***Titolo***

Progettazione e descrizione di una base di dati relazionale per la gestione e memorizzazione di rubriche telefoniche avanzate.

**Introduzione**

**Analisi del problema**

Si vuole progettare una base di dati che permetta all’utente di memorizzare e gestire una rubrica telefonica avanzata. L’utente potrà accedere alla propria rubrica oppure registrarsi per poterne creare una nuova, per effettuare l’accesso e la registrazione sarà necessario fornire e-mail e password valide.  
La rubrica sarà costituita da contatti, in cui verranno mantenute tutte le informazioni principali, e ad esso saranno associati indirizzi di posta elettronica, indirizzi fisici, numeri di telefono e account di messaging. I contatti potranno essere organizzati in gruppi a cui sarà possibile aggiungere o rimuovere partecipanti.

**Requisiti identificati**

La rubrica deve essere in grado di consentire la memorizzazione delle informazioni riguardanti i contatti, ovvero nome, cognome, una o più e-mail ed eventualmente una foto profilo. Un contatto può essere inserito tra i preferiti e può essere partecipante di uno o più gruppi. I gruppi avranno un nome, una data di creazione e una descrizione.

Ad ogni contatto può essere associato un indirizzo fisico principale e uno o più secondari, che specifichino via, città, zipcode e nazione.

Tutti i contatti devono essere dotati di almeno un numero di telefono mobile ed un numero di telefono fisso, e per ogni telefono mobile può essere indicato un telefono fisso a cui verranno reindirizzate

eventuali chiamate senza risposta e, analogamente per ogni telefono fisso deve essere indicato un telefono

mobile per il reindirizzamento.

Agli stessi indirizzi fisici e numeri di telefono fissi può corrispondere anche più di un contatto.

Inoltre, ad ogni contatto verranno associati gli account a sistemi di messaging. Per ognuno di questi account verrà memorizzato il fornitore, il nickname, la frase di benvenuto e l’indirizzo e-mail collegato che dovrà necessariamente esistere tra gli account di posta già salvati per il contatto.

**Progettazione concettuale**

*\*class diagram non ristrutturato\**

**Analisi delle ridondanze**

**Analisi degli indentificativi**

**Rimozione degli attributi multipli**

Email

**Rimozione degli attributi composti**

**Partizione/Accorpamento delle associazioni**

**Rimozione delle gerarchie**

*\*class diagram ristrutturato\**

**Dizionario delle classi**

**User**: utente proprietario della rubrica

* E-mail (string): e-mail del proprietario della rubrica
* Password (string): password di accesso alla rubrica
* Nickname (string): nome identificativo del proprietario della rubrica

**Contact**: istanza di un contatto salvato nella rubrica

* ID: (string) codice di identificazione del contatto all’interno della rubrica
* Name: (string): nome del contatto
* Surname: (string): cognome del contatto
* Propic: (string): percorso sul disco del file contenente l’immagine profilo del contatto

**Address:** indirizzo fisico associato ad uno o più contatti

* Street: (string) stringa contenente nome della strada e numero civico dell’indirizzo
* ZipCode (string): codice postale (CAP) dell’indirizzo
* City (string): nome della città dell’indirizzo
* Province (string): identificativo della provincia relativa alla città
* Country (string): nazione dell’indirizzo
* Main (boolean): indica se è l’indirizzo principale del contatto. Può esistere solo un indirizzo principale per contatto

**Email**: Indirizzo di posta elettronica associato al contatto

* E-mail (string): stringa contenente l’indirizzo di posta elettronica
* Main (boolean): indica se è l’e-mail principale del contatto. Può esistere solo una e-mail principale per contatto

**Social:** account social o di messaging associato ad un contatto

* supplierName (string): nome del fornitore del social o dell’app di messaging
* nickname (string): nome identificativo dell’account del contatto
* assEmail (string): e-mail associata all’account del contatto. Deve esistere tra le e-mail già salvate per il contatto
* welcome

