Team Java Lounge

IS420: Database Application Development

Deliverable IV

December 5th, 2017

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Charles Douglas

Grayson King

Luvneet Singh

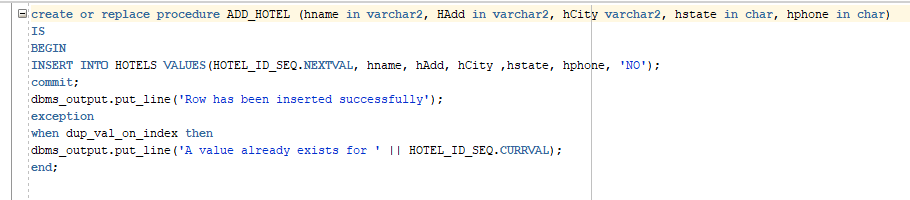
Jay Early

Josiah Tillett

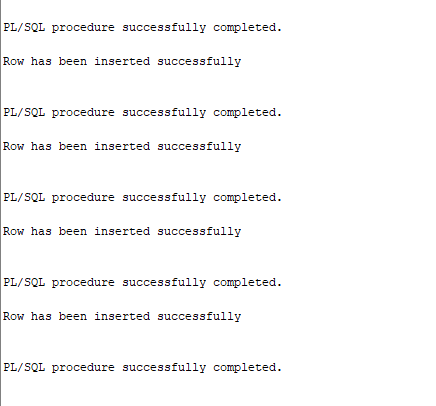
University of Maryland, Baltimore County

College of Engineering and Information Systems

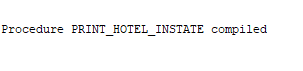
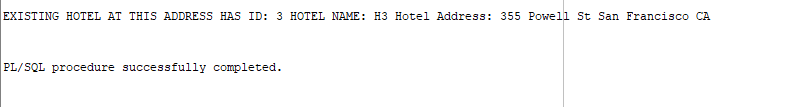
Jay Early {1, 2, 3, 13}

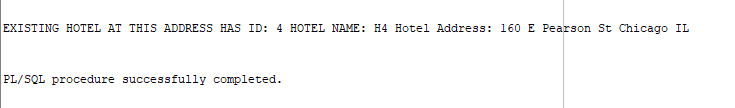
1.[\*] Add a new hotel: Create a new hotel with appropriate information about the hotel as input parameters   




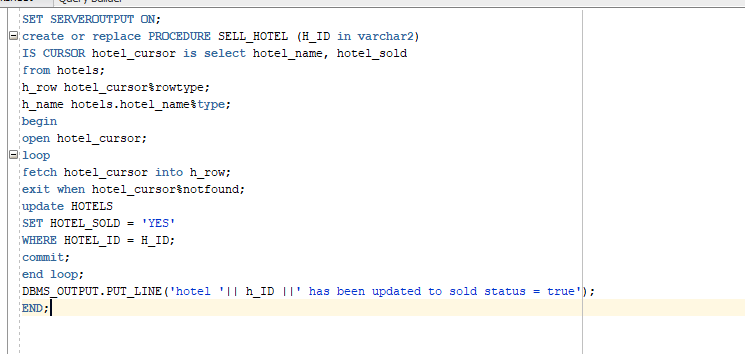


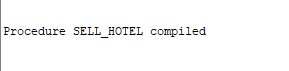
2. [\*] Find a hotel: Provide as input the address of the hotel and return its hotel ID  

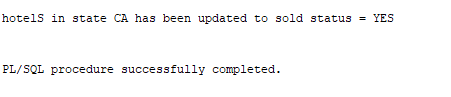

  


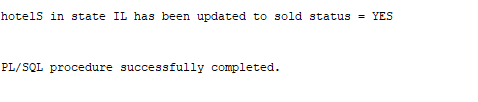


3. [\*] Sell existing hotel: Sell a hotel by providing its hotel ID. Mark it as sold, do not delete the record.









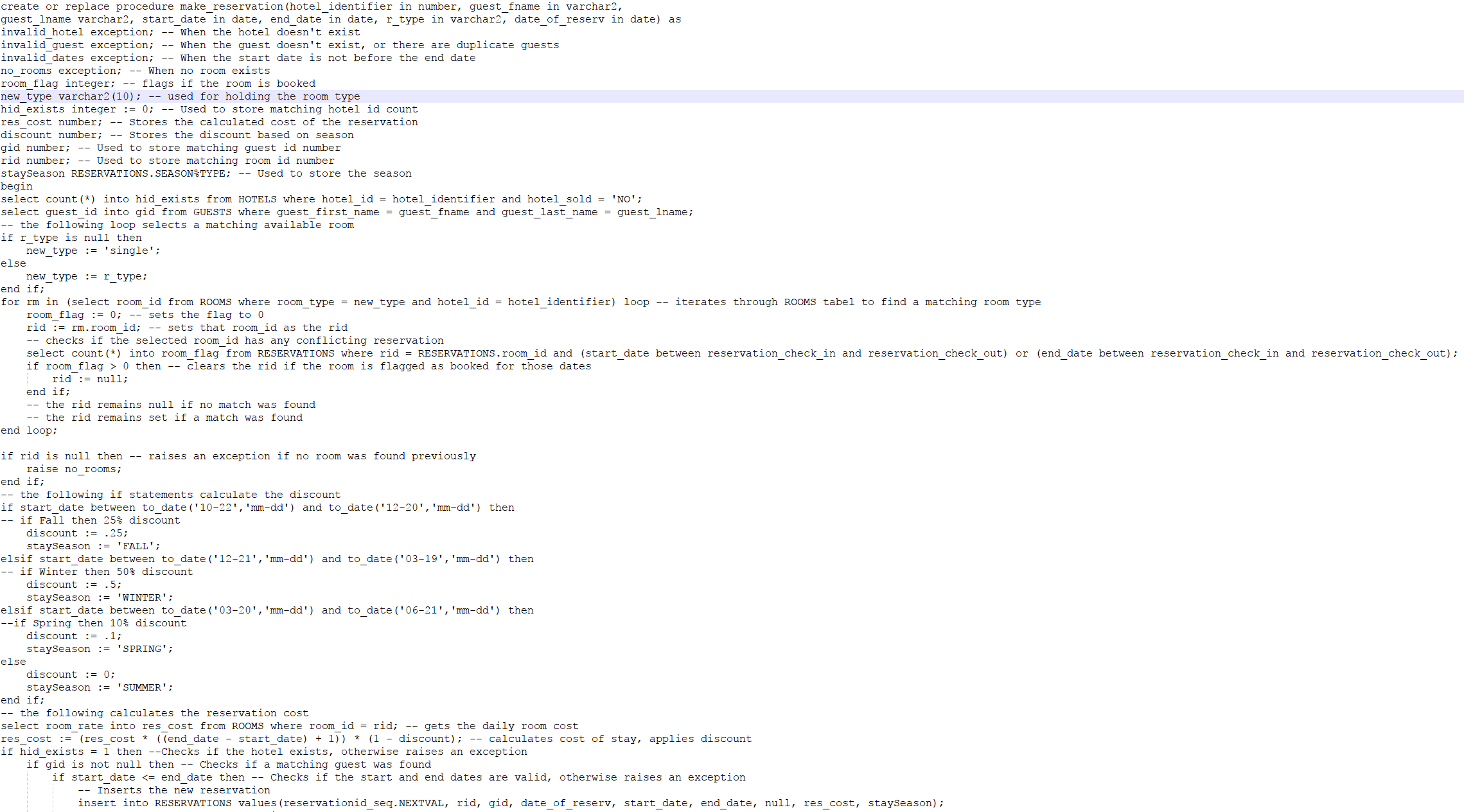


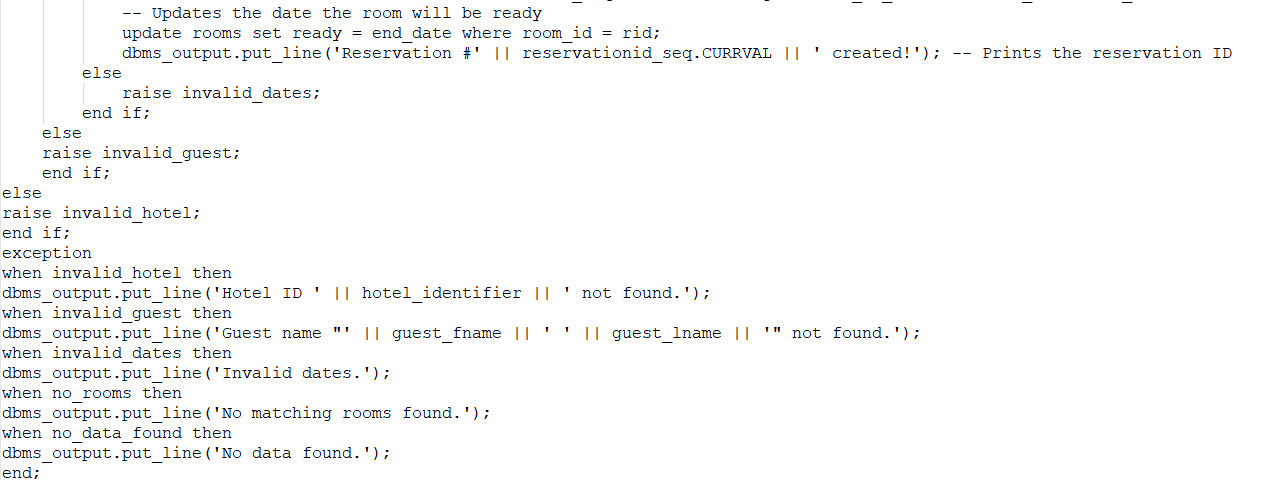
4. [\*\*] Make a reservation: Input parameters: Hotel, guest’s name, start date, end dates,

room type, date of reservation, etc. Output: reservation ID (this is called confirmation

code in real-life ).

