

The background of the page features a large, light blue watermark of the University of Pisa seal. The seal is circular and contains a central figure of a cherub (putto) with curly hair, looking upwards. The text "UNIVERSITAS PISANA" is written around the top inner edge of the seal, and "MDCCCXXXIII" (1833) is at the bottom. A small star-like emblem is located at the bottom center of the seal.

UNIVERSITÀ DEGLI STUDI DI PISA

Facoltà di Ingegneria

Corso di Laurea Triennale in Ingegneria Informatica

RMS
Rental Management System

A cura dello studente **Giamir Buoncristiani (451400)**

ANNO ACCADEMICO 2012/2013

Traccia progetto

Sommario

La base di dati viene utilizzata da un insieme di agenzie sparse sul territorio, ognuna delle quali gestisce un insieme di proprietari di una o più abitazioni da dare in affitto ad inquilini separatamente stanza per stanza.

Le agenzie curano per i conduttori e gli inquilini una serie di servizi relativi agli appartamenti affittati; alcuni di questi servizi sono messi a disposizione anche di utenti esterni. Quindi gli utilizzatori della base di dati possono essere registrati con ruoli diversi.

1 Specifiche di progetto

Ogni appartamento è composto da un insieme di stanze, ciascuna con una funzione definita (cucina, bagno, camera, ecc.).

Ogni stanza ha un arredamento composto da un insieme di mobili o oggetti (inventario). Ogni inquilino presente o passato è mantenuto nel database (con i suoi dati anagrafici) associato alla stanza in cui ha abitato da una certa data ad un'altra (una persona può abitare stanze diverse in periodi diversi, ma anche la stessa stanza in periodi diversi) finché rimane in gestione all'agenzia. Un inquilino può affittare più stanze, comunque in generale risulterà affittuario, ad esempio, esclusivo della sua camera e in condivisione della cucina, di un bagno, del salotto, del terrazzo ecc. La quota fissa di spesa per l'affitto viene fissata dal conduttore (o proprietario) per ogni inquilino al momento del contratto.

L'agenzia mantiene, per ogni appartamento, lo scadenzario dei pagamenti che ogni persona residente in quell'appartamento deve effettuare. Le spese possono essere condivise tra più inquilini. I pagamenti passati vengono mantenuti in modo da avere la possibilità di valutare l'affidabilità di un inquilino rispetto alle scadenze da rispettare.

Rispetto agli inquilini, l'agenzia mette a disposizione anche la visione dello stato della dispensa di ogni appartamento: elenco dei beni (viveri, detersivi, ecc) con indicazione della proprietà di ognuno. Per quanto riguarda i viveri, memorizza anche la data di scadenza. Ugualmente l'agenzia gestisce l'agenda dei turni delle pulizie di ogni appartamento, condivisi tra gli affittuari dei singoli appartamenti, con la data prevista e la data in cui sono state realmente effettuate. Le pulizie possono anche riguardare le parti condominiali nel caso un appartamento sia collocato in un condominio; oppure il giardino, per gli appartamenti che lo hanno in uso.

Il sistema è utilizzato da: Inquilini, Proprietari, Utenti esterni, Agenzie.

1.1.1 Inquilino

Può visualizzare lo scadenzario dei pagamenti, visualizzare/modificare l'agenda dei turni delle pulizie relativamente alla casa in cui abita e per il periodo in cui ci abita. Può chiedere una modifica all'inventario della casa in cui abita. Visualizza/modifica lo stato della dispensa per capire cosa comprare o buttare via.

1.1.2 Proprietario

Visualizza/modifica lo scadenzario dei pagamenti e l'inventario delle stanze per le abitazioni di cui è proprietario; può anche visualizzare l'agenda dei turni delle pulizie delle sue case per valutare la "qualità" di un inquilino che è suo affittuario, oppure valutare un inquilino di un'altra casa in gestione all'agenzia che vuole spostarsi in una casa di sua proprietà.

1.1.3 Utente esterno

Può essere un potenziale nuovo inquilino che vuole valutare il tipo di casa e "la qualità" dei suoi abitanti attuali prima di trasferirvi; quindi vedrà l'occupazione, lo scadenzario e l'agenda delle pulizie.

1.1.4 Agenzia

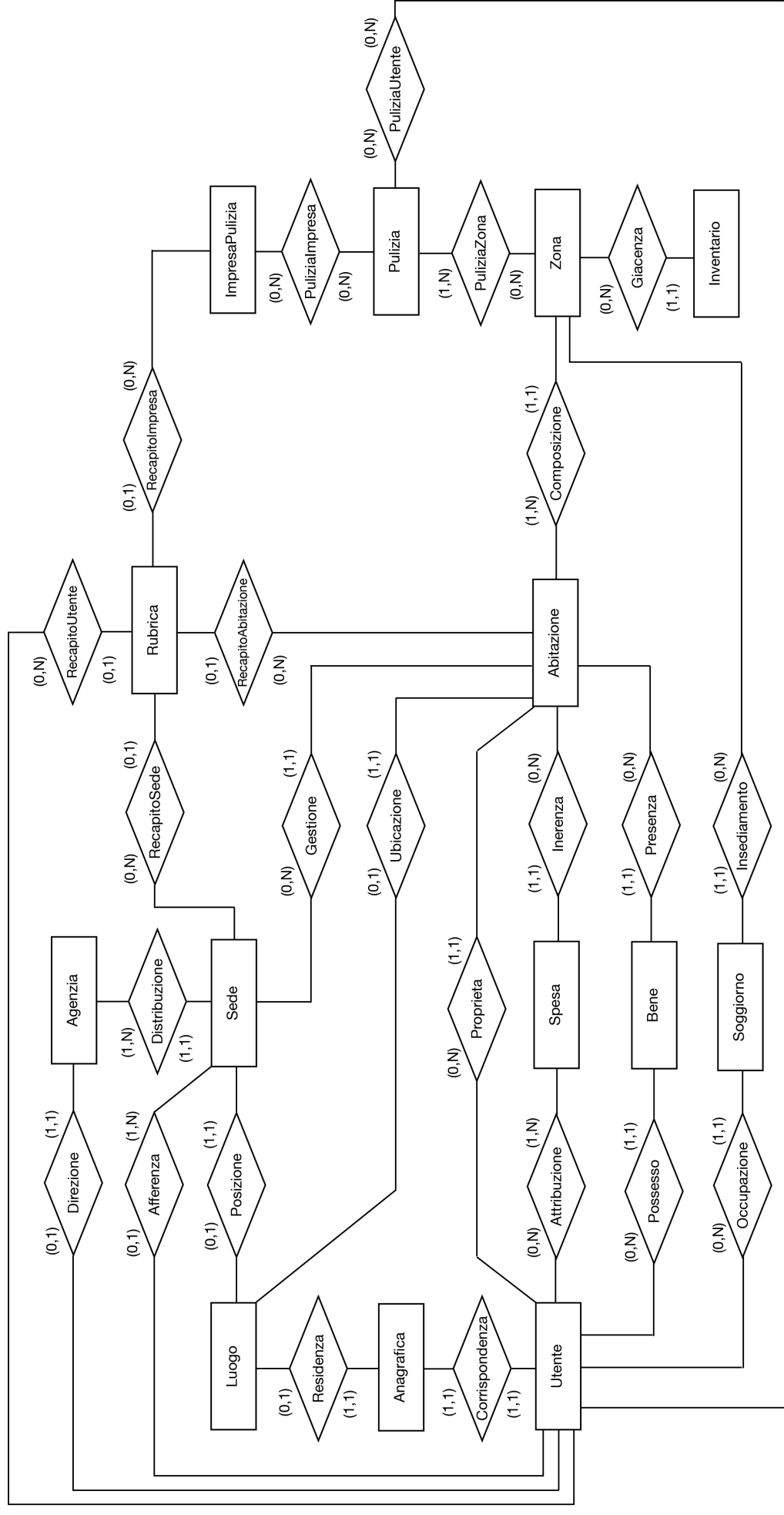
Le agenzie si occupano di stipulare i contratti di affitto e quindi modificare la lista degli inquilini o delle stanze affittate da ognuno di loro. Mantengono i contatti con i proprietari per avere il consenso alla stipula dei contratti. Inseriscono nuovi pagamenti nello scadenzario.

Assunzioni iniziali

Di seguito sono riportate le scelte concettuali più rilevanti fatte dal progettista comprensive di motivazioni.

- L'assunzione più importante che è stata fatta è senz'altro quella di incentrare tutto il sistema RMS intorno all'entità Utente. Chiunque voglia accedere al database dovrà farlo autenticandosi come singolo individuo, sia egli un inquilino, un utente esterno, un proprietario, un direttore o un dipendente di un'agenzia immobiliare. Ogni utente una volta autenticato avrà ovviamente privilegi diversi a seconda della figura.
- Le agenzie non gestiscono direttamente le singole abitazioni ma lo fanno mediante le sedi (può essere soltanto una) dislocate sul territorio.
- È stata inoltre prevista una entità rubrica telefonica dove è possibile registrare i recapiti di abitazioni, utenti, sedi di agenzie e imprese di pulizia.

