**Abstract**

L’agenzia immobiliare X vuole automatizzare parte della propria gestione tramite una base di dati, precisamente la sezione riguardante la registrazione dello storico dei contratti portati a termine, i dati riguardanti le abitazioni vendute e la gestione del proprio personale nelle varie sedi e il loro rapporto con la propria clientela. Tale base di dati verrà inoltre utilizzata a fini statistici per l’eventuale espansione dell’agenzia.

Tale base di dati dovrà essere gestita da un pannello di controllo accessibile via browser.

**Descrizione dei requisiti**

Si vuole creare una base di dati che gestisca prima di tutto il personale e le sedi e secondariamente i rapporti tra il personale e la clientela, nonché lo storico dei contratti portati a compimento e in atto.

Ogni sede è caratterizzata dai seguenti campi:

* La città in cui è presente la sede
* Il numero di sede, all’interno della città

Ogni sede ha dei dipendenti, ognuno dei quali è definito dalle seguenti caratteristiche:

* Matricola, univoca in tutta l’agenzia
* Il nome
* Il cognome
* Il Numero di Telefono

Inoltre ogni dipendente può essere classificato come:

* Direttore di sede
* Agenti immobiliari
* Personale di segreteria, il quale sarà dedicato a fissare appuntamenti tra clienti e agenti immobiliari, di cui ci interessano data, ora, personale e cliente coinvolti.

Di ogni cliente ci interessano:

* Codice Fiscale
* Nome
* Cognome
* Data di Nascita

Inoltre ogni cliente può essere:

* Cedente di un immobile
* Cessionario

Di ogni abitazione trattata, ci interessano:

* L’indirizzo

Inoltre, altri attributi di interesse possono essere:

* Nel caso l’abitazione fosse un appartamento: il numero d’interno
* Nel caso fosse una casa essa potrà essere classificata come:
  + Casa a schiera
  + Casa Singola
  + Casa multifamiliare, nel caso ci interessa anche la sezione (18/A, 18/B, …)
  + Altri tipi non meglio specificati di cui non ci interessano ulteriori informazioni.

I contratti possono essere classificati secondo due criteri:

* Il loro stato
  + Portati a termine - in tal caso ci interessa la data di accettazione
  + In corso
* La loro tipologia
  + Di Compravendita - Di cui ci interessa il prezzo di vendita ed Il valore dell’eventuale mutuo richiesto per l’acquisto
  + Di Locazione (Il cosiddetto affitto) - Di cui ci interessa la rata mensile

Inoltre, riguardo ai contratti, ci interessano le seguenti informazioni aggiuntive:

* La data di stipulazione
* L’agente proponente
* I clienti coinvolti - con l’accorgimento che più clienti possono acquistare o vendere una stessa abitazione (comproprietà).

**Regole Aziendali da tenere in considerazione:**

1. Un dirigente di una sede deve afferire a tale sede
2. Una sede ha un solo direttore ed un direttore può dirigere una sola sede
3. Un agente non può avere due appuntamenti che hanno luogo nella stessa data ed ora

**Analisi dei Requisiti**

Sede:

* Numero
* Città

Dipendente

* Matricola
* Nome
* Cognome
* Numero Telefono

Cliente

* Codice Fiscale
* Nome
* Cognome
* Data di Nascita

Abitazione

* Indirizzo
* Sezione (Se multifamiliare)
* Interno (Se Appartamento)

Contratto

* Tipologia (Compravendita/locazione)
* Stato (In corso/Terminato)
* Data di stipulazione
* Valore Mutuo
* Agente proponente
* Clienti coinvolti

**Progettazione Concettuale**

Entità e attributi

|  |
| --- |
| Frasi Riguardanti Sede:  Ogni sede è caratterizzata dai seguenti campi:   * La città in cui è presente la sede * Il numero di sede, all’interno della città |

Sede:

* Numero: smallint

Città (Entità di utilità):

* Nome: string(20)
* Provincia: string(20)
* Regione: string(20)

|  |
| --- |
| Frasi Riguardanti Dipendente:  Ogni sede ha dei dipendenti, ognuno dei quali è definito dalle seguenti caratteristiche:   * Matricola, univoca in tutta l’agenzia * Il nome * Il cognome * Il numero di telefono   Inoltre ogni dipendente può essere classificato come:   * Direttore di sede * Agenti immobiliari * Personale di segreteria [...] |

Dipendente:

* Matricola: string(7)
* Nome: string(20)
* Cognome: string(20)
* Numero Telefono: string(10)

Direttore

Agente Immobiliare

Personale di Segreteria

|  |
| --- |
| Frasi riguardanti Appuntamento:   * [...] il quale sarà dedicato a fissare appuntamenti tra clienti e agenti immobiliari, di cui ci interessano data, ora, personale e cliente coinvolti. |

Appuntamento:

* Data: date
* Ora: time

|  |
| --- |
| Frasi Riguardanti Cliente:  Di ogni cliente ci interessano:   * Codice Fiscale * Nome * Cognome * Data di Nascita   Inoltre ogni cliente può essere:   * Cedente di un immobile * Cessionario |

Cliente:

* Codice Fiscale: string(16)
* Nome: string(20)
* Cognome: string(20)
* Data\_Nascita: date

Cedente

Cessionario

|  |
| --- |
| Frasi Riguardanti Abitazione:  Di ogni abitazione trattata, ci interessano:   * L’indirizzo   Inoltre, altri attributi di interesse possono essere:   * Nel caso l’abitazione fosse un appartamento: il numero d’interno * Nel caso fosse una casa essa potrà essere classificata come:   + Casa a schiera   + Casa Singola   + Casa multifamiliare, nel caso ci interessa anche la sezione (18/A, 18/B, …)   + Altri tipi non meglio specificati di cui non ci interessano ulteriori informazioni. |

