Deliverable 2

Team 6

Caleb Massey

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

/\*======================================================================\*/

/\* Checks if the room is available during the date \*/

create or replace function CheckBookAvailable(temp\_b\_id int, new\_in\_date date, new\_out\_date date)

return varchar2 as free char;

cursor check\_cursor is

select b\_id, check\_in\_date, check\_out\_date from booking

where booking.r\_id = (select r\_id from booking where temp\_b\_id = b\_id)

and b\_id != temp\_b\_id;

check\_row check\_cursor%rowtype;

keep\_count int := 0;

no\_other\_reservations exception;

begin

for check\_row in check\_cursor loop

/\*Checks for conflicting date\*/

keep\_count := keep\_count + 1;

if new\_in\_date between check\_row.check\_in\_date and check\_row.check\_out\_date then

return to\_char('Occupied');

else

/\*Checks for conflicting date\*/

if new\_out\_date between check\_row.check\_in\_date and check\_row.check\_out\_date then

return to\_char('Occupied');

else

/\*Checks for conflicting date\*/

if check\_row.check\_in\_date > new\_in\_date and new\_out\_date > check\_row.check\_in\_date then

return to\_char('Occupied');

else

/\*Runs if there are no conflicting nights\*/

return to\_char('Free');

end if;

end if;

end if;

end loop;

if keep\_count = 0 then

raise no\_other\_reservations;

end if;

exception when no\_other\_reservations then

return to\_char('Free');

end;

/\*---------------------------------------------------------------------------------------------------------------------------\*/

/\* temp\_id is a booking id \*/

create or replace procedure update\_reservation\_date(temp\_id number, new\_in\_date date, new\_out\_date date)

as

temp\_b\_date date;

Date\_Occupied exception;

Begin

if CheckBookAvailable(temp\_id, new\_in\_date, new\_out\_date) = 'Free' then

select b\_date into temp\_b\_date from booking where b\_id = temp\_id;

dbms\_output.put\_line('Booking ID: ' || temp\_id || ' updated Check In Date to ' ||

new\_in\_date || ' and Check Out Date to ' || new\_out\_date || ' Booking Date is ' || temp\_b\_date);

update booking set check\_in\_date = new\_in\_date, check\_out\_date = new\_out\_date,

room\_cost = price\_from\_date(temp\_b\_date, new\_out\_date)

where b\_id = temp\_id;

else

raise Date\_Occupied;

end if;

exception when Date\_Occupied then

dbms\_output.put\_line('New date is currently occupied');

end;

/\*---------------------------------------------------------------------------------------------------------------------------\*/

set serveroutput on;

execute update\_reservation\_date(12, to\_date('15-APR-18','dd-mon-yy'), to\_date('27-APR-18','dd-mon-yy'));

/\*---------------------------------------------------------------------------------------------------------------------------\*/

select \* from booking;

/\*======================================================================\*/

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

/\*======================================================================\*/

set serveroutput on;

/\*temp\_id is a Hotel ID\*/

create or replace procedure show\_reservations(temp\_id number)

as

cursor reserve\_cursor is

select booking.b\_id, booking.r\_id, booking.b\_date,

booking.check\_in\_date, booking.check\_out\_date,

booking.num\_guests, booking.has\_reservation

from booking, room, hotel

where hotel.h\_id = room.h\_id

and room.r\_id = booking.r\_id

and hotel.h\_id = temp\_id;

reserve\_row reserve\_cursor%rowtype;

keep\_count int := 0;

no\_reservation exception;

begin

for reserve\_row in reserve\_cursor loop

if reserve\_row.b\_date is NULL then

dbms\_output.put\_line('Reservation ID: ' || reserve\_row.b\_id || ', Room ID: ' ||

reserve\_row.r\_id || ', Booking Date: NULL, Check In Date: ' ||

reserve\_row.check\_in\_date || ', Check Out Date: ' || reserve\_row.check\_out\_date || ', Number of Guests: ' ||

reserve\_row.num\_guests || ', Has Reservations: ' || reserve\_row.has\_reservation);

else

dbms\_output.put\_line('Reservation ID: ' || reserve\_row.b\_id || ', Room ID: ' ||

reserve\_row.r\_id || ', Booking Date: ' || reserve\_row.b\_date || ', Check In Date: ' ||

reserve\_row.check\_in\_date || ', Check Out Date: ' || reserve\_row.check\_out\_date || ', Number of Guests: ' ||

reserve\_row.num\_guests || ', Has Reservations: ' || reserve\_row.has\_reservation);

end if;

keep\_count := keep\_count + 1;

end loop;

if keep\_count = 0 then

raise no\_reservation;

end if;

exception when no\_reservation then

dbms\_output.put\_line('No reservations for hotel id ' || temp\_id);

end;

/\*======================================================================\*/

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

/\*======================================================================\*/

set serveroutput on;

create or replace procedure customer\_reservations(search\_name varchar2)

as

cursor customer\_cursor is

select booking.b\_id, booking.r\_id, booking.b\_date, booking.check\_in\_date, booking.check\_out\_date, booking.num\_guests, booking.has\_reservation

from customer, booking

where customer.c\_id = booking.c\_id

and customer.c\_name = search\_name;

customer\_row customer\_cursor%rowtype;

keep\_count int := 0;

no\_reservation exception;

begin

for customer\_row in customer\_cursor loop

dbms\_output.put\_line('Booking ID: ' || customer\_row.b\_id || ' Room ID: ' || customer\_row.r\_id || ' Check In Date: ' || customer\_row.check\_in\_date

|| ' Check Out Date: ' || customer\_row.check\_out\_date || ' Number Of Guests: ' || customer\_row.num\_guests || ' Has Reservation: ' || customer\_row.has\_reservation);

keep\_count := keep\_count + 1;

end loop;

if keep\_count = 0 then

raise no\_reservation;

end if;

exception when no\_reservation then

dbms\_output.put\_line('Customer ' || search\_name || ' has no reservations');

end;

/\*---------------------------------------------------------------------------------------------------------------------------\*/

execute customer\_reservations('Tommy Lee');

select \* from customer;

select \* from booking;

/\*======================================================================\*/