

COSC 3P32 Group Project

Parker TenBroeck: 7376726, pt21zs

Ivy Gifford: 7404775, ag21os

division of labour

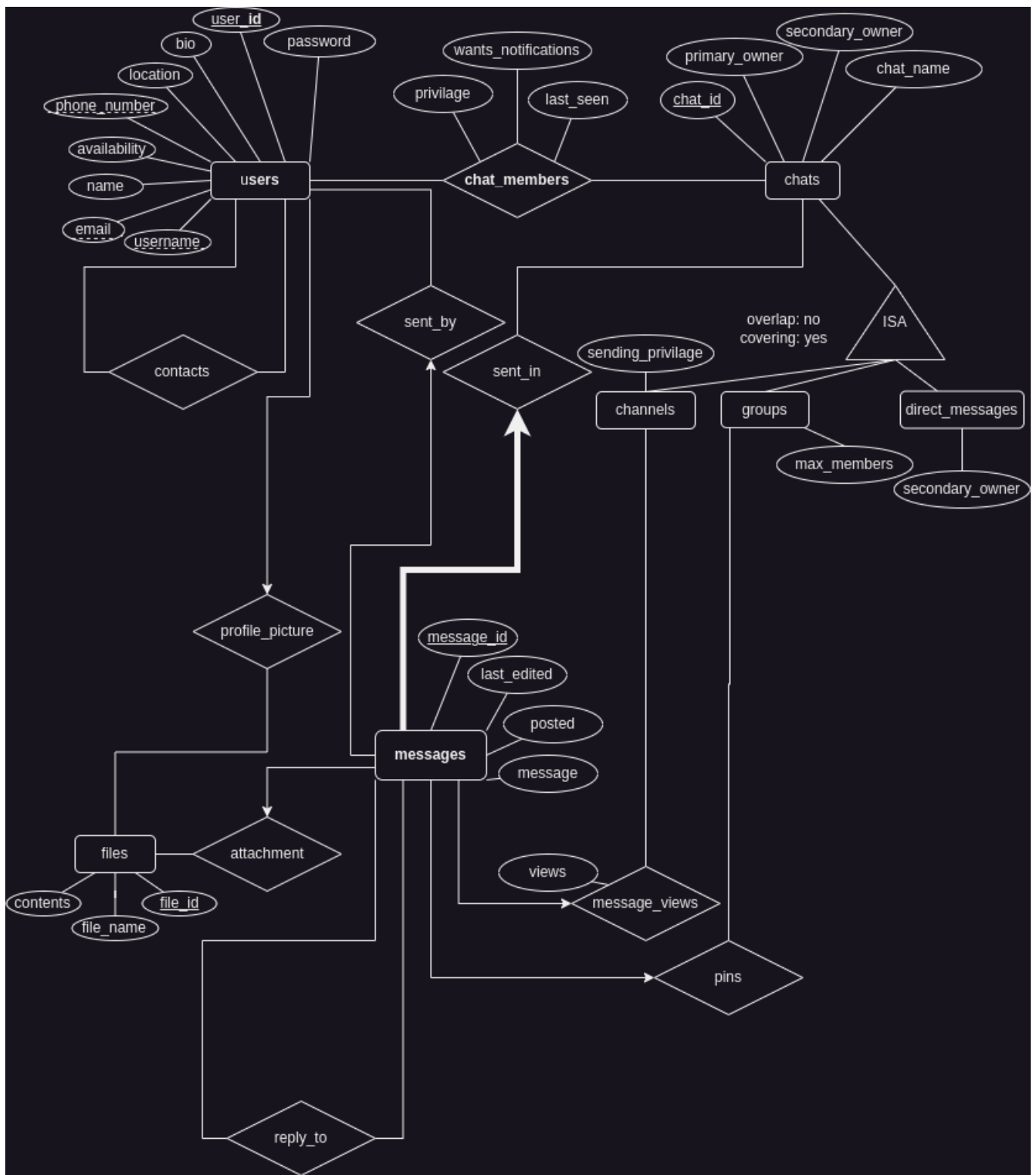
- q1 ivy
- q2 ivy
- q3 ivy
- q4 database - parker
- q4 demo data - ivy
- q5 statements - parker
- q5 demo data - ivy
- q6 parker

q1

the sent by relation does not have total participation as when a user is deleted, the message's sent by field will be set to null indicating the message was sent by a deleted user

the chats ISA constraint is not overlapping as a chat can be only a channel, group or direct message but not multiple and it is covering because it must always be one of the three

another thing not modeled is the join links on the groups. this is because we are using the format `http://sitename/join_chat/<chat_id>` as our join links so links are already stored as the id of the chat



q2

Files(**file_id: INTEGER**, file_name: TEXT, contents: BLOB)

Users(**user_id: INTEGER**, phone_number: TEXT, name: TEXT, email: TEXT, location: TEXT, username: TEXT, password: TEXT, bio: TEXT, availability: INTEGER, pfp_file_id: INTEGER)
pfp_file_id references Files

Contacts(**user_id: INTEGER**, **contact_user_id: INTEGER**)

user_id and contact_user_id reference Users

Messages(**message_id: INTEGER**, message: TEXT, reply_to: INTEGER, attachment: INTEGER, timestamp: BIGINT, last_edited: BIGINT, sender_id: INTEGER, chat_id: INTEGER, views: INTEGER, pinned: boolean)

sender_id references Users

chat_id references Chats

while there are 3 tables in the ISA relationship with chats, channels, groups, and direct_messages, defined in the erd, we found that we could have their unique fields nullable and differentiate between them by just getting the ones that are not null. This simplifies the queries needed greatly and made it better for our group's concrete implementation of the assignment for question 6

if there is a secondary owner its a direct message

if there is a sending privilege then its a channel

otherwise its a group

Chats (**chat_id: INTEGER**, primary_owner: INTEGER, secondary_owner: INTEGER, sending_privilege: INTEGER, track_views: BOOLEAN, max_members: INTEGER, chat_name: text)

primary_owner references Users

secondary_owner references Users

Chat_Members (**chat_id: INTEGER**, **member_id: INTEGER**, privilege: INTEGER, wants_notifications: BOOLEAN, last_seen: INTEGER,)

chat_id references Chats

member_id references Users

q3

Files(file_id: INTEGER, file_name: TEXT, contents: BLOB)

