UMBC
IS420 Database Application Development
Spring 2018
Group Project
Hotel Management System

Team Everest

Developed By:

Kabin Dulal
Diwakar Sharma
Sijan Karki

05/10/2017

Individual Tasks

Sijan Karki: Member 1 Plus 18 and 23

Numbers (1, 2, 3, 4, 5, 6, 18, 23)

Kabin Dulal: Member 2 Plus 16, 19 and 22

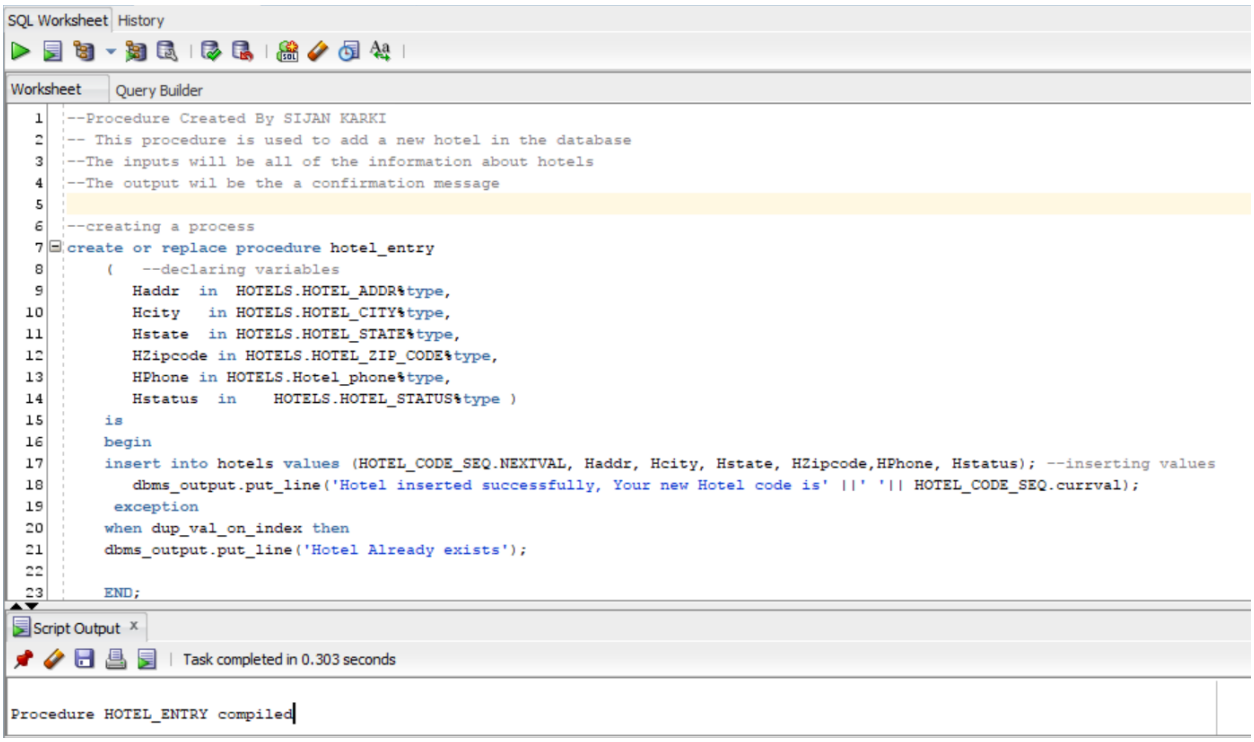
Numbers (7, 8, 9, 10, 16, 19, 22)

Diwakar Sharma: Member 3 Plus 17, 20 and 21

Numbers (11, 12, 12, 14, 15, 17, 20, 21)

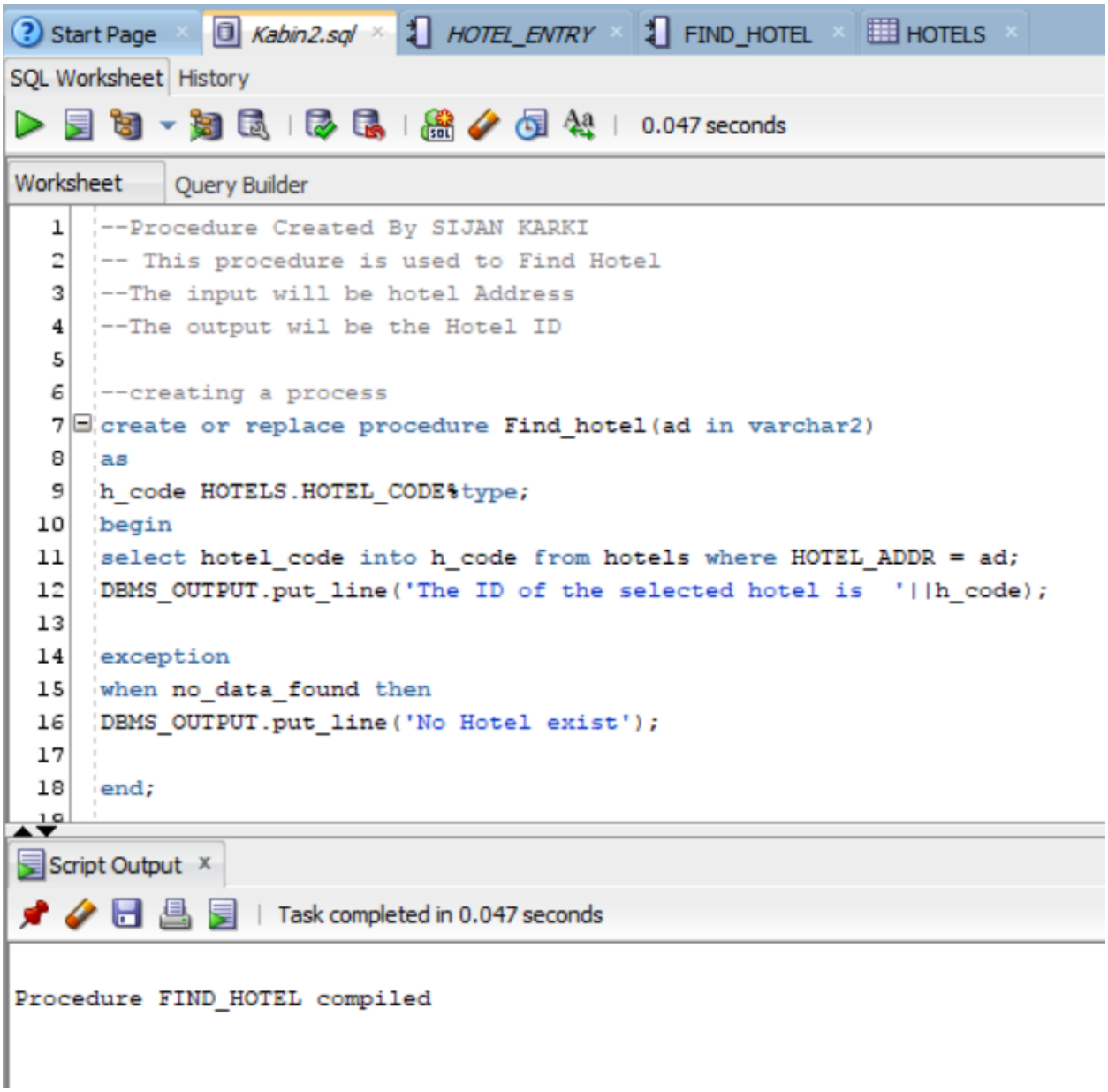
- Member 1: Sijan Karki
    1. Add a new hotel: Create a new hotel with appropriate information about the hotel as input parameters

```
--Procedure Created By SIJAN KARKI
-- This procedure is used to add a new hotel in the database
--The inputs will be all of the information about hotels
--The output wil be the a confirmation message

--creating a process
create or replace procedure hotel_entry
    (   --declaring variables
       Haddr  in  HOTELS.HOTEL_ADDR%type,
       Hcity   in HOTELS.HOTEL_CITY%type,
       Hstate  in HOTELS.HOTEL_STATE%type,
       HZipcode in HOTELS.HOTEL_ZIP_CODE%type,
       HPhone in HOTELS.Hotel_phone%type,
       Hstatus  in    HOTELS.HOTEL_STATUS%type )
    is
    begin
    insert into hotels values (HOTEL_CODE_SEQ.NEXTVAL, Haddr, Hcity, Hstate,
HZipcode,HPhone, Hstatus); --inserting values
       dbms_output.put_line('Hotel inserted successfully, Your new Hotel code
is' ||' '|| HOTEL_CODE_SEQ.currval);
     exception
    when dup_val_on_index then
    dbms_output.put_line('Hotel Already exists');

    END;
```

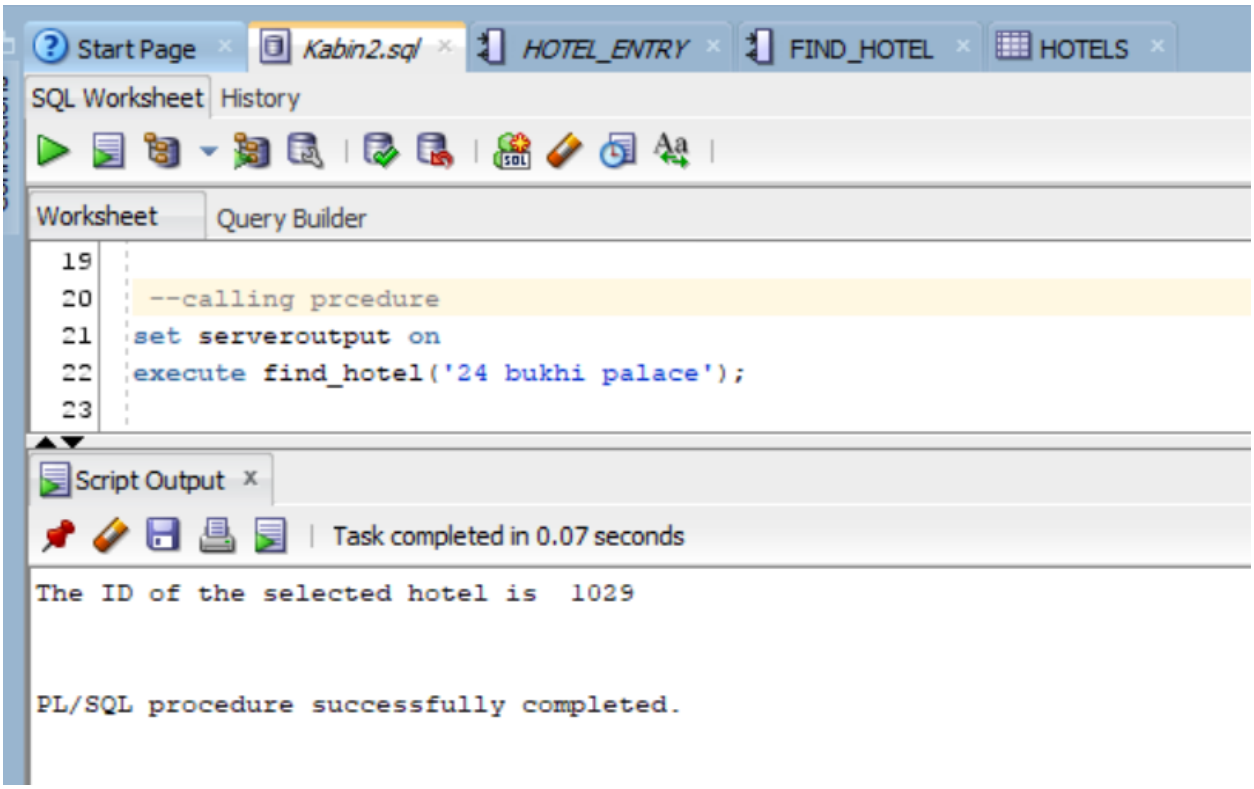


2. Find a hotel: Provide as input the address of the hotel and return its hotel ID

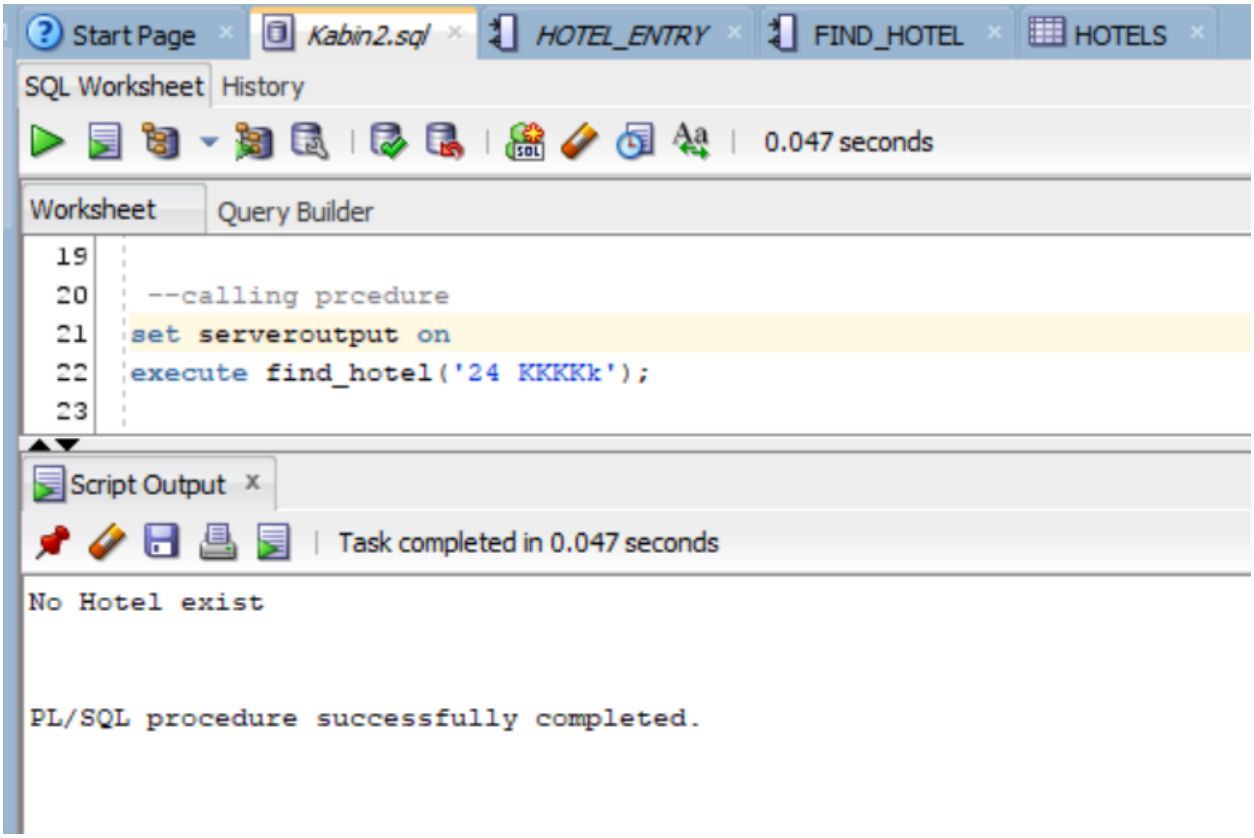
```
--Procedure Created By SIJAN KARKI
-- This procedure is used to Find Hotel
--The input will be hotel Address
--The output wil be the Hotel ID

--creating a process
create or replace procedure Find_hotel(ad in varchar2)
as
h_code HOTELS.HOTEL_CODE%type;
begin
select hotel_code into h_code from hotels where HOTEL_ADDR = ad;
DBMS_OUTPUT.put_line('The ID of the selected hotel is  '||h_code);

exception
when no_data_found then
DBMS_OUTPUT.put_line('No Hotel exist');

end;
```




```
--calling prcedure
set serveroutput on
execute find_hotel('24 bukhi palace');
```



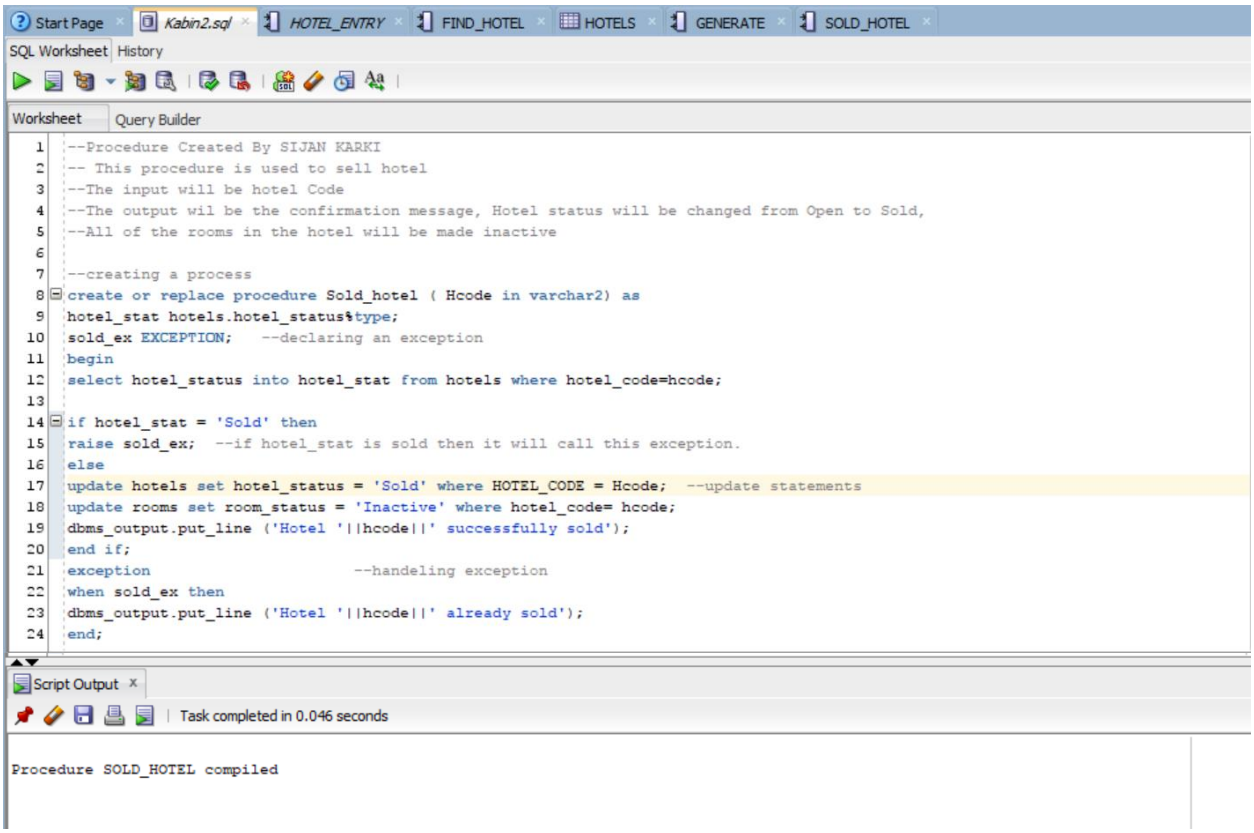

```
--calling prcedure
set serveroutput on
execute find_hotel('24 KKKKk');
```



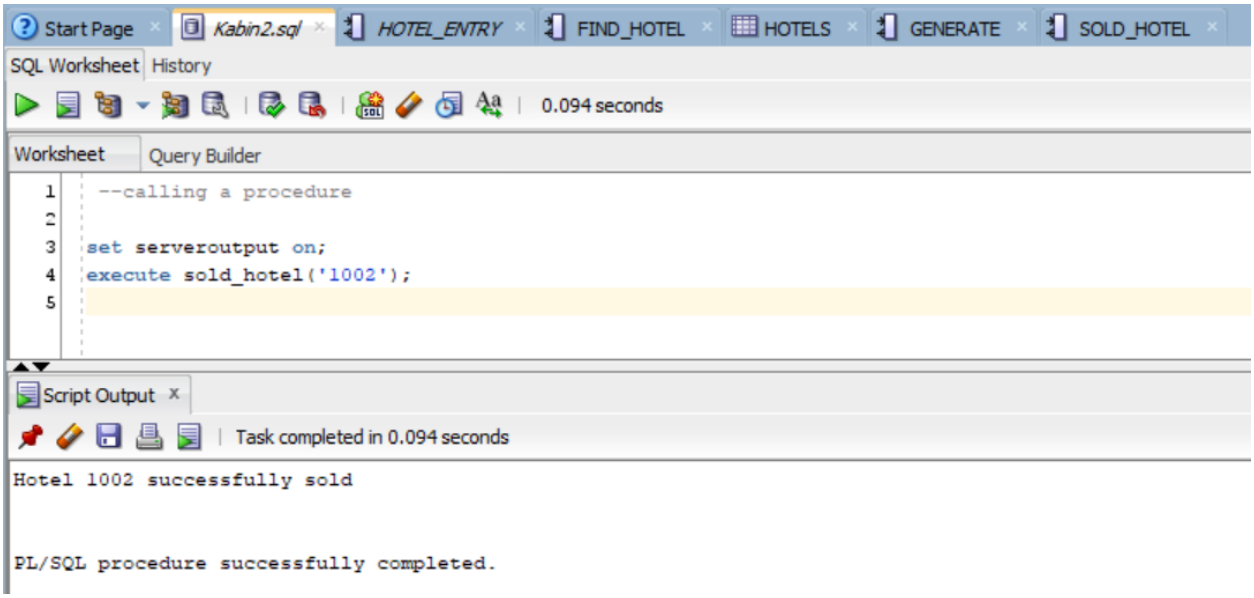
3. Sell existing hotel: Sell a hotel by providing its hotel ID. Mark it as sold, do not delete the record.

```
--Procedure Created By SIJAN KARKI
-- This procedure is used to sell hotel
--The input will be hotel Code
--The output wil be the confirmation message, Hotel status will be changed
from Open to Sold,
--All of the rooms in the hotel will be made inactive

--creating a process
create or replace procedure Sold_hotel ( Hcode in varchar2) as
hotel_stat hotels.hotel_status%type;
sold_ex EXCEPTION;   --declaring an exception
begin
select hotel_status into hotel_stat from hotels where hotel_code=hcode;

if hotel_stat = 'Sold' then
raise sold_ex;  --if hotel_stat is sold then it will call this exception.
else
update hotels set hotel_status = 'Sold' where HOTEL_CODE = Hcode;  --update
statements
update rooms set room_status = 'Inactive' where hotel_code= hcode;
dbms_output.put_line ('Hotel '||hcode||' successfully sold');
end if;
exception                       --handeling exception
```

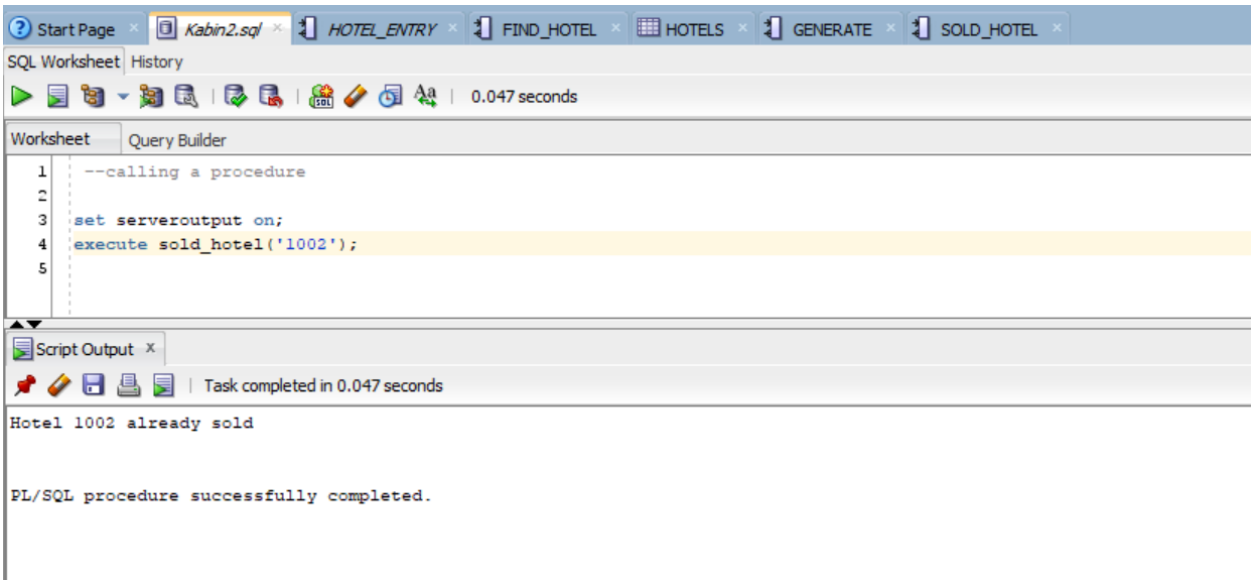
```
when sold_ex then
dbms_output.put_line ('Hotel '||hcode||' already sold');
end;
```




```
--calling a procedure

set serveroutput on;
execute sold_hotel('1002');
```



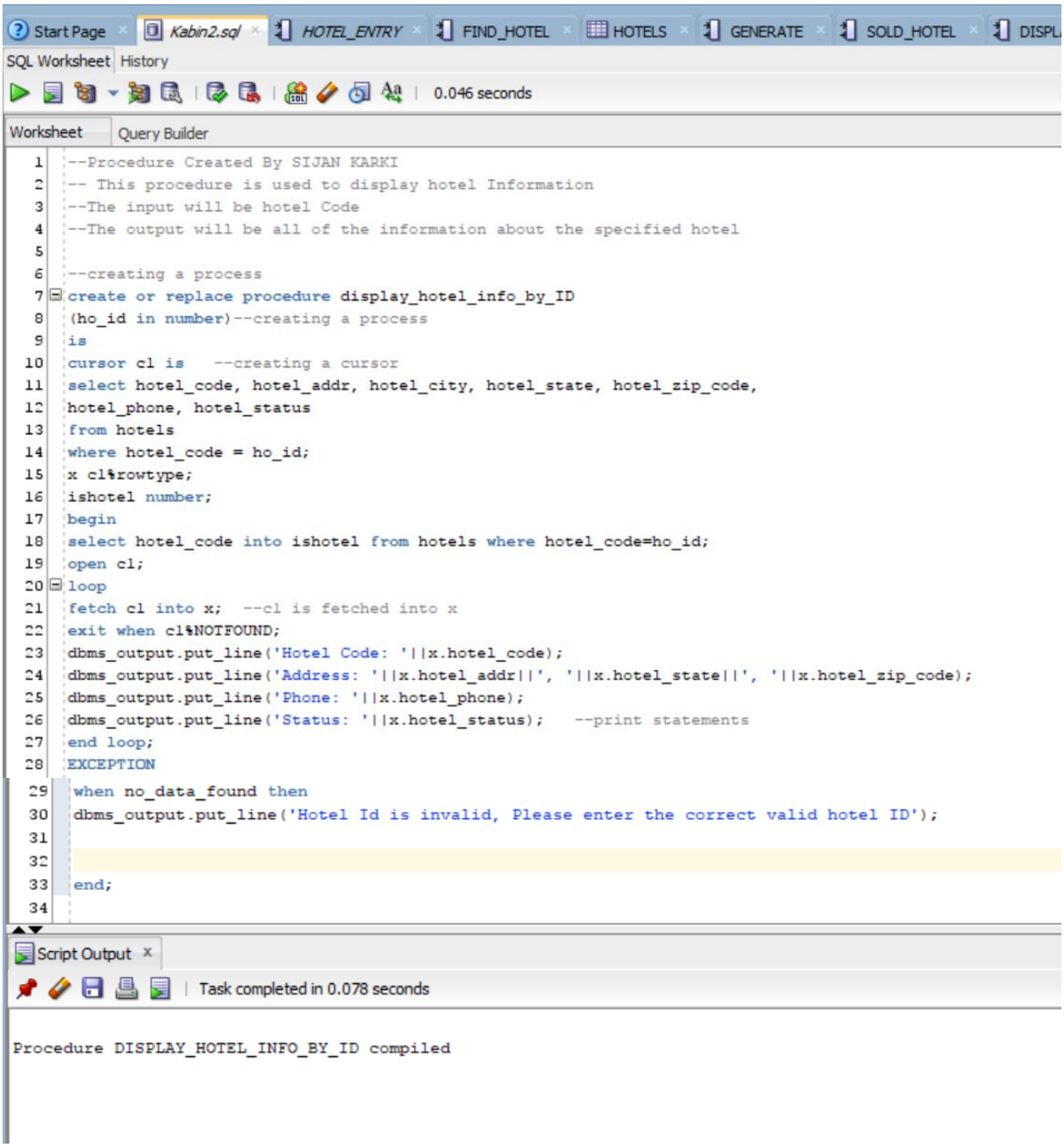

```
--calling a procedure

set serveroutput on;
execute sold_hotel('1002');
```



