

Final Project

Hotel Management



|  |  |
| --- | --- |
| Group Member | Contact Email |
| Kishan | [kishaneshiram@gmail.com](mailto:kishaneshiram@gmail.com) |
| Saros | [saroshmalik269@gmail.com](mailto:saroshmalik269@gmail.com) |
| Peiran Liu | [liupeiran9324@outlook.com](mailto:liupeiran9324@outlook.com) |
| Zeyu Ma | [zma29@my.centennialcollege.ca](mailto:zma29@my.centennialcollege.ca) |

January 9, 2018

Contents

[Hotel Management 1](#_Toc503255759)

[Problem Description 1](#_Toc503255760)

[Use case 1: Find employee by job 2](#_Toc503255761)

[Use case 2: Display all employees 2](#_Toc503255762)

[Use case 3: Find room information and payment method based on room size 2](#_Toc503255763)

[Use case 4: Check room level 2](#_Toc503255764)

[Use case 5: Find request and number of guests by room 2](#_Toc503255765)

[Use case 6: Save room history before modifying 2](#_Toc503255766)

[Use case 7: Change payment method 2](#_Toc503255767)

[Creation of table, relation queries 3](#_Toc503255768)

[Employee Table 3](#_Toc503255769)

[Room Inventory Table 4](#_Toc503255770)

[Work Orders Table 5](#_Toc503255771)

[Guest Table 8](#_Toc503255772)

[Reservation Table 9](#_Toc503255773)

[Special Request Table 11](#_Toc503255774)

[ERD Diagram (Data Model and Relations) 12](#_Toc503255775)

[Program Source Code 13](#_Toc503255776)

[Procedures 13](#_Toc503255777)

# Hotel Management

## Problem Description

Orange-Well hotel is an established property in the heart of Western Ontario. The hotel is managed and owned through a family-oriented business structure. The property has been handed down from generation to generation and the issue the hotel faces comes from attracting a larger clientele as it is only known to locals and by word of mouth. The hotel has decided to invest more in their marketing campaign and have decided to partner with online advertisers to increase their occupancy. In order to accommodate for this the hotel is going to need a new database system. Their reservation system is not up to date and they are still following the old filing system of paper registrations and storing guest payment information in physical filing cabinets. Currently the hotel only accepts registrations through walk-in and over the phone which will change as the new system is applied. The problem about the current registration system is that employees must handle customer orders and requests manually which is time consuming and error pron. Further, it requires more employees to be hired to handle the substantial number of guest registrations. The challenge for the Orange-Well hotel is keeping hardcopies of all visited guest which requires space to store the file and is inconvenient when retrieving guest payment information. The switch to a modern database system will enable the hotel to operate smoothly. The new system will be able to store the guest information including the address and contact while also securing the guests payment information for later reference. This will also allow employees to directly enter information into the database and will also allow for online registration to the database as well.

## Use case 1: Find employee by job

The hotel manager needs to find employees in a certain field.

## Use case 2: Display all employees

The hotel manager needs to know about all the employees

## Use case 3: Find room information and payment method based on room size

The hotel manager needs to know about the general information about rooms and how did guests paid for it by the size of the room

## Use case 4: Check room level

The hotel manager needs to check the level of the room based on the size of the room. Different rooms will be categorized into different sizes based on their square feet.

## Use case 5: Find request and number of guests by room

The hotel manager would like to know which room has special request and the number of guests made the request.

## Use case 6: Save room history before modifying

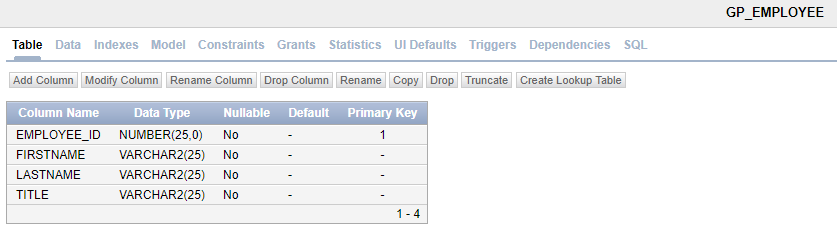
The hotel manager would like to save the previous history of a certain room before having the room reconstructed.

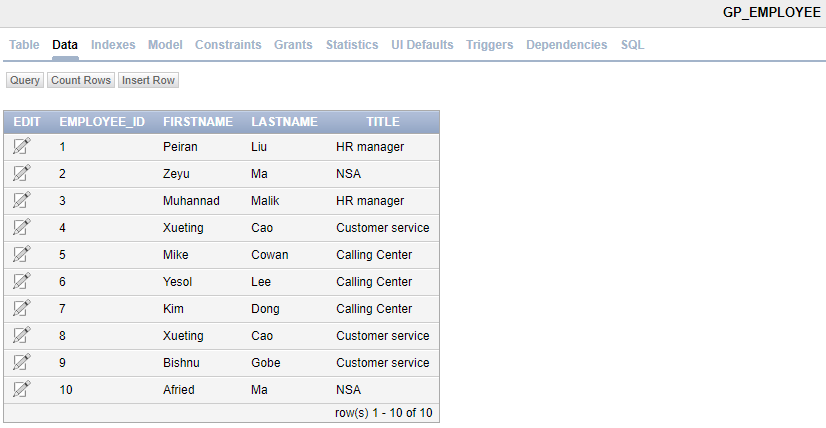
## Use case 7: Change payment method

The hotel manager needs to know the updated version of payment method, due to a change of guests’ id numbers.

