

Report del progetto di Basi di Dati

Vincenzo Arceri - VR360465
Matteo Calabria - VR363871

24 Maggio 2014

1 Requisiti della base di dati

Si vuole progettare un sistema informativo per gestire le informazioni relative alla gestione di un'agenzia immobiliare.

L'agenzia mette in vendita, per conto dei suoi clienti, immobili che possono essere di tre tipi: appartamenti in palazzina, villette a schiera, ville singole. Per ogni immobile il sistema memorizza: un codice univoco, l'indirizzo, la città, la superficie totale, il numero di vani e una descrizione. Per gli appartamenti in palazzina si registra anche il piano a cui si trova l'appartamento, mentre per le ville singole si registra anche la superficie del giardino. Ogni tentativo di vendita di un immobile viene registrato indicando il cliente per conto del quale si gestisce la tentata vendita, l'immobile oggetto della vendita, un prezzo minimo, e il periodo di tempo per cui il prezzo è valido. Un immobile rimasto invenduto la prima volta può essere messo in vendita più volte in periodi diversi. Per ogni cliente il sistema registra: il suo codice fiscale, il nome, il cognome, la città di residenza, il numero di telefono ed eventualmente il numero di cellulare. Si registrano inoltre login e password per l'accesso ai servizi disponibili via web.

Per ogni vendita il sistema registra i contatti di potenziali acquirenti interessati ad un dato immobile messo in vendita registrando: il giorno in cui si è verificato il contatto, il cognome, il nome e il numero di telefono dell'interessato insieme al prezzo offerto.

Per ogni immobile venduto il sistema memorizza i dati dell'acquirente: cognome, nome, codice fiscale, numero di telefono, il prezzo finale e la data in cui la vendita è stata decisa. Successivamente registra la data della compravendita vera e propria e il nome del notaio che redige l'atto formale di compravendita.

Infine il sistema registra gli agenti immobiliari che lavorano per l'agenzia memorizzando: un codice univoco, il nome, il cognome, la data di assunzione, l'indirizzo e la città di residenza. Per ogni compravendita il sistema registra anche l'agente immobiliare che l'ha seguita.

2 Progettazione della base di dati

2.1 Progettazione concettuale della base di dati

Per la progettazione della base di dati abbiamo utilizzato il modello Entità-Relazione, come si evince dalla Figura 1. Nella nostra progettazione abbiamo assunto che un cliente possa mettere in vendita lo stesso immobile più di una

volta a patto che tali tentativi di vendita siano validi in periodi disgiunti. Un acquirente può essere diviso in acquirente potenziale ed acquirente effettivo e per questo motivo abbiamo optato per la generalizzazione dei concetti: l'identificatore viene posto sui figli della generalizzazione che nel caso degli acquirenti effettivi è il codice fiscale, mentre per quelli potenziali è la terna nome, cognome e telefono (a livello fisico inseriremo un id univoco per comodità). Per quanto riguarda la vendita di un immobile, abbiamo deciso di modellarla tramite una relazione ternaria: un agente può concludere zero o più vendite, un acquirente effettivo può aggiudicarsi uno o più tentativi di vendita ed un tentativo di vendita si riferisce solamente ad un agente e un acquirente.

2.2 Progettazione logica e fisica della base di dati

Il modello relazionale costruito a partire dal modello E/R rappresentato nella sezione precedente è il seguente:

Immobile(*CodiceImmobile*, *IndirizzoImmobile*, *CittaImmobile*, *SuperficieImmobile*, *DescrizioneImmobile*, *NumeroVani*, *TipoImmobile*:{*Palazzina*, *Schiera*, *Singola*}, *NumeroPiano**, *SuperficieGiardino**)
Cliente(*CodiceFiscaleCliente*, *NomeCliente*, *CognomeCliente*, *CittaResidenzaCliente*, *TelefonoCliente*, *CellulareCliente**, *UsernameCliente*, *PasswordCliente*)
TentativoVendita(*DataInizioTentativo*, *ImmobileInVendita*, *CodiceFiscaleVenditore*, *PrezzoMinimo*, *DataFineTentativo*, *CodiceAgenteVendita**, *CodiceFiscaleAcquirente**, *Notaio**, *PrezzoFinale**, *DataConcordata**, *DataEffettiva**)
AcquirentePotenziale(*NomeAcquirentePot*, *CognomeAcquirentePot*, *TelefonoAcquirentePot*)
AcquirenteEffettivo(*CodiceFiscaleAcquirenteEff*, *NomeAcquirenteEff*, *CognomeAcquirenteEff*, *TelefonoAcquirenteEff*)
Contatto(*NomeAcquirenteP*, *CognomeAcquirenteP*, *TelefonoAcquirenteP*, *Cliente*, *DataInizio*, *Immobile*, *GiornoContatto*, *Offerta*)
Agente(*CodiceAgente*, *NomeAgente*, *CognomeAgente*, *DataAssunzione*, *CittaResidenzaAgente*, *IndirizzoAgente*)

