

CSCI 331 (DB design)

```
Use Email;
create table Email_account (
Email_ID varchar(50) not null primary key,
Owner_firstname varchar(20),
Owner_lastname varchar(20),
Email_opened date,
Email_closed date,
Email_password char(15),
Email_passwordex date,
Email_ID_status char(10));

INSERT INTO Email_account VALUES ('blee@outlook.com','Bo','lee','2020-02-23'
,'123ab!','2025-02-20','active');

INSERT INTO Email_account VALUES ('sally@yahoo.com','Sally','Smith','2021-02-
22',null,'234cd@','2023-02-20','active');

INSERT INTO Email_account VALUES ('Brown@hotmail.com','John','Brown','2020-02-
21',null,'456ef#','2025-02-20','active');

INSERT INTO Email_account VALUES ('yao@yahoo.com','yao','ming','2021-02-
20',null,'789gh$','2025-02-20','active');

INSERT INTO Email_account VALUES ('ivy@live.com','Blue','Ivy','2022-02-
23',null,'1011ij%','2025-02-20','active');
```

```
Use Email;
create table Email_message (
Email_ID_from varchar(50),
Email_date date,
Email_subject varchar(100),
Email_msgbody varchar(3000));
INSERT INTO Email_message VALUES ('blee@outlook.com' , '2021-12-23' , 'Text program' , I
wanted to welcome you to our text program. Our newest way to reach customers. We look
forward to chatting with you more soon!');

INSERT INTO Email_message VALUES ('blee@outlook.com' , '2021-12-27' , 'latest deals and
discounts' , 'Thanks for joining our text club! We'll share our latest deals and discounts right
here, exclusively with you.');

INSERT INTO Email_message VALUES ('blee@outlook.com' , '2022-02-05' , 'Text club' , I
wanted to reach out and personally thank you for signing up to our text club. I look forward
to chatting more soon.);

INSERT INTO Email_message VALUES ('blee@outlook.com' , '2022-02-10' , 'Arctic
Club' , 'welcome to the Arctic Club texting list. Here, we'll share lineups, specials, and
```

events with you each month.');

INSERT INTO Email_message VALUES ('blee@outlook.com','2022-02-11','Real person','wanted to introduce myself, I'm Pete. Just dropping in to let you know I'm a real person, not a robot, and I'm here to answer any of your questions');

INSERT INTO Email_message VALUES ('blee@outlook.com','2022-03-19','Gym membership','I wanted to drop in and see if you had any questions about your new gym membership. I'll be here if you need anything!');

INSERT INTO Email_message VALUES ('blee@outlook.com','2022-04-01','Spectrum team','thank you for sharing your number with the Spectrum team. Please reply with Yes if this is the best way to reach you!');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-04-20','Sales rep','I wanted to introduce myself. My name is Bea and I'll be your dedicated sales rep. Don't hesitate to reach out if you need anything. That's what I'm here for.');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-04-26','Check in','we haven't heard from you in a while. We wanted to check in and see if there was anything we could help you with?');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-05-15','Auto Z.','Chris here from Auto Z. I wanted to reach out and verify that this is still the best number to reach you at?');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-05-19','Text program','I wanted to welcome you to our text program. Our newest way to reach customers. We look forward to chatting with you more soon!');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-05-24','latest deals and discounts','Thanks for joining our text club! We'll share our latest deals and discounts right here, exclusively with you.');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-05-28','Text club','I wanted to reach out and personally thank you for signing up to our text club. I look forward to chatting more soon.');

INSERT INTO Email_message VALUES ('sally@yahoo.com','2022-05-30','Arctic Club','welcome to the Arctic Club texting list. Here ,we'll share lineups, specials, and events with you each month.');

INSERT INTO Email_message VALUES ('Brown@hotmail.com','2022-06-19',' Real person','wanted to introduce myself, I'm Pete. Just dropping in to let you know I'm a real person, not a robot, and I'm here to answer any of your questions');

INSERT INTO Email_message VALUES ('Brown@hotmail.com','2022-06-20','Gym membership','I wanted to drop in and see if you had any questions about your new gym

membership. I'll be here if you need anything!');

INSERT INTO Email_message VALUES ('Brown@hotmail.com' , '2022-06-26', 'Spectrum team', 'thank you for sharing your number with the Spectrum team. Please reply with Yes if this is the best way to reach you!');

INSERT INTO Email_message VALUES ('Brown@hotmail.com' , '2022-07-07', 'Sales rep', 'I wanted to introduce myself. My name is Bea and I'll be your dedicated sales rep. Don't hesitate to reach out if you need anything. That's what I'm here for.');

INSERT INTO Email_message VALUES ('Brown@hotmail.com' , '2022-07-24', 'Check in', 'we haven't heard from you in a while. We wanted to check in and see if there was anything we could help you with?');

INSERT INTO Email_message VALUES ('Brown@hotmail.com' , '2022-07-31', 'Auto Z.', 'Chris here from Auto Z. I wanted to reach out and verify that this is still the best number to reach you at?');

INSERT INTO Email_message VALUES ('Brown@hotmail.com' , '2022-08-30', 'Text program', 'I wanted to welcome you to our text program. Our newest way to reach customers. We look forward to chatting with you more soon!');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-09-05', 'latest deals and discounts', 'Thanks for joining our text club! We'll share our latest deals and discounts right here, exclusively with you.');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-09-13', 'Text club', 'I wanted to reach out and personally thank you for signing up to our text club. I look forward to chatting more soon.');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-11-11', 'Arctic Club', 'welcome to the Arctic Club texting list. Here, we'll share lineups, specials, and events with you each month.');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-12-01', 'Real person', 'wanted to introduce myself, I'm Pete. Just dropping in to let you know I'm a real person, not a robot, and I'm here to answer any of your questions');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-12-05', 'Text program', 'I wanted to drop in and see if you had any questions about your new gym membership. I'll be here if you need anything!');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-12-05', 'latest deals and discounts', 'thank you for sharing your number with the Spectrum team. Please reply with Yes if this is the best way to reach you!');

INSERT INTO Email_message VALUES ('yao@yahoo.com' , '2022-12-05', 'Text club', 'I wanted to introduce myself. My name is Bea and I'll be your dedicated sales rep. Don't

CSCI 331 (DB design)

hesitate to reach out if you need anything. That's what I'm here for.');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Arctic Club','we haven't heard from you in a while. We wanted to check in and see if there was anything we could help you with?');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Real person','Chris here from Auto Z. I wanted to reach out and verify that this is still the best number to reach you at?');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Gym membership','I wanted to welcome you to our text program. Our newest way to reach customers. We look forward to chatting with you more soon!');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Spectrum team','Thanks for joining our text club! We'll share our latest deals and discounts right here, exclusively with you.');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Sales rep','I wanted to reach out and personally thank you for signing up to our text club. I look forward to chatting more soon.');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Check in','welcome to the Arctic Club texting list. Here, we'll share lineups, specials, and events with you each month.');

INSERT INTO Email_message VALUES ('ivy@live.com','2022-12-05','Auto Z.','wanted to introduce myself, I'm Pete. Just dropping in to let you know I'm a real person, not a robot, and I'm here to answer any of your questions');

alter table Email_message add Msg_ID int NOT NULL AUTO_INCREMENT primary key;