## Creation of table, relation queries

### Employee Table





CREATE TABLE GP\_EMPLOYEE

(EMPLOYEE\_ID NUMBER(25) NOT NULL,

FIRSTNAME VARCHAR2(25) NOT NULL,

LASTNAME VARCHAR2(25) NOT NULL,

TITLE VARCHAR2(25) NOT NULL,

CONSTRAINT gp\_employee\_employee\_id\_pk PRIMARY KEY (EMPLOYEE\_ID));

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0001,'Peiran','Liu','HR manager');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0002,'Zeyu','Ma','NSA');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0003,'Muhannad','Malik','HR manager');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0004,'Xueting','Cao','Customer service');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0005,'Mike','Cowan','Calling Center');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0006,'Yesol','Lee','Calling Center');

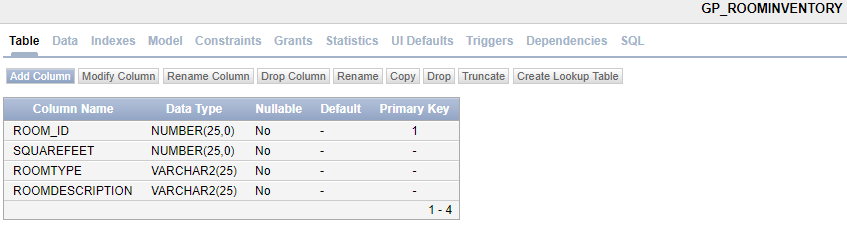
insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0007,'Kim','Dong','Calling Center');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0008,'Xueting','Cao','Customer service');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0009,'Bishnu','Gobe','Customer service');

insert into GP\_Employee (Employee\_ID,FirstName,LastName,Title) values(0010,'Afried','Ma','NSA');

### Room Inventory Table



CREATE TABLE GP\_ROOMINVENTORY

(ROOM\_ID NUMBER(25) NOT NULL,

SQUAREFEET NUMBER(25) NOT NULL,

ROOMTYPE VARCHAR2(25) NOT NULL,

ROOMDESCRIPTION VARCHAR2(25) NOT NULL, --changed to ROOMDESCRIPTION

CONSTRAINT gp\_roominventory\_room\_id\_pk PRIMARY KEY (ROOM\_ID));

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(001,200,'Single','face to sun');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(002,400,'Double','has a really huge bed');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(003,160,'Single','has a really big table');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(004,450,'King','very good');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(005,130,'Single','good');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(006,130,'Single','trash');

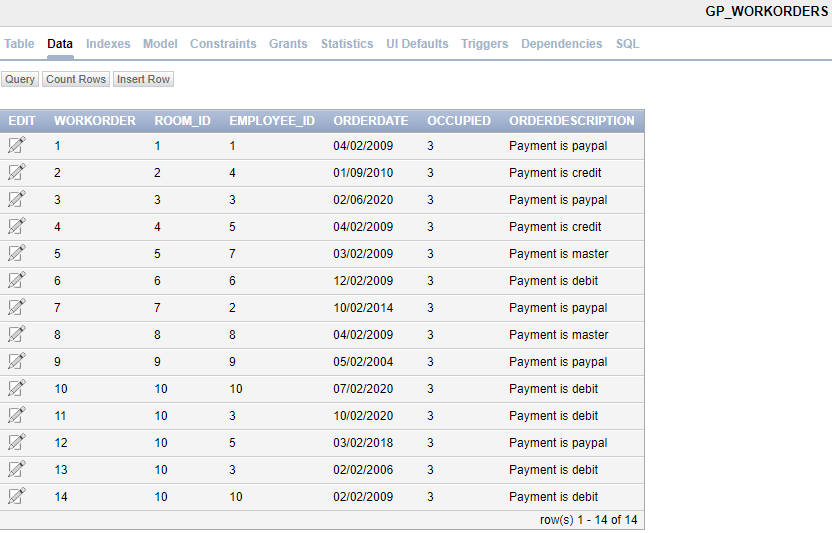
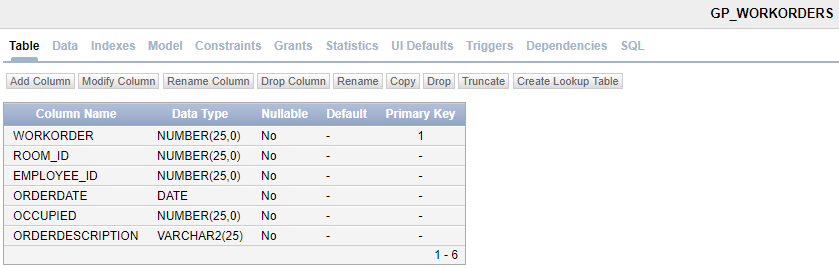
insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(007,140,'Single','bad');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(008,80,'Single','really small');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(009,600,'double','Party room special');

insert into GP\_RoomInventory (Room\_ID,SquareFeet,RoomType,ROOMDESCRIPTION) values(010,600,'double','Party room special2');

### Work Orders Table



CREATE TABLE GP\_WORKORDERS

(WORKORDER NUMBER(25) NOT NULL,

ROOM\_ID NUMBER(25) NOT NULL,

EMPLOYEE\_ID NUMBER(25) NOT NULL,

ORDERDATE DATE NOT NULL, --DATE is a key word, it has been changed to ORDERDATE for readability

OCCUPIED NUMBER(25)NOT NULL,

ORDERDESCRIPTION VARCHAR2(25)NOT NULL, --description is a keyword goddammit!!!!!!!

CONSTRAINT gp\_workorders\_workorder\_pk PRIMARY KEY (WORKORDER),

CONSTRAINT gp\_workorders\_room\_id\_fk FOREIGN KEY (ROOM\_ID)

REFERENCES gp\_roominventory(ROOM\_ID),

CONSTRAINT gp\_workorders\_employee\_id\_fk FOREIGN KEY (EMPLOYEE\_ID)

REFERENCES gp\_employee(EMPLOYEE\_ID));

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(001,001,0001,TO\_DATE('02-APR-09','DD-MON-YY'),3,'Payment is paypal');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(002,002,0004,TO\_DATE('09-JAN-10','DD-MON-YY'),3,'Payment is credit');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(003,003,0003,TO\_DATE('06-FEB-20','DD-MON-YY'),3,'Payment is paypal');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(004,004,0005,TO\_DATE('02-APR-09','DD-MON-YY'),3,'Payment is credit');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(005,005,0007,TO\_DATE('02-MAR-09','DD-MON-YY'),3,'Payment is master');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(006,006,0006,TO\_DATE('02-DEC-09','DD-MON-YY'),3,'Payment is debit');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(007,007,0002,TO\_DATE('02-OCT-14','DD-MON-YY'),3,'Payment is paypal');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(008,008,0008,TO\_DATE('02-APR-09','DD-MON-YY'),3,'Payment is master');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(009,009,0009,TO\_DATE('02-MAY-04','DD-MON-YY'),3,'Payment is paypal');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(010,010,0010,TO\_DATE('02-JUL-20','DD-MON-YY'),3,'Payment is debit');

