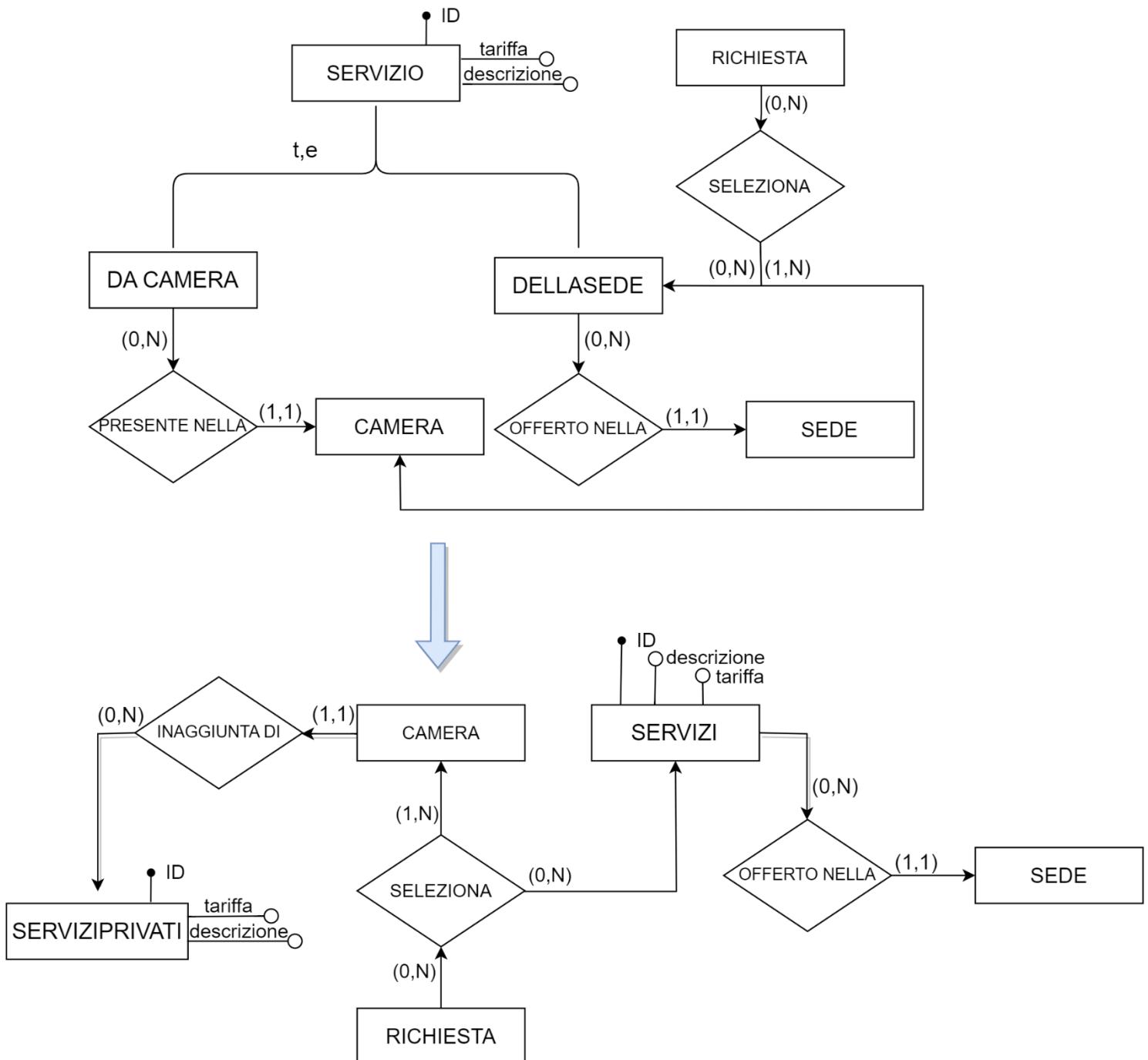
La prima tratta dell'entità **Personale** che presenta tre entità figlie: Segretario, Gestore e Addetto. Effettuare un collasso verso il basso avrebbe poco senso poiché i miglioramenti all'efficienza della base di dati non sarebbero significativi, inoltre, allo stato attuale non sarebbe giustificato un aumento dei collegamenti.

Pertanto, si ritiene opportuno effettuare un **collasso verso l'alto**, aggiungendo un **attributo "ruolo"** all'entità **Personale**.



Per l'entità Servizio invece, visto le diverse partecipazioni nelle associazioni delle due entità figlie, si predilige un **collasso verso il basso**, applicabile poiché la copertura è totale ed esclusiva.

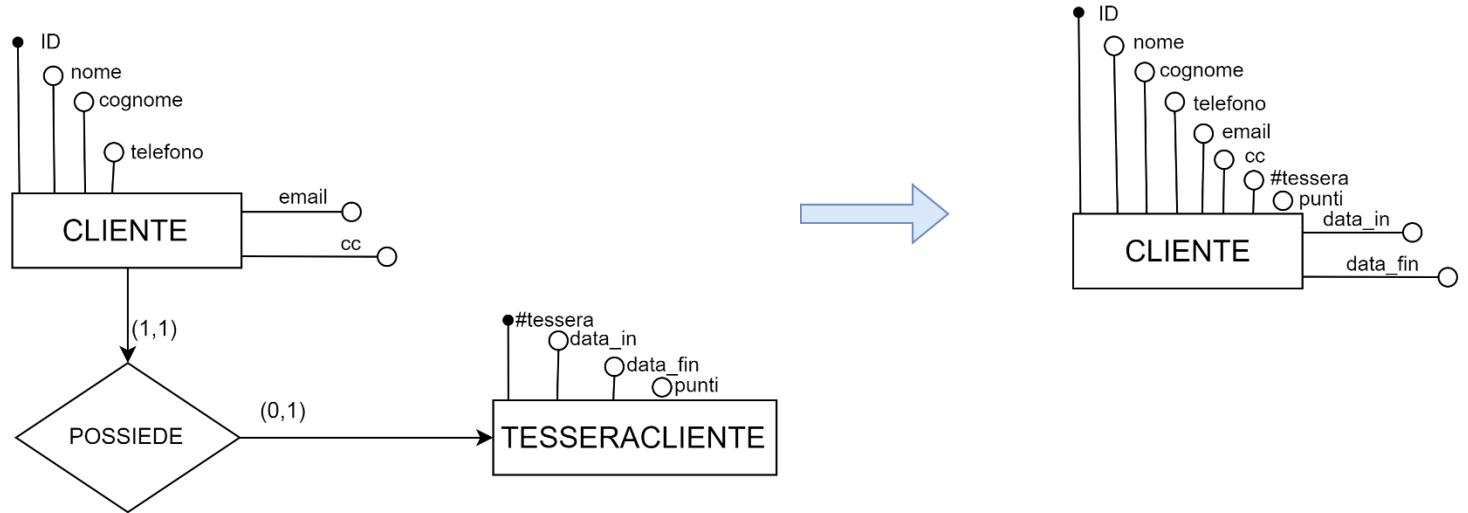
In questo caso le entità figlie saranno due entità diverse: **ServiziPrivati**, per quando riguarda i servizi presenti in una camera e prenotati insieme ad essa, e **Servizi**, i servizi in generale offerti dalla sede e selezionabili nella richiesta per una prenotazione.



Accorpamento delle entità

Cliente ha Tessera Cliente

Si considera la relazione 1 a 1 tra le entità Cliente e Tessera Cliente. Si ritiene opportuno effettuare un accorpamento dell'entità TesseraCliente ai fini di ridurre la complessità spaziale delle entità.

