|  |
| --- |
| Close-up image showing the leaf-sides of two oversized books side-by-side on a bookshelf, with additional books in soft focus background |
| PLSQL Project  Advanced Databases |
| |  |  |  | | --- | --- | --- | | Reonne Benoy-A00268832 | 5/14/20 | Athlone Institute of Technology | |

Contents

[Question 1 1](#_Toc40381076)

[Question 2 4](#_Toc40381077)

[Question 3 7](#_Toc40381078)

[Question 4 10](#_Toc40381079)

[Stage 1 10](#_Toc40381080)

[Stage 2 13](#_Toc40381081)

[Stage 3 17](#_Toc40381082)

PLSQL Project

# Question 1

Clear Screen

Set Serveroutput on Size 100000

Drop Table Concert\_Capacity\_Tracker;

Create Table Concert\_Capacity\_Tracker(Concert\_Id Number(3), Group\_Name Varchar2(15), Increase Number(3), Description Varchar2(50), User\_Name Varchar2(10), Exec\_Date Date);

Create or Replace Procedure Raise\_Concert\_Capacity(Conc\_Id In Integer, Increase In Integer) Is

V\_Group\_Name Concert.Group\_Name%Type;

V\_Category Concert.Category%Type;

V\_Capacity Concert.Capacity%Type;

V\_Actual\_Attendance Concert.Actual\_Attendance%Type;

V\_Ticket\_Price Concert.Ticket\_Price%Type;

T\_Price\_Missing Exception;

Illegal\_Category Exception;

Increase\_Too\_Low Exception;

/\*Reonne Benoy-A00268832\*/

Begin

Dbms\_Output.Put\_Line('View Data Before Update');

Select Group\_Name, Category, Capacity, Actual\_Attendance, Ticket\_Price Into V\_Group\_Name, V\_Category, V\_Capacity, V\_Actual\_Attendance, V\_Ticket\_Price

From Concert

Where Concert.Concert\_Id = Conc\_Id;

Dbms\_Output.Put\_Line('Group\_Name:'|| V\_Group\_Name||'Original Capacity:'||V\_Capacity);

If V\_Ticket\_Price Is Null Then

Raise T\_Price\_Missing;

Elsif Increase<=500 Then

Raise Increase\_Too\_Low;

Elsif V\_Category Not In ('ROCK','CLASSICAL') Then

Raise Illegal\_Category;

Else

Update Concert

Set Capacity = Capacity + Increase

Where Concert.Concert\_Id = Conc\_Id;

End If;

Dbms\_Output.Put\_Line('Output After Update');

Select Group\_Name,Capacity Into V\_Group\_Name, V\_Capacity

From Concert

Where Concert.Concert\_Id = Conc\_Id;

Dbms\_Output.Put\_Line('Group Name:' || V\_Group\_Name|| 'Updated Capacity:'||V\_Capacity);

Exception

When T\_Price\_Missing Then

Insert Into Concert\_Capacity\_Tracker Values(Conc\_Id,V\_Group\_Name,Increase,'Ticker Price is Missing',User,Sysdate);

When Illegal\_Category Then

Insert Into Concert\_Capacity\_Tracker Values(Conc\_Id,V\_Group\_Name,Increase,'No Concert Category as such', User, Sysdate);

