Deliverable 3

Team 6

Caleb Massey

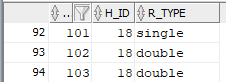
Change a reservation RoomType: Input the reservation ID and change reservation room type if there is availability for that room type during the reservation’s date interval

Before running Pictures:

Booking



Room



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

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

create or replace function CheckAvailable(temp\_h\_id int, temp\_r\_id int, in\_date date, out\_date date, new\_r\_type varchar2)

return varchar2 as free char;

cursor check\_cursor is

select booking.r\_id, check\_in\_date, check\_out\_date from room, hotel, booking

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

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

and hotel.h\_id = temp\_h\_id

and r\_type = new\_r\_type;

check\_row check\_cursor%rowtype;

begin

for check\_row in check\_cursor loop

/\*Checks for conflicting date\*/

if 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 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 > in\_date and 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;

end;

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

set serveroutput on;

create or replace procedure UpdateRoomType(temp\_b\_id number, new\_r\_type varchar2)

as

temp\_r\_id int; /\*Temporary track r\_id\*/

keep\_count int := 0; /\*Keeps count of if booking has been updated yet\*/

occupied exception; /\*exception for if no rooms available of type new\_r\_type\*/

row\_count int := 0; /\*Counts the number of rows in date\_cursor, to know if empty\*/

/\* Finds the rooms that are in that hotel, with the r\_type being changed to\*/

cursor type\_cursor is

select hotel.h\_id, room.r\_id, r\_type from room, hotel

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

and hotel.h\_id = (select hotel.h\_id from hotel, room, booking

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

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

and booking.b\_id = temp\_b\_id)

and r\_type = new\_r\_type;

type\_row type\_cursor%rowtype;

/\* Finds the rooms that are in that hotel, with the r\_type being changed to, and is booked at any time\*/

cursor date\_cursor is

select hotel.h\_id, room.r\_id, check\_in\_date, check\_out\_date from room, hotel, booking

where booking.r\_id = room.r\_id

and hotel.h\_id = room.h\_id

and hotel.h\_id = (select hotel.h\_id from hotel, room, booking

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

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

and booking.b\_id = temp\_b\_id)

and r\_type = new\_r\_type;

date\_row date\_cursor%rowtype;

begin

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

for date\_row in date\_cursor loop

row\_count := row\_count + 1;

end loop;

/\*Loops through all rooms that are in the hotel of the desired r\_type\*/

for type\_row in type\_cursor loop

dbms\_output.put\_line('Type Row ' || type\_row.r\_id);

/\*Loops through all rooms that are in booking and hotel of the same r\_type\*/

for date\_row in date\_cursor loop

dbms\_output.put\_line('Date Row ' || date\_row.r\_id );

/\*Checks if the room is in both cursors\*/

if type\_row.r\_id = date\_row.r\_id then

dbms\_output.put\_line('Pre-Check ' ||type\_row.h\_id|| ' ' || date\_row.r\_id|| ' ' ||date\_row.check\_in\_date|| ' ' ||date\_row.check\_out\_date|| ' ' ||type\_row.r\_type);

/\*Checks if the room in booking is available during the date the original booking is set for \*/

if CheckAvailable(date\_row.h\_id, date\_row.r\_id, date\_row.check\_in\_date, date\_row.check\_out\_date, type\_row.r\_type) = 'Free' then

dbms\_output.put\_line('Passed ' ||type\_row.r\_id || ' and ' || date\_row.r\_id ||' is Free');

/\*Checks if the new room has been updated yet\*/

if keep\_count = 0 then

keep\_count := keep\_count + 1;

/\*update booking set booking.r\_id = type\_row.r\_id where booking.b\_id = temp\_b\_id;\*/

dbms\_output.put\_line('1 Update Complete to room ' || type\_row.r\_id || ' for ' || temp\_r\_id);

end if;

/\*Runs if the room was not available\*/

else

if CheckAvailable(type\_row.h\_id, type\_row.r\_id, date\_row.check\_in\_date, date\_row.check\_out\_date, type\_row.r\_type) = 'Occupied' then

dbms\_output.put\_line (type\_row.r\_id || ' and ' || date\_row.r\_id ||' is Occupied');

end if;

end if;

/\*If this runs it means that room has not been booked yet and is therefor free \*/

else

dbms\_output.put\_line('Failed ' || type\_row.h\_id|| ' ' || date\_row.r\_id|| ' ' ||date\_row.check\_in\_date|| ' ' ||date\_row.check\_out\_date|| ' ' ||type\_row.r\_type);

/\*Checks if the new room has been updated yet\*/

if keep\_count = 0 then

keep\_count := keep\_count + 1;

/\*update booking set booking.r\_id = type\_row.r\_id where booking.b\_id = temp\_b\_id;\*/

dbms\_output.put\_line('2 Update Complete to room ' || type\_row.r\_id || ' for ' || temp\_r\_id);

end if;

end if;

end loop;

if row\_count = 0 then

/\*Runs if there is no rooms in booking at all\*/

if keep\_count = 0 then

keep\_count := keep\_count + 1;

/\*update booking set booking.r\_id = type\_row.r\_id where booking.b\_id = temp\_b\_id;\*/

dbms\_output.put\_line('3 Update Complete to room ' || type\_row.r\_id || ' for ' || temp\_r\_id);

end if;

end if;

end loop;

/\*If nothing can be updated, runs exception, case is no rooms of this type in hotel or no room replacement available\*/

if keep\_count = 0 then

Raise occupied;

end if;

exception when occupied then

dbms\_output.put\_line ('No Rooms available of type ' || new\_r\_type);

end;

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

set serveroutput on;

execute UpdateRoomType(11, 'double');

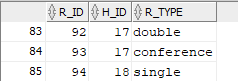
/\*======================================================================\*/

After Running Code:

Booking



Room



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.

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

create or replace procedure calculate\_income

as

total\_income int; /\*Variable that holds the sum of service prices and room costs for the month\*/

sum\_price int; /\*The sum of all the service prices for that month\*/

temp\_r\_cost int; /\*Holds the sum of all the room costs for that month, discount included\*/

month\_date date; /\*Keeps track of the dates as mon-yyyy instead of dd-mon-yyyy\*/

/\*Splits into all months a room is being rented\*/

cursor income\_cursor IS

select distinct to\_char(booking.check\_out\_date, 'MON-YYYY') as month\_date

from hotel, room, booking

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

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

order by month\_date;

income\_row income\_cursor%rowtype;

begin

for income\_row in income\_cursor loop

/\*Finds the sum of service prices for each month, can be NULL\*/

select sum(distinct s\_price) into sum\_price from hotel, room, booking, service

where hotel.h\_id = room.h\_id and room.r\_id = booking.r\_id and booking.b\_id = service.b\_id and

to\_char(booking.check\_out\_date, 'MON-YYYY') like '%' || income\_row.month\_date || '%';

/\*Finds the sum of room costs for each month\*/

select sum(room\_cost) into temp\_r\_cost from hotel, room, booking

where hotel.h\_id = room.h\_id and room.r\_id = booking.r\_id and

to\_char(booking.check\_out\_date, 'MON-YYYY') like '%' || income\_row.month\_date || '%';

/\*Figures out if there were any services provided and adds up the total income\*/

if sum\_price is null then

total\_income := temp\_r\_cost;

else

total\_income := temp\_r\_cost + sum\_price;

end if;

/\*Prints out each months total income\*/

dbms\_output.put\_line(' Income Total for ' || income\_row.month\_date || ' is ' || total\_income);

end loop;

end;

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

set serveroutput on;

execute calculate\_income;

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