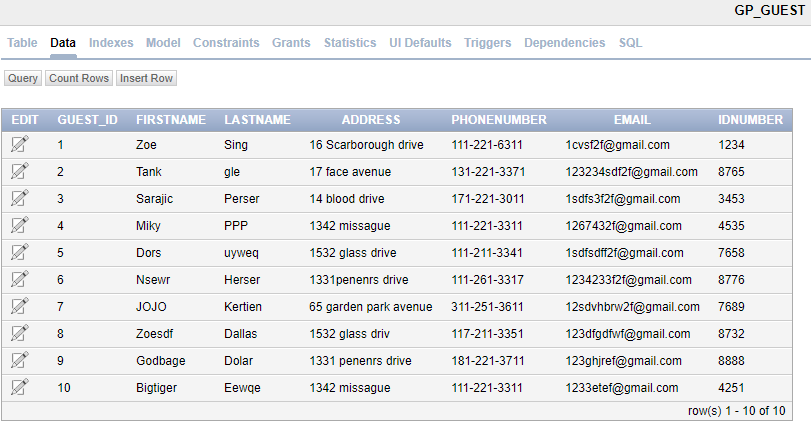
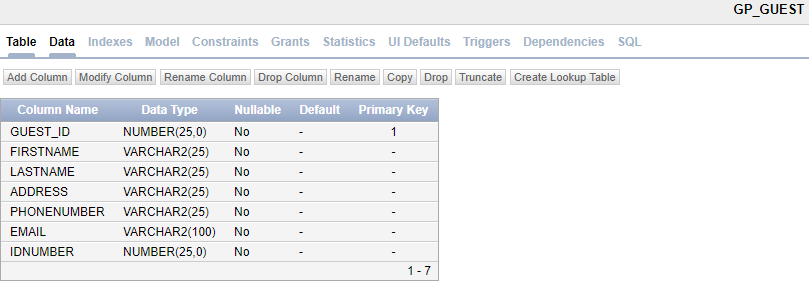
insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(011,010,0003,TO\_DATE('02-OCT-20','DD-MON-YY'),3,'Payment is debit');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(012,010,0005,TO\_DATE('02-MAR-18','DD-MON-YY'),3,'Payment is paypal');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(013,010,0003,TO\_DATE('02-FEB-06','DD-MON-YY'),3,'Payment is debit');

insert into GP\_WORKORDERs (WorkOrder,Room\_ID,Employee\_ID,ORDERDATE,Occupied,OrderDescription) values(014,010,0010,TO\_DATE('02-FEB-09','DD-MON-YY'),3,'Payment is debit');

### Guest Table



CREATE TABLE GP\_GUEST

(GUEST\_ID NUMBER(25) NOT NULL,

FIRSTNAME VARCHAR2(25) NOT NULL,

LASTNAME VARCHAR2(25) NOT NULL,

ADDRESS VARCHAR2(25) NOT NULL,

PHONENUMBER VARCHAR2(25) NOT NULL,

EMAIL VARCHAR2(100) NOT NULL,

IDNUMBER NUMBER(25) NOT NULL,

CONSTRAINT gp\_guest\_guest\_id\_pk PRIMARY KEY (GUEST\_ID));

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(001,'Zoe','Sing','16 Scarborough drive','111-221-6311','1cvsf2f@gmail.com',1234);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(002,'Tank','gle','17 face avenue','131-221-3371','123234sdf2f@gmail.com',8765);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(003,'Sarajic','Perser','14 blood drive','171-221-3011','1sdfs3f2f@gmail.com',3453);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(004,'Miky','PPP','1342 missague','111-221-3311','1267432f@gmail.com',4535);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(005,'Dors','uyweq','1532 glass drive','111-211-3341','1sdfsdff2f@gmail.com',7658);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(006,'Nsewr','Herser','1331penenrs drive','111-261-3317','1234233f2f@gmail.com',8776);

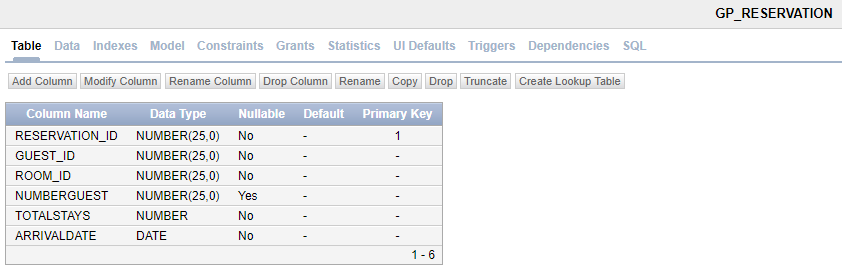
insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(007,'JOJO','Kertien','65 garden park avenue','311-251-3611','12sdvhbrw2f@gmail.com',7689);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(008,'Zoesdf','Dallas','1532 glass driv','117-211-3351','123dfgdfwf@gmail.com',8732);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(009,'Godbage','Dolar','1331 penenrs drive','181-221-3711','123ghjref@gmail.com',1237);

insert into GP\_Guest(Guest\_ID,FirstName,LastName,Address,PhoneNumber,Email,IDNumber) values(010,'Bigtiger','Eewqe','1342 missague','111-221-3311','1233etef@gmail.com',4251);

### Reservation Table



CREATE TABLE GP\_RESERVATION

(RESERVATION\_ID NUMBER(25) NOT NULL,

GUEST\_ID NUMBER(25) NOT NULL,

ROOM\_ID NUMBER(25) NOT NULL,

NUMBERGUEST NUMBER(25),

TOTALSTAYS NUMBER NOT NULL,

ARRIVALDATE DATE NOT NULL,

CONSTRAINT gp\_res\_reservation\_id\_pk PRIMARY KEY (RESERVATION\_ID),

CONSTRAINT gp\_res\_guest\_id\_fk FOREIGN KEY (GUEST\_ID)