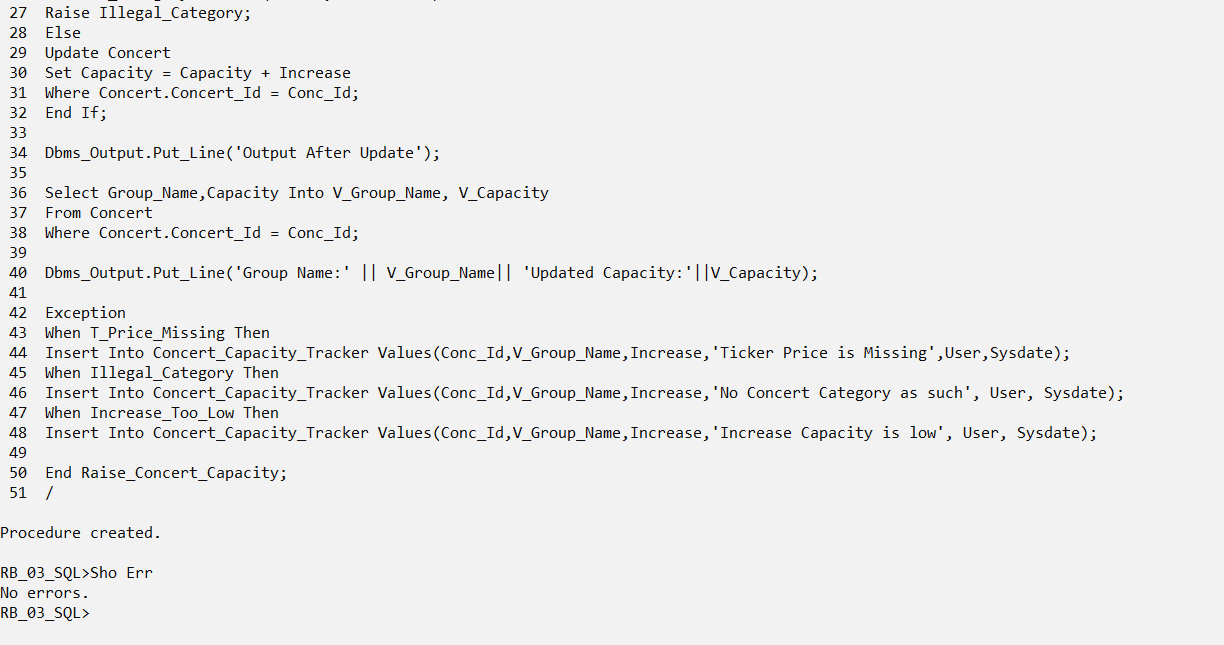
When Increase\_Too\_Low Then

Insert Into Concert\_Capacity\_Tracker Values(Conc\_Id,V\_Group\_Name,Increase,'Increase Capacity is low', User, Sysdate);

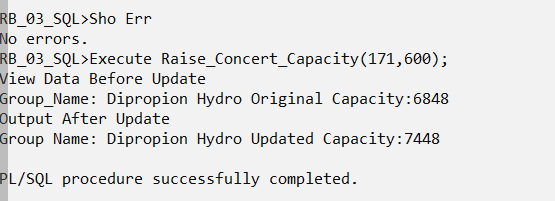
End Raise\_Concert\_Capacity;

/

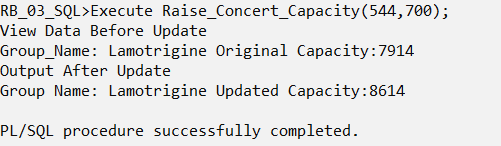
Sho Err



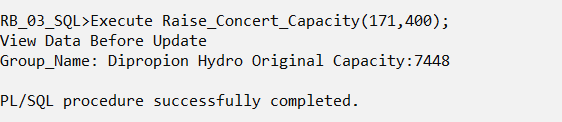
Execute Raise\_Concert\_Capacity(171,600);

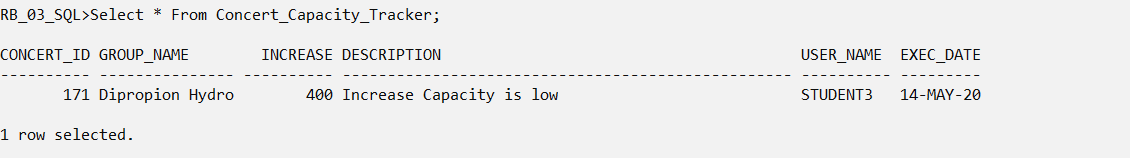


Execute Raise\_Concert\_Capacity(544,700);



Execute Raise\_Concert\_Capacity(171,400);





Insert Into Concert(Concert\_Id,Group\_Name,Category,Ticket\_Price) Values (777,'Quary','Rock',Null);

Execute Raise\_Concert\_Capacity(777,600);

Insert Into Concert(Concert\_Id,Group\_Name,Category,Ticket\_Price) Values (999,'Bond','Athem',40.05);

Execute Raise\_Concert\_Capacity(999,700);

Rollback;

# Question 2

Clear Screen

Set Serveroutput on Size 1000000

Set Linesize 180

Drop Table DML\_Journalist\_log;

Create Table DML\_Journalist\_log(Username Varchar2(15), Exec\_Date Date, Rows\_Changed Number(3));

Create Or Replace Procedure Modify\_LowEarnings\_Journ As

Cursor Journalist\_Cur Is Select Journalist\_Id, Forename, Surname, Annual\_Earnings

From Journalist

Order By Annual\_Earnings

For Update Of Annual\_Earnings;

V\_Journalist\_Rec Journalist\_Cur%Rowtype;

V\_Total\_Earnings\_Bottom5 Number(9,2):=0;

V\_Num\_Journalist\_Changed Number(3):=0;

