1. Print out the total number of events, the average number of users registered per event, and the average number of users on wait list.

Comment- It Print out the total number of events, the average number of users registered per event, and the average number of users on wait list.

```
create or replace PROCEDURE ShowEventSummary
IS
Cursor c1 is select eid from event;
t_location LOCATIONS.LDESC%type;
t_eventid event.eid%type;
t_event event%rowtype;
t_participantnumber integer;
t_waitnumber integer;
t_maxp_event integer;  -- Max participant
t_max_waitrank integer;
n_waitlist integer;
t_userid integer;

total_event integer;
loop_count_id integer;

a_participant float;
a_waitlist float;

BEGIN

  select count(*) into total_event from event;
  dbms_output.put_line('Total number of Event is : ' || total_event);

  a_participant := 0;
  a_waitlist := 0;

  Open c1;
  Loop
    fetch c1 into t_eventid;
    exit when c1%notfound;
    --dbms_output.put_line('Wait list ID: ' || t_userid || ' has been sent the message.');

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

    select count(*) into t_waitnumber from waitlist
       where EID = t_eventid;

    dbms_output.put_line('Event ID: ' || t_eventid
    || ' --- Par No.  no. : ' || t_participantnumber || ' / ' || t_waitnumber);
```

```
    a_participant := a_participant + t_participantnumber;
    a_waitlist := a_waitlist + t_waitnumber;

  End loop;
  Close c1;

  dbms_output.put_line('Average number of users registered per event : ' ||
ROUND(a_participant/total_event,2));
  dbms_output.put_line('Average number of users waitlist per event : ' ||
ROUND(a_waitlist/total_event,2));


END;
```

Test case 1 : To show event summary, we can run the procedure by
Exec ShowEventSummary

```
Total number of Event is : 12
Event ID: 1 --- Par No.  no. : 1 / 0
Event ID: 2 --- Par No.  no. : 2 / 0
Event ID: 3 --- Par No.  no. : 0 / 0
Event ID: 4 --- Par No.  no. : 0 / 0
Event ID: 5 --- Par No.  no. : 0 / 0
Event ID: 6 --- Par No.  no. : 5 / 0
Event ID: 7 --- Par No.  no. : 0 / 0
Event ID: 8 --- Par No.  no. : 4 / 3
Event ID: 9 --- Par No.  no. : 0 / 0
Event ID: 10 --- Par No.  no. : 0 / 0
Event ID: 11 --- Par No.  no. : 0 / 0
Event ID: 12 --- Par No.  no. : 0 / 0
Average number of users registered per event : 1
Average number of users waitlist per event : .25
```

then check the accuracy by add in 2 participants randomly.
And rerun the report again by
exec ParticipantRegister(1,4);
exec ParticipantRegister(2,4);
exec ShowEventSummary();

2 more new users will be registered in to event 4.
And the report should have some changes.

As below picture, The system updated the report correctly. And the average number of user increase while number of waitlist unchanged as below:

```
PL/SQL procedure successfully completed.

Total number of Event is : 12
Event ID: 1 --- Par No.  no. : 1 / 0
Event ID: 2 --- Par No.  no. : 2 / 0
Event ID: 3 --- Par No.  no. : 0 / 0
Event ID: 4 --- Par No.  no. : 2 / 0
Event ID: 5 --- Par No.  no. : 0 / 0
Event ID: 6 --- Par No.  no. : 5 / 0
Event ID: 7 --- Par No.  no. : 0 / 0
Event ID: 8 --- Par No.  no. : 4 / 3
Event ID: 9 --- Par No.  no. : 0 / 0
Event ID: 10 --- Par No.  no. : 0 / 0
Event ID: 11 --- Par No.  no. : 0 / 0
Event ID: 12 --- Par No.  no. : 0 / 0
Average number of users registered per event : 1.17
Average number of users waitlist per event : .25
```

