

University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: November 26th, 2023

Group Number: 45

Name	Student #	CS ID	Email Address
Michael Perkins	35844802	mperki02	michael.alexander.perkins@gmail.com
Siddharth Nand	76648070	sidnand	snand233@gmail.com
Saif Karnawi	92839034	sfkrnwi	saifkarnawi2000@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your email address, and then let us assign you to a TA for your project supervisor.) In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

2. A single SQL script that can be used to create all the tables and data in the database. If you are using multiple scripts while developing, ensure you concatenate them and hand in only a SINGLE SQL script.

```
drop table SUPPORTTEAM cascade constraints;
drop table "User" cascade constraints;
drop table ADMIN cascade constraints;
drop table ACCOUNT cascade constraints;
drop table MAILBOX cascade constraints;
drop table CUSTOMMAILBOX cascade constraints;
drop table EMAIL cascade constraints;
drop table ATTACHMENT cascade constraints;
drop table CONTACT cascade constraints;
drop table USERCONTACTS cascade constraints;

DROP SEQUENCE email_sequence;
DROP SEQUENCE mailbox_sequence;
DROP SEQUENCE custom_mailbox_sequence;
DROP SEQUENCE userContact_sequence;

CREATE TABLE SupportTeam (

    teamCode NUMBER,
    country VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,

    PRIMARY KEY (teamCode)

);

CREATE TABLE "User" (

    phoneNumber NUMBER,
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,

    PRIMARY KEY (phoneNumber)

);

CREATE TABLE Admin (

    agentNumber NUMBER,
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,
    worksFor NUMBER NOT NULL,

    PRIMARY KEY (agentNumber),
    FOREIGN KEY (worksFor) REFERENCES SupportTeam(teamCode) ON DELETE CASCADE

);
```

```

);

CREATE TABLE Account (

    username VARCHAR(20),
    dateCreated DATE NOT NULL,
    ownersPhoneNumber NUMBER NOT NULL,
    adminsNumber NUMBER NOT NULL,

    PRIMARY KEY (username),
    FOREIGN KEY (ownersPhoneNumber) REFERENCES "User" (phoneNumber) ON DELETE CASCADE,
    FOREIGN KEY (adminsNumber) REFERENCES Admin(agentNumber) ON DELETE CASCADE

);

CREATE TABLE Mailbox (

    mailboxID NUMBER,
    ownersUsername VARCHAR(50) NOT NULL,

    PRIMARY KEY (mailboxID),
    FOREIGN KEY (ownersUsername) REFERENCES Account(username) ON DELETE CASCADE

);

CREATE TABLE CustomMailbox (

    mailboxID NUMBER,
    ownersUsername VARCHAR(50) NOT NULL,
    dateCreated DATE NOT NULL,
    customLabel VARCHAR(20) NOT NULL,

    PRIMARY KEY (mailboxID),
    FOREIGN KEY (ownersUsername) REFERENCES Account(username) ON DELETE CASCADE

);

CREATE TABLE Email (

    emailID NUMBER,
    sender VARCHAR(20) NOT NULL,
    recipient VARCHAR(20) NOT NULL,
    body VARCHAR(1000) NOT NULL,
    mailboxID NUMBER NOT NULL,

    PRIMARY KEY (emailID),
    FOREIGN KEY (sender) REFERENCES Account(username) ON DELETE CASCADE,
    FOREIGN KEY (recipient) REFERENCES Account(username) ON DELETE CASCADE,
    FOREIGN KEY (mailboxID) REFERENCES Mailbox(mailboxID) ON DELETE CASCADE

);

```

```

CREATE TABLE Attachment (

    fileName VARCHAR(50),
    emailID NUMBER NOT NULL,
    fileType VARCHAR(5) NOT NULL,

    PRIMARY KEY (fileName, emailID),
    FOREIGN KEY (emailID) REFERENCES Email(emailID) ON DELETE CASCADE

);

CREATE TABLE Contact (

    emailAddress VARCHAR(50),
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,

    PRIMARY KEY (emailAddress)

);

CREATE TABLE UserContacts (

    userContactID NUMBER,
    usersPhoneNumber NUMBER NOT NULL,
    contactsEmailAddress VARCHAR(50) NOT NULL,

    PRIMARY KEY (userContactID),
    FOREIGN KEY (usersPhoneNumber) REFERENCES "User" (phoneNumber) ON DELETE CASCADE

);

CREATE SEQUENCE email_sequence
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

CREATE SEQUENCE mailbox_sequence
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

CREATE SEQUENCE custom_mailbox_sequence
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