Use Email;

```
create table Msg_map(  
Msg_ID int ,  
Email_ID varchar(50),  
Folder_ID Varchar(50)  
);
```

```
INSERT INTO Msg_map VALUES (1,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (2,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (3,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (4,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (5,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (6,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (7,'sally@yahoo.com','Inbox');  
INSERT INTO Msg_map VALUES (8,'Brown@hotmail.com','Inbox');
```

```
INSERT INTO Msg_map VALUES (9,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (10,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (11,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (12,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (13,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (14,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (15,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (16,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (17,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (18,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (19,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (20,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (21,'yao@yahoo.com','Inbox');
INSERT INTO Msg_map VALUES (22,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (23,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (24,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (25,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (26,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (27,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (28,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (29,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (30,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (31,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (32,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (33,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (34,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (35,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (8,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (9,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (10,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (11,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (12,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (13,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (14,'sally@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (15,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (16,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (17,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (18,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (19,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (20,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (21,'Brown@hotmail.com','Sent');
INSERT INTO Msg_map VALUES (22,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (23,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (24,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (25,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (26,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (27,'yao@yahoo.com','Sent');
INSERT INTO Msg_map VALUES (28,'yao@yahoo.com','Sent');
```

CSCI 331 (DB design)

```
INSERT INTO Msg_map VALUES (29,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (30,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (31,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (32,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (33,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (34,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (35,'ivy@live.com','Sent');
INSERT INTO Msg_map VALUES (1,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (2,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (3,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (4,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (5,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (6,'blee@outlook.com','Sent');
INSERT INTO Msg_map VALUES (7,'blee@outlook.com','Sent');

INSERT INTO Msg_map VALUES (1,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (1,'ivy@live.com','Inbox');
INSERT INTO Msg_map VALUES (3,'Brown@hotmail.com','Inbox');
INSERT INTO Msg_map VALUES (15,'blee@outlook.com','Inbox');
INSERT INTO Msg_map VALUES (15,'ivy@live.com','Inbox');

INSERT INTO Msg_map VALUES (35,'yao@yahoo.com','Inbox');
```

```
Create table CC(
Msg_ID int,
cc_email varchar (50)
);
```

```
INSERT INTO CC VALUES (1,'Brown@hotmail.com');
INSERT INTO CC VALUES (1,'ivy@live.com');
INSERT INTO CC VALUES (3,'Brown@hotmail.com');
```

```
Create table BCC(
Msg_ID int,
bcc_email varchar (50)
);
```

```
INSERT INTO BCC VALUES (15,'blee@outlook.com');
INSERT INTO BCC VALUES (15,'ivy@live.com');
INSERT INTO BCC VALUES (35,'yao@yahoo.com');
```

```
Create table Email_attach (
Msg_ID int,
Msg_attach Varchar (40));
```

```
INSERT INTO Email_attach VALUES (1,'book.pdf');
INSERT INTO Email_attach VALUES (1,'Logo.jpg');
INSERT INTO Email_attach VALUES( 34,'Logo.jpg');
INSERT INTO Email_attach VALUES (34, 'book.pdf');
INSERT INTO Email_attach VALUES (35, 'lecture.pdf');
```

```
Create table receiver (
Msg_ID int ,
Email_ID_to Varchar (50));
INSERT INTO receiver VALUES (1,'sally@yahoo.com');
INSERT INTO receiver VALUES (2,'sally@yahoo.com');
INSERT INTO Receiver VALUES (3,'sally@yahoo.com');
INSERT INTO receiver VALUES (4,'sally@yahoo.com');
INSERT INTO receiver VALUES (5,'sally@yahoo.com');
INSERT INTO receiver VALUES (6,'sally@yahoo.com');
INSERT INTO receiver VALUES (7,'sally@yahoo.com');
INSERT INTO receiver VALUES (8,'Brown@hotmail.com');
INSERT INTO receiver VALUES (9,'Brown@hotmail.com');
INSERT INTO receiver VALUES (10,'Brown@hotmail.com');
INSERT INTO receiver VALUES (11,'Brown@hotmail.com');
INSERT INTO receiver VALUES (12,'Brown@hotmail.com');
INSERT INTO receiver VALUES (13,'Brown@hotmail.com');
INSERT INTO receiver VALUES (14,'Brown@hotmail.com');
INSERT INTO receiver VALUES (15,'yao@yahoo.com');
INSERT INTO receiver VALUES (16,'yao@yahoo.com');
INSERT INTO receiver VALUES (17,'yao@yahoo.com');
INSERT INTO receiver VALUES (18,'yao@yahoo.com');
INSERT INTO receiver VALUES (19,'yao@yahoo.com');
INSERT INTO receiver VALUES (20,'yao@yahoo.com');
INSERT INTO receiver VALUES (21,'yao@yahoo.com');
INSERT INTO receiver VALUES (22,'ivy@live.com');
INSERT INTO receiver VALUES (23,'ivy@live.com');
INSERT INTO receiver VALUES (24,'ivy@live.com');
INSERT INTO receiver VALUES (25,'ivy@live.com');
INSERT INTO receiver VALUES (26,'ivy@live.com');
INSERT INTO receiver VALUES (27,'ivy@live.com');
INSERT INTO receiver VALUES (28,'ivy@live.com');
INSERT INTO receiver VALUES (29,'blee@outlook.com');
INSERT INTO receiver VALUES (30,'blee@outlook.com');
INSERT INTO receiver VALUES (31,'blee@outlook.com');
INSERT INTO receiver VALUES (32,'blee@outlook.com');
INSERT INTO receiver VALUES (33,'blee@outlook.com');
INSERT INTO receiver VALUES (34,'blee@outlook.com');
INSERT INTO receiver VALUES (35,'blee@outlook.com');
```

1. Identify the number of messages and total size by email accounts. Display four columns: email account, account owner name, number of messages and total size.

Display the account with the greatest storage requirements first.

use email;

Select

```
Email_account.Email_ID,Email_account.Owner_firstname,Email_account.Owner_lastname,  
count(Msg_Map.Msg_ID) as 'number of messege',SUM(CHAR_LENGTH(  
Email_message.Email_msgbody))as'total size'  
from Email_account, Email_message ,Msg_Map  
Where Email_account.Email_ID= Msg_Map.Email_ID and Email_message.Msg_ID =  
Msg_Map.Msg_id  
group by (Email_ID)  
ORDER BY (SUM(CHAR_LENGTH( Email_message.Email_msgbody))) DESC;
```

CSCI 331 (DB design)

```

use Email;
Select Email_account.Email_ID,Email_account.Owner_firstname,Email_account.Owner_lastname,
count(Msg_Map.Msg_ID) as 'number of message',SUM(CHAR_LENGTH(Email_message.Email_msgbody))as'total size'
from Email_account,Email_message ,Msg_Map
Where Email_account.Email_ID= Msg_Map.Email_ID and Email_message.Msg_ID = Msg_Map.Msg_id
group by (Email_ID)
ORDER BY (SUM(CHAR_LENGTH(Email_message.Email_msgbody))) DESC;

```

Email_ID	Owner_firstname	Owner_lastname	number of message	total size
ivy@live.com	Blue	Ivy	16	1997
Brown@hotmail.com	John	Brown	16	1975
yao@yahoo.com	xiao	ming	15	1811
blue@outlook.com	Bo	lee	15	1639
sally@yahoo.com	Sally	Smith	14	1688

Action Output

Time	Action	Response	Duration / Fetch Time
1341 18:55:01	Select Email_account.Email_ID,Email_account.Owner_firstname,Email_account.Owner_lastname, count(Msg_Map.Msg_ID) as 'nu...	5 row(s) returned	0.0020 sec / 0.00001...