Entity Relationship Diagram (ER)



N.B - Per evitare di appesantire lo schema ER gli attributi di ogni entità e di ogni relazione sono descritti dettagliatamente in apposite tabelle nelle pagine che seguono.

Entità	Descrizione	Attributi	Identificatore
Abitazione	<p>Abitazione o appartamento gestito da una sede di una agenzia.</p> <p>L'abitazione ha un attributo ridondante giardino che indica se ha almeno una zona di tipologia giardino.</p> <p>L'attributo giardino può assumere valore Y oppure N a seconda che il giardino sia o meno presente nell'abitazione.</p>	<p>✓ IdAbitazione</p> <p>✓ Giardino (R)</p>	IdAbitazione
Agenzia	<p>Agenzia distribuita sul territorio grazie a varie sedi locali (almeno una per ogni agenzia).</p> <p>L'agenzia ha una ragione sociale e la partita iva.</p> <p>Possiede inoltre una email principale diversa da quelle delle singole sedi dislocate sul territorio per comunicazioni generali che non riguardano le singole abitazioni gestite.</p>	<p>✓ Partitalva</p> <p>✓ RagSociale</p> <p>✓ Email</p> <p>✓ NumCaseGestite (R)</p>	Partitalva
Anagrafica	<p>Qui vengono raccolti tutti i dati anagrafici di ogni singolo utente registrato al RMS.</p> <p>Anagrafica è composta da codice fiscale, nome, cognome, data e luogo di nascita, cittadinanza e sesso dell'utente.</p> <p>Inoltre grazie alle relazioni con la rubrica telefonica ed i luoghi si è in grado di risalire ai recapiti telefonici e alla residenza di ogni utente.</p>	<p>✓ CodFiscale</p> <p>✓ Nome</p> <p>✓ Cognome</p> <p>✓ DataNascita</p> <p>✓ LuogoNascita</p> <p>✓ Cittadinanza</p> <p>✓ Sesso</p>	CodFiscale
Bene	<p>Raccoglie i beni di proprietà di un inquilino presenti in un'abitazione.</p> <p>Il bene può essere di tipo alimentare oppure generico.</p> <p>Nel caso in cui non sia un bene di tipo alimentare l'attributo "DataScadenza" assume valore NULL.</p>	<p>✓ IdBene</p> <p>✓ Tipo</p> <p>✓ Nome</p> <p>✓ Descrizione</p> <p>✓ DataScadenza</p>	IdBene

ImpresaPulizia	<p>Elenco delle imprese di pulizia registrate al RMS.</p> <p>Per ogni impresa si specifica partita iva, ragione sociale ed una email per eventuali comunicazioni.</p> <p>Di ogni impresa è possibile conoscere telefono, fax, ecc. perché collegata alla rubrica telefonica.</p>	<p>✓ Partitalva</p> <p>✓ RagSociale</p> <p>✓ Email</p>	Partitalva
Inventario	<p>Qui sono raccolti singolarmente gli oggetti che giacciono in una determinata zona dell'abitazione.</p> <p>Ad ogni oggetto è assegnato un identificativo, la tipologia (mobile, sedia, ecc.) e il suo stato (utilizzabile o non) .</p>	<p>✓ IdOggetto</p> <p>✓ Tipologia</p> <p>✓ Stato</p>	IdOggetto
Luogo	<p>Raccolta degli indirizzi che vengono immessi nel RMS.</p> <p>I luoghi possono riferirsi a residenze, sedi di agenzia e ubicazioni di abitazioni.</p> <p>Ogni luogo è composto oltre che da un identificatore, dall'indirizzo (via, frazione, num. civico, ecc.), dal comune, dalla provincia, dal CAP e dallo stato.</p> <p>L'indirizzo si può ulteriormente scomporre nel caso si voglia fare interrogazioni molto dettagliate.</p>	<p>✓ IdLuogo</p> <p>✓ Indirizzo</p> <p>✓ Comune</p> <p>✓ Provincia</p> <p>✓ CAP</p> <p>✓ Stato</p>	IdLuogo
Pulizia	<p>Elenco dei turni di pulizie.</p> <p>Ogni turno è composto da un identificativo, dalla data prevista e dalla data in cui realmente è stata effettuata la pulizia.</p> <p>Ogni pulizia è collegata alla zona/e in cui deve essere effettuata e dall'utente/i o l'impresa/e a cui ne è stato affidato il compito.</p> <p>Si precisa che nel momento in cui l'agenzia immette un nuovo turno di pulizia il campo DataEffettiva avrà valore NULL finché non sarà eseguita la pulizia.</p>	<p>✓ IdPulizia</p> <p>✓ DataPrevista</p> <p>✓ DataEffettiva</p>	IdPulizia
Rubrica	<p>Raccolta dei recapiti telefonici che vengono immessi nel RMS.</p> <p>Ogni singolo recapito è composto, oltre che da un identificativo, dal numero e dalla tipologia (fisso, cellulare, fax).</p>	<p>✓ IdTel</p> <p>✓ Numero</p> <p>✓ Tipologia</p>	IdTel

Sede	<p>Sede locale di un'agenzia. Si assume che un'agenzia debba avere almeno una sede. Alla sede deve afferire almeno un utente (impiegato). Ogni sede è collegata ad un'agenzia, ad un indirizzo e ad un recapito telefonico.</p>	✓	IdSede	IdSede
		✓	Email	
Soggiorno	<p>Elenco dei soggiorni. Ogni soggiorno è collegato ad un utente ed una zona di una abitazione. Un soggiorno si compone, oltre che da un identificatore, dalla data di inizio, dalla data di fine e dalla quota pattuita con il proprietario dell'abitazione.</p>	✓	IdSoggiorno	IdSoggiorno
		✓	DataInizio	
		✓	DataFine	
		✓	Quota	
Spesa	<p>Qui si inseriscono tutte le spese con relative scadenze che uno o più utenti di un'abitazione devono pagare. Ogni spesa ha, oltre il proprio identificatore e la propria descrizione, una data di scadenza, una data di effettivo pagamento, un importo ed una eventuale mora che può assumere valore diverso da 0 solo se DataPagata > DataScadenza. Si precisa che nel momento in cui l'agenzia immette una nuova spesa i campi DataPagata e Mora avranno valore NULL finché non sarà saldata la spesa.</p>	✓	IdSpesa	IdSpesa
		✓	Descrizione	
		✓	DataScadenza	
		✓	DataPagata	
		✓	Importo	
		✓	Mora	
Utente	<p>L'utente è l'entità su cui è basato il sistema di accesso al RMS. L'utente per accedere al sistema dovrà autenticarsi con username e password. Si assume che la password sia composta da 32 cifre esadecimali ottenute mediante l'algoritmo crittografico di hashing MD5. Ad ogni utente è associata anche una email in modo tale che lo sviluppatore possa implementare funzioni in grado di avvisare automaticamente circa scadenze e comunicazioni varie.</p>	✓	Username	Username
		✓	Password	
		✓	Email	

Zona

Elenco delle zone presenti nelle varie abitazioni inserite all'interno del RMS classificate per tipologia.
Così un'abitazione che comprende anche un giardino e una parte condominiale sarà composta da una zona di tipologia "giardino" ed un'altra di tipologia "area condominiale".

√ IdZona

IdZona

√ Tipologia

Relazione	Descrizione	Componenti	Attributi
Afferenza	Afferenza di un utente ad una sede di una determinata agenzia (impiegato di una sede di un'agenzia).	<div>✓ Utente</div> <div>✓ Sede</div>	NO
Attribuzione	Attribuzione di una spesa ad uno o più utenti.	<div>✓ Utente</div> <div>✓ Spesa</div>	NO
Composizione	Composizione dell'abitazione da più zone specifiche.	<div>✓ Abitazione</div> <div>✓ Zona</div>	NO
Corrispondenza	Corrispondenza tra l'utente ed i suoi dati anagrafici.	<div>✓ Utente</div> <div>✓ Anagrafica</div>	NO
Direzione	Direzione da parte di un utente di una agenzia. Ogni agenzia ha un solo direttore.	<div>✓ Utente</div> <div>✓ Agenzia</div>	NO
Distribuzione	Distribuzione delle sedi di un'agenzia sul territorio.	<div>✓ Agenzia</div> <div>✓ Sede</div>	NO
Gestione	Gestione delle abitazioni da parte di una sede locale di un'agenzia.	<div>✓ Sede</div> <div>✓ Abitazione</div>	NO
Giacenza	Giacenza di elementi di arredo in una zona.	<div>✓ Zona</div> <div>✓ Inventario</div>	NO
Inerenza	Inerenza di una spesa con l'abitazione.	<div>✓ Spesa</div> <div>✓ Abitazione</div>	NO
Insedimento	Zona legata ad un determinato periodo di soggiorno.	<div>✓ Zona</div> <div>✓ Soggiorno</div>	NO