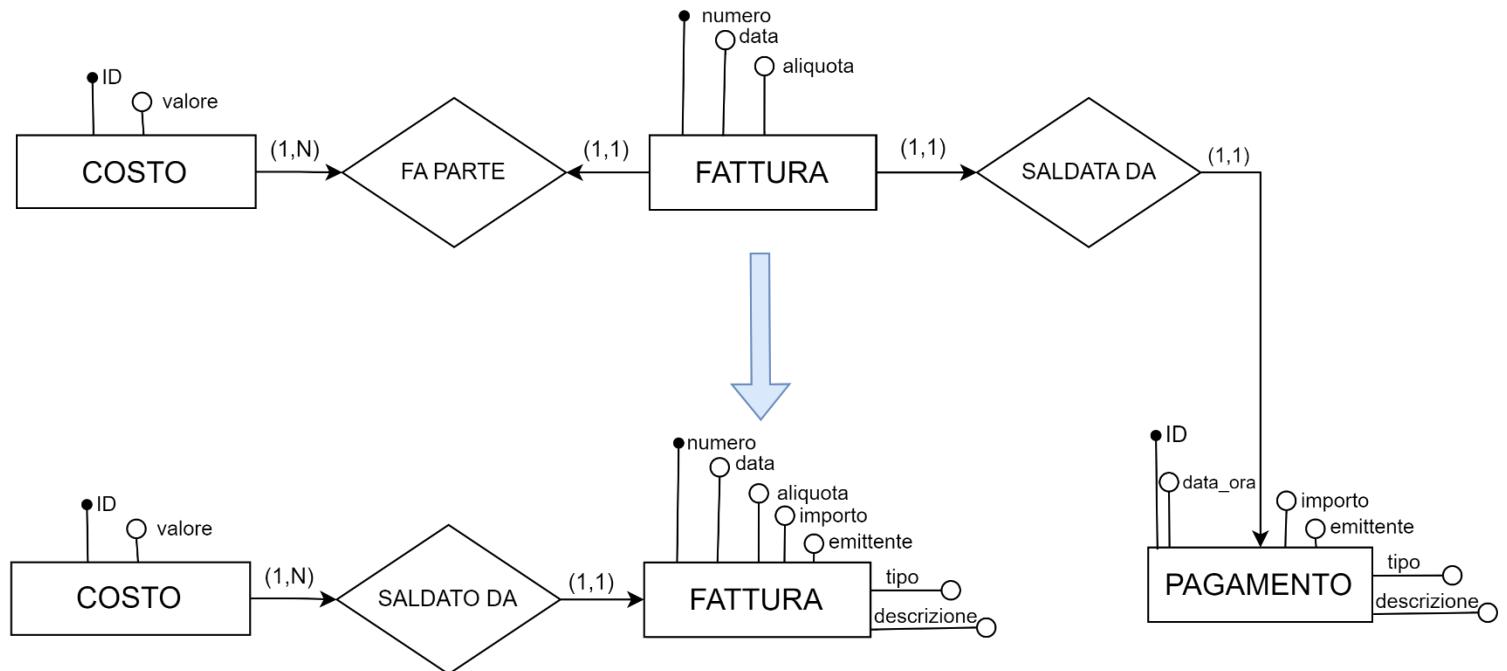


Costo fa parte di Fattura

La relazione non può essere semplificata poichè una fattura può contenere più costi.

Fattura saldata da Pagamento:

Considerano questa relazione 1 a 1 e che entrambe le entità sono entità importanti per motivi finanziari, quindi consultate insieme, si ritiene opportuno l'accorpamento delle entità.



Prenotazione possiede Costo

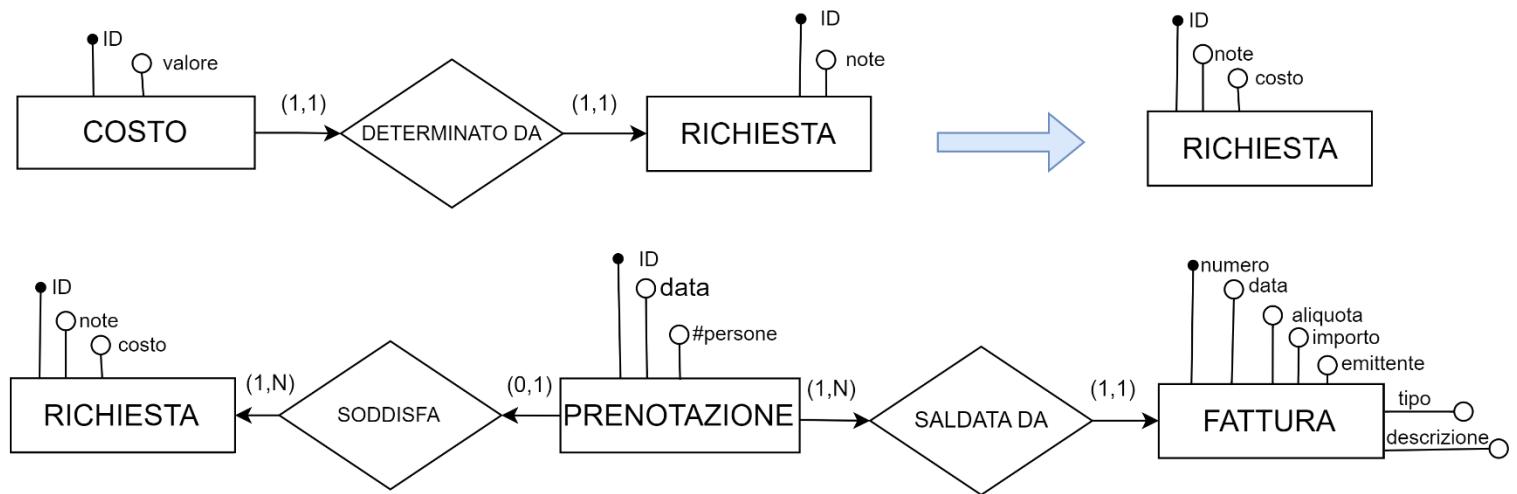
La relazione non può essere semplificata poichè una Prenotazione può contenere più Costi (un costo per ogni richiesta).

Prenotazione soddisfa Richiesta

La relazione non può essere semplificata poichè una Prenotazione può soddisfare più Richieste del cliente.

Costo determinato da Richiesta

Considerano questa relazione 1 a 1 la relazione può essere semplificata.