Added by Grayson

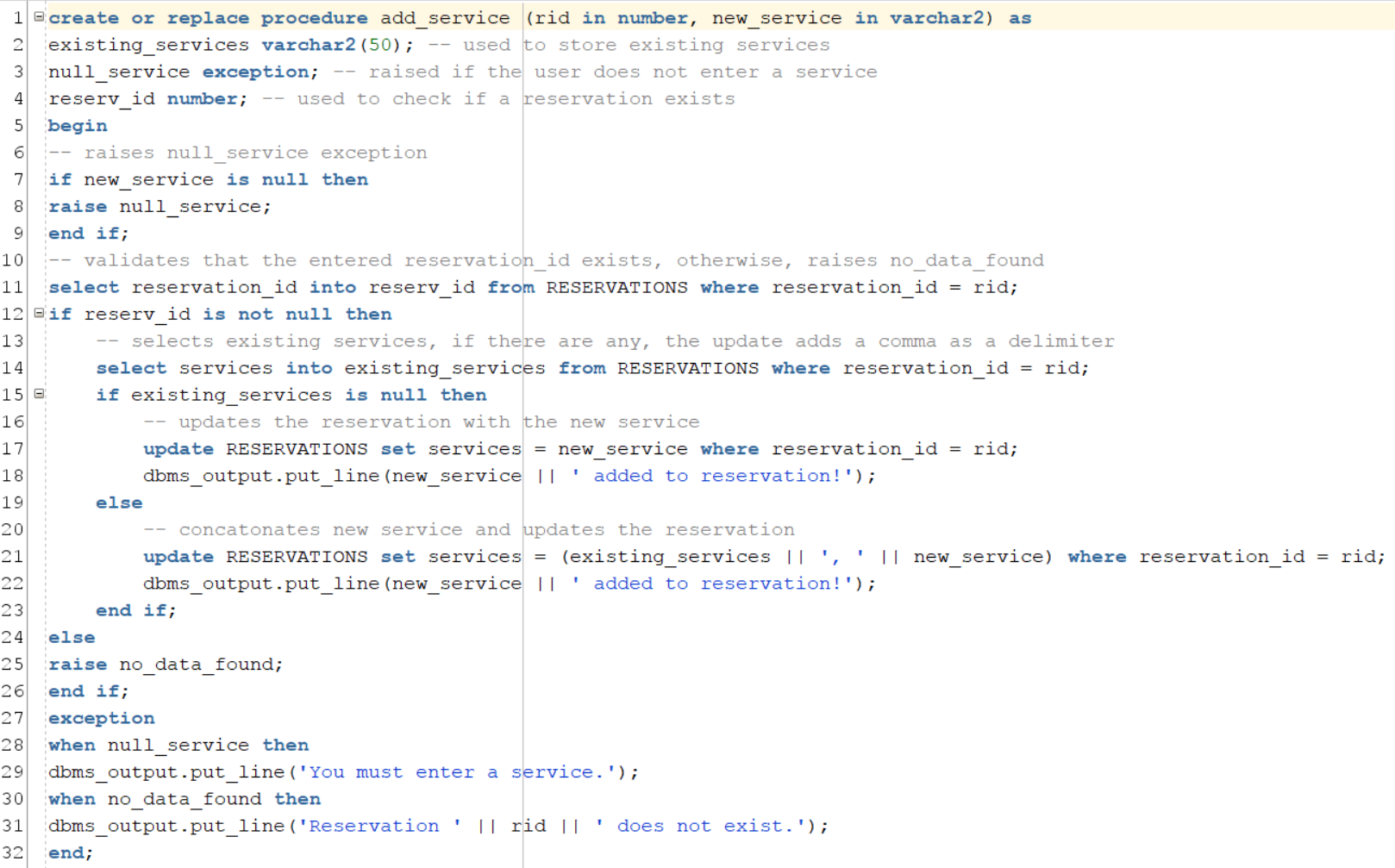




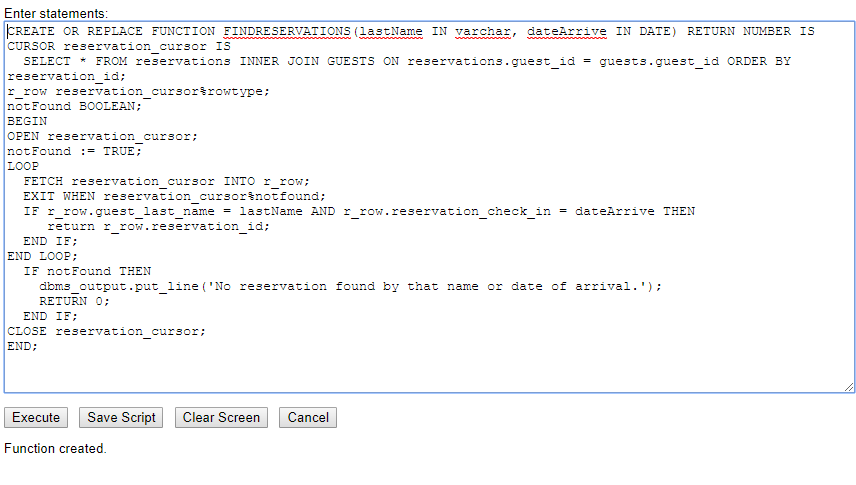


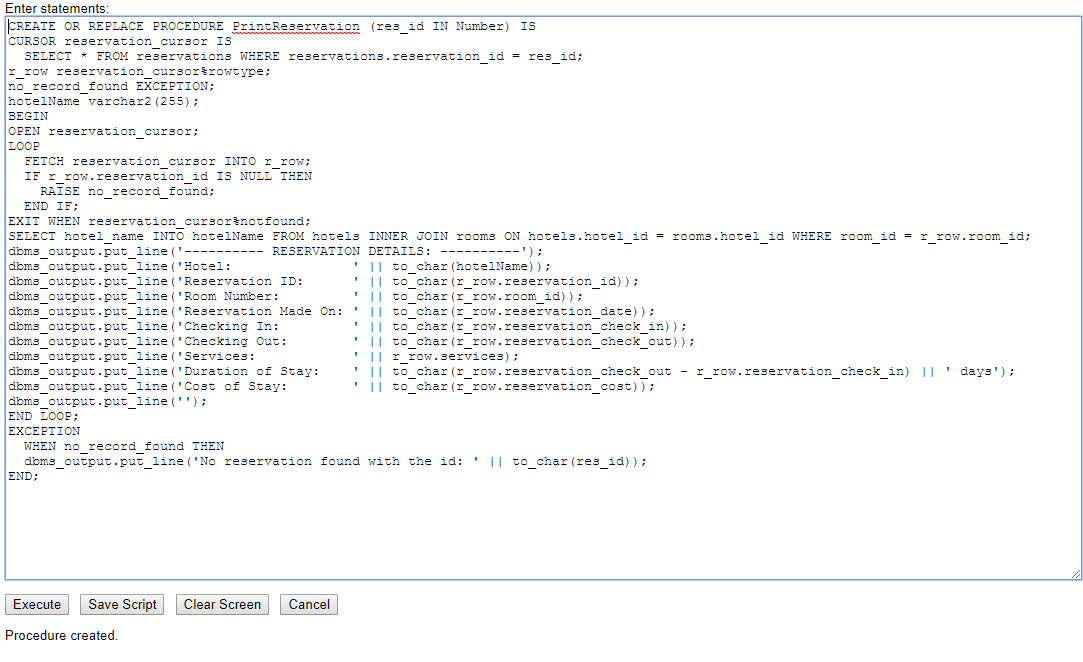
6. [\*] Add a service to a reservation: Input: ReservationID, specific service. Add it to the reservation for a particular date. Multiple services are allowed on a reservation for the same date

Added by Grayson

.

