**----------------------------------------------------------------------------------------------------------------**

create or replace PROCEDURE create\_hotel(

street\_address in varchar2,

city in varchar2,

state in varchar2,

zip\_code in varchar2,

phone\_number in varchar2,

max\_small\_room\_avail in number,

max\_medium\_room\_avail in number,

max\_large\_room\_avail in number

)

--Creates a hotel give the following information

--@param street\_address - the address of the street

--@param city, the city the hotel is located in

--@param state, the state the hotel is located in

--@param zip\_code, the zipcode the hotel is located in

--@param phone\_number, the phone number of the hotel

--@param max\_small\_room\_avail, the number of small rooms available for a hotel

--@param max\_medium\_room\_avail, the number of medium rooms available for a hotel

--@param max\_large\_room\_avail, the number of large rooms available for a hotel

AS

able\_to\_insert\_hotel boolean := true;

BEGIN

--check to make sure all values are not null

if(street\_address is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Street Address is null, please add a street address');

elsif(city is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('city is null, please add a city');

elsif(state is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('State is null, please add a state');

elsif(zip\_code is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Zip code is null, please add a zip code');

elsif(length(zip\_code) > 5) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Zip code length is too long, please enter a valid zip code');

elsif(phone\_number is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Phone number is null, please add a phone number');

elsif(max\_small\_room\_avail is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Number of medium rooms available is null, please add number of medium rooms available');

elsif(max\_medium\_room\_avail is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Number of medium rooms available is null, please add number of medium rooms available');

elsif(max\_large\_room\_avail is null) then

able\_to\_insert\_hotel := false;

dbms\_output.put\_line('Number of large rooms available is null, please add number of large rooms available');

end if;

if(able\_to\_insert\_hotel) then

INSERT INTO HOTEL --(hotel\_id, street\_address, city, state, zip\_code, phone\_number, max\_small\_room\_avail, max\_medium\_room\_avail, max\_large\_room\_avail)

VALUES (hotel\_id\_seq.nextval, street\_address, city, state, zip\_code, phone\_number, max\_small\_room\_avail, max\_medium\_room\_avail, max\_large\_room\_avail);

else

dbms\_output.put\_line('Unable to insert hotel with the provided following info: ' || chr(10) ||

'street\_address: ' || street\_address || chr(10) ||

'city: ' || city || chr(10) ||

'state: ' || state || chr(10) ||

'zip\_code: ' || zip\_code || chr(10) ||

'phone\_number: ' || phone\_number || chr(10) ||

'max\_small\_room\_avail: ' || max\_small\_room\_avail || chr(10) ||

'max\_medium\_room\_avail: ' || max\_medium\_room\_avail || chr(10) ||

'max\_large\_room\_avail: ' || max\_large\_room\_avail);

end if;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX

THEN dbms\_output.put\_line('Hotel already exists!');

END;



SET serveroutput ON;

CREATE OR REPLACE function find\_hotel(

find\_street\_address in varchar2,

find\_city in varchar2,

find\_state in varchar2,

find\_zip\_code in varchar2

)

--Find a hotel based on the following information

--@param street\_address, the address of the hotel

--@param city, the city the hotel is located in

--@param state, the state the hotel is located in

--@param zip\_code, the zip code of the city

--@return number, the hotel\_id

RETURN number

IS

h\_id number;

BEGIN

select hotel.hotel\_id

into h\_id

from hotel hotel

where hotel.street\_address = find\_street\_address and

hotel.city = find\_city and

hotel.state = find\_state and

hotel.zip\_code = find\_zip\_code;

dbms\_output.put\_line('Hotel Id: ' || h\_id);

return h\_id;

EXCEPTION

when no\_data\_found

then dbms\_output.put\_line('Hotel not found');

END;

declare

find\_street\_address varchar2(100) := '1000 Hilltop Circle';

find\_city varchar2(50) := 'Baltimore';

find\_state varchar2(2) := 'MD' ;

find\_zip\_code varchar2(5) := '21250';

hotel\_id number;

begin

hotel\_id := find\_hotel(find\_street\_address, find\_city, find\_state, find\_zip\_code);

dbms\_output.put\_line('Looking for Hotel id with information:' || chr(10) ||

'---------------------------------------------------------' || chr(10) ||

'street\_address: ' || find\_street\_address || chr(10) ||

'city: ' || find\_city || chr(10) ||

'state: ' || find\_state || chr(10) ||

'zip\_code: ' || find\_zip\_code);