high to low DESC, low to high ASC

- Send a new message to sally@yahoo.com. Identify the SQL operations to create and send the message. In addition, Display the new message in the SENT folder and the new message in Sally's INBOX folder.

```
INSERT INTO Email.Email_message(Email_ID_from,Email_date,Email_subject,Email_msgbody)
VALUES('Brown@hotmail.com','2022-12-04', 'Question 2', 'Im doing question 2.');
```

```
use Email;
INSERT INTO Msg_Map
values (36,'sally@yahoo.com','Inbox');
```

CSCI 331 (DB design)

```
INSERT INTO Msg_Map  
values(36,'Brown@hotmail.com','Sent');
```

```
INSERT INTO receiver  
values(36,'sally@yahoo.com');
```

```
use Email;  
Select Email_account.Email_ID,Email_account.Owner_firstname,Email_account.Owner_lastname,  
Msg_map.Msg_ID,Msg_map.Folder_ID,Email_message.Email_msgbody  
from Email_account, Email_message, Msg_map  
where Email_account.Email_ID = Msg_map.Email_ID and Msg_map.Msg_ID  
=Email_message.Msg_ID and  
Msg_map.Msg_ID = 36;
```

Project 3 user

Administration Schemas

Query 58 | Msg_map | Email_account | Email_message

SCHEMAS Filter objects

1 • use Email;
2 • Select Email_account.Email_ID,Email_account.Owner_firstname,
3 | Email_account.Owner_lastname, Msg_map.Msg_ID, Msg_map.Folder_ID,Email_message.Email_msgbody
4 | from Email_account, Email_message, Msg_map
5 | where Email_account.Email_ID = Msg_map.Email_ID and Msg_map.Msg_ID =Email_message.Msg_ID and
6 | Msg_map.Msg_ID = 36;
7
8

Result Grid | Filter Rows: Search Export:

Email_ID	Owner_firstname	Owner_lastname	Msg_ID	Folder_ID	Email_msgbody
sally@yahoo.com	Sally	Smith	36	Inbox	I'm doing question 2.
Brown@hotmail.com	John	Brown	36	Sent	I'm doing question 2.

Object Info Session

Table: Msg_map

Columns:

Msg_ID	int
Email_ID	varchar(50)
Folder_ID	varchar(50)

Action Output

Time	Action	Response	Duration / Fetch Time
13... 19:08:0/	Select Email_account.Email_ID,Email_account.Owner_firstname,Email_account.Owner_lastname,Msg_map.Msg_ID,Msg_map.Folder_ID,Email_message.Email_msgbody from Email_account, Email_message, Msg_map where Email_account.Email_ID = Msg_map.Email_ID and Msg_map.Msg_ID =Email_message.Msg_ID and Msg_map.Msg_ID = 36;	2 row(s) returned	0.00005 sec / 0.000...

3. Move a message from the INBOX to the Flagged folder for the account sally@yahoo.com. Identify the SQL operations required to implement. In addition, display the INBOX and SENT folder before and after the operation.

```
SELECT * FROM Email.Msg_map  
where Msg_ID =36 and Email_ID ='sally@yahoo.com';
```

The screenshot shows the MySQL Workbench interface. The query editor window contains the following SQL code:

```
1 •  SELECT * FROM Email.Msg_map  
2   where Msg_ID =36 and Email_ID ='sally@yahoo.com';
```

The result grid shows one row of data:

	Msg_ID	Email_ID	Folder_ID
▶	36	sally@yahoo.com	Inbox

Below the result grid, the status bar indicates "Msg_map 2" and "Read Only". The action output pane shows the executed query and its duration:

Action	Time	Response	Duration / Fetch Time
SELECT * FROM Email.Msg_map where Msg_ID =...	13...	1 row(s) returned	0.00038 sec / 0.000...

```
UPDATE Msg_Map SET Folder_ID = 'Flagged' WHERE Msg_ID = 36 and  
Folder_id= 'Inbox';
```

```
SELECT * FROM Email.Msg_map  
where Msg_ID =36 and Email_ID ='sally@yahoo.com';
```

CSCI 331 (DB design)

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Schemas, Query 58, Msg_map, Email_account, Email_message, and another Msg_map tab. Below the tabs is a toolbar with various icons. The main area contains a query editor with the following SQL code:

```

1  SELECT * FROM Email.Msg_map
2  where Msg_ID =36 and Email_ID ='sally@yahoo.com';
3

```

The results grid displays one row of data:

Msg_ID	Email_ID	Folder_ID
36	sally@yahoo.com	Flagged

Below the results grid, the message "Msg_map 3" is displayed. At the bottom of the interface, there is a status bar showing the time (T3/0 19:15:27), the query (SELECT * FROM Email.Msg_map where Msg_ID =...), the number of rows returned (1 row(s) returned), and the duration (0.00055 sec / 0.0000...).

4. Create a SQL view to display all messages in the INBOX folder for sally@yahoo.com. Display the columns Received Date, FROM, TO, Subject. Display the most recent message first.

Select Email_message.Email_date as 'Received date',
 Email_message.Email_ID_from as 'From',
 receiver.Email_ID_to as 'To',
 Email_message.Email_subject as 'Subject' ,
 Msg_map.Folder_ID as 'Folder_Name' # to verify
 from Email_message,receiver, Msg_map
 Where Email_message.Msg_ID= receiver.Msg_ID
 and receiver.Msg_ID = Msg_map.Msg_ID and
 Msg_map.Email_ID= 'sally@yahoo.com'
 and Msg_map.Folder_ID = 'Inbox'

CSCI 331 (DB design)

ORDER BY (Email_message.Email_date) DESC;

```

1 • Select Email_message.Email_date as 'Received date',
2   Email_message.Email_ID_from as 'From',
3   receiver.Email_ID_to as 'To',
4   Email_message.Email_subject as 'Subject',
5   Msg_map.Folder_ID as 'Folder_Name' # to verify
6   from Email_message,receiver, Msg_map
7   Where Email_message.Msg_ID= receiver.Msg_ID
8   and receiver.Msg_ID = Msg_map.Msg_ID and
9   Msg_map.Email_ID= 'sally@yahoo.com'
10  and Msg_map.Folder_ID = 'Inbox'
11  ORDER BY (Email_message.Email_date) DESC;
12

```

Received date	From	To	Subject	Folder_Name
2022-04-01	blee@outlook.com	sally@yahoo.com	Spectrum team	Inbox
2022-03-19	blee@outlook.com	sally@yahoo.com	Gym membership	Inbox
2022-02-11	blee@outlook.com	sally@yahoo.com	Real person	Inbox
2022-02-10	blee@outlook.com	sally@yahoo.com	Arctic Club	Inbox
2022-02-05	blee@outlook.com	sally@yahoo.com	Text club	Inbox
2021-12-27	blee@outlook.com	sally@yahoo.com	latest deals and discounts	Inbox
2021-12-23	blee@outlook.com	sally@yahoo.com	Text program	Inbox

Action Output

Time	Action	Response	Duration / Fetch Time
13/2 19:19:31	Select Email_message.Email_date as 'Received d...	/ row(s) returned	0.0012 sec / 0.00001...

