



Progettazione e sviluppo di una Base di dati relazionale per la gestione di una rubrica telefonica avanzata

Oleksandr Sosovskyy Francesco Magri
819383418043 89432498

Inserire la data

Indice

1	Requisiti identificati	3
1.1	Requisiti sui dati	3
1.2	requisiti sulle operazioni	3
2	Progettazione concettuale	4
2.1	Class Diagram	4
2.2	Analisi della ristrutturazione del Class Diagram	4
2.2.1	Analisi delle ridondanze	4
2.2.2	Analisi degli identificativi	4
2.2.3	Rimozione degli attributi multipli	4
2.2.4	Rimozione degli attributi composti	5
2.2.5	Rimozione delle gerarchie	5
2.2.6	Partizione/Accorpamento delle associazioni	5
2.3	Class Diagram ristrutturato	5
2.4	Dizionario delle classi	5
2.5	Dizionario delle associazioni	6
3	Progettazione logica	8
3.1	Schema logico	8

Capitolo 1

Requisiti identificati

Si progetterà una base di dati relazionale per la gestione di una rubrica telefonica avanzata. Il sistema memorizza e gestisce le informazioni relative ai contatti di un utente. Viene ammessa la possibilità per più persone di usare separatamente la propria rubrica, i cui contatti mantengono la completa indipendenza informativa rispetto a eventuali omonimi presenti in altre rubriche. Un indirizzo elettronico associa un contatto al suo account presso i sistemi di messaging. La Base di dati ignora il modo in cui le suddette informazioni esterne vengano reperite: si ammette che vengano recuperate dal dispositivo in uso, inserite nella Base di dati e, alla dichiarazione della corretta email, associate al contatto in questione.

1.1 Requisiti sui dati

La Base di dati gestisce le seguenti classi di dati:

- **Rubrica** indica gli utenti che hanno aggiunto una personale *rubrica*. Il sistema consente di gestire le rubriche di più utenti;
- **Elemento** indica l'atomo concettuale di ogni *Rubrica*: senza almeno un *elemento* è vuota. L'*elemento* è una generalizzazione delle entità presenti in una *Rubrica*, vale a dire i *contatti* e i *gruppi*;
- **Contatto** indica gli *elementi* di una *Rubrica* che possiedono qualche numero di telefono. I numeri associati a un contatto possono essere più di uno e a ciascuno sono associabili possibili descrizioni (come: mobile, fisso, ufficio, ecc.). Proprio come per i numeri, si rappresentano gli indirizzi fisici e le email con record di attributi semplici. Altri attributi di un *contatto* sono il nome, il secondo nome e il cognome;
- **Gruppo** indica gli *elementi* che sono collezione (anche vuote) di *contatti*;
- **Account** indica la collezione di informazioni associate a una email registrata su qualche sistema di messaging. Tali informazioni sono il nome del fornitore, l'indirizzo email, il nickname e la frase di benvenuto.

1.2 requisiti sulle operazioni

La Base di dati consente le seguenti operazioni sui dati:

- Aggiunta e cancellazione di contatti e gruppi;
- Manipolazione delle informazioni di un *Contatto* o di un *Gruppo*;
- Ricerca di *contatti* per nome, email, account di messaging e per numero di telefono.

Capitolo 2

Progettazione concettuale

2.1 Class Diagram

Foto

2.2 Analisi della ristrutturazione del Class Diagram

2.2.1 Analisi delle ridondanze

Il Class Diagram non presenta ridondanze da eliminare. Tuttavia, al termine della ristrutturazione è possibile che alcune istanze di dati possano risultare ridondanti. Il sistema ammette che *contatti* diversi possano condividere, concettualmente, numeri telefonici, indirizzi o, se in rubriche diverse, email uguali. La scelta progettuale di mantenere possibili duplicati per ogni *contatto* è però inevitabile. Il funzionamento del modello relazionale impone che informazioni come il numero di telefono siano riscritte per ogni *contatto* che ha lo stesso numero nella Base di dati. Si tratta di chiavi esterne che necessariamente metteranno in relazione un contatto con i suoi numeri di telefono.