15. Print out the top K events with the most participants (only counting those registered), the top K locations with most number of events, and top K events with the highest average ratings. K is an input parameter.

```
create or replace PROCEDURE ShowTopEvent(input_k in integer)
IS

Cursor c1 is select event.eid, count (*) as ptotal from event,participant where
                event.eid = participant.eid and participant.pcflag ='0' and
                event.ecflag ='0'
                group by event.eid
                order by count (*) desc;
cursor c2 is select event.lid, count (*) as ltotal from event,locations where
                event.lid = locations.lid and
                event.ecflag ='0'
                group by event.lid
                order by count (*) desc;
Cursor c3 is select event.eid, avg (prating) as prate from event,participant where
                event.eid = participant.eid and participant.pcflag ='0' and
                event.ecflag ='0' and participant.prating <> 0
                group by event.eid
                order by avg(prating) desc;


t_c1 c1%rowtype;
t_c2 c2%rowtype;
t_c3 c3%rowtype;

total_event integer;
loop_count_id integer;

a_participant float;
a_waitlist float;

BEGIN

  select count(*) into total_event from event;
  dbms_output.put_line('Total number of Event is : ' || total_event);

  a_participant := 0;
  a_waitlist := 0;

dbms_output.put_line('=========Top Participated Event=========');
  Open c1;
  Loop
    fetch c1 into t_c1;
    dbms_output.put_line('Event ID: '|| t_c1.eid);
    dbms_output.put_line('Participant No: '|| t_c1.ptotal);
```

```
  exit when c1%notfound or c1%rowcount = input_k;
 End loop;

  dbms_output.put_line('==============Top Location==============');
 Open c2;
 Loop
  fetch c2 into t_c2;
  dbms_output.put_line('Location ID: '|| t_c2.lid);
  dbms_output.put_line('Number of event in this location: '|| t_c2.ltotal);

  exit when c2%notfound or c2%rowcount = input_k;
 End loop;

 dbms_output.put_line('============Top Rated Event=============');
 Open c3;
 Loop
  fetch c3 into t_c3;
  dbms_output.put_line('Event ID: '|| t_c3.eid);
  dbms_output.put_line('Event Rating Score : '|| t_c3.prate);

  exit when c3%notfound or c3%rowcount = input_k;

 End loop;

 Close c1;
 Close c2;
 Close c3;

END;
```

Test case 1:

Select top 2 events with the most participants
the top 2 locations with most number of events,
and top 2 events with the highest average ratings.
By running
Exec ShowTopEvent(2)

Report show top 2 (participated events, locations, rated events)

```
Total number of Event is : 12
=========Top Participated Event=========
Event ID: 6
Participant No: 5
Event ID: 8
Participant No: 4
=============Top Location==============
Location ID: 1
Number of event in this location: 2
Location ID: 6
Number of event in this location: 2
============Top Rated Event============
Event ID: 2
Event Rating Score : 4
Event ID: 8
Event Rating Score : 2
```

Prove the correct of feature by adding
We register event 2 to make event 2 has the highest participant.
We update event 1 with all 5 score of rating
And create event to make ‘Common’ location (EID = 3) to be the most popular location

```
exec ParticipantRegister(6,2);
exec ParticipantRegister(7,2);
exec ParticipantRegister(8,2);
exec ParticipantRegister(9,2);
exec UserEventReview(2, 1, 5, 'Great Event !!');
exec UserEventReview(1, 1, 5, 'Great Event !!');
exec CreateEvent('Test 1', timestamp '2020-05-29 09:00:30.00',
timestamp '2020-05-28 09:00:30.00', 'Common', 'http://www.umbc.edu', 1 ,'test error');
exec CreateEvent('Test 1', timestamp '2021-05-29 09:00:30.00',
timestamp '2021-05-28 09:00:30.00', 'Common', 'http://www.umbc.edu', 1 ,'test error');
exec CreateEvent('Test 1', timestamp '2022-05-29 09:00:30.00',
timestamp '2022-05-28 09:00:30.00', 'Common', 'http://www.umbc.edu', 1 ,'test error');
```

The new report show as below by run Exec ShowTopEvent(2);

```
Total number of Event is : 15
=========Top Participated Event=========
Event ID: 2
Participant No: 7
Event ID: 6
Participant No: 5
=============Top Location==============
Location ID: 3
Number of event in this location: 4
Location ID: 6
Number of event in this location: 2
============Top Rated Event=============
Event ID: 1
Event Rating Score : 5
Event ID: 2
Event Rating Score : 4
```

New top participated event is now event 2.
New Top location is now 'Common' ID = 3
New Top rated event is now Event ID 1.

Try Exec ShowTopEvent(3);
The report will printout 3 Top participated event, 3 top locations and 3 top rating event in the screen.

```
Total number of Event is : 15
=========Top Participated Event=========
Event ID: 2
Participant No: 7
Event ID: 6
Participant No: 5
Event ID: 8
Participant No: 4
=============Top Location==============
Location ID: 3
Number of event in this location: 4
Location ID: 6
Number of event in this location: 2
Location ID: 1
Number of event in this location: 2
============Top Rated Event=============
Event ID: 1
Event Rating Score : 5
Event ID: 2
Event Rating Score : 4
Event ID: 8
Event Rating Score : 2
```