Wassup

Progettazione e Implementazione di una Base di Dati per Software di Messaggistica Peer-to-Peer

Andrea Pedini - Elia Renzoni

Sessione Estiva a.a 2024/2025

Relazione per il Progetto del Corso di Basi di Dati - Informatica Applicata Università degli Studi di Urbino Carlo Bo.

Indice

1 Specifica del Problema

Si intende sviluppare un'applicazione di messaggistica simile all'app **Whatsapp**, che possa mettere in comunicazione diversi utenti in giro per il mondo, consentendo loro di comunicare non solo via messaggi testuali, ma anche tramite chiamate vocali. **Wassup**, il nome dell'applicazione da sviluppare, a differenza delle sue omologhe deve essere Peer-to-Peer, in particolare la sua infrastruttura di rete deve rispondere ad un modello di overlay network del tutto simile a quello di una rete P2P non strutturata e completamente decentralizzata. Data l'architettura decentrallizata non saranno presenti punti di comunicazione centralizzata, salvo le cosidette *hot cache* - per riprendere il termine utilizzato dal protocollo Gnutella - che consentono l'entrata nella rete dei nuovi nodi.
La relazione verterà principalmente verso la progettazione e la realizzazione del database, pertanto gli elementi riguardanti i cosidetti *communication steps* di un algoritmo distribuito non saranno trattati. Wassup deve consetire ad ogni singolo utente che accede per la prima volta all'app di registrarsi e di modificare le impostazioni a proprio piacimento. La registrazione si deve effettuare inserendo il proprio nome, cognome, nome utente e numero di telefono. Una volta fatto ciò l'utente è in grado di cercare altre persone nella stessa rete, per fare ciò deve inviare una query di ricerca alla rete indicando il nome utente e il numero di telefono della persona da cercare. Se il flooding della query va a buon fine l'utente, che ha iniziato la ricerca, si vedrà ritornare l'indirizzo IP del nodo della persona cercata. A questo punto sarà possibile inziare una chat; questa può contenere le seguenti categorie di messaggi:

- Messaggi Testuali;
- Messaggi Vocali;
- Messaggi Multimediali;
- Sticker;
- Sondaggi;
- Reaction ai Messaggi;
- Bozze di Messaggi.

Queste categorie di messaggi sono presenti anche nei gruppi di cui un utente può far parte.
Ogni utente avrà una lista di contatti, questi possono essere eventualmente inseriti in una blacklist e l'inserimento in una blacklist produce lo stesso effetto di un utente bloccato, quindi non ne saranno più visibili i nuovi messaggi o chiamate.
Le chiamate possono essere sia tra due persone che di gruppo; l'informazione dell'avvenuta chiamata dovrà essere memorizzata e dovrà contenere la data d'inzio, la data di fine e le informazioni dei partecipanti.
La natura P2P dell'applicazione non consente di avere un sistema di notifiche al pari di applicazioni come Telegram o Instagram; le notifiche di nuovi messaggi o chiamate saranno reperibili solo quando il processo mittente sarà attivo, in caso contrario le notifiche, così come i messaggi da inviare, saranno effettivamente inoltrate solo quando l'utente di destinazione sarà attivo.

2 Progettazione Concettuale

2.1 Modellazione Utente, Contatto e Gruppo

Un utente può disporre di un numero `N` di contatti e gruppi. La relazione è `(0,N)` perchè all'inizio non possiede nessuna delle precedenti relazioni.

Le entità Contatto e Gruppo sono le uniche entità in relazione diretta `(1,1)` con l'utente in quanto il sistema è P2P, quindi si considera utente colui che possiede il nodo, mentre tutti gli altri utenti della rete sono visti come contatti. In accordo con queste decisioni un contatto o un gruppo può appartenere solamente ad un utente.