2.2.2 Analisi degli identificativi

Potenziali identificativi potrebbero essere gli attributi già presenti nelle classi. La scelta di un *nome* come identificativo, tuttavia, è problematica dal punto di vista implementativo della Base di dati. Usare come identificativo di un contatto il suo nome, anche col secondo nome e cognome, vincolerebbe ogni utente che fa uso di una propria rubrica a evitare nomi che sono stati già scelti anche da contatti che fanno uso della propria rubrica. Problemi logici di questo genere sono risolvibili aggiungendo chiavi tecniche, o *codici*, che fungano da identificativi. Si ragiona diversamente per la classe *Rubrica*: è bene che non ci siano ambiguità tra gli utenti che hanno una propria *rubrica*, pertanto sarebbe superfluo aggiungere codici.

2.2.3 Rimozione degli attributi multipli

Nella classe *Contatto* sono presenti i seguenti attributi multipli: Numeri, Email e Indirizzi. Tali attributi portano informazioni non eliminabili dalla Base di dati, ma possono essere anche aggiunti un numero arbitrario di volte. Queste ragioni sono sufficienti per suggerire la creazione di classi per ciascuno degli attributi multipli. Trattandosi anche di attributi multipli, la discussione è ripresa nella sezione successiva.

2.2.4 Rimozione degli attributi composti

Gli attributi composti coincidono con gli attributi multipli. È naturale l'introduzione di classi per ciascuno degli attributi, con ciascuna classe avente come attributi i campi degli attributi composti.

2.2.5 Rimozione delle gerarchie

La classe *Elemento* rappresenta una generalizzazione delle classi *Contatto* e *Gruppo*. Si decide di rimuoverla poiché ha un mero valore concettuale.

2.2.6 Partizione/Accorpamento delle associazioni

Le associazioni individuate sono essenziali nell'indicare relazioni logiche tra le Entità. Si sceglie di non introdurre partizionamenti oppure accorpamenti.

2.3 Class Diagram ristrutturato

Foto

2.4 Dizionario delle classi

Classe	Descrizione	Attributi
Rubrica	Descrittore delle rubriche aggiunte	Utente_ID (varchar) : Chiave tecnica. Identifica l'utente proprietario di una rubrica.
Gruppo	Descrittore di una collezione di contatti in rubrica	Gruppo_ID (integer) : Chiave tecnica. Identifica univocamente ciascun gruppo; Nome(vvarchar) : Nome del gruppo.
Contatto	Descrittore di un'entità contatto	Contatto_ID (integer) : Chiave tecnica. Identifica univocamente ogni contatto; Nome (varchar) : indica il primo nome del contatto; SecondoNome (varchar) :indica il secondo nome del contatto; Cognome (varchar) :indica il cognome del contatto; Foto (blobs) : immagine associata al contatto.
Email	Descrittore di un'entità indirizzo di posta elettronica di un contatto	Email_ID (integer) : Chiave tecnica. Identifica univocamente un'email di un contatto; IndirizzoEmail (varchar) : stringa dell'email; Descrizione (varchar) : informazione aggiuntiva di un email (privata, ufficio, ecc).