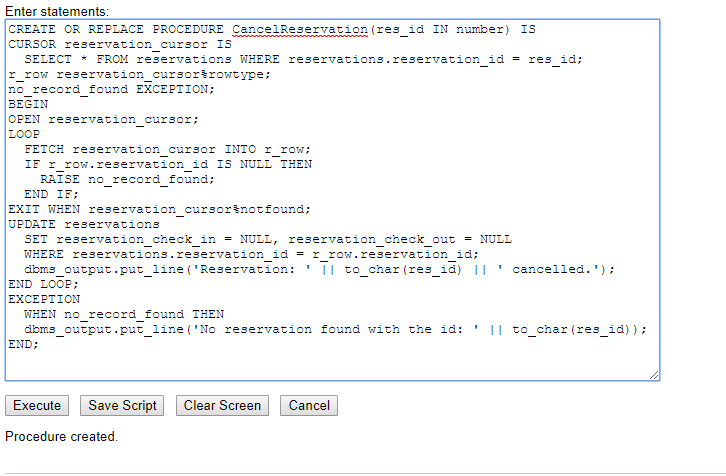


**7. FindReservation & PrintReservation Josiah (\*)**



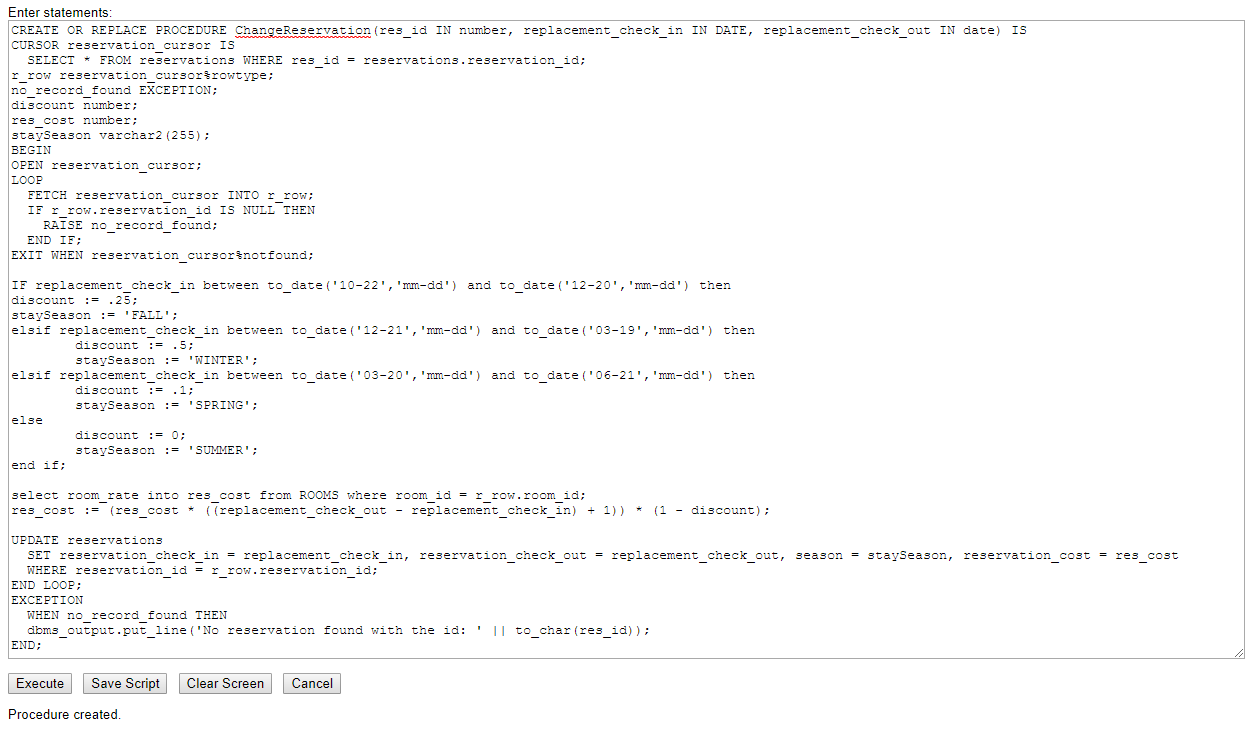
---------------------------

**8. CancelReservation Josiah (\*)**

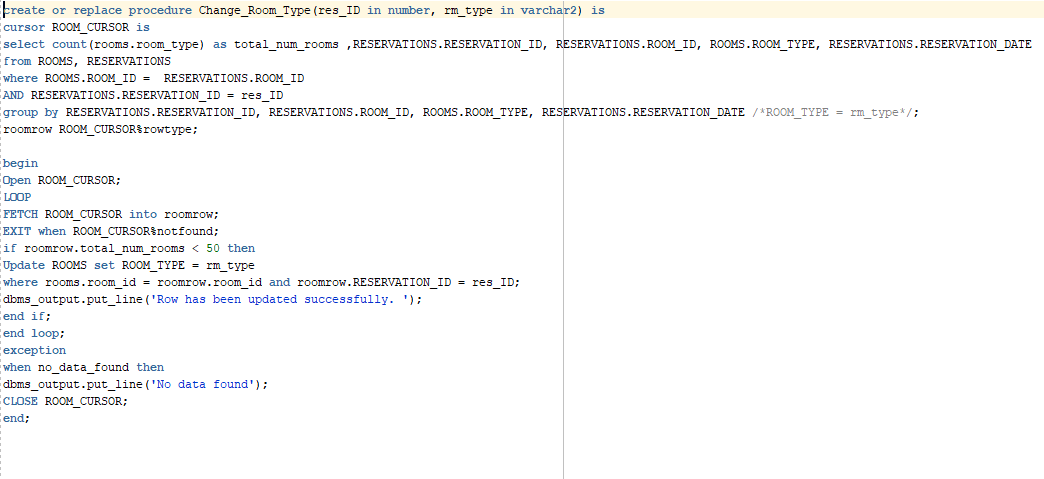


-------------------------------------------------

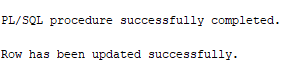
**9. ChangeReservation Josiah (\*\*)**

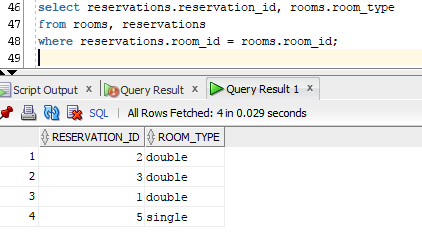


10. \*\*Change\_Room\_Type -charles douglas

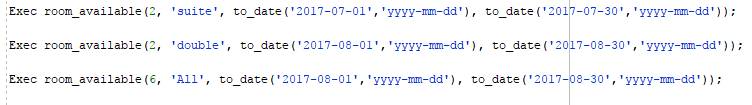








11. \*\*Rooms\_Available -Charles Douglas



12. [\*] AvailabilityOfRoomHotelPerInterval: Input a room type, hotel ID, date interval and return the number of available rooms during that time interval

create or replace Function GetAvailableRoom --function to get the number of rooms

(H\_ID IN NUMBER, Start\_Date IN DATE, End\_Date IN DATE, r\_type IN VARCHAR2) --assigns what form the values would be in

return number

IS num\_rooms number;

BEGIN

select count(ROOMS.room\_id) into num\_rooms

from hotels, rooms, reservations

where hotels.hotel\_id = rooms.hotel\_id --joins hotels and rooms

and ROOMS.room\_id = reservations.ROOM\_ID --joins reservations and rooms

and hotels.hotel\_id = h\_id --sets hotels.hotel\_id to h\_id

AND ROOMS.ROOM\_TYPE = r\_type --sets rooms.room\_type to r\_type

and RESERVATIONS.RESERVATION\_CHECK\_IN NOT between start\_date and end\_date

and RESERVATIONS.RESERVATION\_CHECK\_OUT NOT BETWEEN start\_date and end\_date; --makes sure check in and check out aren’t between their dates

RETURN NUM\_ROOMS;

END;

declare --uses the number of rooms to see which rooms are available

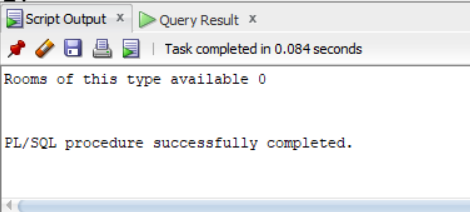
total\_num number;

begin

total\_num := GetAvailableRoom(1,to\_date('2017-10-30','YYYY-MM-DD'), to\_date('2018-12-31', 'YYYY-MM-DD'),'economy');

dbms\_output.put\_line('Rooms of this type available ' || total\_num);

end;



13. [\*\*\*] RoomCheckoutReceipt: Input: ReservationID Output:

* 1. Guest name
  2. Room number, rate per day and possibly multiple rooms (if someone reserved several rooms)
  3. Services rendered per date, type, and amount
  4. Discounts applied (if any)
  5. Total amount to be paid   
       
     create or replace procedure Room\_Checkout\_Reciept(resID in number) IS  
     cursor reciept\_cursor is select HOTELS.HOTEL\_ID HOTEL\_ID,  
     HOTELS.HOTEL\_NAME HOTEL\_NAME,  
     HOTELS.HOTEL\_ADDRESS HOTEL\_ADDRESS,  
     HOTELS.HOTEL\_CITY HOTEL\_CITY,  
     HOTELS.HOTEL\_STATE HOTEL\_STATE,  
     HOTELS.HOTEL\_PHONE HOTEL\_PHONE,  
     HOTELS.HOTEL\_SOLD HOTEL\_SOLD,  
     GUESTS.GUEST\_ID GUEST\_ID,  
     GUESTS.GUEST\_FIRST\_NAME,  
     GUESTS.GUEST\_LAST\_NAME,  
     GUESTS.GUEST\_ADDRESS,  
     GUESTS.GUEST\_CITY,  
     GUESTS.GUEST\_STATE,  
     ROOMS.ROOM\_ID ROOM\_ID,  
     ROOMS.HOTEL\_ID HOTEL\_ID\_0,  
     ROOMS.ROOM\_CAPACITY ROOM\_CAPACITY,  
     ROOMS.ROOM\_RATE ROOM\_RATE,  
     ROOMS.READY READY,  
     ROOMS.CLEANED CLEANED,  
     ROOMS.SMOKING SMOKING,  
     ROOMS.ROOM\_TYPE ROOM\_TYPE,  
     RESERVATIONS.RESERVATION\_ID RESERVATION\_ID,  
     RESERVATIONS.ROOM\_ID ROOM\_ID\_1,  
     RESERVATIONS.RESERVATION\_DATE RESERVATION\_DATE,  
     RESERVATIONS.RESERVATION\_CHECK\_IN RESERVATION\_CHECK\_IN,  
     RESERVATIONS.RESERVATION\_CHECK\_OUT RESERVATION\_CHECK\_OUT,  
     RESERVATIONS.SERVICES SERVICES,  
     RESERVATIONS.SEASON,  
     TRANSACTIONS.TRANSACTION\_ID TRANSACTION\_ID,  
     TRANSACTIONS.GUEST\_ID GUEST\_ID\_2,  
     TRANSACTIONS.RESERVATION\_ID RESERVATION\_ID\_3,  
     TRANSACTIONS.CREDIT\_CARD CREDIT\_CARD,  
     TRANSACTIONS.TRANSACTION\_TOTAL TRANSACTION\_TOTAL,  
     TRANSACTIONS.TRANSACTION\_DATE TRANSACTION\_DATE,  
     SERVICES.SERVICE\_NAME, SERVICES.SERVICE\_COST  
     from HOTELS, GUESTS, ROOMS, RESERVATIONS, TRANSACTIONS, SERVICES  
     where HOTELS.HOTEL\_ID = ROOMS.HOTEL\_ID  
     and ROOMS.ROOM\_ID = RESERVATIONS.ROOM\_ID  
     AND RESERVATIONS.RESERVATION\_ID = TRANSACTIONS.RESERVATION\_ID  
     and guests.guest\_id = reservations.guest\_id  
     and RESERVATIONS.RESERVATION\_ID = resID  
     and RESERVATIONS.SERVICES = services.service\_name;  
      -- Denotes argument passed in  
     reciept\_row reciept\_cursor%rowtype;  
     totalAmount number := 0;  
     --services varchar2(50);  
     lengthOfStay number;  
     totalServiceCost number;  
     totalROOM number;  
     discountRate number := 0;  
     discountAmount number:= 0;  
     numRooms number := 1; --Number of Rooms booked (will be changed later)  
     servName SERVICES.SERVICE\_NAME%type;  
     servCost SERVICES.SERVICE\_COST%type;  
     BEGIN  
     open reciept\_cursor;  
     loop  
     fetch reciept\_cursor into reciept\_row;  
     EXIT when reciept\_cursor%NOTFOUND;  
     lengthOfStay := reciept\_row.RESERVATION\_CHECK\_OUT - reciept\_row.RESERVATION\_CHECK\_IN;  
     totalROOM := reciept\_row.ROOM\_RATE \* lengthOfStay;  
     totalServiceCost := reciept\_row.service\_cost \* lengthOfStay;  
     totalAmount := totalROOM + totalServiceCost - discountAmount;  
     IF reciept\_row.SEASON = 'WINTER' THEN -- When season is winter then applicable discount is applied  
     discountRate := .50;  
     discountAmount := discountRate \* totalROOM;  
     end if;  
      if reciept\_row.season = 'SPRING' then --If Spring time then applicable discount is taken into account.  
      discountRate := .10;  
      totalROOM := totalROOM - (discountRate \* totalROOM);  
      end if;  
      if reciept\_row.season = 'SUMMER' then --If Spring time then applicable discount is taken into account.  
      discountRate := .0;  
      totalROOM := totalROOM - (discountRate \* totalROOM);  
      end if;  
     discountAmount := totalROOM \* discountRate;  
       