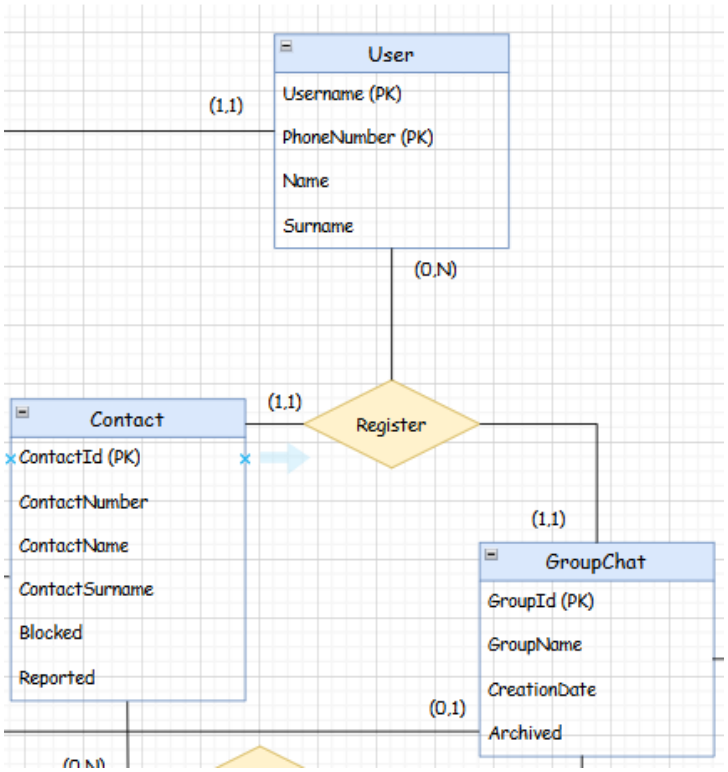

Figura 1: Utente, Contatto e Gruppo

2.2 Modellazione Contatto, Chat e Membro di un Gruppo

L'entità Contatto viene collegata con l'entità Chat e Gruppo, ma con relazioni diverse. Infatti un contatto è in relazione `(0,1)` conl'entità Chat in quanto un utente può disporre di un contatto ma non aver intrattenuto una chat con esso, ma in entrambi i casi le chat sono singolari e in relazione esattamente con un contatto. Diversa è la relazione tra Gruppo e Contatto, dove il Gruppo ha una cardinalità di `(1, N)` con l'entità Contatto.

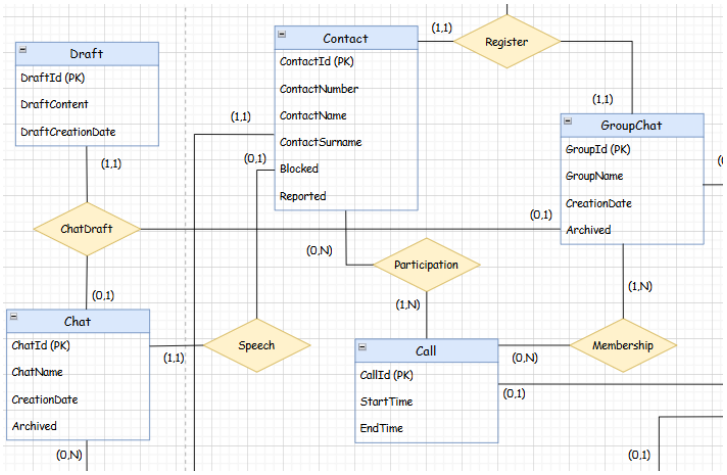

Figura 2: Chat, Gruppo e Contatto

2.3 Modellazione dell'entità Bozza e Messaggio

L'entità Messaggio è stata specializzata con le entità che rispecchiano le diverse tipologie di messaggio. Queste tipologie possono far parte di Media, Sondaggio e Allegato. L'entità Media riguarda le immagini, video e messaggi vocali; mentre l'entità Attachment riguarda gli allegati come file di testo. La Bozza, come suggerisce il nome stesso, indica un messaggio

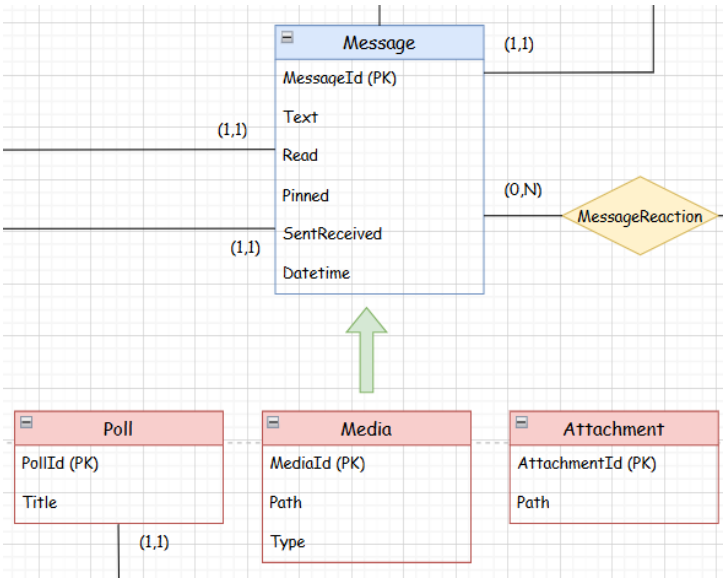

Figura 3: Specializzazione dell'Entità Messaggio

che non è ancora stato inviato.
Le entità Messaggio e Bozza sono collegate con le entità Chat e Gruppo. Sia una chat che un gruppo possono contenere un numero arbitrario di messaggi, questi possono anche non esserci nel caso in cui una chat o un gruppo siano appena stati creati. Quindi le entità Chat e Gruppo hanno una cardinalità `(0, N)` con l'entità Messaggio. Viceversa l'entità Messaggio è in relazione `(1, 1)` sia con Chat che con Gruppo, questo perchè per noi ogni messaggio deve essere identificato univocamente e può appartenere ad un gruppo o chat specifico.

Per quanto riguarda l'entità Bozza un gruppo e una chat possono contenere una sola bozza e questa può appartenere ad un solo gruppo o chat. Sono quindi mutualmente in una relazione di cardinalità `(1, 1)`.

2.4 Modellazione dell'entità Reaction

Un utente può reagire ad un messaggio di qualsiasi tipo. Un messaggio può contenere un numero arbitrario di reazioni, queste possono anche non essserci; mentre una reaction può appartanere solo ad un messaggio specifico. Quindi l'entità Messaggio è in relazione `(0, N)` con l'entià Reaction, mentre l'entità Reaction è in relazione `(1, 1)` con l'entità Messaggio.

