1. Create an event with title, description, start date and time, end date and time, location, an optional url, and organizer id. The procedure needs to check whether any other event at the same location has overlap duration with the new event. If so, print a message saying the event conflicts with an existing event. Otherwise, insert the event with input values into event table and print a message with new event ID.

Comment- The procedure checks whether there is any other event at same time and if it overlaps with other events and displays ‘'Sorry there is an event overlapping + same location with the time you provided**’ if not it will be’ your event is registered with ID’**

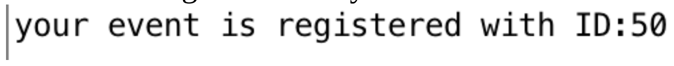
```
create or replace procedure CreateEvent(title in varchar, startdatetime in timestamp,
enddatetime in timestamp, location in varchar, user_url in varchar
,organiser_id in int,error_msg in varchar) IS


l_title varchar(20);
l_locationdesc varchar(20);
l_locationID number;
l_eid number;
l_sdate timestamp;
l_edate timestamp;
l_orgID number;
l_url varchar(100);
new_eid number;
flag number;

BEGIN
l_locationdesc := location;
l_title := title;
l_sdate := startdatetime;
l_edate := enddatetime;
l_orgID := organiser_id;
l_url := user_url;

--flag := 0;
--dbms_output.put_line(location);

select count(*)
into flag
from event e,locations l
where e.lid = l.lid and l.ldesc = location and ((sdate >= l_sdate and sdate <= l_edate)
or (l_sdate >= sdate and l_sdate <= edate));

--select count(*) into new_eid
```

```
--from event;
select lid into l_locationID
from locations l
where l.ldesc = location;


--dbms_output.put_line(flag);

if flag > 0 then
dbms_output.put_line('Sorry there is an event overlapping with the time you provided');
else
insert into event values(eid_seq.nextval,l_title,l_locationdesc,l_sdate,l_edate,
httpuritype.createuri(
l_url),0,l_orgID,l_locationID);

new_eid := eid_seq.currval;
dbms_output.put_line('your event is registered with ID:' ||new_eid);
end if;



exception
  when no_data_found then
      dbms_output.put_line('No data found');



end;
```

Test case 1: Create the event with correct data.

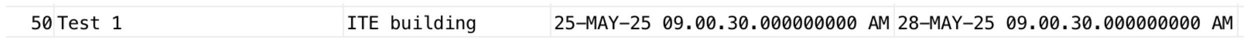
```
set serveroutput on;
declare
begin
 CreateEvent('Test 1', timestamp '2025-05-25 09:00:30.00',
timestamp '2025-05-28 09:00:30.00', 'ITE building', 'http://www.umbc.edu', 1 ,'test error');
end;
```

Get the message from the system as:



```
your event is registered with ID:50
```

In the event table we will see:



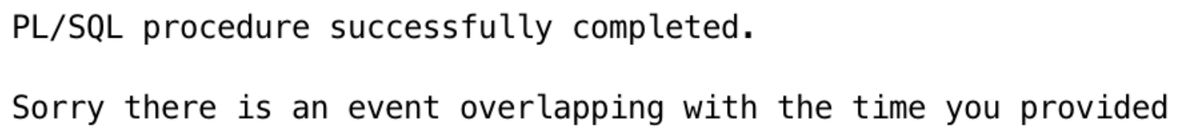
50	Test 1	ITE building	25-MAY-25 09.00.30.000000000 AM	28-MAY-25 09.00.30.000000000 AM

Test case 2:

Overlapping time:

```
begin
CreateEvent('Test 2', timestamp '2025-05-25 09:00:30.00',
timestamp '2025-05-28 09:00:30.00', 'ITE building', 'http://www.umbc.edu', 2 ,'test error');
end;
```

We will get the error message as

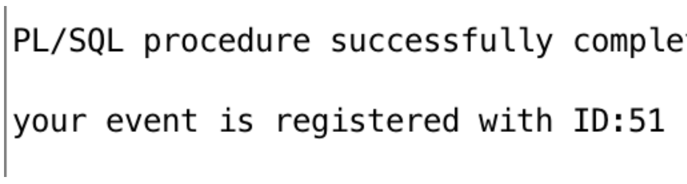


```
PL/SQL procedure successfully completed.

Sorry there is an event overlapping with the time you provided
```

Test case 3:
Overlapping time but in different location.

```
begin

CreateEvent('Test 2', timestamp '2025-05-25 09:00:30.00',timestamp '2025-05-28 09:00:30.00',
'Common', 'http://www.umbc.edu', 2 ,'test error');

end;
```

We will give the message as



```
PL/SQL procedure successfully completed.

your event is registered with ID:51
```

2. List people registered for an event by providing event id. The procedure prints out name of participants, their email addresses, and whether the participant is faculty, staff, or student.