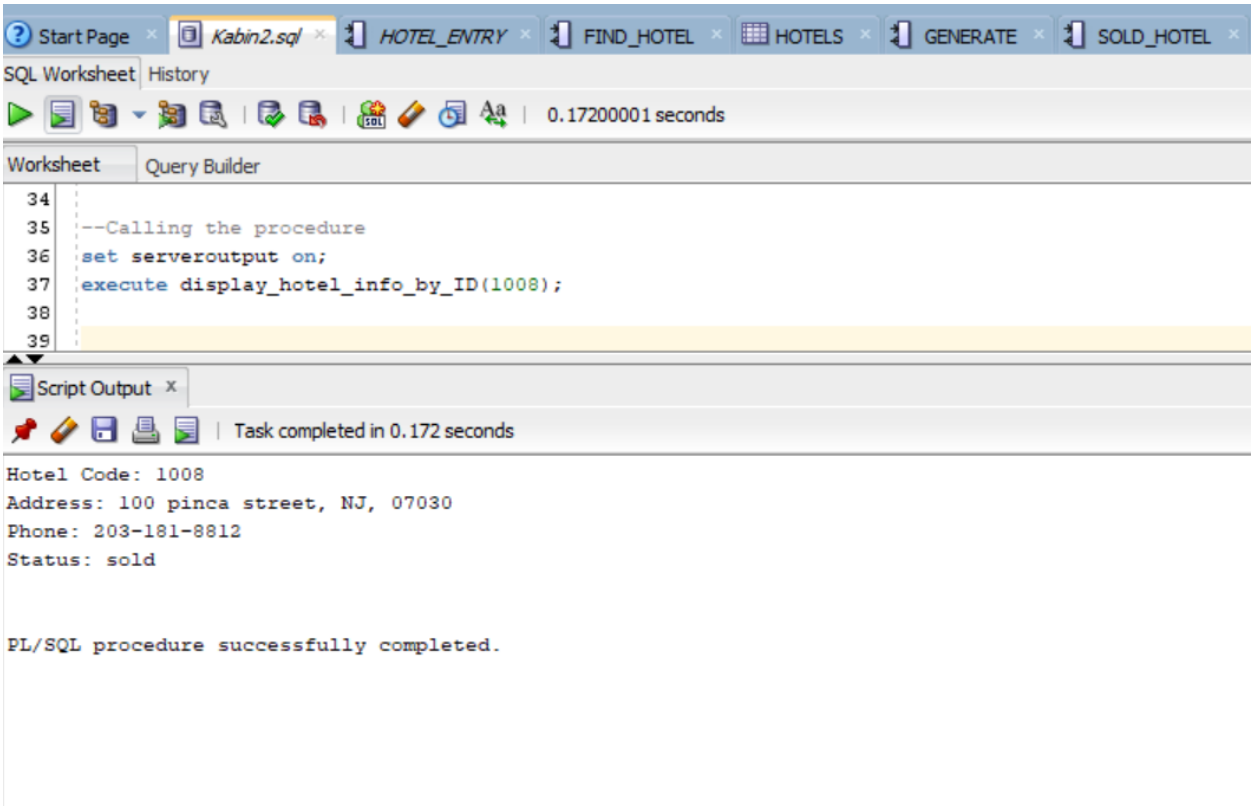
4. Display hotel info: Given a hotel ID, display all information about that hotel

```
--Procedure Created By SIJAN KARKI
-- This procedure is used to display hotel Information
--The input will be hotel Code
--The output will be all of the information about the specified hotel

--creating a process
create or replace procedure display_hotel_info_by_ID
(ho_id in number)--creating a process
```

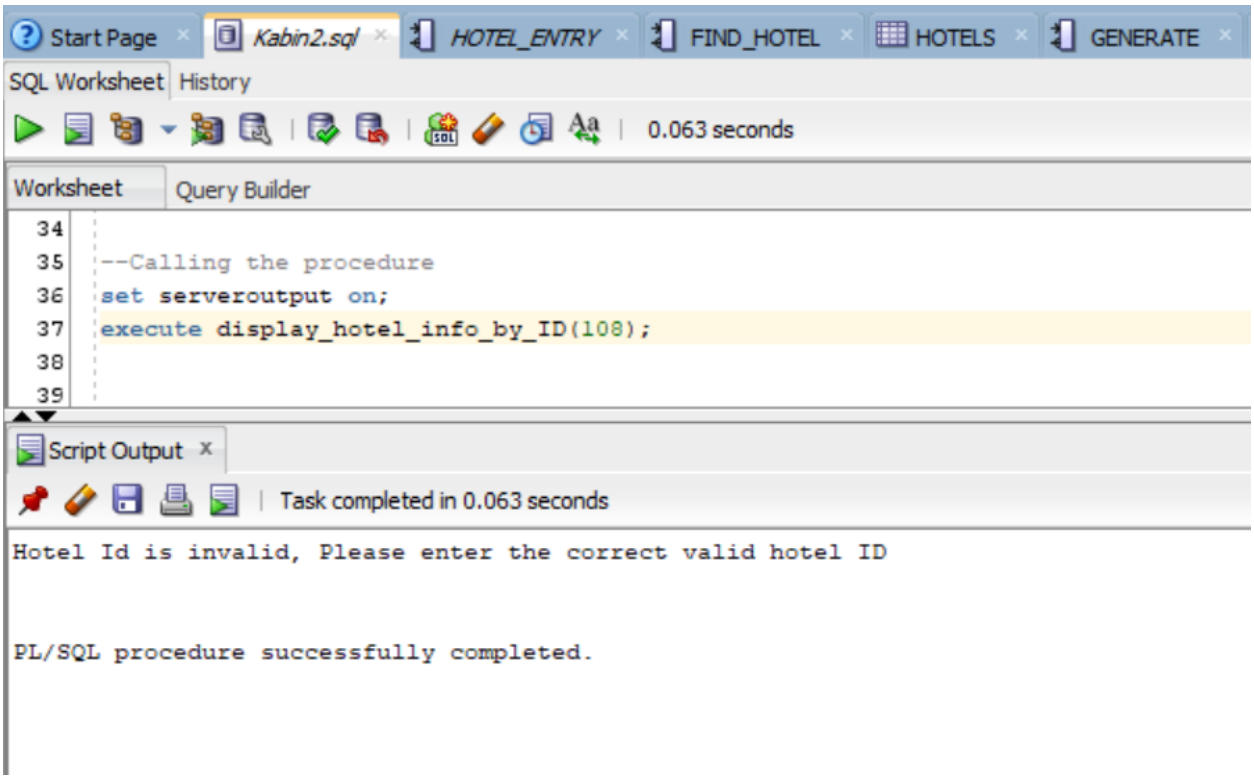
```
is
cursor c1 is   --creating a cursor
select hotel_code, hotel_addr, hotel_city, hotel_state, hotel_zip_code,
hotel_phone, hotel_status
from hotels
where hotel_code = ho_id;
x c1%rowtype;
ishotel number;
begin
select hotel_code into ishotel from hotels where hotel_code=ho_id;
open c1;
loop
fetch c1 into x;  --c1 is fetched into x
exit when c1%NOTFOUND;
dbms_output.put_line('Hotel Code: '||x.hotel_code);
dbms_output.put_line('Address: '||x.hotel_addr||', '||x.hotel_state||',
'||x.hotel_zip_code);
dbms_output.put_line('Phone: '||x.hotel_phone);
dbms_output.put_line('Status: '||x.hotel_status);   --print statements
end loop;
EXCEPTION
when no_data_found then
dbms_output.put_line('Hotel Id is invalid, Please enter the correct valid
hotel ID');


end;
```




```
set serveroutput on;
execute display_hotel_info_by_ID(1008);
```

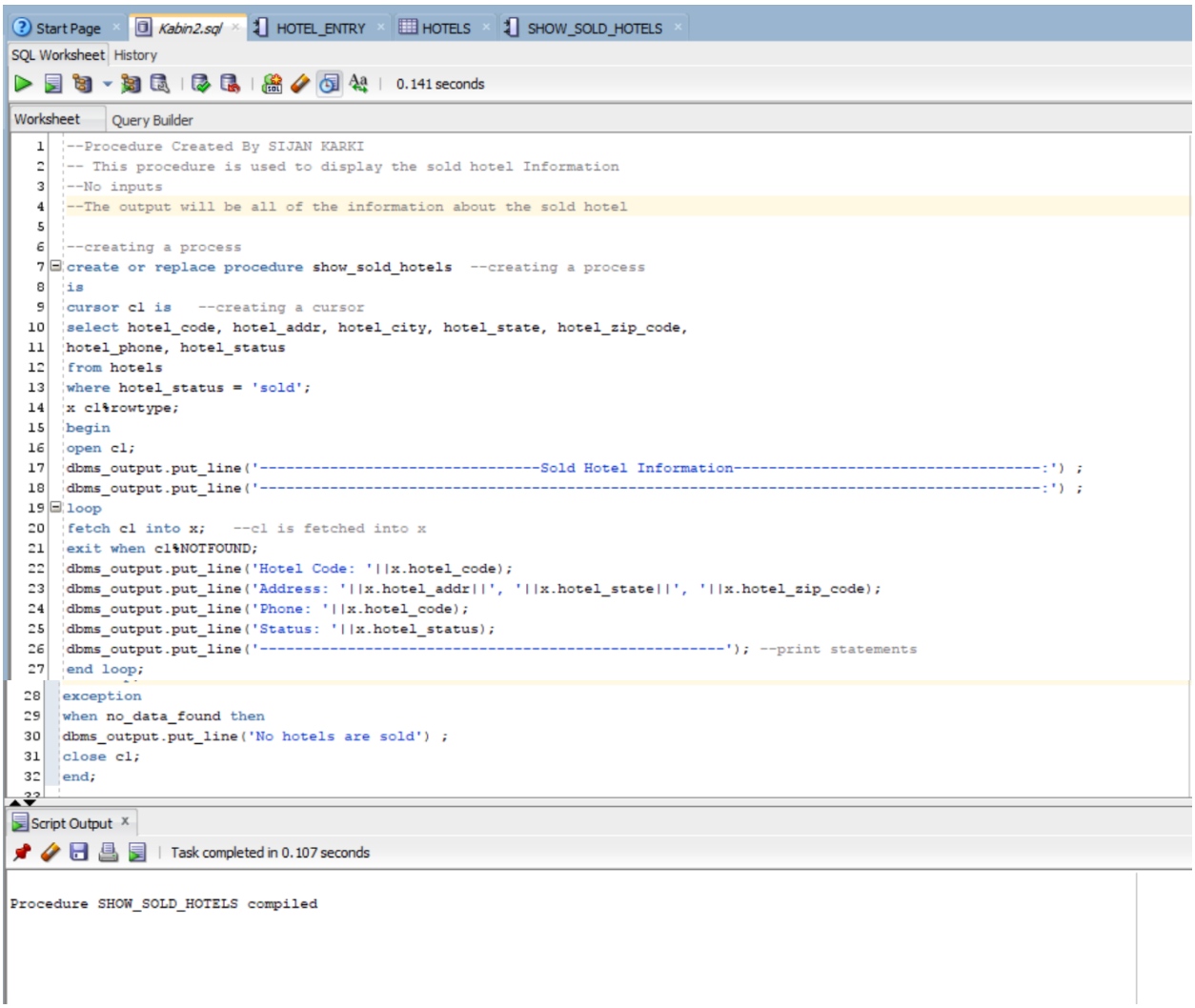



```
set serveroutput on;
execute display_hotel_info_by_ID(108);
```

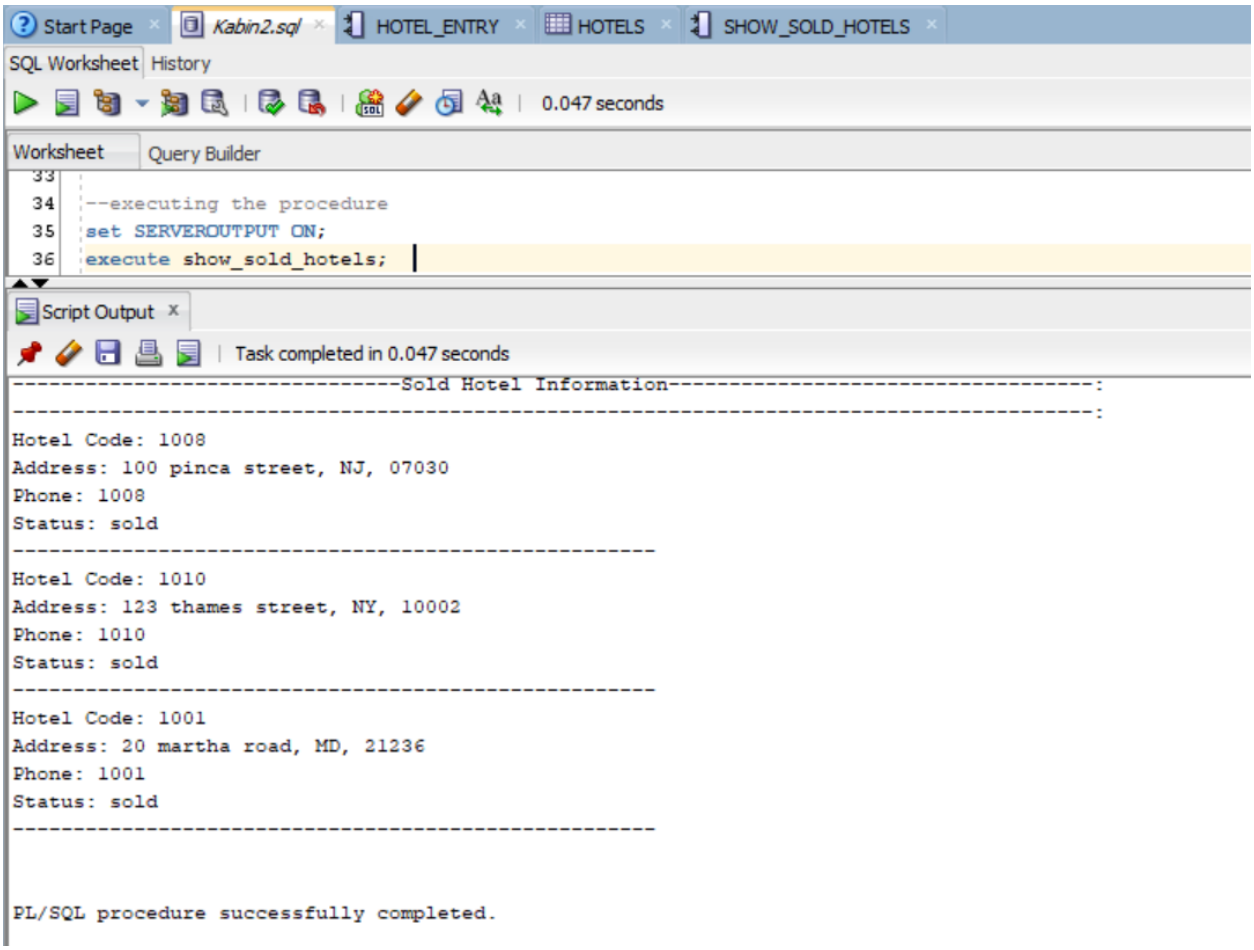


5. Sold Hotels: Print all sold hotel information. Show hotel ID, location, etc.

```
--Procedure Created By SIJAN KARKI
-- This procedure is used to display the sold hotel Information
--No inputs
--The output will be all of the information about the sold hotel

--creating a process
create or replace procedure show_sold_hotels  --creating a process
is
cursor c1 is   --creating a cursor
select hotel_code, hotel_addr, hotel_city, hotel_state, hotel_zip_code,
hotel_phone, hotel_status
from hotels
where hotel_status = 'sold';
x c1%rowtype;
begin
open c1;
dbms_output.put_line('--------------------------------Sold Hotel Information-
---------------------------------:') ;
dbms_output.put_line('------------------------------------------------------
---------------------------------:') ;
loop
fetch c1 into x;   --c1 is fetched into x
exit when c1%NOTFOUND;
dbms_output.put_line('Hotel Code: '||x.hotel_code);
dbms_output.put_line('Address: '||x.hotel_addr||', '||x.hotel_state||',
'||x.hotel_zip_code);
dbms_output.put_line('Phone: '||x.hotel_code);
dbms_output.put_line('Status: '||x.hotel_status);
dbms_output.put_line('------------------------------------------------------
'); --print statements
end loop;
exception
when no_data_found then
dbms_output.put_line('No hotels are sold') ;
close c1;
end;
```



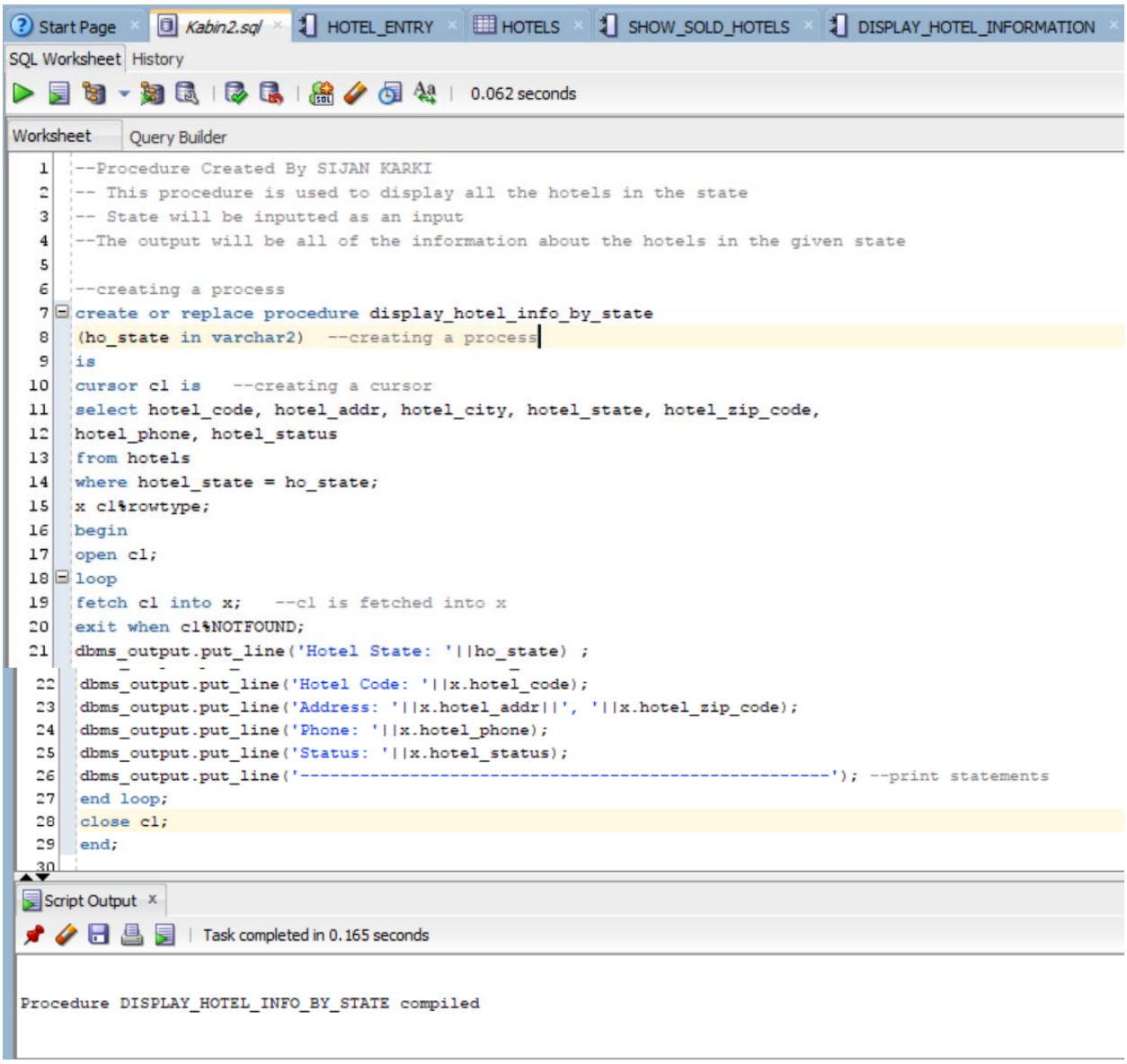

```
--executing the procedure
set SERVEROUTPUT ON;
execute show_sold_hotels;
```



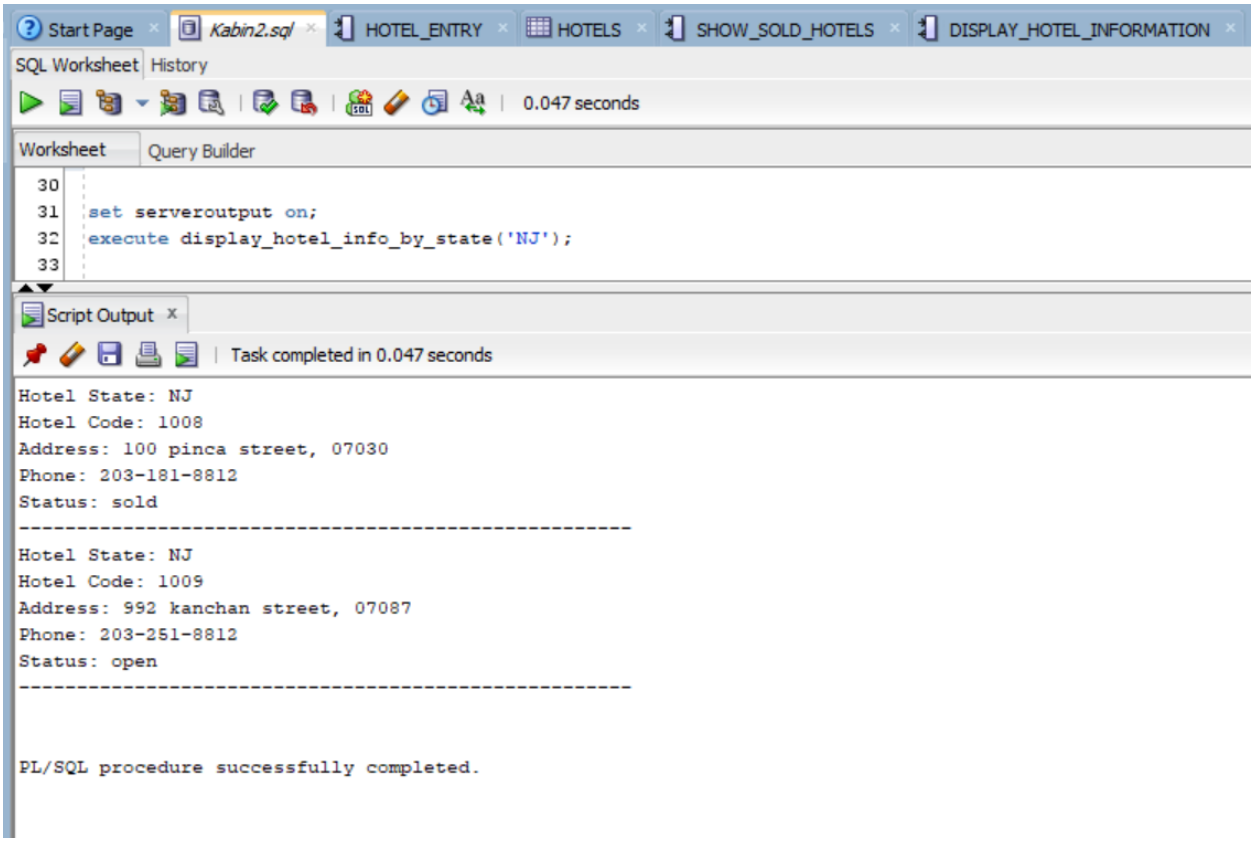
6. Report Hotels In State: Given a state, display information of all hotels in that particular state.

```
--Procedure Created By SIJAN KARKI
-- This procedure is used to display all the hotels in the state
-- State will be inputted as an input
--The output will be all of the information about the hotels in the given
state

--creating a process
create or replace procedure display_hotel_info_by_state
(ho_state in varchar2)  --creating a process
is
cursor c1 is   --creating a cursor
select hotel_code, hotel_addr, hotel_city, hotel_state, hotel_zip_code,
hotel_phone, hotel_status
from hotels
where hotel_state = ho_state;
x c1%rowtype;
begin
open c1;
loop
fetch c1 into x;   --c1 is fetched into x
exit when c1%NOTFOUND;
```

```
dbms_output.put_line('Hotel State: '||ho_state) ;
dbms_output.put_line('Hotel Code: '||x.hotel_code);
dbms_output.put_line('Address: '||x.hotel_addr||', '||x.hotel_zip_code);
dbms_output.put_line('Phone: '||x.hotel_phone);
dbms_output.put_line('Status: '||x.hotel_status);
dbms_output.put_line('--------------------------------------------------
'); --print statements
end loop;
close c1;
end;
```



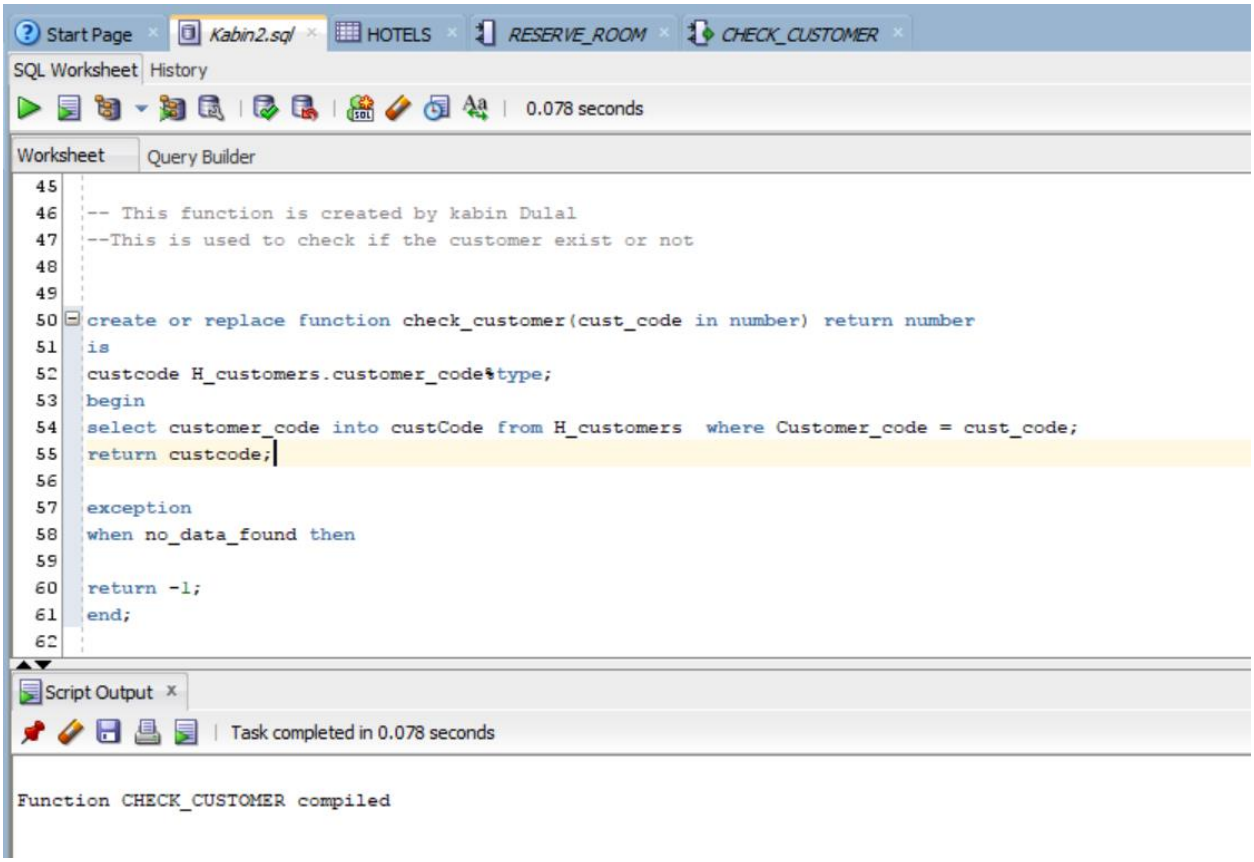

```
set serveroutput on;
execute display_hotel_info_by_state('NJ');
```