Indirizzo	Descrittore di un'entità indirizzo fisico di un contatto	Indirizzo_ID (integer): Chiave tecnica. Identifica univocamente un indirizzo fisico di un contatto; Via (varchar): indica la via dell'indirizzo; Città (varchar): indica la città dell'indirizzo; Nazione(string): indica la nazione dell'indirizzo; CAP (string): indica il codice postale dell'indirizzo; Tipo(TipoA): indica se l'indirizzo è principale o secondario
Telefono	Descrittore di un'entità numero telefonico di un contatto	Telefono_ID (integer): Chiave tecnica. Identifica univocamente un numero di telefono di un contatto; Numero (varchar): stringa del numero; Descrizione (varchar): informazione aggiuntiva di un numero di telefono (mobile, fisso, ufficio ecc).
Account	Descrittore delle informazioni recuperate da un account	Account_ID (varchar): Chiave tecnica. Identifica l'utente proprietario di una rubrica; Fornitore (varchar): indica il fornitore da cui è recuperato l'account (Whatsapp, Telegram, Teams, ecc.); IndirizzoEmail (varchar): indirizzo email associato a un account; FraseStato (varchar): frase di benvenuto associata a un account; Nickname(vvarchar): indica il nickname associato a un account.

2.5 Dizionario delle associazioni

Associazione	Descrizione	Classi coinvolte
AppartenenzaGruppo	Esprime l'appartenenza di un gruppo a una rubrica	Rubrica [1]: indica la rubrica che contiene un gruppo; Gruppo[0..*]: indica il gruppo appartenente a una rubrica.
AppartenenzaContatto	Esprime l'appartenenza di un contatto a una rubrica	Rubrica [1]: indica la rubrica che contiene un contatto; Contatto[0..*]: indica il contatto appartenente a una rubrica.

Composizione	Esprime l'appartenenza di un contatto a un gruppo	Gruppo [0...*]: indica il gruppo che contiene un contatto; Contatto[0..*]: indica il contatto appartenente a un gruppo.
Associa	Esprime l'associazione di un account a un contatto	Contatto [0...*]: indica il contatto a cui è associato un account; Account [0..*]: indica l'account che è associato a un contatto.
RecapitoEmail	Esprime l'appartenenza di una email ad un contatto	Email [0..*]: indica un'entità email appartenente ad un contatto; Contatto[1]: indica il contatto a cui è associata l'entità email.
RecapitoIndirizzo	Esprime l'appartenenza di un'entità indirizzo fisico ad un contatto	Indirizzo [1..*]: indica un'entità indirizzo fisico appartenente ad un contatto; Contatto[1]: indica il contatto a cui è associata l'entità indirizzo.
RecapitoTelefono	Esprime l'appartenenza di un'entità numero di telefono ad un contatto	Telefono [1..*]: indica un'entità numero di telefono appartenente ad un contatto; Contatto[1]: indica il contatto a cui è associata l'entità numero di telefono.
AppartenenzaContatto	Esprime l'appartenenza di un contatto a una rubrica	Rubrica [1]: indica la rubrica che contiene un contatto; Contatto[0..*]: indica il contatto appartenente a una rubrica.

Capitolo 3

Progettazione logica

3.1 Schema logico

Rubrica	(<u>Utente_ID</u>)
Contatto	(<u>Contatto_ID</u> , Nome, SecondoNome, Cognome, Foto, <u>Rubrica_FK</u>) Rubrica_FK → Rubrica.Utente_ID
Gruppo	(<u>Gruppo_ID</u> , Nome, <u>Rubrica_FK</u>) Rubrica_FK → Rubrica.Utente_ID
Account	(<u>Account_ID</u> , Fornitore, IndirizzoEmail, FraseStato, Nickname)
Associa	(<u>Contatto_ID</u> , <u>Account_ID</u>) Contatto_ID → Contatto.Contatto_ID Account_ID → Account.Account_ID
Email	(<u>Email_ID</u> , IndirizzoEmail, Descrizione, <u>Contatto_ID</u>) Contatto_ID → Contatto.Contatto_ID
Indirizzo	(<u>Indirizzo_ID</u> , Via, Città, CAP, Nazione, Tipo, <u>Contatto_ID</u>) Contatto_ID → Contatto.Contatto_ID
Telefono	(<u>Telefono_ID</u> , Numero, Descrizione, <u>Contatto_ID</u>) Contatto_ID → Contatto.Contatto_ID