REFERENCES gp\_guest(GUEST\_ID),

CONSTRAINT gp\_res\_room\_id\_fk FOREIGN KEY (ROOM\_ID)

REFERENCES gp\_roominventory(ROOM\_ID));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(001,001,010,3,6,TO\_DATE('02-APR-09','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(002,002,009,4,7,TO\_DATE('02-APR-10','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(003,003,008,5,8,TO\_DATE('02-APR-11','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(004,004,007,6,9,TO\_DATE('02-APR-15','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(005,005,006,7,10,TO\_DATE('02-APR-17','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(006,006,005,8,11,TO\_DATE('02-APR-19','DD-MON-YY'));

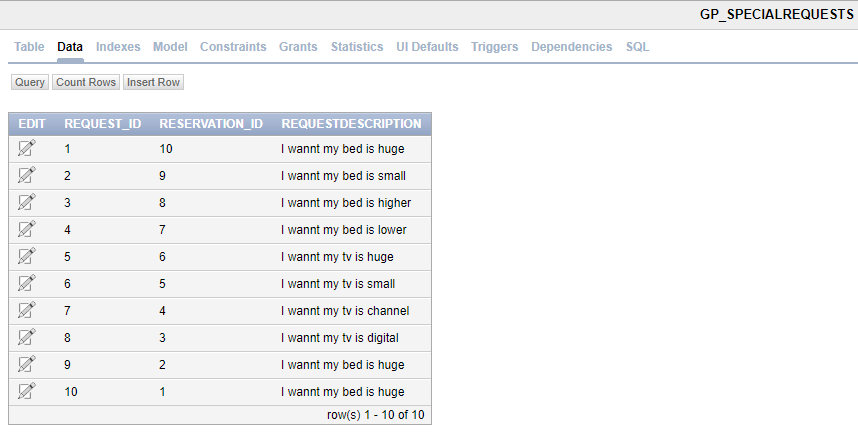
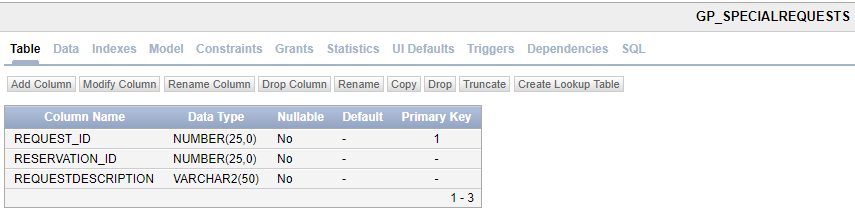
insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(007,007,004,9,12,TO\_DATE('02-APR-20','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(008,008,003,10,13,TO\_DATE('02-APR-24','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(009,009,002,11,14,TO\_DATE('02-APR-26','DD-MON-YY'));

insert into GP\_Reservation (Reservation\_ID,Guest\_ID,Room\_ID,NumberGuest,TotalStays,ArrivalDate) values(010,010,001,12,15,TO\_DATE('02-APR-28','DD-MON-YY'));

### Special Request Table



CREATE TABLE GP\_SPECIALREQUESTS

(REQUEST\_ID NUMBER(25) NOT NULL,

RESERVATION\_ID NUMBER(25) NOT NULL,

REQUESTDESCRIPTION VARCHAR2(50) NOT NULL, --changed to REQUESTDESCRIPTION

CONSTRAINT gp\_spec\_request\_id\_pk PRIMARY KEY (REQUEST\_ID),

CONSTRAINT gp\_spec\_res\_id\_fk FOREIGN KEY (RESERVATION\_ID)

REFERENCES gp\_reservation(RESERVATION\_ID));

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(001,010,'I wannt my bed is huge');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(002,009,'I wannt my bed is small');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(003,008,'I wannt my bed is higher');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(004,007,'I wannt my bed is lower');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(005,006,'I wannt my tv is huge');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(006,005,'I wannt my tv is small');

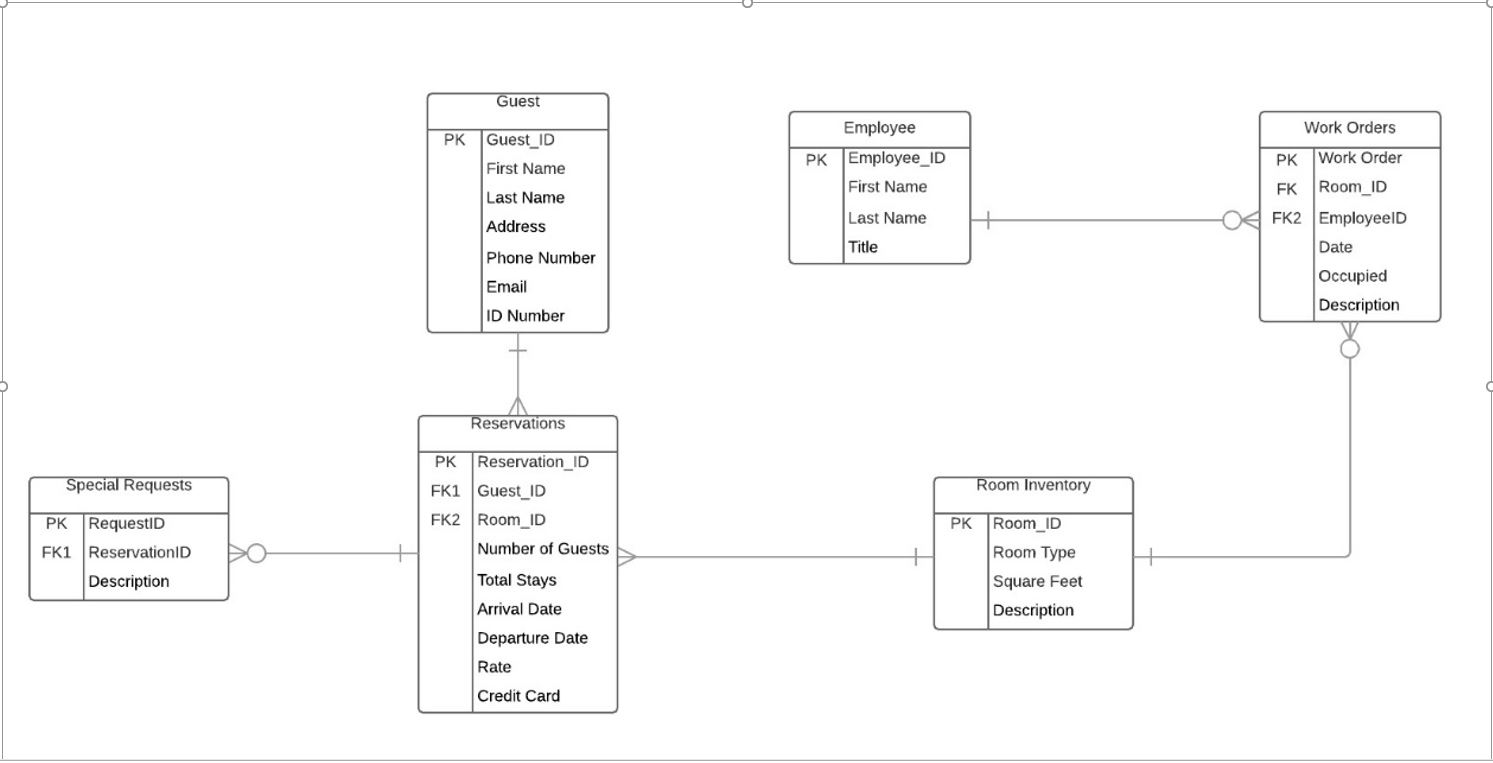
insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(007,004,'I wannt my tv is channel');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(008,003,'I wannt my tv is digital');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(009,002,'I wannt my bed is huge');