     END LOOP;  
     if reciept\_row.transaction\_id is null then  
     raise no\_data\_found;  
     else  
     dbms\_output.put\_line('-------------------------- ' || reciept\_row.HOTEL\_NAME || ' -----------------------------');  
     dbms\_output.put\_line(' Guest Reciept ');  
     dbms\_output.put\_line('Transaction # ' || reciept\_row.transaction\_ID);  
     dbms\_output.put\_line('Guest Name: ' || reciept\_row.GUEST\_FIRST\_NAME || ' ' || reciept\_row.GUEST\_LAST\_NAME);  
     dbms\_output.put\_line('ROOM #' || reciept\_row.ROOM\_ID);  
     dbms\_output.put\_line('Rooms Booked: ' || numRooms);  
     dbms\_output.put\_line('Daily Rate: $' || reciept\_row.ROOM\_RATE);  
     dbms\_output.put\_line('Duration of stay: ' || lengthofStay || ' nights ');  
     dbms\_output.put\_line('Services booked: ' || reciept\_row.service\_name);  
     dbms\_output.put\_line('Service Charge p/day: $' || reciept\_row.service\_cost);  
     dbms\_output.put\_line('Room Charge: $'|| totalROOM);  
     dbms\_output.put\_line('Total Services Charge: $'|| totalServiceCost);  
     dbms\_output.put\_line('---------------------------------------------------------------------');  
     dbms\_output.put\_line('Applied Season Discounts: -' || discountAmount || ' (' || reciept\_row.season || ') ');  
     dbms\_output.put\_line('---------------------------------------------------------------------');  
     dbms\_output.put\_line('Total Amount Due: $'|| totalAmount);  
     dbms\_output.put\_line('CREDIT - Payment Applied: (' || totalAmount || ') ');  
     dbms\_output.put\_line('CREDIT CARD XXXX' || reciept\_row.credit\_card);  
     dbms\_output.put\_line('----------------------------------------------------------------------');  
     dbms\_output.put\_line('NET Outstanding Balance: $0.00');  
     dbms\_output.put\_line('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_');  
     end if;  
     exception  
     when no\_data\_found then  
     dbms\_output.put\_line('No data found for this reservation #');  
     close reciept\_cursor;  
     END;

14. \* Sold\_hotels: Print all sold hotel information. Show ID, location, etc.

create or replace PROCEDURE Sold\_Hotel\_Information IS

CURSOR HOTEL\_CURSOR IS --gets values into a cursor

SELECT \*

FROM HOTELS

WHERE HOTEL\_SOLD = 'YES'; --looks for hotels that are sold

HOTEL\_ROW HOTEL\_CURSOR%ROWTYPE;

BEGIN

OPEN HOTEL\_CURSOR;

LOOP --loops through hotel\_cursor

FETCH HOTEL\_CURSOR INTO HOTEL\_ROW; --uses the cursor to look through hotel\_row

EXIT WHEN HOTEL\_CURSOR%NOTFOUND; --exits cursor when no values are found

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(HOTEL\_ROW.HOTEL\_ID || '' || HOTEL\_ROW.HOTEL\_NAME || '' || HOTEL\_ROW.HOTEL\_ADDRESS ||''|| HOTEL\_ROW.HOTEL\_PHONE); --outputs the hotel in order

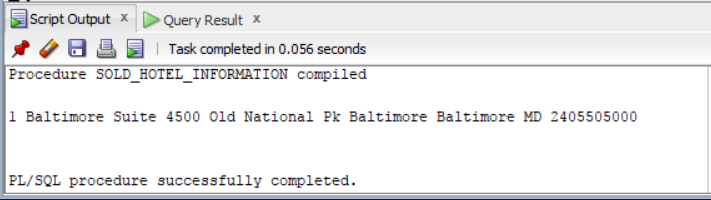
close HOTEL\_CURSOR; --closes the cursor

EXCEPTION --raises an exception for when there is no data found

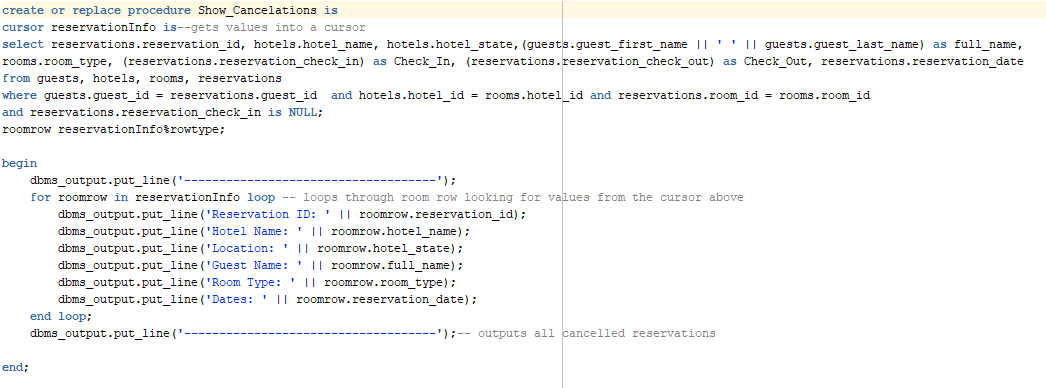
WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Could not locate any Sold Hotels'); --outputs an error if any sold hotels were not found

END;

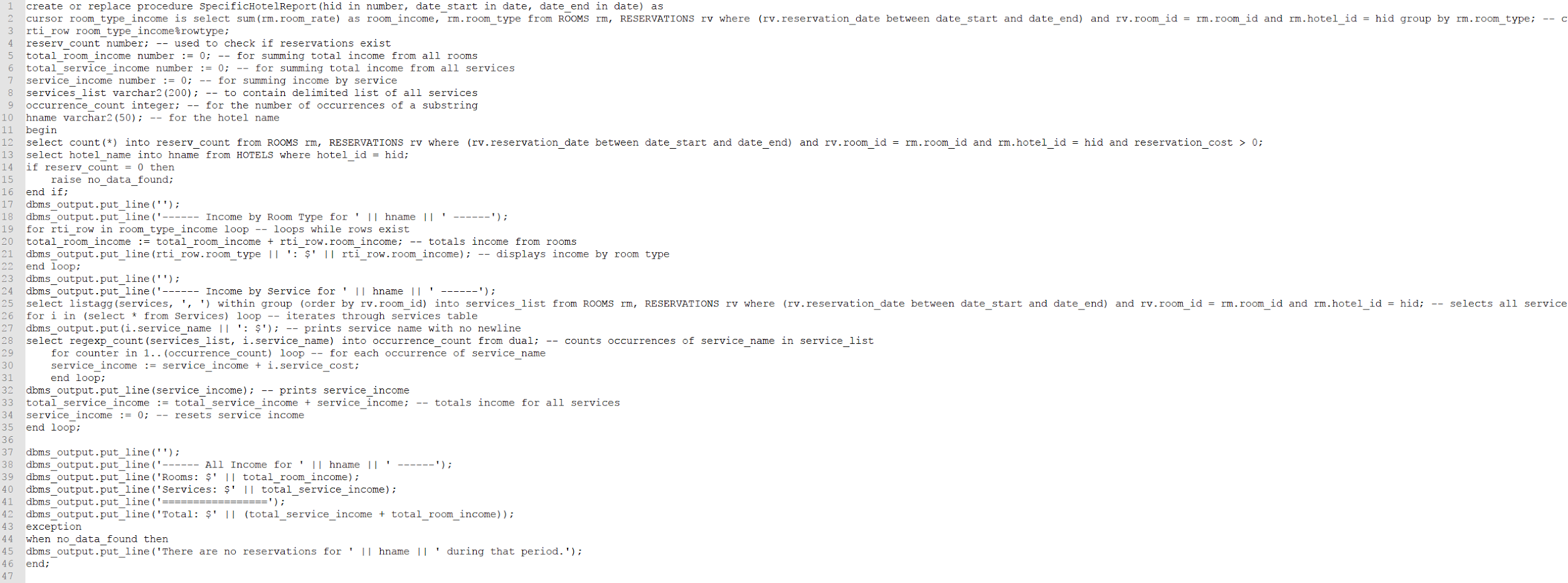


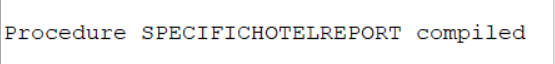
15. \*Show\_Cancelations -charles douglas

