

# AUST CSE 4126 Spring 2018 - Project Report

## Multi-Location Hotel Management System

### With Remote Branches Facilities

**Sarwar Saif**

ID – 15.01.04.091

Email: [saifmu6@gmail.com](mailto:saifmu6@gmail.com)

Year: 4<sup>th</sup> | Semester: 1<sup>st</sup> | Section: B2

Date: 13-October-2018

**Group Members:**

Masud Rana | ID - 15.01.04.090

Samiur Rahman | ID - 15.01.04.092

---

## 1. Introduction

Our project is about a 'Multi-location Hotel' which has two branches along with a head office. Our aim is to make a management system for the hotels located in different regions. As we all know hotel is necessary for travelers but in Bangladesh hardly few hotels has the facility of reservation on remote branches without any help from the third party. So, there is huge opportunity in this sector to improve by creating multi-location hotel management systems for the head office to control their branches more efficiently and effectively and deliver more values to the customer. That's why we decided to work on this project. Before doing our project we make a study on the management system of "Pan Pacific Sonargaon." & "InterContinental".

## 2. Language and Software Tools:

### 2.1. Language:

**PL/SQL**(Procedural Language/Structured Query Language) is **Oracle** Corporation's procedural extension for **SQL** and the **Oracle** relational database.

### 2.2. Softwares:

Oracle 11g Express Edition

Oracle SQL developer for real time emulation of remote sites using single pc.

#### 2.2.1 Why Oracle SQL Developer?

It was very hard for me to write code for different sites using multiple laptops as all of us had pc. So, it took me 3 days to find out something that can help me with real time emulation which will give me the same environment of multiple laptop setup or virtual box and pc environment. But all of this techniques has no proper ways of error detection which becomes very hard and annoying to evaluate the codes properly. "Oracle SQL Developer" gives me the opportunity to work in the similar environment by creating different "User Connection" like remote database and then creating of different "Site Links" where 'Head Office' uses "LocalHost" and other branches uses my IPV4 address 192.168.1.102 and helps me to run my code pretty easily. It's an amazing tool I must say.

## 3. Sites and Tables

There are three different sites in our project. "Dhaka", "Chittagong" & "Noakhali". "Dhaka" site is our main branch and "Chittagong" & "Noakhali" branches are our sub branches. Relational schemas for each site are given below:

### 3.1. Dhaka Site

- branch (branchId, district)
- staff (staffId, jobId, firstName, lastName, address, bookingId)
- guest (guestId, firstName, lastName, address, occupation, telephone)
- booking (bookingId, guestId, staffId, roomNo, numberOfGuest, bookingDate, checkedin, checkout, telephone, branchId)
- paymentMethod (methodId, methodName)
- payment (paymentId, methodId, dateofpay, bookingId, totalPaid)
- room (branchId, roomNo, roomtypeId, available)
- roomType (roomtypeId, roomType, price, details)
- job (jobId, jobType)
- salary (jobId, amount)

### 3.2. Chittagong Site

- staff (staffId, jobId, firstName, lastName, address, bookingId)
- guest (guestId, firstName, lastName, address, occupation, telephone)
- booking (bookingId, guestId, staffId, roomNo, numberOfGuest, bookingDate, checkedin, checkout, telephone)
- paymentMethod (methodId, methodName)
- payment (paymentId, methodId, dateofpay, bookingId, totalPaid)
- room (roomNo, roomtypeId, available)
- roomType (roomtypeId, roomType, price, details)
- job (jobId, jobType)
- salary (jobId, amount)

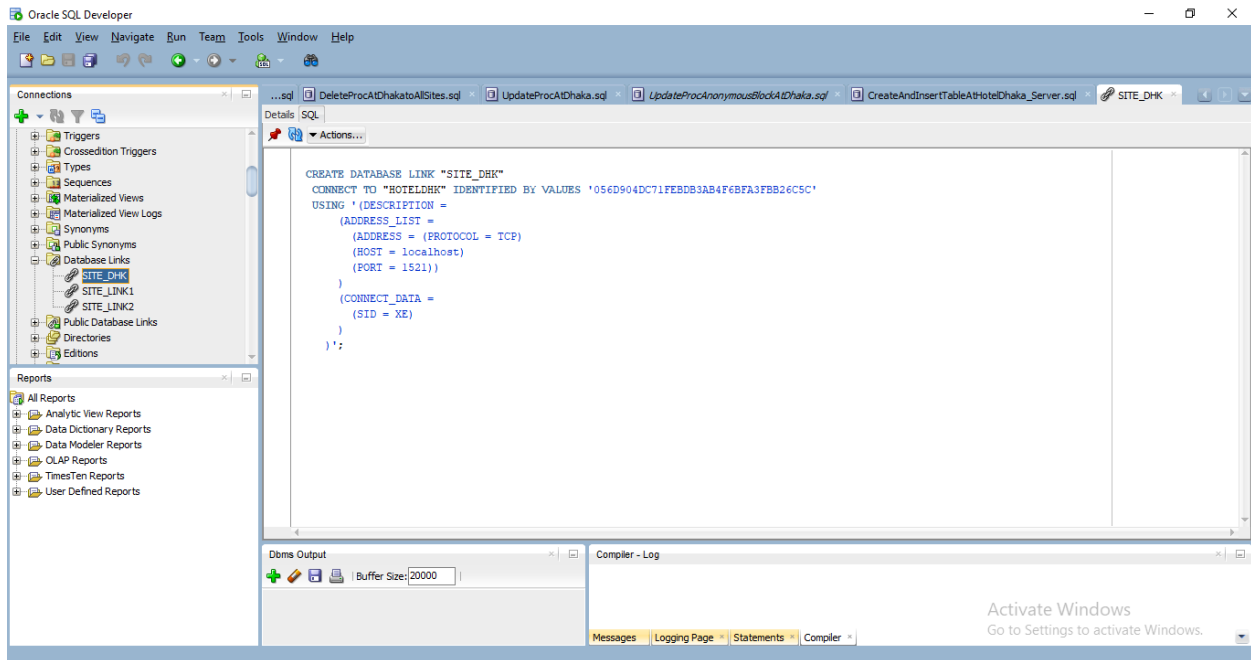
### 3.3. Noakhali Site

- staff (staffId, jobId, firstName, lastName, address, bookingId)
- guest (guestId, firstName, lastName, address, occupation, telephone)
- booking (bookingId, guestId, staffId, roomNo, numberOfGuest, bookingDate, checkedin, checkout, telephone)
- paymentMethod (methodId, methodName)
- payment (paymentId, methodId, dateofpay, bookingId, totalPaid)
- room (roomNo, roomtypeId, available)
- roomType (roomtypeId, roomType, price, details)
- job (jobId, jobType)
- salary (jobId, amount)

## 4. Database links

Here we use three links for our project. They are given below:-

- “site\_dhk” for connecting with “Dhaka”
- “site\_ctg” for connecting with “Chittagong”
- “site\_nhk” for connecting with “Noakhali”



**Fig 4.1:** Site Link Code Of Dhaka As “site\_dhk”

```
CREATE DATABASE LINK "SITE_CTG"
CONNECT TO "HOTELCTG" IDENTIFIED BY VALUES '053172CAB0B08BDD16346ECC11C8D30AB0'
USING '(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = 192.168.1.102)
(PORT = 1521))
)
(CONNECT_DATA =
(SID = XE)
)
)';
```