insert into GP\_SpecialRequests(Request\_ID,Reservation\_ID,requestDescription) values(010,001,'I wannt my bed is huge');

## ERD Diagram (Data Model and Relations)



## Program Source Code

### Procedures

#### Procedure 1 with Cursor: Find Employee based on job

CREATE OR REPLACE

PROCEDURE EMP\_SP\_findEmp (

p\_findJob VARCHAR2,

p\_emp\_refcur IN OUT SYS\_REFCURSOR

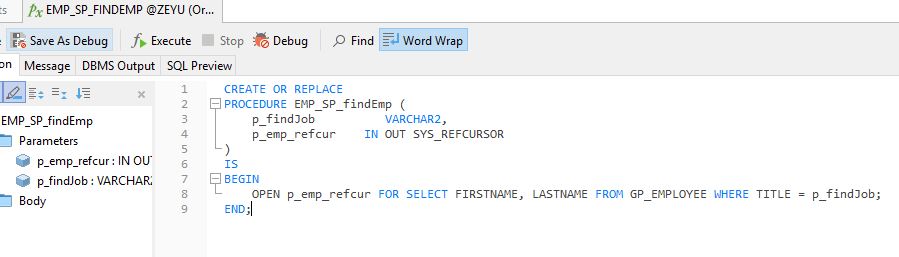
)

IS

BEGIN

OPEN p\_emp\_refcur FOR SELECT FIRSTNAME, LASTNAME FROM GP\_EMPLOYEE WHERE TITLE = p\_findJob;

END;



#### Test Procedure 1 with Cursor

DECLARE

V\_EFIRST GP\_EMPLOYEE.FIRSTNAME % TYPE;

V\_ELAST GP\_EMPLOYEE.LASTNAME % TYPE;

V\_JOB GP\_EMPLOYEE.TITLE % TYPE := 'Calling Center';

V\_EMP\_REFCUR SYS\_REFCURSOR;

BEGIN

DBMS\_OUTPUT.PUT\_LINE ( 'EMPLOYEES WITH JOB ' || V\_JOB );

DBMS\_OUTPUT.PUT\_LINE ( 'FIRSTNAME LASTNAME' );

DBMS\_OUTPUT.PUT\_LINE ( '--------- --------' );

EMP\_SP\_FINDEMP ( V\_JOB, V\_EMP\_REFCUR );

LOOP

FETCH V\_EMP\_REFCUR INTO V\_EFIRST,

V\_ELAST;

EXIT

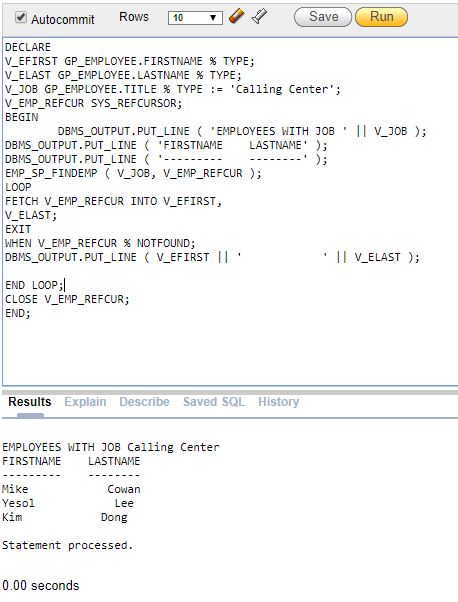
WHEN V\_EMP\_REFCUR % NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE ( V\_EFIRST || ' ' || V\_ELAST );

END LOOP;

CLOSE V\_EMP\_REFCUR;

END;



#### Procedure 2 with Cursor: Display employee list

CREATE OR REPLACE

procedure EMP\_SP\_showEmployee

as

cursor C\_employee is select \* from GP\_EMPLOYEE;

r\_employee C\_employee%ROWTYPE;

begin

open C\_employee;

loop

fetch C\_employee into r\_employee;

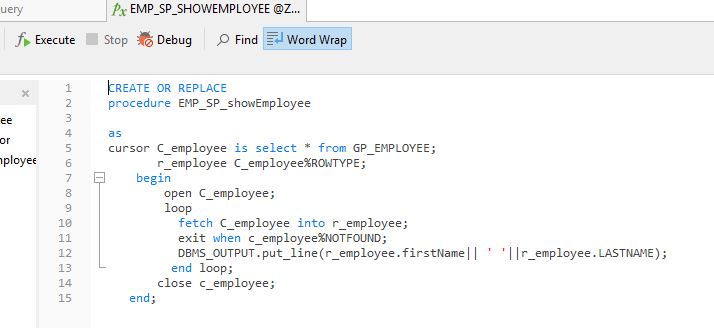
exit when c\_employee%NOTFOUND;