Comment-It prints people with event id and print out whether participants are faculty, staff or students.

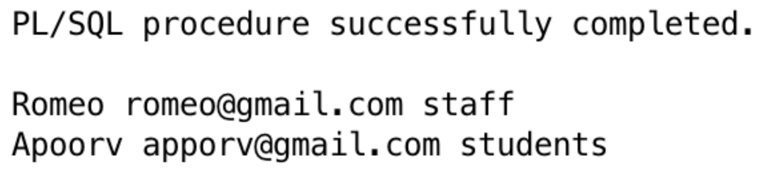
```
Create or replace procedure ParticipantEventList(e_id in integer) IS

username varchar(20);
useremail varchar(20);
ut_desc varchar(20);
event_ID number;

Cursor c1 is select uname,uemail,utdesc
from usertable utb,usertypes uty,participant p
where eid = e_id and utb.userid = p.userid and utb.utid = uty.utid and p.pcflag='0';

begin

select e.eid into event_ID
from event e
where e.eid=e_id;

if event_ID is not null then
open c1;
loop
```

```
fetch c1 into username,useremail,ut_desc;
exit when c1%notfound;
dbms_output.put_line(username||' '||useremail||' '||ut_desc);
end loop;
close c1;
end if;
exception
  when no_data_found then
   dbms_output.put_line('Wrong event ID');
end;
```

Test Case 1: Lists participants for an event after inputting right event ID.

exec ParticipantEventList(2);

After Execution:

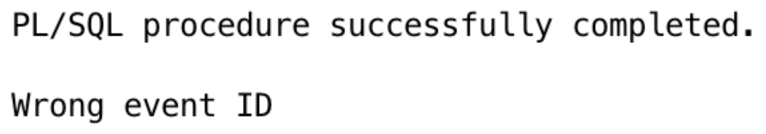


```
PL/SQL procedure successfully completed.

Romeo romeo@gmail.com staff
Apoorv apporv@gmail.com students
```

Test Case 2: User inputs wrong event ID.

exec ParticipantEventList(24);

After execution:



```
PL/SQL procedure successfully completed.

Wrong event ID
```

3. List people on wait list of an event by providing the event id. The procedure prints out names of users on the wait list, their email addresses, and whether they are faculty, staff, or students.

Comment- It Returns average rating of an event and total number of participants and number of people on wait list

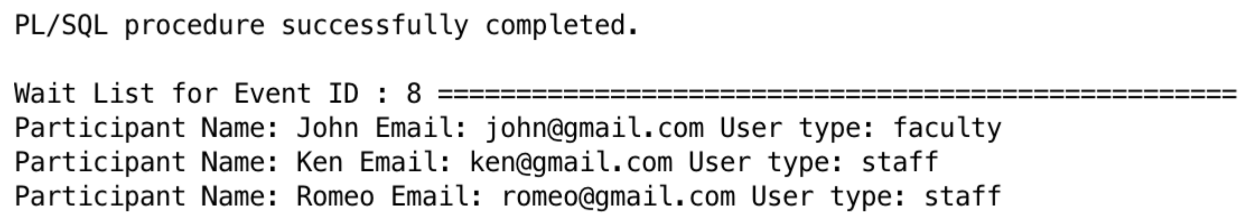
```
create or replace procedure ParticipantEvenWaittList (e_eid in integer) is

cursor c1 is select waitlist.userid, u.uname, u.uemail, usertypes.utdesc
from waitlist, event, usertable u, usertypes
where waitlist.eid = event.eid and waitlist.userid = u.userid and event.eid = e_eid
and waitlist.userid = u.userid and u.utid = usertypes.utid;
```

```
t_c1 c1%rowtype;
t_eventid event.eid%type;
begin

  select eid into t_eventid from event where eid = e_eid and ecflag = '0';
  if t_eventid is not null then
        dbms_output.put_line('Wait List for Event ID : ' || e_eid ||
        ' ===================================================');
        open c1;
        loop
            fetch c1 into t_c1;
            exit when c1%notfound;
            dbms_output.put_line('Participant Name: ' || t_c1.uname
            || ' Email: ' || t_c1.uemail || ' User type: ' || t_c1.utdesc);

        end loop;
        if c1%rowcount = 0 then
        dbms_output.put_line('No Wait list');
        end if;
        close c1;
  end if;

exception
  when no_data_found then
   dbms_output.put_line('Wrong event ID');

end;
```