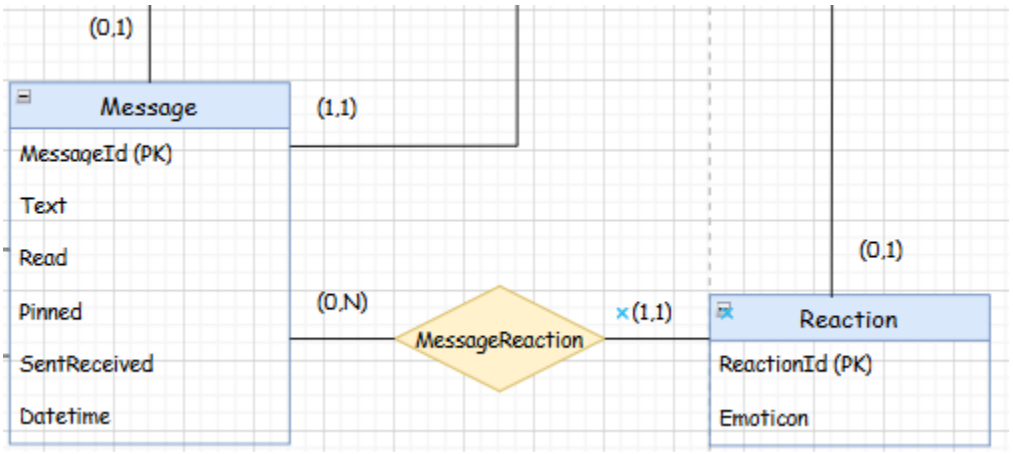

Figura 4: Reaction e Messaggio

2.5 Modellazione dell'entità Impostazione

L'utente una volta registrato deve impostare le impostazioni per la prima volta, queste possono essere modificate successivamente.

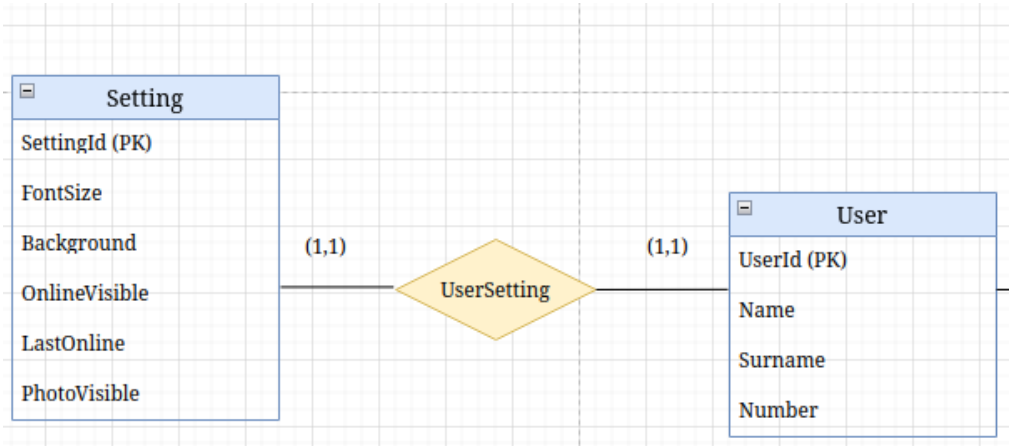

Figura 5: Impostazione, Utente

2.6 Modellazione dell'entità Notifica

Una notifica può avvenire a causa della ricezione di un messaggio, di una reaction ad un messaggio o di una chiamata. L'entità Notification è quindi in relazione con le entità Message, Reaction e Call. In generale una notifica è relativa ad un messaggio, reaction o chiamata univoca, in quanto ognuna di queste è univoca. Tuttavia, pensando al sistema nell'utilizzo effettivo, un utente che sta visualizzando una chat, se riceve un messaggio dalla persona in questione non deve scatenare alcuna notifica perchè visualizza immediatamente il messaggio.

Pertanto la relazione tra Message, Reaction e Call con l'entità Notification ha cardinalità `(0,1)`, e `(1,1)` nel verso opposto.