F(i, n, c)

i -> i, n, c

this is in BCNF as the only fd is the min key

Users(user_id: INTEGER, phone_number: TEXT, name: TEXT, email: TEXT, location: TEXT, username: TEXT, password: TEXT, bio: TEXT, availability: INTEGER, pfp_file_id: INTEGER)

U(id, pn, n, e, l, u, p, b, a)

id -> id, pn, n, e, l, u, p, b, a

pn -> id, pn, n, e, l, u, p, b, a

e -> id, pn, n, e, l, u, p, b, a

u -> id, pn, n, e, l, u, p, b, a

id, pn, e, and u are all minimum candidate keys and are the only FDs on the relation so this relation is in BCNF

Contacts(user_id: INTEGER, contact_user_id: INTEGER)

C(u, c)

u, c -> u, c

the only members of this relation are part of the minimum candidate key, this relation is in BCNF

Messages(message_id: INTEGER, message: TEXT, reply_to: INTEGER, attachment: INTEGER, timestamp: BIGINT, last_edited: BIGINT, sender_id: INTEGER, chat_id: INTEGER, views: INTEGER, pinned: boolean)

M(id, m, r, a, t, l, s, c, v, b)

id -> id, m, r, a, t, l, s, c, v, b

the only FD within this relation is id which is the min key therefore it is in BCNF

Chats (**chat_id: INTEGER**, primary_owner: INTEGER, secondary_owner: INTEGER, sending_privilage: INTEGER, track_views: BOOLEAN, max_members INTEGER, chat_name text)

CH(id, p, sec, sen, t, m, c)

id -> id, p, sec, sen, t, m, c

the only FD within this relation is id which is the min key therefore it is in BCNF

Chat_Members (**chat_id: INTEGER, member_id: INTEGER**, privilage: INTEGER, wants_notifications: BOOLEAN, last_seen INTEGER,)

CM(c, m, p, w, l)

c,m -> c, m, p, w, l

since c,m is the minimum candidate key and the only FD that holds this relation is in BCNF

q4

database:

```
CREATE TABLE files (  
    file_id SERIAL PRIMARY KEY,  
    file_name VARCHAR(150) NOT NULL,  
    contents BYTEA NOT NULL  
);  
  
CREATE TABLE users (  
    user_id SERIAL PRIMARY KEY,  
    phone_number VARCHAR(15) NOT NULL CHECK(length(name) >= 3) UNIQUE,  
    name VARCHAR(150) NOT NULL CHECK(length(name) >= 3),  
    email VARCHAR(150) NOT NULL CHECK(length(name) >= 3) UNIQUE,  
    location VARCHAR(150) NOT NULL CHECK(length(name) >= 3),
```

```

    username VARCHAR(50) NULL CHECK(length(name) >= 3) UNIQUE,
    password VARCHAR(50) NOT NULL CHECK(length(name) >= 3),
    bio VARCHAR(250) NOT NULL DEFAULT '',
    availability INTEGER NOT NULL DEFAULT 0,
    pfp_file_id INTEGER REFERENCES files(file_id)
);

CREATE TABLE contacts (
    user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
    contact_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE
CASCADE,
    PRIMARY KEY(user_id, contact_user_id)
);

CREATE TABLE chats (
    chat_id SERIAL PRIMARY KEY,
    primary_owner INTEGER NOT NULL REFERENCES users(user_id) ON DELETE
CASCADE,
    secondary_owner INTEGER NULL REFERENCES users(user_id) ON DELETE
CASCADE,

    sending_privilage INTEGER NOT NULL,

    track_views BOOLEAN NOT NULL,
    max_members INTEGER NULL,
    chat_name VARCHAR(100) NULL
);

CREATE TABLE messages (
    message_id SERIAL PRIMARY KEY,
    message TEXT,
    reply_to INTEGER NULL REFERENCES messages(message_id) ON DELETE SET
NULL,
    attachment_id INTEGER NULL REFERENCES files(file_id),
    posted BIGINT NOT NULL,
    last_edited BIGINT,
    sender_id INTEGER NULL REFERENCES users(user_id) ON DELETE SET NULL,
    chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
    views INTEGER NULL,
    pinned BOOLEAN NOT NULL DEFAULT FALSE
);

CREATE TABLE chat_members (
    chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
    member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
    privilage INTEGER NOT NULL,

```

```

wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
last_seen BIGINT NOT NULL DEFAULT 0,

PRIMARY KEY (chat_id, member_id)
);