Abitazione:

* Codice: string(7)
* Via: string(20)
* Numero Civico: smallint

Casa Multifamiliare:

* Sezione: string(2)

Appartamento:

* Interno: smallint

|  |
| --- |
| Frasi Riguardanti Contratto:  I contratti possono essere classificati secondo due criteri:   * Il loro stato   + Portati a termine - in tal caso ci interessa la data di accettazione   + In corso * La loro tipologia   + Di Compravendita - Di cui ci interessa il prezzo di vendita e Il valore dell’eventuale mutuo richiesto per l’acquisto   + Di Locazione (Il cosiddetto affitto) - Di cui ci interessa la rata mensile   Inoltre, riguardo ai contratti, ci interessano le seguenti informazioni aggiuntive:   * La data di stipulazione * L’agente proponente * I clienti coinvolti - con l’accorgimento che più clienti possono acquistare o vendere una stessa abitazione (comproprietà). |

Contratto:

* Data di Proposta: date

Contratto di Compravendita:

* Prezzo di vendita: integer
* Valore Mutuo: integer

Contratto di Locazione:

* Rata mensile: integer

Contratto Completato:

* Data di firma: date

Relazioni

*Direttore - Sede*: Direzione

* Un Direttore Dirige una sola sede
* Ogni sede ha un solo direttore

*Dipendente - Sede*: Afferenza

* Ogni sede ha più dipendenti
* Ogni dipendente lavora in una sola sede

*Sede - Città*: Presenza

* Ogni sede è presente in una sola città
* Ogni città può avere zero o più sedi

*Abitazione - Città*: Presenza

* Ogni abitazione di nostro interesse è presente in una sola città
* Ogni città può avere zero o più abitazioni di nostro interesse

*Segretario - Appuntamento*: Creazione

* Ogni segretario può creare più appuntamenti o nessuno
* Ogni appuntamento può essere fissato da un solo segretario

*Cliente - Appuntamento:* Presenza

* Ogni appuntamento è fissato con più clienti
* Un cliente può avere più appuntamenti

*Appuntamento - Agente:* Presenza

* Un appuntamento è tenuto da un solo agente
* Un agente partecipa a più appuntamenti

*Agente - Contratto:* Proposta

* Un agente propone più contratti
* Un contratto è proposto da un solo agente

*Contratto - Abitazione:* Riguardante

* Un contratto riguarda una sola abitazione
* Un’abitazione può avere più contratti all’attivo

*Cedente - Contratto:* Cessione

* Un cedente fa parte uno o più contratti di cessione (riguardanti diverse abitazioni)
* Un contratto di cessione ha più cedenti

*Cessionario - Contratto*: Acquisizione

* Un cessionario fa parte di più contratti di acquisizione diversi
* Un contratto di acquisizione ha più cessionari

Generalizzazioni:

Cliente:

* Cliente cedente
* Cliente Cessionario

Dipendente:

* Direttore
* Segretario
* Agente

Abitazione:

* Appartamento
* Casa
  + A schiera
  + Multifamiliare
  + Singola

Contratto:

* Di Compravendita
* Di Locazione

oppure

* Completato
* In Corso

Diagramma ER a pagina 20.

**Progettazione Logica:**

Ci baseremo sulle seguenti operazioni comuni:

1. Creazione di un appuntamento - ~25 volte al giorno
2. Conteggio dei contratti completati da un certo agente in un certo arco di tempo: ~20 volte al mese
3. Conteggio dei contratti completati in una sede in un certo arco di tempo recente: ~1 volta a settimana
4. Conteggio dei contratti (in corso e completati, cioè proposti) in una data città: ~1 volta al giorno
5. Individuazione del direttore di una sede e conteggio dei dipendenti: ~2 volte al mese
6. Proposta di un nuovo contratto: ~20 volte al giorno

**Analisi delle Ridondanze**

Vi è un dato ridondante all’interno dell’entità Città: Contratti\_Proposti. Che può essere ricavato da altri dati del database. Le operazioni influenzate da tale ridondanza sono le operazioni 4 e 6

**Analisi Prestazionale**

|  |  |  |
| --- | --- | --- |
| **Tavola Delle Operazioni** | | |
| Operazione | Tipo | Frequenza |
| Operazione 1 | Interattivo | 50 al Giorno |
| Operazione 2 | Interattivo | 20 al Mese |
| Operazione 3 | Interattivo | 1 a Settimana |
| Operazione 4 | Batch | 1 al Giorno |
| Operazione 5 | Batch | 2 al Mese |
| Operazione 6 | Interattivo | 20 al Giorno |

Inoltre faremo riferimento alla seguente Tavola dei volumi:

|  |  |  |
| --- | --- | --- |
| **Tavola Dei Volumi** | | |
| Concetto | Tipo | Volume |
| Dipendente | Entità | 260 |
| Direttore | Entità | 20 |
| Segretario | Entità | 40 |
| Agente | Entità | 200 |
| Sede | Entità | 20 |
| Città | Entità | 10 |
| Abitazione | Entità | 1000 |
| Casa | Entità | 350 |
| Singola | Entità | 105 |
| Multifamiliare | Entità | 210 |
| A schiera | Entità | 35 |
| Appartamento | Entità | 650 |
| Contratto | Entità | 2000 |
| Completato | Entità | 1900 |
| In corso | Entità | 100 |
| Compravendita | Entità | 600 |
| Locazione | Entità | 1400 |
| Cliente | Entità | 500 |
| Cedente | Entità | 250 |
| Cessionario | Entità | 250 |
| Appuntamento | Entità | 1000 |
| Direzione | Relazione | 20 |
| Creazione | Relazione | 1000 |
| Presenza (Appuntamento-Cliente) | Relazione | 500000 |
| Acquisizione | Relazione | 1000 |
| Cessione | Relazione | 1000 |
| Proposta | Relazione | 2000 |
| Presenza (Appuntamento-Agente) | Relazione | 1000 |
| Riguardante | Relazione | 2000 |
| Presenza (Abitazione-Città) | Relazione | 1000 |
| Presenza (Sede-Città) | Relazione | 20 |
| Afferenza | Relazione | 260 |