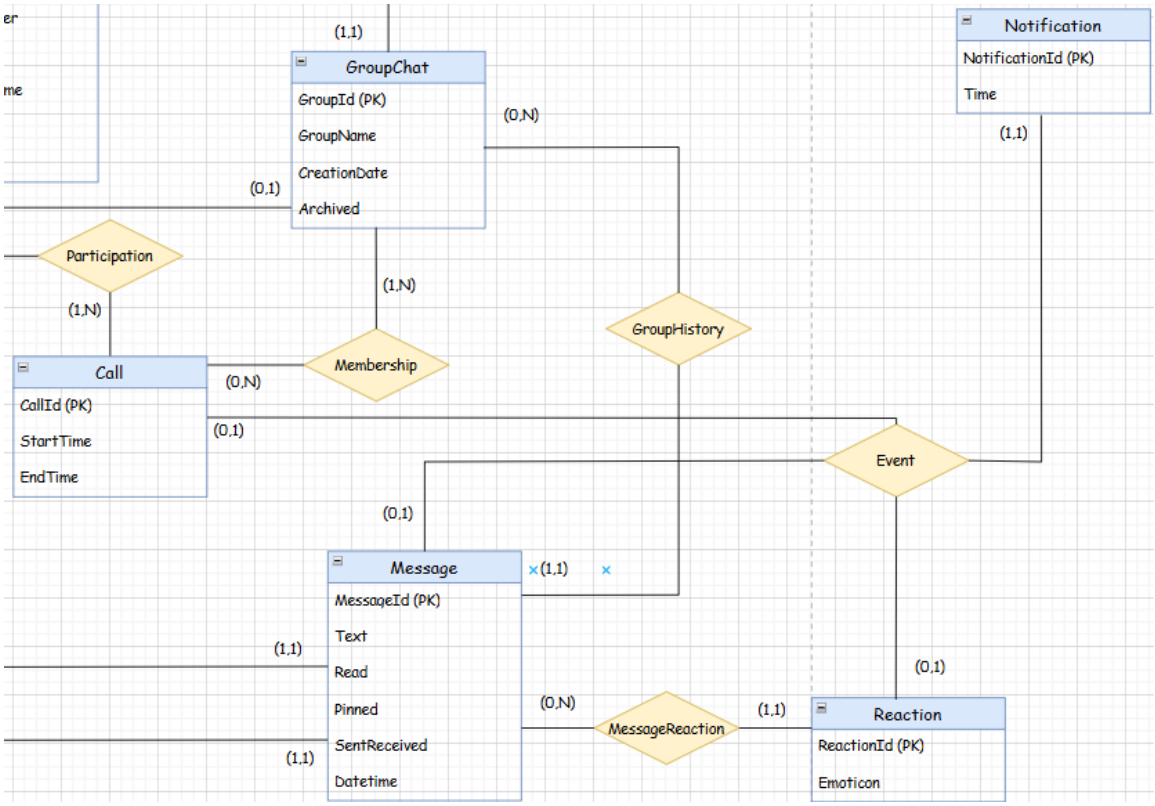

Figura 6: Notifica, Messaggio e Reazione

3 Progettazione Logica

3.1 Ristrutturazione Modello Concettuale

3.1.1 Eliminazione delle Generalizzazioni

Nel modello E-R viene specializzata solo l'entità **Message** attraverso le entità **Media**, **Poll** e **Attachement**. La strategia di eliminazione delle generalizzazioni è quella dell'accorpamento delle entità figlie in quella genitore. Questo soluzione introduce inevitabilmente dei valori `NULL` che possono essere largamente evitati seguendo altre strade. Tuttavia accorpando le entità figlie in quella genitore si facilita il processo di query delle entry della tabella in caso di caricamento del contenuto delle chat, questo perchè si avrebbe un solo punto di ricerca.
Il modello E-R può essere così aggiornato:

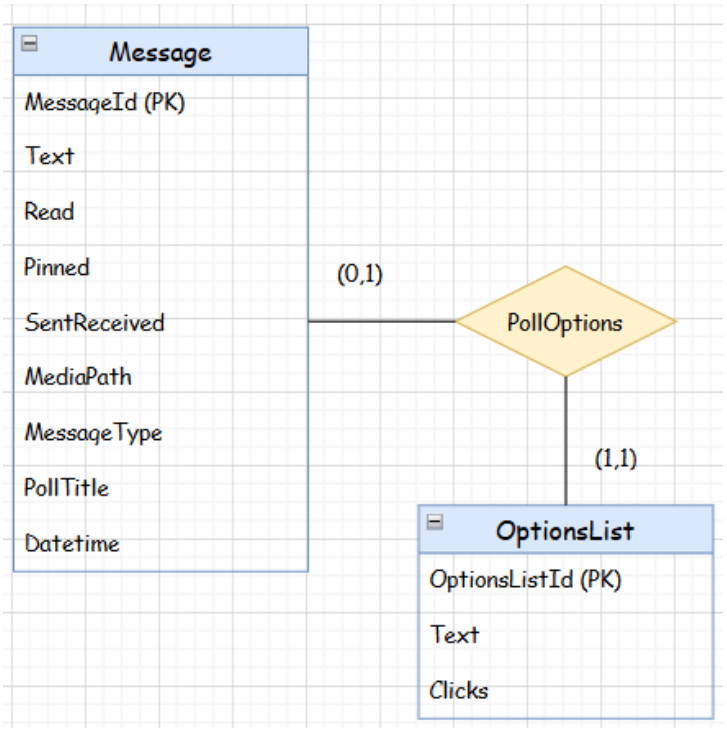

Figura 7: Aggiornamento Modello E-R

3.1.2 Eliminazioni Attributi Multivalore e Modifica dei Concetti

Il modello E-R non prevede attributi multi-valore in quanto ogni entità con attributi con `N` possibili valori è stata relazionata con altre entità in grado contenere questi valori.
Il seguente listato mostra come sono state gestite le entità con attributi aventi valori multipli.

```
/* I nomi in maiuscolo identificano le entita coinvolte.
* Le freccie indicano i collegamenti tra entita.
* Le entita che si trovano a destra sono quelle che
* contengono i valori multipli degli attributi delle
* entita che si trovano a sinstra.
*/

CALL -> CONTACT;
USER -> CONTACT;
USER -> GROUPCHAT;
GROUPCHAT -> CONTACT;
MESSAGE -> REACTION;
CHAT -> MESSAGE;
GROUPCHAT -> MESSAGE;
```