Occupazione	Utente legato ad un determinato periodo di soggiorno.	√ Utente √ Soggiorno	NO
Posizione	Posizione geografica di una determinata sede.	√ Sede √ Luogo	NO
Possesso	Possesso di un bene da parte di un utente.	√ Utente √ Bene	NO
Presenza	Presenza di un bene in una abitazione.	√ Bene √ Abitazione	NO
Proprieta	Proprietà di un'abitazione da parte di un utente.	√ Utente √ Abitazione	NO
PuliziaImpresa	Relazione che lega la pulizia all'impresa/e a cui è stato affidato il compito di eseguirla.	√ ImpresaPulizia √ Pulizia	NO
PuliziaUtente	Relazione che lega la pulizia all'utente/i a cui è stato affidato il compito di eseguirla.	√ Utente √ Pulizia	NO
PuliziaZona	Relazione che lega la pulizia alla zona in cui deve essere effettuata.	√ Zona √ Pulizia	NO
RecapitoAbitazione	Relazione che lega l'abitazione alla rubrica telefonica del RMS.	√ Abitazione √ Rubrica	NO
RecapitoImpresa	Relazione che lega l'impresa di pulizie alla rubrica telefonica del RMS.	√ Impresa √ Rubrica	NO
RecapitoSede	Relazione che lega la sede di un'agenzia alla rubrica telefonica del RMS.	√ Sede √ Rubrica	NO

RecapitoUtente	Relazione che lega i dati anagrafici di un utente alla rubrica telefonica del RMS.	√	Anagrafica	NO
		√	Rubrica	
Residenza	Residenza di un determinato utente.	√	Anagrafica	NO
		√	Luogo	
Ubicazione	Ubicazione dell'abitazione in una determinata area geografica.	√	Abitazione	NO
		√	Luogo	

Individuazione ridondanze

Dall'analisi delle Entità emergono 2 ridondanze:

Giardino attributo dell'entità **Abitazione**:

si può ricavare se un abitazione possiede o meno un giardino controllando se tra le zone di cui si compone è presente almeno una zona con l'attributo tipologia uguale a giardino.

NumCaseGestite attributo dell'entità **Agenzia**:

si può ricavare il numero di case gestite da un'agenzia controllando quante case gestisce ogni sua singola sede e facendo una semplice somma.

Individuazione operazioni da compiere sul database

n.	Operazione	Frequenza
1	Assegnamento di un inquilino ad una certa zona.	100 al giorno
2	Inserire una nuova abitazione.	10 al giorno
3	Inserire una nuova zona.	100 al giorno
4	Stampare il numero di case che gestisce una determinata agenzia.	5 al giorno
5	Stampare l'indirizzo delle abitazioni che dispongono di giardino.	50 al giorno
6	Stampare username degli inquilini che hanno soggiornato o soggiornano in abitazioni gestite da una certa sede.	1 al giorno
7	Aggiungere una nuova spesa inerente ad una determinata abitazione che coinvolge un determinato inquilino.	500 al giorno
8	Controllare tutti i viveri scaduti ed inviare una email di avviso agli utenti che ne sono proprietari.	1 al giorno

Tavola dei volumi

Concetto	Tipo	Volume
Abitazione	E	1000
Agenzia	E	10
Anagrafica	E	10000
Bene	E	15000
ImpresaPulizia	E	50
Inventario	E	20000
Luogo	E	11040
Pulizia	E	50000
Rubrica	E	10000
Sede	E	40
Soggiorno	E	20000
Spesa	E	50000
Utente	E	10000
Zona	E	5000
Afferenza	R	120
Attribuzione	R	100000
Composizione	R	5000
Corrispondenza	R	10000
Direzione	R	10
Distribuzione	R	40
Gestione	R	1000
Giacenza	R	20000
Inerenza	R	50000
Insedimento	R	20000
Occupazione	R	20000
Posizione	R	40
Possesso	R	15000
Presenza	R	15000
Proprieta	R	1000
PuliziaImpresa	R	25000
PuliziaUtente	R	25000
PuliziaZona	R	50000
RecapitoAbitazione	R	800
RecapitoImpresa	R	100
RecapitoSede	R	100
RecapitoUtente	R	9000
Residenza	R	10000
Ubicazione	R	1000

Tavole degli accessi

Operazione 1

Assegnamento di un inquilino ad una certa abitazione.

Concetto	Costrutto	Accessi	Tipo
Soggiorno	E	1	S
Occupazione	R	1	S
Insedimento	R	1	S

$100 \times 3 = 300$ accessi in scrittura

$100 \times 0 = 0$ accessi in lettura

TOTALE 600

Operazione 2 (con ridondanza NumCaseGestite)

Inserire una nuova abitazione.

Concetto	Costrutto	Accessi	Tipo
Abitazione	E	1	S
Ubicazione	R	1	S
Composizione	R	5	S
Gestione	R	1	S
Proprietà	R	1	S
Sede	E	1	L
Distribuzione	R	1	L
Agenzia	E	1	L
Agenzia	E	1	S

$10 \times 10 = 100$ accessi in scrittura

$10 \times 3 = 30$ accessi in lettura

TOTALE 230

Operazione 2 (senza ridondanza NumCaseGestite)

Inserire una nuova abitazione.

Concetto	Costrutto	Accessi	Tipo
Abitazione	E	1	S
Ubicazione	R	1	S
Composizione	R	5	S
Gestione	R	1	S
Proprietà	R	1	S

$10 \times 9 = 90$ accessi in scrittura

$10 \times 0 = 0$ accessi in lettura

TOTALE 180

Operazione 3 (con ridondanza Giardino)

Inserire una nuova zona.

Concetto	Costrutto	Accessi	Tipo
Zona	E	1	S
Composizione	R	1	S
Abitazione	E	1	L
Abitazione	E	1	S

$100 \times 3 = 300$ accessi in scrittura

$100 \times 1 = 100$ accessi in lettura

TOTALE 700

Operazione 3 (senza ridondanza Giardino)

Inserire una nuova zona.

Concetto	Costrutto	Accessi	Tipo
Zona	E	1	S
Composizione	R	1	S

$100 \times 2 = 200$ accessi in scrittura

$100 \times 0 = 0$ accessi in lettura

TOTALE 400

Operazione 4 (con ridondanza NumCaseGestite)

Stampare il numero di case che gestisce una determinata agenzia.

Concetto	Costrutto	Accessi	Tipo
----------	-----------	---------	------

Agenzia	E	1	L
---------	---	---	---

$5 \times 0 = 0$ accessi in scrittura

$5 \times 1 = 10$ accessi in lettura

TOTALE 5

Operazione 4 (senza ridondanza NumCaseGestite)

Stampare il numero di case che gestisce una determinata agenzia.

Concetto	Costrutto	Accessi	Tipo
----------	-----------	---------	------

Agenzia	E	1	L
---------	---	---	---

Distribuzione	R	4	L
---------------	---	---	---

Sede	E	4	L
------	---	---	---

Gestione	R	100	L
----------	---	-----	---

$5 \times 0 = 0$ accessi in scrittura

$5 \times 109 = 545$ accessi in lettura

TOTALE 545

Operazione 5 (con ridondanza Giardino)

Stampare l'indirizzo delle abitazioni che dispongono di giardino.

Si assume che 1 abitazione su 5 disponga di giardino.

Concetto	Costrutto	Accessi	Tipo
----------	-----------	---------	------

Abitazione	E	200	L
------------	---	-----	---

Ubicazione	R	200	L
------------	---	-----	---

Luogo	E	200	L
-------	---	-----	---

$50 \times 0 = 0$ accessi in scrittura

$50 \times 600 = 30000$ accessi in lettura

TOTALE 30000

Operazione 5 (senza ridondanza Giardino)

Stampare l'indirizzo delle abitazioni che dispongono di giardino.
Si assume che 1 abitazione su 5 disponga di giardino.

Concetto	Costrutto	Accessi	Tipo
Zona	E	200	L
Composizione	R	200	L
Abitazione	E	200	L
Ubicazione	R	200	L
Luogo	E	200	L

$50 \times 0 = 0$ accessi in scrittura

$50 \times 1000 = 50000$ accessi in lettura

TOTALE 50000

Operazione 6

Stampare username degli inquilini che hanno soggiornato o soggiornano in abitazioni gestite da una certa sede.
Considerando: utenti inquilini circa 8000, zone di tipo camera (affittabili) circa 2000.

Soggiorno è 500 perché $50(\text{zon.aff.op6}) : X(\text{sog.op6}) = 2000(\text{zon.aff}) : 20000(\text{sog})$ $X = 500$

Utente è 200 perché $500(\text{sog.op6}) : Y(\text{ut.op6}) = 20000(\text{sog}) : 8000(\text{ut.inq})$ $Y = 200$

Concetto	Costrutto	Accessi	Tipo
Sede	E	1	L
Gestione	R	25	L
Abitazione	E	25	L
Composizione	R	125	L
Zona	E	125	L
Insediamiento	R	500	L
Soggiorno	E	500	L
Occupazione	R	500	L
Utente	E	200	L

$1 \times 0 = 0$ accessi in scrittura

$1 \times 2001 = 2001$ accessi in lettura

TOTALE 2001

Operazione 7

Aggiungere una nuova spesa inerente ad una determinata abitazione che coinvolge un determinato inquilino.