Test Case 1: Shows the participants wait list after inputting the correct event ID

exec ParticipantEvenWaittList(8);

After execution:

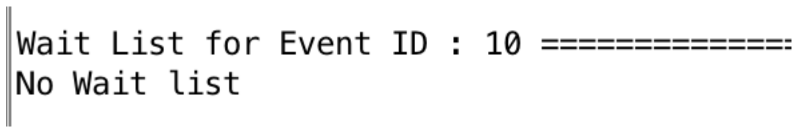


```
PL/SQL procedure successfully completed.

Wait List for Event ID : 8 ===================================================
Participant Name: John Email: john@gmail.com User type: faculty
Participant Name: Ken Email: ken@gmail.com User type: staff
Participant Name: Romeo Email: romeo@gmail.com User type: staff
```

Test Case 2: Show no wait list for event that has no waitlist.

exec ParticipantEventWaitList(10);

We get the message show “No wait list”



```
Wait List for Event ID : 10 ==============
No Wait list
```

Comment-It list people on waitlist, their name ,their email addresses and type of users.

4. Return average rating of an event and total number of participants and number of people on wait list.

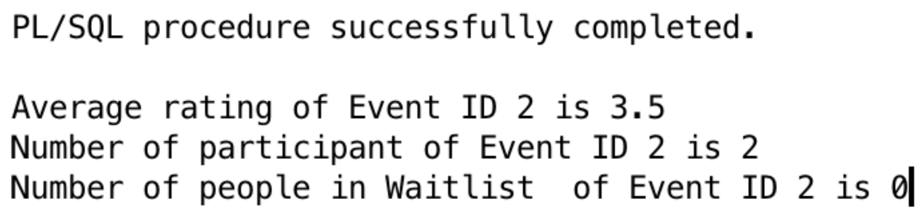
Comment- It Returns average rating of an event and total number of participants and number of people on wait list

```
create or replace procedure average_rating(e_eid in integer) is

avg_rating float;
noofpartici integer;
noofwaitlist integer;
temp integer;

begin
select count(*) into temp from event where eid=e_eid;

if temp <> 0 then
select avg(prating) into avg_rating from participant
where eid = e_eid and pcflag = '0';

dbms_output.put_line('Average rating of Event ID '|| e_eid ||' is ' || avg_rating );

select count(*) into noofpartici from participant
where eid=e_eid and pcflag = '0';

dbms_output.put_line('Number of participant of Event ID '|| e_eid ||' is ' || noofpartici );

select count(*) into noofwaitlist from waitlist
where eid=e_eid;

dbms_output.put_line('Number of people in Waitlist  of Event ID '|| e_eid ||' is ' || noofwaitlist );

else

dbms_output.put_line('Event ID ' || e_eid ||' doesnt exist');

end if;

end;
```

Test case 1: See the rating for general event 2.

exec average_rating(2);

We will get the output as below:

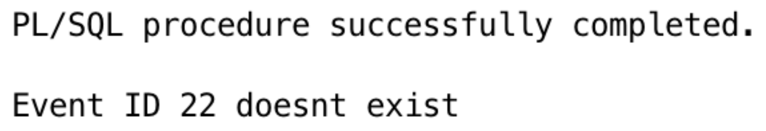


```
PL/SQL procedure successfully completed.

Average rating of Event ID 2 is 3.5
Number of participant of Event ID 2 is 2
Number of people in Waitlist  of Event ID 2 is 0
```

Test case 2: No event exists.

exec average_rating(22);

You will get the error message saying that the event does not exist.



```
PL/SQL procedure successfully completed.

Event ID 22 doesnt exist
```

8. Cancel an event. Please generate a message (by inserting into the message table) for each person who has registered or on wait list saying that the event has been cancelled. You can use a flag in event table to indicate that the event has been cancelled (so you don't need to delete the event and registration records).