con i seguenti vincoli d'integrità (per chiarezza abbiamo utilizzato i nomi delle relazioni in maiuscolo):

- TENTATIVOVENDITA.Immobile → IMMOBILE
- TENTATIVOVENDITA.CodiceFiscaleVenditore → CLIENTE
- CONTATTO.NomeAcquirenteP, CONTATTO.CognomeAcquirenteP, CONTATTO.TelefonoAcquirenteP → ACQUIRENTEPOTENZIALE
- CONTATTO.Cliente, CONTATTO.DataInizio, CONTATTO.Immobile → TENTATIVOVENDITA
- TENTATIVOVENDITA.CodiceAgenteVendita → AGENTE
- TENTATIVOVENDITA.CodiceFiscaleAcquirente → ACQUIRENTEEFFETTIVO

Per quanto riguarda la generalizzazione dell'immobile abbiamo deciso, nella traduzione dal modello E/R a quello relazionale, di accorporare i figli sul padre, mentre per la generalizzazione dell'acquirente abbiamo deciso di accorporare sui figli, poiché hanno relazioni ed attributi differenti e poiché la generalizzazione è totale sovrapposta.

A livello fisico abbiamo deciso di introdurre degli id per maggiore comodità, come si evince dallo script SQL di creazione della base di dati.

3 Requisiti della progettazione dell'applicazione web

Le informazioni contenute nella base di dati devono essere presentate in un sito web. Tale sito deve essere strutturato in schemi di pagina come descritto; è possibile estendere la struttura aggiungendo ulteriori schemi di pagina che mostrano ulteriori dettagli:

- **Home page**, dove si riporta:
 - Presentazione dell'azienda;
 - Elenco dei tipi di immobile attualmente in vendita indicando per ogni tipo il numero (quantità) di tentativi di vendita in corso (il tipo è un link verso lo schema di pagina **TentativiDiVenditaPage**);
 - Sede dell'azienda e contatti.
- **TentativiDiVenditaPage**, dove si presentano le seguenti informazioni:
 - Il tipo di immobili a cui si riferisce la pagina;
 - L'elenco dei tentativi di vendita di immobili di questo tipo; riportare nell'elenco: il codice univoco, l'indirizzo, la città e la superficie totale dell'immobile (il codice è un link verso lo schema di pagina **ImmobilePage**).
- **ImmobilePage**, dove si presentano le seguenti informazioni:
 - Il codice univoco, l'indirizzo, la città, la superficie totale, il numero di vani e la descrizione dell'immobile;
 - Una form che consente ad un potenziale acquirente di inserire un'offerta per l'acquisto dell'immobile indicando: il cognome, il nome e il numero di telefono dell'interessato e il prezzo offerto. Se l'offerta viene registrata correttamente il sistema ritorna alla pagina **ImmobilePage**;
 - Una form che consente all'utente per conto del quale si gestisce la tentata vendita, di collegarsi alla pagina che mostra le offerte pervenute (**OffertePage**).
- **OffertePage**, dove si presentano le seguenti informazioni:
 - L'elenco delle offerte pervenute per l'immobile riportando: il giorno in cui l'offerta è stata registrata, il cognome, il nome e il numero di telefono dell'interessato e il prezzo offerto.