DBMS\_OUTPUT.put\_line(r\_employee.firstName|| ' '||r\_employee.LASTNAME);

end loop;

close c\_employee;

end;

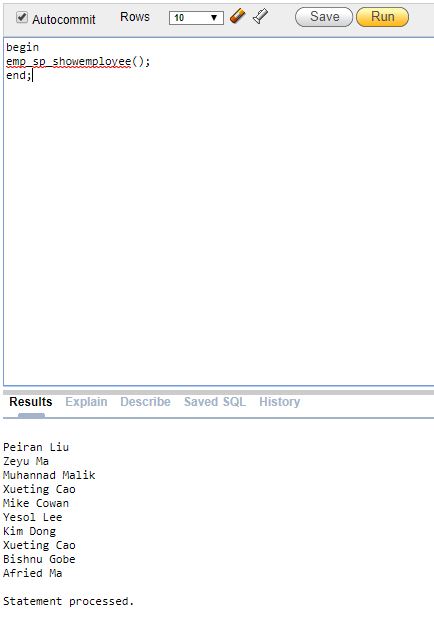


#### Test Procedure 2 with Cursor

begin

emp\_sp\_showemployee();

end;



#### Procedure 3 with Cursor: Find room info and payment method based on room size

CREATE OR REPLACE

procedure pro\_select

(r\_sq in number)

IS

CURSOR cur\_select is

select GP\_WORKORDERS.WORKORDER,GP\_WORKORDERS.ORDERDESCRIPTION,GP\_ROOMINVENTORY.ROOMTYPE,GP\_ROOMINVENTORY.ROOMDESCRIPTION from GP\_WORKORDERS inner join GP\_ROOMINVENTORY using (ROOM\_ID) where GP\_ROOMINVENTORY.SQUAREFEET = r\_sq;

TYPE type\_select is RECORD(

workorder GP\_WORKORDERS.WORKORDER%type,

orderdescription GP\_WORKORDERS.ORDERDESCRIPTION%type,

roomtype GP\_ROOMINVENTORY.ROOMTYPE%type,

roomdescription GP\_ROOMINVENTORY.ROOMDESCRIPTION%type);

rec\_select type\_select;

begin

open cur\_select;

loop

fetch cur\_select into rec\_select;

EXIT when cur\_select%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(rec\_select.workorder|| ' '||rec\_select.ORDERDESCRIPTION||' '||rec\_select.ROOMTYPE||' '||rec\_select.ROOMDESCRIPTION);

end loop;

close cur\_select;

end;



#### Test Procedure 3 with Cursor

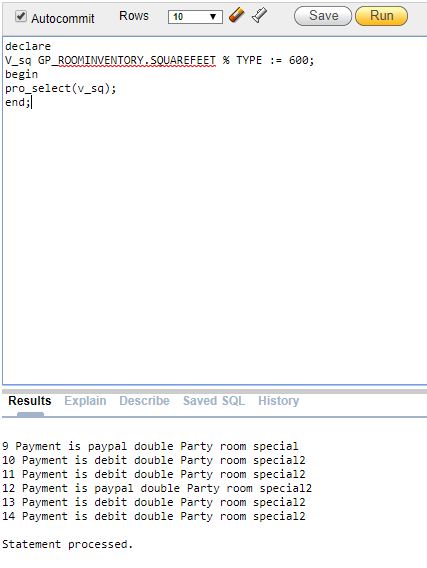
declare

V\_sq GP\_ROOMINVENTORY.SQUAREFEET % TYPE := 600;

begin

pro\_select(v\_sq);

end;



#### Procedure 4: Check room level

CREATE OR REPLACE

procedure check\_room\_level

(r\_cap in number)

as

lv\_id GP\_ROOMINVENTORY.room\_id%type;

lv\_squ GP\_ROOMINVENTORY.SQUAREFEET%type;

begin

select room\_id,squarefeet into lv\_id,lv\_squ from GP\_ROOMINVENTORY where ROOM\_ID = r\_cap;

DBMS\_OUTPUT.PUT\_LINE(lv\_id||' '||lv\_squ);

if lv\_squ>500 then

DBMS\_OUTPUT.PUT\_LINE('THIS is a hugn room');

elsif lv\_squ>400 then

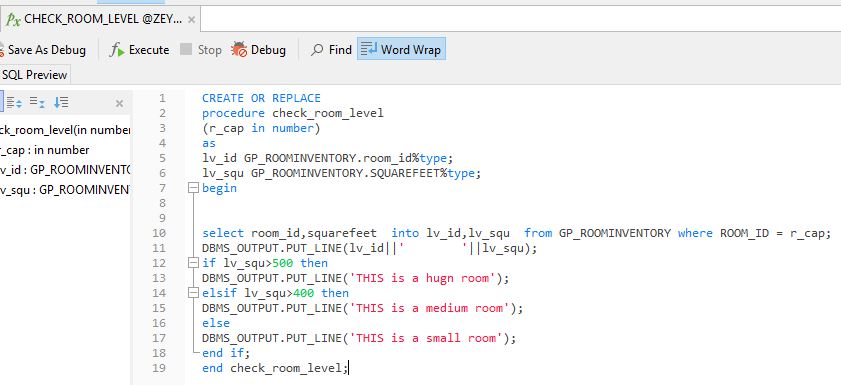
DBMS\_OUTPUT.PUT\_LINE('THIS is a medium room');

else

DBMS\_OUTPUT.PUT\_LINE('THIS is a small room');

end if;

end check\_room\_level;