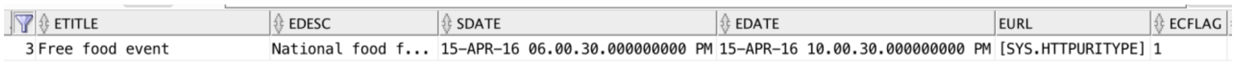
Comment-It cancels the event and generates message for each participant who has registered for event saying event is cancelled and flag generated for cancelled event.

```
create or replace procedure event_cancellation (e_eid in integer) is
cursor c1 is select userid from participant where eid = e_eid and pcflag = '0';
cursor c2 is select userid from waitlist where eid = e_eid;
r_userid participant.userid%type;
w_userid waitlist.userid%type;

final_message varchar(50);
temp integer;
begin
select count(*) into temp from event where eid=e_eid;

if temp <>0 then
update event set ecflag = '1' where eid = e_eid;

--final_message := concat(z,y);
open c1;
loop
   fetch c1 into r_userid;
   exit when c1%notfound;
   insert into message values(msg_seq.nextval,r_userid,'Event ID : ' || e_eid || ' has been
cancelled.',systimestamp);
   dbms_output.put_line('Message has been sent to user id: ' || r_userid);
end loop;
```

```
close c1;
update participant set pcflag = '1' where eid = e_eid;
--dbms_output.put_line('Registered User insert Complete');

open c2;
loop
    fetch c2 into w_userid;
    exit when c2%notfound;
    insert into message values(msg_seq.nextval,w_userid,'Event ID : ' || e_eid || ' has been
cancelled.',systimestamp);
    dbms_output.put_line('Message has been sent to user id: ' || w_userid);
end loop;
close c2;
delete from waitlist where eid = e_eid;
else

dbms_output.put_line('Event ID ' || e_eid ||' doesn’t exist');

end if;


end;
```

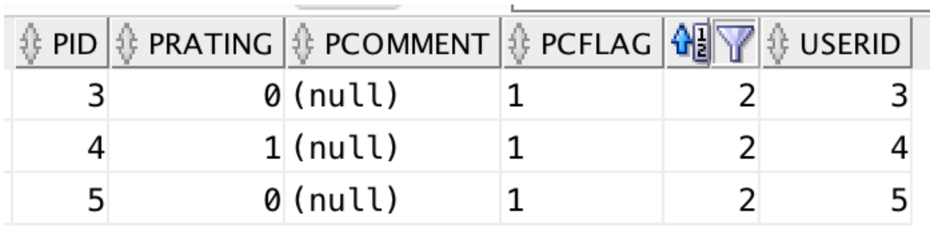
Test case 1: Event cancel (Existing Event ID)
By running
exec event_cancellation (3);

The cancel flag has been updated to 1 in the event table



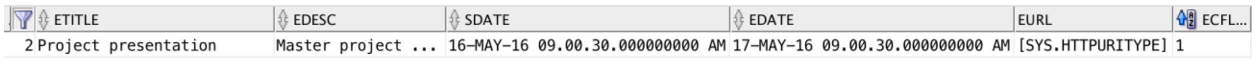
	ETITLE	EDESC	SDATE	EDATE	EURL	ECFLAG
3	Free food event	National food f...	15-APR-16 06.00.30.000000000 PM	15-APR-16 10.00.30.000000000 PM	[SYS.HTTPURITYPE]	1

Test case 2: Cancel the event which has participants
exec event_cancellation (2);



PID	PRATING	PCOMMENT	PCFLAG		USERID
3	0	(null)	1	2	3
4	1	(null)	1	2	4
5	0	(null)	1	2	5

The event 2 has been set the cancelled flag.

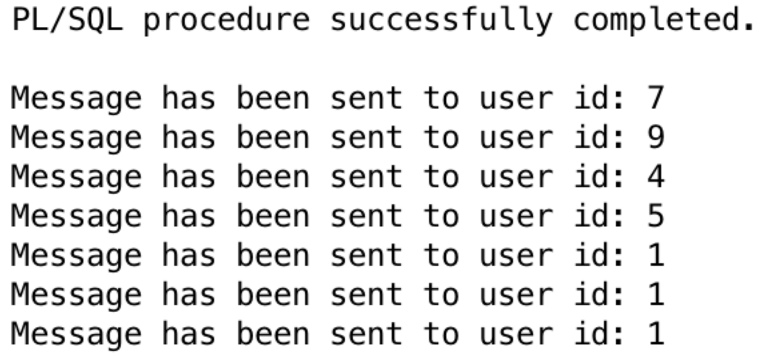


	ETITLE	EDESC	SDATE	EDATE	EURL	ECFL...
2	Project presentation	Master project ...	16-MAY-16 09.00.30.000000000 AM	17-MAY-16 09.00.30.000000000 AM	[SYS.HTTPURITYPE]	1

Test case 3: Cancel the event which has participant and waitlist.