5. Identify all messages with the term Auto_Z in the subject or body. Display the columns Received Date, FROM, TO, Subject. Display the most recent message first.

use Email;

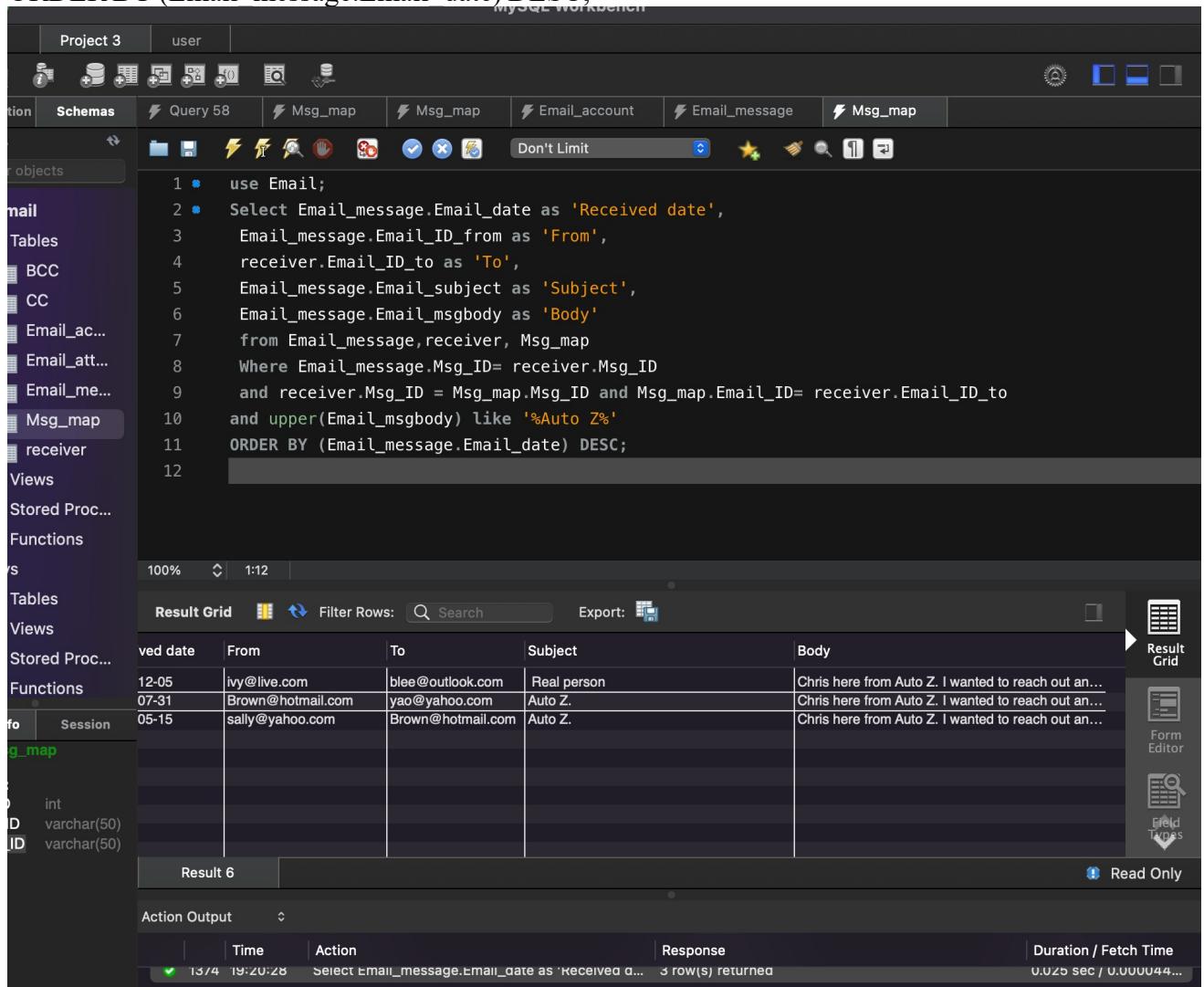
```

Select Email_message.Email_date as 'Received date',
Email_message.Email_ID_from as 'From',
receiver.Email_ID_to as 'To',
Email_message.Email_subject as 'Subject',
Email_message.Email_msgbody as 'Body' #to verify
from Email_message,receiver, Msg_map
Where Email_message.Msg_ID= receiver.Msg_ID

```

CSCI 331 (DB design)

and receiver.Msg_ID = Msg_map.Msg_ID and Msg_map.Email_ID= receiver.Email_ID_to
 and upper(Email_msgbody) like '%Auto Z%'
 ORDER BY (Email_message.Email_date) DESC;



The screenshot shows the MySQL Workbench interface. The top navigation bar includes 'Project 3' and 'user'. Below the toolbar, the schema browser shows 'Email' and its tables: BCC, CC, Email_ac..., Email_att..., Email_me..., Msg_map, receiver, and Views. The 'Msg_map' table is selected. The main area contains a SQL editor with the following query:

```

1 • use Email;
2 • Select Email_message.Email_date as 'Received date',
3     Email_message.Email_ID_from as 'From',
4     receiver.Email_ID_to as 'To',
5     Email_message.Email_subject as 'Subject',
6     Email_message.Email_msgbody as 'Body'
7     from Email_message,receiver, Msg_map
8     Where Email_message.Msg_ID= receiver.Msg_ID
9     and receiver.Msg_ID = Msg_map.Msg_ID and Msg_map.Email_ID= receiver.Email_ID_to
10    and upper(Email_msgbody) like '%Auto Z%'
11    ORDER BY (Email_message.Email_date) DESC;
12

```

The results grid displays three rows of data:

Received date	From	To	Subject	Body
12-05	ivy@live.com	blee@outlook.com	Real person	Chris here from Auto Z. I wanted to reach out an...
07-31	Brown@hotmail.com	yao@yahoo.com	Auto Z.	Chris here from Auto Z. I wanted to reach out an...
05-15	sally@yahoo.com	Brown@hotmail.com	Auto Z.	Chris here from Auto Z. I wanted to reach out an...

At the bottom, the action output shows the query executed and the result: '3 row(s) returned'.

6. Delete one message. Discuss your approach to delete messages and identify the SQL operations required to implement. Display the folder before and after the deletion.

```

SELECT folder_ID,Email_ID, count(folder_ID) as 'Messages in folders' FROM
    Email.Msg_map
where Email_ID ='sally@yahoo.com'
group by (folder_ID);

```

CSCI 331 (DB design)

The screenshot shows the MySQL Workbench interface with the following details:

- Project:** Project 3
- Schemas:** user
- Query Editor:** Query 58 (Msg_map)
- Query:**

```
1 *  SELECT folder_ID,Email_ID, count(folder_ID) as 'Messages in folders' FROM Email.Msg_map
2   where Email_ID ='sally@yahoo.com'
3   group by (folder_ID);
4
```
- Result Grid:**

folder_ID	Email_ID	Messages in fold...
Inbox	sally@yahoo.com	7
Sent	sally@yahoo.com	7
Flagged	sally@yahoo.com	1

Result 13 rows
- Action Output:**

Action	Time	Response	Duration / Fetch Time
1382 19:26:50	SELECT folder_ID,Email_ID, count(folder_ID) as ...	3 row(s) returned	0.00043 sec / 0.000...

```
use Email;
DELETE FROM Msg_map WHERE Folder_ID = 'flagged' and Msg_ID = 36 and
Email_ID = 'sally@yahoo.com';
```

```
SELECT folder_ID,Email_ID, count(folder_ID) as 'Messages in folders' FROM
Email.Msg_map
where Email_ID ='sally@yahoo.com'
group by (folder_ID);
```

CSCI 331 (DB design)

```

1 •  SELECT folder_ID,Email_ID, count(folder_ID) as 'Messages in folders'  FROM Email.Msg_map
2   where Email_ID ='sally@yahoo.com'
3     group by (folder_ID);
4
5
6

```

folder_ID	Email_ID	Messages in fold...
Inbox	sally@yahoo.com	7
Sent	sally@yahoo.com	7

Result 14

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 13...	19:28:55	SELECT folder_ID,Email_ID, count(folder_ID) as '...	2 row(s) returned	0.0017 sec / 0.00001...