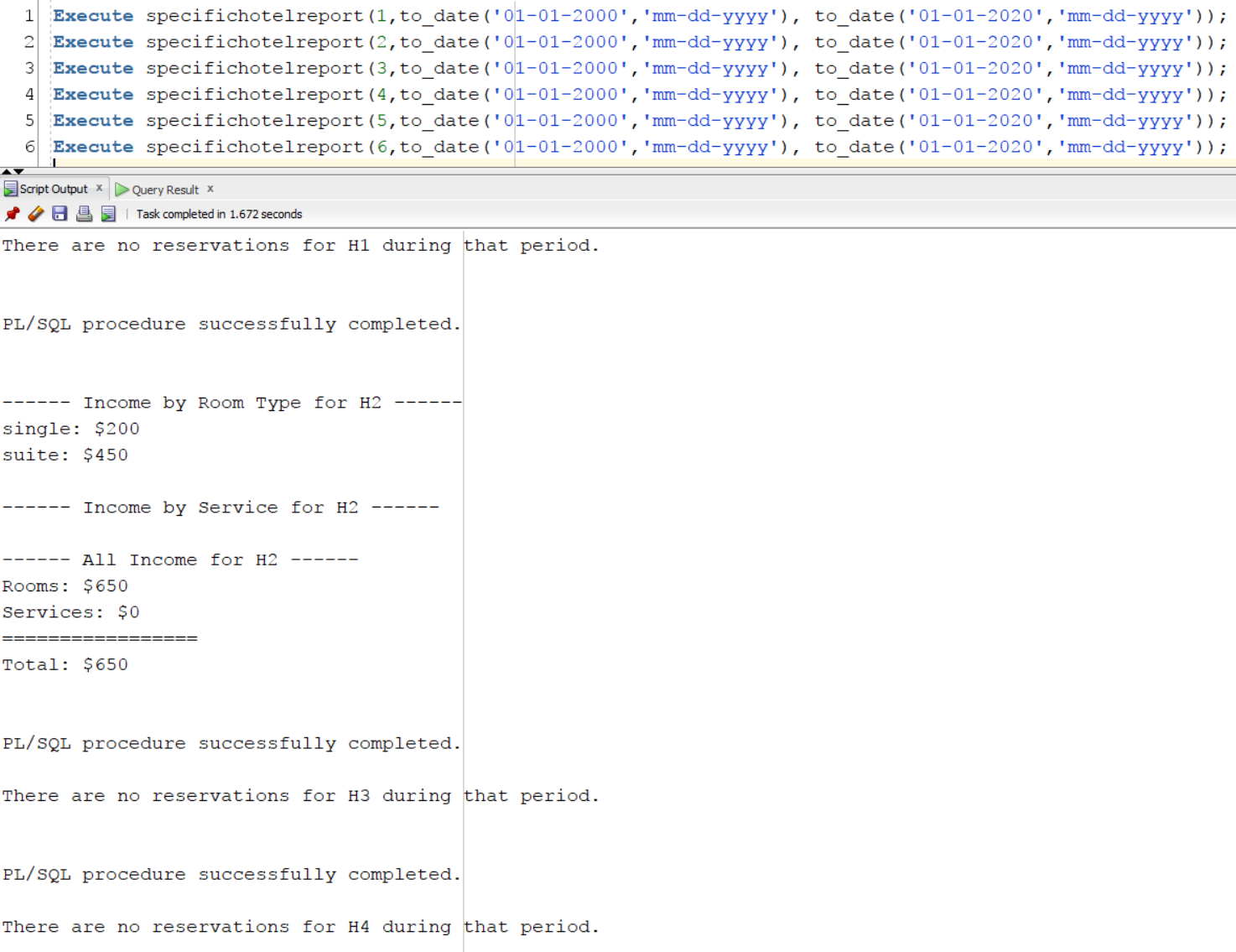




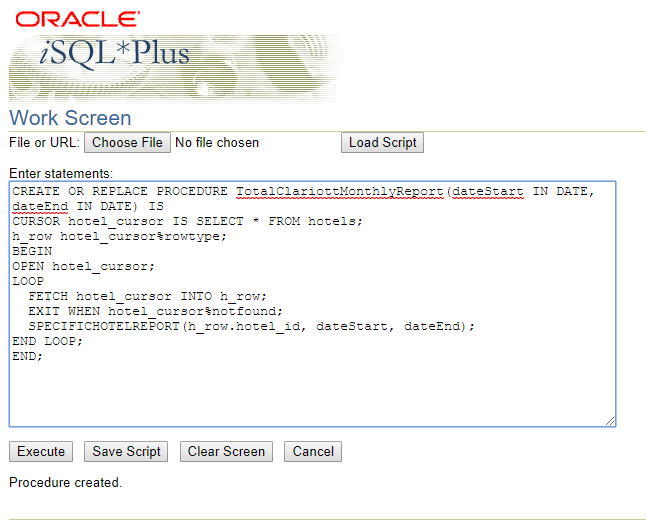
16. [\*\*] SpecificHotelReport:





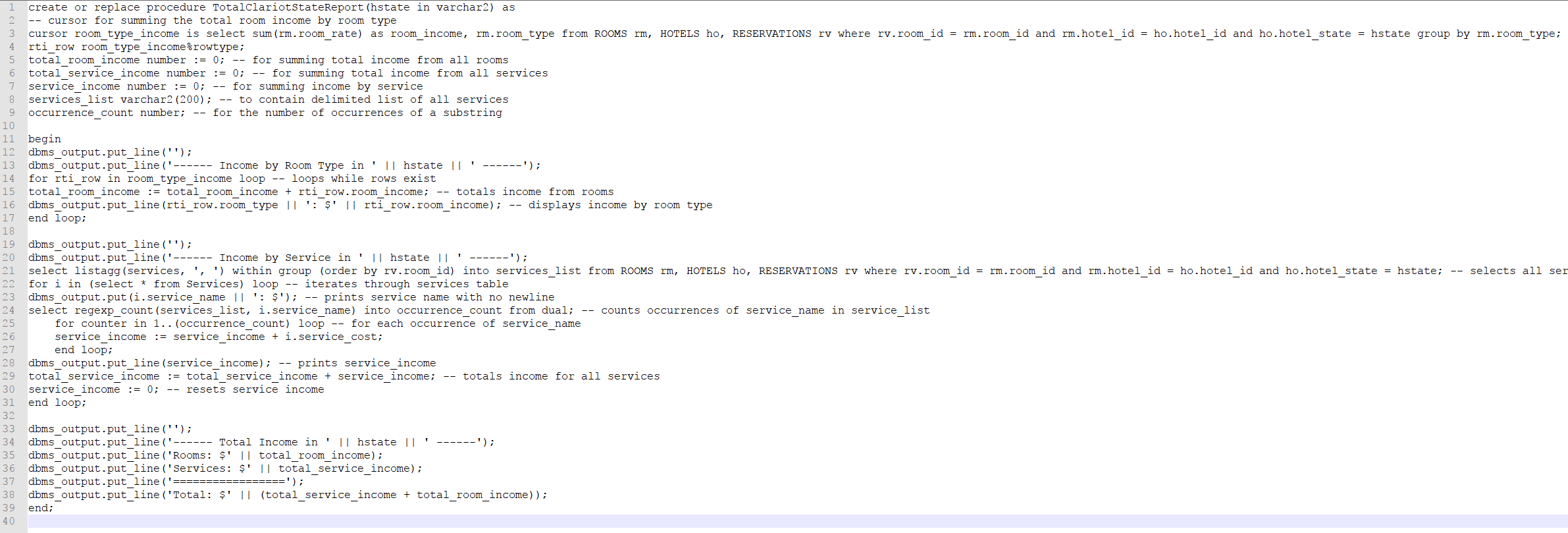


**17. TotalClariottMonthlyReport (\*\*) Josiah**



18. [\*\*] TotalClariottStateReport: Input is state. Print total income from all sources of all hotels by room type and service type in the given state. Include discounts.

Added by Grayson King





Scenario for testing the various operations and reports. Create a text file that makes the

appropriate calls to the PL/SQL stored procedures and functions that are created for the Hotel

Management System. This text file will be used to demonstrate how your software operates.

The list of actions below need to be translated into calls to your existing PL/SQL stored

procedures and functions. Each line identifies the minimum parameters to be entered in the call,

if your code requires additional parameters, feel free to supply.

1. Add a hotel: Added by Jay

a. Add a new hotel called H1 in New York, NY  
 exec add\_hotel('H1', '500 5th ave', 'New York', 'NY', '9174302828');

b. Add a new hotel called H2 in Baltimore, MD  
 exec add\_hotel('H2', '200 E Pratt St', 'Baltimore', 'MD', '4102271422');

c. Add a new hotel called H3 in San Francisco, CA  
 exec add\_hotel('H3', '355 Powell St', 'San Francisco', 'CA', '4154271400');

d. Add a new hotel called H4 in Chicago, IL  
 exec add\_hotel('H4', '160 E Pearson St', 'Chicago', 'IL', '3127871440');

e. Add a new hotel called HH1 in New York, NY  
 exec add\_hotel('HH1', '1050 Park Ave', 'New York', 'NY', '3472123250');

f. Add a new hotel called HH2 in Baltimore, MD

exec add\_hotel('HH2', '301 Light st', 'Baltimore', 'MD', '4103042200');

2. Find a hotel: Added by Jay

a. Find the hotel you entered in CA

exec PRINT\_HOTEL\_INSTATE('CA');

b. Find the first hotel you entered in NY

exec PRINT\_HOTEL\_INSTATE('NY');

3. Sell the hotels in CA and IL: Added by Jay  
exec SELL\_HOTEL\_BYSTATE('CA');

exec SELL\_HOTEL\_BYSTATE('IL');

4. Reservations: Added by Grayson

a. Make a reservation at hotel H2 by John Smith from Aug 1 – Aug 10 for a suite

Execute make\_reservation(2, 'John', 'Smith', to\_date('08-01-2017','mm-dd-yyyy'), to\_date('08-10-2017','mm-dd-yyyy'), 'suite', sysdate);

b. Make a reservation at hotel H3 (already sold)

Execute make\_reservation(3, 'John', 'Smith', to\_date('08-01-2017','mm-dd-yyyy'), to\_date('08-10-2017','mm-dd-yyyy'), null, sysdate);

c. Make a reservation at hotel H2 by John Smith from Feb 1 – Feb 10 for a single

Room

Execute make\_reservation(2, 'John', 'Smith', to\_date('02-01-2017','mm-dd-yyyy'), to\_date('02-10-2017','mm-dd-yyyy'), 'single', sysdate);

d. Make a reservation at hotel HH2 by Nina Chase from Aug 1 – Aug 10 for a

double room

Execute make\_reservation(6, 'Nina', 'Chase', to\_date('08-01-2017','mm-dd-yyyy'), to\_date('08-10-2017','mm-dd-yyyy'), 'double', sysdate);