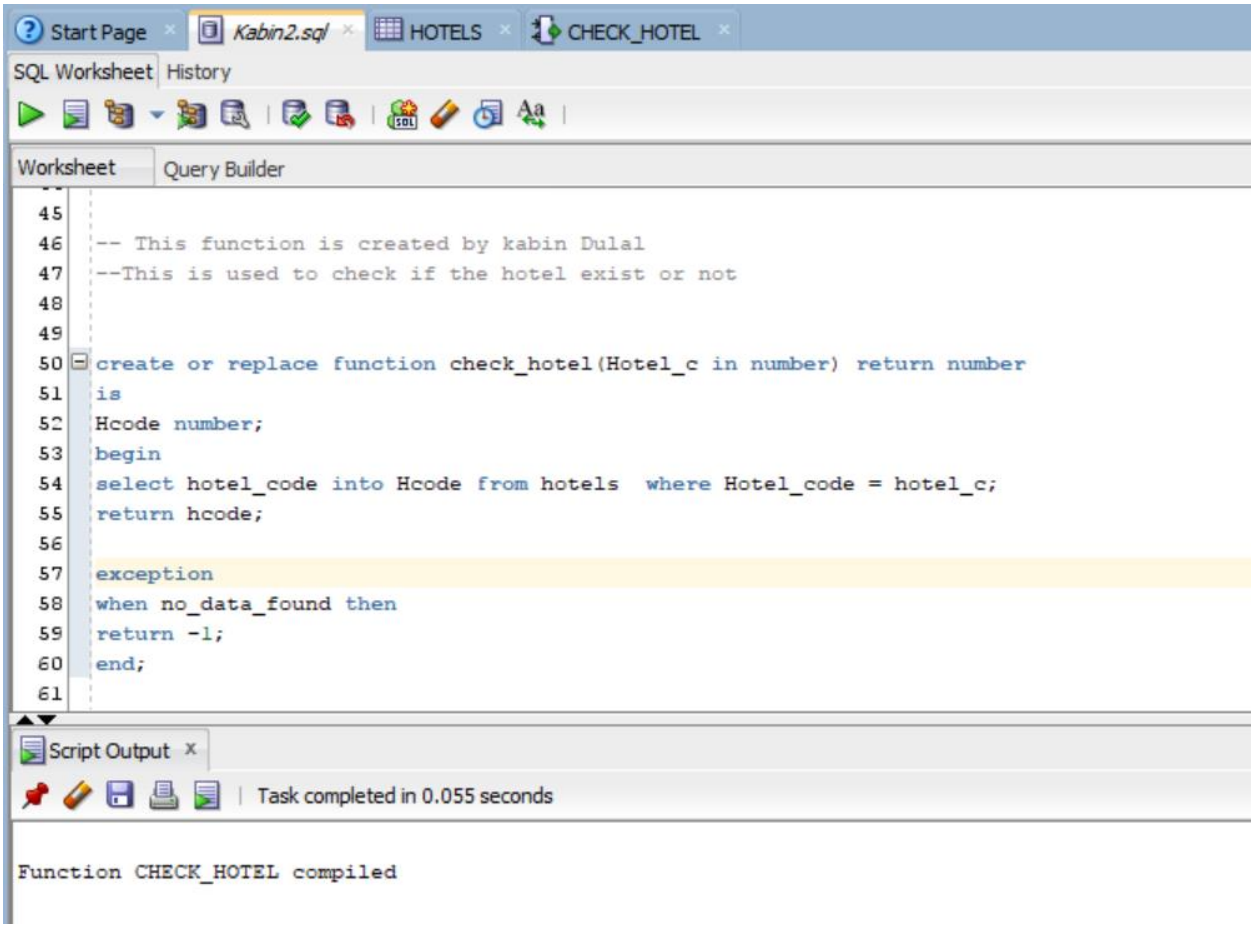
- Member 2: Kabin Dulal
    7. Make a reservation: Input parameters: Hotel ID, guest's name, start date, end date, room type, date of reservation, etc. Output: reservation ID (this is called confirmation code in real-life). NOTE: Only one guest per reservation. However, the same guest can make multiple reservations.

```
This function is created by kabin Dulal
--This is used to check if the customer exist or not


create or replace function check_customer(cust_code in number) return number
is
custcode H_customers.customer_code%type;
begin
select customer_code into custCode from H_customers  where Customer_code =
cust_code;
return custcode;

exception
when no_data_found then

return -1;
```

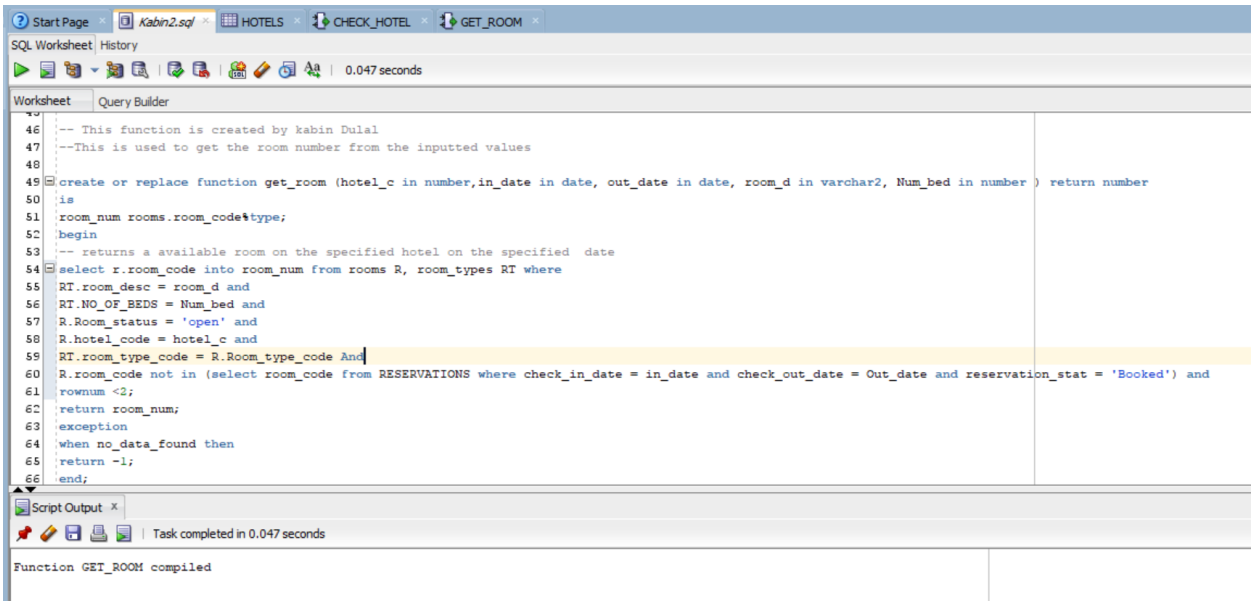
```
end;
```




```
-- This function is created by kabin Dulal
--This is used to check if the hotel exist or not


create or replace function check_hotel(Hotel_c in number) return number
is
Hcode number;
begin
select hotel_code into Hcode from hotels  where Hotel_code = hotel_c;
return hcode;

exception
when no_data_found then
return -1;
end;
```



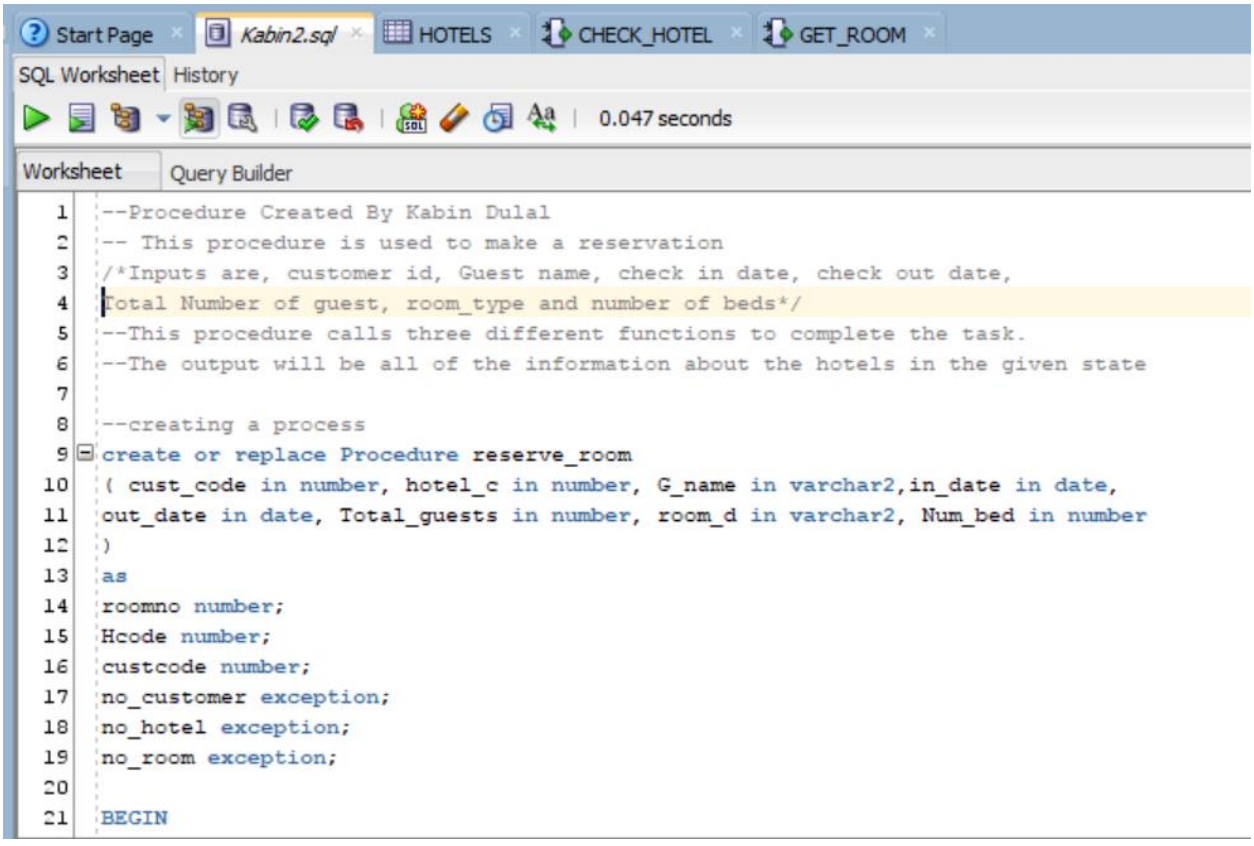

```
-- This function is created by kabin Dulal
--This is used to get the room number from the inputted values

create or replace function get_room (hotel_c in number,in_date in date,
out_date in date, room_d in varchar2, Num_bed in number ) return number
is
room_num rooms.room_code%type;
begin
-- returns a available room on the specified hotel on the specified  date
select r.room_code into room_num from rooms R, room_types RT where
RT.room_desc = room_d and
RT.NO_OF_BEDS = Num_bed and
R.Room_status = 'open' and
R.hotel_code = hotel_c and
RT.room_type_code = R.Room_type_code And
R.room_code not in (select room_code from RESERVATIONS where check_in_date =
in_date and check_out_date = Out_date and reservation_stat = 'Booked') and
rownum <2;
return room_num;
exception
when no_data_found then
return -1;
end;
```



-----The Following procedure that calls all the above function and makes a reservation

```
--Procedure Created By Kabin Dulal
-- This procedure is used to make a reservation
/*Inputs are, customer id, Guest name, check in date, check out date,
Total Number of guest, room_type and number of beds*/
--This procedure calls three different functions to complete the task.
--The output will Reservation number or any error message

--creating a process
create or replace Procedure reserve_room
( cust_code in number, hotel_c in number, G_name in varchar2,in_date in date,
out_date in date, Total_guests in number, room_d in varchar2, Num_bed in
number
)
as
roomno number;
Hcode number;
custcode number;
no_customer exception;
no_hotel exception;
no_room exception;

BEGIN
Hcode:=  check_hotel(hotel_c);--calling  Check_hotel function
Custcode:= Check_customer (cust_code);--calling  check_customer function
roomno:= get_room(Hotel_c, in_date, out_date, room_d, Num_bed);-- Gets rom
number and set to RoomNo
--check if anything went wrong and prints what went wrong
if hcode<1 then
dbms_output.put_line ('Hotel '||Hotel_c||' Does not Exist');
elsif Custcode<1 then
```

```
dbms_output.put_line ('The customer ID is invalid, Please Enter the corrrect
Customer ID');
dbms_output.put_line ('If You are New Customer, Please Sign UP');
elsif roomno<1 then
dbms_output.put_line ('There are no room available of this type in this
hotel, Please Change your Room types or look for a new hotel');
else
Insert into
RESERVATIONS(RESERVATION_CODE,GUEST_NAME,CHECK_IN_DATE,CHECK_OUT_DATE,RESERVA
TION_DATE,
NO_OF_GUESTS,CUSTOMER_CODE,HOTEL_CODE,ROOM_CODE,RESERVATION_STAT)
values
(RESERVE_SEQ.NEXTVAL,G_name,TO_DATE(in_date,'DD-MON-
RR'),To_date(Out_date,'DD-MON-RR'),
To_date(sysdate,'DD-MON-RR'),total_guests,custcode,hcode,Roomno,'Booked');

DBMS_output.put_line ('Congratulations, your reservation is booked,');
DBMS_OUTPUT.PUT_LINE('Your reservation code is: ' || RESERVE_SEQ.CURRVAL);

end if;

END;
```



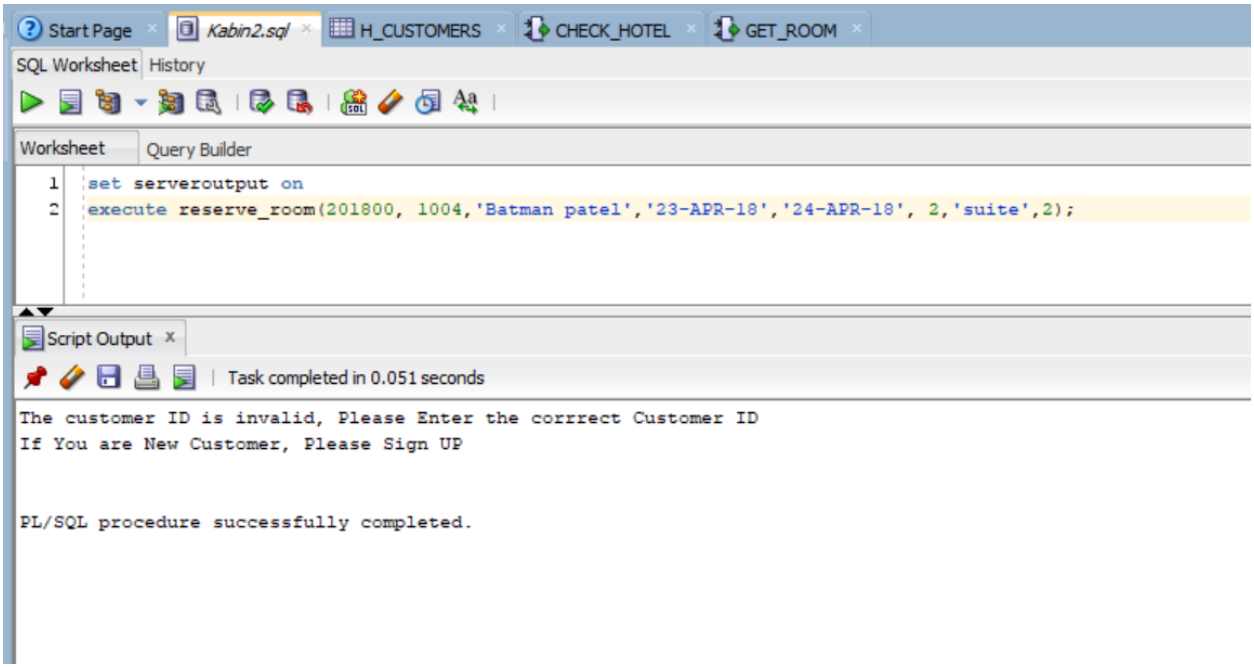
```
Hcode:=  check_hotel(hotel_c);--calling  Check_hotel function
Custcode:= Check_customer (cust_code);--calling  check_customer function
roomno:= get_room(Hotel_c, in_date, out_date, room_d, Num_bed);-- Gets rom number and set to RoomNo
--check if anything went wrong and prints what went wrong
if hcode<1 then
dbms_output.put_line ('Hotel '||Hotel_c||' Does not Exist');
elsif Custcode<1 then
dbms_output.put_line ('The customer ID is invalid, Please Enter the corrrect Customer ID');
dbms_output.put_line ('If You are New Customer, Please Sign UP');
elsif roomno<1 then
dbms_output.put_line ('There are no room available of this type in this hotel, Please Change your Room types or look for a new hotel');
else
Insert into RESERVATIONS(RESERVATION_CODE,GUEST_NAME,CHECK_IN_DATE,CHECK_OUT_DATE,RESERVATION_DATE,
NO_OF_GUESTS,CUSTOMER_CODE,HOTEL_CODE,ROOM_CODE,RESERVATION_STAT)
values
(RESERVE_SEQ.NEXTVAL,G_name,TO_DATE(in_date,'DD-MON-RR'),To_date(Out_date,'DD-MON-RR'),
To_date(sysdate,'DD-MON-RR'),total_guests,custcode,hcode,Roomno,'Booked');

DBMS_output.put_line ('Congratulations, your reservation is booked,');
DBMS_OUTPUT.PUT_LINE('Your reservation code is: ' || RESERVE_SEQ.CURRVAL);

end if;

END;
```

```
Script Output
Task completed in 0.251 seconds

Procedure RESERVE_ROOM compiled
```

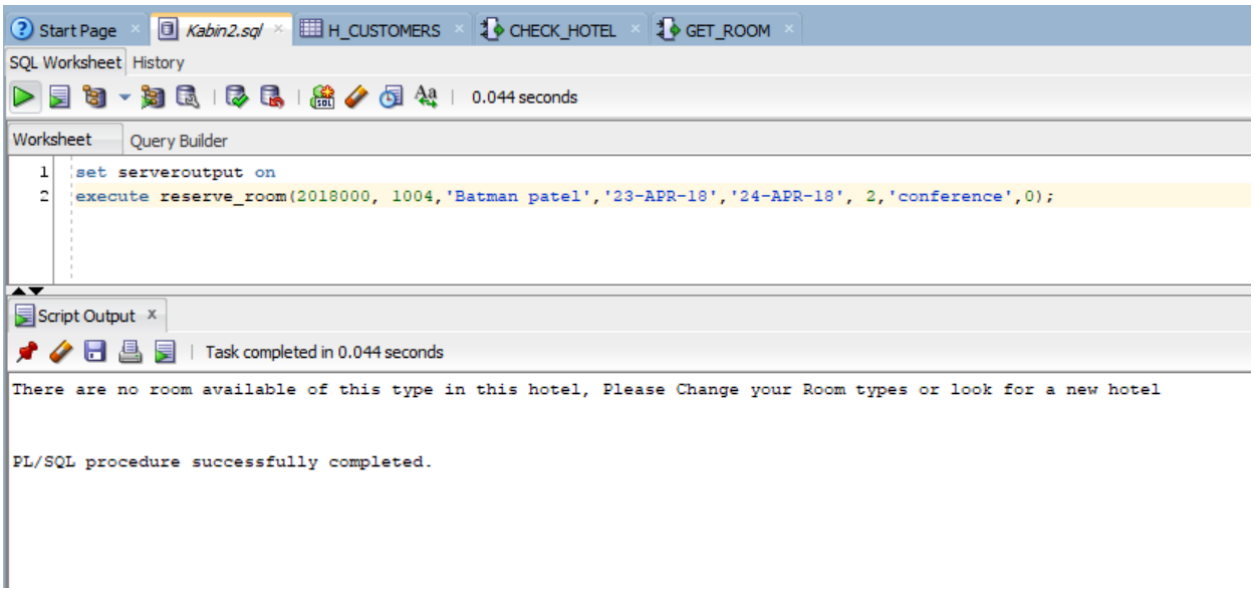
```
set serveroutput on
execute reserve_room(2018000, 1004,'Batman patel','23-APR-18','24-APR-18',
2,'suite',2);
```

```
Start Page | Kabin2.sql | H_CUSTOMERS | CHECK_HOTEL | GET_ROOM
SQL Worksheet | History
Worksheet | Query Builder
1  set serveroutput on
2  execute reserve_room(2018000, 1004,'Batman patel','23-APR-18','24-APR-18', 2,'suite',2);

Script Output
Task completed in 0.107 seconds

Congratulations, your reservation is booked,
Your reservation code is: 100043


PL/SQL procedure successfully completed.
```

```
set serveroutput on
execute reserve_room(201800, 1004,'Batman patel','23-APR-18','24-APR-18',
2,'suite',2);
```



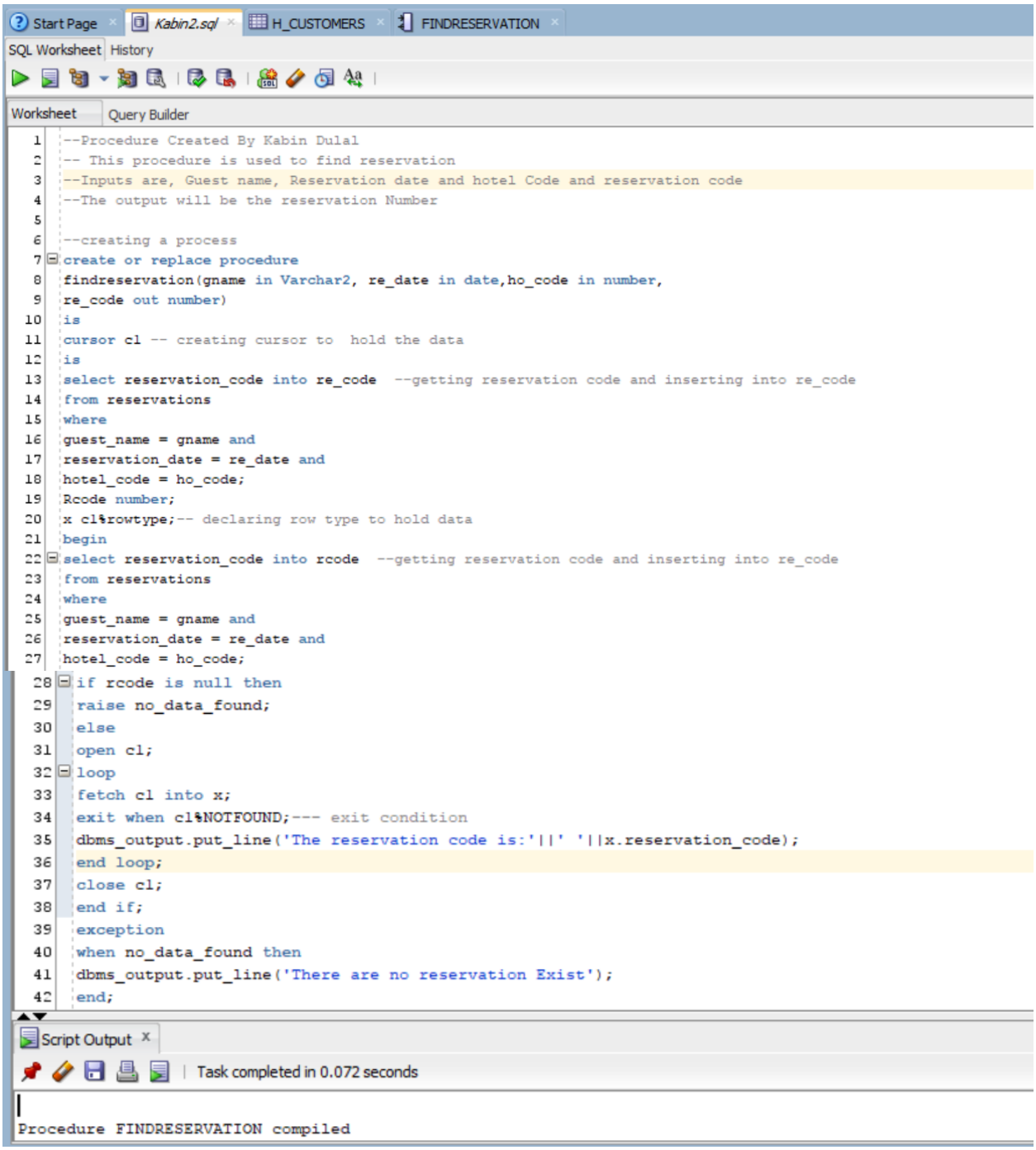

```
set serveroutput on
execute reserve_room(2018000, 1004,'Batman patel','23-APR-18','24-APR-18',
2,'conference',0);
```



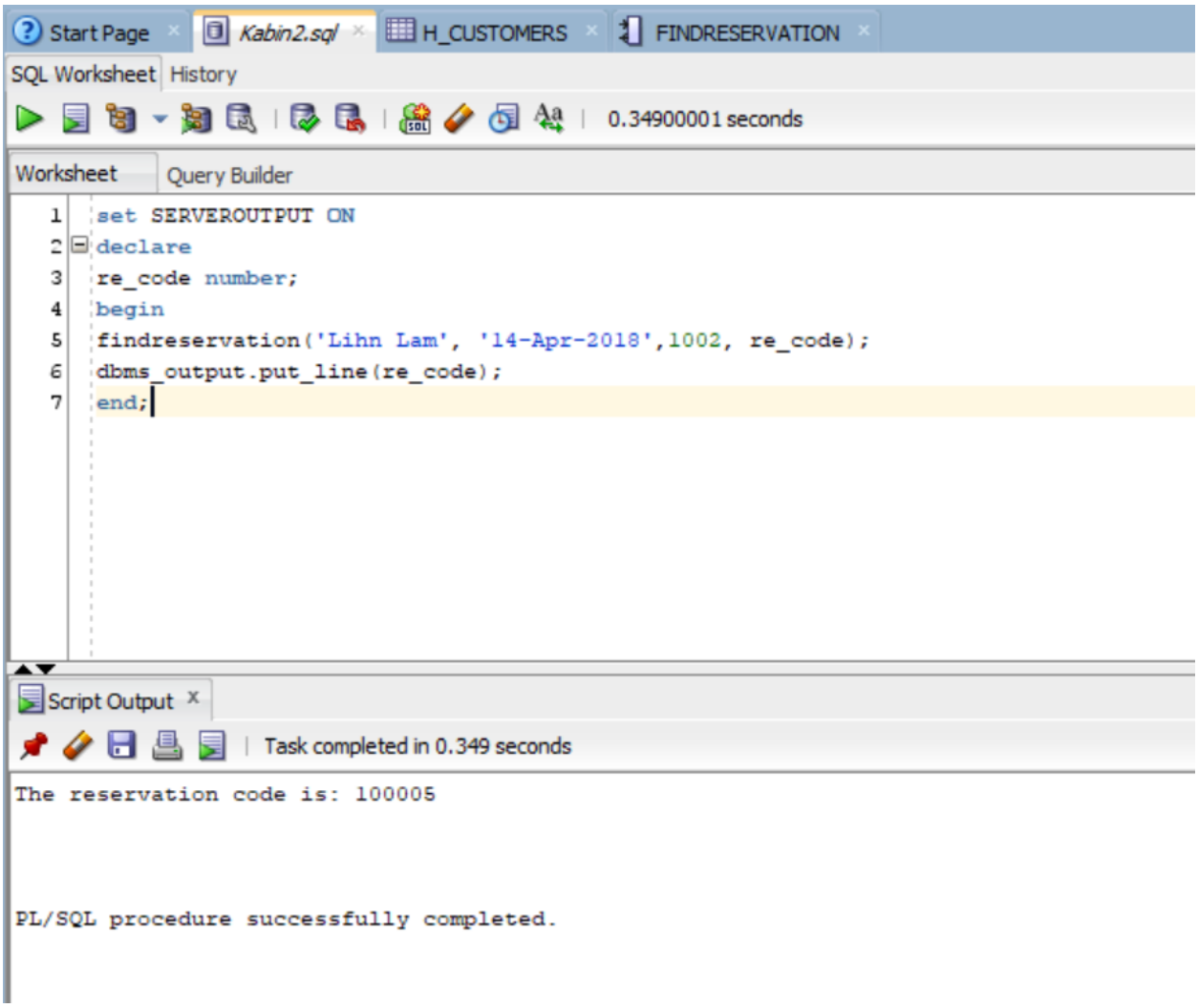
8. Find a reservation: Input is guest's name and date, hotel ID. Output is reservation ID

```
--Procedure Created By Kabin Dulal
-- This procedure is used to find reservation
--Inputs are, Guest name, Reservation date and hotel Code and reservation
code
```