**Fig 4.2:** Site Link Code Of Chittagong As “site\_ctg”

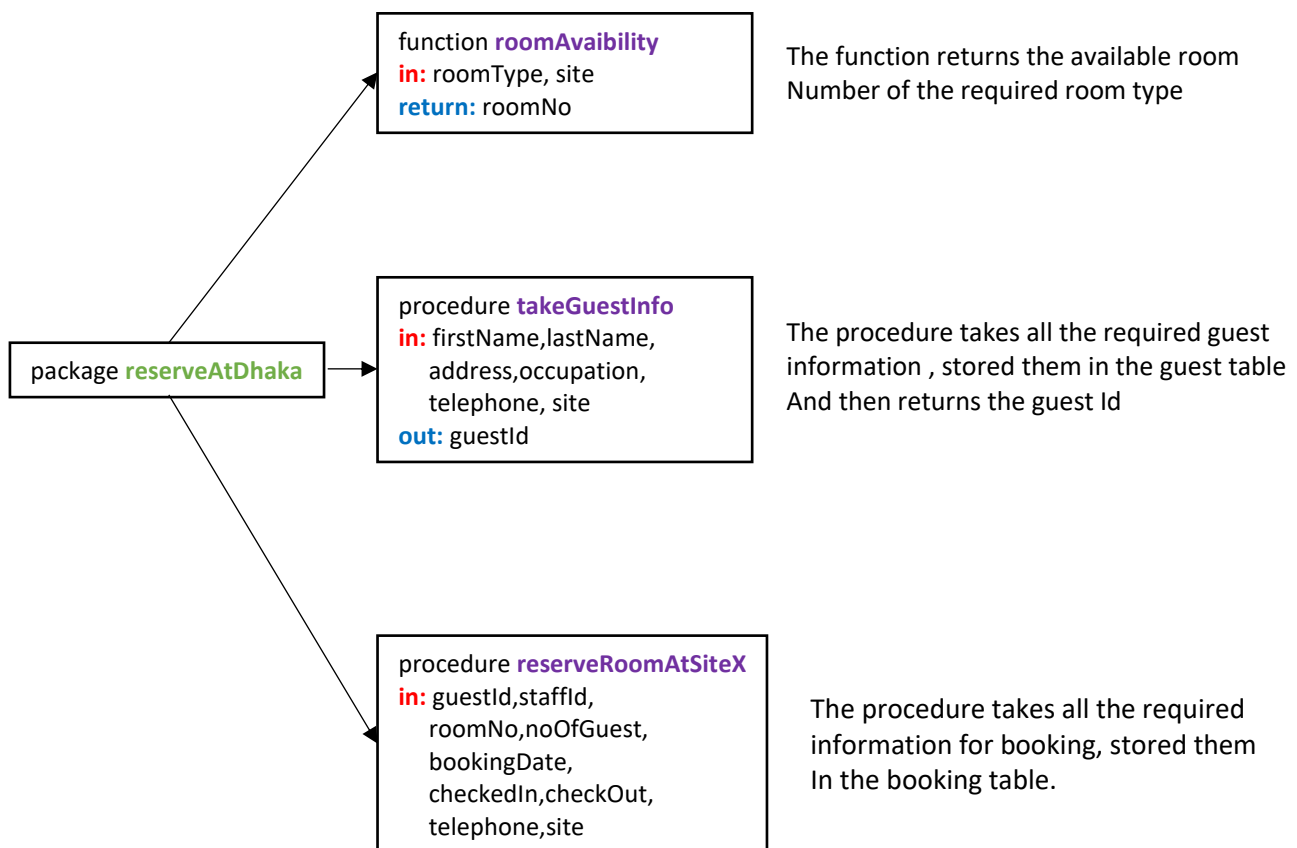
```
CREATE DATABASE LINK "SITE_NHK"
CONNECT TO "HOTELNHK" IDENTIFIED BY VALUES '05A8C2016CFBB8CC4F244116386FCECC8D'
USING '(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)
(HOST = 192.168.1.102)
(PORT = 1521))
)
(CONNECT_DATA =
(SID = XE)
)
)';
```

**Fig 4.3:** Site Link Code Of Noakhali As “site\_nhk”

## 5. Global & Local Applications

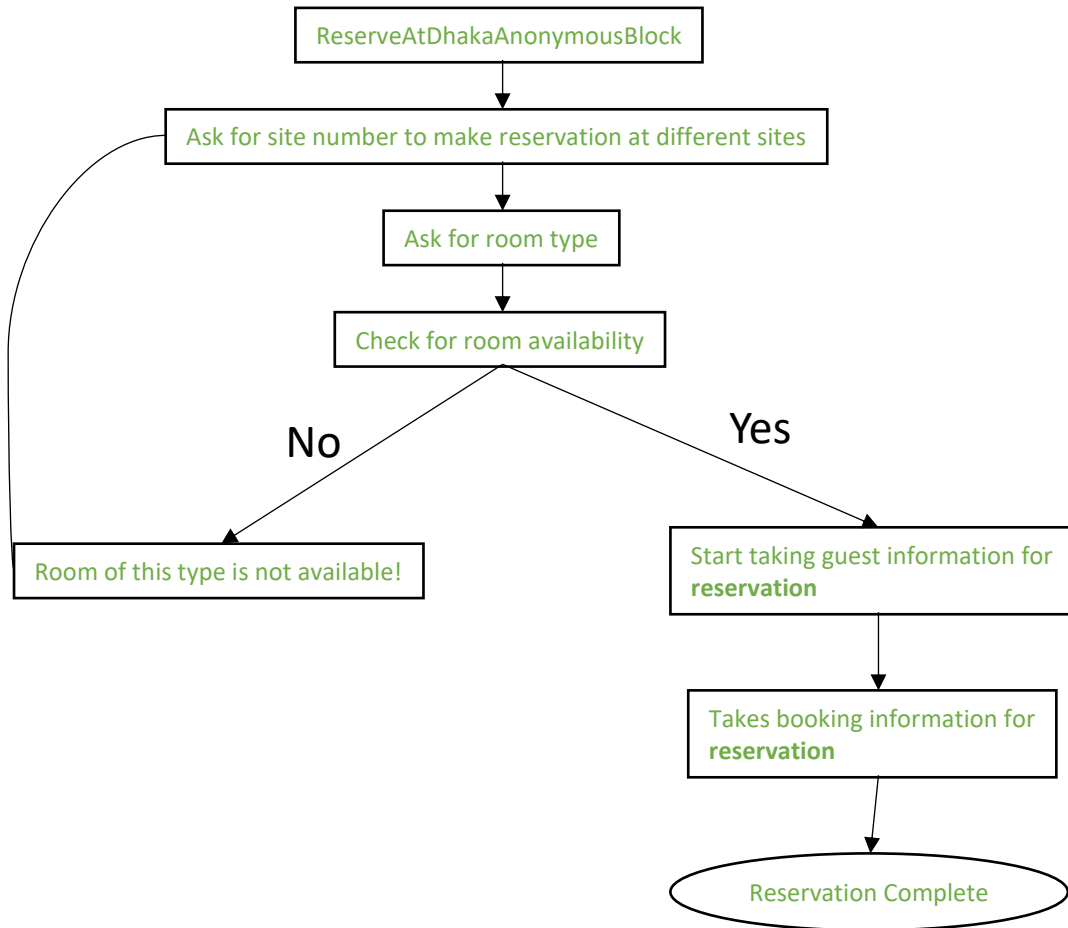
## Dhaka Branch (Server)