3.2 Traduzione in Modello Logico

```
/*
* Modello Logico
* PK -> Primary Key.
* FK -> Foreign Key.
**/

USER(Username: PK, PhoneNumber, Name, Surname).


CALL(CallId: PK, StartTime, EndTime).


CHAT(ChatId: PK, CreationDate, Archived, ContactId: FK).


GROUPCHAT(GroupId: PK, GroupName, CreationDate, Archived, GroupOwner: FK).


CONTACT(ContactId: PK, ContactName, ContactNumber, ContactSurname, Blocked, Reported, ContactOwner: FK).


REACTION(ReactionId: PK, Emoticon, MessageID: FK).


MESSAGE(MessageId: PK, Text, Read, Pinned, SentReceived, DateTime,
MessageType, MediaPath, PollTitle, ChatId: FK, GroupId: FK, AuthorId: FK).


DRAFT(DraftId: PK, DraftContent, DraftCreationDate, ChatId: FK, GroupId: FK).


SETTING(SettingId: PK, FontSize, Background, OnlineVisible, LastOnline, PhotoVisible, User: FK).


MEMBERSHIP(GroupId: FK, ContactId: FK).


NOTIFICATION(NotificationId: PK, Time, MessageId: FK, ReactionId: FK, CallId: FK).


OPTIONLIST(OptionListId: PK, Text, Clicks, MessageId: FK).


PARTECIPATION(CallId: FK, ContactId: FK).
```

4 RDBMS Scelto

Abbiamo scelto SQLite come DBMS per la sua leggerezza e per la sua semplicitià. Queste caratteristiche si riflettono perfettamente con la natura Peer-to-Peer dell'applicazione. Infatti il database non sarà in hosting dentro macchine prestanti come quelle affitabili in cloud, bensì è molto probabile che sia memorizzato localmente dentro un personal computer o persino un telefono, per questo ci è sembrato necessario selezionare un DBMS molto minimale e performante allo stesso tempo.

5 Implementazione delle Query

5.1 Data Definition Language

```
/* Creazione della Tabella Call */
CREATE TABLE Call (
    CallId INTEGER,
    StartTime TEXT NOT NULL,
    EndTime TEXT NOT NULL,
    PRIMARY KEY(CallId)
)

/* Creazione della Tabella Setting */
CREATE TABLE Setting (
    SettingId INTEGER,
    FontSize INTEGER NOT NULL DEFAULT 12,
    Background TEXT NOT NULL,
    OnlineVisible INTEGER NOT NULL DEFAULT 1,
    LastOnline INTEGER NOT NULL DEFAULT 1,
    PhotoVisible INTEGER NOT NULL DEFAULT 1,
    User TEXT NOT NULL,
    PRIMARY KEY(SettingId),
    FOREIGN KEY(User) REFERENCES User(Username),
    CHECK(OnlineVisible IN (0, 1)),
    CHECK(LastOnline IN (0, 1)),
    CHECK(PhotoVisible IN (0, 1))
)

/* Creazione della tabella Contact */
CREATE TABLE Contact (
    ContactId INTEGER,
    ContactName TEXT NOT NULL,
    ContactNumber TEXT NOT NULL,
    ContactSurname TEXT NOT NULL,
    Blocked INTEGER DEFAULT 0,
    Reported INTEGER DEFAULT 0,
    ContactOwner TEXT NOT NULL,
    PRIMARY KEY(ContactId),
    FOREIGN KEY(ContactOwner) REFERENCES User(Username),
    CHECK(Blocked IN (0, 1)),
    CHECK(Reported IN (0, 1))
)

/* Creazione della tabella GroupChat */
CREATE TABLE GroupChat (
    GroupName TEXT NOT NULL,
    GroupOwner TEXT NOT NULL,
    GroupId INTEGER,
    CreationDate TEXT NOT NULL,
    Archived INTEGER DEFAULT 0,
    PRIMARY KEY(GroupId),
    FOREIGN KEY(GroupOwner) REFERENCES User(Username)
)

/* Creazione della tabella Membership */
CREATE TABLE Membership (
    GroupId TEXT,
    ContactId INTEGER,
    PRIMARY KEY(GroupId,ContactId),
    FOREIGN KEY(ContactId) REFERENCES Contact(ContactId),
    FOREIGN KEY(GroupId) REFERENCES GroupChat(GroupId)
)
```

```
/* Creazione della tabella User */
CREATE TABLE User (
    Name TEXT NOT NULL,
    Surname TEXT NOT NULL,
    PhoneNumber TEXT NOT NULL,
    Username TEXT,
    PRIMARY KEY(Username)
)

/* CReazione della tabella Participation */
CREATE TABLE Participation (
    CallId INTEGER,
    ContactId INTEGER,
    FOREIGN KEY(CallId) REFERENCES Call(CallId),
    FOREIGN KEY(ContactId) REFERENCES Contact(ContactId)
)

/* Creazione della tabella Chat */
CREATE TABLE Chat (
    CreationDate DATE NOT NULL,
    Archived INTEGER DEFAULT 0,
    ChatId INTEGER,
    ContactId INTEGER NOT NULL,
    PRIMARY KEY(ChatId),
    FOREIGN KEY(ContactId) REFERENCES Contact(ContactId),
    CHECK(Archived IN (0, 1))
)

/* Creazione della tabella Message */
CREATE TABLE Message (
    MessageId INTEGER,
    Text TEXT NOT NULL,
    Read INTEGER NOT NULL,
    Pinned INTEGER DEFAULT 0,
    ChatId INTEGER,
    Time TEXT NOT NULL,
    SentReceived INTEGER NOT NULL,
    GroupId INTEGER,
    Date TEXT NOT NULL,
    MediaPath TEXT,
    MessageType TEXT NOT NULL,
    PollTitle TEXT,
    AuthorId INTEGER,
    PRIMARY KEY(MessageId),
    FOREIGN KEY(AuthorId) REFERENCES Contact(ContactId),
    FOREIGN KEY(ChatId) REFERENCES Chat(ChatId),
    FOREIGN KEY(GroupId) REFERENCES GroupChat(GroupId),
    CHECK(Read IN (0, 1)),
    CHECK(Pinned IN (0, 1)),
    CHECK(SentReceived IN (0, 1))
)

/* Creazione della tabella OptionsList */
CREATE TABLE OptionsList (
    OptionsListId INTEGER,
    Text TEXT NOT NULL,
    Clicks INTEGER DEFAULT 0,
    PollId INTEGER NOT NULL,
    PRIMARY KEY(OptionsListId),
    FOREIGN KEY(PollId) REFERENCES Message(MessageId)
)

/* Creazione della tabella Draft */
```

```
CREATE TABLE Draft (
    DraftId INTEGER,
    DraftContent TEXT NOT NULL,
    DraftCreationDate TEXT NOT NULL,
    ChatId INTEGER,
    GroupId INTEGER,
    PRIMARY KEY(DraftId),
    FOREIGN KEY(ChatId) REFERENCES Chat(ChatId),
    FOREIGN KEY(GroupId) REFERENCES GroupChat(GroupId)
)

/* Creazione della tabella Reaction */
CREATE TABLE Reaction (
    ReactionId INTEGER,
    Emoticon TEXT NOT NULL,
    MessageId INTEGER NOT NULL,
    PRIMARY KEY(ReactionId),
    FOREIGN KEY(ReactionId) REFERENCES Message(MessageId)
)

/* Creazione della tabella Notification */
CREATE TABLE Notification (
    NotificationId INTEGER,
    Time TEXT NOT NULL,
    MessageId INTEGER,
    ReactionId INTEGER,
    CallId INTEGER,
    PRIMARY KEY(NotificationId),
    FOREIGN KEY(CallId) REFERENCES Call(CallId),
    FOREIGN KEY(MessageId) REFERENCES Message(MessageId),
    FOREIGN KEY(ReactionId) REFERENCES Reaction(ReactionId)
)
```