```

```

CREATE SEQUENCE userContact_sequence
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;

INSERT INTO SupportTeam VALUES (1, 'Canada', 'Toronto');
INSERT INTO SupportTeam VALUES (2, 'Canada', 'Vancouver');
INSERT INTO SupportTeam VALUES (3, 'Canada', 'Montreal');
INSERT INTO SupportTeam VALUES (4, 'United States', 'New York');
INSERT INTO SupportTeam VALUES (5, 'United States', 'Los Angeles');

INSERT INTO "User" VALUES (7786652266, 'Siddharth', 'Nand');
INSERT INTO "User" VALUES (4432348866, 'Michael', 'Perkins');
INSERT INTO "User" VALUES (3441128899, 'Saif', 'Karnawi');
INSERT INTO "User" VALUES (5557771111, 'Jennifer', 'Smith');
INSERT INTO "User" VALUES (6668882222, 'Lora', 'Hill');

INSERT INTO Admin VALUES (1010, 'Dave', 'Colby', 1);
INSERT INTO Admin VALUES (1011, 'Emma', 'Stome', 1);
INSERT INTO Admin VALUES (1012, 'Jessica', 'Khali', 2);
INSERT INTO Admin VALUES (1013, 'Jone', 'Doe', 3);
INSERT INTO Admin VALUES (1014, 'Jane', 'Doe', 4);

INSERT INTO Account VALUES ('siddharthnand', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 7786652266, 1010);
INSERT INTO Account VALUES ('michaelperkins', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 4432348866, 1011);
INSERT INTO Account VALUES ('saifkarnawi', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 3441128899, 1012);
INSERT INTO Account VALUES ('jennifersmith', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 5557771111, 1013);
INSERT INTO Account VALUES ('lorahill', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 6668882222, 1014);
INSERT INTO Account VALUES ('k_smith', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 6668882222, 1014);

INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'siddharthnand');
INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'michaelperkins');
INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'saifkarnawi');
INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'jennifersmith');
INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'lorahill');
INSERT INTO Mailbox VALUES (mailbox_sequence.nextval, 'k_smith');

INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'siddharthnand', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Important');
INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'michaelperkins', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Work');
INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'saifkarnawi', TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Family');

```

```

INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'jennifersmith',
TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Friends');
INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'lorahill',
TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Important');
INSERT INTO CustomMailbox VALUES (custom_mailbox_sequence.nextval, 'k_smith',
TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Junk');

INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'michaelperkins', 'Hello, how
are you?', 2);
INSERT INTO Email VALUES (email_sequence.nextval, 'michaelperkins', 'siddharthnand', 'I am good,
how are you?', 1);
INSERT INTO Email VALUES (email_sequence.nextval, 'saifkarnawi', 'siddharthnand', 'Can we have a
meeting tomorrow at 5pm?', 1);
INSERT INTO Email VALUES (email_sequence.nextval, 'saifkarnawi', 'michaelperkins', 'Can we delay
it till next week?', 2);
INSERT INTO Email VALUES (email_sequence.nextval, 'michaelperkins', 'saifkarnawi', 'I am going to
be late for dinner tonight', 3);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'saifkarnawi', 'Ok, I will
wait for you', 3);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'saifkarnawi', 'You did not
get the job offer', 3);
INSERT INTO Email VALUES (email_sequence.nextval, 'saifkarnawi', 'siddharthnand', 'Ok, I will
wait for you', 1);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'michaelperkins', 'Email test
message.', 2);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'saifkarnawi', 'Your late for
work!', 3);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'saifkarnawi', 'Nice to meet
you', 3);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'michaelperkins', 'Email test
message 2', 2);
INSERT INTO Email VALUES (email_sequence.nextval, 'siddharthnand', 'michaelperkins', 'Congrats on
your job offer!', 2);
INSERT INTO Email VALUES (email_sequence.nextval, 'saifkarnawi', 'siddharthnand', 'Your late
again', 1);
INSERT INTO Email VALUES (email_sequence.nextval, 'saifkarnawi', 'siddharthnand', 'Where are the
keys!?', 1);

INSERT INTO Contact VALUES ('siddharthnand@gmail.com', 'Siddharth', 'Nand');
INSERT INTO Contact VALUES ('michaelperkins@gmail.com', 'Michael', 'Perkins');
INSERT INTO Contact VALUES ('saifkarnawi@mail.com', 'Saif', 'Karnawi');
INSERT INTO Contact VALUES ('nand_s@yahoo.com', 'Siddharth', 'Nand');
INSERT INTO Contact VALUES ('jennifersmith@mail.com', 'Jennifer', 'Smith');
INSERT INTO Contact VALUES ('admin@ubc.ca', 'UBC', 'Admin');

INSERT INTO Attachment VALUES ('attachment1', 1, 'pdf');
INSERT INTO Attachment VALUES ('attachment2', 2, 'doc');
INSERT INTO Attachment VALUES ('attachment3', 3, 'html');
INSERT INTO Attachment VALUES ('attachment4', 4, 'zip');
INSERT INTO Attachment VALUES ('attachment5', 5, 'pdf');

```

```

INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 7786652266,
'michaelperkins@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 7786652266,
'saifkarnawi@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 7786652266, 'nand_s@gmail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 7786652266, 'admin@ubc.ca');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 7786652266,
'jennifersmith@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 4432348866,
'michaelperkins@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 4432348866,
'saifkarnawi@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 4432348866, 'nand_s@gmail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 4432348866, 'admin@ubc.ca');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899,
'jennifersmith@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899,
'michaelperkins@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899,
'saifkarnawi@mail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899, 'nand_s@gmail.com');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899, 'admin@ubc.ca');
INSERT INTO UserContacts VALUES (userContact_sequence.nextval, 3441128899,
'jennifersmith@mail.com');

```

3.a. A short description of the final project, and what it accomplished.