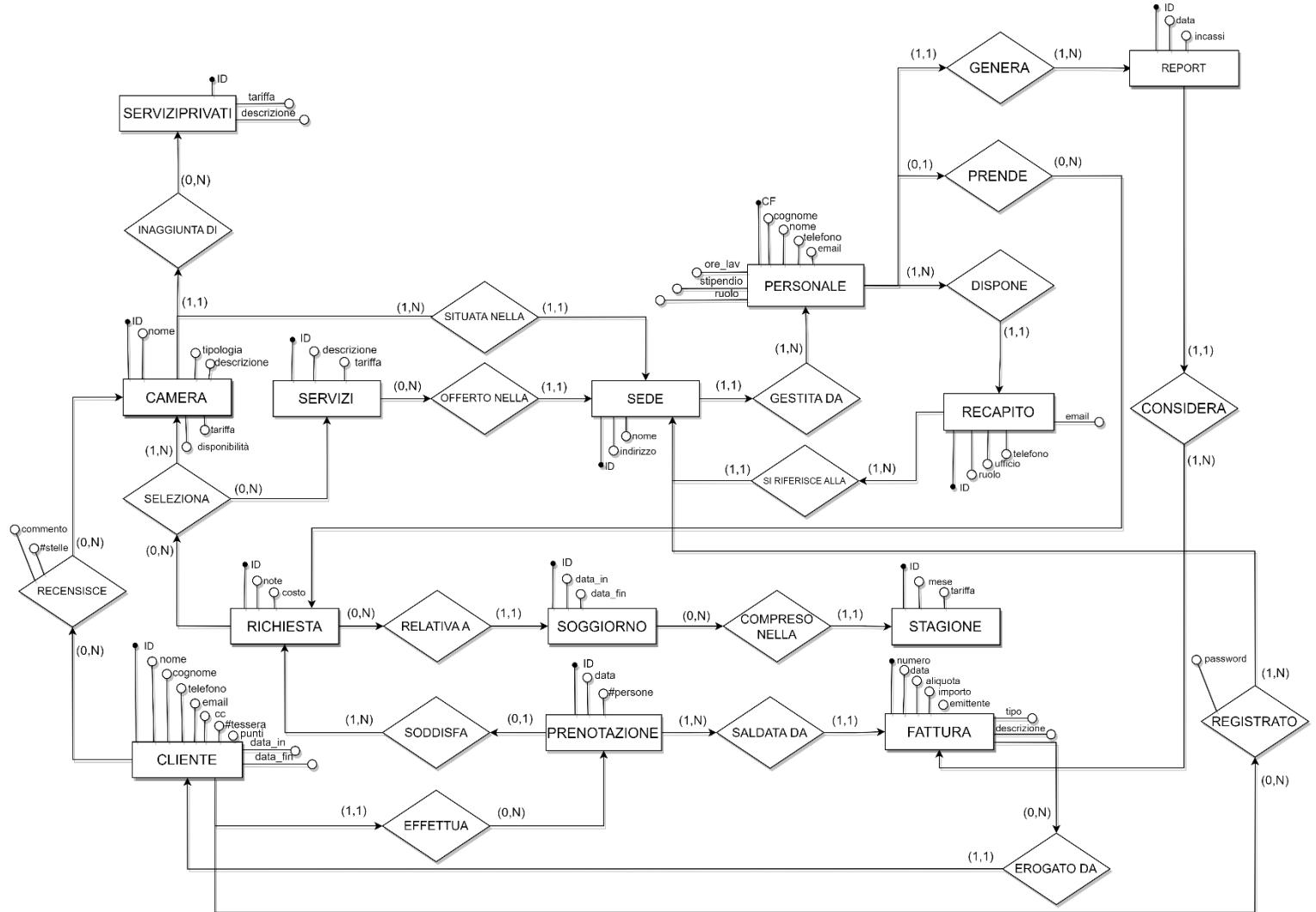
Scelta degli identificatori principali

Le entità presentano già i loro principali identificatori. Nello specifico, la quasi totalità delle entità presentano un identificativo denominato come l'entità a cui appartiene (**id_nomeEntità**), ad esempio, Cliente ha come identificativo `id_cliente`.

ENTITA'	Identificatore
Cliente	<code>Id</code>
TesseraCliente	<code>Id_tessera</code>
Pagamento	<code>Id_pagamento</code>
Fattura	<code>numero</code>
Costo	<code>Id_costo</code>
Prenotazione	<code>Id_prenotazione</code>
Soggiorno	<code>Id_soggiorno</code>
Richiesta	<code>Id_richiesta</code>
Camera	<code>Id_camera</code>
Servizio	<code>Id_servizio</code>
ServizioPrivato	<code>Id_servizio</code>
Stagione	<code>Id_stagione</code>
Sede	<code>Id_sede</code>
Recapito	<code>Id_recapito</code>
Personale	<code>CF</code>
Report	<code>Id_report</code>

Schema E-R finale ristrutturato

Il risultato delle precedenti considerazioni è il seguente schema E-R ristrutturato, finale.



Verso il modello relazionale

Avendo completato la fase di ristrutturazione, si procede a generare il modello logico equivalente.

Traduzione delle associazioni **molti a molti**:

- **Recensisce:** La relazione "Recensisce", posta tra le entità "Cliente" e "Camera", viene tradotta in una nuova entità, denominata "Recensioni". Tale entità viene identificata da un id, ottiene inoltre due altri attributi "commento" e "num_stelle" e farà riferimento a "Cliente" e "Camera".
- **Registrato:** La relazione tra "Cliente" e "Sede", che vedeva coinvolti prima della ristrutturazione dello schema le due entità "TesseraCliente" e "Sede", sarà trasformata in una nuova entità denominata "Registrazione". Tale entità verrà identificata tramite un id, avrà un attributo "password" e farà riferimento alle due entità "Cliente" e "Sede".
- **Seleziona:** La relazione tra "Richiesta" e "Camera" e tra "Richiesta" e "Servizi", sarà trasformata in una nuova entità denominata "Selezione", che farà riferimento alle 2 entità coinvolte rispettivamente.

Traduzione delle associazioni **uno a molti**:

Tutte le traduzioni delle associazioni uno a molti verranno effettuate inglobando un riferimento dell'entità che partecipa con molteplicità 1 nell'entità con molteplicità n nell'associazione.

- **Effettua**, la relazione tra le entità "Cliente" e "Prenotazione" viene inglobata nell'entità Prenotazione, la quale acquisisce quindi un attributo in riferimento all'entità Cliente.
- **Erogato da**, la relazione tra le entità "Cliente" e "Fattura" viene inglobata nell'entità Fattura, la quale acquisisce quindi un attributo in riferimento all'entità Cliente.
- **Soddisfa**, la relazione tra le entità "Richiesta" e "Prenotazione" viene inglobata nell'entità Richiesta, la quale acquisisce quindi un attributo in riferimento all'entità Prenotazione.
- **Relativa a**, la relazione tra le entità "Camera" e "Sede" viene inglobata nell'entità Camera, la quale acquisisce quindi un attributo in riferimento all'entità Sede.
- **Compreso nella**, la relazione tra le entità "Soggiorno" e "Stagione" viene inglobata nell'entità Soggiorno, la quale acquisisce quindi un attributo in riferimento all'entità Stagione.
- **In aggiunta di**, la relazione tra le entità "Camera" e "ServiziPrivati" viene inglobata nell'entità ServiziPrivati, la quale acquisisce quindi un attributo in riferimento all'entità Camera.
- **Situata nella**, la relazione tra le entità "Camera" e "Sede" viene inglobata nell'entità Camera, la quale acquisisce quindi un attributo in riferimento all'entità Sede.
- **Offerto nella**, la relazione tra le entità "Servizi" e "Sede" viene inglobata nell'entità Servizi, la quale acquisisce quindi un attributo in riferimento all'entità Sede.
- **Si riferisce alla**, la relazione tra le entità "Recapito" e "Sede" viene inglobata nell'entità Recapito, la quale acquisisce quindi un attributo in riferimento all'entità Sede.
- **Gestita da**, la relazione tra le entità "Sede" e "Personale" viene inglobata nell'entità Personale, la quale acquisisce quindi un attributo in riferimento all'entità Sede.
- **Dispone**, la relazione tra le entità "Personale" e "Recapito" viene inglobata nell'entità Personale, la quale acquisisce quindi un attributo in riferimento all'entità Recapito.
- **Prende**, la relazione tra le entità "Personale" e "Richiesta" viene inglobata nell'entità , la quale acquisisce quindi un attributo in riferimento all'entità .

- **Genera**, la relazione tra le entità “Personale” e “Report” viene inglobata nell’entità Report, la quale acquisisce quindi un attributo in riferimento all’entità Personale.
- **Considera**, la relazione tra le entità “Report” e “Fattura” viene inglobata nell’entità Fattura, la quale acquisisce quindi un attributo in riferimento all’entità Report.

Traduzione delle associazioni **uno a uno**:

In questa associazione il riferimento può essere contenuto indifferentemente da una qualsiasi entità partecipante, per cui la scelta è puramente soggettiva.