### Global Application (Works at the Server site/ Dhaka)

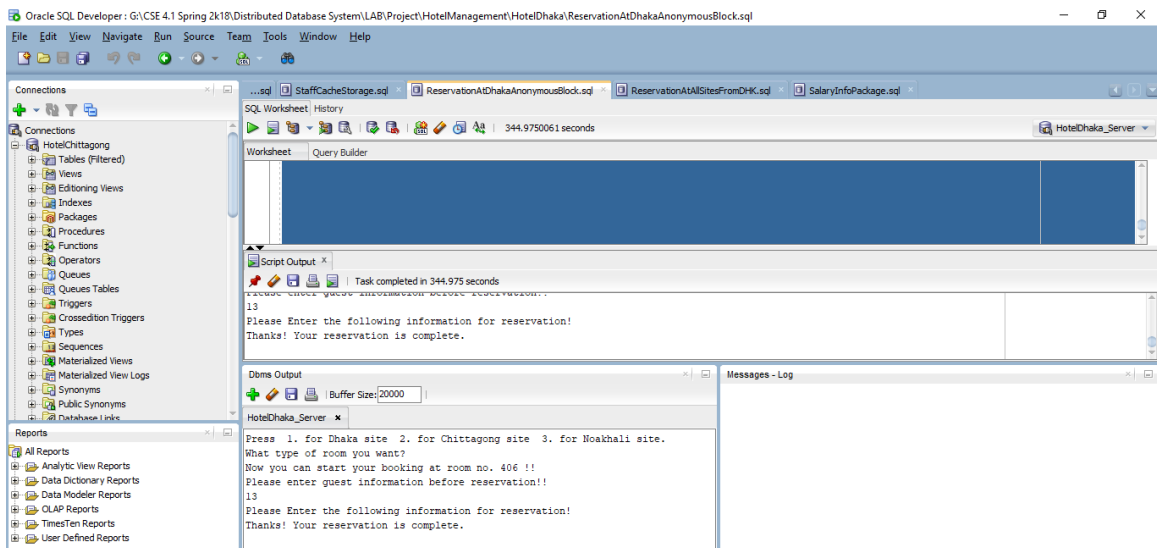
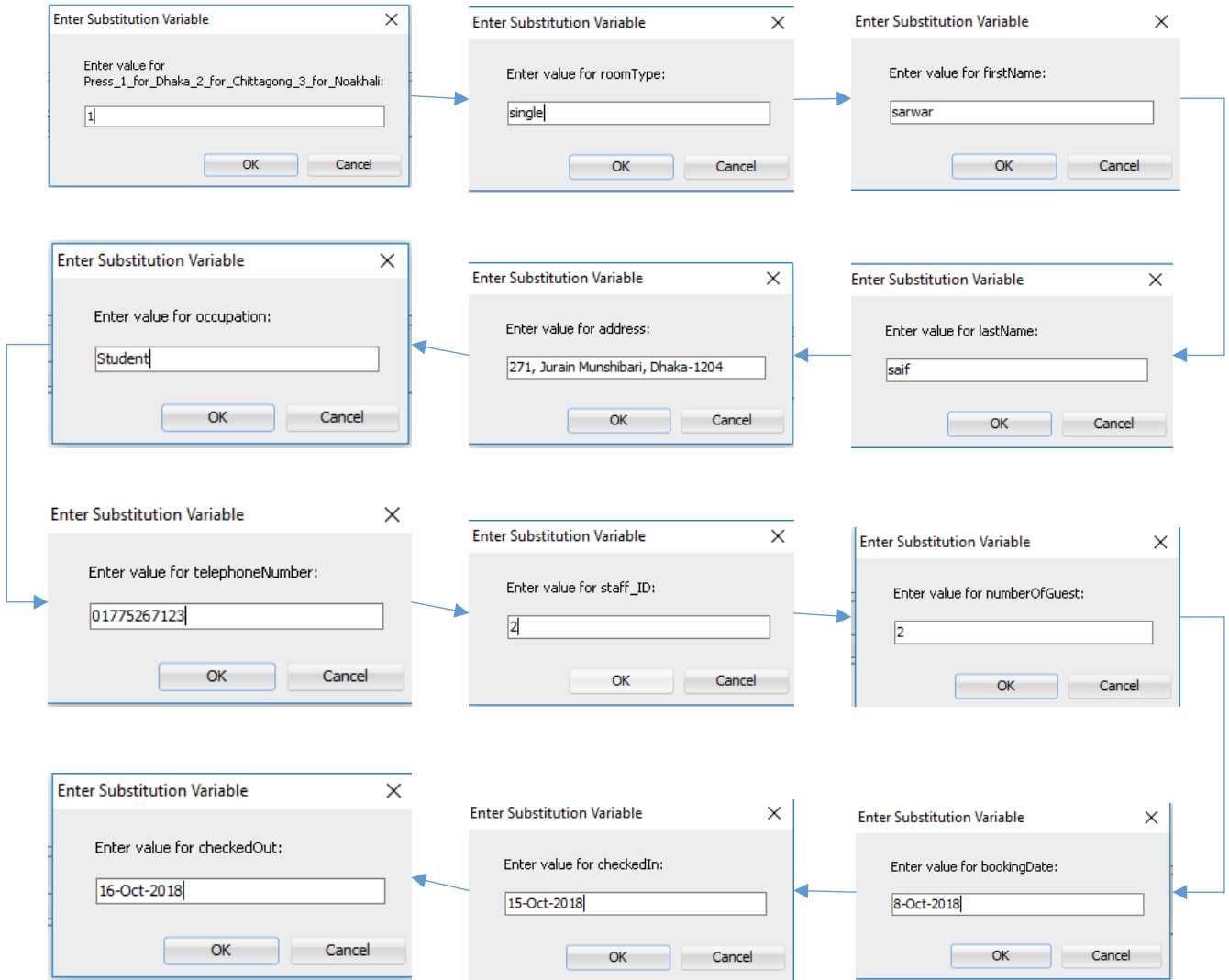


# Dhaka Branch (Server)

Global Application (Works at the Server site/ Dhaka)