Our project is a database for an email service. Our database holds all the information that this service requires and would need access to in order to function correctly. It will function similarly to the large email providers such as Gmail.

b. A description of how your final schema differed from the schema you turned in. i. If the final schema differed, explain why. Note that turning in a final schema that's different from what you planned is fine, we just want to know what changed and why.

Our final schema does not differ from the schema we submitted for milestone 2 beyond minor formatting changes.

c. A copy of the schema and screenshots that show what data is present in each relation after the SQL script from item #2 is run. i. Do not use DDL statements for this deliverable. List the relational schemas with the primary key attributes underlined and foreign keys bolded. For example, Student(sid, name, dob). ii. You can complete this deliverable with screenshots from SQL Plus or create a representation of your relational instances

through a spreadsheet program. No matter what you use, be sure to clearly label which relation a given instance refers to.

Email (emailID, mailboxID, sender, recipient, body)

```
SQL> select * from EMAIL;
rows will be truncated
```

EMAILID	SENDER	RECIPIENT	BODY
21	mperk	mperk	dasdasdsad
1	siddharthnand	michaelperkins	Hello, how are you?
2	michaelperkins	siddharthnand	I am good, how are you?
3	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?
4	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?
5	michaelperkins	siddharthnand	Hi, how are you doing today?
6	saifkarnawi	michaelperkins	Can we delay it till next week?
7	michaelperkins	saifkarnawi	I am going to be late for dinner tonight
8	siddharthnand	saifkarnawi	Ok, I will wait for you
9	siddharthnand	saifkarnawi	You did not get the job offer
10	saifkarnawi	siddharthnand	Ok, I will wait for you
11	siddharthnand	michaelperkins	Email test message.
12	siddharthnand	saifkarnawi	Your late for work!
13	siddharthnand	saifkarnawi	Nice to meet you
14	siddharthnand	michaelperkins	Email test message 2
15	siddharthnand	michaelperkins	Congrats on your job offer!
16	saifkarnawi	siddharthnand	Your late again
17	saifkarnawi	siddharthnand	Where are the keys!?
19	mperk2	mperk	dasdasd
20	mperk	mperk	dasdasdsa

Attachment (fileName, emailID, fileType)

```
SQL> select * from ATTACHMENT;
```

FILENAME	EMAILID	FILET
attachment1	1	pdf
attachment2	2	doc
attachment3	3	html
attachment4	4	zip
attachment5	5	pdf

Account (username, ownersPhoneNumber, adminsNumber, dateCreated)

```
SQL> select * from ACCOUNT;
```

USERNAME	DATECREAT	OWNERSPHONENUMBER	ADMINSNUMBER
siddharthnand	01-JAN-19	7786652266	1010
michaelperkins	01-JAN-19	4432348866	1011
saifkarnawi	01-JAN-19	3441128899	1012
jennifersmith	01-JAN-19	5557771111	1013
lorahill	01-JAN-19	6668882222	1014
k_smith	01-JAN-19	6668882222	1014

Admin (agentNumber, firstName, lastName, worksFor)


```
SQL> select * from ADMIN;
```

AGENTNUMBER	FIRSTNAME	LASTNAME	WORKSFOR
1010	Dave	Colby	1
1011	Emma	Stome	1
1012	Jessica	Khali	2
1013	Jone	Doe	3
1014	Jane	Doe	4

User (phoneNumber, firstName, lastName)

```
SQL> select * from "User";
```

PHONENUMBER	FIRSTNAME	LASTNAME
7786652266	Siddharth	Nand
4432348866	Michael	Perkins
3441128899	Saif	Karnawi
5557771111	Jennifer	Smith
6668882222	Lora	Hill

Mailbox (mailboxID, accountUsername)

```
SQL> select * from MAILBOX;
```

MAILBOXID	OWNERSUSERNAME
1	siddharthnand
2	michaelperkins
3	saifkarnawi
4	jennifersmith
5	lorahill
6	k_smith
7	siddharthnand
8	michaelperkins
9	saifkarnawi
10	jennifersmith
11	lorahill

MAILBOXID	OWNERSUSERNAME
12	k_smith

CustomMailbox (mailboxID, accountUsername, dateCreated, customLabel)

```
SQL> select * from CustomMailbox;
```

MAILBOXID	OWNERSUSERNAME	DATECREAT	CUSTOMLABEL
16	mperk	28-NOV-23	Trash
15	mperk2	28-NOV-23	Sports
7	siddharthnand	01-JAN-19	Important
8	michaelperkins	01-JAN-19	Work
9	saifkarnawi	01-JAN-19	Family
10	jennifersmith	01-JAN-19	Friends
11	lorahill	01-JAN-19	Important
12	k_smith	01-JAN-19	Junk
13	mperk	28-NOV-23	Sports

9 rows selected.