/\*Reonne Benoy-A00268832\*/

Begin

Open Journalist\_Cur;

Dbms\_Output.Put\_Line('List of 5 Journalists with the lowest annual earnings');

For V\_Count In 1..5

Loop

Fetch Journalist\_Cur Into V\_Journalist\_Rec;

Exit When Journalist\_Cur%Notfound;

Dbms\_Output.Put\_Line('Journalist ID: '||V\_Journalist\_Rec.Journalist\_Id);

Dbms\_Output.Put\_Line('Name: '|| V\_Journalist\_Rec.Forename|| ' '|| V\_Journalist\_Rec.Surname);

Dbms\_Output.Put\_Line('Annual\_Earnings: '|| V\_Journalist\_Rec.Annual\_Earnings);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

V\_Total\_Earnings\_Bottom5:=V\_Total\_Earnings\_Bottom5 + V\_Journalist\_Rec.Annual\_Earnings;

Update Journalist

Set Annual\_Earnings = Annual\_Earnings + 2000

Where Current Of Journalist\_Cur;

V\_Num\_Journalist\_Changed := V\_Num\_Journalist\_Changed + 1;

End Loop;

Close Journalist\_Cur;

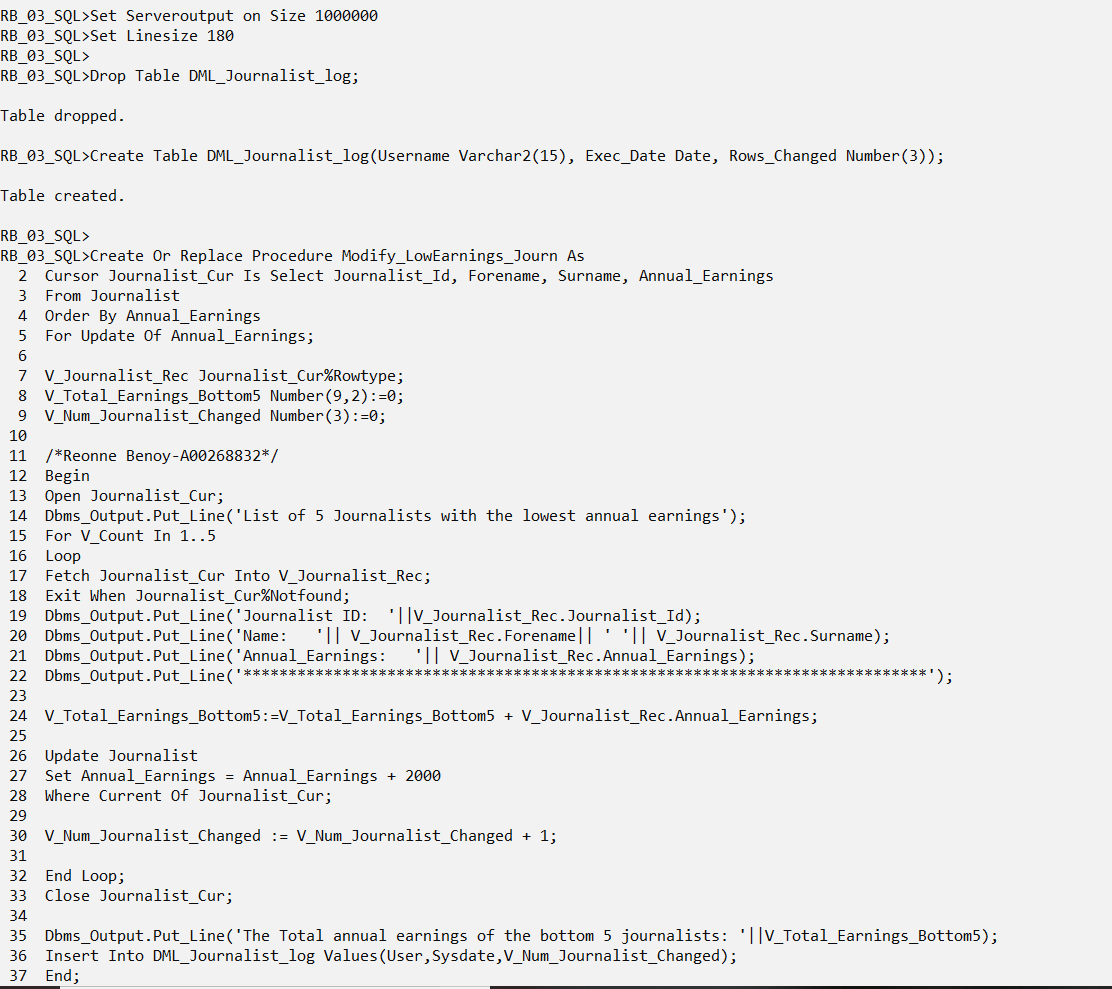
Dbms\_Output.Put\_Line('The Total annual earnings of the bottom 5 journalists: '||V\_Total\_Earnings\_Bottom5);