**Commenti:**

* Consideriamo una media di 2 sedi per città, sparse per un totale di 10 città
* Vi sono mediamente 2 segretari per sede
* Ogni sede ha mediamente 10 agenti
* Consideriamo una media di 100 abitazioni acquistate o vendute per città nel ciclo vitale del DB di cui
  + ~65% sono Appartamenti
  + ~35% sono Case vere e proprie, di cui
    - ~10% sono a schiera
    - ~30% sono singole
    - ~60% sono multifamiliari (bifamiliari, trifamiliari, ecc…)
* Consideriamo che mediamente ogni abitazione è oggetto di 2 contratti, di cui:
  + Il 70% è di locazione
  + Il 30% è di compravendita
* Consideriamo inoltre che mediamente solo il 5% dei contratti in un dato momento risulta attivo (non ratificato da entrambe le parti)
* Consideriamo che il 50% dei clienti farà la parte del cedente, mentre il restante 50% sarà cessionario.
* Consideriamo che circa 50 persone per città siano clienti dell’azienda in oggetto
* Prendiamo in considerazione che mediamente ogni cliente ha bisogno di 2 appuntamenti prima di richiedere una proposta di contratto
* Consideriamo che gli acquisti/vendite in comproprietà siano una parte molto trascurabile del volume d’affari, avendo mediamente solo 1 compratore ed 1 venditore per contratto

**Analisi prestazionale:**

Date le operazioni e i volumi di cui sopra, possiamo procedere a costruire le tavole degli accessi, considerando gli accessi in scrittura con peso doppio rispetto a quelli in lettura:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operazione 4 In Presenza di ridondanza** | | | | |
| Concetto | Costrutto | Accessi | Tipo di Accesso | Peso |
| Città | Entità | 1 | Lettura | 1 |
| Peso di una operazione | | | | 1 |
| Numero di operazioni (al mese) | | | | 30 |
| **Peso Totale** | | | | **30** |
| **Operazione 6 In presenza di ridondanza** | | | | |
| Concetto | Costrutto | Accessi | Tipo di Accesso | Peso |
| Contratto | Entità | 1 | Scrittura | 2 |
| Acquisizione | Relazione | 1 | Scrittura | 2 |
| Cessione | Relazione | 1 | Scrittura | 2 |
| Cedente | Entità | 1 | Lettura | 1 |
| Cessionario | Entità | 1 | Lettura | 1 |
| Proposta | Relazione | 1 | Scrittura | 2 |
| Agente | Entità | 1 | Lettura | 1 |
| Riguardante | Relazione | 1 | Scrittura | 2 |
| Abitazione | Entità | 1 | Lettura | 1 |
| Presenza | Relazione | 1 | Lettura | 1 |
| Città | Entità | 1 | Scrittura | 2 |
| Peso Operazione | | | | 17 |
| Numero di operazioni (al mese) | | | | 20 |
| **Peso Totale** | | | | **340** |
| **Operazione 4 in assenza di Ridondanza** | | | | |
| Concetto | Costrutto | Accessi | Tipo di Accesso | Peso |
| Città | Entità | 1 | Lettura | 1 |
| Presenza | Relazione | 1 | Lettura | 1 |
| Abitazione | Entità | 100 | Lettura | 100 |
| Riguardante | Relazione | 200 | Lettura | 200 |
| Peso Operazione | | | | 302 |
| Numero di operazioni (al mese) | | | | 30 |
| **Peso Totale** | | | | **9060** |
| **Operazione 6 In assenza di ridondanza** | | | | |
| Contratto | Entità | 1 | Scrittura | 2 |
| Acquisizione | Relazione | 1 | Scrittura | 2 |
| Cessione | Relazione | 1 | Scrittura | 2 |
| Cedente | Entità | 1 | Lettura | 1 |
| Cessionario | Entità | 1 | Lettura | 1 |
| Proposta | Relazione | 1 | Scrittura | 2 |
| Agente | Entità | 1 | Lettura | 1 |
| Riguardante | Relazione | 1 | Scrittura | 2 |
| Peso Operazione | | | | 13 |
| Numero di operazioni (al mese) | | | | 20 |
| **Peso totale** | | | | **260** |

Confronto delle operazioni:

|  |  |  |
| --- | --- | --- |
|  | **Con Ridondanza** | **Senza Ridondanza** |
| **Operazione 4** | 30 | 9060 |
| **Operazione 6** | 340 | 260 |

Com’è possibile vedere, vi è un forte vantaggio nel mantenere la ridondanza, soprattutto per l’operazione 4.

Eliminazione delle generalizzazioni:

Generalizzazione Totale Contratto - Di Compravendita/Di Locazione:

Si decide di integrare questa generalizzazione verso il padre

* unendo gli attributi Prezzo e Rata in un solo attributo prezzo
* Portando “valore mutuo” al padre
* creando un attributo “tipo”

Generalizzazione Totale Contratto - Completato/ In Corso

Si decide anche qui di integrare verso il padre

* Lo stato del contratto verrà identificato dalla presenza o meno della data di firma/ratifica, che ora potrà avere valori nulli.

Generalizzazione Parziale Casa - A schiera ….