- **Saldata da**, la relazione tra le entità “Prenotazione” e “Fattura” viene inglobata nell’entità Prenotazione, la quale acquisisce quindi un attributo in riferimento all’entità Fattura.

Schema logico

Cliente (id_cliente, nome, cognome, telefono, mail, cc, num_tessera, punti, data_in, data_fine)

Recensione (id_cliente, id_camera, num_stelle, commento)

Registrazione (id_cliente, id_sede, password)

Prenotazione (id_prenotazione, data, num_persone, id_cliente, id_fattura)

Richiesta (id_richiesta, note, costo, id_prenotazione, id_soggiorno, id_personale)

Selezione (id_selezione, id_richiesta, id_camera, id_servizio)

Soggiorno (id_soggiorno, data_in, data_fin, id_stagione)

Stagione (id_stagione, mese, tariffa)

Fattura (numero, data, aliquota, importo, emittente, tipo, descrizione, id_cliente, id_prenotazione, id_report)

Camera (id_camera, nome, tipologia, descrizione, tariffa, disponibilità, id_sede)

ServizioPrivato (id_servizio, tariffa, descrizione, id_camera)

Servizio (id_servizio, descrizione, tariffa, id_sede)

Sede (id_sede, indirizzo, nome)

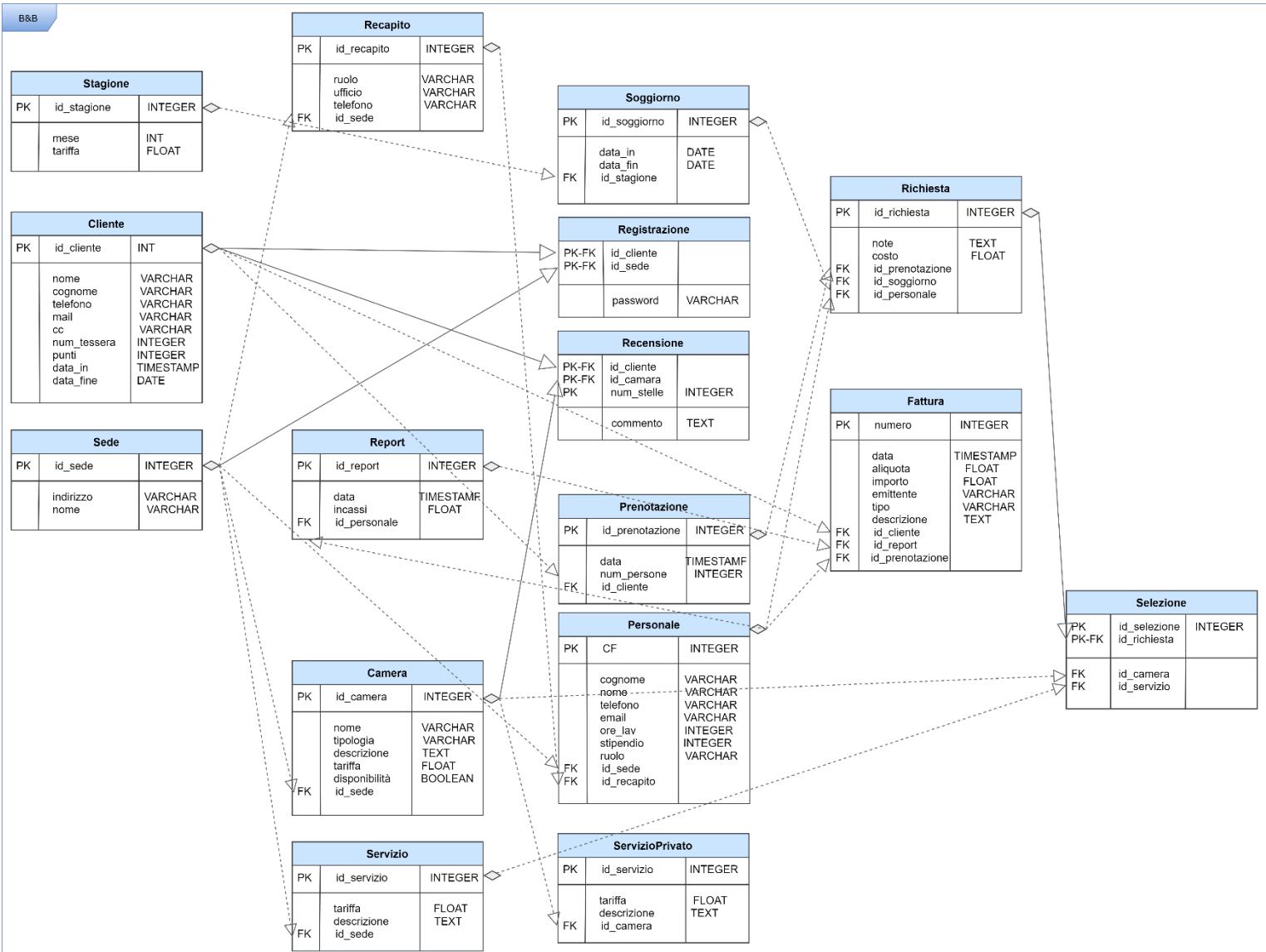
Personale (CF, cognome, nome, telefono, email, ore_lav, stipendio, ruolo, id_sede, id_recapito)

Recapito (id_recapito, ruolo, ufficio, telefono, id_sede)

Report (id_report, data, incassi, id_personale)

Note: la chiave primaria è sottolineata con una riga continua, le chiavi esterne sono tratteggiate con un tratto tratteggiato.

Diagramma UML





4. Progettazione Fisica



La progettazione fisica prende come input lo schema logico definito nella fase della progettazione logica e lo completa con la specifica dei parametri di memorizzazione dei dati, che riguardano principalmente l'organizzazione dei file e degli indici. Il **modello fisico** dipende fortemente dal DBMS utilizzato ma può essere diverso anche all'interno di DBMS della stessa famiglia. In questo caso verrà utilizzato il **DBMS MySQL**.

A questo documento vengono allegati i file di definizione della base di dati, inseriti all'interno della cartella “`src/sql\ operations/`”.

Premessa: traduzione linguistica

Allo scopo di incrementare la manutenibilità della base di dati, lo schema logico ottenuto precedentemente verrà tradotto in lingua inglese come segue:

Customer (customer_id, first_name, last_name, phone, email, banking count, num_card, points, start_date, end_date)

Review (customer_id, room_id, num_stars, comment)

Signup (customer_id, location_id, password)

Reservation (reservation_id, date, #people, customer_id)

Request (request_id, notes, cost, reservation_id, sojourn_id, staff_id, id_season)

Selection (id_selection, request_id, room_id, service_id)

Sojourn (id_sojourn, date_in, date_fin)

Season (id_season, month, rate)

Invoice (number, date, rate, amount, issuer, type, description, customer_id, report_id, reservation_id)

Room (room_id, name, type, description, rate, availability, location_id)

PrivateService (service_id, rate, description, room_id)

Service (service_id, description, rate, location_id)