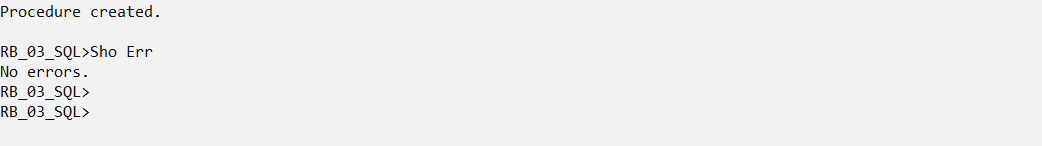
Insert Into DML\_Journalist\_log Values(User,Sysdate,V\_Num\_Journalist\_Changed);

End;

/

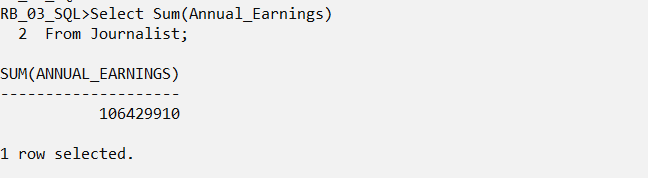
Sho Err



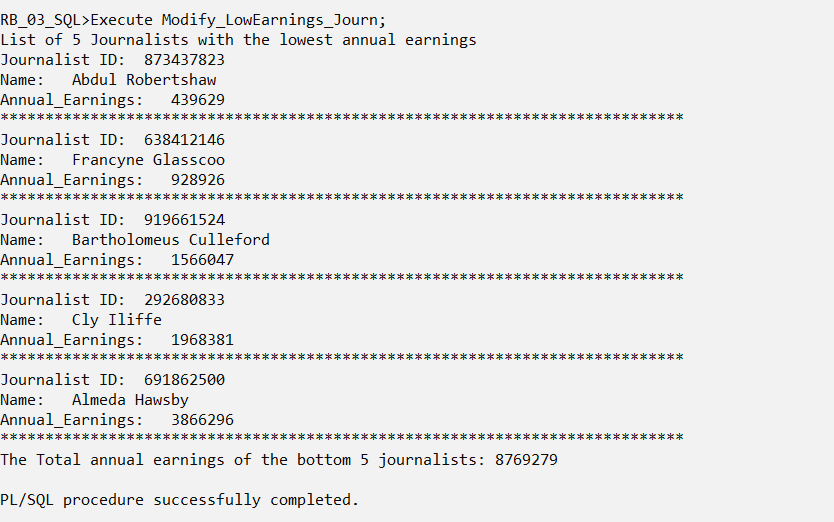


Select Sum(Annual\_Earnings)

From Journalist;

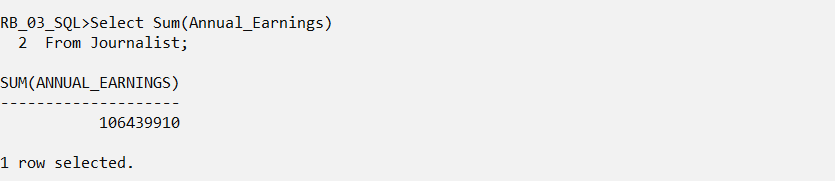


Execute Modify\_LowEarnings\_Journ;

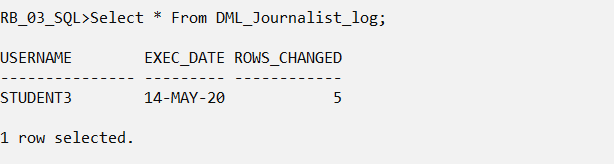


Select Sum(Annual\_Earnings)

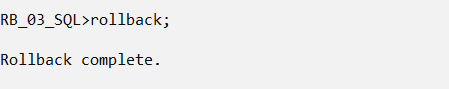
From Journalist;



Select \* From DML\_Journalist\_log;



Rollback;



# Question 3

Clear Screen

Set Serveroutput On Size 1000000

Drop Table Deleted\_Newspaper\_Log;

Create Table Deleted\_Newspaper\_Log(NewspaperID Number(8), Username Varchar2(15),Exec\_Date Date);