# Dhaka Branch (Server)



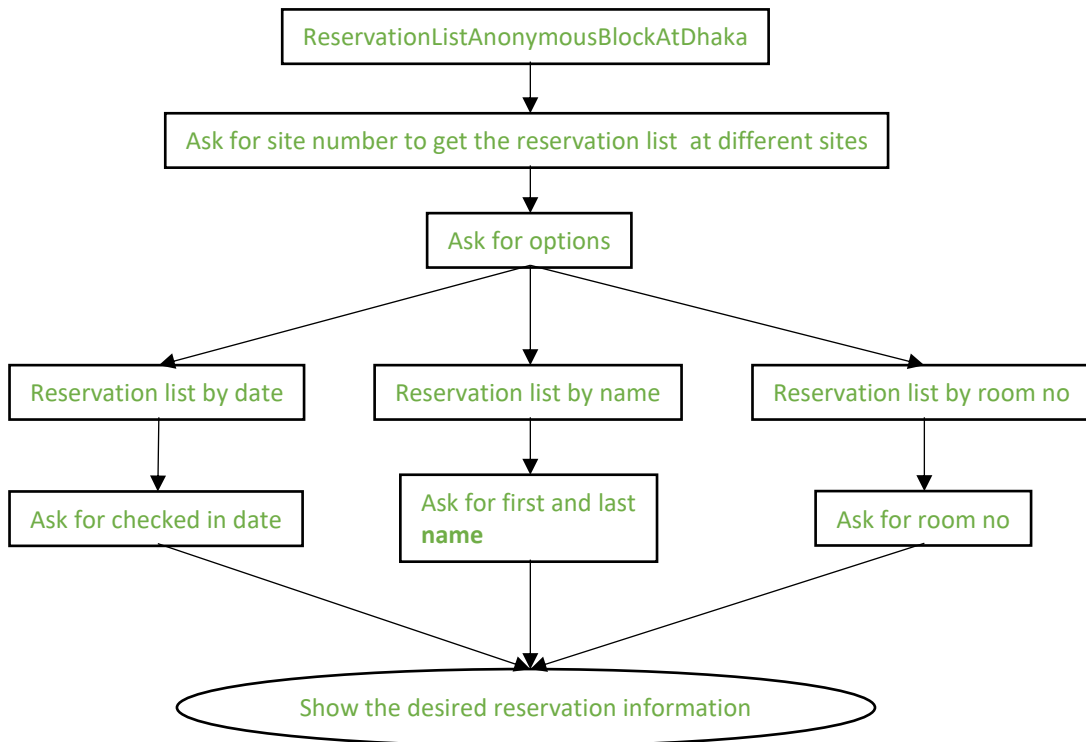
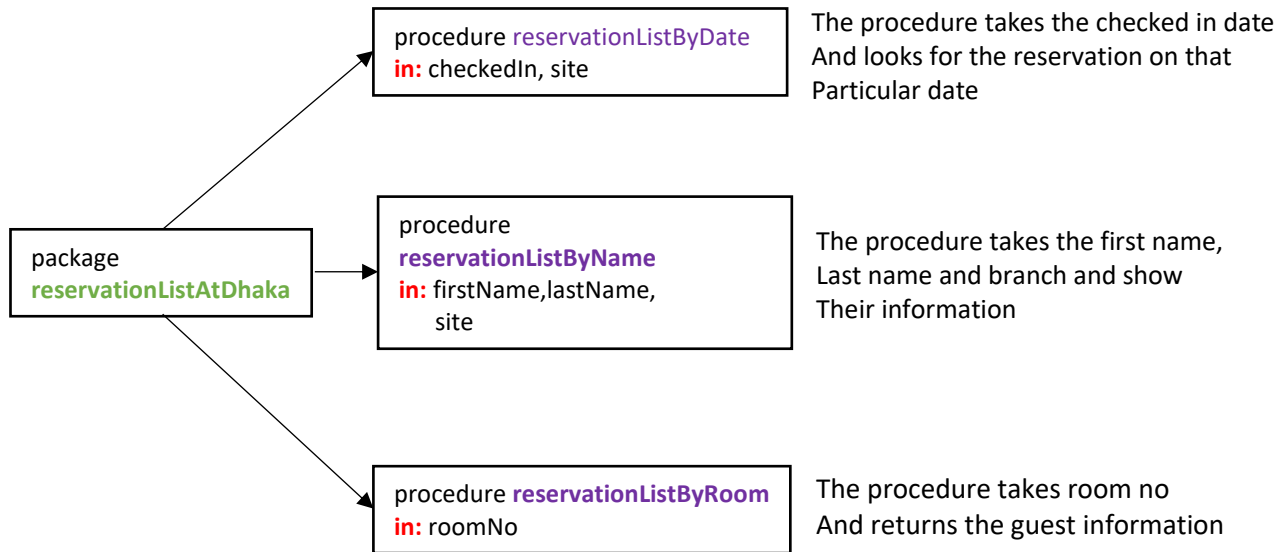
# Dhaka Branch (Server)

BOOKINGID	GUESTID	STAFFID	ROOMNO	NUMBEROFGUEST	BOOKINGDATE	CHECKEDIN	CHECKOUT	TELEPHONE	BRANCHID
1	1	1	1102	2	12-DEC-17	12-dec-2017	13-dec-2017	1765448952	1
2	2	2	1202	3	10-NOV-17	10-nov-2017	12-nov-2017	1865548962	1
3	3	3	1206	2	11-DEC-17	11-dec-2017	12-dec-2017	1965648932	1
4	4	4	1302	4	12-DEC-17	12-dec-2017	14-dec-2017	1555448952	1
5	5	5	2102	2	12-DEC-17	12-dec-2017	13-dec-2017	1765448952	2
6	6	6	2202	3	10-NOV-17	10-nov-2017	12-nov-2017	1865548962	2
7	7	7	2206	2	11-DEC-17	2:30pm	12:00am	1965648932	2
8	8	8	2302	4	12-DEC-17	2:30pm	11:00am	1555448952	2
9	9	9	3102	2	12-DEC-17	12-dec-2017	13-dec-2017	1765448952	3
10	10	10	3202	3	10-NOV-17	10-nov-2017	12-nov-2017	1865548962	3
11	11	11	3206	2	11-DEC-17	11-dec-2017	12-dec-2017	1965648932	3
12	12	12	3302	4	12-DEC-17	12-dec-2017	14-dec-2017	1555448952	3
13	13	13	3302	4	12-DEC-17	12-dec-2017	14-dec-2017	1555448952	1
14	14	13	2	406	208-OCT-18	15-Oct-2018	16-Oct-2018	1775267123	1

GUESTID	FIRSTNAME	LASTNAME	ADDRESS	OCCUPATION	TELEPHONE
1	abdullah-al	nasim	cumilla	student	1765448952
2	abdullah-al	hady	dhaka	student	1865548962
3	protty	mushfiq	cumilla	student	1965648932
4	shamit	ahmed	chattagram	student	1765444523
5	zahid	zahid	noakhali	student	1555448952
6	nasim-al	nasim	cumilla	student	1765448952
7	hady-al	hady	dhaka	student	1865548962
8	mushfiq	mushfiq	cumilla	student	1965648932
9	shamit	shamit	chattagram	student	1765444523
10	tanim	tanim	noakhali	student	1555448952
11	tanim	ahmed	chattagram	student	1765444523
12	ahmed	zahid	noakhali	student	1555448952
13	sarwar	saif	271, Jurain Munshibari, Dhaka-1204	Student	1775267123

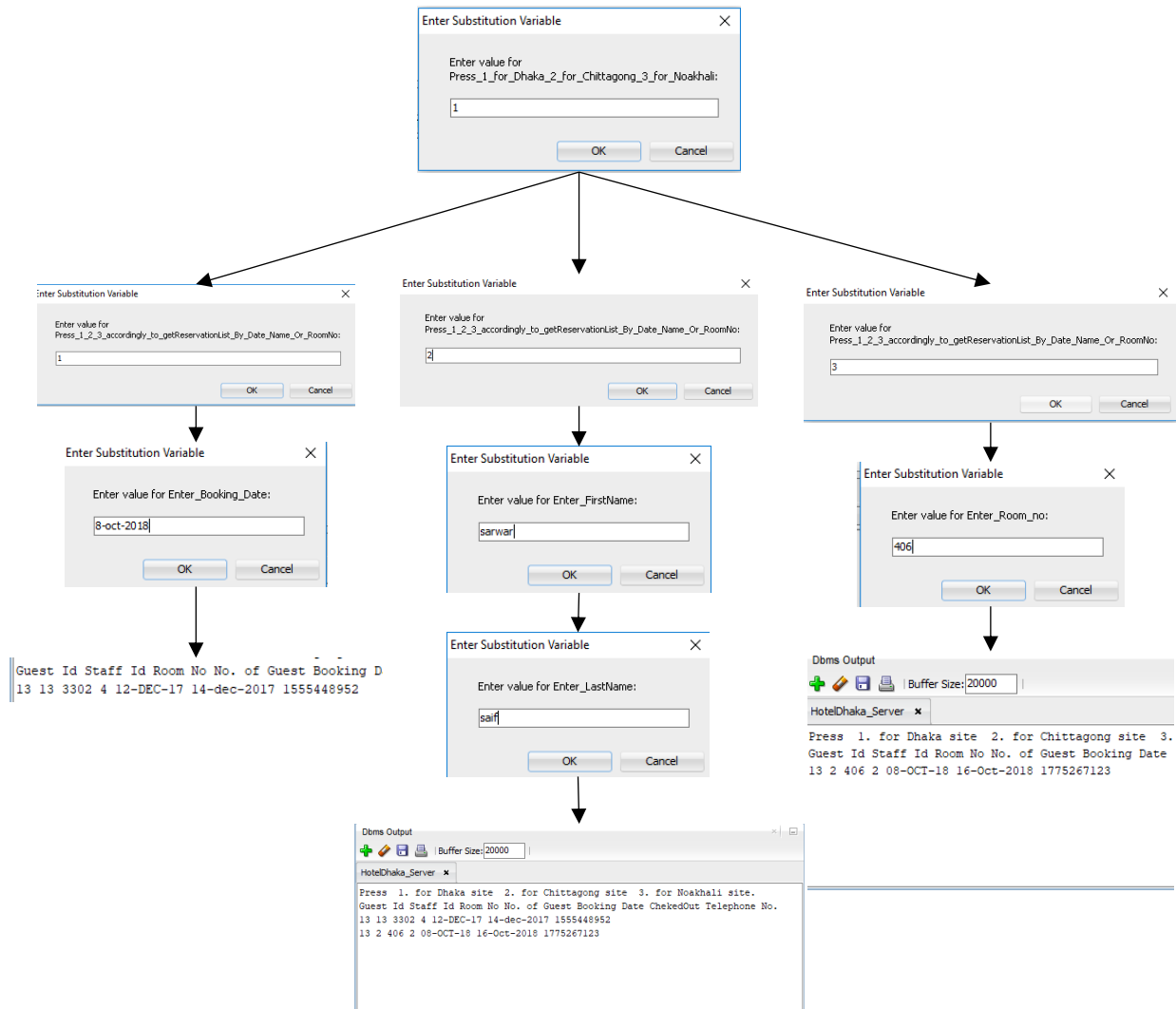
# Dhaka Branch (Server)