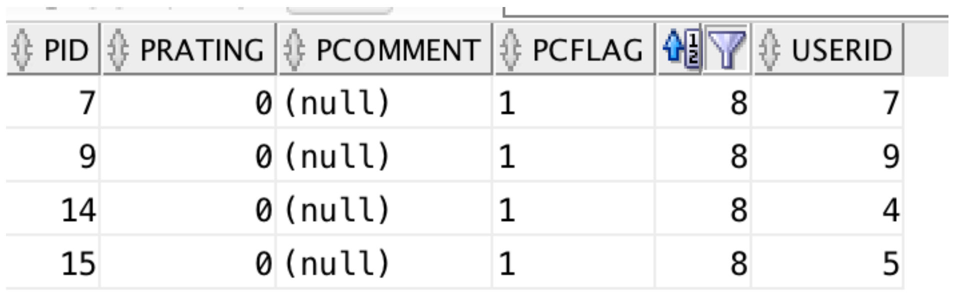
exec event_cancellation (8);

The system show the users that have been sent the message



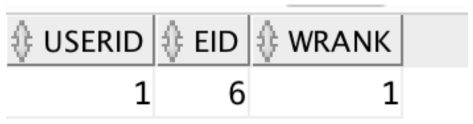
```
PL/SQL procedure successfully completed.

Message has been sent to user id: 7
Message has been sent to user id: 9
Message has been sent to user id: 4
Message has been sent to user id: 5
Message has been sent to user id: 1
Message has been sent to user id: 1
Message has been sent to user id: 1
```

All participants have been cancelled by being set the Pcflag.



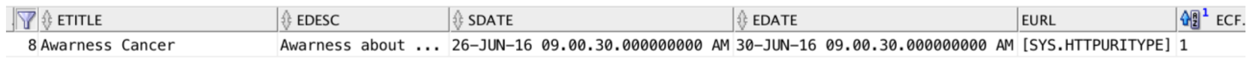
PID	PRATING	PCOMMENT	PCFLAG		USERID
7	0	(null)	1	8	7
9	0	(null)	1	8	9
14	0	(null)	1	8	4
15	0	(null)	1	8	5

No wait list on event 8. (All wait list in the event 8 have been removed)



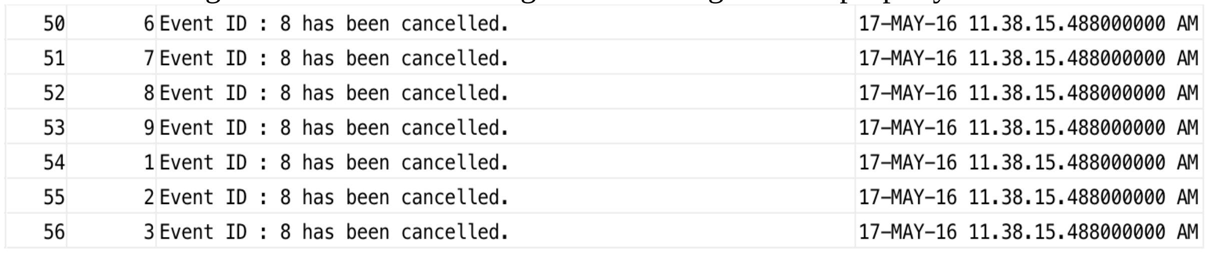
USERID	EID	WRANK
1	6	1

Event 8 has been flagged cancel



	ETITLE	EDESC	SDATE	EDATE	EURL	ECF.
8	Awarness Cancer	Awarness about ...	26-JUN-16 09.00.30.000000000 AM	30-JUN-16 09.00.30.000000000 AM	[SYS.HTTPURITYPE]	1

Check in message table to see the messages have been generated properly.



50	6	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
51	7	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
52	8	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
53	9	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
54	1	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
55	2	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM
56	3	Event ID : 8 has been cancelled.	17-MAY-16 11.38.15.488000000 AM

5. Update the start and end date and time of an event by providing event ID and new start/end date and time. Update the event table if the event exists. If the event does not exist, print out a message saying wrong event ID. Please generate a message for each user who has registered or on wait list with the new date and time of the event.