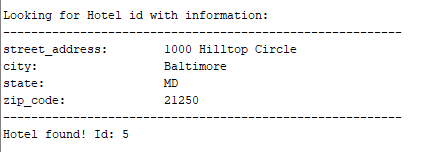
if(hotel\_id is not null) then

dbms\_output.put\_line('---------------------------------------------------------');

dbms\_output.put\_line('Hotel found! Id: ' || hotel\_id);

end if;

end;



SET serveroutput ON;

CREATE OR REPLACE PROCEDURE display\_hotel(

h\_id number

)

--Display a hotels details based on the hotel\_id

--@param hotel\_id the id of the hotel

AS

hotel\_information hotel%rowtype;

BEGIN

select \*

into hotel\_information

from hotel hotel

where hotel.hotel\_id = h\_id;

dbms\_output.put\_line(

'Hotel information' || chr(10) ||

'--------------------------------------------------' || chr(10) ||

'street\_address : ' || hotel\_information.street\_address || chr(10) ||

'city : ' || hotel\_information.city || chr(10) ||

'state : ' || hotel\_information.state || chr(10) ||

'zip\_code : ' || hotel\_information.zip\_code || chr(10) ||

'phone\_number : ' || hotel\_information.phone\_number || chr(10) ||

'max\_small\_room\_avail : ' || hotel\_information.max\_small\_room\_avail || chr(10) ||

'max\_medium\_room\_avail: ' || hotel\_information.max\_medium\_room\_avail || chr(10) ||

'max\_large\_room\_avail : ' || hotel\_information.max\_large\_room\_avail || chr(10) ||

'--------------------------------------------------'

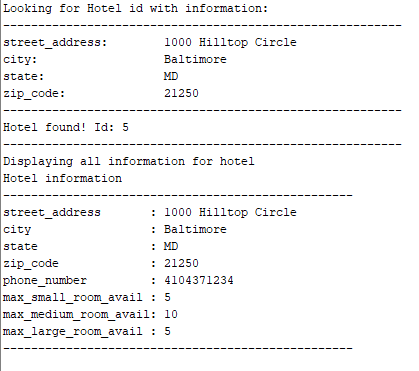
);

EXCEPTION

when no\_data\_found

then dbms\_output.put\_line('Hotel with id:' || h\_id || ' not found');

END;



--solution 7

--- Procedure for creating a reservation ---

--- Creates event based on the all the input parameters required for creating a reservation.---

--- Prints the event reservation confirmation with the reservation ID----

create or replace procedure

reserve\_for\_event

(

hotelid in hotel.hotel\_id%type,

hostname in hosts.host\_name%type,

startdate in event\_reservation.START\_DATE%TYPE,

enddate in event\_reservation.END\_DATE%TYPE,

dateofreservation in event\_reservation.DATE\_OF\_RESERVATION%TYPE,

numofsmallroom in event\_reservation.NUM\_SMALL\_ROOMS%TYPE,

numofmediumroom in event\_reservation.NUM\_MEDIUM\_ROOMS%TYPE,

numoflargeroom in event\_reservation.NUM\_LARGE\_ROOMS%TYPE,

dis in event\_reservation.discount%type,

numberofpeopleattending IN event\_reservation.NUM\_OF\_GUESTS%TYPE ,

eventname in event\_type.event\_name%type,

eventreservationid out event\_reservation.event\_reservation\_id%type

)

-- @param hotelid

-- @param host name

-- @param start date

-- @param end date

-- @param reservation date

-- @param numbers of small rooms

-- @param numbers of medium rooms

-- @param numbers of large rooms

-- @param discount

-- @param no of people attending

-- @param event name

-- @param reservation id

as

eventid event\_type.event\_id%type;

hostid hosts.host\_id%type;

begin

select event\_id into eventid from event\_type where event\_name = eventname;

select host\_id into hostid from hosts where host\_name = hostname;