Contact (emailAddress, firstName, lastName)

```
SQL> select * from contact;
```

EMAILADDRESS	FIRSTNAME	LASTNAME
siddharthnand@gmail.com	Siddharth	Nand
michaelperkins@gmail.com	Michael	Perkins
saifkarnawi@mail.com	Saif	Karnawi
nand_s@yahoo.com	Siddharth	Nand
jennifersmith@mail.com	Jennifer	Smith
admin@ubc.ca	UBC	Admin

6 rows selected.

SupportTeam (teamCode, country, city)

```
SQL> select * from SUPPORTTEAM;
```

TEAMCODE	COUNTRY	CITY
1	Canada	Toronto
2	Canada	Vancouver
3	Canada	Montreal
4	United States	New York
5	United States	Los Angeles

UserContacts (userContactId, usersPhoneNumber, contactsEmail)

```
SQL> select * from USERCONTACTS;
```

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
1	7786652266	michaelperkins@mail.com
2	7786652266	saifkarnawi@mail.com
3	7786652266	nand_s@gmail.com
4	7786652266	admin@ubc.ca
5	7786652266	jennifersmith@mail.com
6	4432348866	michaelperkins@mail.com
7	4432348866	saifkarnawi@mail.com
8	4432348866	nand_s@gmail.com
9	4432348866	admin@ubc.ca
10	3441128899	jennifersmith@mail.com
11	3441128899	michaelperkins@mail.com

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
12	3441128899	saifkarnawi@mail.com
13	3441128899	nand_s@gmail.com
14	3441128899	admin@ubc.ca
15	3441128899	jennifersmith@mail.com

d. A list of all SQL queries used and where it can be found in the code (i.e., file name and line number(s)). For SQL query requirements, check the rubric listed on Canvas for Milestone 4.

INSERT Queries:

- INSERT INTO "User" VALUES (:phone_number, :first_name, :last_name);
index.php, line 99

DELETE

- DELETE FROM CustomMailbox WHERE mailboxID = :mailboxID
account.php, line 278
account.php, line 516

UPDATE

- UPDATE Email SET Email.mailboxID = :newMailboxID WHERE Email.mailboxID = :currentMailboxID
account.php, line 249

SELECTION

- This next query is dynamically made given user input (search box). This input is split by each word and passed into a function with parameter \$query. \$query is an array with the words from search, the function parses it to make sure it is a valid token. Then it generated an SQL query. This is in mailbox.php, lines 68 to 112 (entire function).

Example:

If the user searches: sender is "Jim" and body contains "meeting" or sender is "Joe"

SQL Query would be: SELECT * FROM Email WHERE mailboxID = :mailboxID AND sender = "Jim" AND body LIKE "%meeting%" OR sender = "Joe"

PROJECTION

- SELECT emailAddress FROM Contact
user.php, line 231

JOIN

- SELECT DISTINCT sender FROM CustomMailbox JOIN Email ON CustomMailbox.mailboxID = Email.mailboxID WHERE customLabel = :label
account.php, line 220

AGGREGATION WITH GROUP BY

```
- SELECT mailboxID, COUNT(*) as EMAIL_COUNT FROM Email WHERE recipient =  
:username GROUP BY mailboxID ORDER BY mailboxID ASC;  
account.php, line 541
```

HAVING

```
- SELECT mailboxID FROM Email WHERE recipient = :username GROUP BY mailboxID  
HAVING COUNT(emailID) > 0  
account.php, line 146
```

NESTED AGGREGATION WITH GROUP BY

```
- SELECT COUNT(*) AS contact_count  
- FROM UserContacts  
- GROUP BY usersPhoneNumber  
- HAVING COUNT(*) >= ALL (  
-     SELECT COUNT(*) AS cnt  
-     FROM UserContacts  
-     GROUP BY usersPhoneNumber )  
user.php, lines 185 - 192
```

DIVISION

```
- SELECT U.phoneNumber, U.firstName, U.lastName FROM "User" U  
- WHERE NOT EXISTS (  
-     SELECT C.emailAddress FROM Contact C  
-     WHERE NOT EXISTS (  
-         SELECT UC.usersPhoneNumber FROM UserContacts UC  
-         WHERE UC.contactsEmailAddress = C.emailAddress AND  
-             UC.usersPhoneNumber = U.phoneNumber ) )  
user.php, lines 150 - 160
```

e. Screenshots demonstrating the functionality of each query using the GUI. We want to see a before/during/after progression of events. For example, the before screenshot would be what data is in the table before you run the query, the during screenshot(s) is how the query is triggered using the GUI, and the after screenshot is what data is in your

table afterwards. Please label each set of screenshots with the name of the query it is meant to address (e.g., “Insert Operation”).