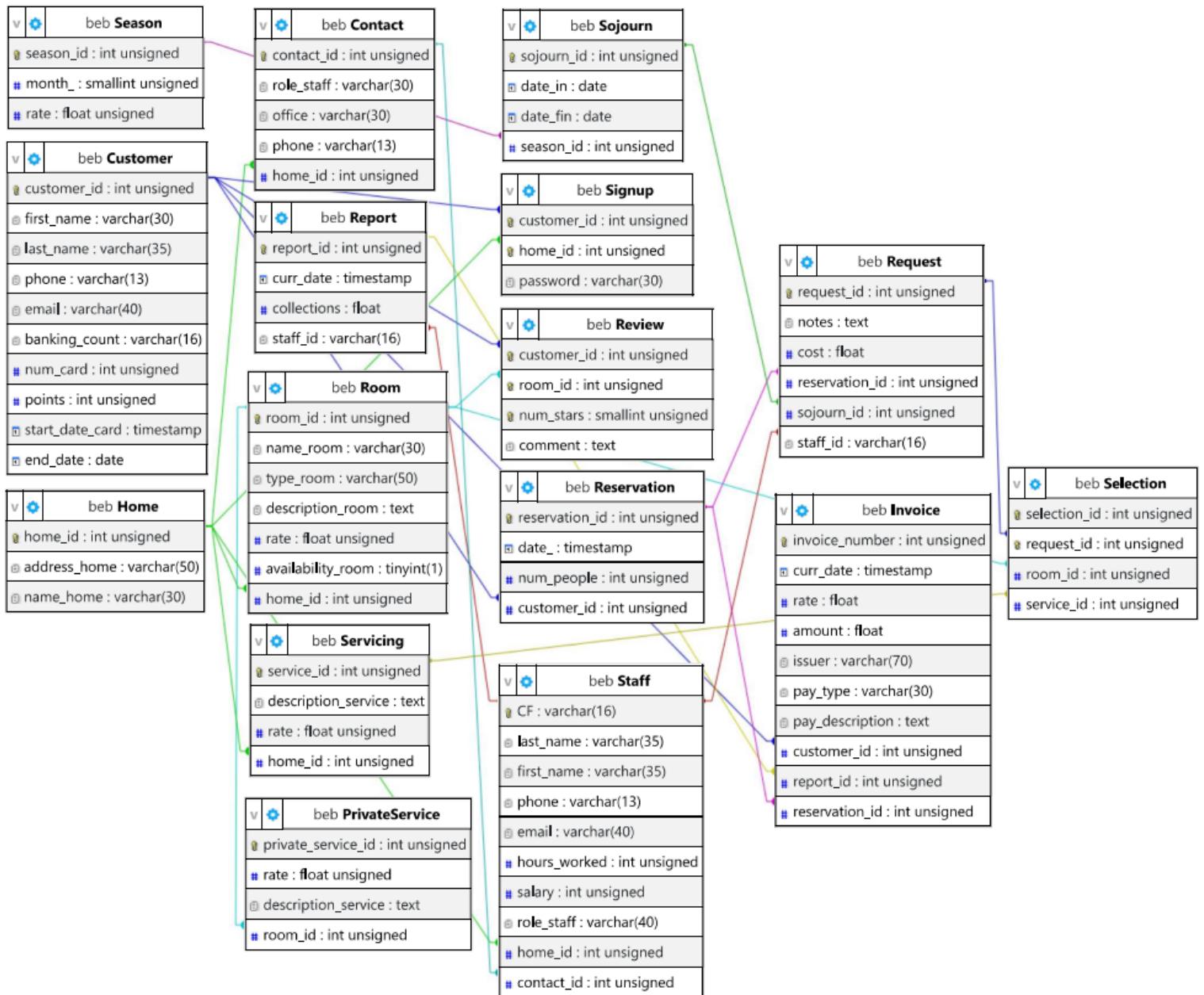
Location (location_id, address, name)

Staff (CF, last_name, first_name, phone, email, hours_worked, salary, role, location_id, id_recapitol)

Contact (contact_id, role, office, phone, location_id)

Report (id_report, date, collections, staff_id)