Global Application (Works at the Server site/ Dhaka)





# Dhaka Branch (Server)



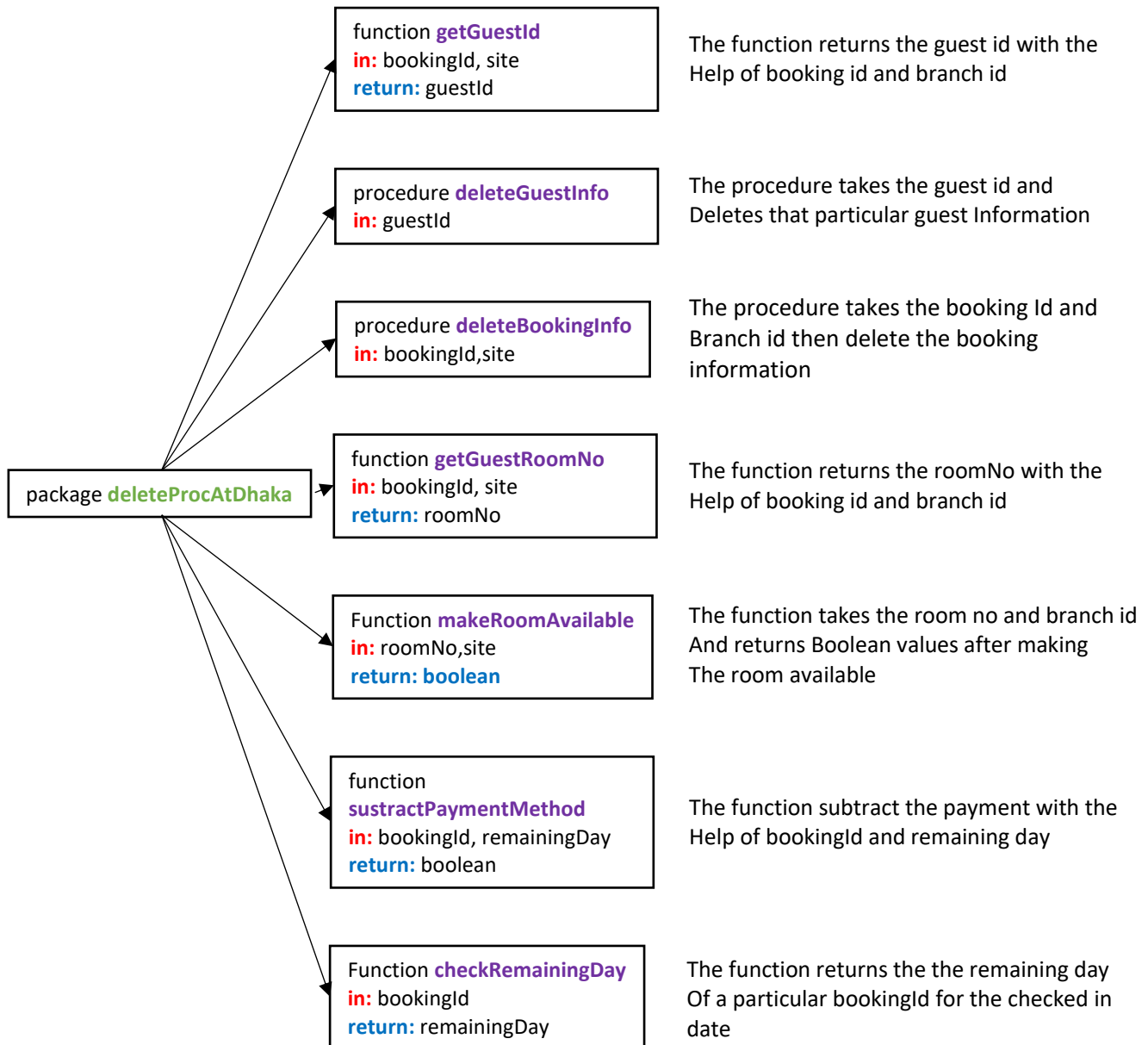
Guest Id Staff Id Room No. of Guest Booking D  
13 13 3302 4 12-DEC-17 14-dec-2017 1555448952

Dbms Output  
HotelDhaka\_Server  
Press 1. for Dhaka site 2. for Chittagong site 3.  
Guest Id Staff Id Room No. of Guest Booking Date  
13 2 406 2 08-OCT-18 16-Oct-2018 1775267123

Dbms Output  
HotelDhaka\_Server  
Press 1. for Dhaka site 2. for Chittagong site 3. for Noakhali site.  
Guest Id Staff Id Room No. of Guest Booking Date ChekedOut Telephone No.  
13 13 3302 4 12-DEC-17 14-dec-2017 1555448952  
13 2 406 2 08-OCT-18 16-Oct-2018 1775267123