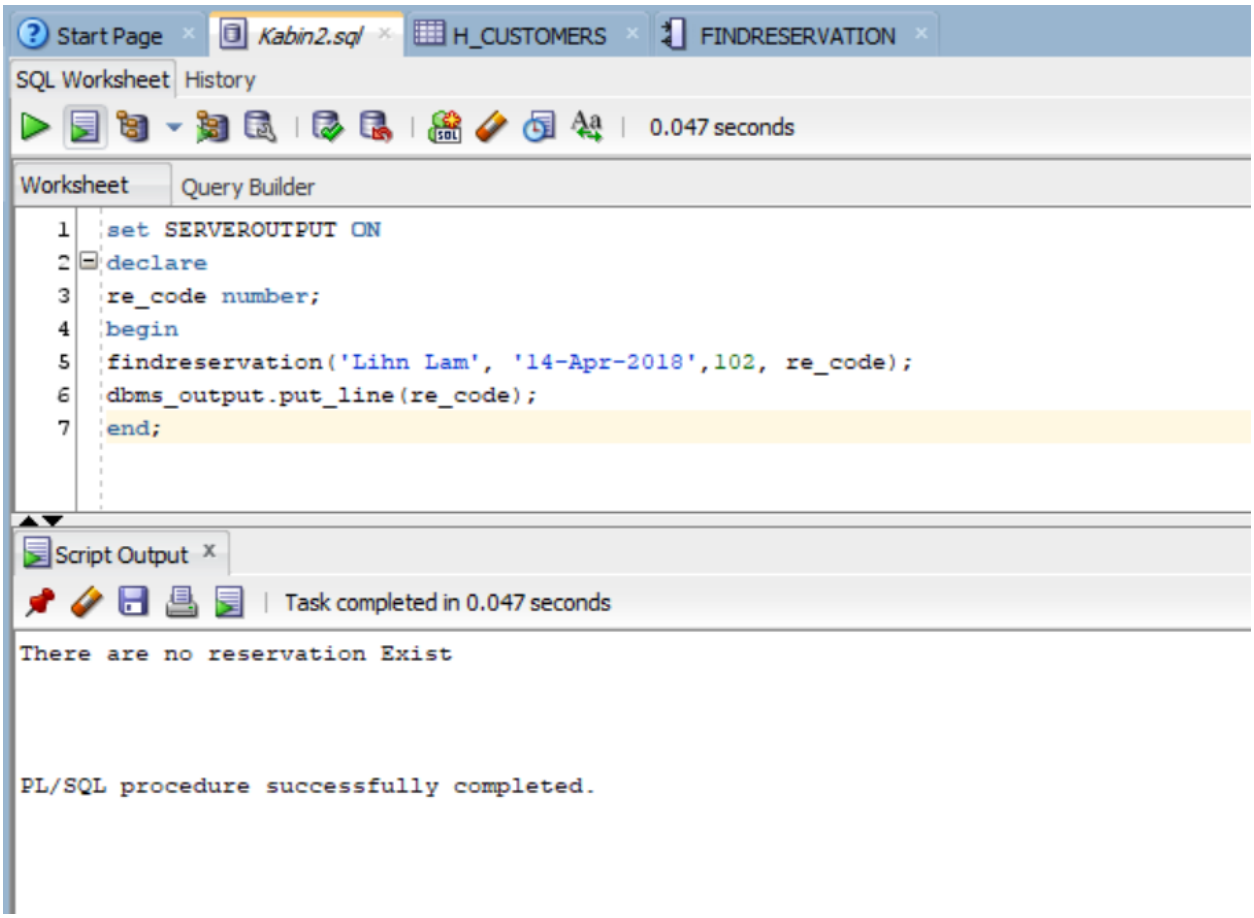
```
--The output will be the reservation Number

--creating a process
create or replace procedure
findreservation(gname in Varchar2, re_date in date,ho_code in number,
re_code out number)
is
cursor c1 -- creating cursor to  hold the data
is
select reservation_code into re_code  --getting reservation code and
inserting into re_code
from reservations
where
guest_name = gname and
reservation_date = re_date and
hotel_code = ho_code;
Rcode number;
x c1%rowtype;-- declaring row type to hold data
begin
select reservation_code into rcode  --getting reservation code and inserting
into re_code
from reservations
where
guest_name = gname and
reservation_date = re_date and
hotel_code = ho_code;
if rcode is null then
raise no_data_found;
else
open c1;
loop
fetch c1 into x;
exit when c1%NOTFOUND;--- exit condition
dbms_output.put_line('The reservation code is:'||' '||x.reservation_code);
end loop;
close c1;
end if;
exception
when no_data_found then
dbms_output.put_line('There are no reservation Exist');
end;
```




```
set SERVEROUTPUT ON
declare
re_code number;
begin
findreservation('Lihn Lam', '14-Apr-2018',1002, re_code);
dbms_output.put_line(re_code);
end;
```



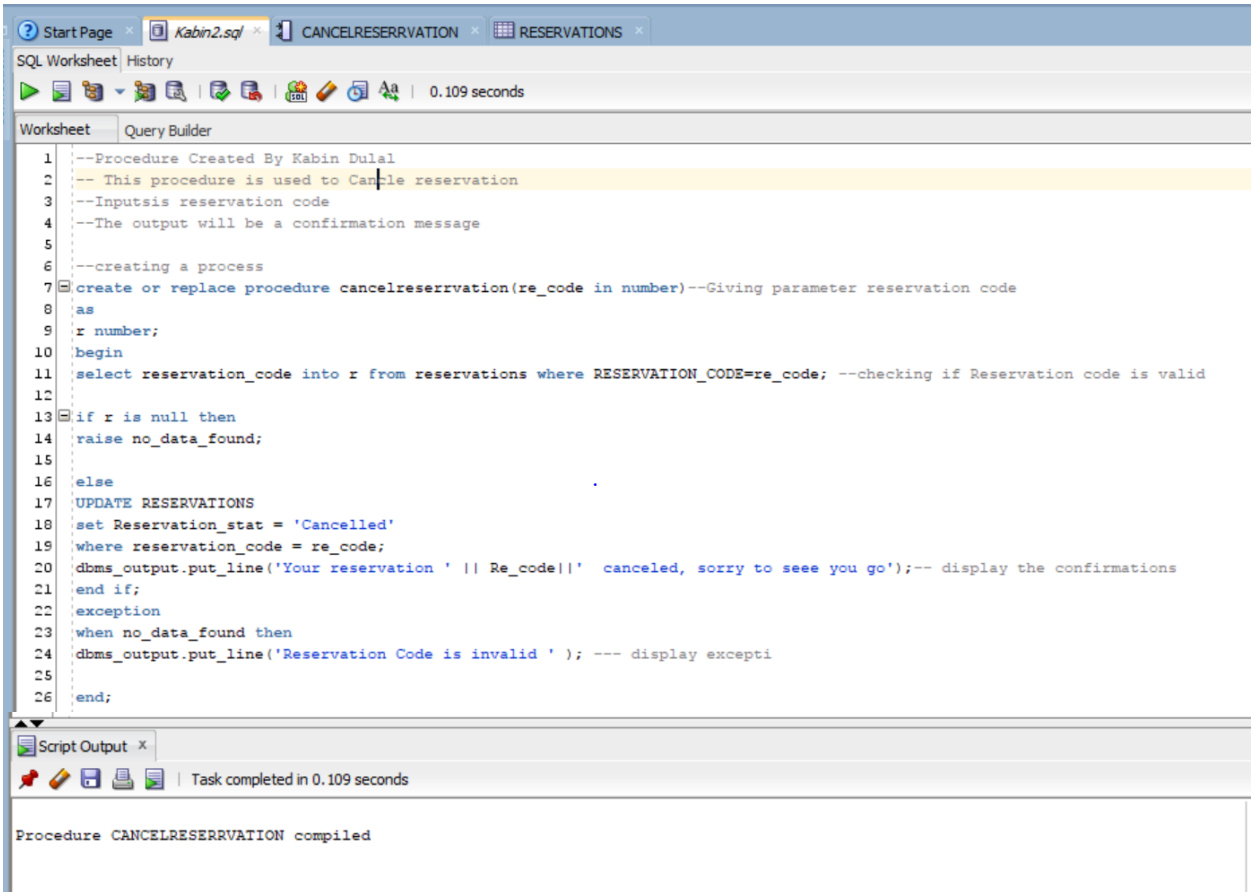

```
set SERVEROUTPUT ON
declare
re_code number;
begin
findreservation('Lihn Lam', '14-Apr-2018',102, re_code);
dbms_output.put_line(re_code);
end;
```



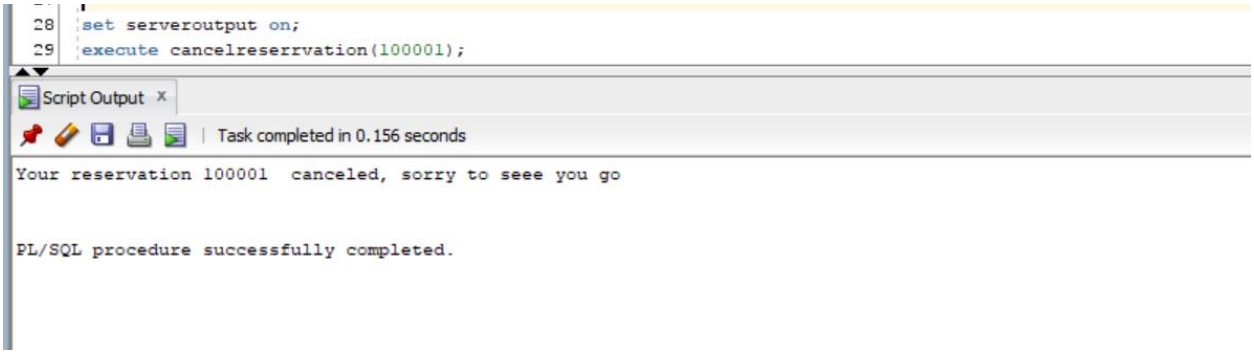
9. Cancel a reservation: Input the reservationID and mark the reservation as cancelled (do NOT delete it)

```
--Procedure Created By Kabin Dulal
-- This procedure is used to Cancle reservation
--Inputsis reservation code
--The output will be a confirmation message

--creating a process
create or replace procedure cancelreserrvation(re_code in number)--Giving
parameter reservation code
as
r number;
begin
select reservation_code into r from reservations where
RESERVATION_CODE=re_code; --checking if Reservation code is valid

if r is null then
raise no_data_found;

else
UPDATE RESERVATIONS
set Reservation_stat = 'Cancelled'
where reservation_code = re_code;
```

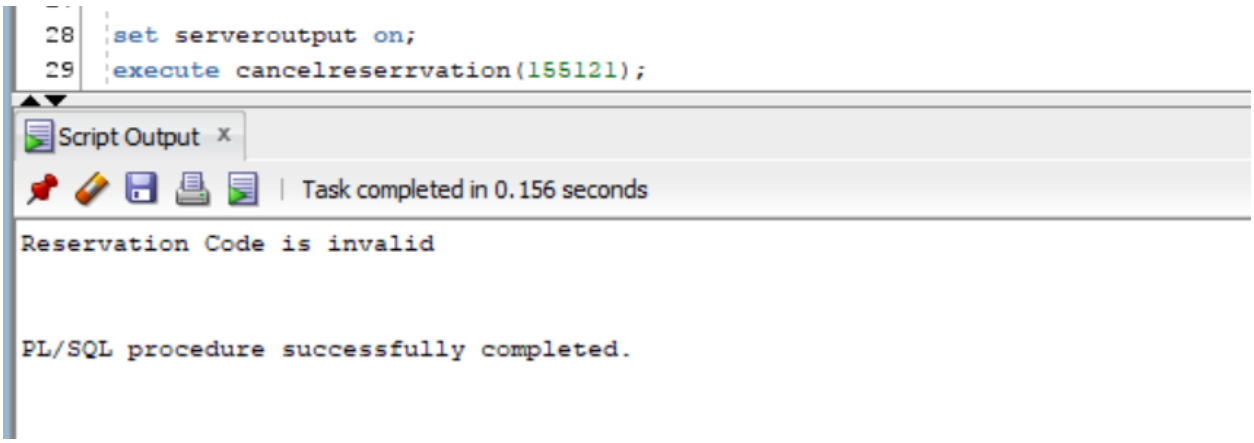
```
dbms_output.put_line('Your reservation ' || Re_code||'  canceled, sorry to
seee you go');-- display the confirmations
end if;
exception
when no_data_found then
dbms_output.put_line('Reservation Code is invalid ' ); --- display excepti

end;
```




```
set serveroutput on;
execute cancelreserrvation(100001);
```



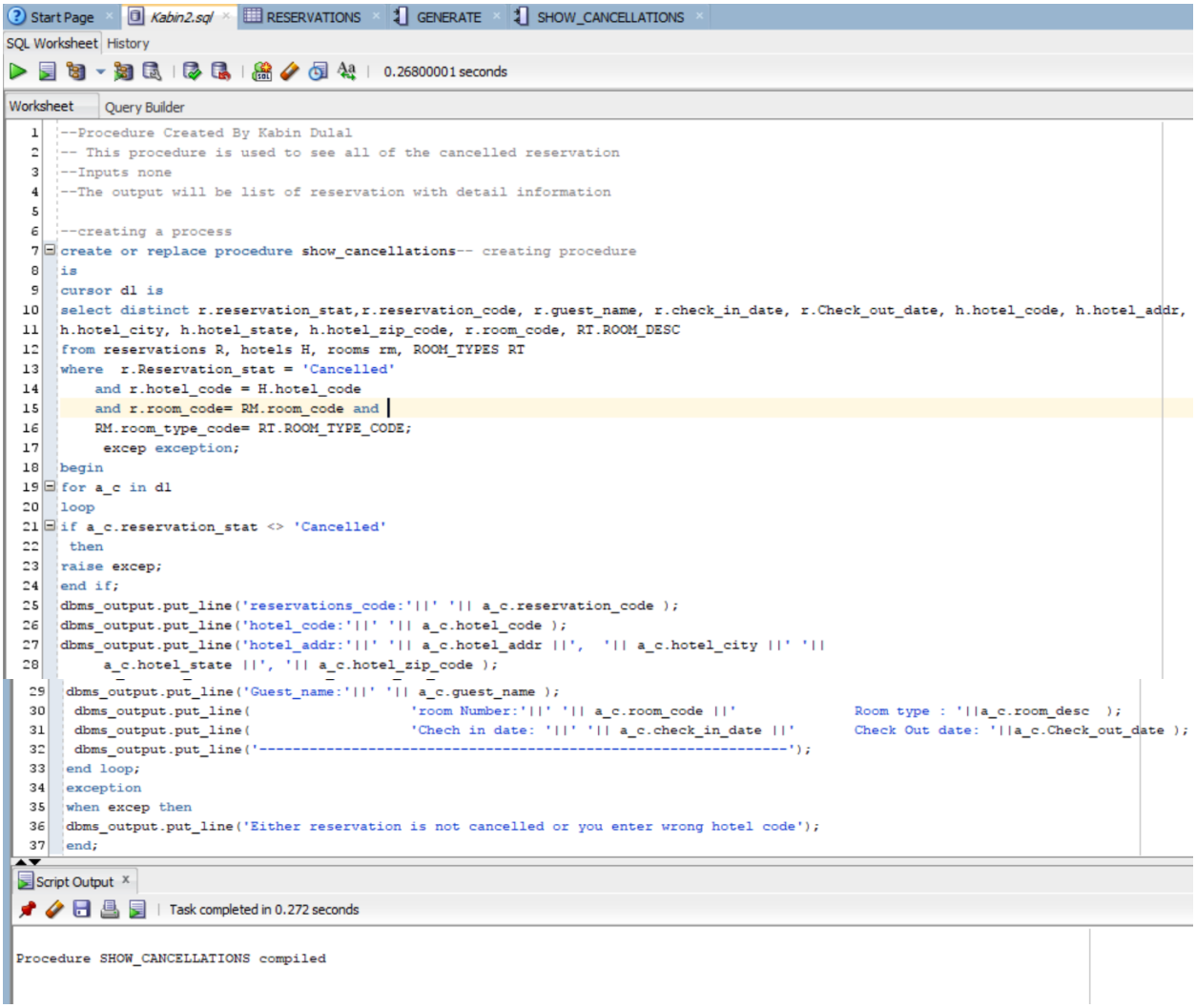
```
set serveroutput on;
execute cancelreserrvation(155121);
```



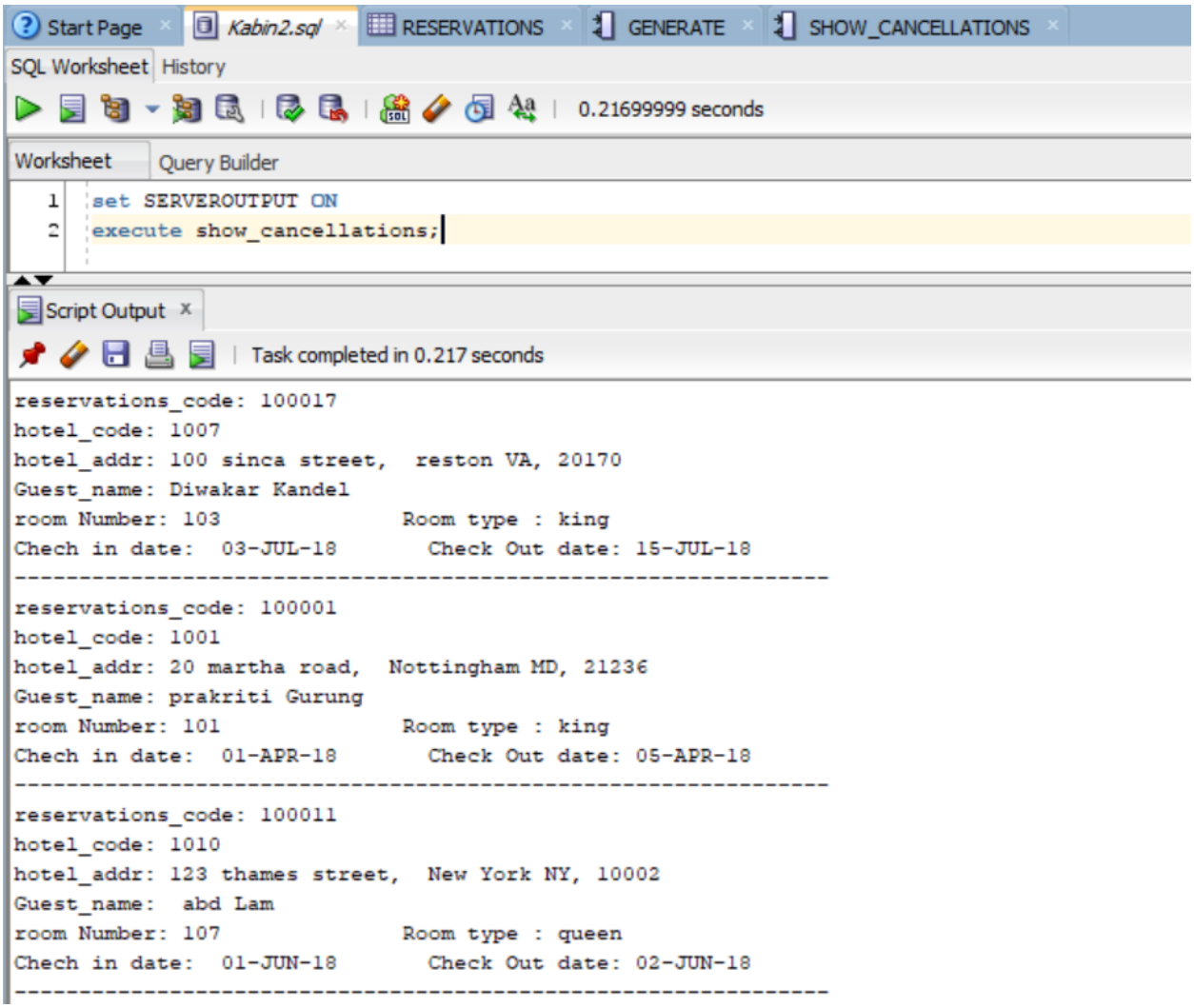
10. ShowCancelations: Print all canceled reservations in the hotel management system. Show reservation ID, hotel name, location, guest name, room type, dates

```
--Procedure Created By Kabin Dulal
-- This procedure is used to see all of the cancelled reservation
--Inputs none
--The output will be list of reservation with detail information

--creating a process
create or replace procedure show_cancellations-- creating procedure
is
cursor d1 is
select distinct r.reservation_stat,r.reservation_code, r.guest_name,
r.check_in_date, r.Check_out_date, h.hotel_code, h.hotel_addr,
h.hotel_city, h.hotel_state, h.hotel_zip_code, r.room_code, RT.ROOM_DESC
from reservations R, hotels H, rooms rm, ROOM_TYPES RT
where  r.Reservation_stat = 'Cancelled'
    and r.hotel_code = H.hotel_code
    and r.room_code= RM.room_code and
    RM.room_type_code= RT.ROOM_TYPE_CODE;
     excep exception;
begin
for a_c in d1
loop
if a_c.reservation_stat <> 'Cancelled'
 then
raise excep;
end if;
dbms_output.put_line('reservations_code:'||' '|| a_c.reservation_code );
dbms_output.put_line('hotel_code:'||' '|| a_c.hotel_code );
dbms_output.put_line('hotel_addr:'||' '|| a_c.hotel_addr ||',  '||
a_c.hotel_city ||' '||
     a_c.hotel_state ||', '|| a_c.hotel_zip_code );
dbms_output.put_line('Guest_name:'||' '|| a_c.guest_name );
```

```
 dbms_output.put_line(                    'room Number:'||' '|| a_c.room_code
||'              Room type : '||a_c.room_desc  );
 dbms_output.put_line(                    'Chech in date: '||' '||
a_c.check_in_date ||'       Check Out date: '||a_c.Check_out_date );
 dbms_output.put_line('-----------------------------------------------------
---------');
end loop;
exception
when excep then
dbms_output.put_line('Either reservation is not cancelled or you enter wrong
hotel code');
end;
```

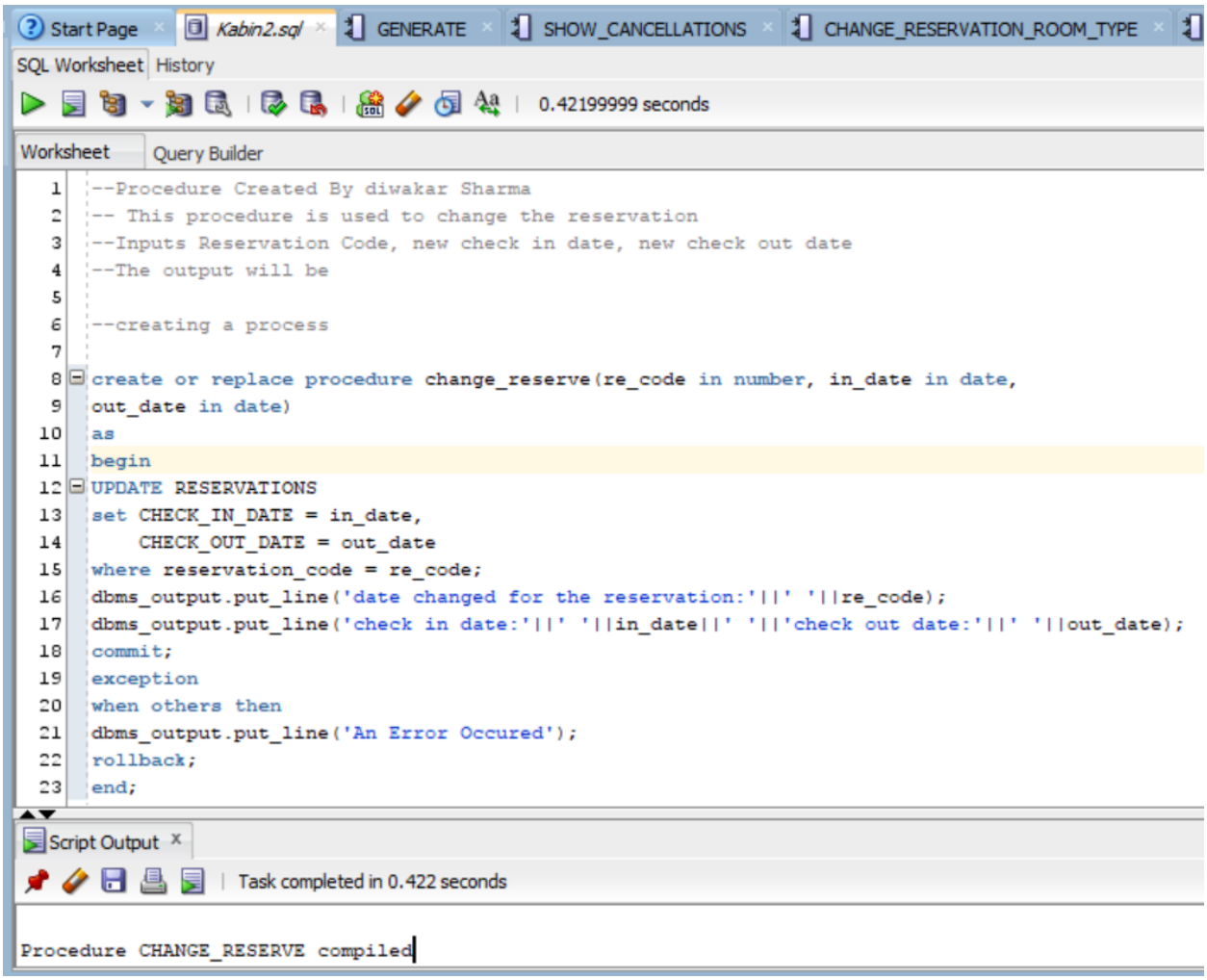



```
set SERVEROUTPUT ON
execute show_cancellations;
```

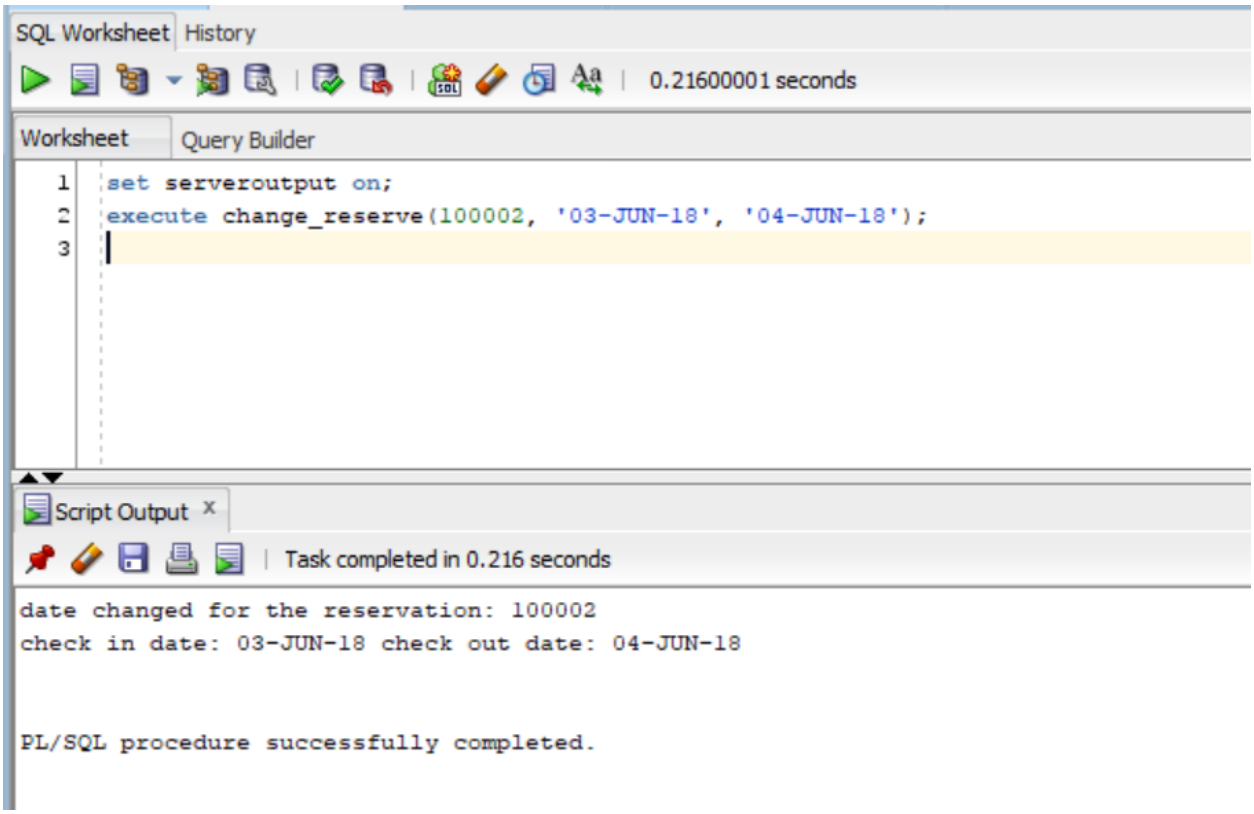

Start Page
Kabin2.sql
RESERVATIONS
GENERATE
SHOW_CANCELLATIONS
SQL Worksheet
History
0.21699999 seconds
Worksheet
Query Builder
1 set SERVEROUTPUT ON
2 execute show_cancellations;
Script Output
Task completed in 0.217 seconds
reservations_code: 100017
hotel_code: 1007
hotel_addr: 100 sinca street, reston VA, 20170
Guest_name: Diwakar Kandel
room Number: 103 Room type : king
Chech in date: 03-JUL-18 Check Out date: 15-JUL-18
---
reservations_code: 100001
hotel_code: 1001
hotel_addr: 20 martha road, Nottingham MD, 21236
Guest_name: prakriti Gurung
room Number: 101 Room type : king
Chech in date: 01-APR-18 Check Out date: 05-APR-18
---
reservations_code: 100011
hotel_code: 1010
hotel_addr: 123 thames street, New York NY, 10002
Guest_name: abd Lam
room Number: 107 Room type : queen
Chech in date: 01-JUN-18 Check Out date: 02-JUN-18

- Member 3: Diwakar Sharma
    11. Change a reservationDate: Input the reservation ID and change reservation start and end date, if there is availability in the same room type for the new date interval

```
--Procedure Created By diwakar Sharma
-- This procedure is used to change the reservation
--Inputs Reservation Code, new check in date, new check out date
--The output will be

--creating a process

create or replace procedure change_reserve(re_code in number, in_date in
date,
out_date in date)
as
begin
UPDATE RESERVATIONS
set CHECK_IN_DATE = in_date,
    CHECK_OUT_DATE = out_date
where reservation_code = re_code;
dbms_output.put_line('date changed for the reservation:'||' '||re_code);
dbms_output.put_line('check in date:'||' '||in_date||' '||'check out
date:'||' '||out_date);
commit;
exception
when others then
dbms_output.put_line('An Error Occured');
rollback;
end;
```