Create Or Replace Procedure Remove\_Newspaper As

Cursor Newspaper\_Cur Is Select Newspaper\_Id, Name, Annual\_Turnover

From Newspaper

Order By Name

For Update;

V\_Newspaper\_Rec Newspaper\_Cur%Rowtype;

V\_Count\_Journalists Number(2):=0;

V\_Count\_Turnover\_Increase Number(2):=0;

V\_Count\_Deleted\_Newspapers Number(2):=0;

/\*Reonne Benoy-A00268832\*/

Begin

Dbms\_Output.Put\_Line('List of all the Newspaper companies ordered by their names');

Open Newspaper\_Cur;

Loop

Fetch Newspaper\_Cur Into V\_Newspaper\_Rec;

Exit When Newspaper\_Cur%Notfound;

Select NVL(Count(\*),0) Into V\_Count\_Journalists

From Journalist, Newspaper

Where Journalist.Newspaper\_Id = Newspaper.Newspaper\_Id;

Dbms\_Output.Put\_Line('Name: '|| V\_Newspaper\_Rec.Name|| 'Annual Turnover:' || V\_Newspaper\_Rec.Annual\_Turnover);

Dbms\_Output.Put\_Line('Number of Journalists employed:' || V\_Count\_Journalists);

If V\_Count\_Journalists = 0 Then

Dbms\_Output.Put\_Line('Newspaper Comapany '|| V\_Newspaper\_Rec.Name||' will be deleted');

Delete

From Newspaper

Where Current Of Newspaper\_Cur;

V\_Count\_Deleted\_Newspapers:= V\_Count\_Deleted\_Newspapers + 1;

Else

Dbms\_Output.Put\_Line('Newspaper Company '|| V\_Newspaper\_Rec.Name||' will get a Turnover increase');

Update Newspaper

Set Annual\_Turnover = Annual\_Turnover + 15000

Where Current Of Newspaper\_Cur;

V\_Count\_Turnover\_Increase:=V\_Count\_Turnover\_Increase + 1;

End If;

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Insert Into Deleted\_Newspaper\_Log Values(V\_Newspaper\_Rec.Newspaper\_Id,User,Sysdate);

End Loop;

Close Newspaper\_Cur;

Dbms\_Output.Put\_Line('Number of Newspapers that have been removed: '|| V\_Count\_Deleted\_Newspapers);

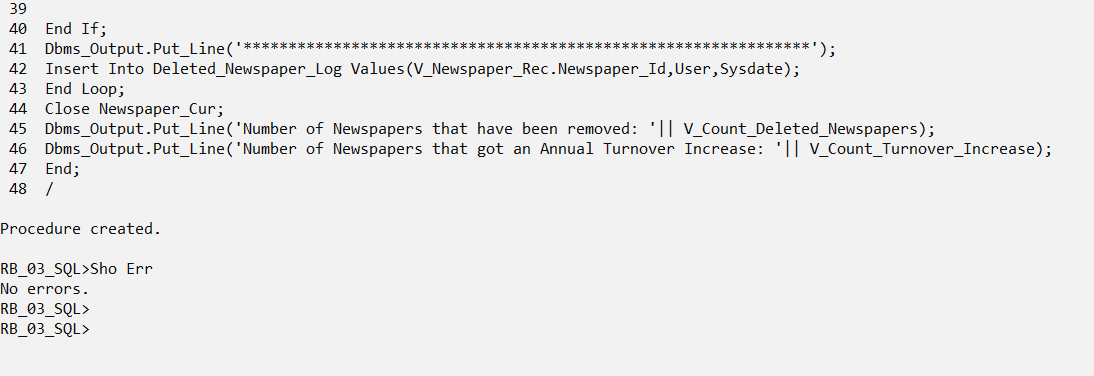
Dbms\_Output.Put\_Line('Number of Newspapers that got an Annual Turnover Increase: '|| V\_Count\_Turnover\_Increase);

End;