INSERT

```
- INSERT INTO "User" VALUES (:phone_number, :first_name, :last_name);
```

The user must sign up to use our service. Below is the signup page:

Signup

Signup

Below is when the user has inputted their name and number:

Signup

Rob

Jones

8897786554

Signup

Before clicking “Signup”, the “User” table does not have this user:

```
SQL> select * from "User";
```

PHONENUMBER	FIRSTNAME	LASTNAME
7786652266	Siddharth	Nand
4432348866	Michael	Perkins
3441128899	Saif	Karnawi
5557771111	Jennifer	Smith
6668882222	Lora	Hill

After clicking “Signup”, the user is redirected to their page:

Email Service

Welcome to your home page, Rob Jones

Select An Account

Add An Account

Delete Account

Login

Contacts Tab

View Contacts

Add A Contact

Delete A Contact

View Average Contacts Per User

Number of Users Who Have All Contacts Added

When the user clicks “Signup”, the query runs. You can see that this user is in the “User” table after clicking “Signup”:

```
SQL> select * from "User";
```

PHONENUMBER	FIRSTNAME	LASTNAME
7786652266	Siddharth	Nand
4432348866	Michael	Perkins
3441128899	Saif	Karnawi
5557771111	Jennifer	Smith
6668882222	Lora	Hill
8897786554	Rob	Jones

DELETE

- DELETE FROM CustomMailbox WHERE mailboxID = :mailboxID

To delete a CustomMailbox, the user can press the button labeled “Delete Mailbox”, and then select a mailbox to delete (note that they cannot delete the ‘General’ Mailbox”).

Email Service [Check Mailbox Status](#) [Check Admin](#)

Account: michaelperkins
[View Mailboxes](#) [Create Mailbox](#) [Delete Mailbox](#) [Move Emails](#) [Back](#) [Logout](#)

Delete a Custom Mailbox

Work [Delete Mailbox](#)

Before pressing “Delete Mailbox”, this is the CustomMailbox table.

```
SQL> SELECT * FROM CustomMailbox;
```

MAILBOXID	OWNERSUSERNAME	DATECREAT	CUSTOMLABEL
7	siddharthnand	01-JAN-19	Important
8	michaelperkins	01-JAN-19	Work
9	saifkarnawi	01-JAN-19	Family
10	jennifersmith	01-JAN-19	Friends
11	lorahill	01-JAN-19	Important
12	k_smith	01-JAN-19	Junk

After pressing “Delete Mailbox”, this is what the GUI looks like:

Email Service [Check Mailbox Status](#) [Check Admin](#)

Account: michaelperkins
[View Mailboxes](#) [Create Mailbox](#) [Delete Mailbox](#) [Move Emails](#) [Back](#) [Logout](#)

Delete a Custom Mailbox

Success! Mailbox has been deleted.
General [Delete Mailbox](#)

And this is what the new CustomMailbox table looks like:

```
SQL> SELECT * FROM CustomMailbox;
```

MAILBOXID	OWNERSUSERNAME	DATECREAT	CUSTOMLABEL
7	siddharthnand	01-JAN-19	Important
9	saifkarnawi	01-JAN-19	Family
10	jennifersmith	01-JAN-19	Friends
11	lorahill	01-JAN-19	Important
12	k_smith	01-JAN-19	Junk

UPDATE

```
- UPDATE Email SET Email.mailboxID = :newMailboxID WHERE Email.mailboxID = :currentMailboxID
```

The user has the option of moving emails between inboxes using the button labeled “Move Emails”, and then selecting the mailboxes to move emails to and from.

Email Service

Check Mailbox StatusCheck Admin

Account: michaelperkins

View MailboxesCreate MailboxDelete MailboxMove EmailsBackLogout

Move Emails

Move All Emails FromGeneraltoWorkMove

Before pressing Move, this is what the MailboxIDs of the current user's emails are:

```
SQL> SELECT mailboxID FROM Email WHERE recipient = 'michaelperkins';
```

MAILBOXID
8
8
8
8
8
2
2

After pressing Move, this is what the GUI looks like:

Email Service

Check Mailbox StatusCheck Admin

Account: michaelperkins

View MailboxesCreate MailboxDelete MailboxMove EmailsBackLogout

Move Emails

Successfully Moved Emails!

Move All Emails FromGeneraltoGeneralMove

And this is what the updated list of Email MailboxIDs look like:

```
SQL> SELECT mailboxID FROM Email WHERE recipient = 'michaelperkins';
```

MAILBOXID

8
8
8
8
8
8
8

SELECTION

This query allows the user to select from a table using any combination of and/or statements. To do this, we have implemented a search feature where the user can search through their email senders and/or message bodies.

When the user opens their mailbox, they will see a search box:

Search Clear

sender body contains or and

Sender	Body	Attachment
michaelperkins	I am good, how are you?	attachment2.doc
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment3.html
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment4.zip
michaelperkins	Hi, how are you doing today?	attachment5.pdf
saifkarnawi	Ok, I will wait for you	
saifkarnawi	Your late again	
saifkarnawi	Where are the keys!?	
saifkarnawi	awdawdwadaw	Assignment_6.ipynb

The user can click on the green and blue buttons to create a search query. For example, the user can use the below query:

Search Clear

sender body contains or and

Sender	Body	Attachment
michaelperkins	I am good, how are you?	attachment2.doc
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment3.html
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment4.zip
michaelperkins	Hi, how are you doing today?	attachment5.pdf
saifkarnawi	Ok, I will wait for you	
saifkarnawi	Your late again	
saifkarnawi	Where are the keys!?	
saifkarnawi	awdawdwadaw	Assignment_6.ipynb

When the user hits search, it converts the user's query to SQL code: `SELECT * FROM Email WHERE mailboxID = :mailboxID AND sender = 'saifkarnawi' AND body LIKE '%?%'`. For this mailbox, mailboxID is 1. When using this SQL query from SQLPlus, the return table is:

```
SQL> SELECT * FROM Email WHERE mailboxID = 1 AND sender = 'saifkarnawi' AND body LIKE '%?%';
rows will be truncated
```

EMAILID	SENDER	RECIPIENT	BODY
3	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?
4	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?
17	saifkarnawi	siddharthnand	Where are the keys!?