**Phone Number**: numero di telefono associato ad uno o più contatti

* Type (string): tipo del numero di telefono. Può essere mobile (MOBILE) oppure fisso (LANDLINE)
* Number (string): numero di telefono (?)
* linkedNumber: (boolean) indica se è un numero di reindirizzamento. Per ogni contatto deve esistere un numero di reindirizzamento tra i numeri fissi, e uno tra i numeri mobili

**Group:** gruppo a cui possono appartenere uno o più contatti all’interno di una rubrica

* ID (string): chiave tecnica. Codice identificativo del gruppo creato da un utente
* Groupname (string): nome del gruppo all’interno della rubrica
* CreationDate (date): data della creazione del gruppo

**Partecipants:** descrittore dell’associazione tra i partecipanti ad un gruppo ed il gruppo

* GroupID: (string) ID del gruppo a cui partecipa il contatto
* ContactID (string): ID del contatto partecipante
* joinDate (date): data di inizio partecipazione al gruppo

**Dizionario delle associazioni**

User-Contact  
Esprime l’appartenenza di un contatto ad una rubrica di un utente

User-Group  
Esprime l’appartenenza di un gruppo ad una rubrica di un utente

Group-Partecipant  
Indica a quale a quale gruppo si riferisce la partecipazione di un contatto

Contact-Partecipant  
Indica quale contatto è partecipante ad un gruppo

Contact-Address  
Indica a quali contatti si riferisce un indirizzo fisico

Contact-Email  
Indica a quale contatto si riferisce una e-mail

Social-Email  
Indica a quale e-mail è collegato un account social

Contact-Social  
Indica a quale contatto è collegato un account social

Contact-PhoneNumber  
Indica a quali contatti si riferisce un numero di telefono

**Schema Logico (bozza: da stabilire formato colonne)**

USER (**E-mail**, Password, Nickname, Notes)

CONTACT (**ContactID**, Name, Surname, Propic, Favorite, User)

*User -> User.E-mail*

GROUP (**GroupID**, Name, CreationDate, Description, User)

*User -> User.E-mail*

PARTECIPANT (ContactID, GroupID, JoinDate)

*ContactID -> Contact.ContactID, GroupID -> Group.GroupID*

ADDRESS (**Street, ZipCode**, City, Province, Country, Main)

E-MAIL (**E-mail**, Main, ContactID)

*ContactID -> Contact.ContactID*

SOCIAL (**SupplierName, AssEmail**, Nickname, welcome, ContactID)

AssEmail -> E-mail.E-mail , *ContactID -> Contact.ContactID*

PHONENUMBER (**Number**, Type, LinkedNumber)

ASSIGNEDADDRESS (ContactID, AddressStr, AddressZip)

*ContactID -> Contact.ContactID, (AddressStr, AddressZip) -> (Address.Street, Address.ZipCode)*

ASSIGNEDPHONE (ContactID, Number)

*ContactID -> Contact.ContactID, Number -> PhoneNumber.Number*

*Vincoli:  
\*L’indirizzo e-mail di un social di un contatto deve essere necessariamente tra gli account di posta già salvati per il contatto. (Traccia)  
\*Ogni contatto deve avere almeno un telefono fisso e un telefono mobile. (Traccia)  
\*Si tenga conto che il sistema accetta contatti che dichiarino stessi indirizzi fisici o numeri telefonici ma non accetta contatti che dichiarano stesse e-mail. (Traccia)  
\*Deve esistere un solo indirizzo e-mail main per contatto. (Traccia)  
\*Social: unicità della coppia di attributi “SupplierName, AssEmail”*

*Trigger:*

*\*Alla rimozione di un partecipante, se il gruppo ha 0 partecipanti viene automaticamente eliminato*

*Procedure:*

*\**

*TO DO*

*-Inserimento di un attributo “Descrizione” a gruppo*

*-Inserimento di un attributo “Note” a User*

*-Inserimento di un attributo “Preferiti” a contatto*

*-Associazione tra social e email*

*-Associazione tra gruppo e utente*

*Cancellare ContactID a Social*

*TO DO LATER*

*Gestire eventuali inserimenti di indirizzi fisici, numeri di telefono ed e-mail uguali (in database o in java?)*