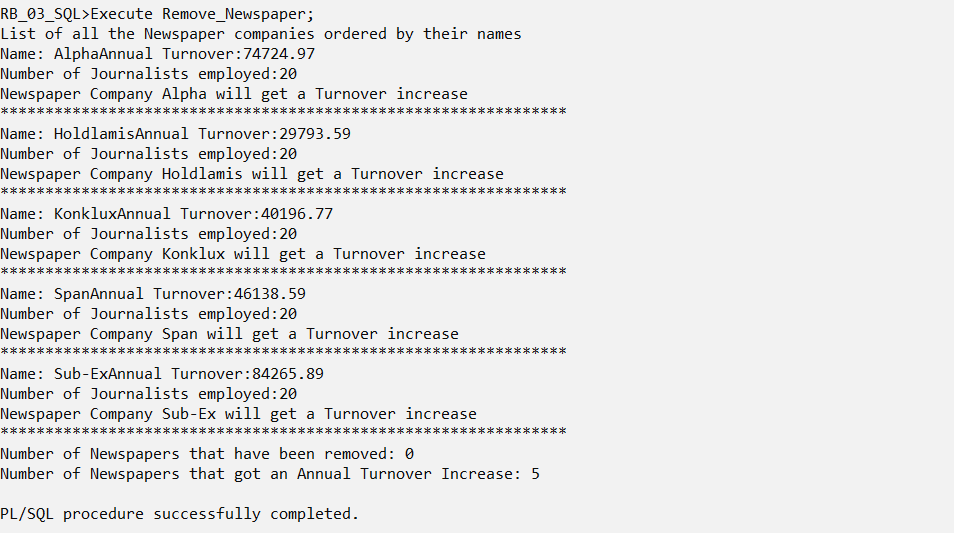
/

Sho Err

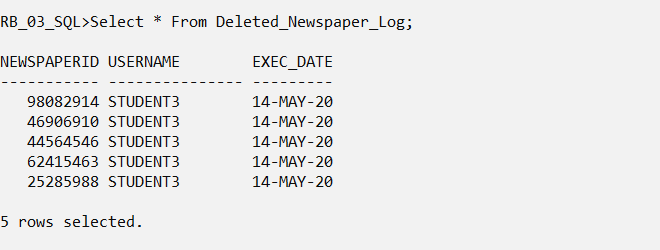




Execute Remove\_Newspaper;



Select \* From Deleted\_Newspaper\_Log;



# Question 4

## Stage 1

Clear Screen

Set Serveroutput On Size 1000000

Create Or Replace Procedure Venue\_Manage(VenueId In Number, Cap\_Prop\_Increase In Number) Is

V\_Name Venue.Name%Type;

V\_City Venue.City%Type;

V\_Category Venue.Category%Type;

V\_Cost Venue.Cost%Type;

V\_Capacity Venue.Capacity%Type;

V\_Actual\_Inc Number(4);

V\_Max\_Cap Venue.Capacity%Type;

V\_Min\_Cap Venue.Capacity%Type;

V\_Avg\_Cap Venue.Capacity%Type;

V\_Max\_Cap\_Cat Venue.Capacity%Type;

V\_Min\_Cap\_Cat Venue.Capacity%Type;

V\_Avg\_Cap\_Cat Venue.Capacity%Type;

/\*Reonne Benoy-A00268832\*/

Begin

Dbms\_Output.Put\_Line('Display Venue details');

Select Name,City,Category,Cost,Capacity Into V\_Name,V\_City,V\_Category,V\_Cost,V\_Capacity

From Venue

Where Venue.Venue\_Id = VenueId;

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: ' || V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Overall Capacity');

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap,V\_Min\_Cap, V\_Avg\_Cap

From Venue;

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Capacity');

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap\_Cat, V\_Min\_Cap\_Cat, V\_Avg\_Cap\_Cat

From Venue

Where Venue.Category = V\_Category;

Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

If V\_Capacity>V\_Avg\_Cap And V\_Capacity<V\_Max\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase;

Elsif V\_Capacity<V\_Avg\_Cap And V\_Capacity>V\_Min\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase\*1.50;

Else

V\_Actual\_Inc:= Cap\_Prop\_Increase\*0.90;

End If;

Dbms\_Output.Put\_Line('Actual Increase of Capacity: ' || V\_Actual\_Inc);

Update Venue

Set Capacity=Capacity + V\_Actual\_Inc

Where Venue.Venue\_Id = VenueId;

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: '|| V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('Overall Capacity After Increase');

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap,V\_Min\_Cap, V\_Avg\_Cap

From Venue;

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Overall Capacity by Categories After Increase');

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap\_Cat, V\_Min\_Cap\_Cat, V\_Avg\_Cap\_Cat