--- Using the sequence to generate the reservation ID ---

eventreservationid := event\_reservation\_id\_seq.nextval;

--- inserting the data after creating the reservation ---

insert into event\_reservation

values (eventreservationid, hotelid, hostid, startdate, enddate, dateofreservation, numofsmallroom, numofmediumroom , numoflargeroom, dis, numberofpeopleattending, eventid);

update hotel set max\_small\_room\_avail = max\_small\_room\_avail - numofsmallroom;

update hotel set max\_medium\_room\_avail = max\_medium\_room\_avail - numofmediumroom;

update hotel set max\_large\_room\_avail = max\_large\_room\_avail - numoflargeroom;

end;

--- output of the reservation confirmation ---

Declare

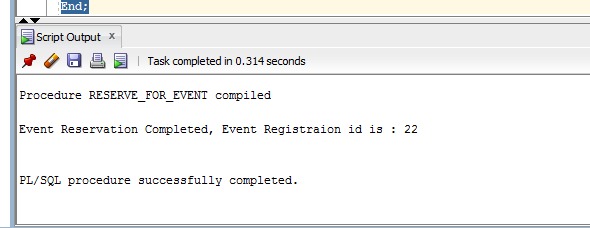
res\_id number;

Begin

reserve\_for\_event(5,'Kajal Pansare', date '2006-08-10', date '2006-08-13', date '2006-08-10',4, 2, 6, 5, 2,'Birthday', res\_id);

Dbms\_output.put\_line('Event Reservation Completed, Event Registraion id is : ' || res\_id);

End;



--solution8

--- Procedure for checking the event for the reservation ---

create or replace procedure

Check\_event\_reservation

(

hostname in hosts.host\_name%type,

dateofres in event\_reservation.date\_of\_reservation%type,

hotelid in hotel.hotel\_id%type

)

is

---@param name of the host

---@param reservation date

---@param hotelid

--- Explicit cursor to fetch and hold the required data---

cursor c1 is select e.event\_reservation\_id from event\_reservation e, hosts h where e.host\_id=h.host\_id

and h.host\_name = hostname and e.hotel\_id = hotelid and e.date\_of\_reservation = dateofres;

eventreservationid event\_reservation.event\_reservation\_id%type;

begin

Open c1;

Loop

--- Fetching the reservation id from the cursor---

fetch c1 into eventreservationid;

exit when c1%notfound;

dbms\_output.put\_line('Event Reservation ID is: ' || eventreservationid );

end loop;

close c1;

--- Handling the exception ---

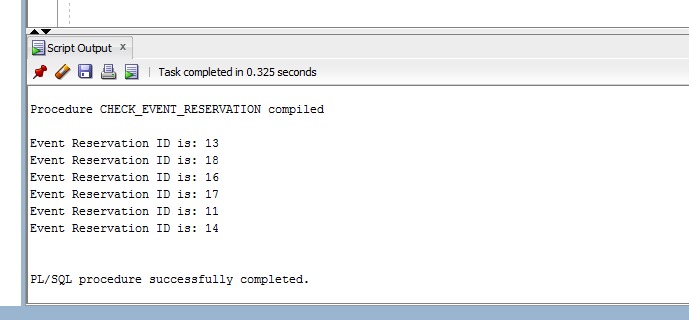
exception

when no\_data\_found then

dbms\_output.put\_line('No event reservation found');

end;

exec Check\_event\_reservation('George Karabatis',date '2006-08-10',5);



Priyanka Ranade ( Member 3)

**-------------------------------------------------------------------------------------------------------------------------------**

--Procedure print\_event

--Show events by person: given a hosts name as an input, find all events under that name

--Print event reservation ID and event name for each event per host

create or replace procedure print\_event(v\_hname hosts.host\_name%type)

is

cursor c1 is select event\_reservation\_id, event\_name

from event\_type,event\_reservation, hosts where

hosts.host\_id=event\_reservation.host\_id and event\_type.event\_id = event\_reservation.event\_id and host\_name=v\_hname;

--@param event\_reservation\_id - unique id for reservation

--@param event\_name - string name of the event

--@param host\_id - unique id for host

--@param event\_id- unique id for event

--variables to hold output from cursor.