3.1 Progettazione logica dell'applicazione web

Di seguito sono riportati gli schemi di pagina dell'applicazione web.

```
page schema HomePage unique (
    TotaleSingole: integer;
    TotaleSchiera: integer;
    TotalePalazzina: integer;
)

page schema TentativiDiVenditaPage (
    TentativiVendita( list_of
        ( link(Codice: string, *ImmaginePage),
          Indirizzo: string;
          Città: string;
          Superficie: int
        )
    )
)

page schema ImmaginePage (
    CodiceImmagine: string,
    IndirizzoImmagine: string,
    Città: string,
    Superficie: integer,
    NumeroVani: integer,
    Descrizione: integer
    NuovaOfferta: form (
        Nome: text,
        Cognome: text,
        Telefono: text,
        Offerta: text,
        FaiOfferta: submit(*HomePage)
    )
    Login: form (
        Username: text,
        Password: password
        Login: submit(*Offerta)
    )
)

page schema OffertePage (
    Offerta(list_of(
        GiornoOfferta: date,
        Cognome: string,
        Nome: string,
        Telefono: string,
        Offerta: integer
    )
)
```

```

DB to page schema HomePage parameter(tipoImmobile) (
    TotaleSingola: SELECT Count(*)
        FROM Immobile, TentativoVendita
        WHERE Immobile.CodiceImmobile =
TentativoVendita.ImmobileInVendita
        AND Immobile.Tipo = ?tipoImmobile?
        AND TentativoVendita.DataFineTentativo >= '26/05/2014'
        AND TentativoVendita.DataInizioTentativo < '26/05/2014'
        AND TentativoVendita.DataEffettiva is null;
    TotaleSchiera: SELECT Count(*)
        FROM Immobile, TentativoVendita
        WHERE Immobile.CodiceImmobile =
TentativoVendita.ImmobileInVendita
        AND Immobile.Tipo = ?tipoImmobile?
        AND TentativoVendita.DataFineTentativo >= '26/05/2014'
        AND TentativoVendita.DataInizioTentativo < '26/05/2014'
        AND TentativoVendita.DataEffettiva is null;
    TotalePalazzina: SELECT Count(*)
        FROM Immobile, TentativoVendita
        WHERE Immobile.CodiceImmobile =
TentativoVendita.ImmobileInVendita
        AND Immobile.Tipo = ?tipoImmobile?
        AND TentativoVendita.DataFineTentativo <= '26/05/2014'
        AND TentativoVendita.DataInizioTentativo > '26/05/2014'
        AND TentativoVendita.DataEffettiva is null;
)

```

```

DB to page schema TentativiDiVenditaPage parameter(tv) (
    TentativiVendita:
    SELECT TentativoVendita.IdTentativoVendita,
        Immobile.CodiceImmobile,
        Immobile.IndirizzoImmobile,
        Immobile.CittaImmobile,
        Immobile.SuperficieImmobile
    FROM Immobile, TentativoVendita
    WHERE Immobile.CodiceImmobile =
        TentativoVendita.ImmobileInVendita
        AND Immobile.Tipo = ?tv?
        AND TentativoVendita.DataFineTentativo <= '26/05/2014'
        AND TentativoVendita.DataInizioTentativo > '26/05/2014'
        AND TentativoVendita.DataEffettiva is null ;
)

```

```

DB to page schema ImmobilePage parameter(tv) (
    CodiceImmobile, IndirizzoImmobile, Città, Superficie, NumeroVani,
    Descrizione:
    SELECT Immobile.CodiceImmobile,
        Immobile.IndirizzoImmobile,
        Immobile.CittaImmobile,
        Immobile.SuperficieImmobile,

```

```

        Immobile.NumeroVani,
        Immobile.DescrizioneImmobile
    FROM Immobile, TentativoVendita
    WHERE Immobile.CodiceImmobile =
        TentativoVendita.ImmobileInVendita
    AND TentativoVendita.IdTentativoVendita = ?tv? ;
)

```

DB to page schema **OffertePage** parameter(tv) (

Offerte:

```

    SELECT Contatto.GiornoContatto, Contatto.Offerta,
        AcquirentePotenziale.NomeAcquirentePot,
        AcquirentePotenziale.CognomeAcquirentePot,
        AcquirentePotenziale.TelefonoAcquirentePot
    FROM Contatto, AcquirentePotenziale
    WHERE Contatto.IdPotenzialeAcquirente =
        AcquirentePotenziale.IdAcquirentePot
    AND Contatto.IdTentativoDiVendita = ?tv? ;
)