When we hit the “Search” button, we get the same values for recipient and body columns:

Search Clear

sender
body contains
or
and

Sender	Body	Attachment
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment3.html
saifkarnawi	Can we have a meeting tomorrow at 5pm?	attachment4.zip
saifkarnawi	Where are the keys!?	

PROJECTION

- `SELECT emailAddress FROM Contact`
user.php, line 231

This query does not result in any data changes. Here is the table before and after the query is run:

```
SQL> select * from contact;
```

EMAILADDRESS	FIRSTNAME	LASTNAME
siddharthnand@gmail.com	Siddharth	Nand
michaelperkins@gmail.com	Michael	Perkins
saifkarnawi@mail.com	Saif	Karnawi
nand_s@yahoo.com	Siddharth	Nand
jennifersmith@mail.com	Jennifer	Smith
admin@ubc.ca	UBC	Admin

6 rows selected.

In the GUI, this query is triggered from the “View Possible Contacts to Add” button. After it is clicked, it shows the contacts below it.

Email Service

Welcome to your home page, Siddharth Nand

Select An Account

Add An Account

Delete Account

Logout

Contacts Tab

View Contacts

Add A Contact

Delete A Contact

Add Contact

Contact Email

Add Contact

View Possible Contacts to Add

admin@ubc.ca
jennifersmith@mail.com
michaelperkins@gmail.com
nand_s@yahoo.com
saifkarnawi@mail.com
siddharthnand@gmail.com

View the Amount of the Most Contacts A User Has

CMail

JOIN

- **SELECT DISTINCT sender FROM CustomMailbox JOIN Email ON CustomMailbox.mailboxID = Email.mailboxID WHERE customLabel = :label**
account.php, line 220

To activate this query, the “List Correspondents in Mailbox” button is pressed on the account.php page.

Email Service

Check Mailbox Status

Check Admin

Account: michaelperkins

View Mailboxes

Create Mailbox

Delete Mailbox

Move Emails

Back

Logout

Select Mailbox

General

Select Mailbox

Get Email Count

List Correspondents in Mailbox

This is the list of emails in this mailbox:

```
SQL> SELECT * FROM Email WHERE mailboxID = 2;
rows will be truncated
```

EMAILID	SENDER	RECIPIENT	BODY
1	siddharthnand	michaelperkins	Hello, how are you?
6	saifkarnawi	michaelperkins	Can we delay it till next week?
11	siddharthnand	michaelperkins	Email test message.
14	siddharthnand	michaelperkins	Email test message 2
15	siddharthnand	michaelperkins	Congrats on your job offer!

After the button is pressed, the following pop up appears.

Check Mailbox Status

Check Admin

Email Service

Account: michaelperkins

View Mailboxes

Create Mailbox

Delete Mailbox

Move Emails

Back

Logout

Select Mailbox

General

Count

List Correspondents in Mailbox

List of correspondents in General:

siddharthnand

saifkarnawi

AGGREGATION WITH GROUP BY

```
- SELECT mailboxID, COUNT(*) as EMAIL_COUNT FROM Email WHERE recipient = :username GROUP BY mailboxID ORDER BY mailboxID ASC;
```

This query counts the number of emails in a mailbox. On the user's account page, they have the ability to select a mailbox from the dropdown and for each mailbox, get the email count.

Account: siddharthnand

View Mailboxes

Create Mailbox

Delete Mailbox

Move Emails

Back

Logout

Select Mailbox

General

Select Mailbox

Get Email Count

List Correspondents in Mailbox

Account: siddharthnand

View Mailboxes
Create Mailbox
Delete Mailbox
Move Emails

BackLogout

Select Mailbox

✓ General
Important

Select Mailbox
Get Email Count
List Correspondents in Mailbox

When the page initially loads, we get the email counts for all mailboxes, then just display the count for any of the user's mailbox.

Before clicking the “Get Email Count” button, we can see that the SQL database has 8 emails in mailboxID 1:

```
SQL> SELECT * FROM Email WHERE recipient = 'siddharthnand';
```

EMAILID	SENDER	RECIPIENT	BODY	MAILBOXID
2	michaelperkins	siddharthnand	I am good, how are you?	
3	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?	1
4	saifkarnawi	siddharthnand	Can we have a meeting tomorrow at 5pm?	1
5	michaelperkins	siddharthnand	Hi, how are you doing today?	1
10	saifkarnawi	siddharthnand	Ok, I will wait for you	1
16	saifkarnawi	siddharthnand	Your late again	1
17	saifkarnawi	siddharthnand	Where are the keys!?	1
21	saifkarnawi	siddharthnand	awdawwadaw	1

8 rows selected.