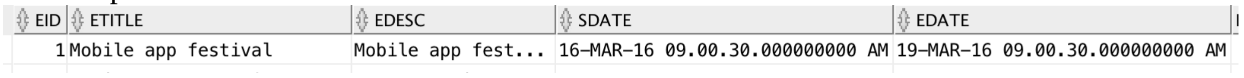
<u>Comments</u>-It Updates the start and end date and time of an event by providing event ID and new start/end date and time and updates the event table if the event exists. If the event does not exist, it print out a message saying wrong event ID. And also generates a message for each user who has registered or on wait list with the new date and time of the event.

```
create or replace PROCEDURE UpdateEvent(event_id in integer, n_sdate in timestamp, n_edate
in timestamp)
IS

Cursor c1 is select userid from waitlist where eid = event_id;
Cursor c2 is select userid from participant where eid = event_id;

t_userid usertable.userid%type;
t_eventid event.eid%type;
flag integer;

BEGIN

  select eid into t_eventid from event where eid = event_id and ecflag = '0';

/*
  -- check duplicate time stamp
  select count(*)
  into flag
  from event e,locations l
  where e.lid = l.lid and e.eid = event_id and ((sdate >= n_sdate and sdate <= n_edate)
  or (n_sdate >= sdate and n_sdate <= edate));
  if flag > 1 then
  dbms_output.put_line('New Timing is overlapped.');
  end if;
  */

  if t_eventid is not null then

      UPDATE event
      SET sdate = n_sdate, edate = n_edate
      WHERE eid = event_id;

  Open c1;
  Loop
    fetch c1 into t_userid;
    exit when c1%notfound;
    dbms_output.put_line('Wait list ID: ' || t_userid || ' has been sent the message.');
    insert into message values(msg_seq.nextval,t_userid, 'The event ID:' || event_id ||
        ' has been changed the time. New Start date is : ' ||
        n_sdate || ' and new end date is : ' || n_edate, systimestamp);
  End loop;
  Close c1;

  Open c2;
  Loop
    fetch c2 into t_userid;
    exit when c2%notfound;
    dbms_output.put_line('User ID: ' || t_userid || ' has been sent the message.');
    insert into message values(msg_seq.nextval,t_userid, 'The event ID:' || event_id ||
        ' has been changed the time. New Start date is : ' ||
        n_sdate || ' and new end date is : ' || n_edate, systimestamp);
  End loop;
  Close c2;

  end if;

  exception
  when no_data_found then
   dbms_output.put_line('Wrong event ID');

END;
```

Test case 1: General update

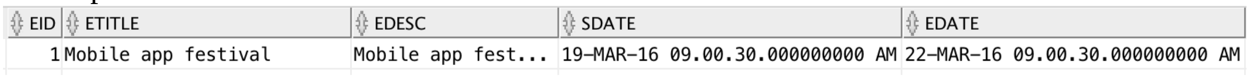
```
Declare
begin
UpdateEvent(1, timestamp '2016-3-19 09:00:30.00', timestamp '2016-3-22 09:00:30.00') ;
end;
```

Before update data



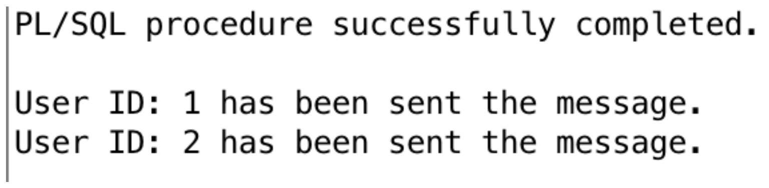
EID	ETITLE	EDESC	SDATE	EDATE
1	Mobile app festival	Mobile app fest...	16-MAR-16 09.00.30.000000000 AM	19-MAR-16 09.00.30.000000000 AM

After update data



EID	ETITLE	EDESC	SDATE	EDATE
1	Mobile app festival	Mobile app fest...	19-MAR-16 09.00.30.000000000 AM	22-MAR-16 09.00.30.000000000 AM

and the messages have been sent to the participants.

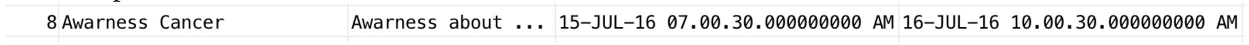


```
PL/SQL procedure successfully completed.

User ID: 1 has been sent the message.
User ID: 2 has been sent the message.
```

Test case 2: Update event that has participants and person on the waitlists.