Si decide di integrare verso il padre (casa) che acquisirà il campo Sezione

Generalizzazione Totale Abitazione - Casa/Appartamento

Anche qui si decide di integrare verso il padre, che acquisirà un attributo Sezione\_o\_Interno

Generalizzazione Totale Dipendente - Direttore + Segretario + Agente

Qui preferiamo mantenere la struttura della generalizzazione, creando 3 relationship che collegheranno le figlie al padre.

Generalizzazione Totale Cliente - Cedente + Cessionario

Qui decidiamo di integrare verso il padre, dato che non vi sono particolari differenze tra cedente e cessionario, rimanendo con due relationship tra cliente e contratto che chiameremo “cedente” e “cessionario”.

**Scelta degli identificatori principali**

Gli identificatori principali saranno quelli inseriti nello schema ER, tranne per:

* la relazione “Contratto” nella quale abbiamo scelto di inserire un attributo “codice” a causa di un forte numero di chiavi esterne che porterebbero ad un appesantimento dello schema logico e relazionale.
* la relazione “Citta” nella quale abbiamo scelto di usare un “codice” per alleggerire le chiavi esterne delle relazioni Sede e Abitazione
* anche la relazione “Appuntamento” farà uso di un codice, per alleggerire la chiave esterna verso presenza

Diagramma ER Ristrutturato a pagina 21.

**Politiche Di Traduzione:**

* La relazione Dipendente - Afferenza - Sede verrà integrata in Dipendente
* La relazione Dipendente - Composizione - Direttore verrà integrata in Direttore
* La relazione Dipendente - Composizione - Agente verrà integrata in Agente
* La relazione Dipendente - Composizione - Segretario verrà integrata in Segretario
* La relazione Direttore - Direzione - Sede verrà integrata in Sede
* La relazione Sede - Presenza - Città verrà integrata in Sede
* La relazione Città - Presenza - Abitazione verrà integrata in Abitazione
* La relazione Contratto - Riguardante - Abitazione verrà integrata in Contratto
* La relazione Contratto - Proposta - Agente verrà integrata in Contratto
* La relazione Cessionario diverrà una relazione a sè
* La relazione Cedente diverrà una relazione a sè
* La relazione Cliente - Presenza - Appuntamento sarà reificata
* La relazione Appuntamento - Creazione - Segretario sarà integrata in Appuntamento
* La relazione Appuntamento - Presenza - Agente sarà integrata in Appuntamento

**Modello Relazionale:**

**Citta**(Codice, Nome, Provincia, Regione, Contratti\_Proposti)

**Abitazione**(Codice, Civico, Via, Sezione\_Interno, Citta, Provincia, Regione)

**Sede**(Numero, Citta, Via, Civico, Direttore)

**Dipendente**(Matricola, Nome, Cognome, Numero\_Telefono, Sede\_Afferenza, Citta\_Afferenza, Provincia\_Afferenza, Regione\_Afferenza)

**Direttore**(Dipendente)

**Agente**(Dipendente)

**Segretario**(Dipendente)

**Cliente**(Codice\_Fiscale, Nome, Cognome, Data\_Nascita)

**Contratto**(Codice, Data\_Proposta, Valore\_Mutuo, Tipo, Prezzo\_o\_Rata, Agente\_Proponente, Abitazione, Data\_Ratifica)

**Cedente**(Cliente, Contratto)

**Cessionario**(Cliente, Contratto)

**Appuntamento**(Codice, Data, Ora, Agente, Segretario)

**Presenza**(Appuntamento, Cliente)

Schema Logico a Pagina 22.

**Funzioni:**

1. **Dato un Codice Fiscale di un cliente ritorna il numero di contratti che ha sottoscritto e ratificato, sia di acquisizione che di cessione**

DROP FUNCTION IF EXISTS Contratti\_Ratificati;

DELIMITER |

CREATE FUNCTION Contratti\_Ratificati(CF VARCHAR(16))

RETURNS INT

DETERMINISTIC READS SQL DATA CONTAINS SQL

BEGIN

DECLARE Ced INT;

DECLARE Ces INT;

DECLARE Tot INT;

SELECT Count(\*) INTO Ced

FROM Cedente JOIN Contratto ON Contratto.Codice = Cedente.Contratto

WHERE Cedente.Cliente = CF AND Contratto.Data\_Ratifica IS NOT NULL;

SELECT Count(\*) INTO Ces

FROM Cessionario JOIN Contratto ON Contratto.Codice = Cessionario.Contratto

WHERE Cessionario.Cliente = CF AND Contratto.Data\_Ratifica IS NOT NULL;

SET Tot=Ced+Ces;

RETURN Tot;

END|

DELIMITER ;

1. **Funzione che dato un dipendente, ne ritorna il ruolo (Direttore, Segretario, Agente)**

DROP FUNCTION IF EXISTS RuoloDipendente;

DELIMITER |

CREATE FUNCTION RuoloDipendente(Matr VARCHAR(6))

RETURNS VARCHAR(20)

DETERMINISTIC READS SQL DATA CONTAINS SQL

BEGIN

DECLARE Ruolo VARCHAR(20);

DECLARE Num INT(3);

SELECT Count(\*) INTO Num

FROM Dipendente JOIN Agente ON Dipendente.Matricola = Agente.Dipendente

WHERE Dipendente.Matricola = Matr;

IF Num > 0 THEN

SET Ruolo="Agente";

END IF;

SELECT Count(\*) INTO Num

FROM Dipendente JOIN Segretario ON Dipendente.Matricola = Segretario.Dipendente

WHERE Dipendente.Matricola = Matr;

IF Num > 0 THEN

SET Ruolo="Segretario";

END IF;

SELECT Count(\*) INTO Num

FROM Dipendente JOIN Direttore ON Dipendente.Matricola = Direttore.Dipendente

WHERE Dipendente.Matricola = Matr;