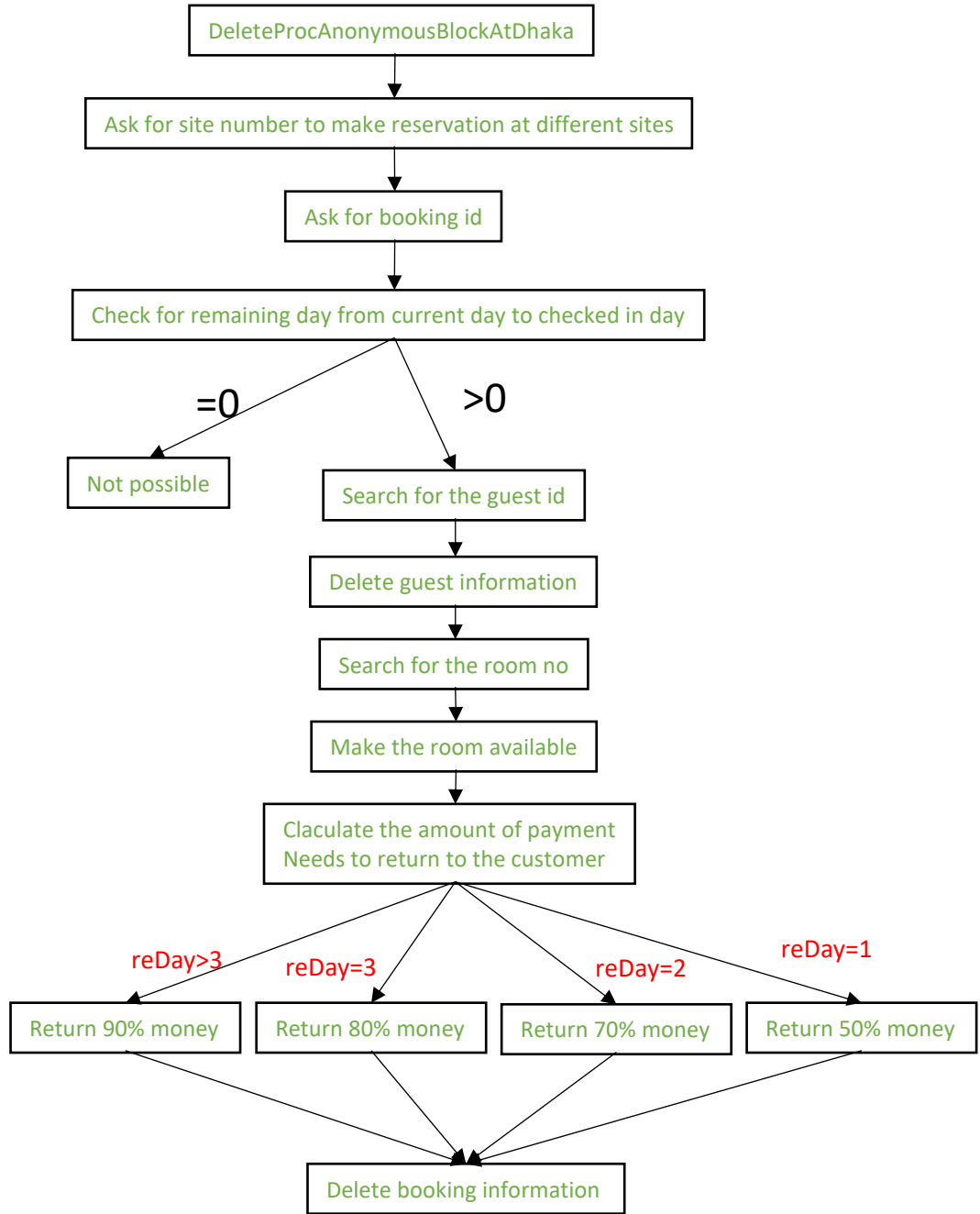
## Dhaka Branch (Server)

### Global Application (Works at the Server site/ Dhaka)



# Dhaka Branch (Server)

Global Application (Works at the Server site/ Dhaka)



# Dhaka Branch (Server)

Enter Substitution Variable

Enter value for Press\_1\_for\_Dhaka\_2\_for\_Chittagong\_3\_for\_Noakhali:

OK Cancel

Enter Substitution Variable

Enter value for Enter\_BookingId:

OK Cancel

8	8 mushfiq	mushfiq	cumilla	student	1965648932
9	9 shamit	shamit	chattagram	student	1765444523
10	10 tanim	tanim	noakhali	student	1555448952
11	11 tanim	ahmed	chattagram	student	1765444523
12	12 ahmed	zahid	noakhali	student	1555448952
13	13 sarwar	saif	271, Jurain, Munshibari, Dhaka-1204	Student	1775267123

7	7 hady-al	hady	dhaka	student	1865548962
8	8 mushfiq	mushfiq	cumilla	student	1965648932
9	9 shamit	shamit	chattagram	student	1765444523
10	10 tanim	tanim	noakhali	student	1555448952
11	11 tanim	ahmed	chattagram	student	1765444523
12	12 ahmed	zahid	noakhali	student	1555448952

9	9	9	9	3102	2 12-DEC-17	12-dec-2017	13-dec-2017	1765448952	3
10	10	10	10	3202	3 10-NOV-17	10-nov-2017	12-nov-2017	1865548962	3
11	11	11	11	3206	2 11-DEC-17	11-dec-2017	12-dec-2017	1965648932	3
12	12	12	12	3302	4 12-DEC-17	12-dec-2017	14-dec-2017	1555448952	3
13	13	13	13	3302	4 12-DEC-17	12-dec-2017	14-dec-2017	1555448952	1
14	14	13	2	406	2 08-OCT-18	15-Oct-2018	16-Oct-2018	1775267123	1

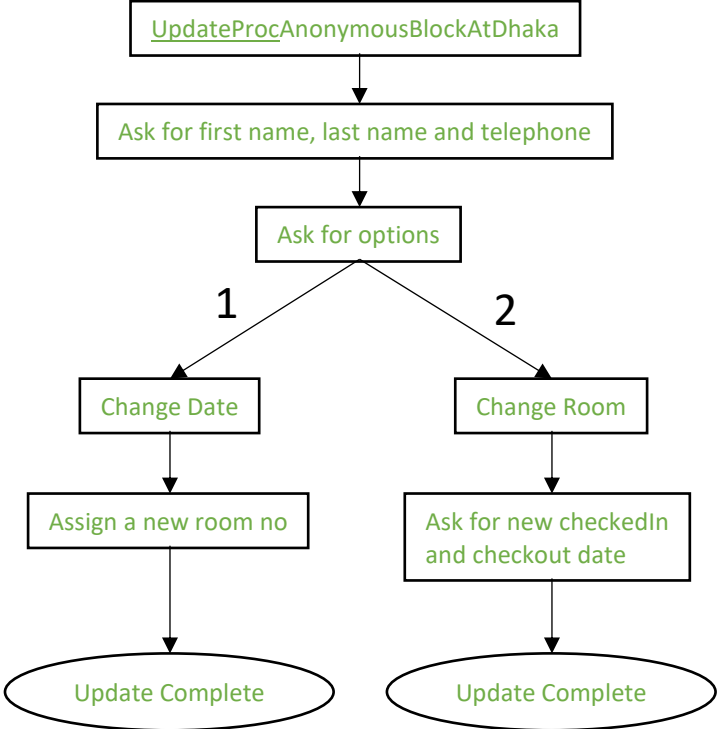
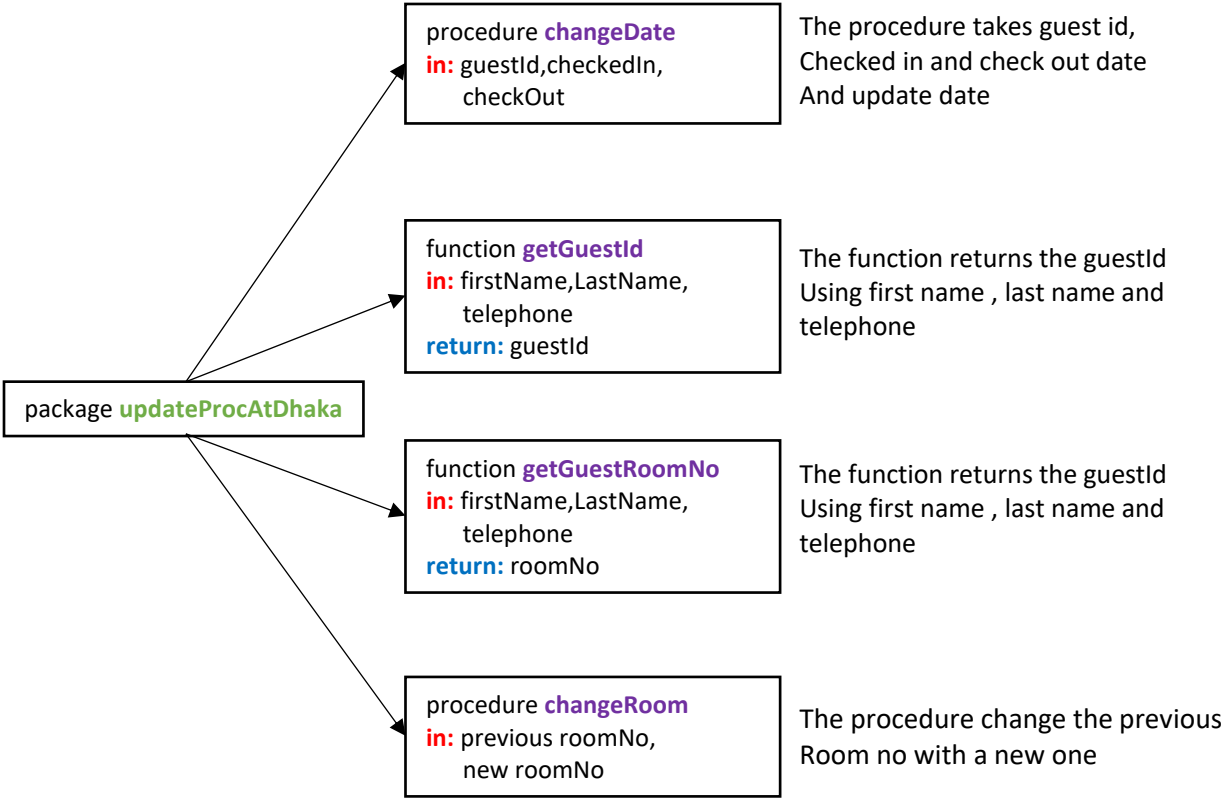
10	10	10	10	3202	3 10-NOV-17	10-nov-2017	12-nov-2017	1865548962	3
11	11	11	11	3206	2 11-DEC-17	11-dec-2017	12-dec-2017	1965648932	3
12	12	12	12	3302	4 12-DEC-17	12-dec-2017	14-dec-2017	1555448952	3
13	13	13	13	3302	4 12-DEC-17	12-dec-2017	14-dec-2017	1555448952	1