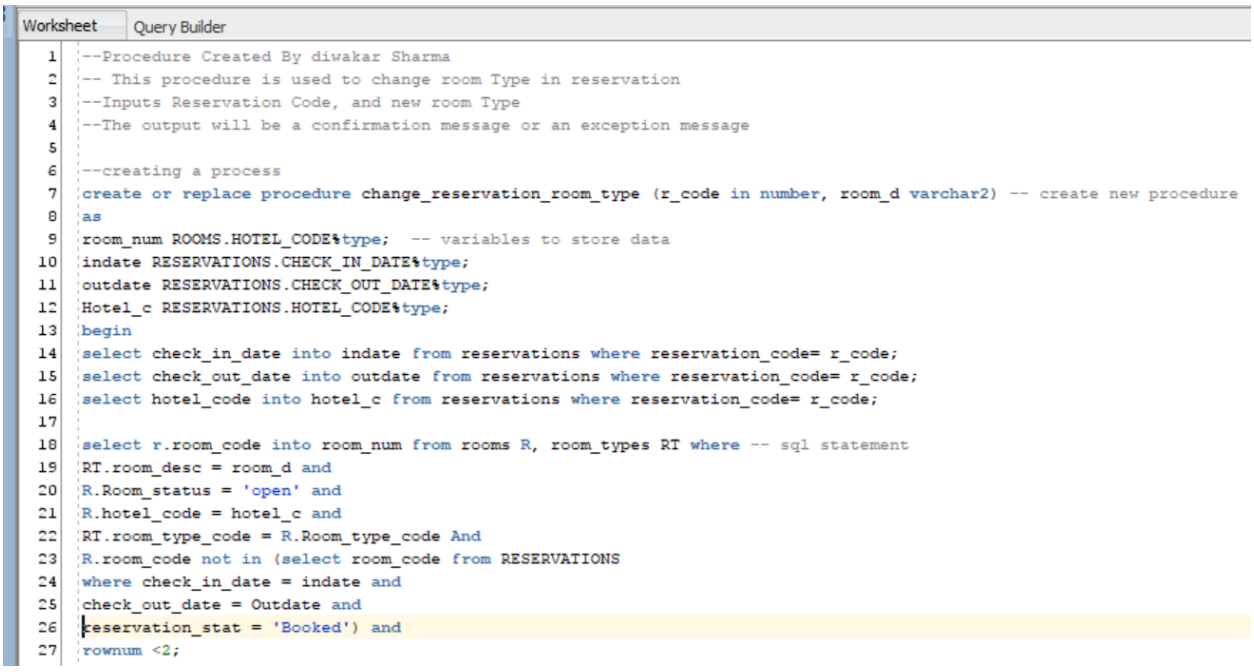

```
set serveroutput on;
execute change_reserve(100002, '03-JUN-18', '04-JUN-18');
```



12. Change a reservationRoomType: Input the reservation ID and change reservation room type if there is availability for that room type during the reservation's date interval

```
--Procedure Created By diwakar Sharma
-- This procedure is used to change room Type in reservation
--Inputs Reservation Code, and new room Type
--The output will be a confirmation message or an exception message

--creating a process
create or replace procedure change_reservation_room_type (r_code in
number, room_d varchar2) -- create new procedure
as
room_num ROOMS.HOTEL_CODE%type;  -- variables to store data
indate RESERVATIONS.CHECK_IN_DATE%type;
outdate RESERVATIONS.CHECK_OUT_DATE%type;
Hotel_c RESERVATIONS.HOTEL_CODE%type;
begin
select check_in_date into indate from reservations where reservation_code=
r_code;
select check_out_date into outdate from reservations where
reservation_code= r_code;
select hotel_code into hotel_c from reservations where reservation_code=
r_code;

select r.room_code into room_num from rooms R, room_types RT where -- sql
statement
```

```
RT.room_desc = room_d and
R.Room_status = 'open' and
R.hotel_code = hotel_c and
RT.room_type_code = R.Room_type_code And
R.room_code not in (select room_code from RESERVATIONS
where check_in_date = indate and
check_out_date = Outdate and
reservation_stat = 'Booked') and
rownum <2;

update reservations
set room_Code = room_num
where reservation_code = r_code;

DBMS_OUTPUT.PUT_LINE('Your room type has been changed to '||room_d ||'Your
new room_number is '||room_num);

exception -- it calls exception
when no_data_found then
DBMS_OUTPUT.PUT_LINE('no room availabel please change your search
criteria');

end;

set SERVEROUTPUT ON
begin
change_reservation_room_type (100043, 'king');
end;
```



```
28
29  update reservations
30  set room_Code = room_num
31  where reservation_code = r_code;
32
33  DBMS_OUTPUT.PUT_LINE('Your room type has been changed to '||room_d ||'Your new room_number is '||room_num);
34
35  exception -- it calls exception
36  when no_data_found then
37  DBMS_OUTPUT.PUT_LINE('no room availabel please change your search criteria');
38
39  end;
```

Script Output

Task completed in 0.125 seconds

```
Procedure CHANGE_RESERVATION_ROOM_TYPE compiled
```

```
set SERVEROUTPUT ON
begin
change_reservation_room_type (100043, 'king');
end;
```

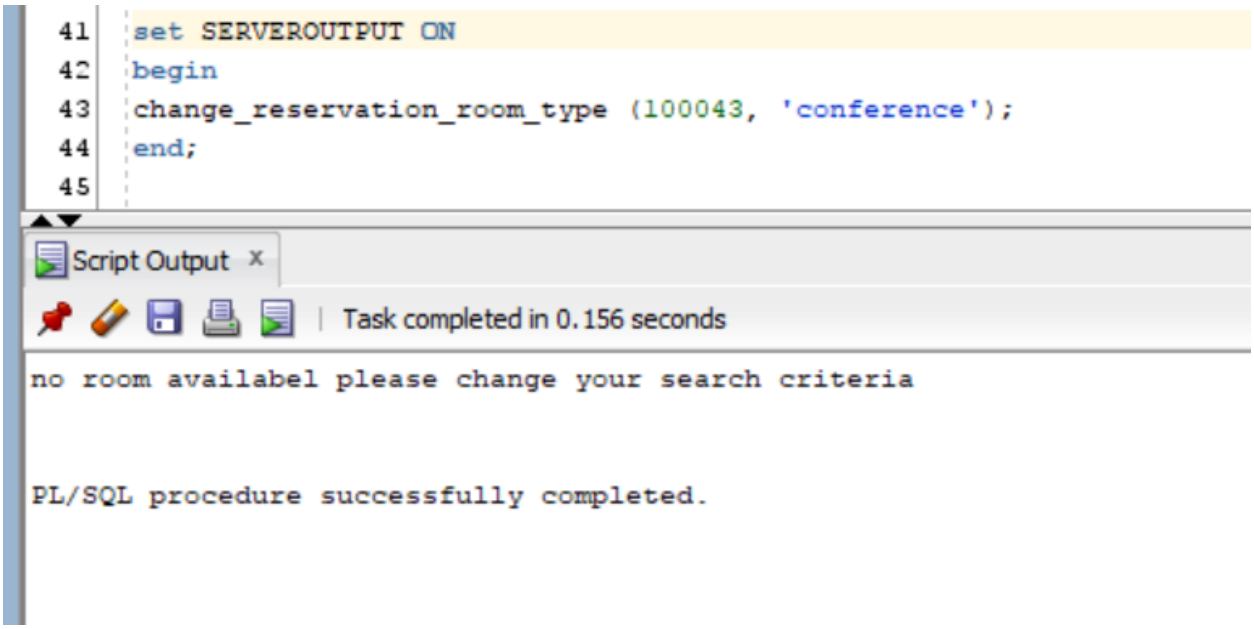
```
41  set SERVEROUTPUT ON
42  begin
43  change_reservation_room_type (100043, 'king');
44  end;
45
```

Script Output

Task completed in 0.25 seconds

```
Your room type has been changed to kingYour new room_number is 111


PL/SQL procedure successfully completed.
```

```
set SERVEROUTPUT ON
begin
change_reservation_room_type (100043, 'conference');
end;
```



13. Show single hotel reservations: Given a hotel ID, show all reservations for that hotel

```
--Procedure Created By diwakar Sharma
-- This procedure is used to view all of the reservation of the single hotel
--Inputs is hotel code
--The output will be list of reservation of the hotel

create or replace procedure show_reservations(h_code number)
is
cursor c1
is
select * from reservations where
hotel_code = h_code;
r2 c1%rowtype;
rc number;
excep exception;
begin
select distinct hotel_code into rc from reservations where hotel_code =
h_code;
if rc <1 then
raise no_data_found;
else
open c1;
loop
fetch c1 into r2;
exit when c1%notfound;
dbms_output.put_line('reservation_code: '|| r2.reservation_code);
dbms_output.put_line('guest_name:  '|| r2.guest_name);
dbms_output.put_line('check_in_date:  '|| r2.check_in_date);
dbms_output.put_line('check_out_date:  '|| r2.check_out_date);
dbms_output.put_line('reservation_date:  '|| r2.reservation_date);
dbms_output.put_line('no_of_guests: '|| r2.no_of_guests);
dbms_output.put_line('customer_code:  '|| r2.customer_code);
```

```
dbms_output.put_line('hotel_code:  '|| r2.hotel_code);
dbms_output.put_line('room_code:  '|| r2.room_code);
dbms_output.put_line('reservation status:  '|| r2.reservation_stat);
dbms_output.put_line('----------------------------------------------------
');
end loop;
close c1;
end if;
exception
when no_data_found then
dbms_output.put_line('this is wrong hotel code, Please enter tghe correct
Hotel ID');
end;
```

Worksheet | Query Builder

```
1  --Procedure Created By diwakar Sharma
2  -- This procedure is used to view all of the reservation of the single hotel
3  --Inputs is hotel code
4  --The output will be list of reservation of the hotel
5
6  create or replace procedure show_reservations(h_code number)
7  is
8  cursor c1
9  is
10 select * from reservations where
11 hotel_code = h_code;
12 r2 c1%rowtype;
13 rc number;
14 excep exception;
15 begin
16 select distinct hotel_code into rc from reservations where hotel_code = h_code;
17 if rc <1 then
18 raise no_data_found;
19 else
20 open c1;
21 loop
22 fetch c1 into r2;
```

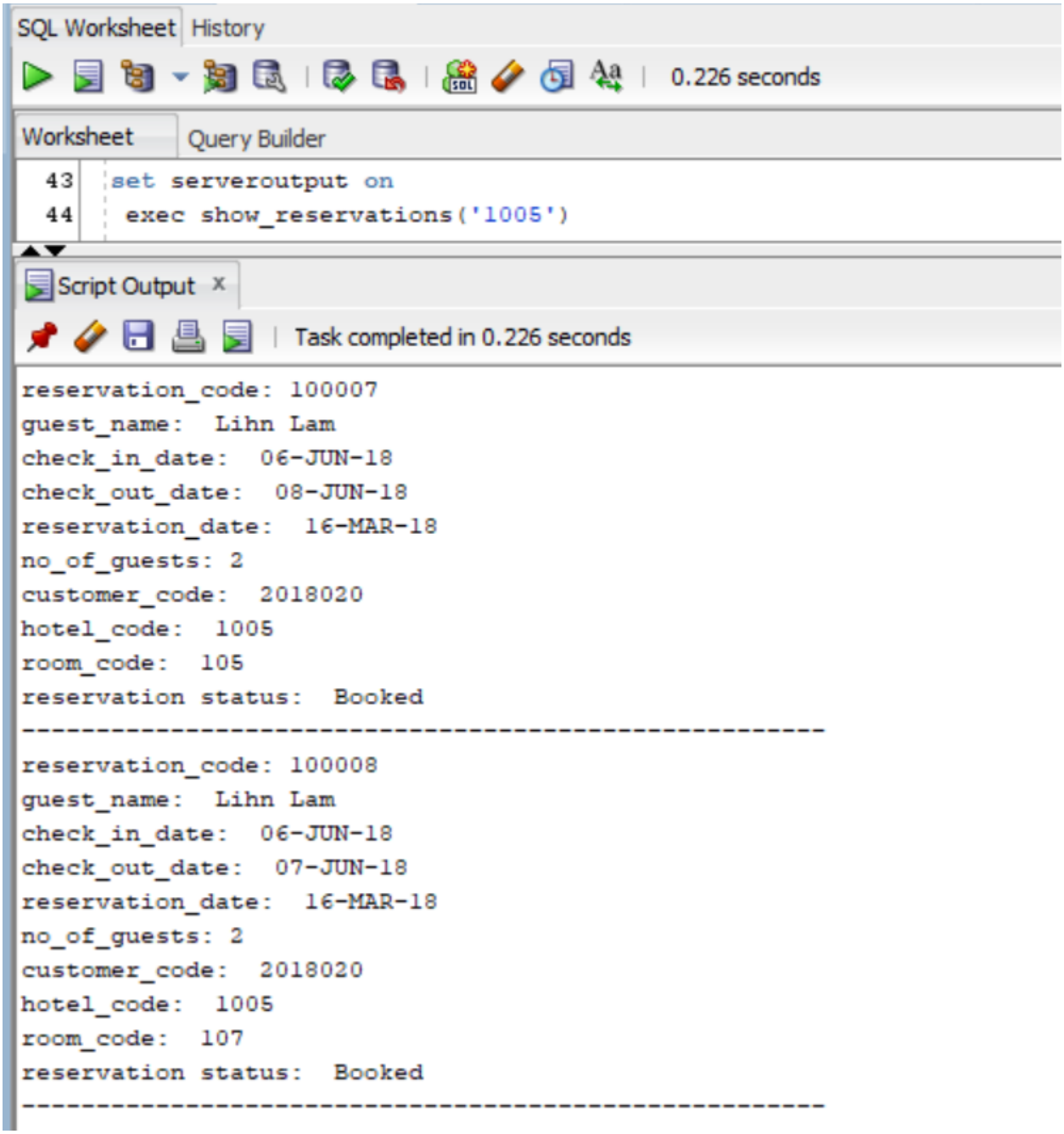
```
23  exit when c1%notfound;
24  dbms_output.put_line('reservation_code: '|| r2.reservation_code);
25  dbms_output.put_line('guest_name:  '|| r2.guest_name);
26  dbms_output.put_line('check_in_date:  '|| r2.check_in_date);
27  dbms_output.put_line('check_out_date:  '|| r2.check_out_date);
28  dbms_output.put_line('reservation_date:  '|| r2.reservation_date);
29  dbms_output.put_line('no_of_guests: '|| r2.no_of_guests);
30  dbms_output.put_line('customer_code:  '|| r2.customer_code);
31  dbms_output.put_line('hotel_code:  '|| r2.hotel_code);
32  dbms_output.put_line('room_code:  '|| r2.room_code);
33  dbms_output.put_line('reservation status:  '|| r2.reservation_stat);
34  dbms_output.put_line('-----------------------------------------------------');
35  end loop;
36  close c1;
37  end if;
38  exception
39  when no_data_found then
40  dbms_output.put_line('this is wrong hotel code, Please enter tghe correct Hotel ID');
41  end;
42
```

Script Output x

Task completed in 0.47 seconds

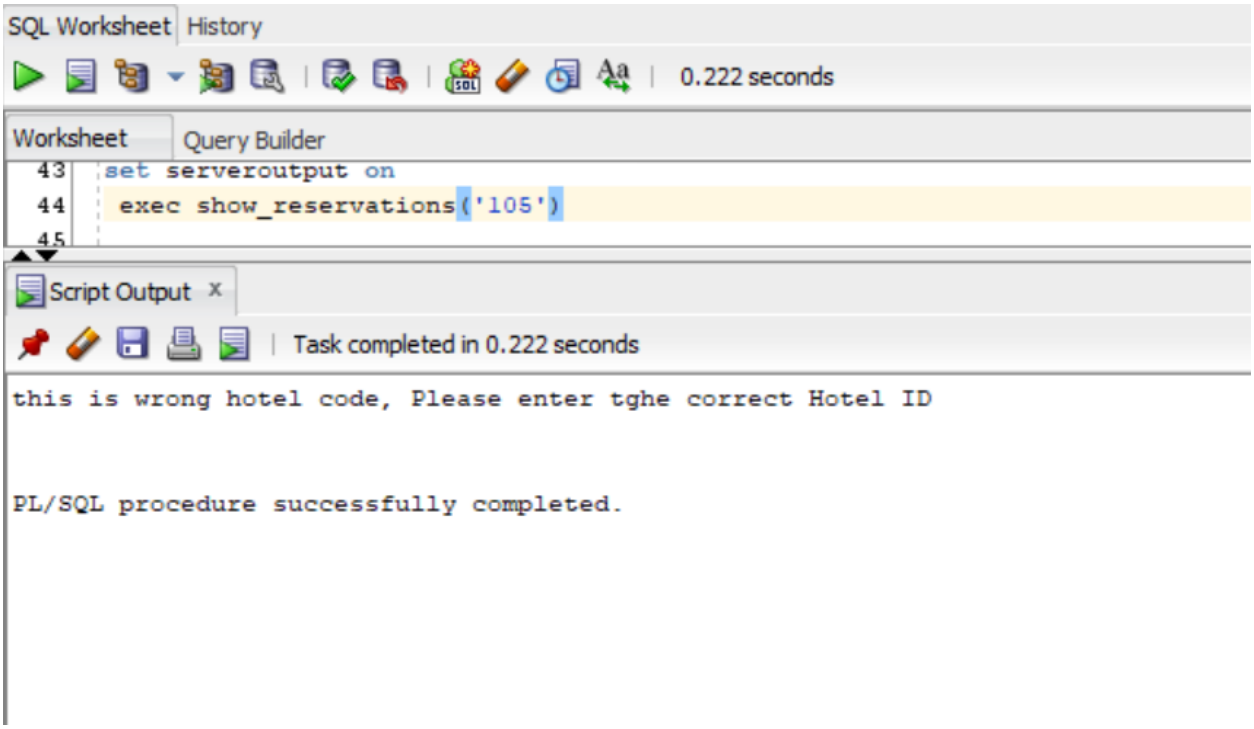
```
Procedure SHOW_RESERVATIONS compiled
```

```
set serveroutput on
 exec show_reservations('1005')
```


SQL Worksheet
History
0.226 seconds
Worksheet
Query Builder
43 set serveroutput on
44 exec show_reservations('1005')
Script Output
Task completed in 0.226 seconds
reservation_code: 100007
guest_name: Lihn Lam
check_in_date: 06-JUN-18
check_out_date: 08-JUN-18
reservation_date: 16-MAR-18
no_of_guests: 2
customer_code: 2018020
hotel_code: 1005
room_code: 105
reservation status: Booked
reservation_code: 100008
guest_name: Lihn Lam
check_in_date: 06-JUN-18
check_out_date: 07-JUN-18
reservation_date: 16-MAR-18
no_of_guests: 2
customer_code: 2018020
hotel_code: 1005
room_code: 107
reservation status: Booked

```
reservation_code: 100023
guest_name:  Padey Solti
check_in_date:  08-MAR-18
check_out_date:  10-MAR-18
reservation_date:  01-OCT-17
no_of_guests: 4
customer_code:  2018023
hotel_code:  1005
room_code:  101
reservation status:  checked out
------------------------------------------------------


PL/SQL procedure successfully completed.
```

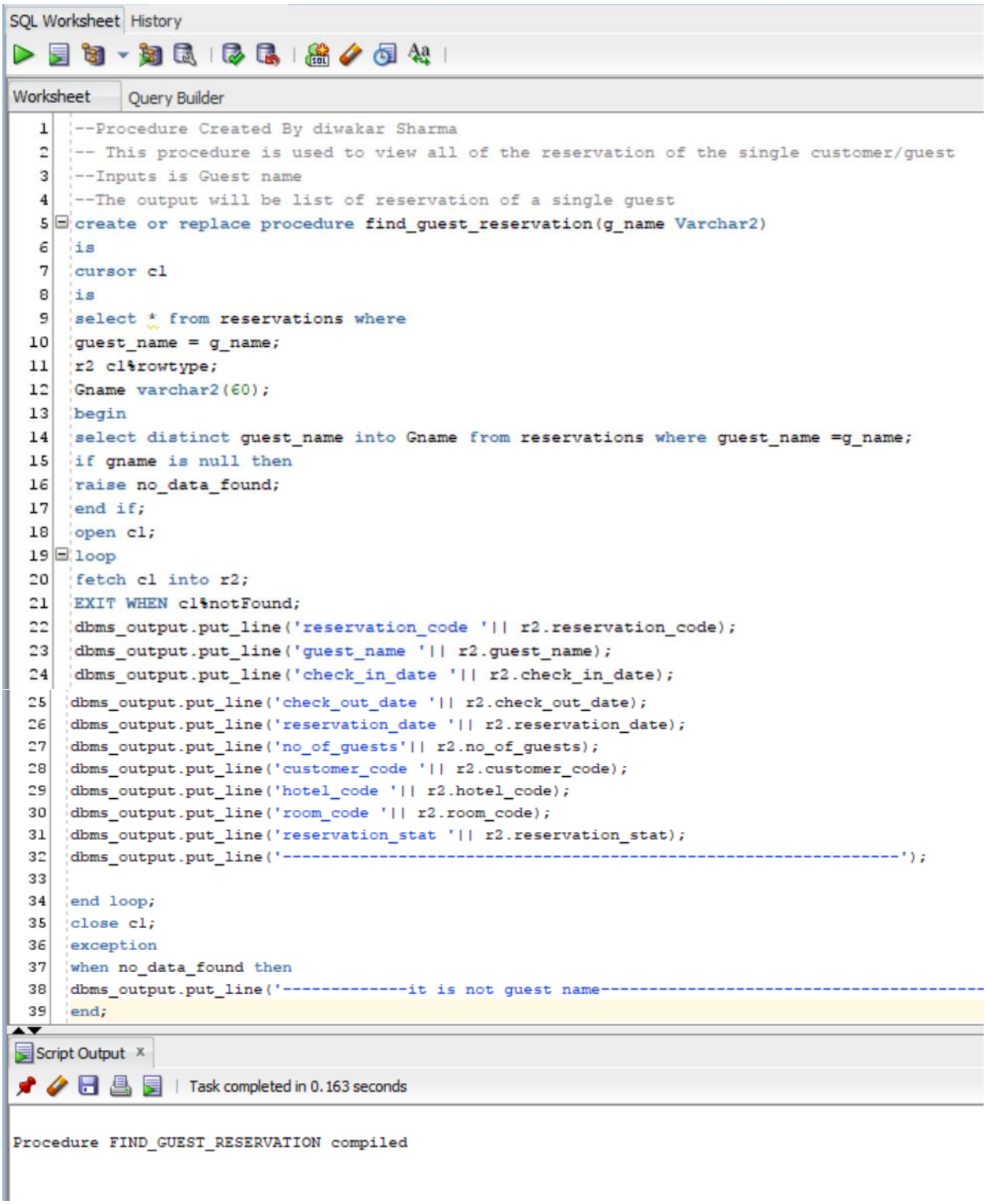
```
set serveroutput on
 exec show_reservations('105')
```



14. Show single guest reservations: Given a guest name, find all reservations under that name

```
--Procedure Created By diwakar Sharma
```

```
-- This procedure is used to view all of the reservation of the single
customer/guest
--Inputs is Guest name
--The output will be list of reservation of a single guest
create or replace procedure find_guest_reservation(g_name Varchar2)
is
cursor c1
is
select * from reservations where
guest_name = g_name;
r2 c1%rowtype;
Gname varchar2(60);
begin
select distinct guest_name into Gname from reservations where guest_name
=g_name;
if gname is null then
raise no_data_found;
end if;
open c1;
loop
fetch c1 into r2;
EXIT WHEN c1%notFound;
dbms_output.put_line('reservation_code '|| r2.reservation_code);
dbms_output.put_line('guest_name '|| r2.guest_name);
dbms_output.put_line('check_in_date '|| r2.check_in_date);
dbms_output.put_line('check_out_date '|| r2.check_out_date);
dbms_output.put_line('reservation_date '|| r2.reservation_date);
dbms_output.put_line('no_of_guests'|| r2.no_of_guests);
dbms_output.put_line('customer_code '|| r2.customer_code);
dbms_output.put_line('hotel_code '|| r2.hotel_code);
dbms_output.put_line('room_code '|| r2.room_code);
dbms_output.put_line('reservation_stat '|| r2.reservation_stat);
dbms_output.put_line('-----------------------------------------------------------
---------');

end loop;
close c1;
exception
when no_data_found then
dbms_output.put_line('it is not guest name');
end;

set serveroutput on;
execute find_reservation_total('ABC Lam');
```




```
set serveroutput on;
execute find_guest_reservation('Lihn Lam');
```

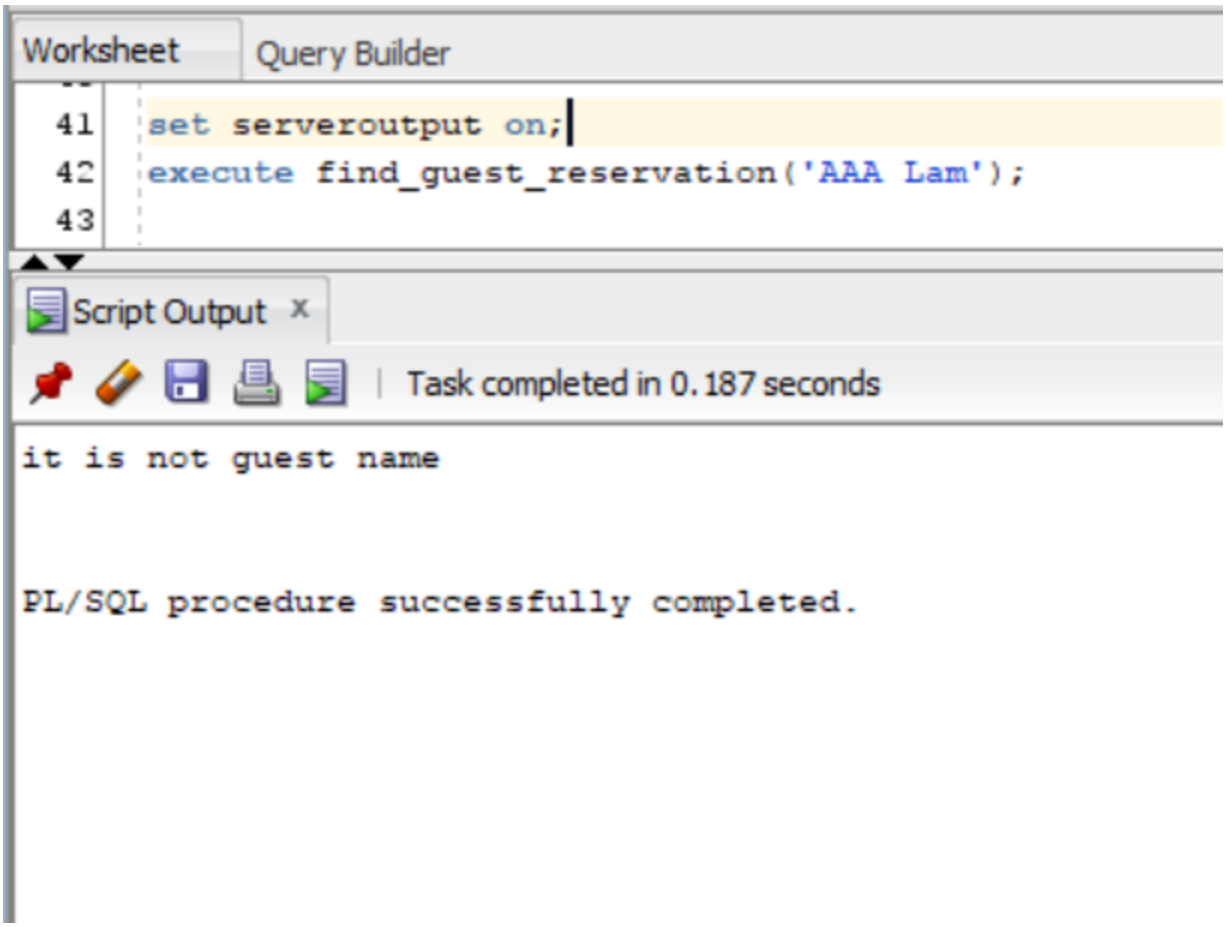
```
41  set serveroutput on;
42  execute find_guest_reservation('Lihn Lam');
43
```

Script Output x

Task completed in 0.179 seconds

```
reservation_code 100007
guest_name Lihn Lam
check_in_date 06-JUN-18
check_out_date 08-JUN-18
reservation_date 16-MAR-18
no_of_guests2
customer_code 2018020
hotel_code 1005
room_code 105
reservation_stat Booked
------------------------------------------------------------------
reservation_code 100008
guest_name Lihn Lam
check_in_date 06-JUN-18
check_out_date 07-JUN-18
reservation_date 16-MAR-18
no_of_guests2
customer_code 2018020
hotel_code 1005
room_code 107
reservation_stat Booked
------------------------------------------------------------------
```

```
set serveroutput on;
execute find_guest_reservation('AAA Lam');
```