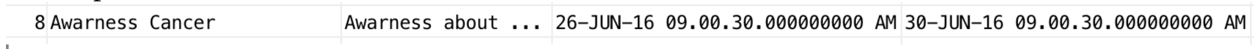
```
declare
begin
UpdateEvent(8, timestamp '2016-6-26 09:00:30.00', timestamp '2016-6-30 09:00:30.00') ;
end;
```

Before update

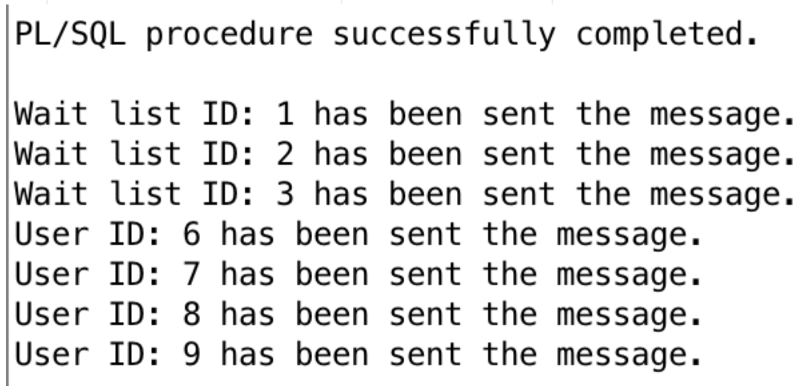


8	Awarness Cancer	Awarness about ...	15-JUL-16 07.00.30.000000000 AM	16-JUL-16 10.00.30.000000000 AM

After update

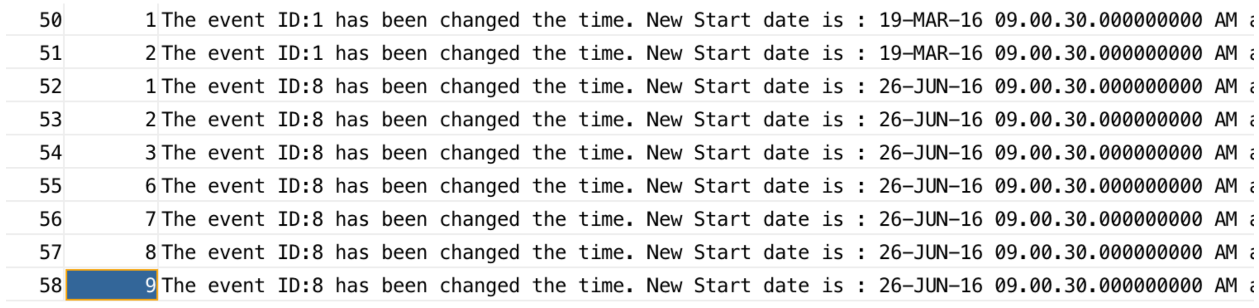


8	Awarness Cancer	Awarness about ...	26-JUN-16 09.00.30.000000000 AM	30-JUN-16 09.00.30.000000000 AM



```
PL/SQL procedure successfully completed.

Wait list ID: 1 has been sent the message.
Wait list ID: 2 has been sent the message.
Wait list ID: 3 has been sent the message.
User ID: 6 has been sent the message.
User ID: 7 has been sent the message.
User ID: 8 has been sent the message.
User ID: 9 has been sent the message.
```

Then, check the message table (The messages have been generated for both test cases)



50	1	The event ID:1 has been changed the time. New Start date is : 19-MAR-16 09.00.30.000000000 AM
51	2	The event ID:1 has been changed the time. New Start date is : 19-MAR-16 09.00.30.000000000 AM
52	1	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
53	2	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
54	3	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
55	6	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
56	7	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
57	8	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM
58	9	The event ID:8 has been changed the time. New Start date is : 26-JUN-16 09.00.30.000000000 AM

6. Search for events with a certain keyword in the title (you can use like), return start and end date and time, full event title, location, url, organizer name and email, and whether the event is full.