1	1	102	1 no
2	1	202	2 no
3	1	206	1 no
4	1	302	3 no
5	1	406	1 no
6	1	407	1 yes
7	1	502	3 yes
8	2	102	1 no

2	1	202	2 no
3	1	206	1 no
4	1	302	3 no
5	1	406	1 yes
6	1	407	1 yes
7	1	502	3 yes
8	2	102	1 no

# Dhaka Branch (Server)

Local Application (Works at the Server site/ Dhaka)



## Local Application (Works at the Server site/ Dhaka) Show Top N salaries

```
clear screen;
CREATE OR REPLACE PACKAGE salaryInfo AS
  TYPE measure_record IS RECORD(
    stId staff.staffId%Type,
    fname staff.firstname%Type,
    lname staff.lastname%TYPE,
    address staff.address%TYPE,
    salary number
  );
  TYPE measure_table IS TABLE OF measure_record;
  FUNCTION get_top_N_salaries(foo NUMBER)
    RETURN measure_table
    PIPELINED;
END salaryInfo;
/
CREATE OR REPLACE PACKAGE BODY salaryInfo AS
  FUNCTION get_top_N_salaries(foo number)
    RETURN measure_table
    PIPELINED IS
    rec measure_record;
    cursor cur is
      SELECT distinct (staff.staffId), staff.firstName, staff.lastName, staff.address,salary.amount
      --INTO rec
      FROM Staff ,Salary
      where staff.jobId=salary.jobId and rownum<foo
      Order By salary.amount DESC
    ;
  BEGIN
    open cur;
    loop
      fetch cur into rec;
      exit when cur%notfound;
      PIPE ROW (rec);
      DBMS_output.Put_line(to_char(rec.stId)||' '||rec.fname||
        ' '||rec.lname||' '||rec.address||' '||to_char(rec.salary));
    end loop;
    close cur;
    RETURN;
  END get_top_N_salaries;
END salaryInfo;
/
SELECT *
FROM table(salaryInfo.get_top_N_salaries(4));
```

Use of TYPE  
attribute

Tried to use the  
Pipelined table  
function  
Which makes the  
code much more  
elegant

## Local Application (Works at the Server site/ Dhaka)

### Show Employee Information from **cache**

```
create or replace FUNCTION one_employee (employee_id_in
IN staff.staffId%TYPE)
RETURN staff%ROWTYPE
RESULT_CACHE RELIES_ON (staff)
IS
  l_employee staff%ROWTYPE;
BEGIN
  SELECT *
  INTO l_employee
  FROM staff
  WHERE staffId = employee_id_in;
  RETURN l_employee;
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN
    /* Return an empty record. */
    RETURN l_employee;
END one_employee;
/
SET SERVEROUTPUT ON;
declare
  l_employee staff%ROWTYPE;
begin
  l_employee:=one_employee(1);
  DBMS_output.Put_line(to_char(l_employee.staffId)||'|'||to_char(l_employee.jobId)||'|'
||l_employee.firstname||'|'
  ||l_employee.lastname||'|' ||l_employee.address||'|' ||to_char(l_employee.bookingId));
End;
/
```

Tried to implement the new feature of Oracle Database 11g Which is “Result\_Cache”

### Why result cache?

This RESULT\_CACHE clause tells Oracle Database that it should remember (store in a special in-memory result cache) each record retrieved for a specific employee ID number. And when a session executes this function and passes in an employee ID that was previously stored, the PL/SQL runtime engine will not execute the function body, which includes that query.

Instead, it will simply retrieve the record from the cache and return that data immediately. The result is much faster retrieval.

In addition, by specifying RELIES\_ON (employees), we inform Oracle Database that if any session commits changes to that table, any data in the result cache drawn from the table must be invalidated. The next call to the one\_employee function would then have to execute the query and retrieve the data fresh from the table.

\*\*\*This makes the query more faster for working with huge datasets

## **Chittagong and Noakhali Branch ( Local Applications)**

### **Reservation Application-**

Package is named reserveAtCtg/ reserveAtNhk.

- procedure takeGuestInfo
- function roomAvaibility
- procedure reserveRoomAtSiteX

The reserve application is almost the same like Dhaka branch but they lack the option to choose sites as the site is fixed as 2 for chittagong and 3 for noakhali as both of them are local applications.