15. Total Monthly Income Report: Calculate and display total income from all sources of all hotels. Totals must be printed by month, and for each month by room type, service type. Include discounts.

```
--Procedure Created By diwakar Sharma
-- This procedure is used to view report by month
--Inputs none
--The output will be list of monthly report
create or replace procedure Monthly_report
is
 cursor c1 is -- this cursor gives  information about first month income
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount
,room_desc from room_income where month=1 group by room_desc;
  cursor c2 is -- this cursor gives information about first mont service
   select sum(service_rate) as S_revenue, service_desc from service_income
where month=1 group by SERVICE_DESC;
    cursor c3 is---- this cursor gives  information about second month
income
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount
,room_desc from room_income where month=2 group by room_desc;
  cursor c4 is---- this cursor gives  information about second month
service
   select sum(service_rate) as S_revenue, service_desc from service_income
where month=2 group by SERVICE_DESC;
    cursor c5 is-- third month
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount
,room_desc from room_income where month=3 group by room_desc;
```

```
  cursor c6 is--third month
   select sum(service_rate) as S_revenue, service_desc from service_income
where month=3 group by SERVICE_DESC;

    R1 c1%rowtype;   -- they store cursor information
    R2 C2%rowtype;
     R3 c3%rowtype;
    R4 C4%rowtype;
     R5 c5%rowtype;
    R6 C6%rowtype;

    Begin
            open c1;
            dbms_output.put_line ('--------------Monthly Sales Report-----
-----------');
            dbms_output.put_line ('------------------------------------
------------');
            dbms_output.put_line ('--------------Report for JANUARY-------
---------');
                dbms_output.put_line ('--------Rooms--------');
            loop
            fetch c1 into R1; exit when c1%notfound;
            dbms_output.put_line (R1.room_desc||'          $'||
r1.R_revenue);
            end loop;
            close c1;

            open c2;
            dbms_output.put_line ('--------Services---------');
            loop
            fetch c2 into r2; exit when c2%notfound;
             dbms_output.put_line (R2.service_desc||'          $'||
R2.S_revenue);
             end loop; close c2;
             open c3;
            dbms_output.put_line ('--------------Report for FEBREARY------
----------');
                dbms_output.put_line ('--------Rooms--------');
            loop
            fetch c3 into R3; exit when c3%notfound;
            dbms_output.put_line (R3.room_desc||'
$'||r3.R_revenue);
            end loop; close c3;
            open c4;
       dbms_output.put_line ('--------Services---------');

            loop
            fetch c4 into r4; exit when c4%notfound;
             dbms_output.put_line (R4.service_desc ||'
$'||R4.S_revenue);
             end loop; close c4;
             open c5;
            dbms_output.put_line ('--------------Report for MARCH---------
-------');
                dbms_output.put_line ('--------Rooms--------');
            loop
            fetch c5 into r5; exit when c5%notfound;
```

```
            dbms_output.put_line (R5.room_desc ||'
$'||R5.r_revenue);
            end loop; close c5;
            open c6;
        dbms_output.put_line ('--------Services---------');
            loop
            fetch c6 into r6; exit when c6%notfound;
            dbms_output.put_line (R6.service_desc||'          $'||
R6.S_revenue);
            end loop; close c6;

 end;
```

Worksheet Query Builder

```
--Procedure Created By diwakar Sharma
-- This procedure is used to view report by month
--Inputs none
--The output will be list of monthly report
create or replace procedure Monthly_report
is
 cursor c1 is -- this cursor gives  information about first month income
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount ,room_desc from room_income where month=1 group by room_desc;
  cursor c2 is -- this cursor gives information about first mont service
   select sum(service_rate) as S_revenue, service_desc from service_income where month=1 group by SERVICE_DESC;
    cursor c3 is---- this cursor gives  information about second month income
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount ,room_desc from room_income where month=2 group by room_desc;
  cursor c4 is---- this cursor gives  information about second month service
   select sum(service_rate) as S_revenue, service_desc from service_income where month=2 group by SERVICE_DESC;
    cursor c5 is-- third month
  Select sum(room_charge)as R_revenue, Sum(discount) as Discount ,room_desc from room_income where month=3 group by room_desc;
  cursor c6 is--third month
   select sum(service_rate) as S_revenue, service_desc from service_income where month=3 group by SERVICE_DESC;

    R1 c1%rowtype;   -- they store cursor information
    R2 C2%rowtype;
     R3 c3%rowtype;
    R4 C4%rowtype;
     R5 c5%rowtype;
    R6 C6%rowtype;

   Begin
          open c1;
          dbms_output.put_line ('--------------Monthly Sales Report----------------');
          dbms_output.put_line ('-------------------------------------------------');
          dbms_output.put_line ('--------------Report for JANUARY----------------');
              dbms_output.put_line ('--------Rooms---------');
          loop
          fetch c1 into R1; exit when c1%notfound;
          dbms_output.put_line (R1.room_desc||'          $'|| r1.R_revenue);
          end loop;
          close c1;

          open c2;
          dbms_output.put_line ('--------Services---------');
          loop
          fetch c2 into r2; exit when c2%notfound;
           dbms_output.put_line (R2.service_desc||'          $'|| R2.S_revenue);
           end loop; close c2;
           open c3;
          dbms_output.put_line ('--------------Report for FEBREARY----------------');
              dbms_output.put_line ('--------Rooms---------');
          loop
          fetch c3 into R3; exit when c3%notfound;
          dbms_output.put_line (R3.room_desc||'          $'||r3.R_revenue);
          end loop; close c3;
          open c4;
      dbms_output.put_line ('--------Services---------');

```

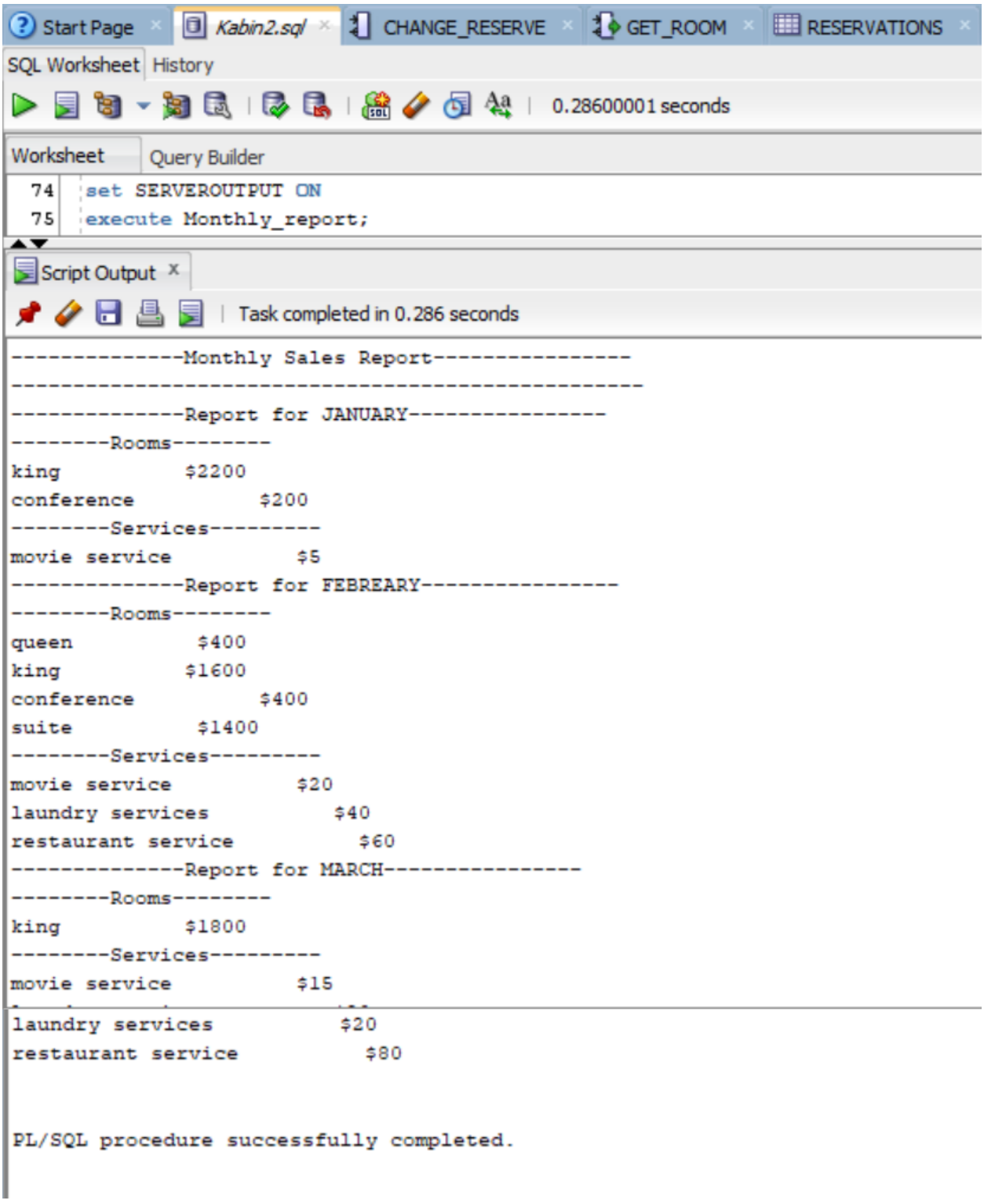
```
55          loop
56          fetch c4 into r4; exit when c4%notfound;
57           dbms_output.put_line (R4.service_desc ||'          $'||R4.S_revenue);
58           end loop; close c4;
59           open c5;
60          dbms_output.put_line ('--------------Report for MARCH----------------');
61              dbms_output.put_line ('--------Rooms--------');
62          loop
63          fetch c5 into r5; exit when c5%notfound;
64           dbms_output.put_line (R5.room_desc ||'          $'||R5.r_revenue);
65           end loop; close c5;
66           open c6;
67      dbms_output.put_line ('--------Services---------');
68           loop
69           fetch c6 into r6; exit when c6%notfound;
70           dbms_output.put_line (R6.service_desc||'          $'|| R6.S_revenue);
71           end loop; close c6;
72
73  end;
74
```

Script Output

Task completed in 0.218 seconds

```
Procedure MONTHLY_REPORT compiled
```

```
set SERVEROUTPUT ON
execute Monthly_report;
```

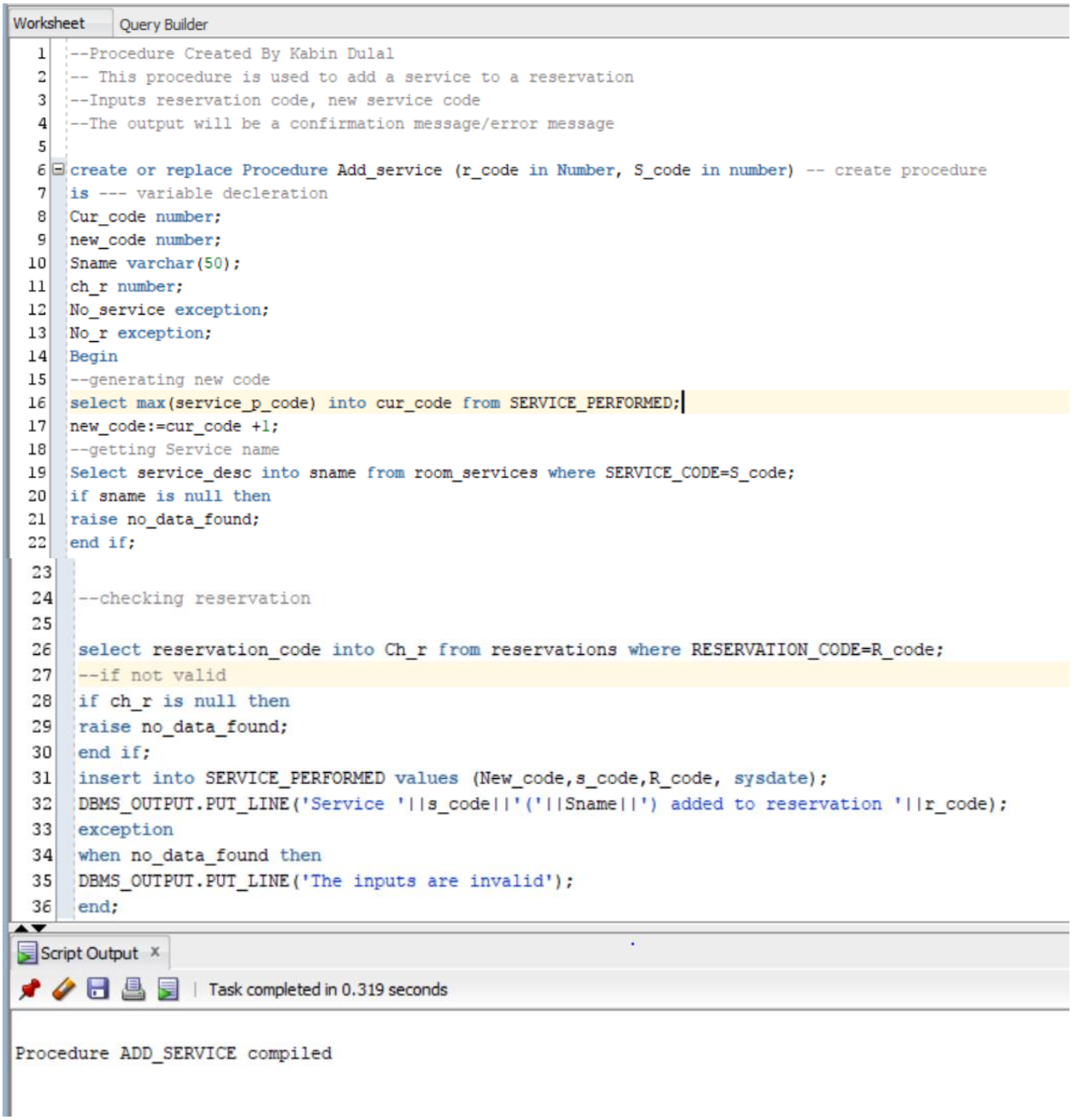

Start Page
Kabin2.sql
CHANGE_RESERVE
GET_ROOM
RESERVATIONS
SQL Worksheet
History
0.28600001 seconds
Worksheet
Query Builder
74 set SERVEROUTPUT ON
75 execute Monthly_report;
Script Output
Task completed in 0.286 seconds
----------Monthly Sales Report----------
----------Report for JANUARY----------
--------Rooms--------
king $2200
conference $200
--------Services---------
movie service $5
----------Report for FEBREARY----------
--------Rooms--------
queen $400
king $1600
conference $400
suite $1400
--------Services---------
movie service $20
laundry services $40
restaurant service $60
----------Report for MARCH----------
--------Rooms--------
king $1800
--------Services---------
movie service $15
laundry services $20
restaurant service $80
PL/SQL procedure successfully completed.

Member 4: Split

16. Add a service to a reservation: Input: ReservationID, specific service. Add the service to the reservation for a particular date. Multiple services are allowed on a reservation for the same date.

----------Kabin Dulal----------

```
--Procedure Created By Kabin Dulal
-- This procedure is used to add a service to a reservation
--Inputs reservation code, new service code
--The output will be a confirmation message/error message

create or replace Procedure Add_service (r_code in Number, S_code in number)
-- create procedure
is --- variable decleration
Cur_code number;
new_code number;
Sname varchar(50);
ch_r number;
No_service exception;
No_r exception;
Begin
--generating new code
select max(service_p_code) into cur_code from SERVICE_PERFORMED;
new_code:=cur_code +1;
--getting Service name
Select service_desc into sname from room_services where SERVICE_CODE=S_code;
if sname is null then
raise no_data_found;
end if;

--checking reservation

select reservation_code into Ch_r from reservations where
RESERVATION_CODE=R_code;
--if not valid
if ch_r is null then
raise no_data_found;
end if;
insert into SERVICE_PERFORMED values (New_code,s_code,R_code, sysdate);
DBMS_OUTPUT.PUT_LINE('Service '||s_code||'('||Sname||') added to reservation
'||r_code);
exception
when no_data_found then
DBMS_OUTPUT.PUT_LINE('The inputs are invalid');
end;
```



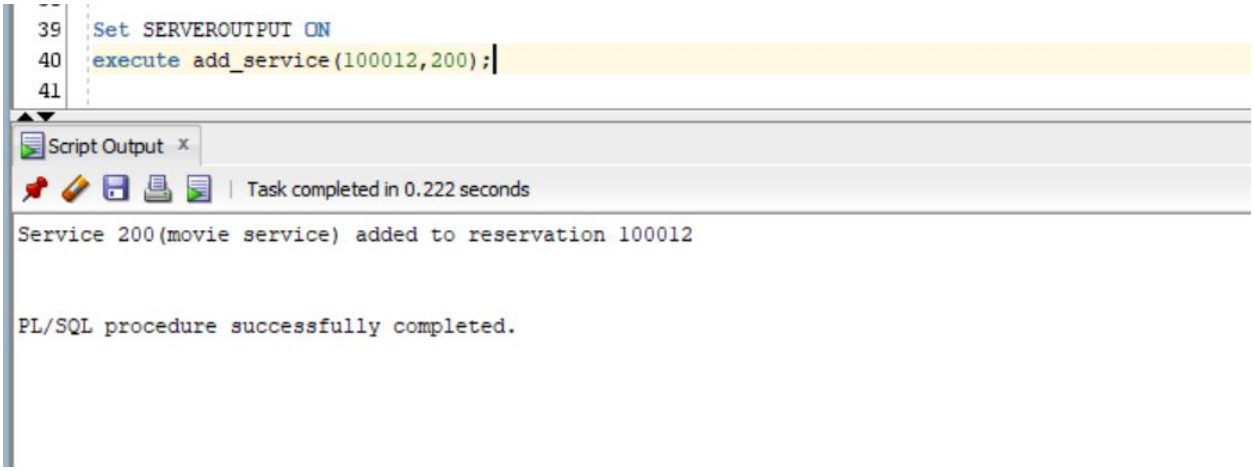
```
--Procedure Created By Kabin Dulal
-- This procedure is used to add a service to a reservation
--Inputs reservation code, new service code
--The output will be a confirmation message/error message

create or replace Procedure Add_service (r_code in Number, S_code in number) -- create procedure
is --- variable decleration
Cur_code number;
new_code number;
Sname varchar(50);
ch_r number;
No_service exception;
No_r exception;
Begin
--generating new code
select max(service_p_code) into cur_code from SERVICE_PERFORMED;
new_code:=cur_code +1;
--getting Service name
Select service_desc into sname from room_services where SERVICE_CODE=S_code;
if sname is null then
raise no_data_found;
end if;

--checking reservation

select reservation_code into Ch_r from reservations where RESERVATION_CODE=R_code;
--if not valid
if ch_r is null then
raise no_data_found;
end if;
insert into SERVICE_PERFORMED values (New_code,s_code,R_code, sysdate);
DBMS_OUTPUT.PUT_LINE('Service '||s_code||'('||Sname||') added to reservation '||r_code);
exception
when no_data_found then
DBMS_OUTPUT.PUT_LINE('The inputs are invalid');
end;
```

Script Output

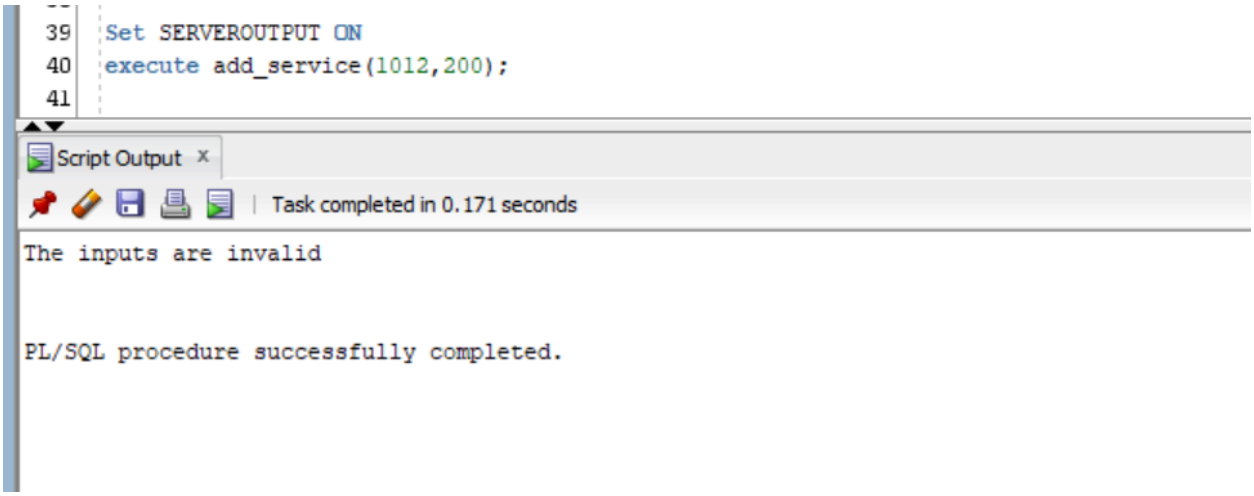
Task completed in 0.319 seconds

```
Procedure ADD_SERVICE compiled
```

```
Set SERVEROUTPUT ON
execute add_service(100012,200);
```




```
Set SERVEROUTPUT ON
execute add_service(1012,200);
```

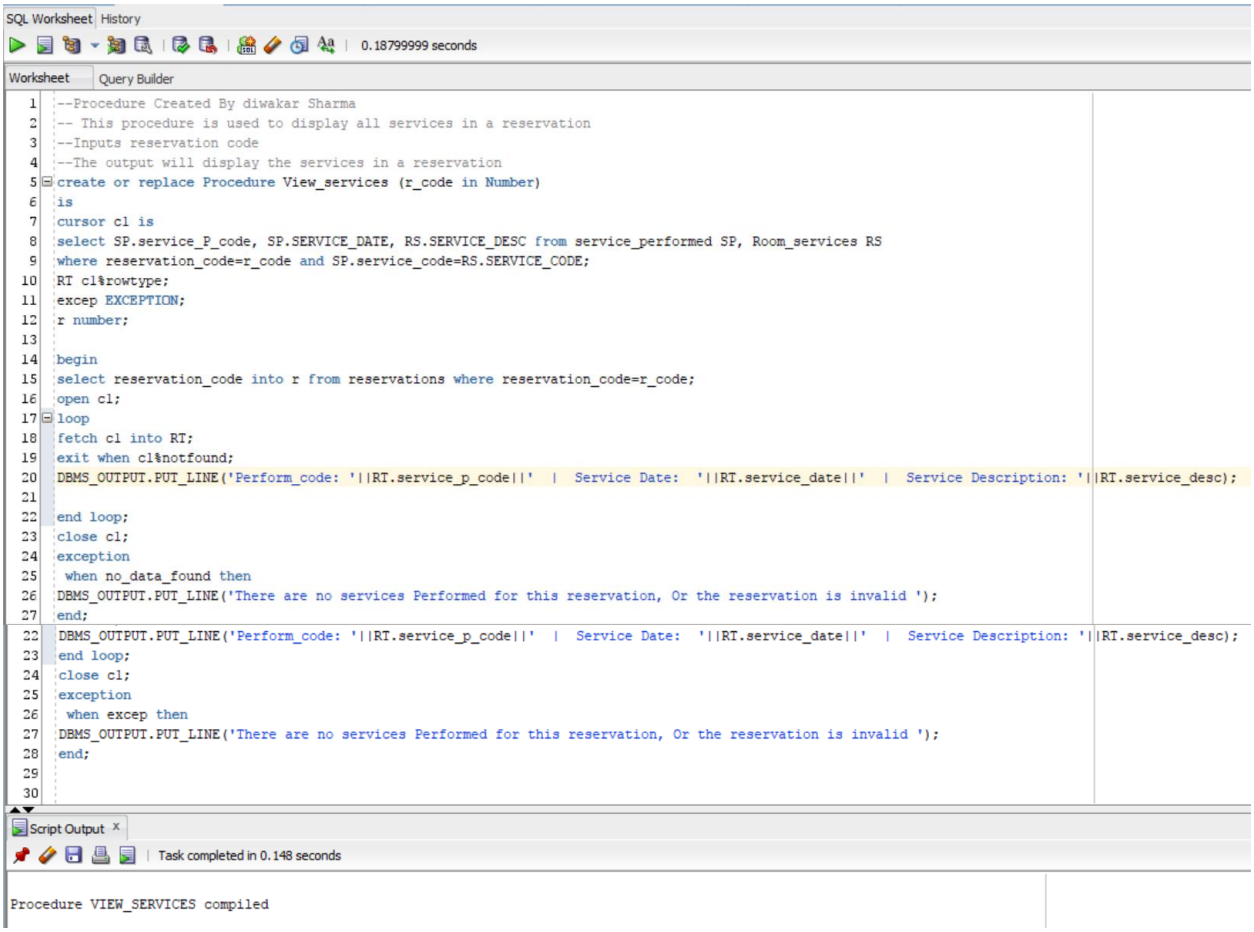


17. Reservation Services Report: Input the reservation ID and display all services on this reservation. Print “no services for this reservation” if none exists.

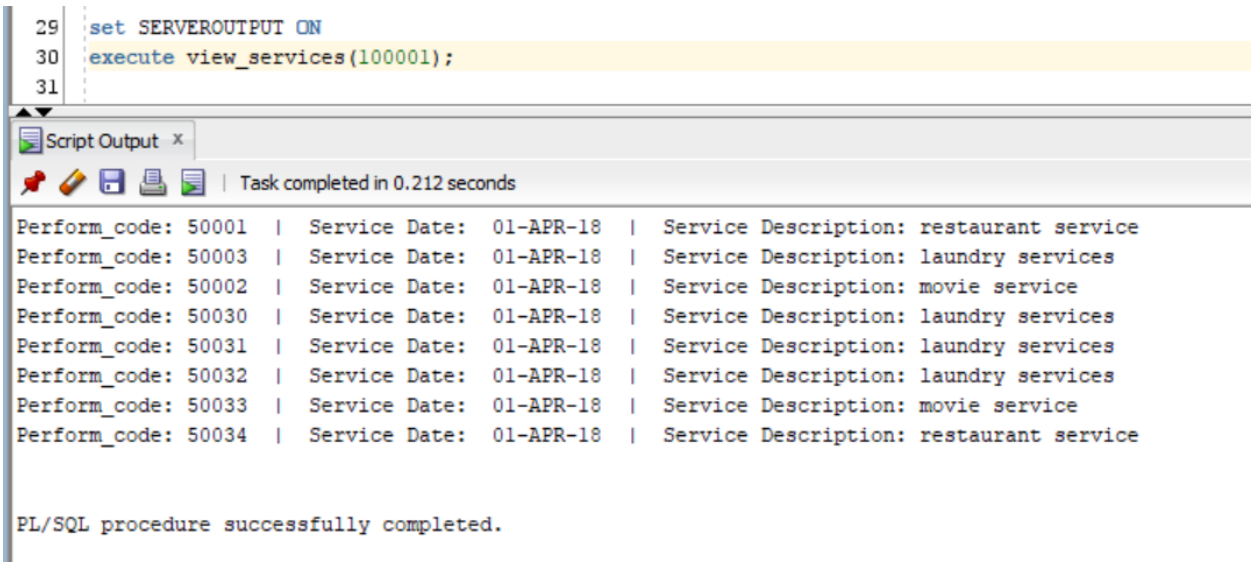
----------Diwakar Sharma----------

```
--Procedure Created By diwakar Sharma
-- This procedure is used to display all services in a reservation
--Inputs reservation code
--The output will display the services in a reservation
create or replace Procedure View_services (r_code in Number)
is
cursor c1 is
select SP.service_P_code, SP.SERVICE_DATE, RS.SERVICE_DESC from
service_performed SP, Room_services RS
where reservation_code=r_code and SP.service_code=RS.SERVICE_CODE;
RT c1%rowtype;
```

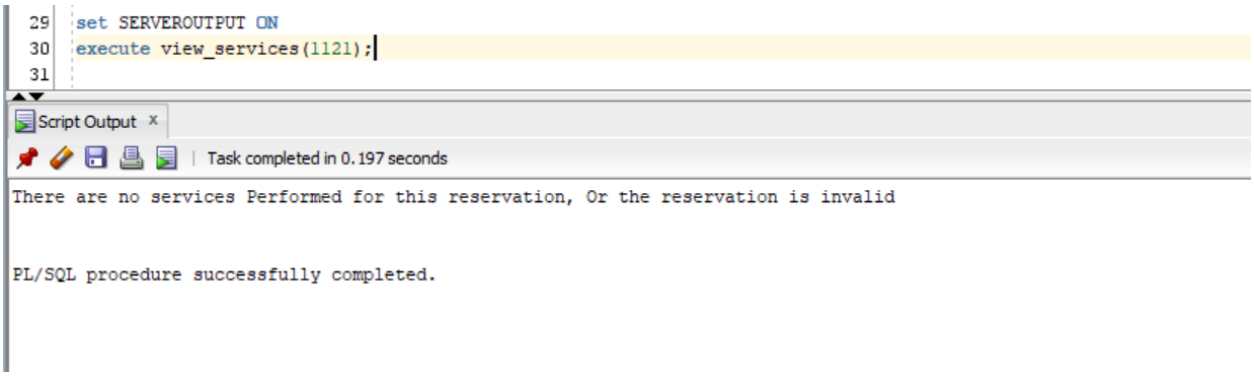
```
excep EXCEPTION;
r number;

begin
select reservation_code into r from reservations where
reservation_code=r_code;
open c1;
loop
fetch c1 into RT;
exit when c1%notfound;
DBMS_OUTPUT.PUT_LINE('Perform_code: '||RT.service_p_code||'  |  Service Date:
'||RT.service_date||'  |  Service Description: '||RT.service_desc);

end loop;
close c1;
exception
 when no_data_found then
DBMS_OUTPUT.PUT_LINE('There are no services Performed for this reservation,
Or the reservation is invalid ');
end;
```




```
set SERVEROUTPUT ON
execute view_services(100001);
```




```
set SERVEROUTPUT ON
execute view_services(1121);
```