5.2 Data Manipulation Language

5.2.1 Inserimento di Nuovi Valori

```
    -- INSERT nella tabella User
INSERT INTO User (Name, Surname, PhoneNumber, Username)
VALUES
('Mario', 'Rossi', '1234567890', 'mrossi'),
('Luigi', 'Verdi', '2345678901', 'lverdi'),
('Giulia', 'Bianchi', '3456789012', 'gbianchi'),
('Anna', 'Neri', '4567890123', 'aneri'),
('Marco', 'Gialli', '5678901234', 'mgialli'),
('Sara', 'Rosa', '6789012345', 'srosa'),
('Paolo', 'Blu', '7890123456', 'pblu'),
('Luca', 'Marrone', '8901234567', 'lmarrone'),
('Elisa', 'Grigi', '9012345678', 'egrigi'),
('Franco', 'Viola', '0123456789', 'fviola');

-- INSERT nella tabella Setting
INSERT INTO Setting (SettingId, FontSize, Background, OnlineVisible, LastOnline, PhotoVisible, User)
VALUES
(1, 14, 'white', 1, 1, 1, 'mrossi'),
(2, 12, 'dark', 0, 1, 1, 'lverdi'),
(3, 16, 'blue', 1, 0, 1, 'gbianchi'),
(4, 12, 'green', 1, 1, 0, 'aneri'),
(5, 15, 'grey', 1, 1, 1, 'mgialli'),
(6, 13, 'yellow', 0, 0, 1, 'srosa'),
(7, 12, 'pink', 1, 1, 1, 'pblu'),
(8, 14, 'orange', 1, 0, 1, 'lmarrone'),
```

```
(9, 12, 'black', 1, 1, 1, 'egrigi'),
(10, 13, 'cyan', 0, 1, 0, 'fviola');

-- INSERT nella tabella Contact
INSERT INTO Contact (ContactId, ContactName, ContactNumber, ContactSurname, Blocked, Reported, ContactOwner)
VALUES
(1, 'Giorgio', '1111111111', 'Verdi', 0, 0, 'mrossi'),
(2, 'Laura', '2222222222', 'Blu', 1, 0, 'lverdi'),
(3, 'Simone', '3333333333', 'Rossi', 0, 1, 'gbianchi'),
(4, 'Martina', '4444444444', 'Neri', 0, 0, 'aneri'),
(5, 'Davide', '5555555555', 'Gialli', 0, 0, 'mgialli'),
(6, 'Irene', '6666666666', 'Rosa', 1, 1, 'srosa'),
(7, 'Enrico', '7777777777', 'Blu', 0, 0, 'pblu'),
(8, 'Tania', '8888888888', 'Marrone', 0, 1, 'lmarrone'),
(9, 'Valeria', '9999999999', 'Grigi', 0, 0, 'egrigi'),
(10, 'Gianluca', '0000000000', 'Viola', 1, 0, 'fviola');

-- INSERT nella tabella GroupChat
INSERT INTO GroupChat (GroupName, GroupOwner, GroupId, CreationDate, Archived)
VALUES
('Famiglia', 'mrossi', 1, '2024-01-01', 0),
('Lavoro', 'lverdi', 2, '2024-02-01', 0),
('Amici', 'gbianchi', 3, '2024-03-01', 1),
('Sport', 'aneri', 4, '2024-04-01', 0),
('Viaggi', 'mgialli', 5, '2024-05-01', 0);

-- INSERT nella tabella Membership
INSERT INTO Membership (GroupId, ContactId)
VALUES
(1, 1),
(1, 2),
(2, 3),
(2, 4),
(3, 5),
(3, 6),
(4, 7),
(4, 8),
(5, 9),
(5, 10);

-- INSERT nella tabella Call
INSERT INTO Call (CallId, StartTime, EndTime)
VALUES
(1, '2024-05-01 10:00:00', '2024-05-01 10:30:00'),
(2, '2024-05-02 11:00:00', '2024-05-02 11:45:00'),
(3, '2024-05-03 12:00:00', '2024-05-03 12:20:00'),
(4, '2024-05-04 13:00:00', '2024-05-04 13:50:00'),
(5, '2024-05-05 14:00:00', '2024-05-05 14:30:00'),
(6, '2024-05-06 15:00:00', '2024-05-06 15:40:00'),
(7, '2024-05-07 16:00:00', '2024-05-07 16:10:00'),