La gestione di un Bed and Breakfast | Alessia Maccarrone



Definizione delle tabelle

La definizione delle tabelle è contenuta all'interno del file **tables.sql**.

```
CREATE DATABASE IF NOT EXISTS beb;
USE beb;

CREATE TABLE IF NOT EXISTS Season (
    /*Stagione (id_stagione, mese, tariffa)
    Season (season_id, month, rate) */
    season_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    month_ SMALLINT UNSIGNED,
    rate FLOAT UNSIGNED
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Customer (
    /*Cliente (id_cliente, nome, cognome, telefono, mail, cc, num_tessera, punti,
    data_in, data_fine)
    Customer (customer_id, first_name, last_name, phone, email, banking_count,
    num_card, pointERS, start_date, end_date)*/
    customer_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(30),
    last_name VARCHAR(35) NOT NULL,
    phone VARCHAR(13) NOT NULL,
    email VARCHAR(40) NOT NULL,
    banking_count VARCHAR(16),
    num_card INTEGER UNSIGNED,
    points INTEGER UNSIGNED,
    start_date_card TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    end_date DATE,
    PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Home (
    /*Sede (id_sede, indirizzo, nome)
    Home (home_id, address, name) */
    home_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    address_home VARCHAR(50),
    name_home VARCHAR(30),
    PRIMARY KEY (`home_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS Contact (
    /*Recapito (id_recapito, ruolo, ufficio, telefono, id_sede)
    Contact (contact_id, role_staff, office, phone, home_id) */
    contact_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    role_staff VARCHAR(30),
    office VARCHAR(30),
    phone VARCHAR(13) NOT NULL,
    home_id INTEGER UNSIGNED,
    FOREIGN KEY (`home_id`) REFERENCES Home(`home_id`),
    PRIMARY KEY (`contact_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Room (
    /*Camera (id_camera, nome, tipologia, descrizione, tariffa, disponibilità,
    id_sede)
    Room (room_id, name, type, description, rate, availability, home_id) */
    room_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    name_room VARCHAR(30),
    type_room VARCHAR(50),
    description_room TEXT,
    rate FLOAT UNSIGNED NOT NULL,
    availability_room BOOLEAN NOT NULL,
    home_id INTEGER UNSIGNED,
    FOREIGN KEY (`home_id`) REFERENCES Home(`home_id`),
    PRIMARY KEY (`room_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Servicing (
    /*Servizio (id_servizio, descrizione, tariffa, id_sede)
    Service (service_id, description, rate, home_id) */
    service_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    description_service TEXT,
    rate FLOAT UNSIGNED NOT NULL,
    home_id INTEGER UNSIGNED,
    FOREIGN KEY (`home_id`) REFERENCES Home(`home_id`),
    PRIMARY KEY (`service_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS PrivateService (
    /*ServizioPrivato (id_servizio, tariffa, descrizione, id_camera)
    PrivateService (service_id, rate, description, room_id) */
    private_service_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    rate FLOAT UNSIGNED,
    description_service TEXT,
    room_id INTEGER UNSIGNED,
    FOREIGN KEY (`room_id`) REFERENCES Room(`room_id`),
    PRIMARY KEY (`private_service_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS Staff (
    /*Personale (CF, cognome, nome, telefono, email, ore_lav, stipendio, ruolo,
id_sede, id_recapito)
    Staff (CF, last_name, first_name, phone, email, hours_worked, salary,
role_staff, home_id, contact_id) */
    CF VARCHAR(16) NOT NULL UNIQUE,
    last_name VARCHAR(35),
    first_name VARCHAR(35),
    phone VARCHAR(13),
    email VARCHAR(40),
    hours_worked INTEGER UNSIGNED,
    salary INTEGER UNSIGNED,
    role_staff VARCHAR(40),
    home_id INTEGER UNSIGNED,
    contact_id INTEGER UNSIGNED,
    FOREIGN KEY (`home_id`) REFERENCES Home(`home_id`),
    FOREIGN KEY (`contact_id`) REFERENCES Contact(`contact_id`),
    PRIMARY KEY (`CF`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS Report (
    /*Report (report_id, data, incassi, id_personale)
    Report (report_id, curr_date, collections, staff_id) */
    report_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    curr_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    collections FLOAT NOT NULL,
    staff_id VARCHAR(16),
    FOREIGN KEY (`staff_id`) REFERENCES Staff(`CF`),
    PRIMARY KEY (`report_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS Invoice (
    /*Fattura (numero, data, aliquota, importo, emittente, tipo, descrizione,
id_cliente, report_id)
    Invoice (invoice_number, curr_date, rate, amount, issuer, type, description,
customer_id, report_id)*/
    invoice_number INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    curr_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    rate FLOAT,
    amount FLOAT NOT NULL,
    issuer VARCHAR(70),
    pay_type VARCHAR(30),
    pay_description TEXT,
    customer_id INTEGER UNSIGNED,
    report_id INTEGER UNSIGNED,
    FOREIGN KEY (`customer_id`) REFERENCES Customer(`customer_id`),
    FOREIGN KEY (`report_id`) REFERENCES Report(`report_id`),
    PRIMARY KEY (`invoice_number`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE IF NOT EXISTS Reservation (
    /*Prenotazione (id_prenotazione, data, num_persone, id_cliente, id_fattura)
    Reservation (reservation_id, date, num_people, customer_id, invoice_id) */
    reservation_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    date_ TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    num_people INTEGER UNSIGNED,
    customer_id INTEGER UNSIGNED,
    invoice_id INTEGER UNSIGNED,
    FOREIGN KEY (`customer_id`) REFERENCES Customer(`customer_id`),
    FOREIGN KEY (`invoice_id`) REFERENCES Invoice(`invoice_number`),
    PRIMARY KEY (`reservation_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Review (
    /*Recensione (id_cliente, id_camera, num_stelle, commento)
    Review (customer_id, room_id, num_stars, comment)*/
    customer_id INTEGER UNSIGNED,
    room_id INTEGER UNSIGNED,
    FOREIGN KEY (`customer_id`) REFERENCES Customer(`customer_id`),
    FOREIGN KEY (`room_id`) REFERENCES Room(`room_id`),
    num_stars SMALLINT UNSIGNED,
    comment TEXT,
    PRIMARY KEY (`customer_id`, `room_id`, `num_stars`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Signup (
    /*Registrazione (id_cliente, id_sede, password)
    Signup (customer_id, home_id, password) */
    customer_id INTEGER UNSIGNED,
    home_id INTEGER UNSIGNED,
    FOREIGN KEY (`customer_id`) REFERENCES Customer(`customer_id`),
    FOREIGN KEY (`home_id`) REFERENCES Home(`home_id`),
    password VARCHAR(30),
    PRIMARY KEY (`customer_id`, `home_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Sojourn (
    /*Soggiorno (id_soggiorno, data_in, data_fin, id_stagione)
    Sojourn (sojourn_id, date_in, date_fin, id_season) */
    sojourn_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    date_in DATE NOT NULL,
    date_fin DATE NOT NULL,
    season_id INTEGER UNSIGNED,
    FOREIGN KEY (`season_id`) REFERENCES Season(`season_id`),
    PRIMARY KEY (`sojourn_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```

CREATE TABLE IF NOT EXISTS Request (
    /*Richiesta (id_richiesta, note, costo, id_prenotazione, id_soggiorno,
    id_personale)
    Request (request_id, notes, cost, reservation_id, sojourn_id, staff_id) */
    request_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    notes TEXT,
    cost FLOAT,
    reservation_id INTEGER UNSIGNED,
    sojourn_id INTEGER UNSIGNED,
    staff_id VARCHAR(16),
    FOREIGN KEY (`reservation_id`) REFERENCES Reservation(`reservation_id`),
    FOREIGN KEY (`sojourn_id`) REFERENCES Sojourn(`sojourn_id`),
    FOREIGN KEY (`staff_id`) REFERENCES Staff(`CF`),
    PRIMARY KEY (`request_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS Selection (
    /*Selezione (id_selezione, id_richiesta, id_camera, id_servizio)
    Selection (selection_id, request_id, room_id, service_id) */
    selection_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
    request_id INTEGER UNSIGNED,
    room_id INTEGER UNSIGNED,
    service_id INTEGER UNSIGNED,
    FOREIGN KEY (`request_id`) REFERENCES Request(`request_id`),
    FOREIGN KEY (`room_id`) REFERENCES Room(`room_id`),
    FOREIGN KEY (`service_id`) REFERENCES Servicing(`service_id`),
    PRIMARY KEY (`selection_id`, `request_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

Si precisa che *alcuni nomi di attributi e tavole* sono stati cambiati in fase di programmazione e non corrispondono con lo schema logico poiché *in contrasto con le parole chiavi riconosciute da MySQL*.

Definizione delle operazioni

La definizione di varie operazioni d'esempio è contenuta all'interno del file operations.sql.

```

--01 : Cliente effettua prenotazione per una camera
DELIMITER //
CREATE PROCEDURE op1(IN num INT, IN client INT)
BEGIN
    INSERT INTO Reservation(num_people, customer_id) VALUES (num, client);
END //
DELIMITER ;

```

```
--02 : Cliente modifica prenotazione (data del soggiorno)
DELIMITER //
CREATE PROCEDURE op2(IN data_i DATE, IN data_f DATE, IN id INT)
BEGIN
    UPDATE Sojourn
    SET date_in = data_i AND date_fin = data_f
    WHERE sojourn_id = id;
END //
DELIMITER ;

--03 : Cliente cancella la prenotazione
DELIMITER //
CREATE PROCEDURE op3(IN id INT)
BEGIN
    DELETE
    FROM Reservation
    WHERE reservation_id = id;
END //
DELIMITER ;

--04 : Ricerca delle prenotazioni di un cliente
DELIMITER //
CREATE PROCEDURE op4(IN id INT)
BEGIN
    SELECT *
    FROM Reservation R, Customer C, Request Re
    WHERE R.customer_id = C.customer_id AND C.customer_id = id AND R.reservation_id
= Re.reservation_id;
END //
DELIMITER ;

--05 : Calcolo del costo della prenotazione
DELIMITER //
CREATE PROCEDURE op5(IN id INT)
BEGIN
    SELECT R.reservation_id, SUM(cost) AS CostoTotale
    FROM Reservation R, Request Re
    WHERE R.reservation_id = Re.reservation_id AND R.reservation_id = id
    GROUP BY R.reservation_id;
END //
DELIMITER ;
```

```
--06 : Visualizzare lo storico delle prenotazioni per un cliente
DELIMITER //
CREATE PROCEDURE op6(IN id INT)
BEGIN
    SELECT *
    FROM Reservation R, Customer C, Request Re, Sojourn S, Selection Se, Servicing
Ser, Room Ro
    WHERE R.customer_id = C.customer_id AND C.customer_id = id AND R.reservation_id
= Re.reservation_id AND Re.sojourn_id = S.sojourn_id AND Se.request_id =
Re.request_id AND Ser.service_id = Se.service_id AND Ro.room_id = Se.room_id;
END //
DELIMITER ;

--07 : Visualizzare i dettagli di una richiesta
DELIMITER //
CREATE PROCEDURE op7(IN id INT)
BEGIN
    SELECT *
    FROM Request Re, Reservation R, Sojourn S, Customer C
    WHERE R.reservation_id = Re.reservation_id AND R.customer_id = C.customer_id
AND Re.sojourn_id = S.sojourn_id;
END //
DELIMITER ;

--08 : Visualizzare i dettagli del soggiorno di un cliente
DELIMITER //
CREATE PROCEDURE op8(IN id INT)
BEGIN
    SELECT date_in, date_fin
    FROM Reservation R, Sojourn S
    WHERE R.sojourn_id = S.sojourn_id AND R.reservation_id = id
END //
DELIMITER ;

--09 : Visualizzare disponibilità di una camera
DELIMITER //
CREATE PROCEDURE op9(IN id INT)
BEGIN
    SELECT availability_room
    FROM Room R
    WHERE room_id = id;
END //
DELIMITER ;
```