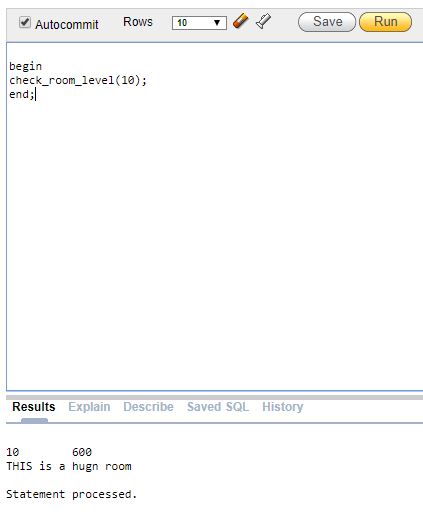


#### Test Procedure 4

begin

check\_room\_level(10);

end;



#### Procedure 5 & 6: Find request and guest by room

CREATE OR REPLACE

PROCEDURE find\_request\_sp

(p\_reid in number, p\_rqds out VARCHAR2)

IS

BEGIN

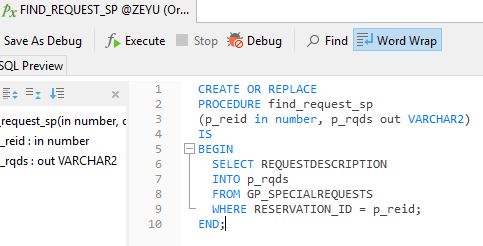
SELECT REQUESTDESCRIPTION

INTO p\_rqds

FROM GP\_SPECIALREQUESTS

WHERE RESERVATION\_ID = p\_reid;

END;



CREATE OR REPLACE

PROCEDURE find\_guestnum\_sp

(p\_roid in GP\_RESERVATION.ROOM\_ID%TYPE,

p\_numg out number,

p\_jrqds out VARCHAR2)

IS

BEGIN

SELECT GP\_RESERVATION.NUMBERGUEST

INTO p\_numg

FROM GP\_RESERVATION

WHERE RESERVATION\_ID = p\_roid;

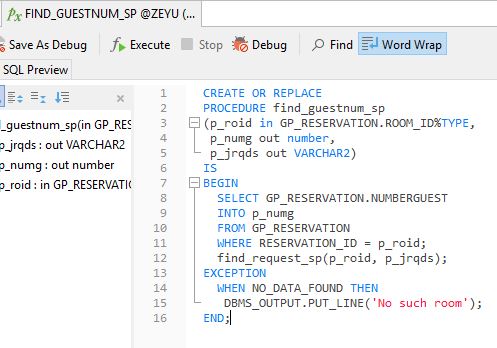
find\_request\_sp(p\_roid, p\_jrqds);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No such room');

END;



#### Test Procedure 5 & 6

DECLARE

lv\_roomid GP\_RESERVATION.ROOM\_ID%TYPE := 3;

lv\_guest\_num NUMBER(10);

lv\_request\_des VARCHAR2(50);

BEGIN

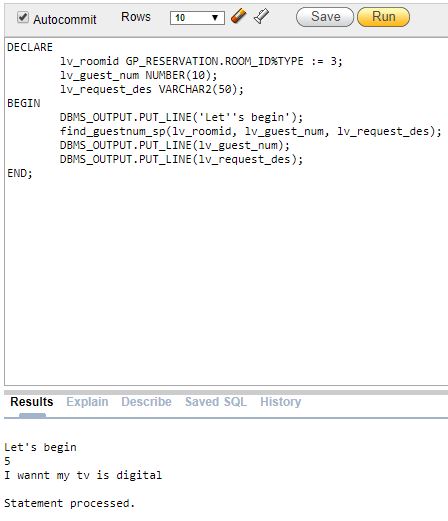
DBMS\_OUTPUT.PUT\_LINE('Let''s begin');

find\_guestnum\_sp(lv\_roomid, lv\_guest\_num, lv\_request\_des);

DBMS\_OUTPUT.PUT\_LINE(lv\_guest\_num);

DBMS\_OUTPUT.PUT\_LINE(lv\_request\_des);

END;



#### Trigger 1: Save room history before modifying

CREATE TABLE GP\_ROOM\_HISTORY

( ROOM\_ID NUMBER(25,0) NOT NULL ENABLE,

SQUAREFEET NUMBER(25,0) NOT NULL ENABLE,

ROOMTYPE VARCHAR2(25) NOT NULL ENABLE,

ROOMDESCRIPTION VARCHAR2(25) NOT NULL ENABLE

) ;

CREATE OR REPLACE TRIGGER GP\_ROOMINVENTORY\_TRG1

BEFORE

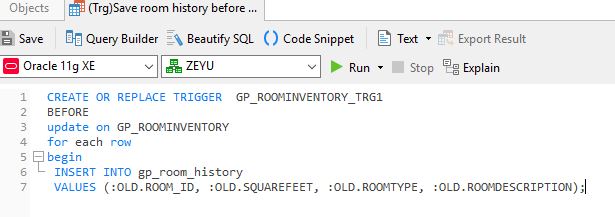
update on GP\_ROOMINVENTORY

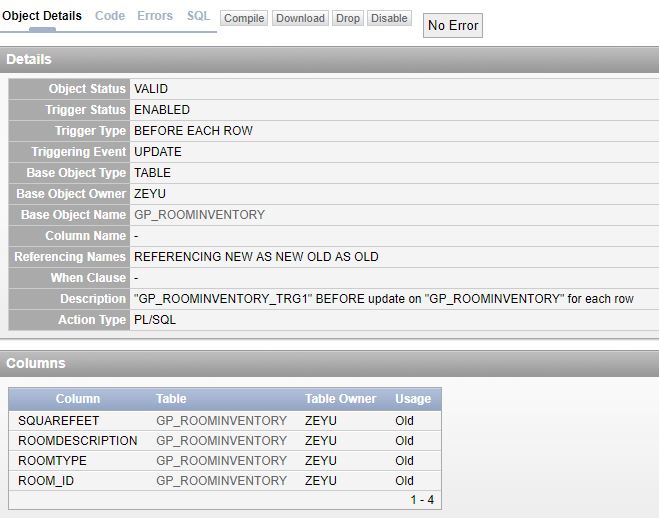
for each row

begin

INSERT INTO gp\_room\_history

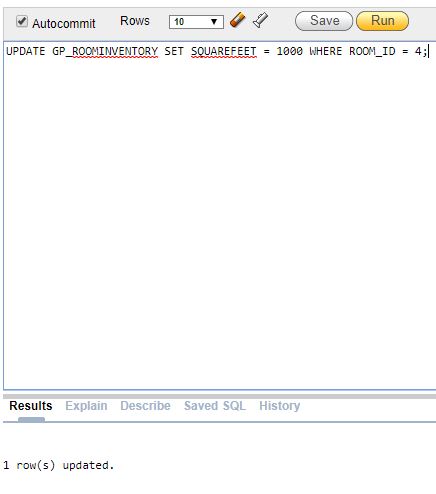
VALUES (:OLD.ROOM\_ID, :OLD.SQUAREFEET, :OLD.ROOMTYPE, :OLD.ROOMDESCRIPTION);