7. Identify and move spam messages to the JUNK folder. Identify the SQL operations required to implement.

```

use Email;
Select Email_message.Email_date as 'Received date',
Email_message.Email_ID_from as 'From',
receiver.Email_ID_to as 'To',
Email_message.Email_subject as 'Subject',
Email_message.Email_msgbody as 'Body',
Msg_map.Folder_ID, Msg_map.Msg_ID
from Email_message,receiver, Msg_map
Where Email_message.Msg_ID= receiver.Msg_ID
and receiver.Msg_ID = Msg_map.Msg_ID and Msg_map.Email_ID=
receiver.Email_ID_to
and upper(Email_msgbody) like '%win 50 million%';

```

CSCI 331 (DB design)

The screenshot shows a database management interface with the following details:

- Project 3** is selected in the top navigation bar.
- user** is selected in the schema dropdown.
- Schemas** is selected in the left sidebar.
- Query 58** is the current query tab.
- Code Area:**

```
1  use Email;
2  Select Email_message.Email_date as 'Received date',
3        Email_message.Email_ID_from as 'From',
4        receiver.Email_ID_to as 'To',
5        Email_message.Email_subject as 'Subject',
6        Email_message.Email_msgbody as 'Body',
7        Msg_map.Folder_ID, Msg_map.Msg_ID
8  from Email_message,receiver, Msg_map
9  Where Email_message.Msg_ID= receiver.Msg_ID
10    and receiver.Msg_ID = Msg_map.Msg_ID and Msg_map.Email_ID= receiver.Email_ID_to
11    and upper(Email_msgbody) like '%win 50 million%';
12
```
- Result Grid:**

Received date	From	To	Subject	Body	Folder_ID	Msg_ID
2022-12-05	ivy@live.com	blee@outlook.com	Check in	welcome to the Arctic Club texting list,Win 50 Mi...	Inbox	34

Action Output:

Time	Action	Response	Duration / Fetch Time
13... 19:32:47	Select Email_message.Email_date as 'Received d...	1 row(s) returned	0.00098 sec / 0.000...

```
use email;
UPDATE Msg_map SET Folder_ID = 'junk' WHERE msg_ID = 34 and
Folder_ID = 'Inbox';
```

CSCI 331 (DB design)

MySQL Workbench

Project 3 user

Schemas: Query 58, Msg_map, Email_message, Email_message

Objects: Schemas, Tables, BCC, CC, Email_ac..., Email_att..., Email_me..., Msg_map, receiver, Views, Stored Proc..., Functions

```

1 •  use Email;
2   Select Email_message.Email_date as 'Received date',
3   Email_message.Email_ID_from as 'From',
4   receiver.Email_ID_to as 'To',
5   Email_message.Email_subject as 'Subject',
6   Email_message.Email_msgbody as 'Body',
7   Msg_map.Folder_ID, Msg_map.Msg_ID
8   from Email_message,receiver, Msg_map
9   Where Email_message.Msg_ID= receiver.Msg_ID
10  and receiver.Msg_ID = Msg_map.Msg_ID and Msg_map.Email_ID= receiver.Email_ID_to
11  and upper(Email_msgbody) like '%win 50 million%';
12

```

Result Grid: Filter Rows: Search: Export:

Received date	From	To	Subject	Body	Folder_ID	Msg_ID
2022-12-05	ivy@live.com	blee@outlook.com	Check in	welcome to the Arctic Club texting list,Win 50 Mi...	junk	34

Action Output: Time: 14... 19:36:01 | Action: Select Email_message.Email_date as 'Received d... | Response: 1 row(s) returned | Duration / Fetch Time: 0.0013 sec / 0.00001...

8. Identify accounts that have not received new messages today (2022-12-05). Display the email address.

```

use Email;
SELECT receiver.Email_ID_to As 'Email ID'
FROM Email_message,receiver
where Email_message.Msg_ID= receiver.Msg_ID
and Email_message.Email_date <> '2022-12-05'
group by (Email_ID_to)

```

CSCI 331 (DB design)

The screenshot shows a database management interface with the following details:

Project 3 - user

Schemas: Email, Msg_map, Email_message, Email_account, Email_message, Email_message

Query 58:

```
1 • use Email;
2 • SELECT receiver.Email_ID_to As 'Email ID'
3   FROM Email_message,receiver
4     where Email_message.Msg_ID= receiver.Msg_ID
5       and Email_message.Email_date <> '2022-12-05'
6         group by (Email_ID_to)
7   ;
8
9
```

Result Grid (Email ID):

Email ID
sally@yahoo.com
Brown@hotmail.com
yao@yahoo.com
ivy@live.com

Action Output (Time, Action, Response, Duration / Fetch Time):

Time	Action	Response	Duration / Fetch Time
19:43:10	SELECT receiver.Email_ID_to As 'Email ID' FROM...	4 row(s) returned	0.00063 sec / 0.000...

Result Grid, Form Editor, Field Types buttons are visible on the right side.

9. Utilize security roles to limit users to edit/view rights to their mailbox, but don't have access to other mailboxes. Identify the SQL operations to implement and demonstrate the functionality of the security roles.

#PLEASE CHECK

```
CREATE VIEW `Sallysmith` AS
SELECT Msg_map.folder_ID, Msg_map.Msg_ID,Email_message.Email_msgbody as
'Messages in folders' FROM Msg_map,Email_message
where Msg_map.Email_ID = 'sally@yahoo.com' and Msg_map.Msg_ID =
Email_message.Msg_ID;
```

```
CREATE VIEW `Bolee` AS
SELECT Msg_map.folder_ID, Msg_map.Msg_ID,Email_message.Email_msgbody as
'Messages in folders' FROM Msg_map,Email_message
where Msg_map.Email_ID = 'blee@outlook.com' and Msg_map.Msg_ID =
Email_message.Msg_ID;
```

```
CREATE VIEW `Johnbrown` AS
SELECT Msg_map.folder_ID, Msg_map.Msg_ID,Email_message.Email_msgbody as
'Messages in folders'
FROM Msg_map,Email_message
where Msg_map.Email_ID = 'Brown@hotmail.com' and Msg_map.Msg_ID =
Email_message.Msg_ID;
```

```
CREATE VIEW `yaoming` AS
SELECT Msg_map.folder_ID, Msg_map.Msg_ID,Email_message.Email_msgbody as
'Messages in folders' FROM Msg_map,Email_message
where Msg_map.Email_ID = 'yao@yahoo.com' and Msg_map.Msg_ID =
Email_message.Msg_ID;
```

```
CREATE VIEW `blueivy` AS
SELECT Msg_map.folder_ID, Msg_map.Msg_ID,Email_message.Email_msgbody as
'Messages in folders'
FROM Msg_map,Email_message
Where Msg_map.Email_ID='ivy@live.com' and Msg_map.Msg_ID =
Email_message.Msg_ID;
```

```
CREATE USER 'bolee'@'localhost' IDENTIFIED BY 'bolee';
GRANT SELECT,Update ON Email.bolee TO 'bolee'@'localhost';
```

```
CREATE USER 'sallysmith'@'localhost' IDENTIFIED BY 'sallysmith';
```

CSCI 331 (DB design)

```
GRANT SELECT,Update ON Email.sallysmith TO 'sallysmith'@'localhost';
```

```
CREATE USER 'yaoming'@'localhost' IDENTIFIED BY 'yaoming';
GRANT SELECT,Update ON Email.yaoming TO 'yaoming'@'localhost';
```

```
CREATE USER 'johnbrown'@'localhost' IDENTIFIED BY 'johnbrown';
GRANT SELECT,Update ON Email.johnbrown TO 'johnbrown'@'localhost';
```

```
CREATE USER 'blueivy'@'localhost' IDENTIFIED BY 'blueivy';
GRANT SELECT,Update ON Email.blueivy TO 'blueivy'@'localhost';
```