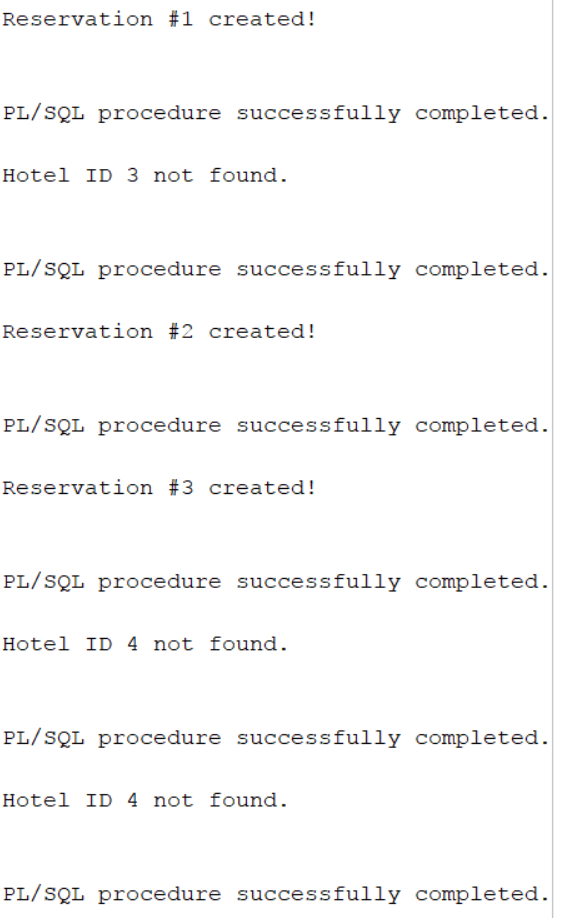
e. Make a reservation at hotel H4 by Pierce Borg from Feb 1 – Feb 10 for a suite

Execute make\_reservation(4, 'Pierce', 'Borg', to\_date('02-01-2017','mm-dd-yyyy'), to\_date('02-10-2017','mm-dd-yyyy'), 'suite', sysdate);

f. Make a reservation at hotel H4 by Pierce Borg from Mar 1 – Mar 10 for a suite

Execute make\_reservation(4, 'Pierce', 'Borg', to\_date('03-01-2017','mm-dd-yyyy'), to\_date('03-10-2017','mm-dd-yyyy'), 'suite', sysdate);





5. Find reservations described above in items 4a, 4c, 4e Josiah

1. dbms\_output.put\_line(to\_char(FINDRESERVATIONS('Smith',to\_date('08-01-2017','mm-dd-yyyy'))));



1. dbms\_output.put\_line(to\_char(FINDRESERVATIONS('Smith',to\_date('02-01-2017','mm-dd-yyyy'))));



1. dbms\_output.put\_line(to\_char(FINDRESERVATIONS('Borg',to\_date('03-01-2017','mm-dd-yyyy'))));

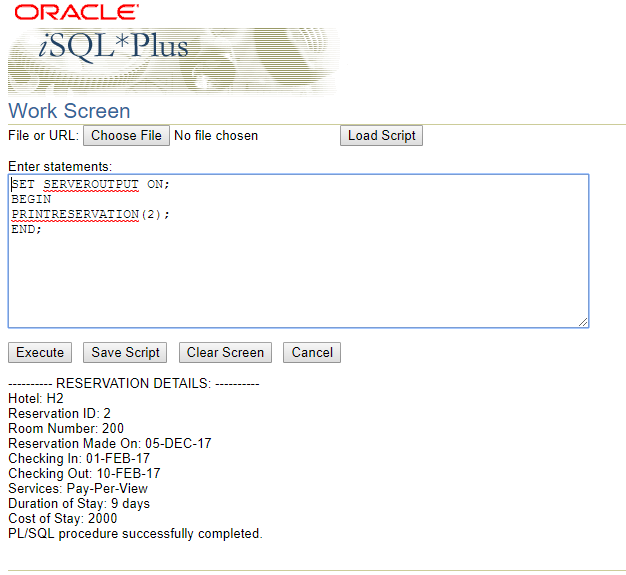


6. Show reservation details for reservations 4a, 4c, and 4e Josiah

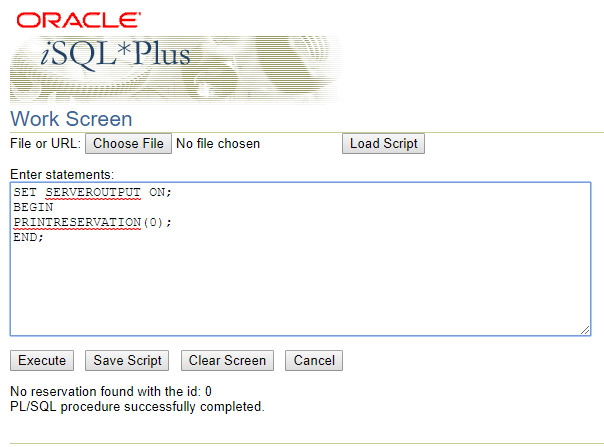
1. PRINTRESERVATION(1);



1. PRINTRESERVATION(2);



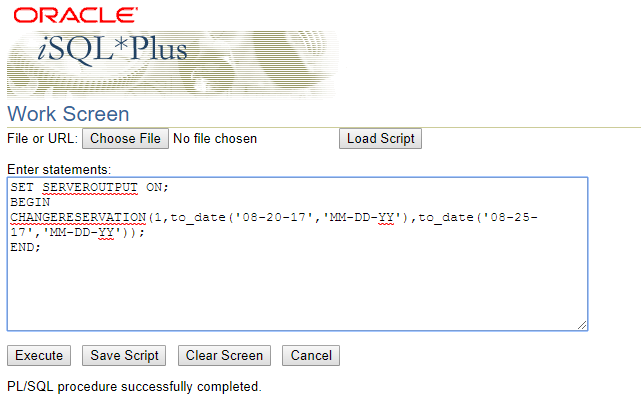
1. PRINTRESERVATION(0);



7. Change the following reservation dates: Josiah

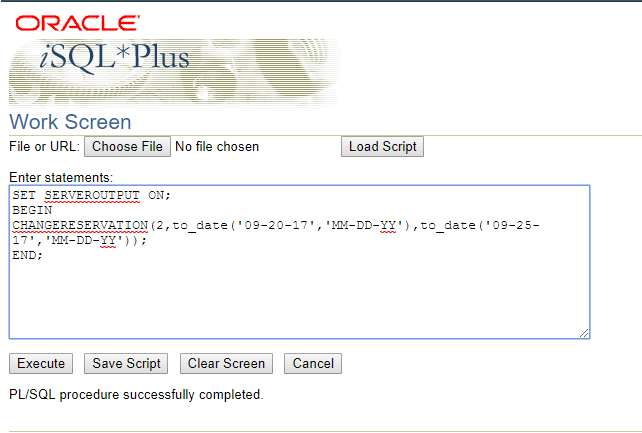
a. For reservation 4a make the new dates from Aug 20-Aug 25

CHANGERESERVATION(1,to\_date('08-20-17','MM-DD-YY'),to\_date('08-25-17','MM-DD-YY'));



b. For reservation 4c make the new dates from Sept 20- Sept 25

CHANGERESERVATION(2,to\_date('09-20-17','MM-DD-YY'),to\_date('09-25-17','MM-DD-YY'));

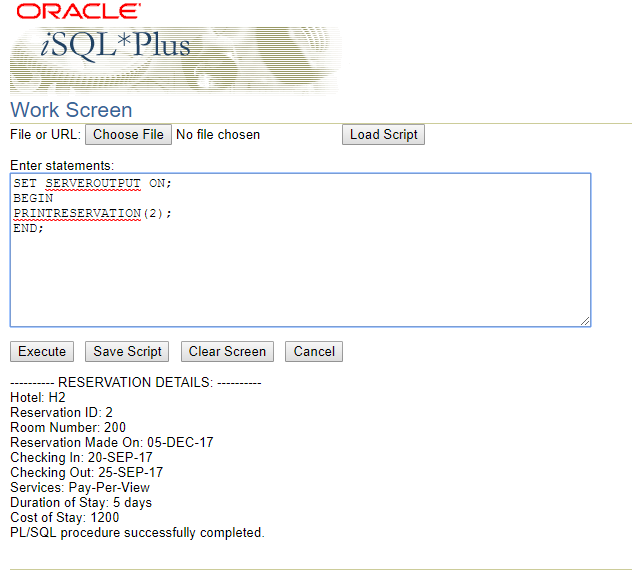


8. Show reservation details for reservations 4a, 4c, and 4e Josiah

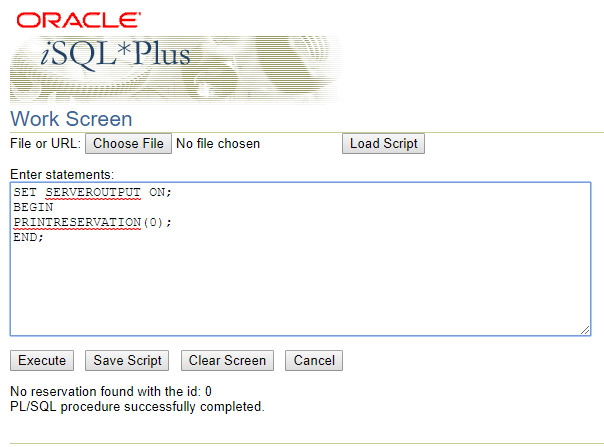
1. PRINTRESERVATION(1);



1. PRINTRESERVATION(2);

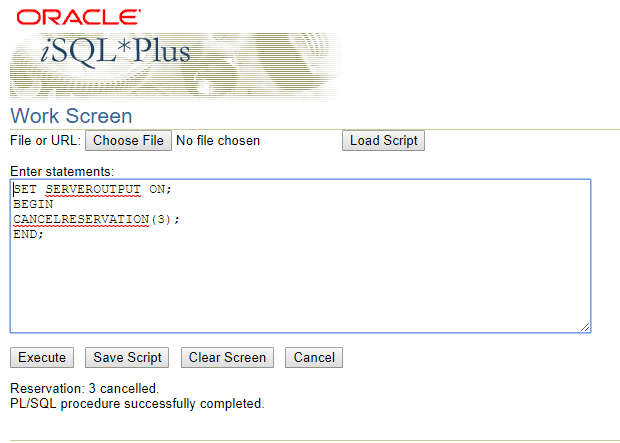


1. PRINTRESERVATION(0);



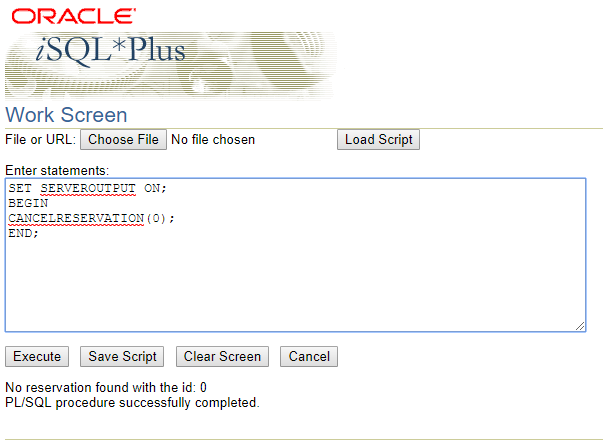
9. Cancel reservation 4d Josiah

CANCELRESERVATION(3);