From Venue

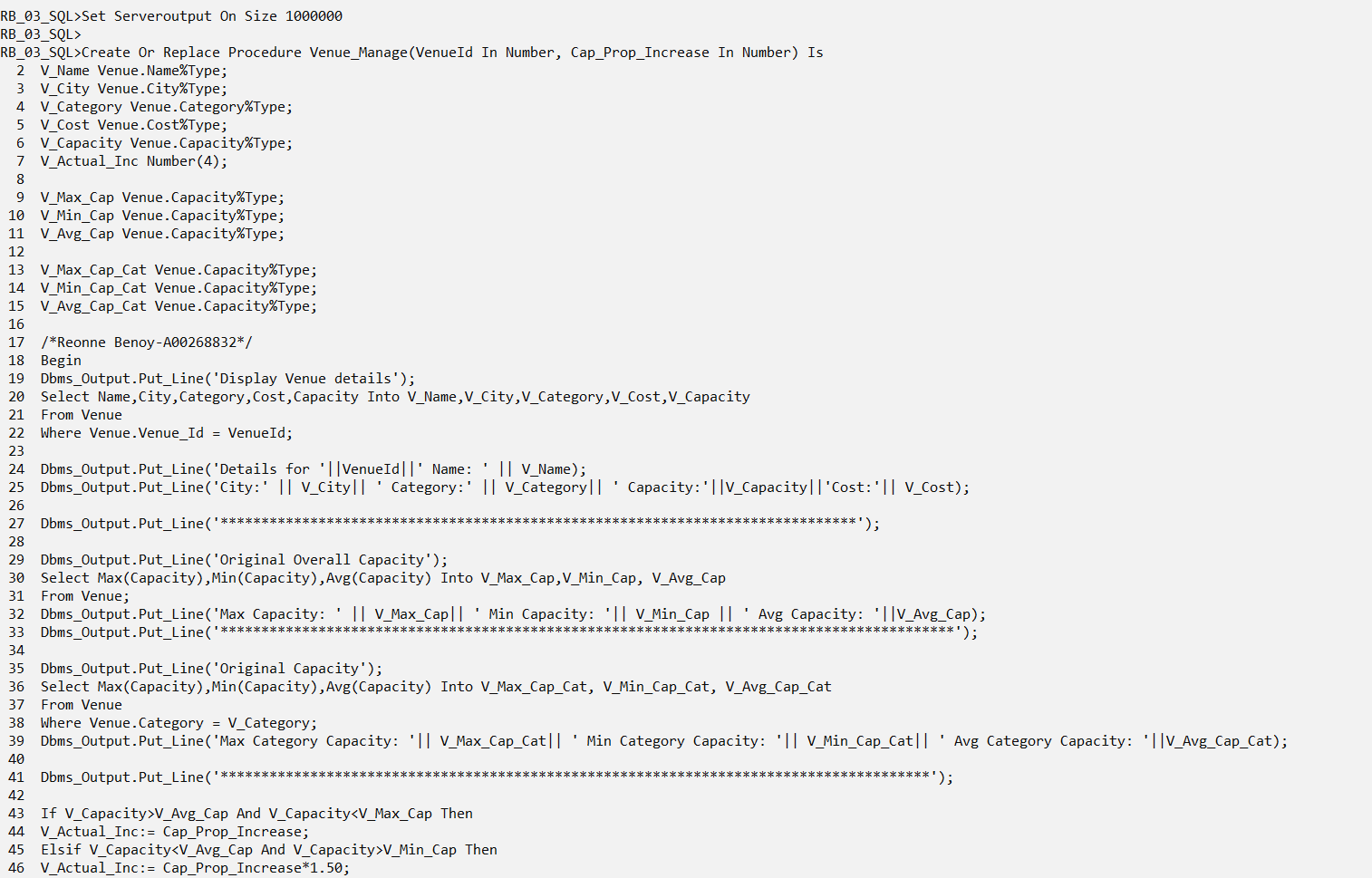
Where Venue.Category = V\_Category;