Concetto	Costrutto	Accessi	Tipo
Spesa	E	1	S
Attribuzione	R	1	S
Inerenza	R	1	S

$500 \times 3 = 1500$ accessi in scrittura

$500 \times 0 = 0$ accessi in lettura

TOTALE 3000

Operazione 8

Controllare tutti i viveri scaduti ed in scadenza nei successivi 3 giorni all'interrogazione ed inviare una email di avviso agli utenti che ne sono proprietari. Si assume che 50 sia la media dei viveri in scadenza o scaduti.

Concetto	Costrutto	Accessi	Tipo
Bene	E	15000	L
Possesso	R	50	L
Utente	E	50	L

$1 \times 0 = 0$ accessi in scrittura

$1 \times 15100 = 15100$ accessi in lettura

TOTALE 15100

Analisi ridondanze

Dall'analisi della ridondanza **NumCaseGestite** emerge che:

Operazione di scrittura: con ridondanza 230 acc/g - senza ridondanza 180 acc/g

Operazione di lettura: con ridondanza 5 acc/g - senza ridondanza 545 acc/g

Dall'analisi della ridondanza **Giardino** emerge che:

Operazione di scrittura: con ridondanza 700 acc/g - senza ridondanza 400 acc/g

Operazione di lettura: con ridondanza 30000 acc/g - senza ridondanza 50000 acc/g

Si decide quindi di mantenere la ridondanza Giardino perché permette di ridurre di circa 20000 gli accessi giornalieri al database mentre si elimina la ridondanza NumCaseGestite perché coinvolta solo in operazioni con frequenza giornaliera piuttosto bassa.

Generazione delle tabelle

Abitazione (IdAbitazione, Proprietario, Ubicazione, Gestione, Giardino)
Agenzia (Partitalva, RagSociale, Email, Direttore)
Anagrafica (CodFiscale, Nome, Cognome, DataNascita, LuogoNascita, Cittadinanza, Sesso, Utente, Residenza)
Attribuzione (Utente, Spesa)
Bene (IdBene, Tipo, Nome, Descrizione, DataScadenza, Abitazione, Possessore)
ImpresaPulizia (Partitalva, RagSociale, Email)
Inventario (IdOggetto, Tipologia, Stato, Zona)
Luogo (IdLuogo, Indirizzo, Comune, Provincia, CAP, Stato)
Pulizia (IdPulizia, DataPrevista, DataEffettiva)
PuliziaImpresa (Pulizia, Impresa)
PuliziaUtente (Pulizia, Utente)
PuliziaZona (Pulizia, Zona)
Rubrica (IdTel, Numero, Tipologia)
RubricaAbitazione (Tel, Abitazione)
RubricaImpresa (Tel, Impresa)
RubricaSede (Tel, Sede)
RubricaUtente (Tel, Utente)
Sede (IdSede, Email, Posizione, Agenzia)
Soggiorno (IdSoggiorno, DataInizio, DataFine, Quota, Zona, Utente)
Spesa (IdSpesa, Descrizione, DataScadenza, DataPagata, Importo, Mora, Abitazione)
Utente (Username, Password, Email, Afferenza)
Zona (IdZona, Tipologia, Abitazione)

Le tipologie di ogni attributo sono deducibili dallo script MySQL di creazione database presente nelle pagine successive; tali informazioni sono state omesse sopra per non rendere dispersiva la traduzione dello schema ER in tabelle.

Vincoli di integrità referenziale

Analizziamo in dettaglio i vincoli di integrità interreferenziale:

Tabella *Abitazione*:

Proprietario è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Ubicazione è foreign key con riferimento all'attributo *IdLuogo* della tabella esterna *Luogo*.

Gestione è foreign key con riferimento all'attributo *IdSede* della tabella esterna *Sede*.

Tabella *Agenzia*:

Direttore è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Tabella *Anagrafica*:

Utente è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Residenza è foreign key con riferimento all'attributo *IdLuogo* della tabella esterna *Luogo*.

Tabella *Attribuzione*:

Utente è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Spesa è foreign key con riferimento all'attributo *IdSpesa* della tabella esterna *Spesa*.

Tabella *Bene*:

Possessore è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Abitazione è foreign key con riferimento all'attributo *IdAbitazione* della tabella esterna *Abitazione*.

Tabella *Inventario*:

Zona è foreign key con riferimento all'attributo *IdZona* della tabella esterna *Zona*.

Tabella *PuliziaImpresa*:

Pulizia è foreign key con riferimento all'attributo *IdPulizia* della tabella esterna *Pulizia*.

Impresa è foreign key con riferimento all'attributo *Partitalva* della tabella esterna *ImpresaPulizia*.

Tabella *PuliziaUtente*:

Pulizia è foreign key con riferimento all'attributo *IdPulizia* della tabella esterna *Pulizia*.

Utente è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Tabella *PuliziaZona*:

Pulizia è foreign key con riferimento all'attributo *IdPulizia* della tabella esterna *Pulizia*.

Zona è foreign key con riferimento all'attributo *IdZona* della tabella esterna *Zona*.

Tabella *RubricaAbitazione*:

Tel è foreign key con riferimento all'attributo *IdTel* della tabella esterna *Rubrica*.

Abitazione è foreign key con riferimento all'attributo *IdAbitazione* della tabella esterna *Abitazione*.

Tabella *RubricaImpresa*:

Tel è foreign key con riferimento all'attributo *IdTel* della tabella esterna *Rubrica*.

Impresa è foreign key con riferimento all'attributo *Partitalva* della tabella esterna *ImpresaPulizia*.

Tabella *RubricaSede*:

Tel è foreign key con riferimento all'attributo *IdTel* della tabella esterna *Rubrica*.

Sede è foreign key con riferimento all'attributo *IdSede* della tabella esterna *Sede*.

Tabella *RubricaUtente*:

Tel è foreign key con riferimento all'attributo *IdTel* della tabella esterna *Rubrica*.

Utente è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Tabella *Sede*:

Agenzia è foreign key con riferimento all'attributo *Partitalva* della tabella esterna *Agenzia*.

Posizione è foreign key con riferimento all'attributo *IdLuogo* della tabella esterna *Luogo*.

Tabella *Soggiorno*:

Utente è foreign key con riferimento all'attributo *Username* della tabella esterna *Utente*.

Zona è foreign key con riferimento all'attributo *IdZona* della tabella esterna *Zona*.

Tabella *Spesa*:

Abitazione è foreign key con riferimento all'attributo *IdAbitazione* della tabella esterna *Abitazione*.

Tabella *Utente*:

Afferenza è foreign key con riferimento all'attributo *IdSede* della tabella esterna *Sede*.

Tabella *Zona*:

Abitazione è foreign key con riferimento all'attributo *IdAbitazione* della tabella esterna *Abitazione*.

Normalizzazione di Boyce-Codd

Si dice che una tabella è in 1NF (prima forma normale) se e solo se:

1. non presenta gruppi di attributi che si ripetono (ossia ciascun attributo è definito su un dominio con valori atomici).
2. esiste una chiave primaria (ossia esiste un insieme di attributi, che identifica in modo univoco ogni tupla della relazione).

Si dice che una tabella è in 2NF (seconda forma normale) se e solo se:

1. è in 1NF.
2. tutti i campi non chiave dipendono funzionalmente dall'intera chiave composta e non da una parte di essa.

Si dice che una tabella è in 3NF (terza forma normale) se e solo se:

1. è in 2NF.
2. tutti gli attributi non-chiave dipendono dalla chiave soltanto, ossia non esistono attributi che dipendono anche da altri attributi non-chiave.

Si dice che una tabella è in BCNF (forma normale di Boyce-Codd) se e solo se:

1. è in 3NF.
2. per ogni dipendenza funzionale non banale $X \rightarrow Y$, X è una superchiave della tabella.

Osservando le tabelle del database RMS è possibile affermare che rispettano tutte la BCNF.
L'unica tabella sulla quale si può fare qualche osservazione è BENE:

BENE

IdBene \rightarrow Nome, Tipo, Descrizione, Possessore, Abitazione

IdBene, Tipo \rightarrow DataScadenza

Notiamo che l'attributo DataScadenza oltre a dipendere dalla chiave IdBene dipende anche dall'attributo Tipo. L'attributo Tipo dipende a sua volta da IdBene quindi la seconda dipendenza funzionale diventa banale e la tabella risulta quindi rispettare la BCNF.

In realtà in base al valore che assume Tipo (alimentare o generico) l'attributo DataScadenza assume significato o meno (NULL). Quindi Tipo ci permette di effettuare un controllo automatico (con un trigger) sull'attributo DataScadenza.