```

```

home > ivy > .dbclient > query > 1708466642633@@@127.0.0.1@5432@cosc3p32@public > up.sql > ...
  Run | New Tab | Copy
1 CREATE TABLE files (
2     file_id SERIAL PRIMARY KEY,
3     file_name VARCHAR(150) NOT NULL,
4     contents BYTEA NOT NULL
5 ); 9ms
6
  Run | New Tab | Copy
7 CREATE TABLE users (
8     user_id SERIAL PRIMARY KEY,
9     phone_number VARCHAR(15) NOT NULL CHECK(length(name) >= 3) UNIQUE,
10    name VARCHAR(150) NOT NULL CHECK(length(name) >= 3),
11    email VARCHAR(150) NOT NULL CHECK(length(name) >= 3) UNIQUE,
12    location VARCHAR(150) NOT NULL CHECK(length(name) >= 3),
13    username VARCHAR(50) NULL CHECK(length(name) >= 3) UNIQUE,
14    password VARCHAR(50) NOT NULL CHECK(length(name) >= 3),
15    bio VARCHAR(250) NOT NULL DEFAULT '',
16    availability INTEGER NOT NULL DEFAULT 0,
17    pfp_file_id INTEGER REFERENCES files(file_id)
18 );
19
  Run | New Tab | Copy
20 CREATE TABLE contacts (
21     user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
22     contact_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,

```

If you are an premium user, can show definition by hover

Result x

Search results Cost: 9ms

```

CREATE TABLE files ( file_id SERIAL PRIMARY KEY, file_name VARCHAR(150) NOT NULL, contents BYTEA NOT NULL )

```

```

home > ivy > .dbclient > query > 1708466642633@@@127.0.0.1@5432@cosc3p32@public > up.sql > ...
  Run | New Tab | Copy
1 CREATE TABLE files (
2     file_id SERIAL PRIMARY KEY,
3     file_name VARCHAR(150) NOT NULL,
4     contents BYTEA NOT NULL
5 );
6
  Run | New Tab | Copy
7 CREATE TABLE users (
8     user_id SERIAL PRIMARY KEY,
9     phone_number VARCHAR(15) NOT NULL CHECK(length(name) >= 3) UNIQUE,
10    name VARCHAR(150) NOT NULL CHECK(length(name) >= 3),
11    email VARCHAR(150) NOT NULL CHECK(length(name) >= 3) UNIQUE,
12    location VARCHAR(150) NOT NULL CHECK(length(name) >= 3),
13    username VARCHAR(50) NULL CHECK(length(name) >= 3) UNIQUE,
14    password VARCHAR(50) NOT NULL CHECK(length(name) >= 3),
15    bio VARCHAR(250) NOT NULL DEFAULT '',
16    availability INTEGER NOT NULL DEFAULT 0,
17    pfp_file_id INTEGER REFERENCES files(file_id)
18 ); 13ms
19
  Run | New Tab | Copy
20 CREATE TABLE contacts (
21     user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
22     contact_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,

```

Result x

Search results Cost: 13ms

```

CREATE TABLE users ( user_id SERIAL PRIMARY KEY, phone_number VARCHAR(15) NOT NULL CHECK(length(name) >= 3) UNIQUE, name VARCHAR(150) NOT NULL CHECK(length(name) >= 3), email VARCHAR(150) NOT NULL CHECK(length(name) >= 3) UNIQUE, location VARCHAR(150) NOT NULL CHECK(length(name) >= 3), username VARCHAR(50) NULL CHECK(length(name) >= 3) UNIQUE, password VARCHAR(50) NOT NULL CHECK(length(name) >= 3), bio VARCHAR(250) NOT NULL DEFAULT "", availability INTEGER NOT NULL DEFAULT 0, pfp_file_id INTEGER REFERENCES files(file_id) )