```

DB from page schema **ImmobilePage** (

NewOfferta:

```

    idAcquirentePotenziale = " "
    idAcquirentePotenziale = (
        SELECT AcquirentePotenziale.IdAcquirentePot
        FROM AcquirentePotenziale
        WHERE AcquirentePotenziale.NomeAcquirentePot = ?Nome?
        AND AcquirentePotenziale.CognomeAcquirentePot = ?Cognome?
        AND AcquirentePotenziale.TelefonoAcquirentePot = ?Telefono?
        if (idAcquirentePotenziale == null) {
            string idAcquirentePotenziale = Random.nextInt().toString();
            INSERT INTO AcquirentePotenziale
                (IdAcquirentePot, NomeAcquirentePot,
                CognomeAcquirentePot, TelefonoAcquirentePot)
            VALUES(idAcquirentePotenziale, ?Nome?, ?Cognome?,
                ?Telefono?)
        }
        INSERT INTO Contatto(IdPotenzialeAcquirente,
            IdTentativoDiVendita, GiornoContatto, Offerta)
        VALUES(idAcquirentePotenziale, ?tv?, '2014-06-04', ?Offerta?)
    )

```

Login:

```

    if (SELECT TentativoVendita.idTentativoVendita
        FROM Cliente, TentativoVendita
        WHERE Cliente.CodiceFiscaleCliente =
            TentativoVendita.CodiceFiscaleVenditore
        AND Cliente.UsernameCliente = ?Username?
        AND Cliente.PasswordCliente = ?Password?
        AND TentativoVendita.idTentativoVendita = ?tv?)
        *OffertaPage
    else

```

) *ErrorPage

3.2 Struttura dell'applicazione web

Il modello di progettazione utilizzato nel nostro progetto è il modello MVC-2 servlet centric:

- **Model:** tale modulo si occupa dell'interazione con il DBMS; viene implementato da una classe Java che nel nostro caso si chiama DBMS.java, contenente tutte le query riportate del page schema;
- **View:** tale modulo si occupa della presentazione dei dati; tale modulo è realizzato tramite le diverse pagine JSP, di cui parleremo ampliamento di seguito.
- **Control:** tale modulo si occupa della interazione fra i precedenti due moduli; viene realizzato dalla classe Main.java che in base a il valore e la presenza di diversi parametri invoca differenti pagine JSP, utilizzando diversi Java Data Bean per comunicare.

La classe Main gestisce richieste HTTP di tipo get e post. Si accettano i seguenti parametri:

- ps: è il parametro che indica la pagina JSP che deve essere lanciata; può assumere i seguenti valori: tvp e imm per la richieste di tipo GET, e inserimento e offerte per le richieste di tipo POST. La classe Main in base ai parametri effettuerà le dovute chiamate ai metodi della classe DBMS per ottenere i dati da passare alla relativa JSP.
- idTent: indica il tentativo di vendita a cui fa riferimento la pagina; ha utilità nel caso di ImmobilePage.jsp e OffertePage.jsp

Nell'applicazione web utilizziamo le seguenti pagine JSP:

- **HomePage.jsp:** è la pagina di home che mostra la quantità di tentativi di vendita attivi divisi per tipologia d'immobile; i tipi sono link alla pagina TentativiDiVenditaPage;
- **TentativiDiVenditaPage.jsp:** è la pagina che mostra i tentativi di vendita per il tipo d'immobile selezionato in HomePage; il codice del relativo immobile è un link ad ImmobilePage;
- **ImmobilePage.jsp:** è la pagina che mostra le informazioni di un dato immobile relativo ad un tentativo di vendita; inoltre sono presenti due form, la prima per effettuare una nuova offerta, come descritto nel page schema, mentre la seconda viene utilizzata per effettuare il login da parte del cliente per cui si gestisce la vendita; se il login va a buon fine viene invocata la pagina OffertePage;
- **OffertePage.jsp:** mostra le offerte effettuate relative all'immobile selezionato nella pagina precedente.
- **ErrorPage.jsp:** è la pagina d'errore, lanciata in caso di URL sbagliato o login errato.