10. Utilize security roles to allow the administrator to view all mailboxes, but can't change any messages. Identify the SQL operations to implement and demonstrate the functionality of the security roles.

```
CREATE USER 'Admin'@'localhost' IDENTIFIED BY 'Admin';
GRANT select ON Email.* TO 'Admin'@'Localhost' WITH GRANT OPTION;
```

CSCI 331 (DB design)

The screenshot shows the MySQL Workbench interface with the following details:

- Project 3 Admin**: The current project and environment.
- Schemas**: A tree view of database schemas including `email`, `es`, `CC`, `C`, `mail_ac...`, `mail_att...`, `mail_me...`, `sg_map`, `ceiver`, `vs`, `ed Proc...`, and `ctions`.
- Query Editor**: Displays the following SQL code:

```
1 • Use Email;
2 • UPDATE Msg_Map SET Folder_ID = 'Flagged' WHERE Msg_ID = 36 and Folder_id= 'Inbox';
3
4
```
- Session History**: Shows the execution history for the `email` schema, including the following actions:

Action Output	Time	Action	Response	Duration / Fetch Time
100%	23:06:56	SELECT * FROM Email.Email_message	36 row(s) returned	0.0087 sec / 0.00003...
	23:07:38	UPDATE Msg_Map SET Folder_ID = 'Flagged' WH...	Error Code: 1046. No database selected Select the de...	0.00031 sec
	23:07:55	Use Email	0 row(s) affected	0.00023 sec
	23:07:55	UPDATE Msg_Map SET Folder_ID = 'Flagged' WH...	Error Code: 1142. UPDATE command denied to user 'A...' for table 'Msg_Map'	0.00031 sec

CSCI 331 (DB design)

The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```
SELECT * FROM Email.Msg_map;
```

The result grid displays the following data:

Msg_ID	Email_ID	Folder_ID
1	sally@yahoo.com	Inbox
3	sally@yahoo.com	Inbox
4	sally@yahoo.com	Inbox
5	sally@yahoo.com	Inbox
6	sally@yahoo.com	Inbox
7	sally@yahoo.com	Inbox
8	Brown@hotmail.com	Inbox
9	Brown@hotmail.com	Inbox
10	Brown@hotmail.com	Inbox
11	Brown@hotmail.com	Inbox
12	Brown@hotmail.com	Inbox
13	Brown@hotmail.com	Inbox
14	Brown@hotmail.com	Inbox
15	yao@yahoo.com	Inbox
16	yao@yahoo.com	Inbox

Below the grid, the status bar shows:

- Action Output
- Time: 23:09:55
- Action: SELECT * FROM Email.Msg_map
- Response: 77 row(s) returned
- Duration / Fetch Time: 0.00043 sec / 0.000...

11. Identify the SQL operations to implement and demonstrate the functionality of the security roles

#SAME ANSWER

12. In one SQL window, move message 1 to a new folder. Don't commit. In another SQL window, move message 1 to different folder. Don't commit. Explain your results. Resolve the problem. Disable the auto commit flag at the top of the window before performing this operation.

```
SELECT @@autocommit = 0;
```

```
CREATE USER 'ALI'@'localhost' IDENTIFIED BY 'Admin';
GRANT All ON Email.* TO 'Admin'@'localhost' WITH GRANT OPTION;
```

CSCI 331 (DB design)

AND MAIN as root.

In MAIN

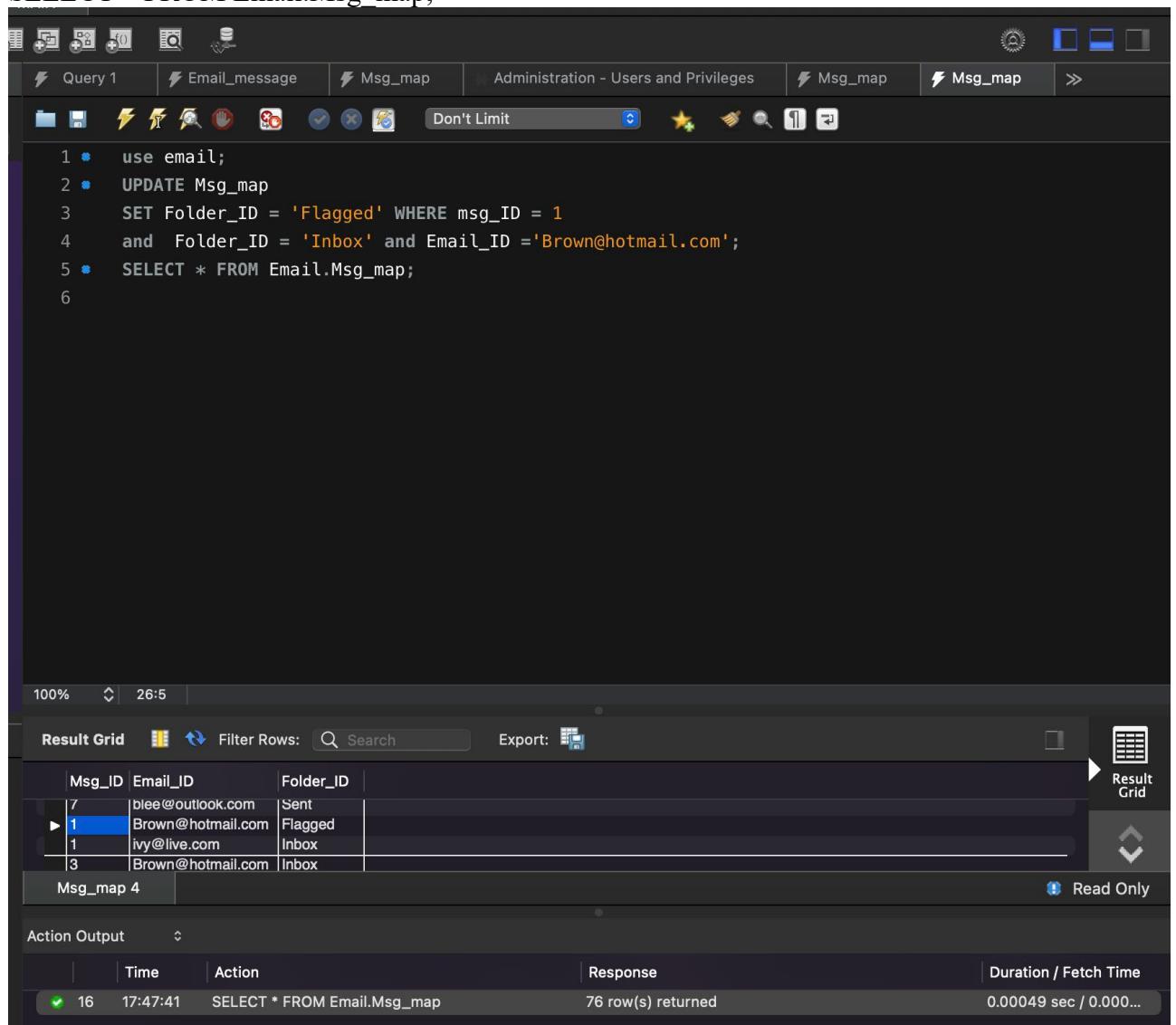
use email;

UPDATE Msg_map

SET Folder_ID = 'Flagged' WHERE msg_ID = 1

and Folder_ID = 'Inbox' and Email_ID ='Brown@hotmail.com';

SELECT * FROM Email.Msg_map;



The screenshot shows the MySQL Workbench interface. The top bar has tabs for "Query 1", "Email_message", "Msg_map", "Administration - Users and Privileges", and two "Msg_map" tabs. Below the tabs is a toolbar with various icons. The main area contains a query editor with the following SQL code:

```
1 • use email;
2 • UPDATE Msg_map
3     SET Folder_ID = 'Flagged' WHERE msg_ID = 1
4     and Folder_ID = 'Inbox' and Email_ID ='Brown@hotmail.com';
5 • SELECT * FROM Email.Msg_map;
6
```

The result grid below shows the data from the Msg_map table:

Msg_ID	Email_ID	Folder_ID
7	blee@outlook.com	Sent
1	Brown@hotmail.com	Flagged
1	ivy@live.com	Inbox
3	Brown@hotmail.com	Inbox

The bottom section shows the "Action Output" with the following details:

Action	Time	Response	Duration / Fetch Time
16	17:47:41	SELECT * FROM Email.Msg_map	76 row(s) returned 0.00049 sec / 0.000...

CSCI 331 (DB design)

In ALI

```
SELECT @@autocommit;
SET AUTOCOMMIT=0;
use email;
UPDATE Msg_map
SET Folder_ID = 'junk' WHERE msg_ID = 1
and Folder_ID = 'Inbox' and Email_ID ='Brown@hotmail.com';
```

The screenshot shows the MySQL Workbench interface with a session titled 'ALI'. In the query editor, the following SQL code is visible:

```
1 #In ALI
2 • use email;
3 • UPDATE Msg_map
4 SET Folder_ID = 'junk' WHERE msg_ID = 1
5 and Folder_ID = 'Inbox' and Email_ID ='Brown@hotmail.com';
6 • SELECT * FROM Email.Msg_map;
7
```

The results pane shows the execution of the UPDATE statement:

Action	Output	Time	Action	Response	Duration / Fetch Time
117	18:06:59	use email		0 row(s) affected	0.000022 sec
118	18:06:59	UPDATE Msg_map SET Folder_ID = 'junk' WHERE...	Running...		?

It says Running and then ALI is **locked out**. Since MAIN is using the row and did not **commit** the new value the row is locked and the table so ALI is locked and also other users. The lock will be released only if MAIN commits the changed table value then the other users will be able to access the table with new value set by MAIN.

13. In one SQL window, delete all messages for one account. Don't commit. In another SQL window, move all messages for the same account to the FRIENDS folder. Don't commit. Explain your results. Resolve the problem. Create a backup of your table before implementing. To create a backup table, enter CREATE TABLE <NEWTABLE> AS SELECT * FROM <ORIGINALTABLE>; COMMIT; Then you can rename a table using the RENAME TABLE commit. Disable the auto commit flag at the top of the window before performing this operation.

```
CREATE TABLE MAP_MSG AS SELECT * FROM Msg_map;  
commit;
```

FROM ALI AND MAIN

```
SELECT @@autocommit=0 ;  
  
#IN MAIN  
Delete from Msg_map  
where Email_ID='sally@yahoo.com';  
SELECT * FROM Email.Msg_map;  
Use Email;
```

CSCI 331 (DB design)

The screenshot shows the MySQL Workbench interface. The top bar has tabs for 'MAIN' and several database connections. Below the tabs is a toolbar with icons for file operations, search, and refresh. The main area contains a query editor with the following code:

```
1 #IN MAIN
2 • SELECT * FROM Email.Msg_map WHERE Email_ID ='sally@yahoo.com';
3 • Use Email;
4 • Delete from Msg_map
5 where Email_ID ='sally@yahoo.com';
6
7
8
```

Below the code is a results grid titled 'Result Grid' with columns 'Msg_ID', 'Email_ID', and 'Folder_ID'. A message 'Msg_map 9' is displayed above the grid. The bottom section shows an 'Action Output' table with the following data:

Action	Time	Response	Duration / Fetch Time
80 18:30:43 Delete from Msg_map where Email_ID ='sally@y...' 14 row(s) affected			0.00080 sec
81 18:30:50 SELECT * FROM Email.Msg_map WHERE Email_ID... 0 row(s) returned			0.00048 sec / 0.000...
82 18:30:50 Use Email 0 row(s) affected			0.00025 sec
83 18:30:50 Delete from Msg_map where Email_ID ='sally@y...' 0 row(s) affected			0.00040 sec

```
SELECT @@autocommit =0;
```

```
SELECT * FROM Email.Msg_map;
UPDATE Msg_map SET Folder_ID = 'Junk'
WHERE Email_ID ='sally@yahoo.com';
```

CSCI 331 (DB design)

MySQL Workbench

ALI MAIN

Schemas: Msg_map, Email_message, Email_message, Msg_map, Email_message, Msg_map, Msg_map, >>

Objects: SELECT * FROM Email.Msg_map;
UPDATE Msg_map SET Folder_ID = 'Junk'
WHERE Email_ID = 'sally@yahoo.com';
;

Result Grid: Msg_ID, Email_ID, Folder_ID

Msg_ID	Email_ID	Folder_ID
3	sally@yahoo.com	Inbox
4	sally@yahoo.com	Inbox
5	sally@yahoo.com	Inbox
6	sally@yahoo.com	Inbox
7	sally@yahoo.com	Inbox
8	Brown@hotmail.com	Inbox
9	Brown@hotmail.com	Inbox
10	Brown@hotmail.com	Inbox

Session message

varchar(50)
date
varchar(100)
varchar(3000)
int AI PK

Action Output

	Time	Action	Response	Duration / Fetch Time
124	18:28:10	SELECT * FROM Email.Msg_map	76 row(s) returned	0.0013 sec / 0.00025...
125	18:28:10	UPDATE Msg_map SET Folder_ID = 'Junk' WHERE Email_ID = 'sally@yahoo.com';	Running...	?

Result Grid, Form Editor

It says Running and then ALI is locked out . Since MAIN is using the row AND DELETED it and did not commit the new value the row is locked and the table so ALI is locked and also other users. The lock will be released only if MAIN **COMMIT** the changed table value or do **ROLLBACK** then the other users (ALI) will be able to access the table with new value set by MAIN.

14. In one SQL window, change the password for account 1. Don't commit. In another SQL window, change the last name of the owner of account 1. Don't commit. Quit both Oracle sessions. Login to Oracle and display all information. Explain your results. Disable the auto commit flag at the top of the windows before performing this operation.

Account 1 is bolee

From MAIN

```
SELECT @@autocommit;
SET AUTOCOMMIT=0;
ALTER USER 'bolee'@'localhost' IDENTIFIED BY 'bopang';
```

From ALI