```



```
20 > Run | New Tab | Copy
21 CREATE TABLE contacts (
22     user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
23     contact_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
24     PRIMARY KEY(user_id, contact_user_id)
25 ); 10ms
26 > Run | New Tab | Copy
27 CREATE TABLE chats (
28     chat_id SERIAL PRIMARY KEY,
29     primary_owner INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
30     secondary_owner INTEGER NULL REFERENCES users(user_id) ON DELETE CASCADE,
31     sending_privilage INTEGER NOT NULL,
32     track_views BOOLEAN NOT NULL,
33     max_members INTEGER NOT NULL,
34     chat_name VARCHAR(100) NULL
35 ); 10ms
36 > Run | New Tab | Copy
37 CREATE TABLE messages (
38     message_id SERIAL PRIMARY KEY,
39     message TEXT,
40     reply_to INTEGER NULL REFERENCES messages(message_id) ON DELETE SET NULL,
41     attachment_id INTEGER NULL REFERENCES files(file_id),
42     posted BIGINT NOT NULL,
43     last_edited BIGINT,
44     sender_id INTEGER NULL REFERENCES users(user_id) ON DELETE SET NULL,
45     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
46     views INTEGER NULL,
47     pinned BOOLEAN NOT NULL DEFAULT FALSE
48 ); 14ms
49 > Run | New Tab | Copy
50 CREATE TABLE chat_members (
51     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
52     member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
53     privilage INTEGER NOT NULL,
54     wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
55     last_seen BIGINT NOT NULL DEFAULT 0,
56     PRIMARY KEY (chat_id, member_id)
57 ); 8ms
```

```
25 > Run | New Tab | Copy
26 CREATE TABLE contacts (
27     user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
28     contact_user_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
29     PRIMARY KEY(user_id, contact_user_id)
30 ); 10ms
31 > Run | New Tab | Copy
32 CREATE TABLE chats (
33     chat_id SERIAL PRIMARY KEY,
34     primary_owner INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
35     secondary_owner INTEGER NULL REFERENCES users(user_id) ON DELETE CASCADE,
36     sending_privilage INTEGER NOT NULL,
37     track_views BOOLEAN NOT NULL,
38     max_members INTEGER NOT NULL,
39     chat_name VARCHAR(100) NULL
40 ); 10ms
41 > Run | New Tab | Copy
42 CREATE TABLE messages (
43     message_id SERIAL PRIMARY KEY,
44     message TEXT,
45     reply_to INTEGER NULL REFERENCES messages(message_id) ON DELETE SET NULL,
46     attachment_id INTEGER NULL REFERENCES files(file_id),
47     posted BIGINT NOT NULL,
48     last_edited BIGINT,
49     sender_id INTEGER NULL REFERENCES users(user_id) ON DELETE SET NULL,
50     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
51     views INTEGER NULL,
52     pinned BOOLEAN NOT NULL DEFAULT FALSE
53 ); 14ms
54 > Run | New Tab | Copy
55 CREATE TABLE chat_members (
56     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
57     member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
58     privilage INTEGER NOT NULL,
59     wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
60     last_seen BIGINT NOT NULL DEFAULT 0,
61     PRIMARY KEY (chat_id, member_id)
62 ); 8ms
```

```
37 > Run | New Tab | Copy
38 CREATE TABLE messages (
39     message_id SERIAL PRIMARY KEY,
40     message TEXT,
41     reply_to INTEGER NULL REFERENCES messages(message_id) ON DELETE SET NULL,
42     attachment_id INTEGER NULL REFERENCES files(file_id),
43     posted BIGINT NOT NULL,
44     last_edited BIGINT,
45     sender_id INTEGER NULL REFERENCES users(user_id) ON DELETE SET NULL,
46     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
47     views INTEGER NULL,
48     pinned BOOLEAN NOT NULL DEFAULT FALSE
49 ); 14ms
50 > Run | New Tab | Copy
51 CREATE TABLE chat_members (
52     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
53     member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
54     privilage INTEGER NOT NULL,
55     wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
56     last_seen BIGINT NOT NULL DEFAULT 0,
57     PRIMARY KEY (chat_id, member_id)
58 ); 8ms
```

```
51 > Run | New Tab | Copy
52 CREATE TABLE chat_members (
53     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
54     member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
55     privilage INTEGER NOT NULL,
56     wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
57     last_seen BIGINT NOT NULL DEFAULT 0,
58     PRIMARY KEY (chat_id, member_id)
59 ); 8ms
60 > Run | New Tab | Copy
61 CREATE TABLE chat_members (
62     chat_id INTEGER NOT NULL REFERENCES chats(chat_id) ON DELETE CASCADE,
63     member_id INTEGER NOT NULL REFERENCES users(user_id) ON DELETE CASCADE,
64     privilage INTEGER NOT NULL,
65     wants_notifications BOOLEAN NOT NULL DEFAULT TRUE,
66     last_seen BIGINT NOT NULL DEFAULT 0,
67     PRIMARY KEY (chat_id, member_id)
68 ); 8ms
```

adding test data:

```
-- insert some users
INSERT INTO users
```

```

(phone_number, name, email, location, username, password)
VALUES
    ('1231', 'ivy', 'ivy@gmail.com', 'ontario', 'ivytime', '123'),
    ('3212', 'parker', 'parker@brock.ca', 'ontario', 'heygrey', '123');

-- insert into files
INSERT INTO files (file_name, contents)
VALUES ('testfile', 'image data');

-- insert a contact
INSERT INTO contacts
    (user_id, contact_user_id)
VALUES
    (1, 2);

-- create a dm
BEGIN;
INSERT INTO chats
    (primary_owner, secondary_owner, sending_privilage, track_views,
max_members)
SELECT
    1, 2, 0, FALSE, 2
WHERE NOT EXISTS
    (SELECT 1 FROM chats WHERE (primary_owner=1 AND secondary_owner=2)
OR (primary_owner=2 AND secondary_owner=1))
RETURNING chat_id;

INSERT INTO chat_members
    (chat_id, member_id, privilage)
VALUES
    (var1, 1, 255),
    (var1, 2, 255);
COMMIT;

-- create group
BEGIN;
INSERT INTO chats
    (primary_owner, sending_privilage, track_views, max_members,
chat_name)
SELECT
    1, 0, FALSE, 2000, 'hello'
WHERE 100>(
    SELECT COUNT(*) FROM
        (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
    LEFT JOIN
        chat_members

```

```

        ON (t1.chat_id=chat_members.chat_id)
        WHERE member_id=0
    )
RETURNING chat_id;

INSERT INTO chat_members
(chat_id, member_id, privilege)
VALUES
(2, 1, 255);
COMMIT;

-- create a channel max members is null as channels do not have a max
BEGIN;
INSERT INTO chats
(primary_owner, sending_privilege, track_views, max_members,
chat_name)
SELECT
    1, 128, TRUE, null, 'this is a channel'
WHERE 100>(
    SELECT COUNT(*) FROM
        (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
    LEFT JOIN
        chat_members
    ON (t1.chat_id=chat_members.chat_id)
    WHERE member_id=1
)
RETURNING chat_id;

INSERT INTO chat_members
(chat_id, member_id, privilege)
VALUES
(3, 1, 255);
COMMIT;

-- join chat 2
INSERT INTO chat_members
(chat_id, member_id, privilege)
SELECT
    2, 2, 0
WHERE
100>(
    SELECT COUNT(*) FROM
        (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
    LEFT JOIN
        chat_members
    ON (t1.chat_id=chat_members.chat_id)

