CS331 HOMEWORK 3

Haiqiang Zou

Student ID: 2717

Due Date: 5/20/2013



1. Create a new message from Sally to Robert. Create the SQL commands to a) create and send the message and b) display all message in Robert’s inbox after Sally sends the message. Display subject, datesent, from, size and priority for only the inbox folder for Robert’s account. The newest message is displayed first.

1. insert into messages

values(mid\_seq.nextval, 'sally@gmail.com','Tomorrow', 'low', '10', sysdate, 'what are you doing?');

insert into mto

values(mtoid\_seq.nextval+1, 50045, 'robert@gmail.com');

insert into storage

values(sid\_seq.nextval, 'sally@gmail.com', 50045, 'Sent');

insert into storage

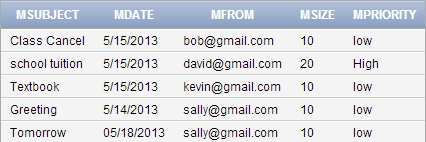
values(sid\_seq.nextval, 'robert@gmail.com', 50045, 'Inbox');

1. select msubject, mdate, mfrom, msize, mpriority

from storage s, messages m

where fname='Inbox' and s.email='robert@gmail.com' and s.mid=m.mid

order by 2 desc;



2. Robert moves a message from the inbox to the friend’s folder. Create the SQL commands to move the message. Display a) Robert’s inbox after the move and b) the friend’s folder after the move. Display subject, datesent, from, size and priority. The newest message is displayed first.

update storage

set fid='20005'

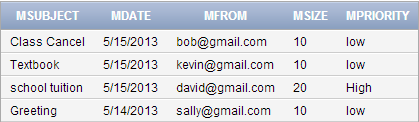
where sid=40001;

1. select Msubject, mdate, mfrom, msize, mpriority

from storage s, messages m, folders f

where s.mid=m.mid and email='robert@gmail.com' and f.fid=s.fid and Fname='Inbox'

order by 2 desc;



1. select Msubject, mdate, mfrom, msize, mpriority

from storage s, messages m, folders f

where s.mid=m.mid and email='robert@gmail.com' and f.fid=s.fid and Fname='Friend'

order by 2 desc;



3. Identify spam when new messages are received. Create the SQL commands to a) identify spam when messages are received and b) move spam messages to the spam folder.

update storage

set fid='20004'

where fid='20001' and mid in (select mid

from messages

where upper(msubject) like '%FREE%' or upper(msubject) like '%BENEFICIARY%' or upper(msubject) like '%CHEAP%' or upper(msubject) like '%DISCOUNT%' or upper(msubject) like '%SAVE UP TO%' or upper(minfo) like '%FREE%' or upper(minfo) like '%BENEFICIARY%' or upper(minfo) like '%CHEAP%' or upper(minfo) like '%DISCOUNT%' or upper(minfo) like '%SAVE UP TO%');

4. The email administrator needs to identify the mailbox size for each account. Display the account, number of messages, size of all messages, earliest message and most recent message. Display one row for each account. The account with the largest mailbox is displayed first.

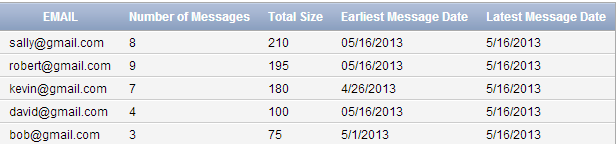
select s.email, count(\*) "Number of Messages", sum(Msize) "Total Size", min(mdate)"Earliest Message Date", max(mdate) "Latest Message Date"

from storage s, messages m

where s.mid=m.mid

group by s.email

order by 3 desc;



5. Sally needs to identify folder information for only her account. Display the account, folder, number of messages, size of messages in this folder, earliest message in this folder and most recent message in this folder. Display one row for each account. The account with the largest mailbox is displayed first.

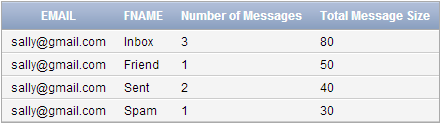
select s.email, fname, count(\*) "Number of Messages", sum(msize) "Total Message Size"

from storage s, messages m

where s.email='sally@gmail.com' and s.mid=m.mid

group by s.email, fname

order by 4 desc;



6. Robert is searching his account for messages sent to Sally recently with the term “I Love You Sally”. Create the SQL to a) perform the search and b) display the responsive messages. Display the subject, datesent, from, size and priority for only the messages responsive to the search. The newest message is displayed first.

select msubject, mdate, mfrom, msize, mpriority, minfo

from messages

where mfrom='robert@gmail.com' and upper(minfo) like '%I LOVE YOU SALLY%'

order by mdate desc;



7. In one SQL window, move message 1 for Sally from the inbox to the friend’s folder. Don’t commit. In another SQL window, move message 1 for Sally (the same message 1) from the inbox to the trash folder. Don’t commit. Explain your results. Resolve the problem.

In the first SQL window I moved a message from Sally’s inbox into her Friend’s folder, after that I tried to move the same message in the second SQL window however it is not allowing me. It seems like there is a write lock in the database because in the second window it says processing but never ends. We can resolve this problem by entering the commit command in the first window then it releases the lock.

8. In one SQL window, delete all messages in Sally’s inbox. Don’t commit. In another SQL window, move Sally’s messages from the inbox to the friend’s 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.

In the first SQL window I delete all messages in Sally’s inbox, In the second SQL window I tried to move Sally’s message into the inbox folder however it is not allowing me. It seems that there is a lock for all Sally’s message because in the second window it says processing but never ends. We can resolve this problem by entering the commit command in the first window then it releases the lock.

9. In one SQL window, delete message 1 from Sally’s inbox. Don’t commit. In another SQL window, forward message 1 in Sally’s inbox to Bob. Don’t commit. Quit both Oracle sessions. Reconnect, Login to SQL and display the inbox for Sally and Bob. Explain your results.

The data turns back into the committed values before I made those two queries because I did not commit after using a change command