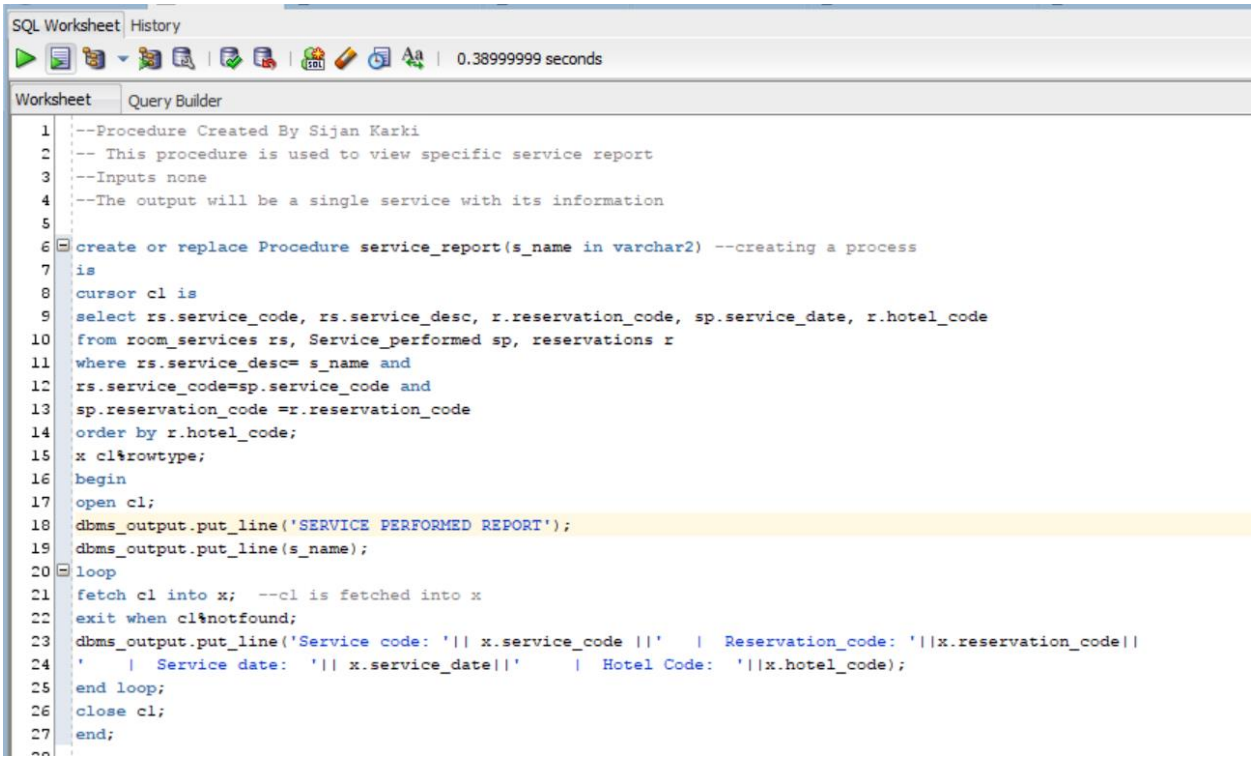


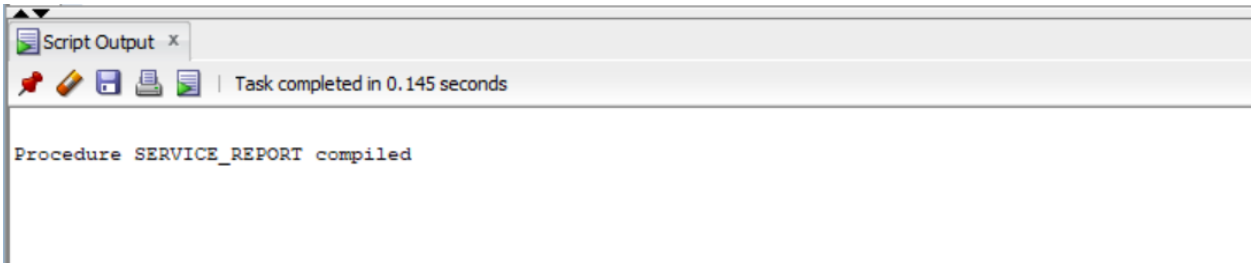
18. Show Specific Service Report: Input the service name, and display information on all reservations that have this service in all hotels

----------Sijan karki----------

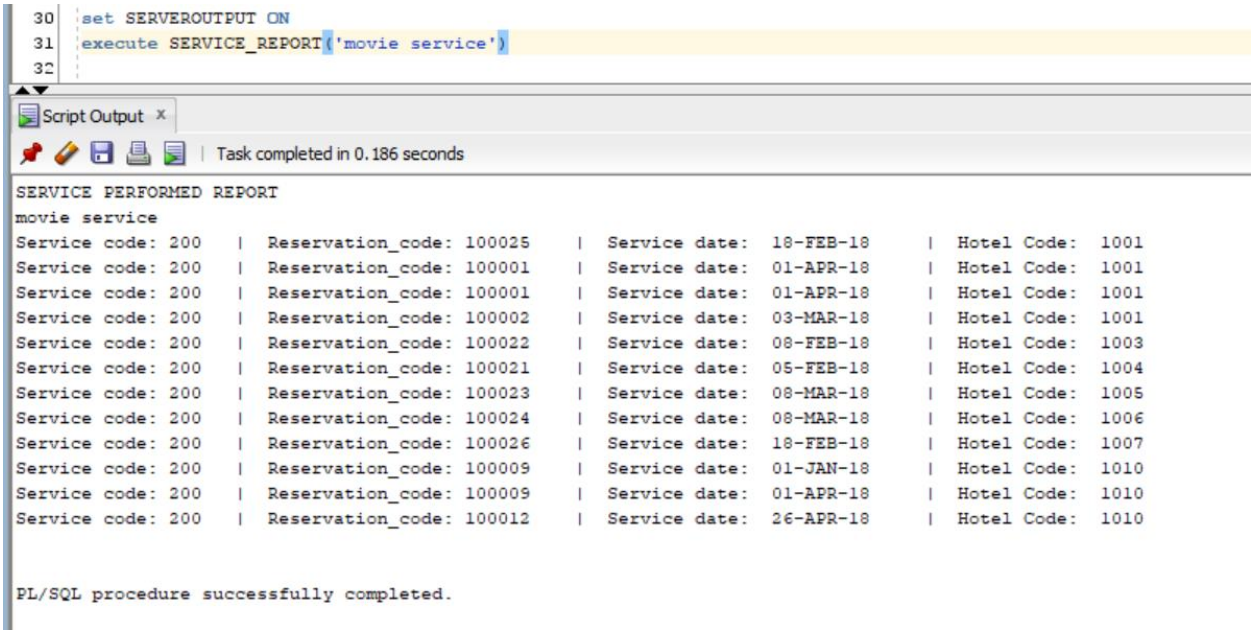
```
--Procedure Created By Sijan Karki
-- This procedure is used to view specific service report
--Inputs none
--The output will be a single service with its information
```

```
create or replace Procedure service_report(s_name in varchar2) --creating
a process
is
cursor c1 is
select rs.service_code, rs.service_desc, r.reservation_code,
sp.service_date, r.hotel_code
from room_services rs, Service_performed sp, reservations r
where rs.service_desc= s_name and
rs.service_code=sp.service_code and
sp.reservation_code =r.reservation_code
order by r.hotel_code;
x c1%rowtype;
begin
open c1;
dbms_output.put_line('SERVICE PERFORMED REPORT');
dbms_output.put_line(s_name);
loop
fetch c1 into x;  --c1 is fetched into x
exit when c1%notfound;
dbms_output.put_line('Service code: '|| x.service_code ||'    |
Reservation_code: '||x.reservation_code||
'     |  Service date:  '|| x.service_date||'      |  Hotel Code:
'||x.hotel_code);
end loop;
close c1;
end;
```






```
set SERVEROUTPUT ON
execute SERVICE_REPORT('movie service')
```




```
set SERVEROUTPUT ON
execute SERVICE_REPORT('restaurant service')
```

```
30  set SERVEROUTPUT ON
31  execute SERVICE_REPORT('restaurant service')
32
```

Script Output

Task completed in 0.198 seconds

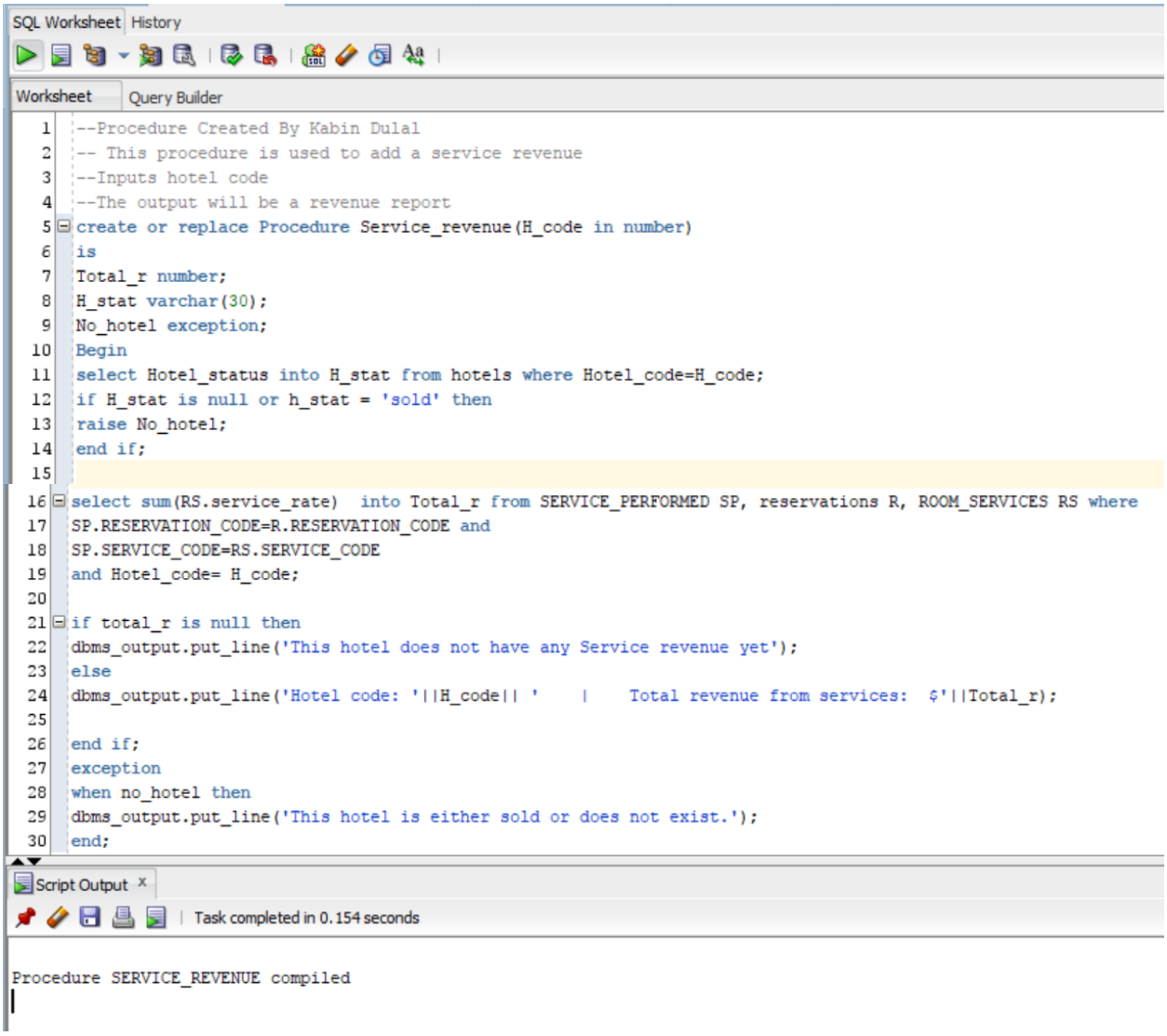
```
SERVICE PERFORMED REPORT
restaurant service
Service code: 100   |  Reservation_code: 100001   |  Service date:  01-APR-18   |  Hotel Code:  1001
Service code: 100   |  Reservation_code: 100001   |  Service date:  01-APR-18   |  Hotel Code:  1001
Service code: 100   |  Reservation_code: 100002   |  Service date:  03-MAR-18   |  Hotel Code:  1001
Service code: 100   |  Reservation_code: 100002   |  Service date:  03-MAR-18   |  Hotel Code:  1001
Service code: 100   |  Reservation_code: 100022   |  Service date:  08-FEB-18   |  Hotel Code:  1003
Service code: 100   |  Reservation_code: 100021   |  Service date:  05-FEB-18   |  Hotel Code:  1004
Service code: 100   |  Reservation_code: 100023   |  Service date:  08-MAR-18   |  Hotel Code:  1005
Service code: 100   |  Reservation_code: 100024   |  Service date:  08-MAR-18   |  Hotel Code:  1006
Service code: 100   |  Reservation_code: 100026   |  Service date:  18-FEB-18   |  Hotel Code:  1007


PL/SQL procedure successfully completed.
```

19. Total Services Income Report: Given a hotelID, calculate and display income from all services in all reservations in that hotel.

----------Kabin Dulal----------

```
--Procedure Created By Kabin Dulal
-- This procedure is used to add a service revenue
--Inputs hotel code
--The output will be a revenue report
create or replace Procedure Service_revenue(H_code in number)
is
Total_r number;
H_stat varchar(30);
No_hotel exception;
Begin
select Hotel_status into H_stat from hotels where Hotel_code=H_code;
if H_stat is null or h_stat = 'sold' then
raise No_hotel;
end if;

select sum(RS.service_rate)  into Total_r from SERVICE_PERFORMED SP,
reservations R, ROOM_SERVICES RS where
SP.RESERVATION_CODE=R.RESERVATION_CODE and
SP.SERVICE_CODE=RS.SERVICE_CODE
and Hotel_code= H_code;

if total_r is null then
dbms_output.put_line('This hotel does not have any Service revenue yet');
else
dbms_output.put_line('Hotel code: '||H_code|| '    |    Total revenue from
services:  $'||Total_r);
```

```
end if;
exception
when no_hotel then
dbms_output.put_line('This hotel is either sold or does not exist.');
end;
```




```
set SERVEROUTPUT ON
execute service_revenue (1005);
```

```
32  set SERVEROUTPUT ON
33  execute service_revenue (1005);
34
```

Script Output

Task completed in 0.182 seconds

```
Hotel code: 1005    |    Total revenue from services:  $35


PL/SQL procedure successfully completed.
```

```
set SERVEROUTPUT ON
execute service_revenue (1009);
```

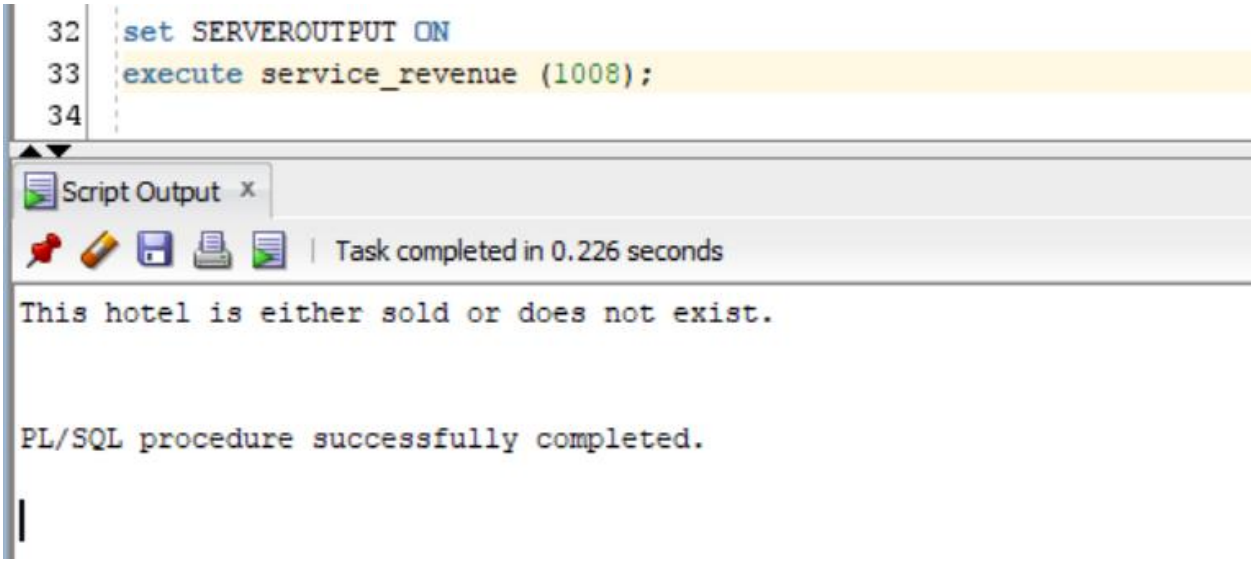
```
32  set SERVEROUTPUT ON
33  execute service_revenue (1009);
34
```

Script Output

Task completed in 0.21 seconds

```
This hotel does not have any Service revenue yet


PL/SQL procedure successfully completed.
```

```
set SERVEROUTPUT ON
execute service_revenue (1008);
```



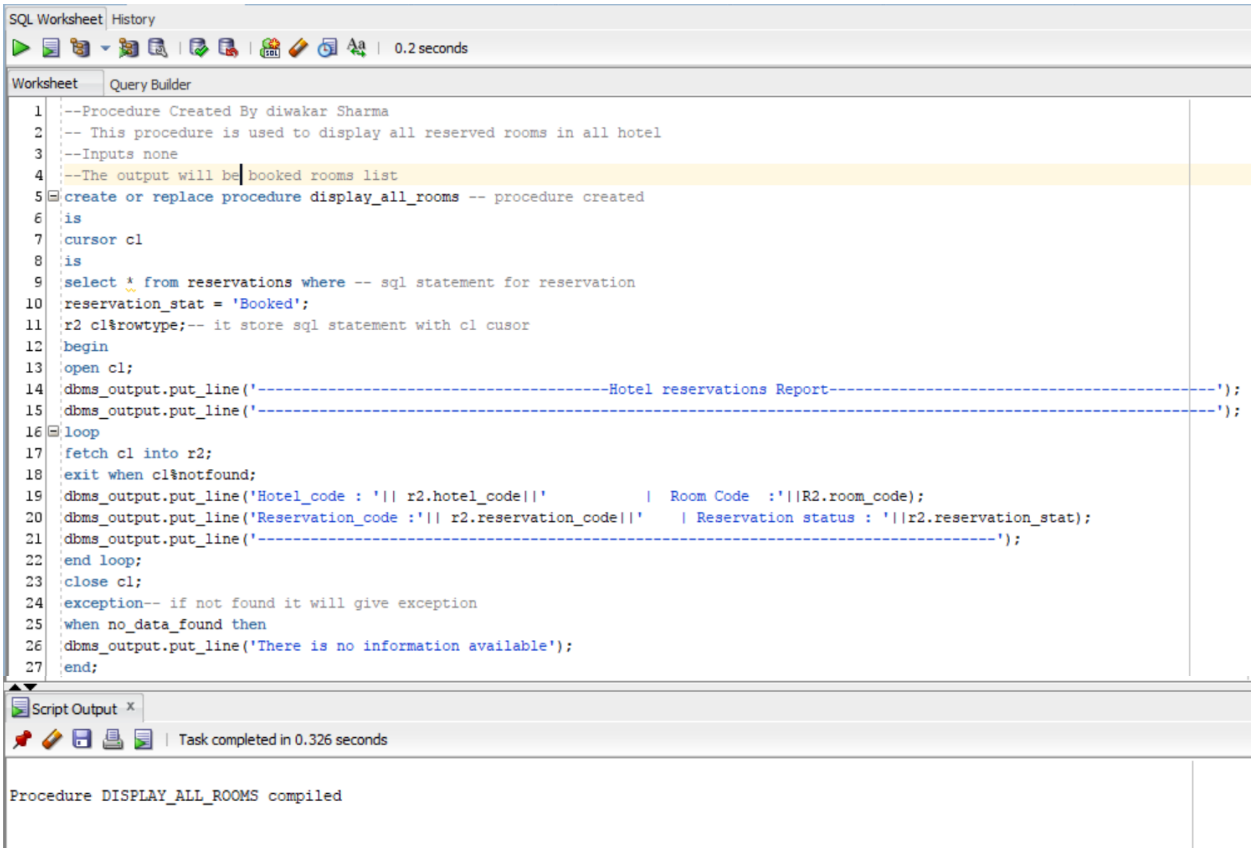
Member 5: Split

20. Show reservations: display all rooms reserved in all hotels

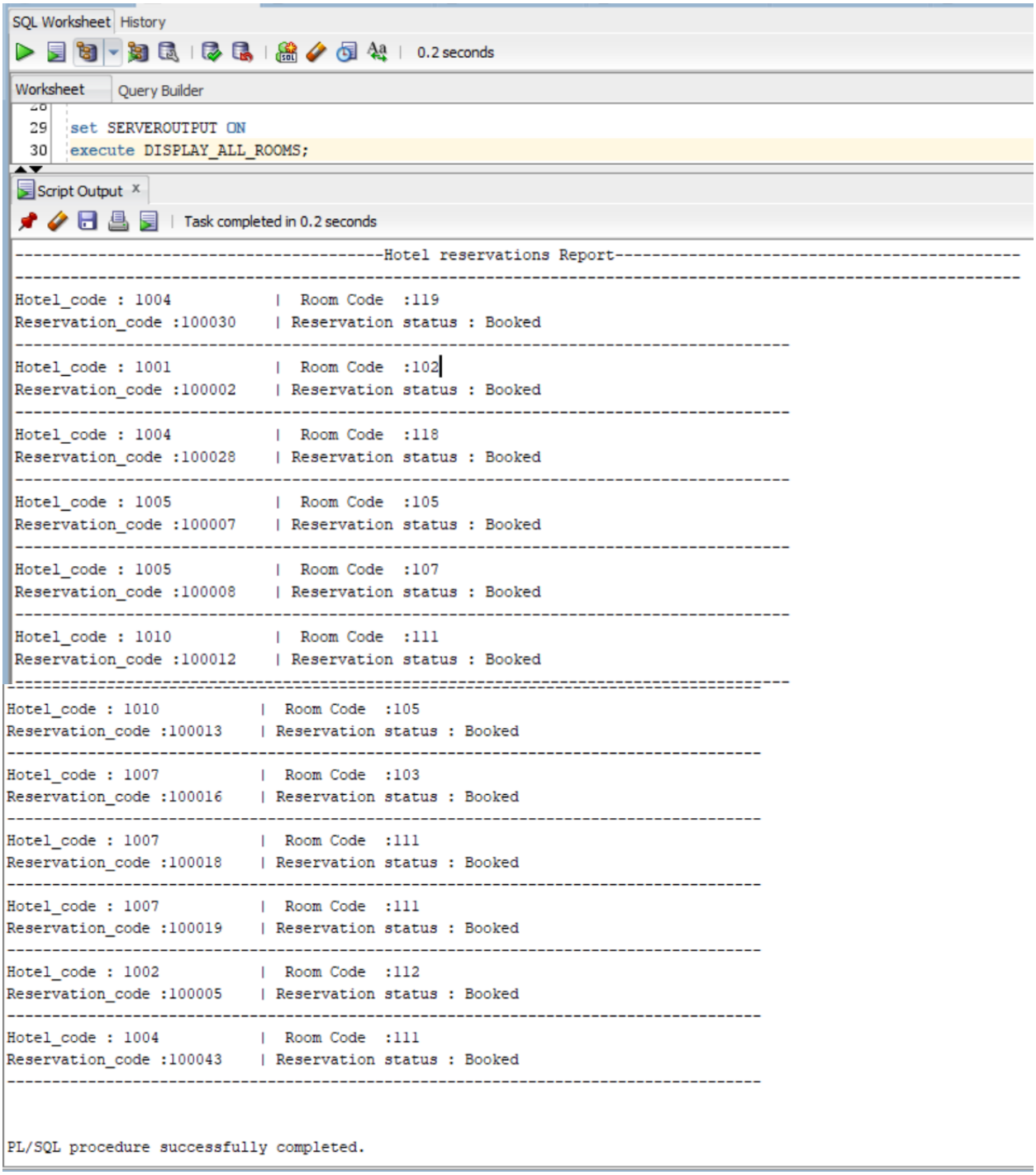
----------Diwakar Sharma----------

```
--Procedure Created By diwakar Sharma
-- This procedure is used to display all reserved rooms in all hotel
--Inputs none
--The output will be booked rooms list
create or replace procedure display_all_rooms -- procedure created
is
cursor c1
is
select * from reservations where -- sql statement for reservation
reservation_stat = 'Booked';
r2 c1%rowtype;-- it store sql statement with c1 cusor
begin
open c1;
dbms_output.put_line('----------------------------------------Hotel
reservations Report-----------------------------------------');
dbms_output.put_line('------------------------------------------------------
-------------------------------------------------');
loop
fetch c1 into r2;
exit when c1%notfound;
dbms_output.put_line('Hotel_code : '|| r2.hotel_code||'           |  Room
Code  :'||R2.room_code);
dbms_output.put_line('Reservation_code :'|| r2.reservation_code||'    |
Reservation status : '||r2.reservation_stat);
dbms_output.put_line('------------------------------------------------------
----------------------------');
end loop;
close c1;
exception-- if not found it will give exception
when no_data_found then
```

```
dbms_output.put_line('There is no information available');
end;
```




```
set SERVEROUTPUT ON
execute DISPLAY_ALL_ROOMS;
```

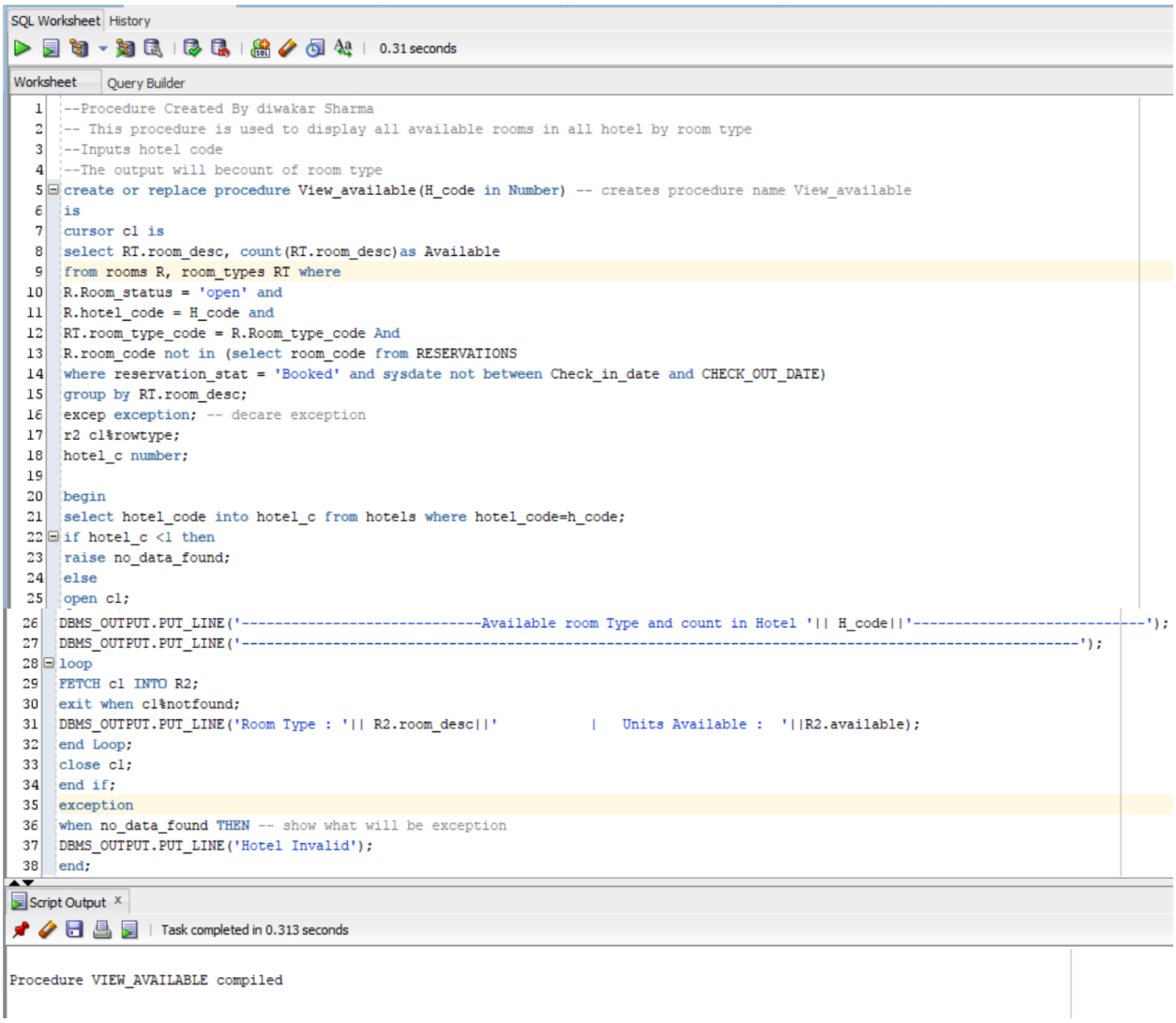


21. Show available rooms by type: Given a hotel ID, display the count of all available rooms by room type.

----------Diwakar Sharma----------

```
--Procedure Created By diwakar Sharma
-- This procedure is used to display all available rooms in all hotel by room type
```

```
--Inputs hotel code
--The output will becount of room type
create or replace procedure View_available(H_code in Number) -- creates
procedure name View_available
is
cursor c1 is
select RT.room_desc, count(RT.room_desc)as Available
from rooms R, room_types RT where
R.Room_status = 'open' and
R.hotel_code = H_code and
RT.room_type_code = R.Room_type_code And
R.room_code not in (select room_code from RESERVATIONS
where reservation_stat = 'Booked' and sysdate not between Check_in_date and
CHECK_OUT_DATE)
group by RT.room_desc;
excep exception; -- decare exception
r2 c1%rowtype;
hotel_c number;

begin
select hotel_code into hotel_c from hotels where hotel_code=h_code;
if hotel_c <1 then
raise no_data_found;
else
open c1;
DBMS_OUTPUT.PUT_LINE('----------------------------Available room Type and
count in Hotel '|| H_code||'----------------------------');
DBMS_OUTPUT.PUT_LINE('------------------------------------------------------
-------------------------------------------');
loop
FETCH c1 INTO R2;
exit when c1%notfound;
DBMS_OUTPUT.PUT_LINE('Room Type : '|| R2.room_desc||'           |   Units
Available :  '||R2.available);
end Loop;
close c1;
end if;
exception
when no_data_found THEN -- show what will be exception
DBMS_OUTPUT.PUT_LINE('Hotel Invalid');
end;
```




```
set serveroutput on
execute View_available(1001);
```

```
41  set serveroutput on
42  execute View_available(1001);
43
44
```