```

```

        WHERE member_id=3
    ) AND
    (
        (SELECT SUM(max_members) FROM chats WHERE chat_id=3)
        >
        (SELECT COUNT(*) FROM chat_members WHERE chat_id=3)
    );

-- join chat 3
INSERT INTO chat_members
    (chat_id, member_id, privilage)
SELECT
    3, 2, 0
WHERE
100>(
    SELECT COUNT(*) FROM
        (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
    LEFT JOIN
        chat_members
    ON (t1.chat_id=chat_members.chat_id)
    WHERE member_id=3
) AND
(
    (SELECT SUM(max_members) FROM chats WHERE chat_id=3)
    >
    (SELECT COUNT(*) FROM chat_members WHERE chat_id=3)
);

--insert some messages
INSERT INTO messages
    (sender_id, chat_id, message, attachment_id, posted, reply_to)
VALUES
    (1, 1, 'hello in a dm', null, 1714528509619, null),
    (2, 1, 'hey!', null, 1714528560684, null),
    (1, 2, 'howdy in a group!', null, 1714528649805, null),
    (2, 2, 'cool!', null, 1714528659417, null),
    (1, 3, 'zoinks in a channel', null, 1714528664131, null),
    (2, 3, 'channel time!', null, 1714528668494, null);

```

```
3  -- insert some users
4  > Run | New Tab | Copy
5  INSERT INTO users
6  (phone_number, name, email, location, username, password)
7  VALUES
8  ('1231', 'ivy', 'ivy@gmail.com', 'ontario', 'ivytime', '123'),
9  ('3212', 'parker', 'parker@brock.ca', 'ontario', 'heygrey', '123'); 8ms
10
11 -- insert into files
12 > Run | New Tab
13 INSERT INTO files (file_name, contents)
14 VALUES (:name, :data);
15
16 -- insert a contact
17 > Run | New Tab
18 INSERT INTO contacts
19 (user_id, contact_user_id)
20 VALUES
21 (0, 1);
22
23 -- create a dm
24 > Run | New Tab | Copy
25 INSERT INTO chats
```

Result x

Search results

Cost: 8ms

INSERT INTO users (phone_number, name, email, location, username, password) VALUES ('1231', 'ivy', 'ivy@gmail.com', 'ontario', 'ivytime', '123'), ('3212', 'parker', 'parker@brock.ca', 'ontario', 'heygrey', '123')

AffectedRows : 2

```
3  -- insert some users
4  > Run | New Tab | Copy
5  INSERT INTO users
6  (phone_number, name, email, location, username, password)
7  VALUES
8  ('1231', 'ivy', 'ivy@gmail.com', 'ontario', 'ivytime', '123'),
9  ('3212', 'parker', 'parker@brock.ca', 'ontario', 'heygrey', '123');
10
11 -- insert into files
12 > Run | New Tab
13 INSERT INTO files (file_name, contents)
14 VALUES ('testfile', 'image data'); 7ms
15
16 -- insert a contact
17 > Run | New Tab
18 INSERT INTO contacts
19 (user_id, contact_user_id)
20 VALUES
21 (0, 1);
22
23 -- create a dm
24 > Run | New Tab | Copy
25 INSERT INTO chats
```

Result x

Search results

Cost: 7ms

INSERT INTO files (file_name, contents) VALUES ('testfile', 'image data')

AffectedRows : 1

```
13
14 -- insert a contact
15 > Run | New Tab
16 INSERT INTO contacts
17 (user_id, contact_user_id)
18 VALUES
19 (1, 2); 4ms
20
21 -- create a dm
22 > Run | New Tab | Copy
23 INSERT INTO chats
```

Result x

Search results


Cost: 4ms

INSERT INTO contacts (user_id, contact_user_id) VALUES (1, 2)

AffectedRows : 1

```
19 -- create a dm
20 > Run | New Tab
21 BEGIN;
22 > Run | New Tab | Copy
23 INSERT INTO chats
24   (primary_owner, secondary_owner, sending_privilage, track_views, max_members)
25 SELECT
26   1, 2, 0, FALSE, 2
27 WHERE NOT EXISTS
28   (SELECT 1 FROM chats WHERE (primary_owner=1 AND secondary_owner=2) OR (primary_owner=2 AND secondary_owner=1))
29 RETURNING chat_id;
30 > Run | New Tab | Copy
31 INSERT INTO chat_members
32   (chat_id, member_id, privilage)
33 VALUES
34   (1, 1, 255),
35   (1, 2, 255);
36 > Run | New Tab
37 COMMIT; 7ms
38 -- create group
39 > Run | New Tab | Copy
40 INSERT INTO chats
41   (primary_owner, sending_privilage, track_views, max_members, chat_name)
42 SELECT
```

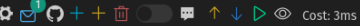
Result x

Search results  Cost: 7ms

COMMIT

```
37 -- create group
38 > Run | New Tab
39 BEGIN;
40 > Run | New Tab | Copy
41 INSERT INTO chats
42   (primary_owner, sending_privilage, track_views, max_members, chat_name)
43 SELECT
44   1, 0, FALSE, 2000, 'hello'
45 WHERE 100>(
46   SELECT COUNT(*) FROM
47     (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
48   LEFT JOIN
49     chat_members
50   ON (t1.chat_id=chat_members.chat_id)
51   WHERE member_id=0
52 )
53 RETURNING chat_id;
54 > Run | New Tab
55 INSERT INTO chat_members
56   (chat_id, member_id, privilage)
57 VALUES
58   (2, 1, 255);
59 > Run | New Tab
60 COMMIT; 3ms
```


Result x

Search results  Cost: 3ms

COMMIT

```
58 -- create a channel max members is null as channels do not have a max
59 > Run | New Tab
60 BEGIN;
61 > Run | New Tab | Copy
62 INSERT INTO chats
63   (primary_owner, sending_privilage, track_views, max_members, chat_name)
64 SELECT
65   1, 128, TRUE, null, 'this is a channel'
66 WHERE 100>(
67   SELECT COUNT(*) FROM
68     (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
69   LEFT JOIN
70     chat_members
71   ON (t1.chat_id=chat_members.chat_id)
72   WHERE member_id=1
73 )
74 RETURNING chat_id;
75 > Run | New Tab
76 INSERT INTO chat_members
77   (chat_id, member_id, privilage)
78 VALUES
79   (3, 1, 255);
80 > Run | New Tab
81 COMMIT; 7ms
82 -- join chat
```