10. Cancel reservation 4f Josiah

CANCELRESERVATION(0);



11. Add services: Added by Grayson

a. For reservation 4a add restaurant services for each day

Execute add\_service(1, 'Daily Restaurant');

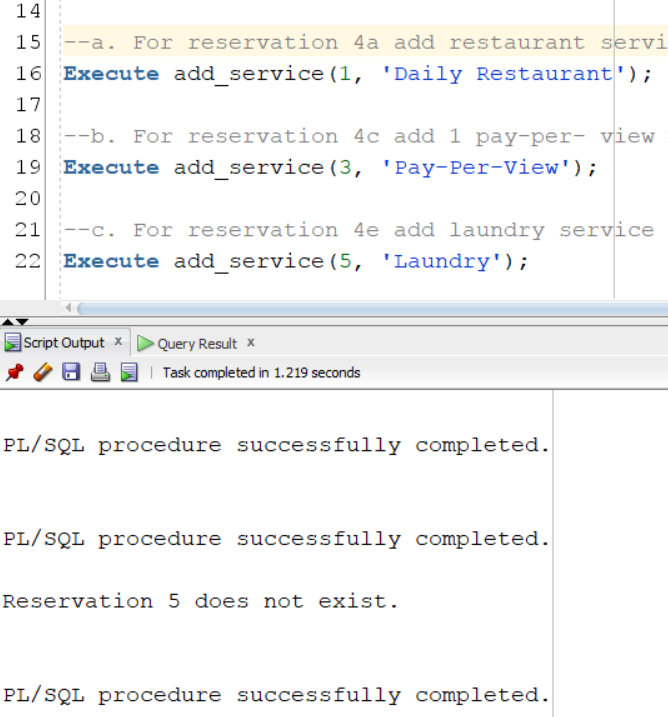
b. For reservation 4c add 1 pay-per- view movie for the first day

Execute add\_service(3, 'Pay-Per-View');

c. For reservation 4e add laundry service for one day (of your choice)

Execute add\_service(5, 'Laundry');

Commit;

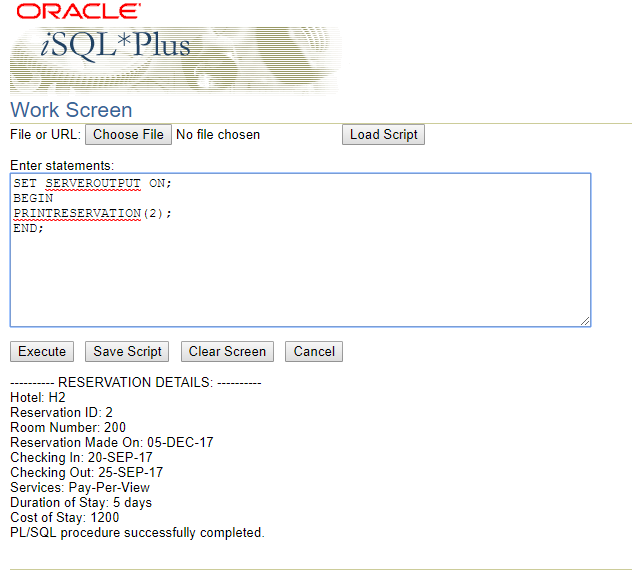


12. Show reservation details for reservations 4a, 4c, and 4e Josiah

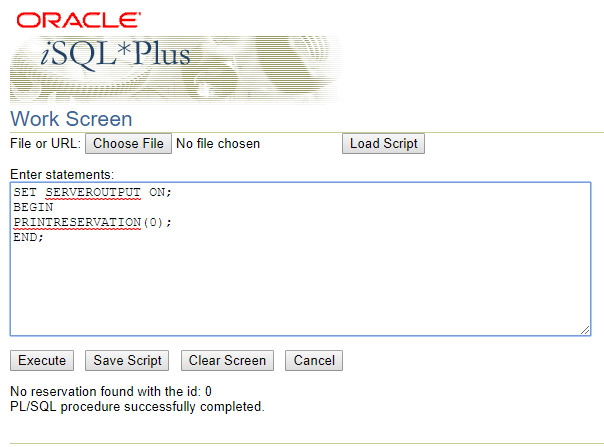
1. PRINTRESERVATION(1);



1. PRINTRESERVATION(2);



1. PRINTRESERVATION(0);



13. Change reservation room type: Added by Charles

a. For reservation 4a from Suite to double

Exec Change\_Room\_Type(1,'double');

b. For reservation 4c from single to double

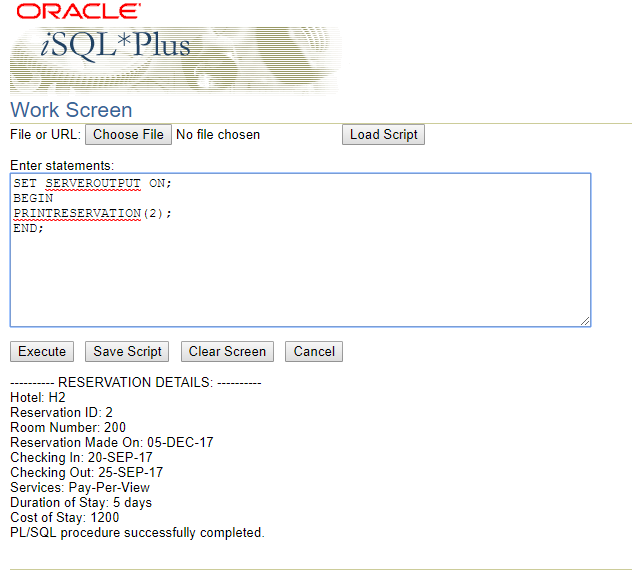
Exec Change\_Room\_Type(2,'double');

14. Show reservation details for reservations 4a, 4c Josiah

1. PRINTRESERVATION(1);



1. PRINTRESERVATION(2);



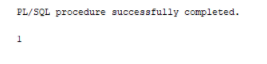
15. Show available suite rooms in H2 from Jul 1- Jul 30

Exec room\_available(2, 'suite', to\_date('2017-07-01','yyyy-mm-dd'), to\_date('2017-07-30','yyyy-mm-dd'));



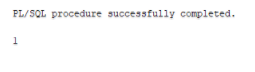
16. Show available double rooms in H2 from Aug 1- Aug 30

Exec room\_available(2, 'double', to\_date('2017-08-01','yyyy-mm-dd'), to\_date('2017-08-30','yyyy-mm-dd'));



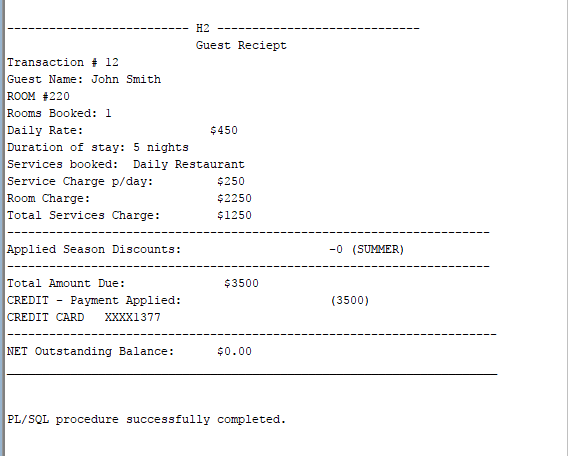
17. Show available rooms in H2 from Aug 1-Aug 30

Exec room\_available(2, 'All', to\_date('2017-08-01','yyyy-mm-dd'), to\_date('2017-08-30','yyyy-mm-dd'));



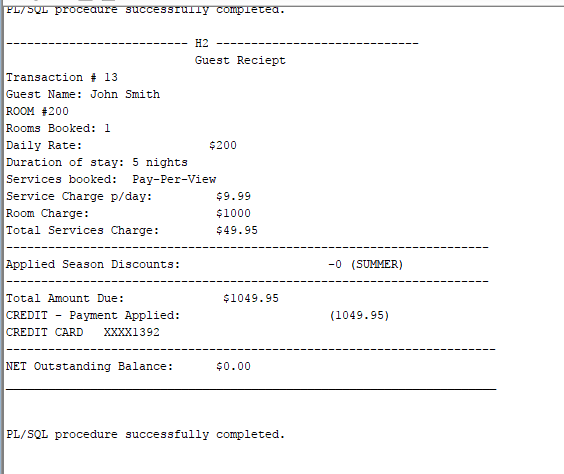
18. Room checkout receipt for John Smith for reservation in 4a Added by Jay

exec room\_checkout\_reciept(1);



19. Room checkout receipt for John Smith for reservation in 4c Added by Jay

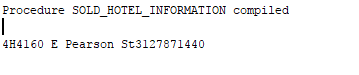
exec room\_checkout\_reciept(2);



20. Room checkout receipt for Pierce Borg for reservation in 4e Added by Jay  
exec room\_checkout\_reciept(4); --canceled

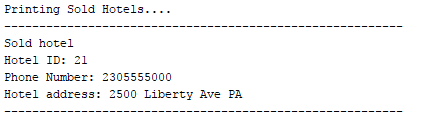
21. Show sold hotels Added by Luvneet

Exec Sold\_Hotel\_Information;



22. Show all cancellations Added by Charles

exec Show\_Cancelations;



23. Specific hotel report: Run it for each hotel you have created Added by Grayson

Execute specifichotelreport(1,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

Execute specifichotelreport(2,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

Execute specifichotelreport(3,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