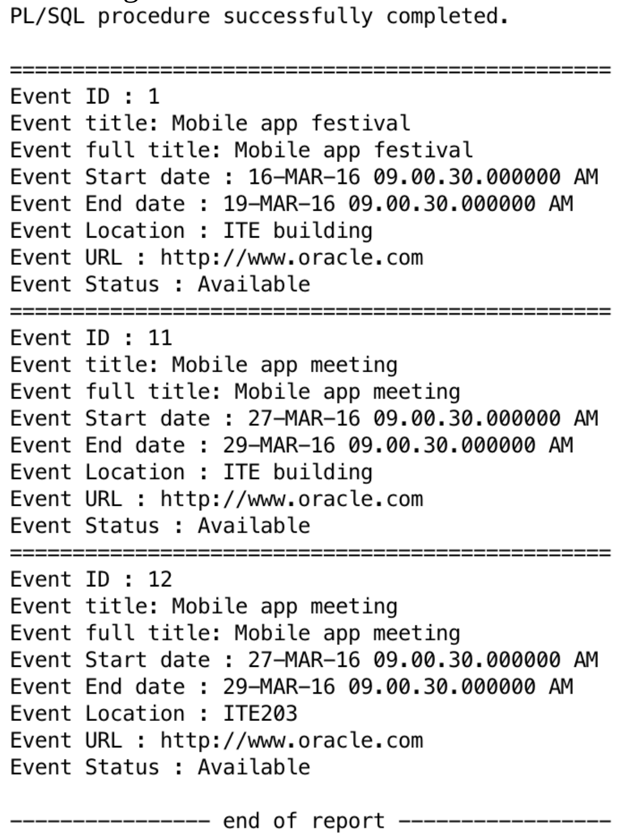
Comments- It Search for events with a certain keyword in the title and returns start and end date and time, full event title, location, url, organizer name and email, and status of the event

```
create or replace PROCEDURE SearchEvent(searchterm in varchar)
IS

t_location LOCATIONS.LDESC%type;
t_eventid event.eid%type;
t_searchterm varchar2(200) := '%'|| searchterm ||'%';
```

```
t_event event%rowtype;
t_participantnumber integer;
t_maxp_event integer;  -- Max participant

Cursor c1 is select eid from event where etitle like t_searchterm;

BEGIN

  Open c1;
  Loop
    fetch c1 into t_eventid;
    exit when c1%notfound;

    select * into t_event from event where eid = t_eventid;
    dbms_output.put_line('=================================================');
    dbms_output.put_line('Event ID : ' || t_event.eid);
    dbms_output.put_line('Event title: ' || t_event.etitle);
    dbms_output.put_line('Event full title: ' || t_event.edesc);
    dbms_output.put_line('Event Start date : ' || t_event.sdate);
    dbms_output.put_line('Event End date : ' || t_event.edate);

    select ldesc into t_location from locations where lid = t_event.lid;

    dbms_output.put_line('Event Location : ' || t_location);
    dbms_output.put_line('Event URL : ' || t_event.eurl.geturl());

    select count(*) into t_participantnumber from participant
    where EID = t_event.eid and pcflag ='0';

    select lcap into t_maxp_event from locations where lid = t_event.lid;
--  dbms_output.put_line('p: ' || t_participantnumber);
--  dbms_output.put_line('m: ' || t_maxp_event);
    if t_participantnumber = t_maxp_event then
        dbms_output.put_line('Event Status : Full');
    else
        dbms_output.put_line('Event Status : Available');
    end if;

  End loop;

  if c1%rowcount = 0 then
  dbms_output.put_line('No data from the search term');
  else
  dbms_output.put_line('                                                  ');
  dbms_output.put_line('---------------- end of report -----------------');
  end if;
```

  Close c1;

END;
Test case 1: search event with a word ‘Mobile’.
Test Run by
exec SearchEvent('Mobile') ;

You will get all event with the word Mobile in their titles as below:


```
PL/SQL procedure successfully completed.

================================================
Event ID : 1
Event title: Mobile app festival
Event full title: Mobile app festival
Event Start date : 16-MAR-16 09.00.30.000000 AM
Event End date : 19-MAR-16 09.00.30.000000 AM
Event Location : ITE building
Event URL : http://www.oracle.com
Event Status : Available
================================================
Event ID : 11
Event title: Mobile app meeting
Event full title: Mobile app meeting
Event Start date : 27-MAR-16 09.00.30.000000 AM
Event End date : 29-MAR-16 09.00.30.000000 AM
Event Location : ITE building
Event URL : http://www.oracle.com
Event Status : Available
================================================
Event ID : 12
Event title: Mobile app meeting
Event full title: Mobile app meeting
Event Start date : 27-MAR-16 09.00.30.000000 AM
Event End date : 29-MAR-16 09.00.30.000000 AM
Event Location : ITE203
Event URL : http://www.oracle.com
Event Status : Available

---------------- end of report -----------------
```

Test case 2: search event with the word ‘your’

```
PL/SQL procedure successfully completed.

================================================
Event ID : 6
Event title: Make your poster
Event full title: poster making competition
Event Start date : 15-JUN-16 09.00.30.000000 AM
Event End date : 16-JUN-16 02.00.30.000000 PM
Event Location : Library
Event URL : http://www.umbc.edu
Event Status : Available
================================================
Event ID : 9
Event title: Paint your mind
Event full title: Painting competition
Event Start date : 15-AUG-16 08.00.30.000000 AM
Event End date : 16-AUG-16 10.00.30.000000 AM
Event Location : ITE204
Event URL : http://www.umbc.edu
Event Status : Available

---------------- end of report -----------------
```