Result x

Search results  Cost: 7ms

COMMIT

```

79 COMMIT;
80
81 -- join chat
82 INSERT INTO chat_members
83 (chat_id, member_id, privilege)
84 SELECT
85 3, 2, 0
86 WHERE
87 100 > (
88 SELECT COUNT(*) FROM
89 (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1
90 LEFT JOIN
91 chat_members
92 ON (t1.chat_id=chat_members.chat_id)
93 WHERE member_id=3
94 ) AND
95 (
96 (SELECT SUM(max_members) FROM chats WHERE chat_id=3)
97 >
98 (SELECT COUNT(*) FROM chat_members WHERE chat_id=3)
99 ); 2ms
100
101 --insert some messages
102 INSERT INTO messages
103 (sender_id, chat_id, message, attachment_id, posted, reply_to)

```

Result

INSERT INTO chat_members (chat_id, member_id, privilege) SELECT 3, 2, 0 WHERE 100 > (SELECT COUNT(*) FROM (SELECT chat_id FROM chats WHERE secondary_owner IS NULL) t1 LEFT JOIN chat_members ON (t1.chat_id=chat_members.chat_id) WHERE member_id=3) AND ((SELECT SUM(max_members) FROM chats WHERE chat_id=3) > (SELECT COUNT(*) FROM chat_members WHERE chat_id=3))

AffectedRows : 0

```

120
121 --insert some messages
122 INSERT INTO messages
123 (sender_id, chat_id, message, attachment_id, posted, reply_to)
124 VALUES
125 (1, 1, 'hello in a dm', null, 1714528509619, null),
126 (2, 1, 'hey!', null, 1714528560684, null),
127 (1, 2, 'howdy in a group!', null, 1714528649805, null),
128 (2, 2, 'cool!', null, 1714528659417, null),
129 (1, 3, 'zoinks in a channel', null, 1714528664131, null),
130 (2, 3, 'channel time!', null, 1714528668494, null); 8ms

```

Result

INSERT INTO messages (sender_id, chat_id, message, attachment_id, posted, reply_to) VALUES (1, 1, 'hello in a dm', null, 1714528509619, null), (2, 1, 'hey!', null, 1714528560684, null), (1, 2, 'howdy in a group!', null, 1714528649805, null), (2, 2, 'cool!', null, 1714528659417, null), (1, 3, 'zoinks in a channel', null, 1714528664131, null), (2, 3, 'channel time!', null, 1714528668494, null)

AffectedRows : 6

q5

a. Retrieve the list of all users

SELECT * FROM users

```

2
3 SELECT * FROM users 3ms
4
5 SELECT * FROM users WHERE availability=1

```

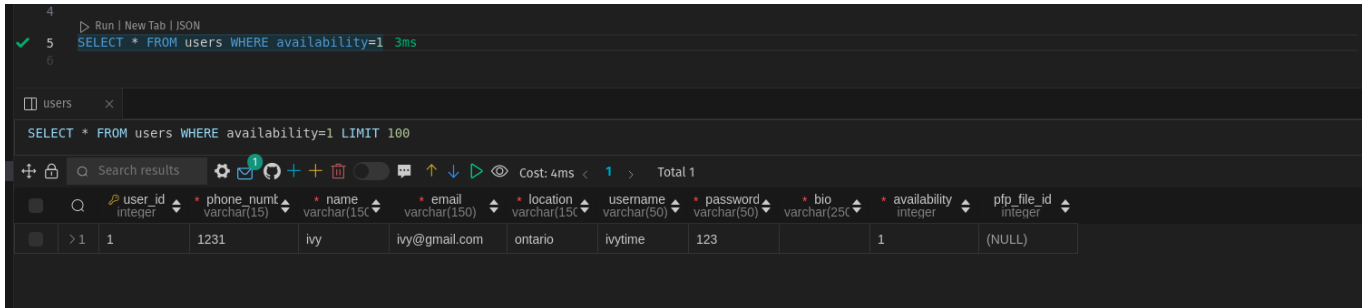
Result

SELECT * FROM users LIMIT 100

	user_id integer	phone_num varchar(15)	name varchar(15)	email varchar(150)	location varchar(15)	username varchar(50)	password varchar(50)	bio varchar(25)	availability integer	pfp_file_id integer
> 1	2	3212	parker	parker@brock.ca	ontario	heygrey	123		0	(NULL)
> 2	1	1231	ivy	ivy@gmail.com	ontario	ivytime	123		1	(NULL)

b. Retrieve the list of all online users.

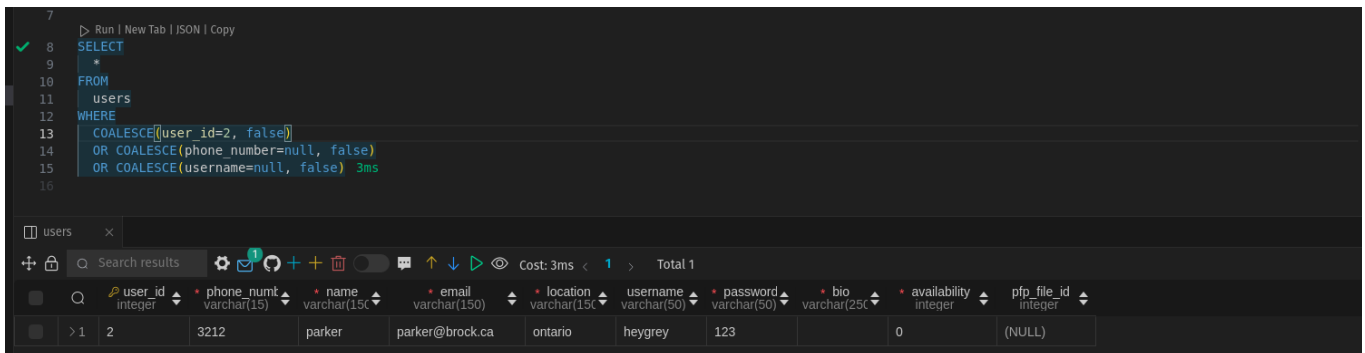
```
SELECT * FROM users WHERE availability=1
```