--of type event\_id and event\_name from event\_type table

v\_event\_res\_id event\_reservation.event\_id%type;

v\_event\_name event\_type.event\_name%type;

begin

open c1;

Loop

--fetching cursor into holding variables v\_event\_Res\_id and v\_event\_name

fetch c1 into v\_event\_res\_id,v\_event\_name;

exit when c1%notfound;

--print event id and name of the event`

dbms\_output.put\_line('Event Reservation ID is: ' || v\_event\_res\_id || ' and ' || 'Event Name is: ' || v\_event\_name );

end loop;

close c1;

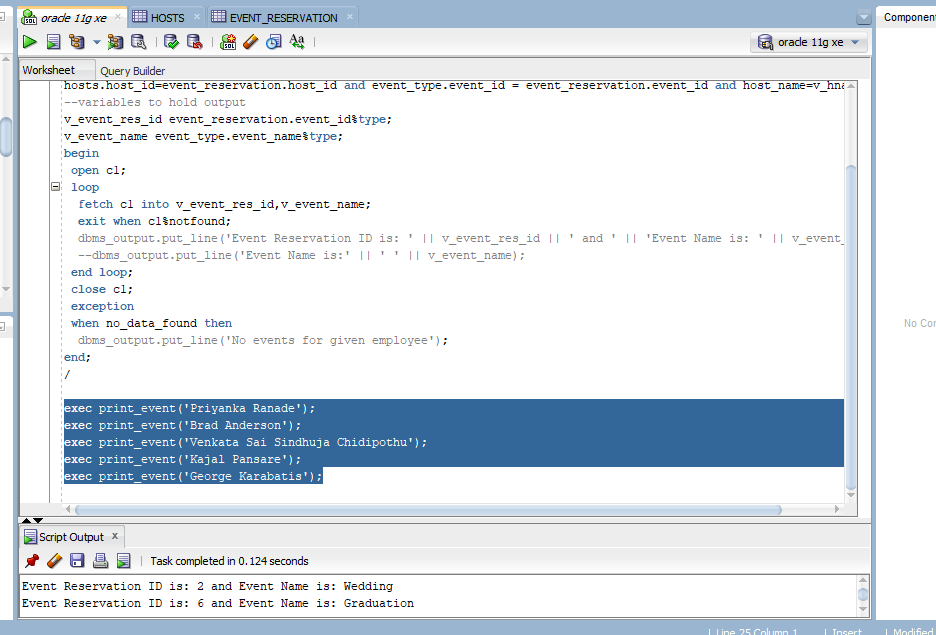
exception

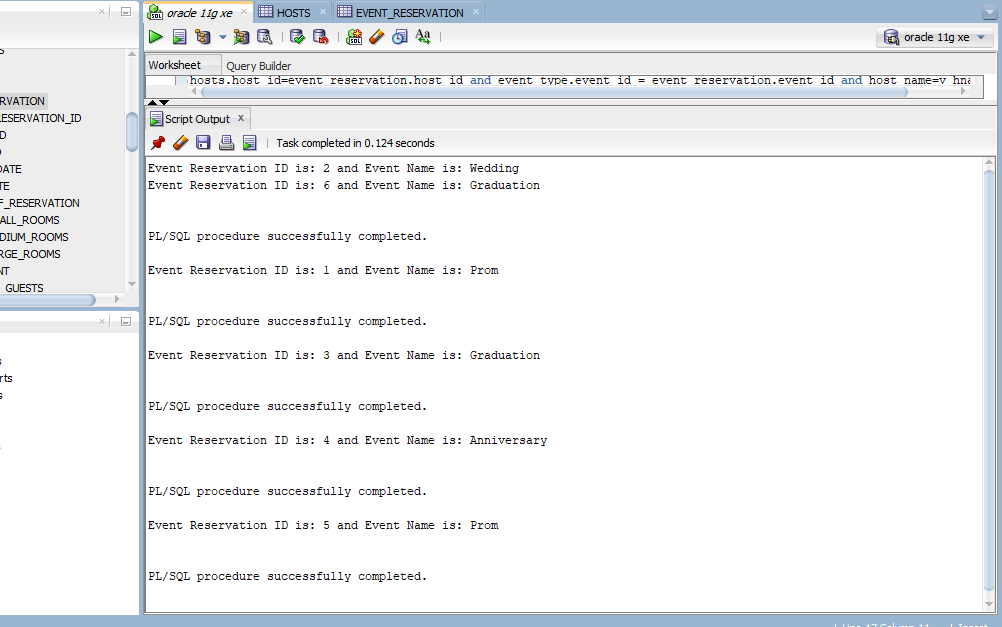
when no\_data\_found then

dbms\_output.put\_line('No events for given employee');

end;

**-- Screenshots of Execution Statements and Output**



****

-- Procedure print\_event\_type

--Show specific event: Given an event type (Birthday, Wedding, etc.)

--display all events of that type in all hotels along with the address of the hotel and the date of the event

-- Taking an event type (Birthday, Wedding, etc) as input, joining event\_reservation, event\_type, and hotel tables

-- to show events of input type, hotel address, and date of event in all hotels

create or replace procedure print\_event\_type(v\_ename event\_type.event\_name%type)

is