Script Output | Task completed in 0.202 seconds

```
----------------------------Available room Type and count in Hotel 1001---------------------------
------------------------------------------------------------------------------------------------------
Room Type : king           |   Units Available :  3
Room Type : queen           |   Units Available :  4
Room Type : conference            |   Units Available :  2
Room Type : suite           |   Units Available :  3


PL/SQL procedure successfully completed.
```

```
set serveroutput on
execute View_available(100031);
```

```
41  set serveroutput on
42  execute View_available(100031);
43
44
```

Script Output | Task completed in 0.193 seconds

```
Hotel Invalid


PL/SQL procedure successfully completed.
```

22. Room Checkout Report: Input: ReservationID Output:
    - Guest name
    - Room number, rate per day and possibly multiple rooms (if someone reserved several rooms)
    - Services rendered per date, type, and amount
    - Discounts applied (if any)
    - Total amount to be paid

----------Kabin Dulal----------

This procedure is used to generate Bill/ It does not make a payment

```
--Procedure Created By Kabin Dulal
-- This procedure is used to check out report
--Inputs reservation code
--The output will be a check out report

create or replace procedure Generate_Bill (r_code in number)
is
rate number;
Cust_fname varchar2(30);
Cust_lname varchar2(30);
G_name varchar2(60);
num_days number;
service_total number;
room_total number;
discount number;
Dis number;
Bill_Total number;
Net_total number;
room_num number;
Num_bed number;
Room_D varchar(50);
H_code number;
no_data EXCEPTION;

begin
rate:=Get_rate(r_code);
    if rate<0 then
        raise no_data;
        end if;
    select check_out_date - check_in_date into Num_days from RESERVATIONS
where reservation_code=r_code;

    select hotel_code into H_code from Reservations where
reservation_code=R_code;

    select sum(RS.Service_rate)into service_total from ROOM_SERVICES RS,
service_performed SP where
        RS.service_code=SP.SERVICE_CODE and
        SP.RESERVATION_CODE = r_code;

    select guest_name into G_name from Reservations where reservation_code=
r_code;

     room_total:= num_days *rate;
     Bill_Total:=(room_total + service_total);
     discount:= get_discount(r_code);
     if discount >0 then
     dis:= ((bill_total* discount)/100);
     Net_total:= (Bill_total-dis);
     else
     dis:=0;
     net_total:= Bill_total;
     End if;
```

```
     select H.customer_Fname, H.customer_lname into cust_fname, cust_lname
from H_CUSTOMERS H, RESERVATIONS R where
        H.customer_code = R.customer_code and
        R.reservation_code = r_code;

        select RT.room_desc,RT.No_of_beds, RM.room_code into Room_d,Num_bed,
Room_num
        from reservations R, Rooms RM, Room_types RT where
            R.reservation_code=r_code and
            RM.hotel_code=H_Code and
            R.room_code=RM.room_code and
            RM.room_type_code =RT.Room_type_code;

     dbms_output.put_line('Billing date:  ' || to_char(SYSDATE,'YYYY-MM-DD
HH24:MI:SS'));
     dbms_output.put_line('Reservation_code : ' || R_code||'      |     Hotel
Code : '|| h_code);
      dbms_output.put_line('Customer Name           : ' || Cust_fname ||'
'|| Cust_lname);
      dbms_output.put_line('Guest Name          : ' || G_name);
      dbms_output.put_line('-------------------------------------------------
----');
        dbms_output.put_line('Room Number: ' || Room_num|| ' |   Room Type:
'|| Room_D||'  |   Number of Beds : '|| Num_bed);

     dbms_output.put_line('Room Total for  ' || Num_days ||' days * $'
||rate||'/day  : $'|| room_total);
     dbms_output.put_line('Service Total                      :  $' ||
service_total);
     dbms_output.put_line('-------------------------------------------------
---');
     dbms_output.put_line('Bill Total: ----------------> $' || Bill_Total);
      dbms_output.put_line('Discount percent :  ' || discount ||'%      -$'||
dis);

dbms_output.put_line('=====================================================')
;
      dbms_output.put_line('Total Amount to be paid:          $'||
net_total);

     Exception
when no_data then
dbms_output.put_line('This reservation does not Exist');
end;
```

SQL Worksheet | History

0.37200001 seconds

Worksheet | Query Builder

```
1  --Procedure Created By Kabin Dulal
2  -- This procedure is used to check out report
3  --Inputs reservation code
4  --The output will be a check out report
5
6  create or replace procedure Generate_Bill (r_code in number)
7  is
8  rate number;
9  Cust_fname varchar2(30);
10 Cust_lname varchar2(30);
11 G_name varchar2(60);
12 num_days number;
13 service_total number;
14 room_total number;
15 discount number;
16 Dis number;
17 Bill_Total number;
18 Net_total number;
19 room_num number;
20 Num_bed number;
21 Room_D varchar(50);
22 H_code number;
```

```
22 H_code number;
23 no_data EXCEPTION;
24
25 begin
26 rate:=Get_rate(r_code);
27     if rate<0 then
28         raise no_data;
29         end if;
30     select check_out_date - check_in_date into Num_days from RESERVATIONS where reservation_code=r_code;
31
32     select hotel_code into H_code from Reservations where reservation_code=R_code;
33
34     select sum(RS.Service_rate)into service_total from ROOM_SERVICES RS, service_performed SP where
35         RS.service_code=SP.SERVICE_CODE and
36         SP.RESERVATION_CODE = r_code;
37
38     select guest_name into G_name from Reservations where reservation_code= r_code;
39
40      room_total:= num_days *rate;
41      Bill_Total:=(room_total + service_total);
42      discount:= get_discount(r_code);
43      if discount >0 then
```

```
    dis:= ((bill_total* discount)/100);
    Net_total:= (Bill_total-dis);
    else
    dis:=0;
    net_total:= Bill_total;
    End if;

    select H.customer_Fname, H.customer_lname into cust_fname, cust_lname from H_CUSTOMERS H, RESERVATIONS R where
       H.customer_code = R.customer_code and
       R.reservation_code = r_code;

       select RT.room_desc,RT.No_of_beds, RM.room_code into Room_d,Num_bed, Room_num
       from reservations R, Rooms RM, Room_types RT where
           R.reservation_code=r_code and
           RM.hotel_code=H_Code and
           R.room_code=RM.room_code and
           RM.room_type_code =RT.Room_type_code;

    dbms_output.put_line('Billing date:  ' || to_char(SYSDATE,'YYYY-MM-DD HH24:MI:SS'));
    dbms_output.put_line('Reservation_code : ' || R_code||'      |     Hotel Code : '|| h_code);
     dbms_output.put_line('Customer Name           : ' || Cust_fname ||'  '|| Cust_lname);
     dbms_output.put_line('Guest Name          : ' || G_name);
     dbms_output.put_line('-----------------------------------------------------');
       dbms_output.put_line('Room Number: ' || Room_num|| ' |   Room Type: '|| Room_D||'  |   Number of Beds : '|| Num_bed);

    dbms_output.put_line('Room Total for  ' || Num_days ||' days * $' ||rate||'/day  : $'|| room_total);
    dbms_output.put_line('Service Total                    :  $' || service_total);
```

```
      dbms_output.put_line('-----------------------------------------------------');
      dbms_output.put_line('Bill Total: ----------------> $' || Bill_Total);
       dbms_output.put_line('Discount percent :  ' || discount ||'%      -$'|| dis);
       dbms_output.put_line('=====================================================');
       dbms_output.put_line('Total Amount to be paid:          $'|| net_total);

      Exception
when no_data then
dbms_output.put_line('This reservation does not Exist');
end;
```

Script Output ×

Task completed in 0.372 seconds

```
Procedure GENERATE_BILL compiled
```

```
set SERVEROUTPUT ON
execute generate(100002);
```

```
83  set SERVEROUTPUT ON
84  execute generate(100002);
85
```

Script Output | Task completed in 0.232 seconds

```
Room Number: 102 |   Room Type: king  |   Number of Beds : 2
Room Total for  1 days * $300/day           : $300
============================================================
Service Total                                :  $55
03-MAR-18   restaurant service|   rate: $20
03-MAR-18   restaurant service|   rate: $20
03-MAR-18   movie service|   rate: $5
03-MAR-18   laundry services|   rate: $10
============================================================
Bill Total: ----------------> $355
Discount percent :  10%      -$35.5
============================================================
Total Amount to be paid                          :$319.5


PL/SQL procedure successfully completed.
```

```
set SERVEROUTPUT ON
execute generate(12302);
```

```
83  set SERVEROUTPUT ON
84  execute generate(12302);
85
```

Script Output | Task completed in 0.197 seconds

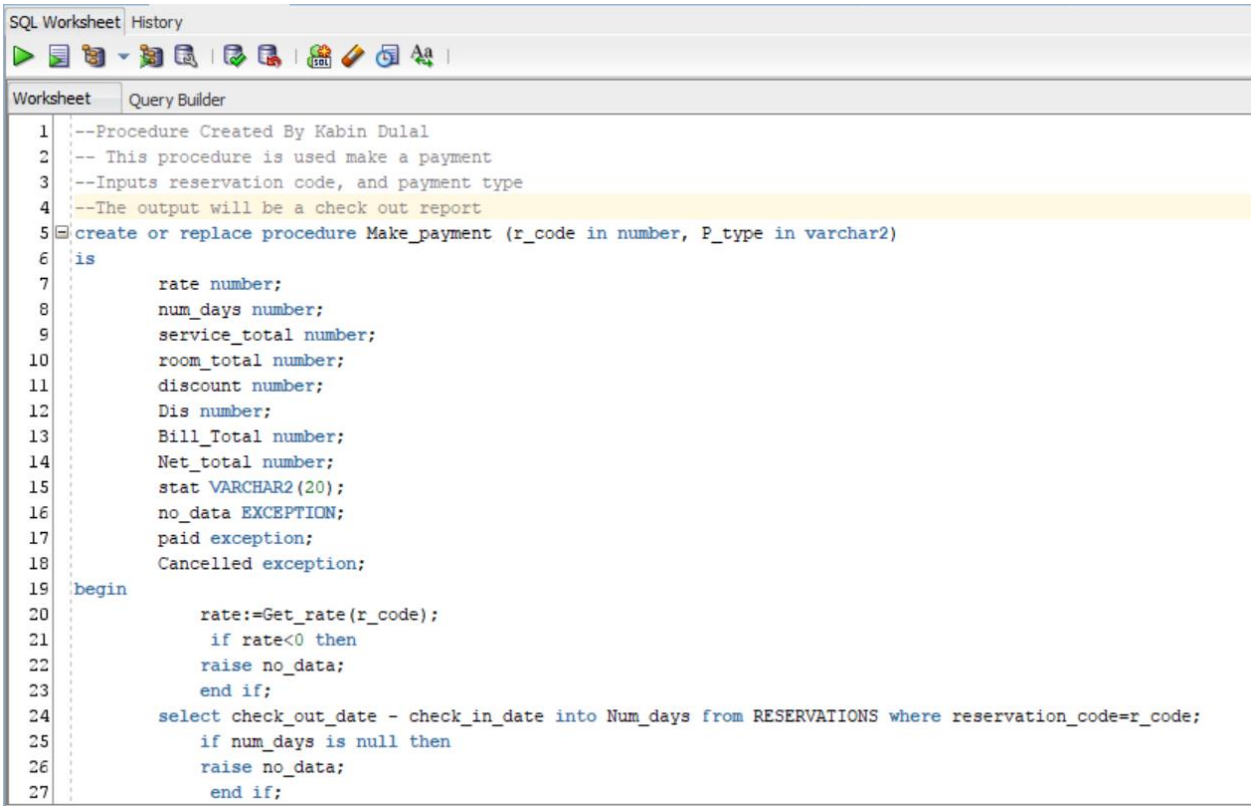
```
This reservation does not Exist


PL/SQL procedure successfully completed.
```

The Following procedure is used to make a payment/ This inserts the data to Bill

```
--Procedure Created By Kabin Dulal
-- This procedure is used make a payment
--Inputs reservation code, and payment type
--The output will be a check out report
```

```
create or replace procedure Make_payment (r_code in number, P_type in
varchar2)
is
        rate number;
        num_days number;
        service_total number;
        room_total number;
        discount number;
        Dis number;
        Bill_Total number;
        Net_total number;
        stat VARCHAR2(20);
        no_data EXCEPTION;
        paid exception;
        Cancelled exception;
begin
            rate:=Get_rate(r_code);
             if rate<0 then
            raise no_data;
            end if;
        select check_out_date - check_in_date into Num_days from RESERVATIONS
where reservation_code=r_code;
            if num_days is null then
            raise no_data;
             end if;
        select sum(RS.Service_rate)into service_total from ROOM_SERVICES RS,
service_performed SP where
            RS.service_code=SP.SERVICE_CODE and
            SP.RESERVATION_CODE = r_code;
        if service_total is null then
            service_total:=0;
         end if;
         select Reservation_stat into stat from reservations where
reservation_code=R_code;
         If stat='checked out' then
         raise paid;
         elsif stat='Cancelled' then
         raise Cancelled;
         end if;

         room_total:= num_days *rate;
         Bill_Total:=(room_total + service_total);
         discount:= get_discount(r_code);
         if discount >0 then
         dis:= ((bill_total* discount)/100);
         Net_total:= (Bill_total- dis);
         else
         dis:=0;
         net_total:= Bill_total;
         End if;

         Insert into Bills values ( Billing_code_seq.nextval,
r_code,room_total,service_total,Bill_total, dis,Net_total,P_type, sysdate);
         update reservations set reservation_stat = 'checked out' where
reservation_code=R_code;
         dbms_output.put_line('Payment for Reservation   ' ||r_code||' is
completed on  '|| to_char(SYSDATE,'YYYY-MM-DD HH24:MI:SS'));
```

```
        dbms_output.put_line('Thank You for the payment');

        Exception
           when no_data then
           dbms_output.put_line('This reservation does not Exist');
           when paid then
           dbms_output.put_line('Reservation is Already paid');
           when Cancelled then
           dbms_output.put_line('Reservation is cancelled, No payment
necessary');

end;
```



```
28          select sum(RS.Service_rate)into service_total from ROOM_SERVICES RS, service_performed SP where
29              RS.service_code=SP.SERVICE_CODE and
30              SP.RESERVATION_CODE = r_code;
31          if service_total is null then
32              service_total:=0;
33           end if;
34           select Reservation_stat into stat from reservations where reservation_code=R_code;
35           If stat='checked out' then
36           raise paid;
37           elsif stat='Cancelled' then
38           raise Cancelled;
39           end if;
40
41           room_total:= num_days *rate;
42           Bill_Total:=(room_total + service_total);
43           discount:= get_discount(r_code);
44           if discount >0 then
45           dis:= ((bill_total* discount)/100);
46           Net_total:= (Bill_total- dis);
47           else
48           dis:=0;
49           net_total:= Bill_total;
50           End if;
51
52           Insert into Bills values ( Billing_code_seq.nextval, r_code,room_total,service_total,Bill_total, dis,Net_total,P_type, sysdate);
53           update reservations set reservation_stat = 'checked out' where reservation_code=R_code;
54           dbms_output.put_line('Payment for Reservation   ' ||r_code||' is completed on  '|| to_char(SYSDATE,'YYYY-MM-DD HH24:MI:SS'));
```

```
55            dbms_output.put_line('Thank You for the payment');
56
57            Exception
58               when no_data then
59               dbms_output.put_line('This reservation does not Exist');
60               when paid then
61               dbms_output.put_line('Reservation is Already paid');
62               when Cancelled then
63               dbms_output.put_line('Reservation is cancelled, No payment necessary');
64
65    end;
```

Script Output

Task completed in 0.156 seconds

```
Procedure MAKE_PAYMENT compiled
```

```
Set SERVEROUTPUT ON
execute MAKE_PAYMENT(100029,'cash');
```

```
67  Set SERVEROUTPUT ON
68  execute MAKE_PAYMENT(100029,'cash');
```

Script Output

Task completed in 0.195 seconds

```
Payment for Reservation   100029 is completed on  2018-05-12 21:52:47
Thank You for the payment


PL/SQL procedure successfully completed.
```

```
Set SERVEROUTPUT ON
execute MAKE_PAYMENT(10012,'cash');
```

```
67  Set SERVEROUTPUT ON
68  execute MAKE_PAYMENT(10012,'cash');
```

Script Output x

Task completed in 0.183 seconds

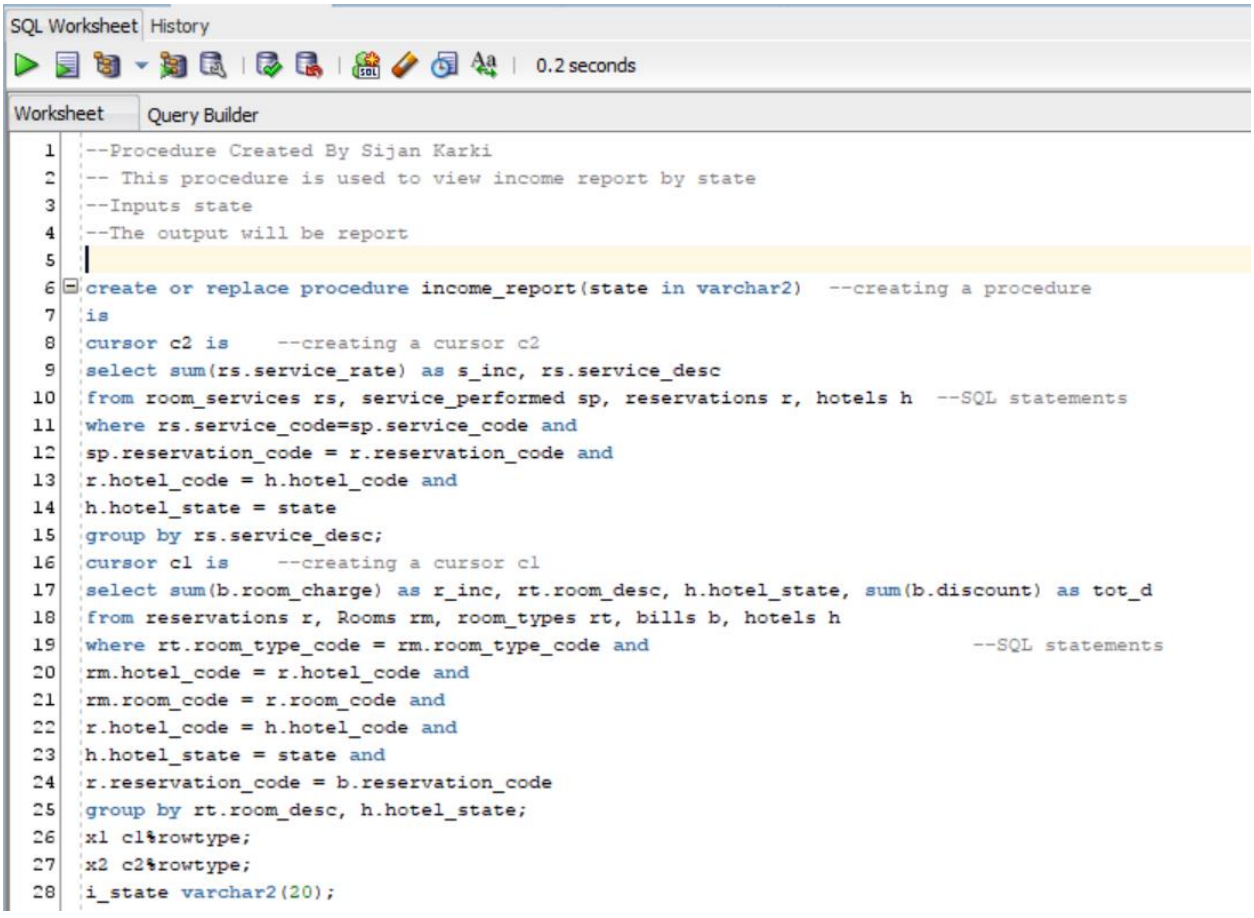
```
This reservation does not Exist


PL/SQL procedure successfully completed.
```

23. Income By State Report: Input is state. Print total income from all sources of all hotels by room type and service type in the given state. Include discounts.

----------**Sijan karki**----------

```
--Procedure Created By Sijan Karki
-- This procedure is used to view income report by state
--Inputs state
--The output will be report

create or replace procedure income_report(state in varchar2)  --creating a
procedure
is
cursor c2 is    --creating a cursor c2
select sum(rs.service_rate) as s_inc, rs.service_desc
from room_services rs, service_performed sp, reservations r, hotels h  --SQL
statements
where rs.service_code=sp.service_code and
sp.reservation_code = r.reservation_code and
r.hotel_code = h.hotel_code and
h.hotel_state = state
group by rs.service_desc;
cursor c1 is    --creating a cursor c1
select sum(b.room_charge) as r_inc, rt.room_desc, h.hotel_state,
sum(b.discount) as tot_d
from reservations r, Rooms rm, room_types rt, bills b, hotels h
where rt.room_type_code = rm.room_type_code and                          --
SQL statements
rm.hotel_code = r.hotel_code and
rm.room_code = r.room_code and
r.hotel_code = h.hotel_code and
h.hotel_state = state and
r.reservation_code = b.reservation_code
group by rt.room_desc, h.hotel_state;
x1 c1%rowtype;
x2 c2%rowtype;
i_state varchar2(20);
No_hotel exception;
Begin
select distinct hotel_state into i_state from hotels where hotel_state =
state;
if i_state is null then      --conditional statement
raise no_data_found;
else

dbms_output.put_line('*****************************************************
**********');
dbms_output.put_line('-------------------------Income report for
"'||state||'"----------------------------');
dbms_output.put_line('*****************************************************
**********');
dbms_output.put_line('INCOME FROM ROOM RESERVATIONS');
open c1;
loop
fetch c1 into x1;  --c1 is fetched into x1
exit when c1%notfound;
```

```
dbms_output.put_line(x1.room_desc||'          : $'||x1.r_inc|| '
| Discount    :  $'|| x1.tot_d);
end Loop;
close c1;
dbms_output.put_line('*****************************************************
**********************');
dbms_output.put_line('INCOME FROM ROOM SERVICES');
open c2;
loop
fetch c2 into x2;   --c1 is fetched into x2
exit when c2%notfound;
dbms_output.put_line(x2.service_desc||'          : $'||x2.S_inc);
end Loop;
end if;
exception    --exception handeling
when no_data_found then
dbms_output.put_line('No hotels in this state or the state is Invalid');
end;
```



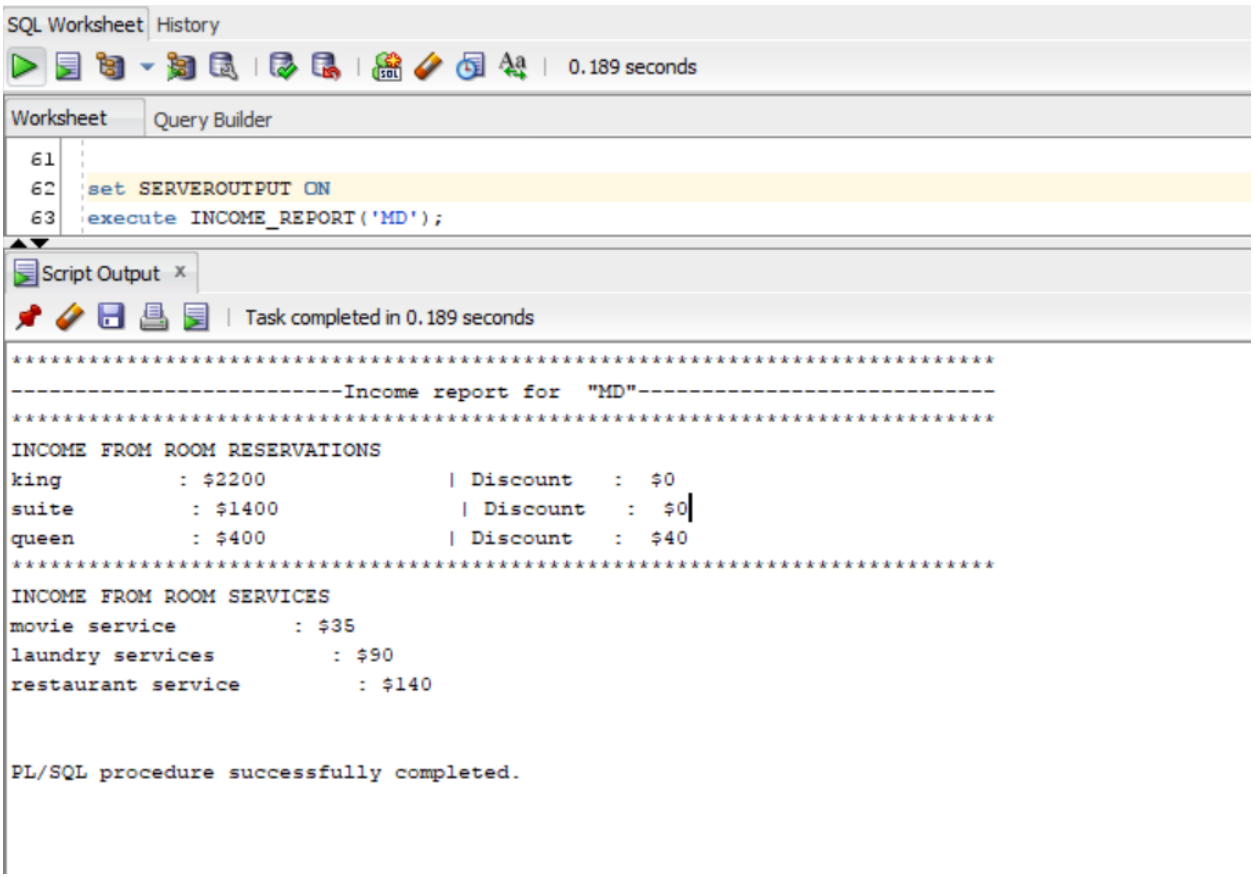
```
No_hotel exception;
Begin
select distinct hotel_state into i_state from hotels where hotel_state = state;
if i_state is null then      --conditional statement
raise no_data_found;
else

dbms_output.put_line('*********************************************************************************');
dbms_output.put_line('-------------------------Income report for  "'||state||'"----------------------------');
dbms_output.put_line('*********************************************************************************');
dbms_output.put_line('INCOME FROM ROOM RESERVATIONS');
open c1;
loop
fetch c1 into x1;  --c1 is fetched into x1
exit when c1%notfound;
dbms_output.put_line(x1.room_desc||'         : $'||x1.r_inc|| '              | Discount   :  $'|| x1.tot_d);
end Loop;
close c1;
dbms_output.put_line('*********************************************************************************');
dbms_output.put_line('INCOME FROM ROOM SERVICES');
open c2;
loop
fetch c2 into x2;   --c1 is fetched into x2
exit when c2%notfound;
dbms_output.put_line(x2.service_desc||'         : $'||x2.S_inc);
end Loop;
end if;
exception    --exception handeling
when no_data_found then
dbms_output.put_line('No hotels in this state or the state is Invalid');
end;
```

Script Output

Task completed in 0.135 seconds

```
Procedure INCOME_REPORT compiled
```

```
set SERVEROUTPUT ON
execute INCOME_REPORT('MD');
```




```
set SERVEROUTPUT ON
execute INCOME_REPORT('FL');
```