(8, '2024-05-08 17:00:00', '2024-05-08 17:45:00'),
(9, '2024-05-09 18:00:00', '2024-05-09 18:20:00'),
(10, '2024-05-10 19:00:00', '2024-05-10 19:30:00');

-- INSERT nella tabella Participation
INSERT INTO Participation (CallId, ContactId)
VALUES
(1, 1),
(1, 2),
(2, 3),
(2, 4),
(3, 5),
(3, 6),
```

```
(4, 7),
(4, 8),
(5, 9),
(5, 10);

-- INSERT nella tabella Chat
INSERT INTO Chat (CreationDate, Archived, ChatId, ContactId)
VALUES
('2024-05-01', 0, 1, 1),
('2024-05-02', 1, 2, 2),
('2024-05-03', 0, 3, 3),
('2024-05-04', 0, 4, 4),
('2024-05-05', 1, 5, 5),
('2024-05-06', 0, 6, 6),
('2024-05-07', 0, 7, 7),
('2024-05-08', 0, 8, 8),
('2024-05-09', 0, 9, 9),
('2024-05-10', 0, 10, 10);

-- INSERT nella tabella Message
INSERT INTO Message (MessageId, Text, Read, Pinned, ChatId, Time, SentReceived, GroupId, Date, MediaPath, MessageType, Po
VALUES
(1, 'Ciao, come va?', 1, 0, 1, '10:00:00', 1, NULL, '2024-05-01', NULL, 'text', NULL, 1),
(2, 'Bene, grazie!', 1, 0, 1, '10:01:00', 0, NULL, '2024-05-01', NULL, 'text', NULL, 2),
(3, 'Guarda questa foto!', 1, 0, 2, '11:00:00', 1, NULL, '2024-05-02', '/media/photo1.jpg', 'image', NULL, 2),
(4, 'Interessante', 1, 0, 3, '12:00:00', 1, NULL, '2024-05-03', NULL, 'text', NULL, 3),
(5, 'Parliamo dopo', 0, 0, 4, '13:00:00', 1, NULL, '2024-05-04', NULL, 'text', NULL, 4),
(6, 'Video allegato', 1, 1, 5, '14:00:00', 1, NULL, '2024-05-05', '/media/video1.mp4', 'video', NULL, 5),
(7, 'Sondaggio: Dove andiamo?', 1, 0, NULL, '15:00:00', 1, 5, '2024-05-06', NULL, 'poll', 'Destinazione preferita', 6),
(8, 'Risposta al sondaggio', 1, 0, NULL, '16:00:00', 1, 5, '2024-05-06', NULL, 'text', NULL, 7),
(9, 'File condiviso', 1, 0, 6, '17:00:00', 1, NULL, '2024-05-07', '/docs/file.pdf', 'document', NULL, 8),
(10, 'Arrivo tra poco', 0, 0, 7, '18:00:00', 1, NULL, '2024-05-08', NULL, 'text', NULL, 9);

-- INSERT nella tabella OptionsList
INSERT INTO OptionsList (OptionsListId, Text, Clicks, PollId)
VALUES
(1, 'Montagna', 3, 7),
(2, 'Mare', 5, 7),
(3, 'Citt darte', 2, 7);


INSERT INTO Draft (DraftId, DraftContent, DraftCreationDate, ChatId, GroupId)
VALUES
(1, 'Scrivo dopo', '2024-05-09 12:00:00', 8, NULL),
(2, 'Non dimenticare la riunione', '2024-05-09 13:00:00', NULL, 2),
(3, 'Porta il documento firmato', '2024-05-09 14:00:00', 9, NULL);

-- INSERT nella tabella Reaction
INSERT INTO Reaction (ReactionId, Emoticon, MessageId)
VALUES
(1, '', 1),
(2, '', 2),
(3, '', 7);

-- INSERT nella tabella Notification
INSERT INTO Notification (NotificationId, Time, MessageId, ReactionId, CallId)
VALUES
(1, '10:01:30', 1, NULL, NULL),
(2, '10:02:00', NULL, 1, NULL),
(3, '15:30:00', 7, NULL, NULL),
(4, '16:05:00', NULL, 3, NULL),
(5, '18:30:00', NULL, NULL, 1),
(6, '19:00:00', 10, NULL, NULL);
```