cursor c1 is select event\_name,street\_address,city,state,zip\_code,start\_date

from event\_reservation, event\_type, hotel

where event\_name= v\_ename

and event\_reservation.event\_id = event\_type.event\_id

and event\_reservation.hotel\_id = hotel.hotel\_id;

--@param event\_reservation\_id - unique id for reservation

--@param event\_name - string name of the event

--@param event\_id - unique event id

--@hotel\_id - unique id for hotel

--variables to hold output

v\_event\_name event\_type.event\_name%type;

v\_street\_addr hotel.street\_address%type;

v\_city hotel.city%type;

v\_state hotel.state%type;

v\_zip\_code hotel.zip\_code%type;

v\_start\_date event\_reservation.start\_date%type;

--using cursor to fetch records of input event type

begin

open c1;

loop

fetch c1 into v\_event\_name,v\_street\_addr,v\_city,v\_state,v\_zip\_code,v\_start\_date;

exit when c1%notfound;

--dbms\_output.put\_line('Event Reservation ID is: ' || v\_event\_res\_id || ' and ' || 'Event Name is: ' || v\_event\_name );

dbms\_output.put\_line('Event Type: ' || v\_event\_name || ',' || ' Street Address: ' || v\_street\_addr || ',' || ' City: ' || v\_city || ',' || ' State: ' || v\_state || ',' || ' Zip Code: ' || v\_zip\_code || ',' || ' Event Date: ' || v\_start\_date);

end loop;

close c1;

