create or replace procedure make\_reservation(hotel\_identifier in number, guest\_fname in varchar2,

guest\_lname varchar2, start\_date in date, end\_date in date, r\_type in varchar2, date\_of\_reserv in date) as

invalid\_hotel exception; -- When the hotel doesn't exist

invalid\_guest exception; -- When the guest doesn't exist, or there are duplicate guests

invalid\_dates exception; -- When the start date is not before the end date

no\_rooms exception; -- When no room exists

room\_flag integer; -- flags if the room is booked

new\_type varchar2(10); -- used for holding the room type

hid\_exists integer := 0; -- Used to store matching hotel id count

res\_cost number; -- Stores the calculated cost of the reservation

discount number; -- Stores the discount based on season

gid number; -- Used to store matching guest id number

rid number; -- Used to store matching room id number

staySeason RESERVATIONS.SEASON%TYPE; -- Used to store the season

begin

select count(\*) into hid\_exists from HOTELS where hotel\_id = hotel\_identifier and hotel\_sold = 'NO';

select guest\_id into gid from GUESTS where guest\_first\_name = guest\_fname and guest\_last\_name = guest\_lname;

-- the following loop selects a matching available room

if r\_type is null then

new\_type := 'single';

else

new\_type := r\_type;

end if;

for rm in (select room\_id from ROOMS where room\_type = new\_type and hotel\_id = hotel\_identifier) loop -- iterates through ROOMS tabel to find a matching room type

room\_flag := 0; -- sets the flag to 0

rid := rm.room\_id; -- sets that room\_id as the rid

-- checks if the selected room\_id has any conflicting reservation

select count(\*) into room\_flag from RESERVATIONS where rid = RESERVATIONS.room\_id and (start\_date between reservation\_check\_in and reservation\_check\_out) or (end\_date between reservation\_check\_in and reservation\_check\_out);

if room\_flag > 0 then -- clears the rid if the room is flagged as booked for those dates

rid := null;

end if;

-- the rid remains null if no match was found

-- the rid remains set if a match was found

end loop;

if rid is null then -- raises an exception if no room was found previously

raise no\_rooms;

end if;

-- the following if statements calculate the discount

if start\_date between to\_date('10-22','mm-dd') and to\_date('12-20','mm-dd') then

-- if Fall then 25% discount

discount := .25;

staySeason := 'FALL';

elsif start\_date between to\_date('12-21','mm-dd') and to\_date('03-19','mm-dd') then

-- if Winter then 50% discount

discount := .5;

staySeason := 'WINTER';

elsif start\_date between to\_date('03-20','mm-dd') and to\_date('06-21','mm-dd') then

--if Spring then 10% discount

discount := .1;

staySeason := 'SPRING';

else

discount := 0;

staySeason := 'SUMMER';

end if;

-- the following calculates the reservation cost

select room\_rate into res\_cost from ROOMS where room\_id = rid; -- gets the daily room cost

res\_cost := (res\_cost \* ((end\_date - start\_date) + 1)) \* (1 - discount); -- calculates cost of stay, applies discount

if hid\_exists = 1 then --Checks if the hotel exists, otherwise raises an exception

if gid is not null then -- Checks if a matching guest was found

if start\_date <= end\_date then -- Checks if the start and end dates are valid, otherwise raises an exception

-- Inserts the new reservation

insert into RESERVATIONS values(reservationid\_seq.NEXTVAL, rid, gid, date\_of\_reserv, start\_date, end\_date, null, res\_cost, staySeason);

-- Updates the date the room will be ready

update rooms set ready = end\_date where room\_id = rid;

dbms\_output.put\_line('Reservation #' || reservationid\_seq.CURRVAL || ' created!'); -- Prints the reservation ID

else

raise invalid\_dates;

end if;

else

raise invalid\_guest;

end if;

else

raise invalid\_hotel;

end if;

exception

when invalid\_hotel then

dbms\_output.put\_line('Hotel ID ' || hotel\_identifier || ' not found.');

when invalid\_guest then

dbms\_output.put\_line('Guest name "' || guest\_fname || ' ' || guest\_lname || '" not found.');

when invalid\_dates then

dbms\_output.put\_line('Invalid dates.');

when no\_rooms then

dbms\_output.put\_line('No matching rooms found.');

when no\_data\_found then

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

end;