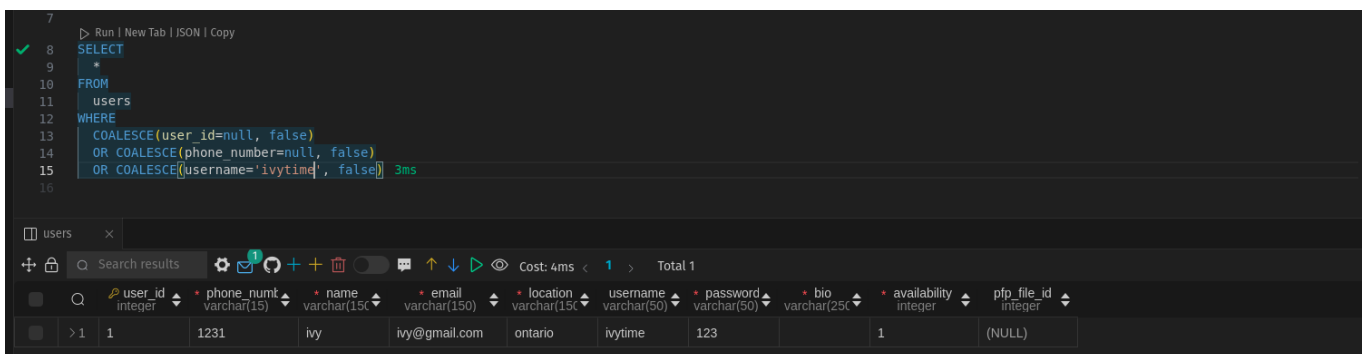
user_id	phone_num	name	email	location	username	password	bio	availability	pfp_file_id
1	1231	ivy	ivy@gmail.com	ontario	ivytime	123	(NULL)	1	(NULL)

c. Given a user (by phone number or unique ID or username), retrieve all information of the user.

```
SELECT
*
FROM
users
WHERE
COALESCE(user_id=:user_id, false)
OR COALESCE(phone_number=:phone_number, false)
OR COALESCE(username=:username, false)
```



user_id	phone_num	name	email	location	username	password	bio	availability	pfp_file_id
2	3212	parker	parker@brock.ca	ontario	heygrey	123	(NULL)	0	(NULL)



user_id	phone_num	name	email	location	username	password	bio	availability	pfp_file_id
1	1231	ivy	ivy@gmail.com	ontario	ivytime	123	(NULL)	1	(NULL)

d. Given a user (by phone number, unique ID or username) retrieve all his/her chats (private chats, normal groups and channels)


```

SELECT
    chat_id
FROM
    chat_members
WHERE member_id=(
    SELECT
        user_id
    FROM
        users
    WHERE
        COALESCE(user_id=:user_id, false)
        OR COALESCE(phone_number=:phone_number, false)
        OR COALESCE(username=:username, false)
)

```

The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:

```

SELECT
  chat_id
FROM
  chat_members
WHERE member_id=(
  SELECT
    user_id
  FROM
    users
  WHERE
    COALESCE(user_id=1, false)
    OR COALESCE(phone_number=null, false)
    OR COALESCE(username=null, false)
)

```

The results pane shows the following data:

chat id
1
2
3

e. For a given chat, retrieve its metadata (chat title, bio, join link (if applicable), etc.)

```

SELECT * FROM chats WHERE chat_id=:chat_id

```

The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:

```

SELECT * FROM chats WHERE chat_id=1

```

The results pane shows the following data:

chat_id	primary_ownership	secondary_ownership	sending_privilege	track_views	max_members	chat_name
1	1	(NULL)	0	false	2000	hello

f. For a given chat, retrieve all its users.

```

SELECT member_id FROM chat_members WHERE chat_id=:chat_id

```

```
33 Run | New Tab | JSON
34 SELECT member_id FROM chat_members WHERE chat_id=:chat_id 3ms
35
```

chat_members x

Search results

member integer

>1 1

g. For a given chat, retrieve all its online users.

```
SELECT
    member_id
FROM
    chat_members
WHERE
    chat_id=:chat_id
    AND 1=(SELECT availability FROM users WHERE
    user_id=chat_members.member_id)
```

```
35 Run | New Tab | JSON | Copy
36 SELECT
37 member_id
38 FROM
39 chat_members
40 WHERE
41 chat_id=:chat_id
42 AND 1=(SELECT availability FROM users WHERE user_id=chat_members.member_id) 4ms
43
```

Result(RO) x

Search results

member id

>1 1

h. For a given chat, retrieve its creator.

```
SELECT
    primary_owner
FROM
    chats
WHERE
    chat_id=:chat_id
```

```
44 SELECT
45     primary_owner
46 FROM
47     chats
48 WHERE
49     chat_id=1 3ms
50
```

chats x

Search results

primary_owne
integer

>1 1

i. For a given chat, retrieve all its admins (including the creator).

```
SELECT
    member_id
FROM
    chat_members
WHERE
    chat_id=:chat_id
    AND privilege>=(SELECT sending_privilage FROM chats WHERE
chat_id=:chat_id)
```

```
52 SELECT
53     member_id
54 FROM
55     chat_members
56 WHERE
57     chat_id=1
58     AND privilege>=(SELECT sending_privilage FROM chats WHERE chat_id=3) 3ms
59
```