Creazione e popolazione RMS tramite script MySQL

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

BEGIN;
CREATE DATABASE IF NOT EXISTS `RMS`;
COMMIT;

USE `RMS`;

-- -----
-- Table structure for `ABITAZIONE`
-- -----
DROP TABLE IF EXISTS `Abitazione`;
CREATE TABLE `Abitazione` (
  `IdAbitazione` int(11) NOT NULL AUTO_INCREMENT,
  `Proprietario` varchar(20) NOT NULL,
  `Ubicazione` int(11) NOT NULL,
  `Gestione` int(11) DEFAULT NULL,
  `Giardino` varchar(1) DEFAULT 'N',
  PRIMARY KEY (`IdAbitazione`),
  FOREIGN KEY (`Gestione`) REFERENCES `Sede` (`IdSede`) ON DELETE SET NULL ON UPDATE CASCADE,
  FOREIGN KEY (`Proprietario`) REFERENCES `Utente` (`Username`) ON DELETE NO ACTION ON UPDATE CASCADE,
  FOREIGN KEY (`Ubicazione`) REFERENCES `Luogo` (`IdLuogo`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `ABITAZIONE`
-- -----
BEGIN;
INSERT INTO `Abitazione` VALUES (1,'Ermenegilda50',7,1,'Y');
INSERT INTO `Abitazione` VALUES (2,'Giuseppe70',14,2,'N');
COMMIT;

-- -----
-- Table structure for `AGENZIA`
-- -----
DROP TABLE IF EXISTS `Agenzia`;
CREATE TABLE `Agenzia` (
  `PartitaIva` varchar(20) NOT NULL,
  `RagSociale` varchar(50) NOT NULL,
  `Email` varchar(40) NOT NULL,
  `Direttore` varchar(20) NOT NULL,
  PRIMARY KEY (`PartitaIva`),
  FOREIGN KEY (`Direttore`) REFERENCES `Utente` (`Username`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `AGENZIA`
-- -----
BEGIN;
INSERT INTO `Agenzia` VALUES ('FR12345678912','Immobiliare','info@immobiliare.com','Remi55');
INSERT INTO `Agenzia` VALUES ('IT01114601006','Stile Casa','info@stilecasa.com','Mario75');
COMMIT;

-- -----
-- Table structure for `ANAGRAFICA`
-- -----
DROP TABLE IF EXISTS `Anagrafica`;
CREATE TABLE `Anagrafica` (
  `CodFiscale` varchar(20) NOT NULL,
  `Nome` varchar(20) NOT NULL,
  `Cognome` varchar(20) NOT NULL,
  `DataNascita` date NOT NULL,
  `LuogoNascita` varchar(20) NOT NULL,
  `Cittadinanza` varchar(20) NOT NULL,
  `Sesso` varchar(1) NOT NULL,
  `Utente` varchar(20) NOT NULL,
  `Residenza` int(11) NOT NULL,
  PRIMARY KEY (`CodFiscale`),
  FOREIGN KEY (`Residenza`) REFERENCES `Luogo` (`IdLuogo`) ON DELETE NO ACTION ON UPDATE CASCADE,
  FOREIGN KEY (`Utente`) REFERENCES `Utente` (`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `ANAGRAFICA`
-- -----
BEGIN;
```

```

INSERT INTO `Anagrafica` VALUES ('ERMTRC50','Ermenegilda','Turchesi','1950-02-01','Napoli','Italiana','F','Ermenegilda50',6);
INSERT INTO `Anagrafica` VALUES ('ERNGLL85','Ernesto','Gialli','1985-03-02','Venezia','Italiana','M','Ernesto85',10);
INSERT INTO `Anagrafica` VALUES ('FRNGLL45','Francesca','Galli','1945-08-09','Agrigento','Italiana','F','Francesca45',16);
INSERT INTO `Anagrafica` VALUES ('GNNARC40','Gennaro','Arancioni','1940-10-14','Bari','Italiana','M','Gennaro40',15);
INSERT INTO `Anagrafica` VALUES ('GSPVRD70','Giuseppe','Verdi','1970-05-12','Lucca','Italiana','M','Giuseppe70',3);
INSERT INTO `Anagrafica` VALUES ('JHNDOE80','John','Doe','1980-02-27','London','Inglese','M','John80',9);
INSERT INTO `Anagrafica` VALUES ('LCRZZR90','Lucrezia','Azzurri','1990-12-09','Firenze','Italiana','F','Lucrezia90',5);
INSERT INTO `Anagrafica` VALUES ('MARRSS75','Mario','Rossi','1975-06-01','Pisa','Italiana','M','Mario75',1);
INSERT INTO `Anagrafica` VALUES ('RMIMRT55','Remi','Martell','1955-08-30','Nizza','Francese','M','Remi55',13);
INSERT INTO `Anagrafica` VALUES ('SVRORS65','Saverio','Orsucci','1965-09-23','Genova','Italiana','M','Saverio65',12);
COMMIT;

```

```

-- -----
-- Table structure for `ATTRIBUZIONE`
-- -----

```

```

DROP TABLE IF EXISTS `Attribuzione`;
CREATE TABLE `Attribuzione` (
  `Utente` varchar(20) NOT NULL,
  `Spesa` int(11) NOT NULL,
  PRIMARY KEY (`Spesa`,`Utente`),
  FOREIGN KEY (`Spesa`) REFERENCES `Spesa`(`IdSpesa`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Utente`) REFERENCES `Utente`(`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `ATTRIBUZIONE`
-- -----

```

```

BEGIN;
INSERT INTO `Attribuzione` VALUES ('Saverio65',1);
INSERT INTO `Attribuzione` VALUES ('Saverio65',2);
INSERT INTO `Attribuzione` VALUES ('Ernesto85',3);
INSERT INTO `Attribuzione` VALUES ('Ernesto85',4);
INSERT INTO `Attribuzione` VALUES ('Saverio65',4);
INSERT INTO `Attribuzione` VALUES ('Ernesto85',5);
INSERT INTO `Attribuzione` VALUES ('Saverio65',5);
INSERT INTO `Attribuzione` VALUES ('Ernesto85',6);
INSERT INTO `Attribuzione` VALUES ('Saverio65',6);
COMMIT;

```

```

-- -----
-- Table structure for `BENE`
-- -----

```

```

DROP TABLE IF EXISTS `Bene`;
CREATE TABLE `Bene` (
  `Idbene` int(11) NOT NULL AUTO_INCREMENT,
  `Tipo` varchar(10) NOT NULL,
  `Nome` varchar(20) NOT NULL,
  `Descrizione` varchar(1000) DEFAULT NULL,
  `DataScadenza` date DEFAULT NULL,
  `Abitazione` int(11) DEFAULT NULL,
  `Possessore` varchar(20) NOT NULL,
  PRIMARY KEY (`Idbene`),
  FOREIGN KEY (`Abitazione`) REFERENCES `Abitazione`(`IdAbitazione`) ON DELETE SET NULL ON UPDATE CASCADE,
  FOREIGN KEY (`Possessore`) REFERENCES `Utente`(`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `BENE`
-- -----

```

```

BEGIN;
INSERT INTO `Bene` VALUES (1,'Alimentare','Yogurt','uno yogurt da mangiare','2013-06-01',1,'Saverio65');
INSERT INTO `Bene` VALUES (2,'Generico','Mazza da Golf','una mazza da golf',NULL,1,'Ernesto85');
COMMIT;

```

```

-- -----
-- Table structure for `IMPRESAPULIZIA`
-- -----

```

```

DROP TABLE IF EXISTS `ImpresaPulizia`;
CREATE TABLE `ImpresaPulizia` (
  `PartitaIva` varchar(20) NOT NULL,
  `RagSociale` varchar(50) NOT NULL,
  `Email` varchar(40) NOT NULL,
  PRIMARY KEY (`PartitaIva`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `IMPRESAPULIZIA`
-- -----

```

```

BEGIN;
INSERT INTO `ImpresaPulizia` VALUES ('IT12345678910','Pulizie Intense SRL','info@pulizieintense.com');
COMMIT;

```

```

-- -----
-- Table structure for `INVENTARIO`
-- -----
DROP TABLE IF EXISTS `Inventario`;
CREATE TABLE `Inventario` (
  `IdOggetto` int(11) NOT NULL AUTO_INCREMENT,
  `Tipologia` varchar(20) NOT NULL,
  `Stato` varchar(1) NOT NULL,
  `Zona` int(11) NOT NULL,
  PRIMARY KEY (`IdOggetto`),
  FOREIGN KEY (`Zona`) REFERENCES `Zona` (`IdZona`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `INVENTARIO`
-- -----
BEGIN;
INSERT INTO `Inventario` VALUES (1,'Mobile','U',1);
INSERT INTO `Inventario` VALUES (2,'Sedia','N',1);
INSERT INTO `Inventario` VALUES (3,'Tavolo','U',1);
INSERT INTO `Inventario` VALUES (4,'Poltrona','U',4);
INSERT INTO `Inventario` VALUES (5,'Lampadario','U',4);
INSERT INTO `Inventario` VALUES (6,'Letto','U',3);
INSERT INTO `Inventario` VALUES (7,'Letto','U',3);
COMMIT;