```
--010 : Aggiungere punti fedeltà nella tessera di un cliente
DELIMITER //
CREATE PROCEDURE op10(IN id INT)
BEGIN
    UPDATE Customer
    SET points = points + 5
    WHERE customer_id = id;
END //
DELIMITER ;

--011 : Visualizzare punti nella tessera di un cliente
DELIMITER //
CREATE PROCEDURE op11(IN id INT)
BEGIN
    SELECT points
    FROM Customer C
    WHERE customer_id = id;
END //
DELIMITER ;

--012 : Visualizzare recensioni camera
DELIMITER //
CREATE PROCEDURE op12(IN id INT)
BEGIN
    SELECT customer_id, num_stars, comment
    FROM Review R
    WHERE room_id = id;
END //
DELIMITER ;

--013 : Gestore genera report
DELIMITER //
CREATE PROCEDURE op13(IN idS INT, IN idR INT)
BEGIN
    INSERT INTO Report(report_id, collections, staff_id)
    SELECT idR, SUM(amount), idS
    FROM Invoice I
    WHERE report_id = NULL;

    UPDATE Invoice
    SET report_id = idR
    WHERE report_id = NULL;
END //
DELIMITER ;
```

```
--014 : Aggiunta di una nuova sede
DELIMITER //
CREATE PROCEDURE op14(IN addr VARCHAR(50), IN n VARCHAR(30))
BEGIN
    INSERT INTO Home(address_home, name_home)
    VALUES (addr, n);
END //
DELIMITER ;
```

Trigger

Per rendere **attiva** la base di dati, si inserisce all'interno di essa un trigger che permette di gestire i vincoli non esprimibili prima esposti.

Sono quindi anche stati implementati alcuni trigger come segue:

Avevo pensato a dei trigger ignorando il fatto che in MySQL non è possibile modificare la tabella su cui è stato scatenato il trigger durante l'esecuzione dello stesso trigger.

```
/*trigger che una volta inserito un soggiorno aggiunge la stagione di appartenenza*/

DELIMITER //
CREATE TRIGGER T_Soggiorno
AFTER INSERT ON Sojourn
FOR EACH ROW
BEGIN
    DECLARE inizio INT;
    DECLARE stagione INT;

    SELECT MONTH(S.date_in) INTO inizio /*prendiamo il mese di inizio del soggiorno*/
        FROM Sojourn S
        WHERE new.sojourn_id = S.sojourn_id;

    SELECT season_id INTO stagione /*salviamo l'id della stagione corrispondente a quel mese*/
        FROM Season Se
        WHERE Se.month_ = inizio;

    UPDATE Sojourn S
    SET season_id = stagione
    WHERE new.sojourn_id = S.sojourn_id; /*aggiungiamo la stagione al soggiorno*/
END //
DELIMITER ;
```

Per risolvere questo problema, ho considerato l'utilizzo di un altro tipo di oggetto (come una stored procedure) per gestire la modifica della tabella. Ottenendo la seguente procedura:

```
/*procedura per aggiornare la stagione del soggiorno*/
DELIMITER //
CREATE PROCEDURE P_Soggiorno(IN sojourn_id INT)
BEGIN
    DECLARE inizio INT;
    DECLARE stagione INT;

    SELECT MONTH(S.date_in) INTO inizio
    FROM Sojourn S
    WHERE S.sojourn_id = sojourn_id;

    SELECT season_id INTO stagione
    FROM Season Se
    WHERE Se.month_ = inizio;

    UPDATE Sojourn S
    SET S.season_id = stagione
    WHERE S.sojourn_id = sojourn_id;
END //
DELIMITER ;
```

Lo stesso problema vale per questo trigger.

```
/*trigger che una volta inserita una richiesta calcola il costo*/

DELIMITER //
CREATE TRIGGER T_Costo
AFTER INSERT ON Request
FOR EACH ROW
BEGIN
    DECLARE costocamere INT;
    DECLARE costoservizi INT;
    DECLARE numgiorni INT;
    DECLARE cliente INT;

    SELECT SUM(Ro.rate) INTO costocamere /*somma le tariffe di tutte le camere*/
    FROM Request Re, Room Ro, Selection S
    WHERE new.request_id = Re.request_id AND S.request_id = Re.request_id AND
S.room_id = Ro.room_id;

    SELECT SUM(Se.rate) INTO costoservizi /*somma le tariffe di tutti i servizi*/
    FROM Request Re, Selection S, Servicing Se
    WHERE new.request_id = Re.request_id AND S.request_id = Re.request_id AND
S.service_id = Se.service_id;

    SELECT DATEDIFF(So.date_fin, So.date_in) INTO numgiorni /*calcoliamo i giorni
del soggiorno*/
    FROM Request Re, Sojourn So
    WHERE new.request_id = Re.request_id AND Re.sojourn_id = So.sojourn_id;

    UPDATE Request R
    SET R.cost = (((costocamere +
costoservizi)*numgiorni)*sconto)/100 /*calcoliamo il costo con lo sconto*/
    WHERE new.request_id = R.request_id;

END //
DELIMITER ;
```

Risolvendo con la seguente procedura:

```
DELIMITER //
CREATE PROCEDURE calculate_cost(IN new_request_id INT)
BEGIN
    DECLARE costocamere INT;
    DECLARE costoservizi INT;
    DECLARE numgiorni INT;
    DECLARE cliente INT;

    SELECT SUM(Ro.rate) INTO costocamere /*somma le tariffe di tutte le camere*/
    FROM Request Re, Room Ro, Selection S
    WHERE new_request_id = Re.request_id AND S.request_id = Re.request_id AND S.room_id =
Ro.room_id;

    SELECT SUM(Se.rate) INTO costoservizi /*somma le tariffe di tutti i servizi*/
    FROM Request Re, Selection S, Servicing Se
    WHERE new_request_id = Re.request_id AND S.request_id = Re.request_id AND S.service_id =
Se.service_id;

    SELECT DATEDIFF(So.date_fin, So.date_in) INTO numgiorni /*calcoliamo i giorni del soggiorno*/
    FROM Request Re, Sojourn So
    WHERE new_request_id = Re.request_id AND Re.sojourn_id = So.sojourn_id;

    UPDATE Request R
    SET R.cost = (costocamere + costoservizi)*numgiorni /*calcoliamo il costo senza lo sconto*/
    WHERE new_request_id = R.request_id;
END //
DELIMITER ;
```



```
DELIMITER //
CREATE PROCEDURE calculate_cost_invoice(IN new_invoice_id INT)
BEGIN
    DECLARE costorischiesta FLOAT;
    DECLARE num_persone FLOAT;
    DECLARE sconto FLOAT;

    SELECT Re.cost, R.num_people, C.points INTO costorischiesta, num_persone, sconto
    FROM Invoice I, Reservation R, Request Re, Customer C
    WHERE I.invoice_number = new_invoice_id AND I.reservation_id = R.reservation_id AND
R.reservation_id = Re.reservation_id AND C.customer_id = R.customer_id;

    UPDATE Invoice I
    SET I.amount = ((costorischiesta * num_persone) + ((costorischiesta * num_persone) * I.rate /
100)) - sconto
    WHERE I.invoice_number = new_invoice_id;
END //
DELIMITER ;
```

```

/*trigger che una volta registrato un utente ad una sede aggiunge il numero della tessera*/

DELIMITER //
CREATE TRIGGER T_tessera
AFTER INSERT ON Signup
FOR EACH ROW
BEGIN
    UPDATE Customer C
    SET C.num_card = RAND(new.customer_id)
    WHERE C.customer_id = new.customer_id;
END //
DELIMITER ;

```

Popolamento tabelle

È stato preparato un file per il popolamento delle tabelle in modo da potere già visualizzare alcuni dati iniziali nel database, il nome del file è “table_population.sql”.