5.2.2 Ricerca dei Dati

```
/* Login */
SELECT *
FROM User
WHERE Username = 'mrossi';

/* Caricamento delle Chat */
SELECT Chat.ChatId, Contact.ContactName, Contact.ContactSurname, Contact.ContactNumber, Chat.CreationDate,
Chat.Archived
FROM Chat INNER JOIN Contact ON Chat.ContactId = Contact.ContactId
WHERE Contact.ContactOwner = 'mrossi';

/* Caricamento dei Gruppi */
SELECT *
FROM GroupChat
WHERE GroupOwner = 'mrossi';

/* Caricamento dei Messaggi delle Chat */
SELECT Text, MessageType, MediaPath, Read, SentReceived, Date, Time
FROM Message
WHERE ChatId = 5;



/* Caricamento dei Messaggi dei Gruppi */
SELECT M.Text, M.MessageType, M.MediaPath, M.Read, M.SentReceived, M.Date, M.Time, C.ContactName
FROM Message M
JOIN Contact C ON M.AuthorId = C.ContactId
WHERE GroupId = 5;

/* Caricamento delle Chiamate */
SELECT *
FROM Call
WHERE CallId IN (
    SELECT CallId FROM Participation WHERE ContactId IN (
           SELECT ContactId FROM Contact WHERE ContactOwner = 'mrossi'
    )
)

/* Caricamento delle Bozze per Chat */
SELECT DraftContent
FROM Draft
WHERE ChatId = 9;

/* Caricamento delle Bozze per un Gruppo */
SELECT DraftContent
FROM Draft
WHERE GroupId = 2;

/* Caricamento delle Impostazioni */
SELECT *
FROM Setting WHERE User = 'mrossi';

/* Caricamento dei Membri di un Gruppo */
SELECT *
FROM Contact
WHERE ContactId IN (
    SELECT ContactId FROM Membership WHERE GroupId = '5'
);

/* Caricamento di tutti i contatti*/
SELECT *
```

```sql
FROM Contact
WHERE ContactOwner = 'mrossi';

/* Caricamento di tutti i contatti bloccati */
SELECT *
FROM Contact
WHERE ContactOwner = 'mrossi' AND Blocked = 1;

SELECT OptionsList.Text, OptionsList.Clicks
FROM OptionsList JOIN Message ON OptionsList.MessageId = Message.MessageId
WHERE Message.ChatName = <Given Chat Name>;

/* Caricamento Completo dei Songaggi in un Gruppo */
SELECT M.MessageId, O.Text, O.Clicks
FROM Message M
JOIN OptionsList O ON M.MessageId = O.PollId
AND M.GroupId = '5';


/* Caricamento delle Reaction a tutti i messaggi di una Chat */
SELECT Emoticon, MessageId
FROM Reaction
WHERE MessageId IN (
      SELECT MessageId
      FROM Message
      WHERE ChatId = 1
);

/* Caricamento delle Reaction a tutti i messaggi di un Gruppo */
SELECT Emoticon, MessageId
FROM Reaction
WHERE MessageId IN (
      SELECT MessageId
      FROM Message
      WHERE GroupId = 5
);

/* Caricamento di tutte le notifiche di un utente */
SELECT DISTINCT N.*
FROM Notification N
LEFT JOIN Message M ON N.MessageId = M.MessageId
LEFT JOIN Reaction R ON N.ReactionId = R.ReactionId
LEFT JOIN Message RM ON R.MessageId = RM.MessageId
LEFT JOIN Call C ON N.CallId = C.CallId
LEFT JOIN Participation P ON C.CallId = P.CallId
LEFT JOIN Contact ContMsg ON M.AuthorId = ContMsg.ContactId
LEFT JOIN Contact ContReact ON RM.AuthorId = ContReact.ContactId
LEFT JOIN Contact ContCall ON P.ContactId = ContCall.ContactId
WHERE ContMsg.ContactOwner = 'mrossi'
   OR ContReact.ContactOwner = 'mrossi'
   OR ContCall.ContactOwner = 'mrossi';


/* Aggiornamento Impostazioni */
UPDATE Setting
SET FontSize = 1
WHERE User = 'mrossi';

UPDATE Setting
SET Background = 'dark'
WHERE User = 'mrossi';
```

```sql
UPDATE Setting
SET OnlineVisible = 0
WHERE User = 'mrossi';

UPDATE Setting
SET LastOnline = 0
WHERE User = 'mrossi';

UPDATE Setting
SET PhotoVisible = 0
WHERE User = 'mrossi';


/* Bloccare Contatti o Segnalarli */
UPDATE Contact
SET Blocked = 1
WHERE ContactOwner = 'egrigi' AND ContactId = 9;

UPDATE Contact
SET Reported = 1
WHERE ContactOwner = 'egrigi' AND ContactId = 9;

/* Archiviare Chat o Gruppi */
UPDATE Chat
SET Archived = 1
WHERE ContactId IN (
      SELECT ContactId
      FROM Contact
      WHERE ContactOwner = 'mrossi'
);

UPDATE GroupChat
SET Archived = 1
WHERE GroupOwner = 'mrossi' AND GroupId = 1;
```