IF Num > 0 THEN

SET Ruolo="Direttore";

END IF;

RETURN Ruolo;

END|

DELIMITER ;

1. **Funzione che dati un codice contratto ed un cliente, ritorna che ruolo ha avuto tale cliente in quel contratto (Cedente o Cessionario)**

DROP FUNCTION IF EXISTS RuoloCliente;

DELIMITER |

CREATE FUNCTION RuoloCliente(Id VARCHAR(16), Contr VARCHAR(6))

RETURNS VARCHAR(20)

DETERMINISTIC READS SQL DATA CONTAINS SQL

BEGIN

DECLARE Ruolo VARCHAR(20);

DECLARE Num INT;

SELECT Count(\*) INTO Num

FROM Cliente JOIN Cedente

ON Cedente.Cliente = Id

WHERE Cedente.Contratto = Contr;

IF Num > 0 THEN

SET Ruolo="Cedente";

END IF;

SELECT Count(\*) INTO Num

FROM Cliente JOIN Cessionario

ON Cessionario.Cliente = Id

WHERE Cessionario.Contratto = Contr;

IF Num > 0 THEN

SET Ruolo="Cessionario";

END IF;

RETURN Ruolo;

END|

DELIMITER ;

**Triggers:**

1. **Quando inserisco un nuovo contratto, la ridondanza in Città viene aggiornata**

DROP TRIGGER IF EXISTS ConteggioContratti;

DELIMITER |

CREATE TRIGGER ConteggioContratti

AFTER INSERT ON Contratto

FOR EACH ROW

BEGIN

UPDATE Citta SET

Contratti\_Proposti = Contratti\_Proposti + 1

WHERE Codice = (

SELECT DISTINCT Citta

FROM Abitazione JOIN Contratto

ON Abitazione.Codice = Contratto.Abitazione

WHERE Abitazione.Codice = new.Abitazione);

END|

DELIMITER ;

1. **All’inserimento di un direttore in una sede, verificare che tale direttore afferisca alla sede assegnatagli**

DROP TRIGGER IF EXISTS VerificaAfferenza;

DELIMITER |

CREATE TRIGGER VerificaAfferenza

BEFORE UPDATE ON Sede

FOR EACH ROW

BEGIN

DECLARE NSede INT;

DECLARE CSede VARCHAR(20);

SELECT Sede\_Afferenza, Citta\_Afferenza INTO NSede, CSede

FROM Dipendente

WHERE Matricola = new.Direttore;

IF new.Numero != NSede OR new.Citta != CSede THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Il dipendente scelto non afferisce alla sede di cui si vuole renderlo direttore, update rifiutato";

END IF;

END|

DELIMITER ;

1. **Serie di 3 triggers che, All'inserimento di un dipendente in un ruolo, dovrebbero controllare che egli non sia già già assegnato ad altri ruoli**

DROP TRIGGER IF EXISTS VerificaRuoloAgente;

DELIMITER |

CREATE TRIGGER VerificaRuoloAgente

BEFORE INSERT ON Agente

FOR EACH ROW

BEGIN

IF RuoloDipendente(new.Dipendente) IS NOT NULL THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Questo Dipendente ha già un ruolo, inserimento fallito";

END IF;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS VerificaRuoloSegretario;

DELIMITER |

CREATE TRIGGER VerificaRuoloSegretario

BEFORE INSERT ON Segretario

FOR EACH ROW

BEGIN

IF RuoloDipendente(new.Dipendente) IS NOT NULL THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Questo Dipendente ha già un ruolo, inserimento fallito";

END IF;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS VerificaRuoloDirettore;

DELIMITER |

CREATE TRIGGER VerificaRuoloDirettore

BEFORE INSERT ON Direttore

FOR EACH ROW

BEGIN

IF RuoloDipendente(new.Dipendente) IS NOT NULL THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Questo Dipendente ha già un ruolo, inserimento fallito";

END IF;

END|

DELIMITER ;

1. **Serie di 2 triggers che Impedisce di inserire un contratto con cedenti e cessionari uguali (o comunque sottoinsiemi di l'un l'altro).**

DROP TRIGGER IF EXISTS VerificaCedentiContratti;

DELIMITER |

CREATE TRIGGER VerificaCedentiContratti

BEFORE INSERT ON Cedente

FOR EACH ROW

BEGIN

DECLARE Cnt INT;

SET Cnt = (SELECT Count(\*)

FROM Cessionario

WHERE new.Contratto = Cessionario.Contratto AND new.Cliente=Cessionario.Cliente);

IF Cnt > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Il record già contenuto nella Tabella Cessionario. Questo comporterebbe che un cliente venda a sè stesso. Inserimento Rifiutato";

END IF;

END|

DELIMITER ;

DROP TRIGGER IF EXISTS VerificaCessionariContratti;

DELIMITER |

CREATE TRIGGER VerificaCessionariContratti

BEFORE INSERT ON Cessionario

FOR EACH ROW

BEGIN

DECLARE Cnt INT;

SET Cnt = (SELECT Count(\*)

FROM Cedente

WHERE new.Contratto = Cedente.Contratto AND new.Cliente=Cedente.Cliente);

IF Cnt > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Il record già contenuto nella Tabella Cedente. Questo comporterebbe che un cliente venda a sè stesso. Inserimento Rifiutato";

END IF;

END|

DELIMITER ;

1. **Impedisce l’inserimento di due appuntamenti con la stessa data ed ora che coinvolgano lo stesso agente.**

DROP TRIGGER IF EXISTS VerificaAppuntamenti;

DELIMITER |

CREATE TRIGGER VerificaAppuntamenti

BEFORE INSERT ON Appuntamento

FOR EACH ROW

BEGIN

DECLARE App INT;

SELECT Count(\*) INTO App

FROM Appuntamento a

WHERE a.Data=new.Data AND a.Ora=new.Ora AND a.Agente=new.Agente;

IF App > 0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT="Questo agente ha già un appuntamento per tali data ed ora! Inserimento Rifiutato";

END IF;

END|

DELIMITER ;

**Query e Procedure**

Alcune query si riferiscono a dati di prova completamente diversi da quelli usati nella versione finale del database.