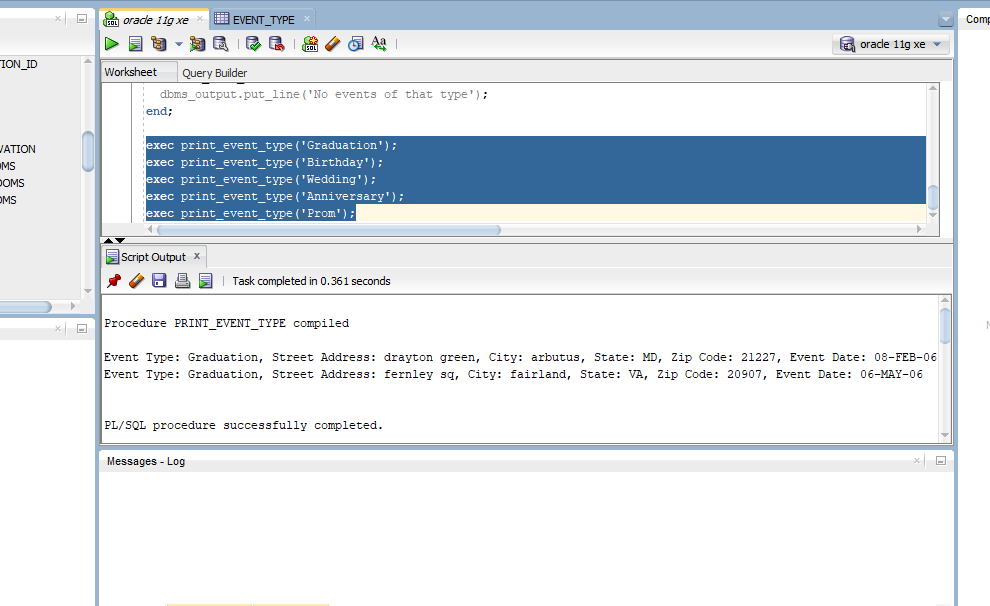
exception

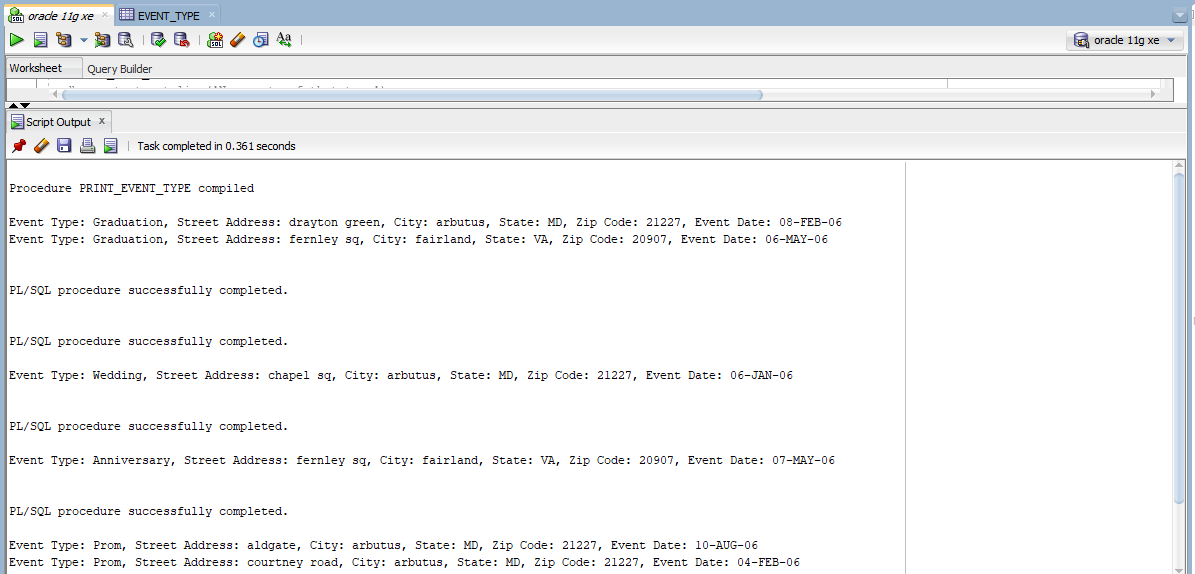
when no\_data\_found then

dbms\_output.put\_line('No events of that type');

end;

**--Screenshot of Execution Statements and Output**

****

****

**----------------------------------------------------------------------------------------------------------------**

Solution 18 :

--procedure reservation\_services\_report

--show services on reservation: given a event reservation as input

--print service name and num of guests for each event per reservation id

CREATE OR REPLACE PROCEDURE reservation\_services\_report(eventreservationid IN event\_reservation.event\_reservation\_id%TYPE)

IS

CURSOR c1 IS SELECT services.service\_type , event\_reservation.num\_of\_guests from services,services\_for\_events,event\_reservation

where

services.service\_id = services\_for\_events.service\_id AND services\_for\_events.event\_reservation\_id = eventreservationid;

--@param service\_type - String for service type

--@param num\_of\_guests - Count of number of guests

--@param service\_id - Unique ID for services

--@param event\_reservation\_id - Unique ID for reservations

--variables to hold output from cursor

--of type service\_type and num\_of\_ guests from services and event\_reservation

servicename SERVICES.service\_type%TYPE;

numofguests event\_reservation.num\_of\_guests%type;

BEGIN

OPEN c1;

LOOP

--fetching cursor into holding variables service name, no of guests---

FETCH c1 into servicename,numofguests;

EXIT WHEN c1%NOTFOUND;

---Printing the output of Service name and no of guests ---

DBMS\_OUTPUT.PUT\_LINE('service name is :' ||servicename);