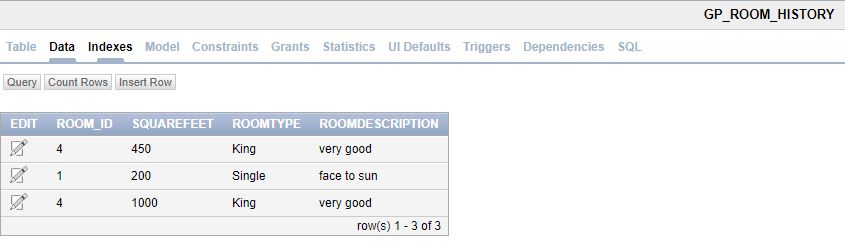




#### Test Trigger 1

UPDATE GP\_ROOMINVENTORY SET SQUAREFEET = 1000 WHERE ROOM\_ID = 4;





#### Trigger 2: Update payment method after Guest change id number

CREATE

OR REPLACE TRIGGER gp\_workorder\_trg

AFTER UPDATE OF IDNUMBER ON gp\_GUEST FOR EACH ROW

DECLARE

CURSOR gp\_reservation\_cur IS SELECT

room\_id,

numberguest,

totalstays

FROM

gp\_reservation ;

lv\_od\_des VARCHAR2 ( 50 );

BEGIN

FOR gp\_reservation\_rec IN gp\_reservation\_cur

LOOP

IF

gp\_reservation\_rec.TOTALSTAYS < 7 THEN

lv\_od\_des := ( 'Payment is master' );

ELSIF

gp\_reservation\_rec.TOTALSTAYS >= 7 AND gp\_reservation\_rec.TOTALSTAYS < 9 THEN

lv\_od\_des := ( 'Payment is debit' );

ELSIF

gp\_reservation\_rec.TOTALSTAYS >= 9 AND gp\_reservation\_rec.TOTALSTAYS < 12 THEN

lv\_od\_des := ( 'Payment is paypal' );

ELSE lv\_od\_des := ( 'Payment is credit' );

END IF;

UPDATE gp\_workorders

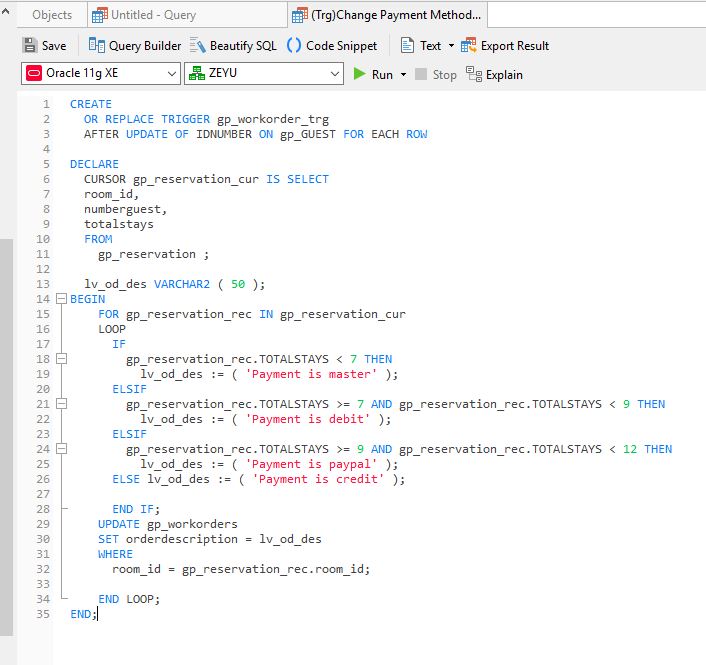
SET orderdescription = lv\_od\_des

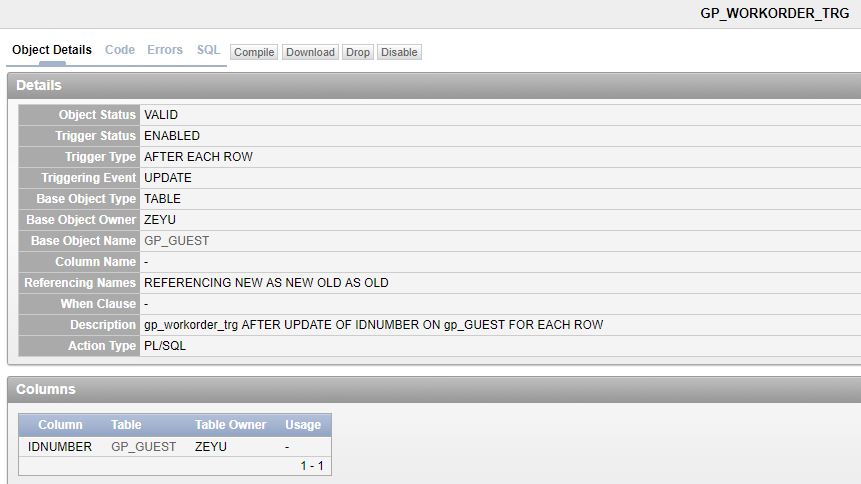
WHERE

room\_id = gp\_reservation\_rec.room\_id;

END LOOP;

END;





#### Test Trigger 2

UPDATE GP\_GUEST

SET IDNUMBER = 8888

WHERE GUEST\_ID = 2;