1. **Ritornare Nome, Cognome, I codici di contratto sottoscritti da tutti i clienti, con il ruolo a cui hanno partecipato a tali contratti, ordinati per codice di contratto**

SELECT Cliente.Nome, Cliente.Cognome, Contratto.Codice AS CodiceContratto, RuoloCliente(Cliente.Codice\_Fiscale, Contratto.Codice) AS Ruolo

FROM Cliente, Contratto

WHERE RuoloCliente(Cliente.Codice\_Fiscale, Contratto.Codice) IS NOT NULL

ORDER BY Contratto.Codice;

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Cognome** | **CodiceContratto** | **Ruolo** |
| Roberto | Benigni | 000001 | Cessionario |
| Alberto | Sordi | 000001 | Cedente |
| Fabio | Fazio | 000002 | Cessionario |
| Filippo | Neviani | 000002 | Cedente |
| Nicoletta | Braschi | 000003 | Cedente |
| Alberto | Sordi | 000003 | Cessionario |

1. **Nome, Cognome, Matricola e Numero di Contratti completati dal dipendente del mese (L’agente che ha completato più contratti nel mese)**

SELECT d.Nome, d.Cognome, d.Matricola, count(c.Codice) as Contratti\_Completati

FROM (Dipendente d JOIN Agente a ON a.Dipendente=d.Matricola)

JOIN Contratto c ON c.Agente\_Proponente = a.Dipendente

WHERE Data\_Ratifica IS NOT NULL

ORDER BY Contratti\_Completati DESC

LIMIT 1;

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Cognome** | **Matricola** | **Contratti\_Completati** |
| Franco | De Roberti | 159754 | 1 |

1. **Elenco di Città, con relative provincie e regioni, con numero di contratti proposti e completati in tali città, ordinate per il numero di contratti completati in ordine decrescente**

SELECT cit.Nome, cit.Provincia, cit.Regione, cit.Contratti\_Proposti,

Count(cont.data\_ratifica) AS Contratti\_Completati

FROM (Contratto cont JOIN Abitazione a ON cont.Abitazione=a.Codice)

JOIN Citta cit ON a.Citta=cit.Codice

GROUP BY cit.Codice

ORDER BY Contratti\_Completati DESC;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome** | **Provincia** | **Regione** | **Contratti\_Proposti** | **Contratti\_Completati** |
| Filadelfia | Vibo Valentia | Calabria | 2 | 1 |
| Casatenovo | Lecco | Lombardia | 2 | 0 |
| Grantorto | Padova | Veneto | 1 | 0 |

1. **Trovare Numero, Città e Numero di Contratti Proposti dalle 3 sedi più oberate (con numero maggiore di contratti)**

SELECT s.Numero AS Numero\_Sede, s.Citta AS Città\_Sede,

Count(c.Codice) as Contratti\_Proposti

FROM (

(

(Contratto c JOIN Agente a ON a.Dipendente=c.Agente\_Proponente)

JOIN Dipendente d ON a.Dipendente=d.Matricola)

JOIN Sede s ON d.Sede\_Afferenza=s.Numero AND d.Citta\_Afferenza=s.Citta)

GROUP BY s.Numero, s.Citta

ORDER BY Contratti\_Proposti DESC

LIMIT 3;

|  |  |  |
| --- | --- | --- |
| **Numero\_Sede** | **Citta\_Sede** | **Contratti\_Proposti** |
| 2 | 5 | 4 |
| 1 | 13 | 4 |
| 1 | 8 | 4 |

1. **Procedura che, dati Numero e Città in cui una sede è collocata, mostri Matricola, Nome e Cognome degli agenti che vi lavorano, insieme a Codice, Data ed Ora dei relativi appuntamenti dei giorni avvenire, ordinati per data ed ora**

DELIMITER |

CREATE PROCEDURE ProssimiAppuntamenti(IN nSede int(11), IN cSede varchar(20))

BEGIN

SELECT d.Matricola, d.Nome, d.Cognome, a.Codice as Codice\_Appuntamento, a.Data, a.Ora

FROM (Appuntamento a JOIN Agente ag ON ag.Dipendente=a.Agente)

JOIN Dipendente d ON d.Matricola=ag.Dipendente

WHERE a.Data >= CURDATE()

AND d.Sede\_Afferenza=nSede  
AND d.Citta\_Afferenza=cSede

ORDER BY a.Data, a.Ora ASC;

END;|

DELIMITER ;

CALL ProssimiAppuntamenti(1, ‘4’);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Matricola** | **Nome** | **Cognome** | **Codice\_Appuntamento** | **Data** | **Ora** |
| 1098004 | Caterina | Bilotta | 15 | 2017-02-14 | 11:00:00 |
| 10728015 | Attilio | Dalla Via | 13 | 2017-02-20 | 18:35:00 |
| 1031919 | Elvira | Bonturi | 12 | 2017-03-18 | 17:00 |

1. **Procedura che, dati Numero e Città in cui una sede è collocata, ritorni Matricola, Nome, Cognome e Ruolo di tutti i dipendenti di tale sede, facendo uso della funzione RuoloDipendente.**

DELIMITER |

CREATE PROCEDURE ElencoDipendenti(IN NumeroSede int(11), IN CittaSede

varchar(20))