### **Reservation List Application-**

- procedure reservationListByDate
- procedure reservationListByName
- procedure reservationListByRoom

### **Update Application-**

- function getGuestId
- procedure changeDate
- function getGuestRoomNo
- procedure changeRoom

### **Delete Application-**

- function checkRemainingDay
- function getGuestId
- procedure deleteGuestInfo
- function getGuestRoomNo
- procedure makeRoomAvailable
- function subtractPaymentMethod
- procedure deleteBookingInfo



## Triggers

```
--payment trigger
set serveroutput on;
create or replace trigger trigger_payment
after insert on payment
for each row
begin
    insert into log_insert_payment
    values(:new.bookingid,:new.amount, sysdate);
    DBMS_OUTPUT.PUT_LINE('payment done');
end;
/
--delete payment trigger
set serveroutput on;
create or replace trigger trigger_payment
after delete on payment
for each row
begin
    DBMS_OUTPUT.PUT_LINE('payment removed');
end;
/
--INSERT TRIGGER
create or replace trigger trigger_new_GUEST
after insert on guest
for each row
begin
    insert into log_guest_insert
    values(:new.guestId, sysdate);
    DBMS_OUTPUT.PUT_LINE('guest info');
end;
/
--delete TRIGGER
create or replace trigger trigger_new_GUEST
after delete on guest
for each row
begin

    DBMS_OUTPUT.PUT_LINE('guest deleted');
end;
/
```

## Triggers

```
set serveroutput on;
create or replace trigger trigger_new_booking
after insert on booking
for each row
begin
    insert into log_booking_insert
    values(:new.bookingId, sysdate);
    DBMS_OUTPUT.PUT_LINE('A new guest has booked');
end;
/
--delete trigger
set serveroutput on;
create or replace trigger trigger_new_booking
after delete on booking
for each row
begin

    DBMS_OUTPUT.PUT_LINE('guest removed');
end;
/
--update branchId
create or replace trigger trigger_update_branch
after update on booking
for each row
begin
    insert into log_update_bookingbranch
    values(:new.bookingId, :old.branchId, :new.branchId, sysdate);
    DBMS_OUTPUT.PUT_LINE('updated branch id');
end;
/
--checkin/checkoyt triggger
create or replace trigger trigger_checkout
after insert on booking
for each row
begin
    insert into log_checkout
    values(:new.bookingId,:new.checkedin,:new.checkout);
end;
/
commit;
```

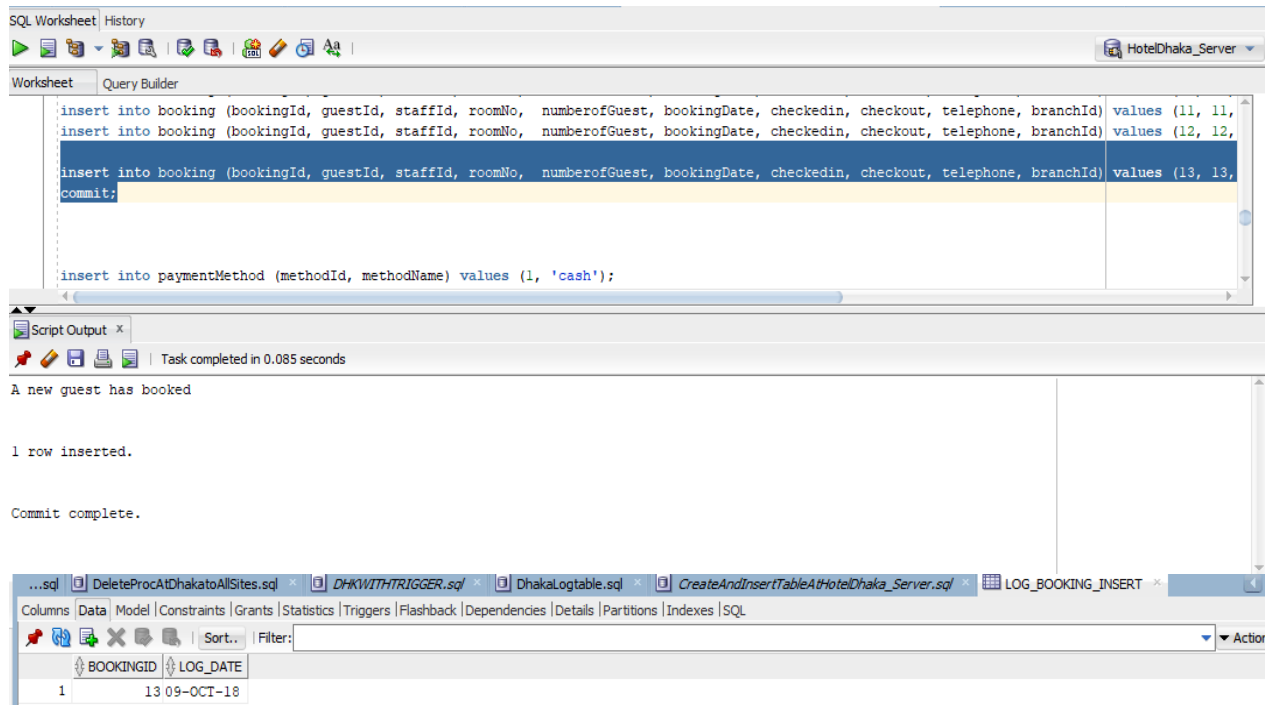


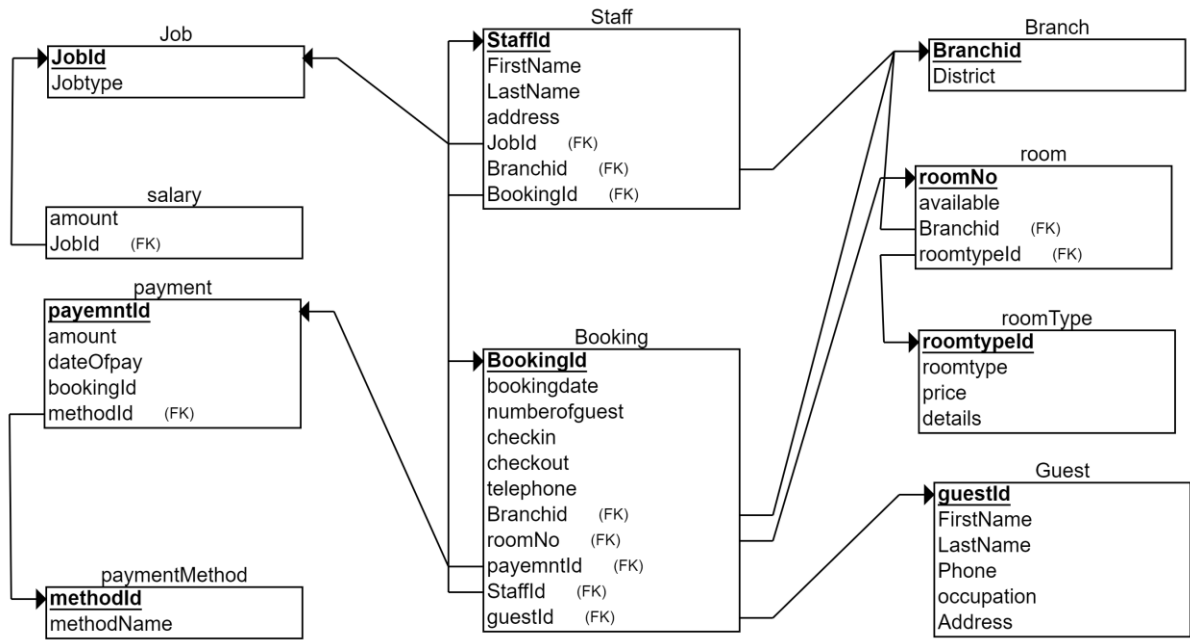
Fig: Trigger after insert

### About my contribution in this project:

As I was busy because of codeware, I could not able to contribute my 100% in the mid evaluation. As a result, the project was not up to your expectation I guess. But after that, I have tried my best to make the project as good as possible. I tried to implement all the things I have learned from the lab and with the help of google. I think the project is not about just writing some codes lines after lines, it needs to be efficient and more detailed. I tried to implement the project in that way. Alhamdulillah, the project hardly has any errors. I tried to learn some new features and I found there are still so many things left to learn. I actually enjoyed doing the project. In Sha Allah, I'll try to learn more things in the near future.

After reading three of our reports, I guess you'll understand all of our contributions in this project.

Thank you!



**Fig: Relational Schema**