-- -----
-- Table structure for `LUOGO`
-- -----
DROP TABLE IF EXISTS `Luogo`;
CREATE TABLE `Luogo` (
  `IdLuogo` int(11) NOT NULL AUTO_INCREMENT,
  `Indirizzo` varchar(50) NOT NULL,
  `Comune` varchar(20) NOT NULL,
  `Provincia` varchar(20) NOT NULL,
  `CAP` int(11) NOT NULL,
  `Stato` varchar(20) NOT NULL,
  PRIMARY KEY (`IdLuogo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `LUOGO`
-- -----
BEGIN;
INSERT INTO `Luogo` VALUES (1,'Via Dante Alighieri, 25','San Giuliano Terme','Pisa',12345,'Italy');
INSERT INTO `Luogo` VALUES (2,'Piazza Garibaldi, 30','Pisa','Pisa',12345,'Italy');
INSERT INTO `Luogo` VALUES (3,'Via Fillungo, 40','Lucca','Lucca',12345,'Italy');
INSERT INTO `Luogo` VALUES (4,'Via Cavour, 45','Roma','Roma',12345,'Italy');
INSERT INTO `Luogo` VALUES (5,'Largo Lucio Lazzarino, 125','Varazze','Savona',12345,'Italy');
INSERT INTO `Luogo` VALUES (6,'Vicolo Corto, 89','Montefalcone','Terni',12345,'Italy');
INSERT INTO `Luogo` VALUES (7,'Viale della Vittoria, 345','Agrigento','Agrigento',12345,'Italy');
INSERT INTO `Luogo` VALUES (8,'Via Ho Chi Min, 34','Nizza','Nizza',12345,'France');
INSERT INTO `Luogo` VALUES (9,'Oxford Street, 321','Liverpool','Liverpool',12345,'England');
INSERT INTO `Luogo` VALUES (10,'Piazza Giulio Cesare, 66','Milano','Milano',12345,'Italy');
INSERT INTO `Luogo` VALUES (11,'Via Marconi, 43','Agrigento','Agrigento',12345,'Italy');
INSERT INTO `Luogo` VALUES (12,'Via Dell\'Orto, 35','Lecce','Lecce',12345,'Italy');
INSERT INTO `Luogo` VALUES (13,'Piazza Voilà, 43','Paris','Paris',12345,'France');
INSERT INTO `Luogo` VALUES (14,'Piazza Dante, 100','Scandicci','Firenze',12345,'Italy');
INSERT INTO `Luogo` VALUES (15,'Via Martiri, 56','Bari','Bari',12345,'Italy');
INSERT INTO `Luogo` VALUES (16,'Piazza delle Vettovaglie, 30','Pisa','Pisa',12345,'Italy');
COMMIT;

-- -----
-- Table structure for `PULIZIA`
-- -----
DROP TABLE IF EXISTS `Pulizia`;
CREATE TABLE `Pulizia` (
  `IdPulizia` int(11) NOT NULL AUTO_INCREMENT,
  `DataPrevista` datetime NOT NULL,
  `DataEffettiva` datetime DEFAULT NULL,
  PRIMARY KEY (`IdPulizia`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `PULIZIA`
-- -----
BEGIN;
INSERT INTO `Pulizia` VALUES (1,'2013-01-01 00:00:00','2013-01-01 00:00:00');
INSERT INTO `Pulizia` VALUES (2,'2013-02-01 00:00:00','2013-02-01 00:00:00');
INSERT INTO `Pulizia` VALUES (3,'2013-03-01 00:00:00','2013-03-01 00:00:00');

```

```

INSERT INTO `Pulizia` VALUES (4,'2013-04-01 00:00:00','2013-04-01 00:00:00');
INSERT INTO `Pulizia` VALUES (5,'2013-05-01 00:00:00','2013-05-01 00:00:00');
INSERT INTO `Pulizia` VALUES (6,'2013-06-01 00:00:00','2013-06-01 00:00:00');
INSERT INTO `Pulizia` VALUES (7,'2013-07-01 00:00:00',NULL);
INSERT INTO `Pulizia` VALUES (8,'2013-08-01 00:00:00',NULL);
INSERT INTO `Pulizia` VALUES (9,'2013-09-01 00:00:00',NULL);
INSERT INTO `Pulizia` VALUES (10,'2013-10-01 00:00:00',NULL);
INSERT INTO `Pulizia` VALUES (11,'2013-11-01 00:00:00',NULL);
INSERT INTO `Pulizia` VALUES (12,'2013-12-01 00:00:00',NULL);
COMMIT;

-- -----
-- Table structure for `PULIZIAIMPRESA`
-- -----
DROP TABLE IF EXISTS `PuliziaImpresa`;
CREATE TABLE `PuliziaImpresa` (
  `Pulizia` int(11) NOT NULL,
  `Impresa` varchar(20) NOT NULL,
  PRIMARY KEY (`Pulizia`,`Impresa`),
  FOREIGN KEY (`Impresa`) REFERENCES `ImpresaPulizia` (`PartitaIva`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Pulizia`) REFERENCES `Pulizia` (`IdPulizia`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `PULIZIAIMPRESA`
-- -----
BEGIN;
INSERT INTO `PuliziaImpresa` VALUES (1,'IT12345678910');
INSERT INTO `PuliziaImpresa` VALUES (3,'IT12345678910');
INSERT INTO `PuliziaImpresa` VALUES (5,'IT12345678910');
INSERT INTO `PuliziaImpresa` VALUES (7,'IT12345678910');
INSERT INTO `PuliziaImpresa` VALUES (9,'IT12345678910');
INSERT INTO `PuliziaImpresa` VALUES (11,'IT12345678910');
COMMIT;

-- -----
-- Table structure for `PULIZIAUTENTE`
-- -----
DROP TABLE IF EXISTS `PuliziaUtente`;
CREATE TABLE `PuliziaUtente` (
  `Pulizia` int(11) NOT NULL,
  `Utente` varchar(20) NOT NULL,
  PRIMARY KEY (`Pulizia`,`Utente`),
  FOREIGN KEY (`Pulizia`) REFERENCES `Pulizia` (`IdPulizia`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Utente`) REFERENCES `Utente` (`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `PULIZIAUTENTE`
-- -----
INSERT INTO `PuliziaUtente` VALUES (2,'Saverio65');
INSERT INTO `PuliziaUtente` VALUES (4,'Ernesto85');
INSERT INTO `PuliziaUtente` VALUES (6,'Ernesto85');
INSERT INTO `PuliziaUtente` VALUES (6,'Saverio65');
INSERT INTO `PuliziaUtente` VALUES (8,'Saverio65');
INSERT INTO `PuliziaUtente` VALUES (10,'Ernesto85');
INSERT INTO `PuliziaUtente` VALUES (12,'Ernesto85');
INSERT INTO `PuliziaUtente` VALUES (12,'Saverio65');

-- -----
-- Table structure for `PULIZIAZONA`
-- -----
DROP TABLE IF EXISTS `PuliziaZona`;
CREATE TABLE `PuliziaZona` (
  `Pulizia` int(11) NOT NULL,
  `Zona` int(11) NOT NULL,
  PRIMARY KEY (`Pulizia`,`Zona`),
  FOREIGN KEY (`Pulizia`) REFERENCES `Pulizia` (`IdPulizia`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Zona`) REFERENCES `Zona` (`IdZona`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
-- Records of `PULIZIAZONA`
-- -----
BEGIN;
INSERT INTO `PuliziaZona` VALUES (1,1);
INSERT INTO `PuliziaZona` VALUES (1,2);
INSERT INTO `PuliziaZona` VALUES (1,3);
INSERT INTO `PuliziaZona` VALUES (1,4);
INSERT INTO `PuliziaZona` VALUES (2,1);
INSERT INTO `PuliziaZona` VALUES (2,2);

```



```

INSERT INTO `PuliziaZona` VALUES (2,3);
INSERT INTO `PuliziaZona` VALUES (2,4);
INSERT INTO `PuliziaZona` VALUES (3,1);
INSERT INTO `PuliziaZona` VALUES (3,2);
INSERT INTO `PuliziaZona` VALUES (3,3);
INSERT INTO `PuliziaZona` VALUES (3,4);
INSERT INTO `PuliziaZona` VALUES (4,1);
INSERT INTO `PuliziaZona` VALUES (4,2);
INSERT INTO `PuliziaZona` VALUES (4,3);
INSERT INTO `PuliziaZona` VALUES (4,4);
INSERT INTO `PuliziaZona` VALUES (5,1);
INSERT INTO `PuliziaZona` VALUES (5,2);
INSERT INTO `PuliziaZona` VALUES (5,3);
INSERT INTO `PuliziaZona` VALUES (5,4);
INSERT INTO `PuliziaZona` VALUES (6,1);
INSERT INTO `PuliziaZona` VALUES (6,2);
INSERT INTO `PuliziaZona` VALUES (6,3);
INSERT INTO `PuliziaZona` VALUES (6,4);
INSERT INTO `PuliziaZona` VALUES (7,1);
INSERT INTO `PuliziaZona` VALUES (7,2);
INSERT INTO `PuliziaZona` VALUES (7,3);
INSERT INTO `PuliziaZona` VALUES (7,4);
INSERT INTO `PuliziaZona` VALUES (8,1);
INSERT INTO `PuliziaZona` VALUES (8,2);
INSERT INTO `PuliziaZona` VALUES (8,3);
INSERT INTO `PuliziaZona` VALUES (8,4);
INSERT INTO `PuliziaZona` VALUES (9,1);
INSERT INTO `PuliziaZona` VALUES (9,2);
INSERT INTO `PuliziaZona` VALUES (9,3);
INSERT INTO `PuliziaZona` VALUES (9,4);
INSERT INTO `PuliziaZona` VALUES (10,1);
INSERT INTO `PuliziaZona` VALUES (10,2);
INSERT INTO `PuliziaZona` VALUES (10,3);
INSERT INTO `PuliziaZona` VALUES (10,4);
INSERT INTO `PuliziaZona` VALUES (11,1);
INSERT INTO `PuliziaZona` VALUES (11,2);
INSERT INTO `PuliziaZona` VALUES (11,3);
INSERT INTO `PuliziaZona` VALUES (11,4);
INSERT INTO `PuliziaZona` VALUES (12,1);
INSERT INTO `PuliziaZona` VALUES (12,2);
INSERT INTO `PuliziaZona` VALUES (12,3);
INSERT INTO `PuliziaZona` VALUES (12,4);
COMMIT;

```