BEGIN

SELECT d.Matricola, d.Nome, d.Cognome, RuoloDipendente(d.Matricola) as Ruolo

FROM Dipendente d

WHERE d.Sede\_Afferenza = NumeroSede

AND d.Citta\_Afferenza = CittaSede;

END;|

DELIMITER ;

CALL ElencoDipendenti(1,’1’);

|  |  |  |  |
| --- | --- | --- | --- |
| **Matricola** | **Nome** | **Cognome** | **Ruolo** |
| 1053484 | Loredana | Fabbris | Segretario |
| 1069841 | Battista | Marsetti | Agente |
| 1070162 | Jacopo | Stocco | Agente |
| 1070654 | Lucia | Cappellari | Direttore |

**Interfaccia Web:**

L’interfaccia web è stata codificata in PHP, HTML5 e CSS3, con un semplice layout composto da header superiore, una sidebar con menu a scomparsa ed una parte dedicata al contenuto.

Questa interfaccia (assieme al database) è stata testata e funzionante facendo uso dei seguenti pacchetti software (oltre alle macchine del laboratorio):

* PHP 7.1.1: Development server locale (tramite comando: php -S localhost:8000 -t .)
* Client Compatibile con MySQL 15.1: MariaDB 10.1.21
* Firefox 51.0 su Gentoo Linux ~AMD64
* QuteBrowser 0.9.1 git con backends Webkit e QTWebEngine su Gentoo Linux ~AMD64
* Chrome 55 su Android 5.0
* Chrome 50.0.2661.89 Su Android 5.1

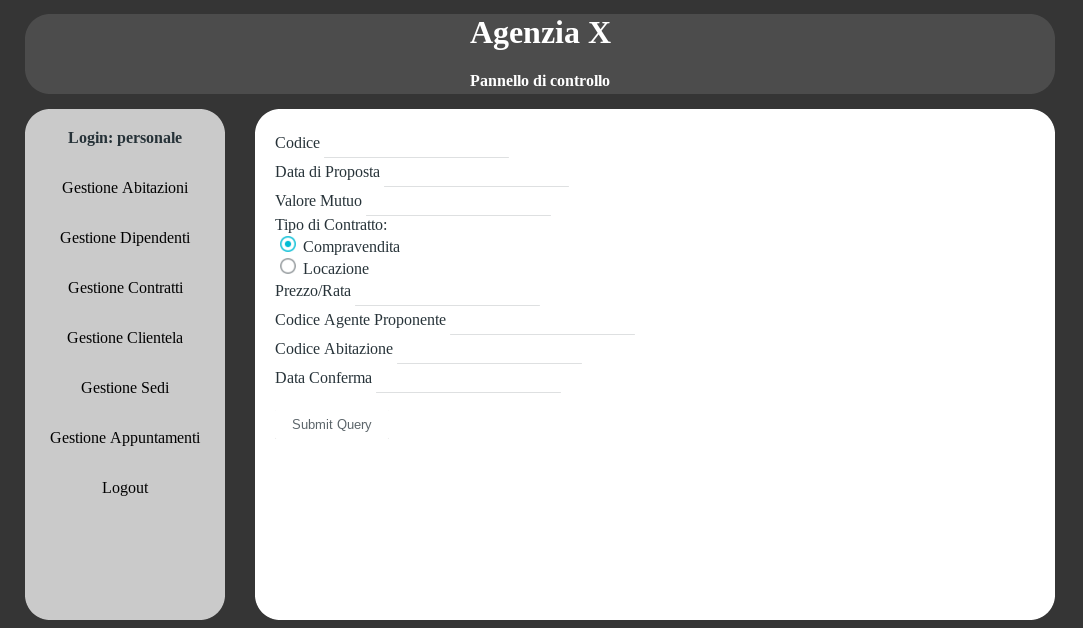
All’apertura della pagina, ci si troverà davanti ad una schermata di login:



È possibile accedere al pannello di gestione con le seguenti credenziali, che per praticità di codifica sono hard-coded all’interno della pagina PHP che andrà a fare la verifica di login:

* **Nome Utente:** personale
* **Password:** password

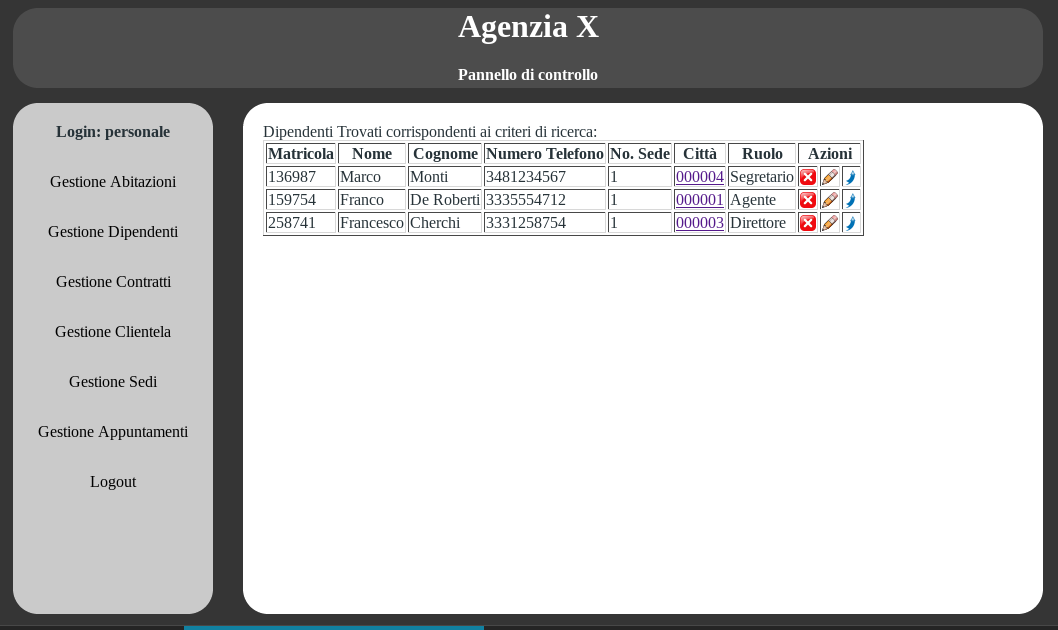
Dopo la login ci si ritroverà subito alla schermata di ricerca contratti:



Nella sidebar si può vedere con che nome utente si è acceduti oltre a varie sezioni tematiche riguardanti la gestione del database, rappresentate da menu pop-up:

* Gestione Abitazioni
  + Registrazione Abitazione
  + Ricerca Abitazioni
* Gestione Dipendenti
  + Aggiunta Dipendenti
  + Assegnazione Dipendente
  + Ricerca Dipendente
* Gestione Contratti
  + Ricerca Contratti
  + Aggiunta contratti
  + Registrazione delle parti contrattuali
* Gestione Clientela
  + Aggiunta Clienti
  + Ricerca Clienti
* Gestione Sedi
  + Aggiunta Sedi
  + Ricerca Sedi
  + Aggiunta Città
  + Ricerca Città
* Gestione Appuntamenti
  + Ricerca Appuntamenti
  + Aggiunta Appuntamenti
  + Aggiunta di clienti partecipanti

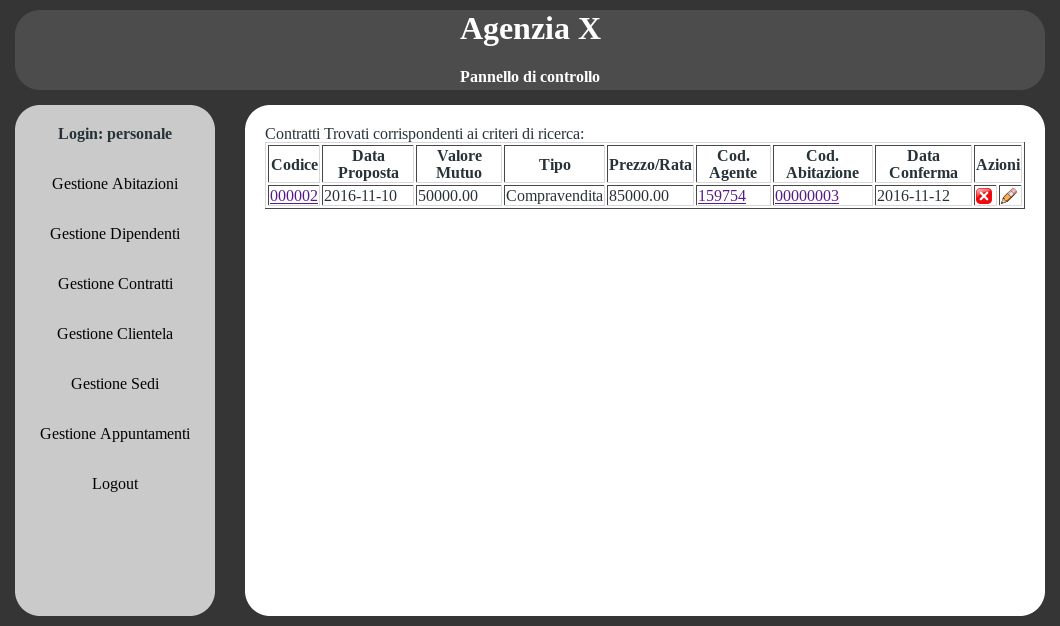
Le operazioni di modifica, aggiornamento ed eliminazione di records vengono fatte dalle schermate dei risultati delle varie ricerche:



Legenda delle icone “Azione”:

* delete.png Cancella la riga dal database, se questo non va ad infrangere vincoli di integrità interni al database, altrimenti porta ad un errore.
* edit.png Permette di modificare la riga, a meno che questo non infranga vincoli di integrità del Database.
* Solleva.png Solleva il dipendente dall’incarico corrente, nel caso un direttore venisse sollevato dall’incarico, il Database farà in modo di settare a NULL il campo “direttore” della sede di afferenza.

Link Rapidi inter-tabellari:



All’interno delle tabelle dei risultati possono comparire dei link che possono rapidamente dare informazioni di interesse coerenti con la natura della ricerca che si è effettuato, nell’esempio in immagine:

* Cliccando sul codice Contratto 000002 si avrà l’elenco delle parti (Cedenti e Cessionari) che sono state coinvolte nel contratto, dando la possibilità di rimuovere quelle aggiunte per errore
* Cliccando sul Codice Agente 159754 si potrà iniziare una ricerca che visualizza tutti i dati dell’agente coinvolto
* Facendo click sul Codice dell’abitazione si avvia una ricerca che mostra l’ubicazione dell’abitazione in oggetto.

Questi link rapidi sono presenti all’interno di tutta l’interfaccia web, come nella ricerca appuntamenti, o nella ricerca di sedi, o nella visualizzazione delle parti coinvolte nei contratti.

Questi link permettono di velocizzare operazioni come sollevare il direttore di una sede conosciuta: basta cerca la sede, cliccare sul codice del direttore e poi cliccare sull’icona “Solleva dall’incarico”.

L’uso del bottone “logout” porterà alla conclusione della sessione, alla sua distruzione, insieme a tutti i cookies settati nel frattempo.

Inserire il codice delle pagine web in questa relazione renderebbe il tutto troppo lungo da leggere (68 files tra interfacce e pagine di esecuzione), quindi abbiamo preferito omettere tutto ciò.