DBMS\_OUTPUT.PUT\_LINE('number of attendees is :' ||numofguests);

END LOOP;

close c1;

--Handling the exception ---

EXCEPTION

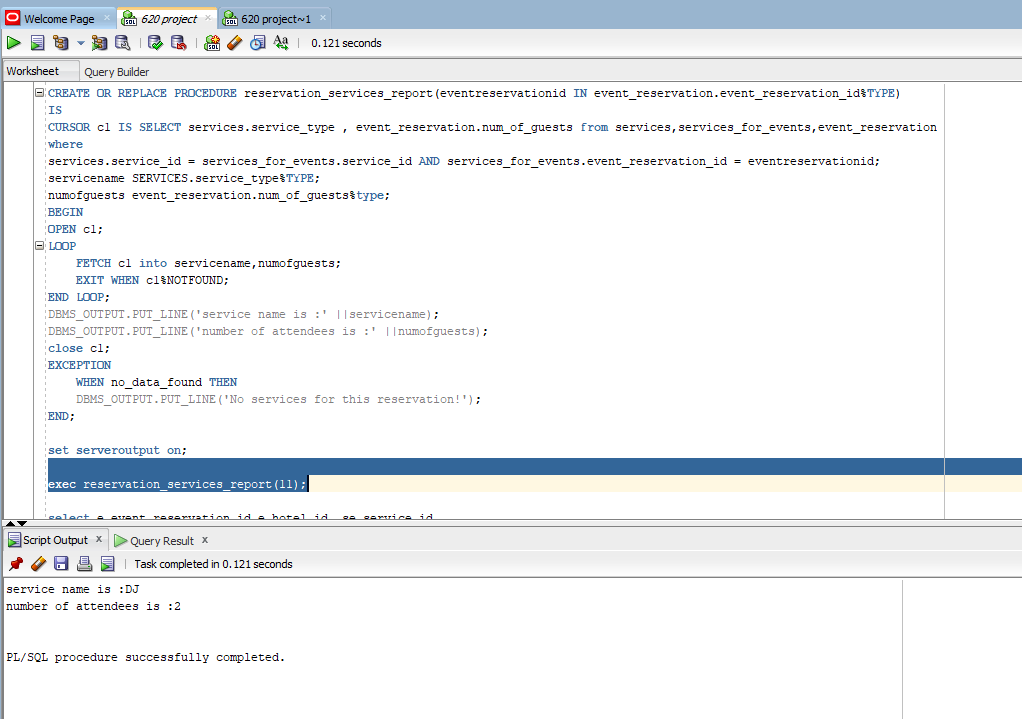
WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No services for this reservation!');

END;

--- Execution of the procedure ---

exec reservation\_services\_report(11);



Solution 19 :

---Procedure specific\_service\_report

---Displaying the reservation by inputing the service name

create or replace procedure specific\_service\_report(service\_name IN services.service\_type%TYPE)

IS

Cursor c1 is

select e.event\_reservation\_id,e.hotel\_id, se.service\_id

from event\_reservation e,services\_for\_events se, services s where e.event\_reservation\_id=se.event\_reservation\_id and s.service\_type = service\_name ;

--- @param reservation\_id

--- @param service\_id

--- @param hotel\_id

--- @param service\_type

--- Variables to handle the output cursor---

eventreservation event\_reservation.event\_reservation\_id%type;

hotelid event\_reservation.hotel\_id%type;

serviceid services.service\_id%type;

BEGIN

OPEN C1;

LOOP

FETCH c1 into eventreservation,hotelid,serviceid;

EXIT WHEN c1%NOTFOUND;

---Outputing the Event reservation,Hotel ID and Service id's after executing the procedure---

DBMS\_OUTPUT.PUT\_LINE('event reservation is :'||eventreservation);

DBMS\_OUTPUT.PUT\_LINE('hotelid is :'||hotelid);

DBMS\_OUTPUT.PUT\_LINE('servicesid is :'||serviceid);

END LOOP;

close c1;

---Handling the required exception ----

EXCEPTION

WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No reservations for this service');

END;

EXEC specific\_service\_report('DJ');