```

-- -----
-- Table structure for `RUBRICA`
-- -----
DROP TABLE IF EXISTS `Rubrica`;
CREATE TABLE `Rubrica` (
  `IdTel` int(11) NOT NULL AUTO_INCREMENT,
  `Numero` varchar(20) NOT NULL,
  `Tipologia` varchar(20) NOT NULL,
  PRIMARY KEY (`IdTel`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `RUBRICA`
-- -----
BEGIN;
INSERT INTO `Rubrica` VALUES (1,'1234 1234567','Cellulare');
INSERT INTO `Rubrica` VALUES (2,'1234 1234567','Fisso');
INSERT INTO `Rubrica` VALUES (3,'1234 1234567','Cellulare');
INSERT INTO `Rubrica` VALUES (4,'1234 1234567','Cellulare');
INSERT INTO `Rubrica` VALUES (5,'1234 1234567','Cellulare');
INSERT INTO `Rubrica` VALUES (6,'1234 1234567','Fax');
INSERT INTO `Rubrica` VALUES (7,'1234 1234567','Fisso');
INSERT INTO `Rubrica` VALUES (8,'1234 1234567','Fax');
COMMIT;

```

```

-- -----
-- Table structure for `RUBRICAABITAZIONE`
-- -----
DROP TABLE IF EXISTS `RubricaAbitazione`;
CREATE TABLE `RubricaAbitazione` (
  `Tel` int(11) NOT NULL,
  `Abitazione` int(11) NOT NULL,
  PRIMARY KEY (`Tel`),
  FOREIGN KEY (`Tel`) REFERENCES `Rubrica`(`IdTel`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Abitazione`) REFERENCES `Abitazione`(`IdAbitazione`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `RUBRICAABITAZIONE`
-- -----

```

```

-----
BEGIN;
INSERT INTO `RubricaAbitazione` VALUES (7,1);
COMMIT;

-----
-- Table structure for `RUBRICAIMPRESA`
-----
DROP TABLE IF EXISTS `RubricaImpresa`;
CREATE TABLE `RubricaImpresa` (
  `Tel` int(11) NOT NULL,
  `Impresa` varchar(20) NOT NULL,
  PRIMARY KEY (`Tel`),
  FOREIGN KEY (`Tel`) REFERENCES `Rubrica`(`IdTel`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Impresa`) REFERENCES `ImpresaPulizia`(`PartitaIva`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of `RUBRICAIMPRESA`
-----
BEGIN;
INSERT INTO `RubricaImpresa` VALUES (8,'IT12345678910');
COMMIT;

-----
-- Table structure for `RUBRICASEDE`
-----
DROP TABLE IF EXISTS `RubricaSede`;
CREATE TABLE `RubricaSede` (
  `Tel` int(11) NOT NULL,
  `Sede` int(11) NOT NULL,
  PRIMARY KEY (`Tel`),
  FOREIGN KEY (`Tel`) REFERENCES `Rubrica`(`IdTel`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Sede`) REFERENCES `Sede`(`IdSede`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of `RUBRICASEDE`
-----
BEGIN;
INSERT INTO `RubricaSede` VALUES (6,1);
COMMIT;

-----
-- Table structure for `RUBRICAUTENTE`
-----
DROP TABLE IF EXISTS `RubricaUtente`;
CREATE TABLE `RubricaUtente` (
  `Tel` int(11) NOT NULL,
  `Utente` varchar(20) NOT NULL,
  PRIMARY KEY (`Tel`),
  FOREIGN KEY (`Tel`) REFERENCES `Rubrica`(`IdTel`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Utente`) REFERENCES `Utente`(`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of `RUBRICAUTENTE`
-----
BEGIN;
INSERT INTO `RubricaUtente` VALUES (1,'Ermenegilda50');
INSERT INTO `RubricaUtente` VALUES (2,'Saverio65');
INSERT INTO `RubricaUtente` VALUES (3,'Ernesto85');
INSERT INTO `RubricaUtente` VALUES (4,'Mario75');
INSERT INTO `RubricaUtente` VALUES (5,'Giuseppe70');
COMMIT;

-----
-- Table structure for `SEDE`
-----
DROP TABLE IF EXISTS `Sede`;
CREATE TABLE `Sede` (
  `IdSede` int(11) NOT NULL AUTO_INCREMENT,
  `Email` varchar(40) NOT NULL,
  `Posizione` int(11) NOT NULL,
  `Agenzia` varchar(20) NOT NULL,
  PRIMARY KEY (`IdSede`),
  FOREIGN KEY (`Agenzia`) REFERENCES `Agenzia`(`PartitaIva`) ON DELETE NO ACTION ON UPDATE CASCADE,
  FOREIGN KEY (`Posizione`) REFERENCES `Luogo`(`IdLuogo`) ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
--   Records of `SEDE`
-- -----

BEGIN;
INSERT INTO `Sede` VALUES (1,'pisa@stilecasa.com',2,'IT01114601006');
INSERT INTO `Sede` VALUES (2,'roma@stilecasa.com',4,'IT01114601006');
INSERT INTO `Sede` VALUES (3,'agrigento@stilecasa.com',11,'IT01114601006');
INSERT INTO `Sede` VALUES (4,'nizza@immobiliare.com',8,'FR12345678912');
COMMIT;

-- -----
--   Table structure for `SOGGIORNO`
-- -----

DROP TABLE IF EXISTS `Soggiorno`;
CREATE TABLE `Soggiorno` (
  `IdSoggiorno` int(11) NOT NULL AUTO_INCREMENT,
  `DataInizio` datetime NOT NULL,
  `DataFine` datetime NOT NULL,
  `Quota` int(11) NOT NULL,
  `Zona` int(11) NOT NULL,
  `Utente` varchar(20) NOT NULL,
  PRIMARY KEY (`IdSoggiorno`),
  FOREIGN KEY (`Zona`) REFERENCES `Zona` (`IdZona`) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`Utente`) REFERENCES `Utente` (`Username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
--   Records of `SOGGIORNO`
-- -----

BEGIN;
INSERT INTO `Soggiorno` VALUES (1,'2010-01-01 00:00:00','2011-01-01 00:00:00',2000,3,'Saverio65');
INSERT INTO `Soggiorno` VALUES (2,'2013-01-01 00:00:00','2014-01-01 00:00:00',3000,3,'Saverio65');
INSERT INTO `Soggiorno` VALUES (5,'2012-01-01 00:00:00','2014-01-01 00:00:00',5000,4,'Ernesto85');
COMMIT;

-- -----
--   Table structure for `SPESA`
-- -----

DROP TABLE IF EXISTS `Spesa`;
CREATE TABLE `Spesa` (
  `IdSpesa` int(11) NOT NULL AUTO_INCREMENT,
  `Descrizione` varchar(100) DEFAULT NULL,
  `DataScadenza` date NOT NULL,
  `DataPagata` date DEFAULT NULL,
  `Importo` int(11) NOT NULL,
  `Mora` int(11) DEFAULT NULL,
  `Abitazione` int(11) NOT NULL,
  PRIMARY KEY (`IdSpesa`),
  FOREIGN KEY (`Abitazione`) REFERENCES `Abitazione` (`IdAbitazione`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
--   Records of `SPESA`
-- -----

BEGIN;
INSERT INTO `Spesa` VALUES (1,'Affitto','2011-01-01','2011-01-01',2000,0,1);
INSERT INTO `Spesa` VALUES (2,'Affitto','2014-01-01',NULL,3000,NULL,1);
INSERT INTO `Spesa` VALUES (3,'Affitto','2014-01-01','2013-01-01',5000,0,1);
INSERT INTO `Spesa` VALUES (4,'Acqua','2013-02-01','2013-01-20',100,0,1);
INSERT INTO `Spesa` VALUES (5,'Luce','2013-02-01','2013-01-25',200,0,1);
INSERT INTO `Spesa` VALUES (6,'Gas','2013-02-01','2013-01-27',100,0,1);
COMMIT;

-- -----
--   Table structure for `UTENTE`
-- -----

DROP TABLE IF EXISTS `Utente`;
CREATE TABLE `Utente` (
  `Username` varchar(20) NOT NULL,
  `Password` varchar(40) NOT NULL,
  `Email` varchar(40) NOT NULL,
  `Afferenza` int(11) DEFAULT NULL,
  PRIMARY KEY (`Username`),
  FOREIGN KEY (`Afferenza`) REFERENCES `Sede` (`IdSede`) ON DELETE SET NULL ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- -----
--   Records of `UTENTE`
-- -----