Result(RO) x

Search results

member id

>1	1
>2	2

j. For a given chat admin, retrieve his/her permissions.

```
SELECT
    privilege
FROM
    chat_members
WHERE
    chat_id=:chat_id
    AND member_id=:member_id
```

```
60 SELECT
61     privilege
62 FROM
63     chat_members
64 WHERE
65     chat_id=1
66     AND member_id=2 2ms
67
```

chat_members x

Search results

Cost: 3ms < 1 > Total 1

Q * privilege integer

> 1	255
-----	-----

k. For a given chat, retrieve all its message history

```
SELECT
    message_id
FROM
    messages
WHERE
    chat_id=:chat_id
ORDER BY posted DESC
```

```
67
68 SELECT
69     message_id
70 FROM
71     messages
72 WHERE
73     chat_id=:chat_id
74     ORDER BY posted DESC 3ms
75
```

messages x

Search results

Cost: 14ms < 1 > Total 2

Q message integer

> 1	14
> 2	13

l. For a given chat, retrieve its message during a specific date-time

```
SELECT
    message_id
FROM
    messages
WHERE
    chat_id=:chat_id
    AND posted<=:before
    AND posted>=:after
ORDER BY posted DESC
```

```
76
77 SELECT
78   message_id
79 FROM
80   messages
81 WHERE
82   chat_id=1
83   AND posted<=1714528509640
84   AND posted>=1714528509600
85 ORDER BY posted DESC 2ms
86
```

messages x

Search results

message integer

>1 13

Cost: 3ms < 1 > Total 1

m. For a given chat, retrieve all messages posted by a user during a specific date-time range

```
SELECT
  message_id
FROM
  messages
WHERE
  chat_id=:chat_id
  AND sender_id=:user_id
  AND posted<=:before
  AND posted>=:after
ORDER BY posted DESC
```

```
87
88 SELECT
89   message_id
90 FROM
91   messages
92 WHERE
93   chat_id=1
94   AND sender_id=1
95   AND posted<=1714528509640
96   AND posted>=1714528509600
97 ORDER BY posted DESC 3ms
98
```

messages x

Search results

message integer

>1 13

Cost: 3ms < 1 > Total 1

n. For a given chat, retrieve its unread messages.

```
SELECT
  message_id
FROM
  messages
WHERE
  chat_id=:chat_id
  AND posted>(
    SELECT
```

```

        MIN(last_seen)
    FROM
        chat_members
    WHERE
        chat_id=:chat_id
)

```

The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:

```

SELECT
    message_id
FROM
    messages
WHERE
    chat_id=1
    AND posted>
    (
        SELECT
            MIN(last_seen)
        FROM
            chat_members
        WHERE
            chat_id=1
    )

```

The results pane shows two rows of data:

message_id
13
14

o. For a given chat, retrieve the last n (say 100) message.

```

SELECT
    message_id
FROM
    messages
WHERE
    chat_id=:chat_id
ORDER BY posted DESC
LIMIT 100

```

The screenshot shows a SQL IDE with a query editor and a results pane. The query in the editor is:

```

SELECT
    message_id
FROM
    messages
WHERE
    chat_id=1
ORDER BY posted DESC
LIMIT 100

```

The results pane shows two rows of data:

message_id
14
13

p. For a given message ID, retrieve all its information.

```

SELECT * FROM messages WHERE message_id=:message_id

```

```
124
125
126 SELECT * FROM messages WHERE message_id=14 2ms
127
```

message_id	message	reply_to	attachment_id	posted	last_edited	sender_id	chat_id	views	pinned
14	hey!	(NULL)	(NULL)	1714528560684	(NULL)	2	1	(NULL)	false

q6

included with this submission is a rust project that will run as a webserver for this project

you can either run it locally or you can use the online demo we have running at

<http://sc.on.underlying.skynet.tpgc.me:42069/login.html>

to run the project locally you will need rust installed see <https://www.rust-lang.org/tools/install> and then you can use `cargo run` within the projects directory

below is example output of the program:

User Login

[Don't have an account? Create one](#)

Email:

Password:

Login

User Registration Form

[Have an account? Login](#)

Phone Number:

Name:


Email:

Location:

Username:

Password:

- User Settings
- Create Chat
- DMs
- Groups
- Channels



Phone Number:

Name:

Email:

Location:

Username:

Password:

Bio:

Browse...

Drag and drop your profile picture here, or click to select

Update

Logout

Delete Account

User Settings

Create Chat

DMs

Groups

Channels

Create Chat

Direct Message

Group

Channel

test

Create Chat

User Settings

Create Chat

DMs

parker

Groups

test

Channels

parker

parker

4/30/24, 9:19 PM

hello

ivy

4/30/24, 9:20 PM

hi

Type your message...

attachment

User Settings

Create Chat

DMs

parker

Groups

test

Channels

test



ivy 4/30/24, 9:18 PM

hello



ivy 4/30/24, 9:18 PM

:)



Delete Edit Reply

Type your message...

attachment

User Settings

Create Chat

DMs

parker

Groups

test

Channels

parker



parker 4/30/24, 9:19 PM

hello



ivy 4/30/24, 9:20 PM

hi



ivy 4/30/24, 9:21 PM reply -> parker hello

neat



ivy 4/30/24, 9:22 PM reply -> parker hello



Type your message...

attachment