When the user clicks on the “Get Email Count” button for the “General” mailbox, which is mailboxID 1:

Account: siddharthnand

View Mailboxes
Create Mailbox
Delete Mailbox
Move Emails

BackLogout

Select Mailbox

General

8 emails in General

Select Mailbox
Get Email Count
List Correspondents in Mailbox

Similarly for the “Important” mailbox (mailboxID != 1):

Account: siddharthnand

View Mailboxes
Create Mailbox
Delete Mailbox
Move Emails

Back
Logout

Select Mailbox

Important
0 emails in Important

Select Mailbox
Get Email Count
List Correspondents in Mailbox

Running the query in SQLPlus where username is “siddharthnand” yields the same results as the GUI shows. We can see that mailboxID 1 (General mailbox) has 8 emails, and the other mailbox has 0:

```
SQL> SELECT mailboxID, COUNT(*) as EMAIL_COUNT FROM Email WHERE recipient = 'siddharthnand' GROUP BY mailboxID ORDER BY mailboxID ASC;

MAILBOXID EMAIL_COUNT
-----
1          8

SQL> |
```

HAVING

- SELECT mailboxID FROM Email WHERE recipient = :username GROUP BY mailboxID
HAVING COUNT(emailID) > 0

account.php, line 146

To activate this query, the “Check Mailbox Status” button is pressed on the account.php page.

Email Service

Check Mailbox StatusCheck Admin

Account: michaelperkins

View Mailboxes
Create Mailbox
Delete Mailbox
Move Emails

Back
Logout

Select Mailbox

Work

Select Mailbox
Get Email Count
List Correspondents in Mailbox

The List of MailboxIDs for this user selected from Emails is shown below:

```
SQL> SELECT mailboxID FROM Email WHERE recipient = 'michaelperkins';
```

MAILBOXID
8
8
8
8
8
2
2

The Mailbox and CustomMailbox Tables are listed below:

MAILBOXID	OWNERSUSERNAME
1	siddharthnand
2	michaelperkins
3	saifkarnawi
4	jennifersmith
5	lorahill
6	k_smith
7	siddharthnand
8	michaelperkins
9	saifkarnawi
10	jennifersmith
11	lorahill

MAILBOXID	OWNERSUSERNAME
12	k_smith

MAILBOXID	OWNERSUSERNAME	DATECREAT	CUSTOMLABEL
7	siddharthnand	01-JAN-19	Important
8	michaelperkins	01-JAN-19	Work
9	saifkarnawi	01-JAN-19	Family
10	jennifersmith	01-JAN-19	Friends
11	lorahill	01-JAN-19	Important
12	k_smith	01-JAN-19	Junk

After the button is pressed, this pop up appears:

Email Service

Check Mailbox Status

Check Admin

Account: michaelperkins

View Mailboxes

Create Mailbox

Delete Mailbox

Move Emails

Back

Logout

X

Mailboxes Currently in Use:

Work

General

Which we can see match the MailboxIDs shown above (8 = 'Work', and 2 is the General Mailbox for this user)

NESTED AGGREGATION WITH GROUP BY

```
- SELECT COUNT(*) AS contact_count
- FROM UserContacts
- GROUP BY usersPhoneNumber
- HAVING COUNT(*) >= ALL (
  - SELECT COUNT(*) AS cnt
  - FROM UserContacts
  - GROUP BY usersPhoneNumber )
```

user.php, lines 185 - 192

This query does not result in data changes, so the data before and after the query is as follows:

```
SQL> SELECT * FROM UserContacts;
```

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
1	7786652266	michaelperkins@mail.com
2	7786652266	saifkarnawi@mail.com
3	7786652266	nand_s@gmail.com
4	7786652266	admin@ubc.ca
5	7786652266	jennifersmith@mail.com
6	4432348866	michaelperkins@mail.com
7	4432348866	saifkarnawi@mail.com
8	4432348866	nand_s@gmail.com
9	4432348866	admin@ubc.ca
10	3441128899	jennifersmith@mail.com
11	3441128899	michaelperkins@mail.com

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
12	3441128899	saifkarnawi@mail.com
13	3441128899	nand_s@gmail.com
14	3441128899	admin@ubc.ca
15	3441128899	jennifersmith@mail.com