```
SELECT @@autocommit=0;
RENAME USER 'bolee'@'localhost' TO 'bopang'@'localhost';
```

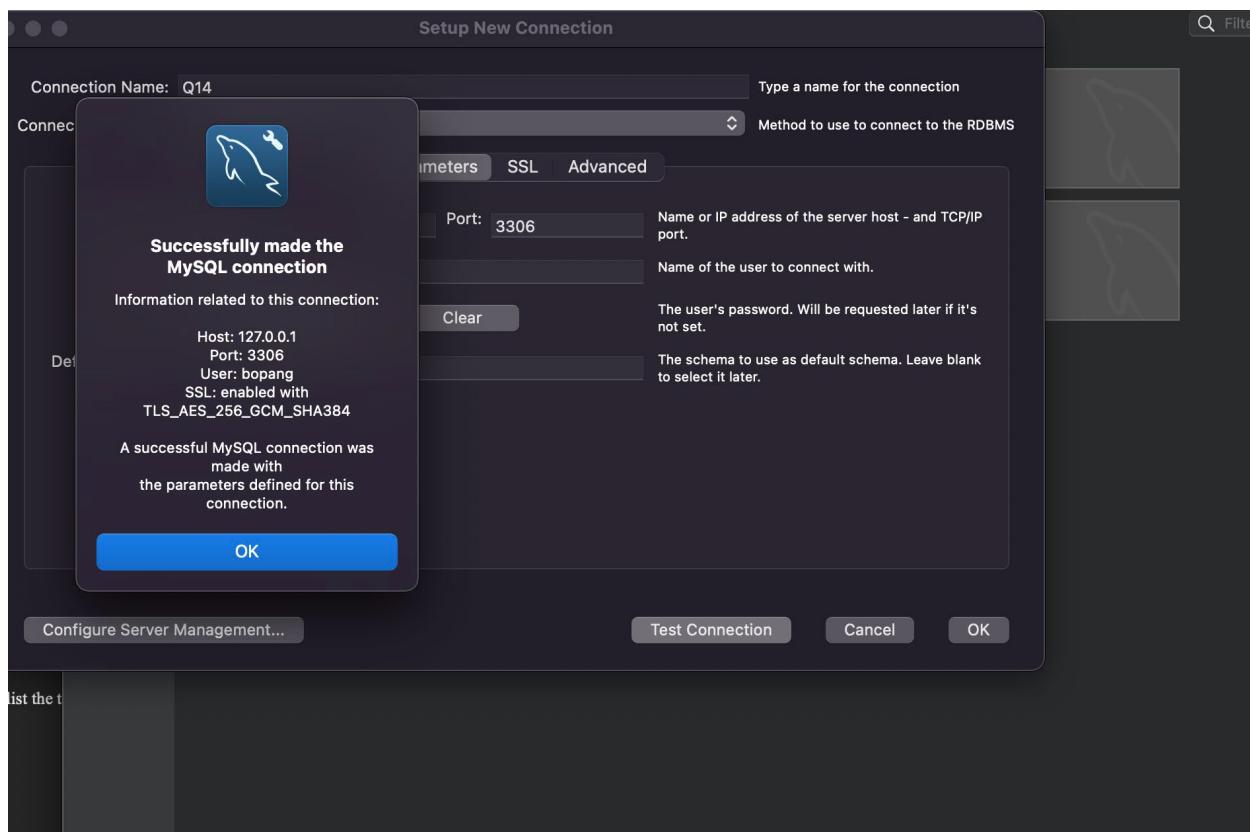
CSCI 331 (DB design)

The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench". Below it is a section titled "MySQL Connections" with a search bar "Filter connections". There are seven connection cards displayed:

- MAIN**: User "root" from host "127.0.0.1:3306".
- ALI**: User "root" from host "127.0.0.1:3306".
- Admin**: User "Admin" from host "127.0.0.1:3306".
- sallysmith**: User "sallysmith" from host "127.0.0.1:3306".
- yaoming**: User "yaoming" from host "127.0.0.1:3306".
- johnbrown**: User "johnbrown" from host "127.0.0.1:3306".
- blueivy**: User "blueivy" from host "127.0.0.1:3306".
- bolee**: User "bolee" from host "127.0.0.1:3306".

bolee still exists as a user but can't log in to bolee anymore. IN users can see another user called **bopang**, lets try connecting to bopang with 2 different passwords

CSCI 331 (DB design)



Logged in successfully to ‘bopang’ with password ‘bopang’

15. Use the SQL DESCRIBE operation to list the table structure for all tables.

CSCI 331 (DB design)

MySQL Workbench

Query 1 | Email_message | Msg_map

1 DESC EMAIL.CC;

Result Grid

Field	Type	Null	Key	Default	Extra
Msg_ID	int	YES		NULL	
cc_email	varchar(50)	YES		NULL	

Result 7

Action Output

Time	Action	Response	Duration / Fetch Time
19:26:56	DESC EMAIL.CC	2 row(s) returned	0.00093 sec / 0.000...

MAIN

Query 1 | Email_message | Msg_map

1 DESC EMAIL.Email_account;

Result Grid

Field	Type	Null	Key	Default	Extra
Email_ID	varchar(50)	NO	PRI	NULL	
Owner_firstname	varchar(20)	YES		NULL	
Owner_lastname	varchar(20)	YES		NULL	
Email_opened	date	YES		NULL	
Email_closed	date	YES		NULL	
Email_password	char(15)	YES		NULL	
Email_passwordex	date	YES		NULL	
Email_ID_status	char(10)	YES		NULL	

Result 8

Action Output

Time	Action	Response	Duration / Fetch Time
19:27:43	DESC EMAIL.Email_account	8 row(s) returned	0.00096 sec / 0.000...

CSCI 331 (DB design)

MySQL Workbench

MAIN

Schemas: EMAIL

Tables: Email_attach

Columns:

Field	Type	Null	Key	Default	Extra
Msg_ID	int	YES		NULL	
Msg_attach	varchar(40)	YES		NULL	

Action Output

Time	Action	Response	Duration / Fetch Time
10 19:28:17	DESC EMAIL.Email_attach	2 row(s) returned	0.000091 sec / 0.000...

MAIN

Schemas: EMAIL

Tables: Email_message

Columns:

Field	Type	Null	Key	Default	Extra
Email_ID_from	varchar(50)	YES		NULL	
Email_date	date	YES		NULL	
Email_subject	varchar(100)	YES		NULL	
Email_msgbody	varchar(3000)	YES		NULL	
Msg_ID	int	NO	PRI	NULL	auto_increment

Action Output

Time	Action	Response	Duration / Fetch Time
11 19:28:50	DESC EMAIL.Email_message	5 row(s) returned	0.000094 sec / 0.000...

CSCI 331 (DB design)

MySQL Workbench

MAIN

Schemas: Query 1 Email_message Msg_map

Objects: DESC EMAIL.MSG_MAP;

Tables: BCC, CC, Email_ac..., Email_att..., Email_me..., Msg_map, receiver

Views:

Stored Proc...:

Functions:

Session: Result Grid, Filter Rows, Search, Export

Result Grid (Table Structure):

Field	Type	Null	Key	Default	Extra
Msg_ID	int	YES		NULL	
Email_ID	varchar(50)	YES		NULL	
Folder_ID	varchar(50)	YES		NULL	

Result 11: Action Output, Time, Action, Response, Duration / Fetch Time

MySQL Workbench

MAIN

Schemas: Query 1 Email_message Msg_map

Filter objects: DESC EMAIL.RECEIVER;

Tables: Email

Columns: receiver

Msg_ID int
Email_ID_t varchar(50)

Result Grid (Table Structure):

Field	Type	Null	Key	Default	Extra
Msg_ID	int	YES		NULL	
Email_ID_to	varchar(50)	YES		NULL	

Result 12: Action Output, Time, Action, Response, Duration / Fetch Time

CSCI 331 (DB design)

MySQL Workbench

Administration Schemas MAIN

Query 1 Email_message Msg_map

DESC EMAIL.BCC;

Result Grid

Field	Type	Null	Key	Default	Extra
Msg_ID	int	YES		NULL	
bcc_email	varchar(50)	YES		NULL	

Result 6 Read Only

Action Output

Time	Action	Response	Duration / Fetch Time
19:26:24	DESC EMAIL.BCC	2 row(s) returned	0.0011 sec / 0.000011...

Object Info Session object selected

Result Grid Form Editor Field Types Query Stats