BEGIN;

```

```

INSERT INTO `Utente` VALUES ('Ermenegilda50','6f3240543cede5da63dc8d44bba868ac','ermenegilda.turchesi@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Ernesto85','6f3240543cede5da63dc8d44bba868ac','ernesto.gialli@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Francesca45','6f3240543cede5da63dc8d44bba868ac','francesca.galli@gmail.com',3);
INSERT INTO `Utente` VALUES ('Gennaro40','6f3240543cede5da63dc8d44bba868ac','gennaro.arancioni@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Giuseppe70','6f3240543cede5da63dc8d44bba868ac','giuseppe.verdi@gmail.com',2);
INSERT INTO `Utente` VALUES ('John80','6f3240543cede5da63dc8d44bba868ac','john.doe@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Lucrezia90','6f3240543cede5da63dc8d44bba868ac','lucrezia.azzurri@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Mario75','6f3240543cede5da63dc8d44bba868ac','mario.rossi@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Remi55','6f3240543cede5da63dc8d44bba868ac','remi.martell@gmail.com',NULL);
INSERT INTO `Utente` VALUES ('Saverio65','6f3240543cede5da63dc8d44bba868ac','saverio.orsucci@gmail.com',NULL);
COMMIT;

```

```

-- -----
-- Table structure for `ZONA`
-- -----

```

```

DROP TABLE IF EXISTS `Zona`;
CREATE TABLE `Zona` (
  `IdZona` int(11) NOT NULL AUTO_INCREMENT,
  `Tipologia` varchar(20) NOT NULL,
  `Abitazione` int(11) NOT NULL,
  PRIMARY KEY (`IdZona`),
  FOREIGN KEY (`Abitazione`) REFERENCES `Abitazione` (`IdAbitazione`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

-- -----
-- Records of `ZONA`
-- -----

```

```

BEGIN;
INSERT INTO `Zona` VALUES (1,'Cucina',1);
INSERT INTO `Zona` VALUES (2,'Bagno',1);
INSERT INTO `Zona` VALUES (3,'Camera',1);
INSERT INTO `Zona` VALUES (4,'Camera',1);
INSERT INTO `Zona` VALUES (5,'Giardino',1);
INSERT INTO `Zona` VALUES (6,'Cucina',2);
INSERT INTO `Zona` VALUES (7,'Bagno',2);
INSERT INTO `Zona` VALUES (8,'Camera',2);
INSERT INTO `Zona` VALUES (9,'Camera',2);
INSERT INTO `Zona` VALUES (10,'Area Condominiale',2);
COMMIT;

```

```

SET FOREIGN_KEY_CHECKS = 1;

```

Creazione trigger per gestione di una ridondanza

```
-- -----
-- Trigger structure for `RIDGIARDINOINS`
-- -----
DROP TRIGGER IF EXISTS RidGiardinoINS;
DELIMITER $$

CREATE TRIGGER RidGiardinoINS
AFTER INSERT ON `Zona` FOR EACH ROW
BEGIN

IF (NEW.Tipologia = 'Giardino')
THEN
UPDATE Abitazione
SET Giardino = 'Y'
WHERE IdAbitazione = NEW.Abitazione;
END IF;

END$$
DELIMITER ;

-- -----
-- Trigger structure for `RIDGIARDINOUP`
-- -----
DROP TRIGGER IF EXISTS RidGiardinoUP;
DELIMITER $$

CREATE TRIGGER RidGiardinoUP
AFTER UPDATE ON `Zona` FOR EACH ROW
BEGIN

IF (NEW.Tipologia = 'Giardino')
THEN
UPDATE Abitazione
SET Giardino = 'Y'
WHERE IdAbitazione = NEW.Abitazione;
END IF;

IF (OLD.Tipologia = 'Giardino')
THEN
UPDATE Abitazione
SET Giardino = 'N'
WHERE IdAbitazione = OLD.Abitazione AND 0 =(
SELECT COUNT(*)
FROM Zona
WHERE Abitazione = OLD.Abitazione AND Tipologia='Giardino');

END IF;

END$$
DELIMITER ;

-- -----
-- Trigger structure for `RIDGIARDINODEL`
-- -----
DROP TRIGGER IF EXISTS RidGiardinoDel;
DELIMITER $$

CREATE TRIGGER RidGiardinoDel
AFTER DELETE ON `Zona` FOR EACH ROW
BEGIN

IF (OLD.Tipologia = 'Giardino')
THEN
UPDATE Abitazione
SET Giardino = 'N'
WHERE IdAbitazione = OLD.Abitazione AND 0 =(
SELECT COUNT(*)
FROM Zona
WHERE Abitazione = OLD.Abitazione AND Tipologia='Giardino');

END IF;

END$$
DELIMITER ;
```

Creazione Trigger per gestione di un vincolo di integrità generico

```
-- -----
-- Trigger structure for `CHKDATASCADENZAINS`
-- -----
DROP TRIGGER IF EXISTS ChkDataScadenzaINS;
DELIMITER $$

CREATE TRIGGER ChkDataScadenzaINS
BEFORE INSERT ON `Bene` FOR EACH ROW
BEGIN

DECLARE msg varchar(255);

IF (NEW.Tipo <> 'Alimentare' AND NEW.DataScadenza IS NOT NULL)
THEN
SET msg = CONCAT('DataScadenza non può contenere alcun valore per gli attributi di Tipo ',CAST(NEW.Tipo AS CHAR));
SIGNAL SQLSTATE '45100' SET MESSAGE_TEXT = msg;
END IF;

IF (NEW.Tipo = 'Alimentare' AND NEW.DataScadenza IS NULL)
THEN
SET msg = 'DataScadenza non può contenere il valore NULL per gli attributi di Tipo Alimentare';
SIGNAL SQLSTATE '45100' SET MESSAGE_TEXT = msg;
END IF;

END$$
DELIMITER ;

-- -----
-- Trigger structure for `CHKDATASCADENZAUP`
-- -----
DROP TRIGGER IF EXISTS ChkDataScadenzaUP;
DELIMITER $$

CREATE TRIGGER ChkDataScadenzaUP
BEFORE UPDATE ON `Bene` FOR EACH ROW
BEGIN

DECLARE msg varchar(255);

IF (NEW.Tipo <> 'Alimentare' AND NEW.DataScadenza IS NOT NULL)
THEN
SET msg = CONCAT('DataScadenza non può contenere alcun valore per gli attributi di Tipo ',CAST(NEW.Tipo AS CHAR));
SIGNAL SQLSTATE '45100' SET MESSAGE_TEXT = msg;
END IF;

IF (NEW.Tipo = 'Alimentare' AND NEW.DataScadenza IS NULL)
THEN
SET msg = 'DataScadenza non può contenere il valore NULL per gli attributi di Tipo Alimentare';
SIGNAL SQLSTATE '45100' SET MESSAGE_TEXT = msg;
END IF;

END$$
DELIMITER ;
```

Considerazioni su possibili miglioramenti

Di seguito si riportano alcune considerazioni su alcuni miglioramenti che potrebbero interessare la base di dati ma che per motivi tempistici e per la natura didattica di questo elaborato non sono stati implementati.

- All'inserimento di una nuova tupla nella tabella "Soggiorno" sarebbe opportuno che un trigger controllasse che la tipologia della zona interessata sia di tipologia "Camera".
- Si potrebbe pensare di realizzare un trigger automatico che nel momento della stipula di un nuovo contratto di soggiorno generi automaticamente una nuova spesa per l'utente contraente con "Importo" pari a "Quota" e "DataScadenza" uguale a "DataFine".
- Un trigger automatico potrebbe stabilire il valore dell'attributo "Mora" nella tabella "Spesa" in base all'inserimento del campo "DataPagata".
- Per quanto riguarda l'entità Zona si potrebbe pensare di inserire un ulteriore attributo "Stato" che assuma valore diverso da NULL (e quindi abbia significato) quando la tipologia è camera e che indichi se al momento dell'interrogazione una determinata camera è occupata o meno. Essendo l'attributo "Stato" deducibile dall'entità "Soggiorno" si dovrebbe necessariamente pensare ad un trigger temporale che gestisca automaticamente l'aggiornamento della ridondanza.
- L'entità Sede potrebbe avere un attributo ridondante Luogo con indicato all'interno il comune dove è posizionata la sede per velocizzare le ricerche.
- Per come è strutturato lo schema ER le zone possono assumere anche tipologia "giardino" e "area condominiale". Questo è corretto finché non si trovano due abitazioni registrate all'interno del RMS che condividono il solito giardino o la solita area condominiale. In tal caso non c'è più una relazione "uno a molti con partecipazione obbligatoria" tra abitazione e zona ma bensì una relazione "molti a molti". Questo implicherebbe la costruzione di una nuova tabella al cui interno si inseriscono due attributi: l'identificativo della zona e quello dell'abitazione di cui fa parte. Per semplicità didattica si assume quindi che le zone di tipologia "area condominiale" e "giardino" di una certa abitazione non possano essere condivise con altre abitazioni registrate al RMS.