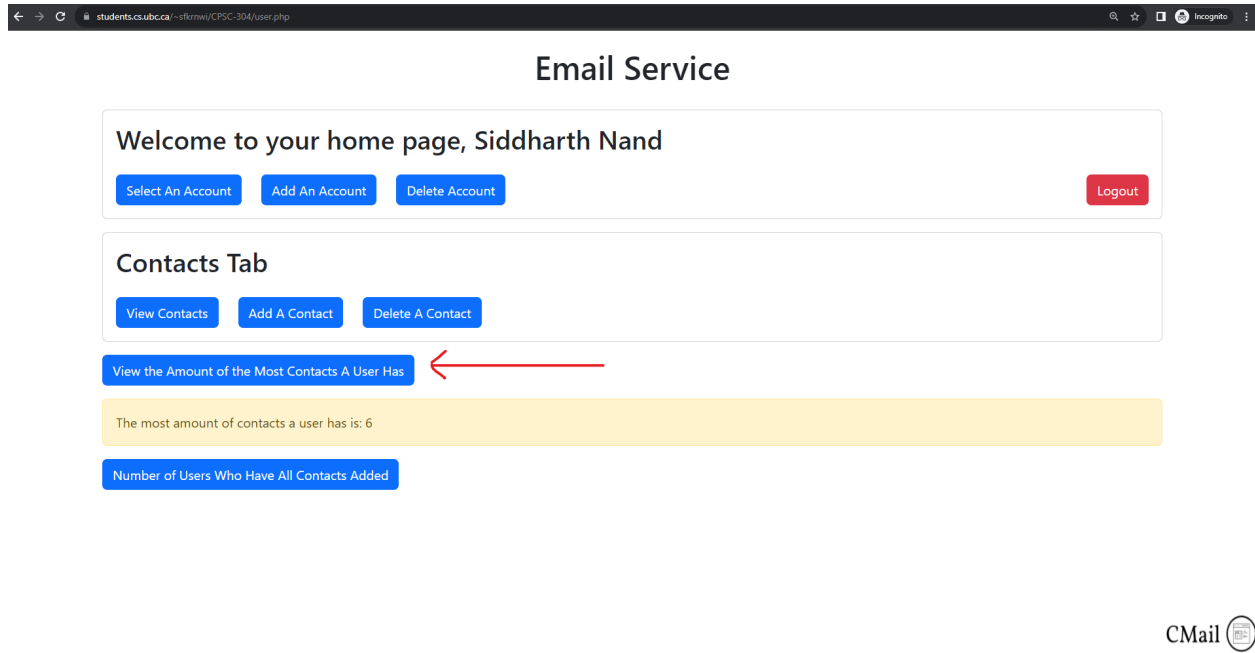
15 rows selected.

And here is the result of the query:

```
SQL> SELECT COUNT(*) AS contact_count FROM UserContacts GROUP BY usersPhoneNumber HAVING COUNT(*) >= ALL (SELECT COUNT(*) AS cnt FROM UserContacts GROUP BY usersPhoneNumber);
```

CONTACT_COUNT
6

The query is triggered from the “View the Amount of the Most Contacts A User Has” button, and it shows the yellow message below it after it is clicked.



DIVISION

```
- SELECT U.phoneNumber, U.firstName, U.lastName FROM \"User\" U
- WHERE NOT EXISTS (
  - SELECT C.emailAddress
  - FROM Contact C
  - WHERE NOT EXISTS (
    - SELECT UC.usersPhoneNumber
    - FROM UserContacts UC
    - WHERE UC.contactsEmailAddress = C.emailAddress AND
      UC.usersPhoneNumber = U.phoneNumber ) )
```

user.php, lines 150 - 160

This query works with multiple tables and it changes none of them in the before and after. Here is the contacts table:


```
SQL> select * from contact;
```

EMAILADDRESS	FIRSTNAME	LASTNAME
siddharthnand@gmail.com	Siddharth	Nand
michaelperkins@gmail.com	Michael	Perkins
saifkarnawi@mail.com	Saif	Karnawi
nand_s@yahoo.com	Siddharth	Nand
jennifersmith@mail.com	Jennifer	Smith
admin@ubc.ca	UBC	Admin

6 rows selected.

And the UserContacts table:

```
SQL> SELECT * FROM UserContacts;
```

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
1	7786652266	michaelperkins@mail.com
2	7786652266	saifkarnawi@mail.com
3	7786652266	nand_s@gmail.com
4	7786652266	admin@ubc.ca
5	7786652266	jennifersmith@mail.com
6	4432348866	michaelperkins@mail.com
7	4432348866	saifkarnawi@mail.com
8	4432348866	nand_s@gmail.com
9	4432348866	admin@ubc.ca
10	3441128899	jennifersmith@mail.com
11	3441128899	michaelperkins@mail.com

USERCONTACTID	USERSPHONENUMBER	CONTACTSEMAILADDRESS
12	3441128899	saifkarnawi@mail.com
13	3441128899	nand_s@gmail.com
14	3441128899	admin@ubc.ca
15	3441128899	jennifersmith@mail.com

15 rows selected.

And here is the result of the query:

```
SQL> SELECT U.phoneNumber, U.firstName, U.lastName FROM "User" U WHERE NOT EXISTS (SELECT C.emailAddress FROM Contact C WHERE NOT EXISTS (SELECT UC.usersPhoneNumber FROM UserContacts UC WHERE UC.contactsEmailAddress = C.emailAddress AND UC.usersPhoneNumber = U.phoneNumber));
```

no rows selected

Finally, the query is triggered from the “Number of Users Who Have All Contacts Added” button, and the output is right below it.

Email Service

Welcome to your home page, Siddharth Nand

[Select An Account](#)[Add An Account](#)[Delete Account](#)[Logout](#)

Contacts Tab

[View Contacts](#)[Add A Contact](#)[Delete A Contact](#)

[View the Amount of the Most Contacts A User Has](#)

[Number of Users Who Have All Contacts Added](#) ←

Currently, there are no users who have every contact added!