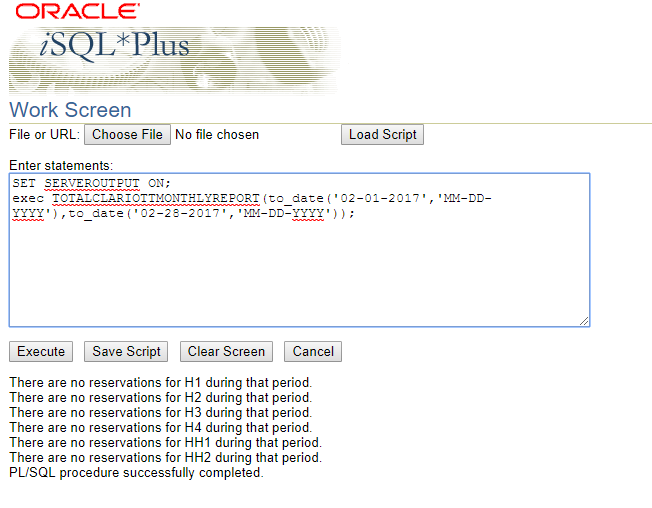
Execute specifichotelreport(4,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

Execute specifichotelreport(5,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

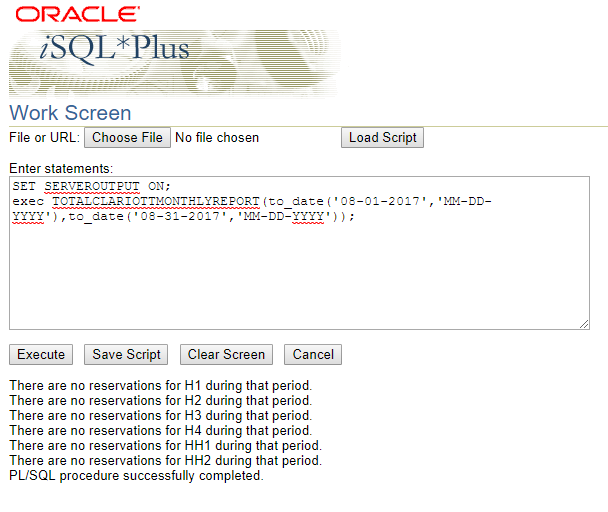
Execute specifichotelreport(6,to\_date('01-01-2000','mm-dd-yyyy'), to\_date('01-01-2020','mm-dd-yyyy'));

24. TotalClariot Monthly: Run this report for the months of Feb, Aug, Jan, September Josiah

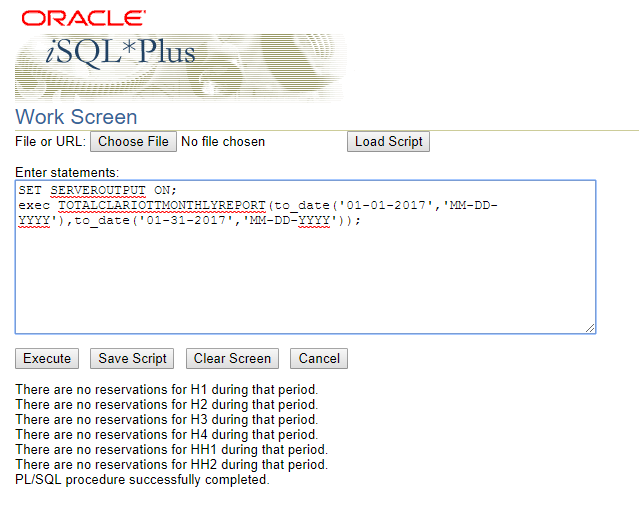
1. Exec TOTALCLARIOTTMONTHLYREPORT(to\_date('02-01-2017','MM-DD-YYYY'),to\_date('02-28-2017','MM-DD-YYYY'));



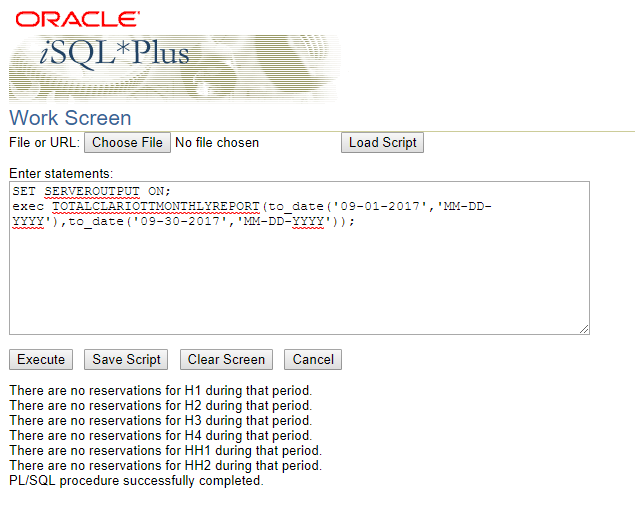
1. TOTALCLARIOTTMONTHLYREPORT(to\_date('08-01-2017','MM-DD-YYYY'),to\_date('08-31-2017','MM-DD-YYYY'));



1. TOTALCLARIOTTMONTHLYREPORT(to\_date('01-01-2017','MM-DD-YYYY'),to\_date('01-31-2017','MM-DD-YYYY'));



1. TOTALCLARIOTTMONTHLYREPORT(to\_date('09-01-2017','MM-DD-YYYY'),to\_date('09-30-2017','MM-DD-YYYY'));



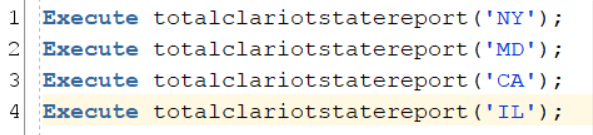
25. Total Clariot State report for NY, MD, CA, IL Added by Grayson

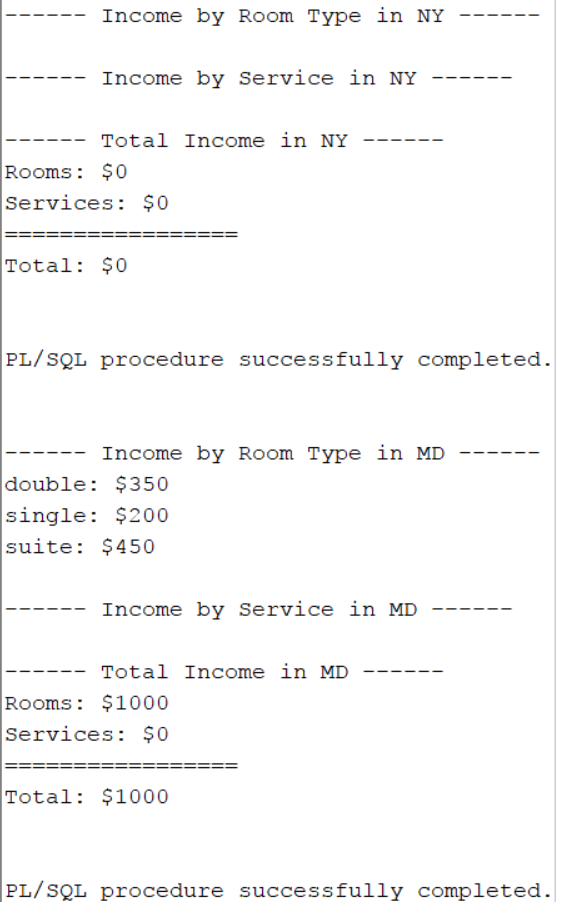
Execute totalclariotstatereport('NY');

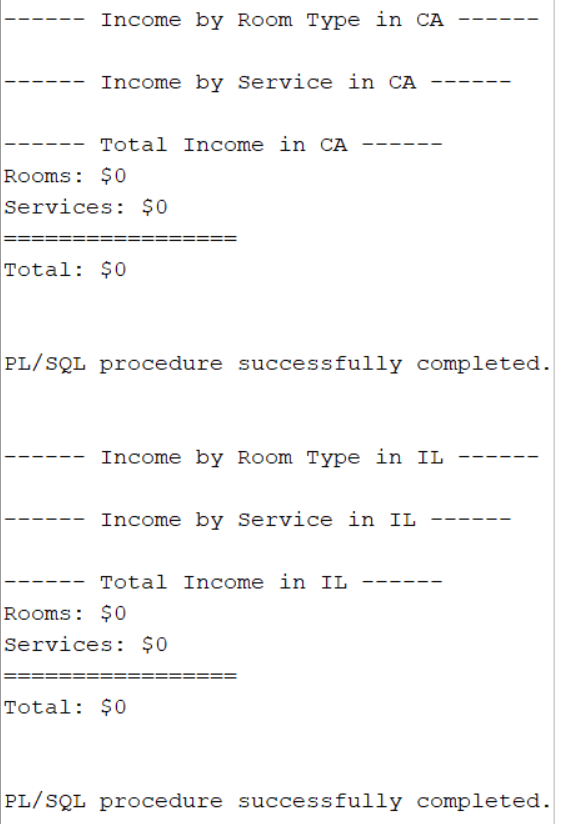
Execute totalclariotstatereport('MD');

Execute totalclariotstatereport('CA');

Execute totalclariotstatereport('IL');



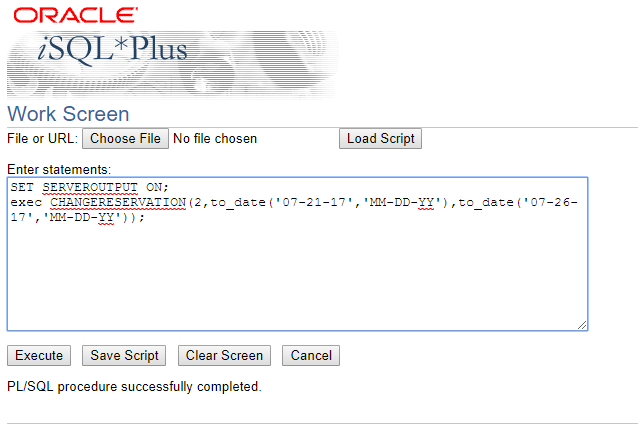




26. Change one reservation date and make it much earlier so that discounts will not apply for

that reservation Josiah

CHANGERESERVATION(2,to\_date('07-21-17','MM-DD-YY'),to\_date('07-26-17','MM-DD-YY'));



27. Total Clariot State report for NY, MD, CA, IL Added by Grayson

Execute totalclariotstatereport('NY');

Execute totalclariotstatereport('MD');

Execute totalclariotstatereport('CA');

Execute totalclariotstatereport('IL');