```

INSERT INTO Season (month_, rate)
VALUES (1, 10),(2, 15),(3, 5),(4, 10),(5, 10),(6, 20),(7, 30),(8, 40),(9, 30),(10, 20),(11, 10),(12, 40);

INSERT INTO Customer(first_name, last_name, phone, email)
VALUES ('Alessia', 'Maccarrone', '1234567890', 'maccalessia@gmail.com'),('Mario', 'Bianchi', '1234567890', 'mario@gmail.com'),('Vittoria', 'Rossi', '1234567890', 'rossi@gmail.com'),('Salvo', 'Catania', '1234567890', 'salvo@gmail.com'),('Gioele', 'Federico', '1234567890', 'fede@gmail.com');

INSERT INTO Home(address_home, name_home)
VALUES ('via catania 1', 'il b&b'),('piazza Italia 209', 'la casa storica'),('via Federico 18', 'Fede b&b');

INSERT INTO Contact(role_staff, office, phone, home_id)
VALUES ('assistenza', 'via Amantea 33', '1234567890', 1), ('prenotazioni', 'via Amantea 33', '1234567890', 1), ('assistenza', 'via Tropea 33', '1234567890', 2), ('assistenza', 'via Tropea 33', '1234567890', 3);

INSERT INTO Room(name_room, type_room, rate, home_id)
VALUES ('camera Deluxe', 'camera singola', 40, 1), ('camera Queen', 'camera doppia', 50, 1), ('camera Matrimoniale standard', 'matrimoniale', 70, 2), ('camera familiare standard', 'tripla', 100, 2), ('camera con vista oceano', 'doppia uso singola', 80, 3), ('Suite King Deluxe', 'camera singola', 40, 3), ('Suite master', 'camera quafrupla', 100, 1), ('camera love', 'matrimoniale', 50, 2), ('camera bimbi', 'camera tripla', 90, 3);

```

```
INSERT INTO Servicing(description_service, rate, home_id)
VALUES ('parcheggio', 5, 1), ('piscina', 10, 2), ('cucina', 5, 3);

INSERT INTO PrivateService(rate, room_id)
VALUES (5, 3), (10, 5), (15, 1), (5, 4);

INSERT INTO Staff(CF, hours_worked, salary, role_staff, contact_id, home_id)
VALUES ('mqwert02h67f789x', 50, 1500, 'segretario', 1, 1), ('asccdfver', 80, 2000,
'addetto della pulizia', 1, 1), ('adcsewe', 40, 3000, 'gestore', 2, 1),
('vererwfa', 50, 1500, 'segretario', 2, 2), ('erfvewwv', 80, 2000, 'addetto della
pulizia', 3, 2), ('vtevberthr', 40, 3000, 'gestore', 3, 2), ('rvbtrnyjge', 50, 1500,
'segretario', 4, 3), ('vrtgtrbhreb', 80, 2000, 'addetto della pulizia', 4, 3),
('bge6ye5bngds', 40, 3000, 'gestore', 4, 3), ('ffvsrtgsthb', 50, 1500,
'segretario', 4, 1), ('cebfkhrfk', 80, 2000, 'addetto della pulizia', 4, 2),
('awjlefefefbe', 40, 3000, 'gestore', 1, 3);

INSERT INTO Report(collections, staff_id)
VALUES (0, 'adcsewe'), (0, 'bge6ye5bngds'), (0, 'vtevberthr');

INSERT INTO Reservation(num_people, customer_id)
VALUES (1, 1), (2, 2), (1, 3), (1, 4), (1, 5);

INSERT INTO Signup(customer_id, home_id)
VALUES (1, 1), (2, 2), (3, 3);

INSERT INTO Sojourn(date_in, date_fin)
VALUES ('2022-09-20', '2022-09-25'), ('2022-10-02', '22-10-07'), ('2022-10-15',
'2022-10-22'), ('2022-11-25', '22-11-30'), ('2022-12-20', '2022-12-25');

INSERT INTO Request(reservation_id, sojourn_id, staff_id)
VALUES (1, 1, 'ffvsrtgsthb'), (2, 2, 'vererwfa'), (3, 3, 'rvbtrnyjge'), (4, 4,
'ffvsrtgsthb'), (5, 5, 'vererwfa');

INSERT INTO Selection(request_id, room_id)
VALUES (1, 1), (2, 2), (3, 3), (4, 4), (5, 5);

INSERT INTO Invoice(issuer, pay_type, customer_id, reservation_id)
VALUES (15, 'bonifico', 1, 1), (15, 'bonifico', 2, 2), (15, 'bonifico', 3, 3), (15,
'bonifico', 4, 4), (15, 'bonifico', 5, 5);

INSERT INTO Review(customer_id, room_id, num_stars)
VALUES (1, 1, 4), (2, 2, 5), (3, 3, 3);
```

Database esportato da phpMyAdmin

Nella cartella “beb.sql.zip”.

5. Interfaccia web



È stata realizzata una piccola demo di prova per effettuare alcune query sul database e visualizzare i dati inseriti su pagine html. La tecnologia utilizzata è Laravel 8 e il codice è stato scritto in linguaggio PHP.

Il progetto con tutte le cartelle e il codice relativo al suo funzionamento verrà incluso all'interno della cartella “`src/webManagement/`”, disponibile su un link di github.

Demo di prova Laravel

Comandi utilizzati per la creazione:

```
#creiamo il Progetto Laravel col seguente comando
composer create-project --prefer-dist laravel/laravel webManagement

#creazione delle migrazioni per ogni tabella del database
php artisan make:model -a NomeTabella
#seguendo l'ordine della creazione delle tabelle in sql
#adesso seguirà la programmazione in php delle tabelle

#per creare le tabelle così come previsto dalle migrazioni
php artisan migrate

#adesso possiamo implementare il modello MODEL VIEW CONTROLLER (MVC)
# quindi per ogni tabella creare una classe MODEL, CONTROLLER e diverse VIEW
```

Interfaccia illustrata

Prima di iniziare attraverso questa riga da terminale possiamo verificare i percorsi della route:

```
php artisan route:list --exception=vendor
```

Per ogni tabella è stata realizzata una pagina di visualizzazione dei record e una o più pagine di inserimento, modifica e cancellazione.

Repository github con corrispondente codice



<https://github.com/MaccarroneAlessia/database-project>

Comandi per l'esecuzione del progetto e il test

Per i seguenti comandi si suppone di aver già installato MySQL, php, phpMyAdmin, Composer e Laravel nel proprio computer. In quanto prerequisiti si consiglia di installarli prima di provare i comandi a seguire nel proprio terminale Linux.



```
git clone https://github.com/MaccarroneAlessia/database-project.git

#per spostare il database in locale
cd database-project/src/sql\ operations/
sudo mysql -u root < mysql_setup.sql
sudo mysql -u root < tables.sql
sudo mysql -u root < operations.sql
sudo mysql -u root < trigger.sql
sudo mysql -u root < table_population.sql

sudo mysql -u admin -padmin #per visualizzare la creazione e le operazioni

#per aprire l'interfaccia web del database di phpMyAdmin
cd /etc/phpmyadmin
php -S localhost:7777

#per visualizzare l'interfaccia web laravel
cd database-project/src/sql\ operations/
sudo mysql -u root < mysql_setup.sql

cd database-project/src/webManagement/
php artisan migrate #per eseguire le migrazioni che creeranno le tabelle
php artisan route:list --exception=vendor #per vedere la lista delle rotte
php artisan serve #per attivare il server
#aprire il browser all'indirizzo localhost:8000
```