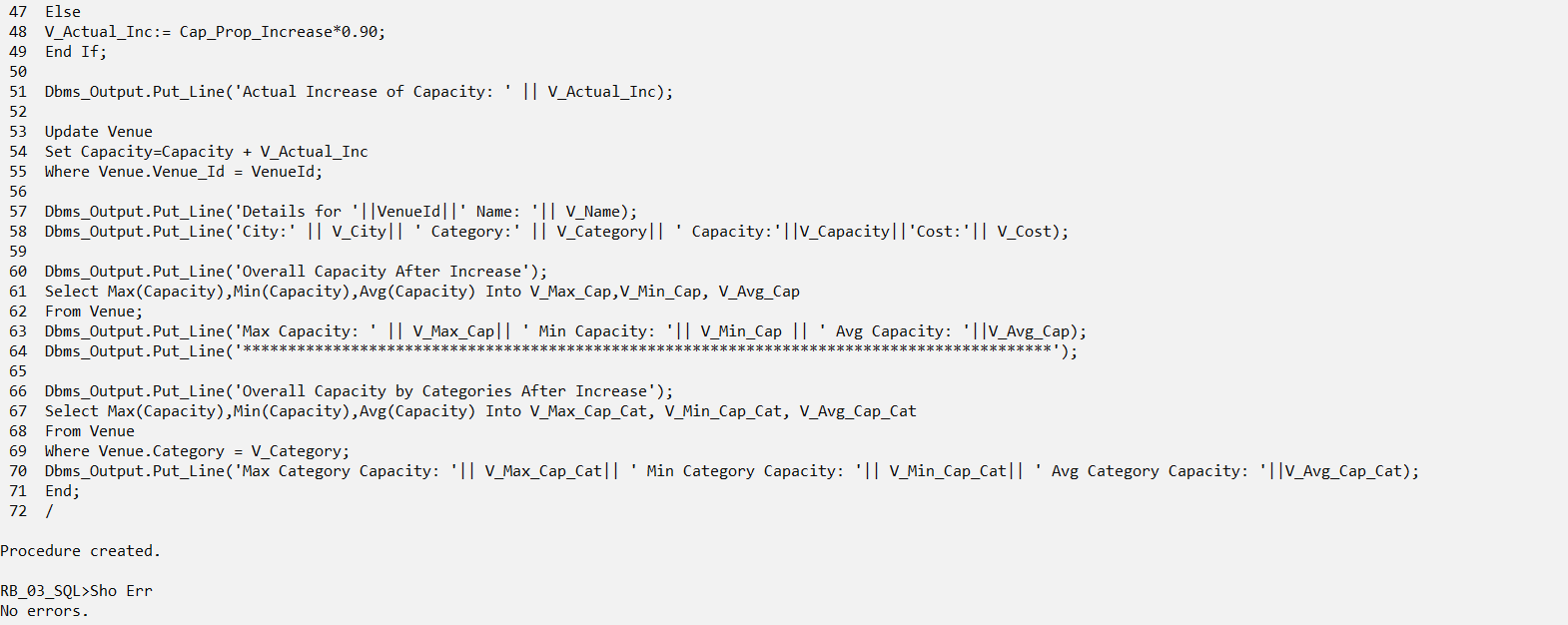
Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

End;

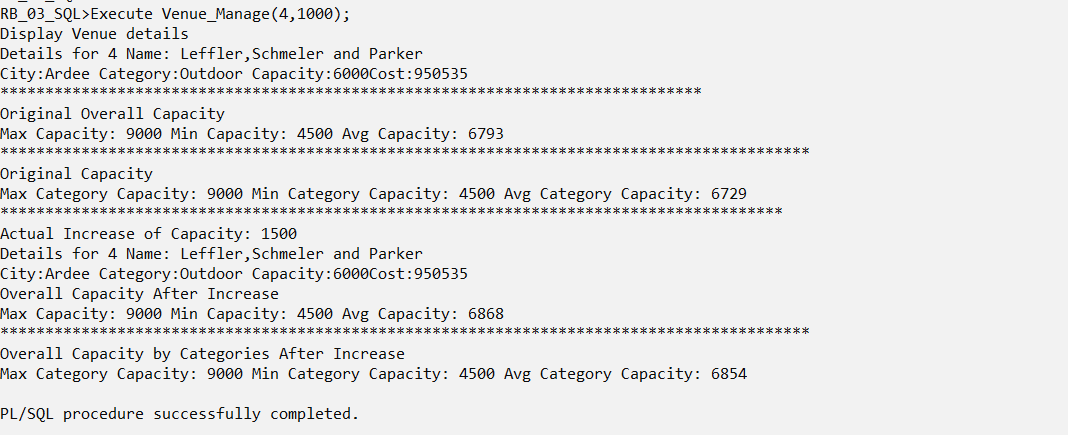
/

Sho Err





Execute Venue\_Manage(4,1000);



Rollback;

## Stage 2

**Stored Procedures and Functions**

Clear Screen

Set Serveroutput On Size 1000000

Create Or Replace Procedure Get\_Venue\_Details(VenueId In Number, V\_Name Out Varchar, V\_City Out Varchar, V\_Category Out Varchar, V\_Capacity Out Number, V\_Cost Out Number) As

Begin

Select Name,City,Category,Cost,Capacity Into V\_Name,V\_City,V\_Category,V\_Cost,V\_Capacity

From Venue

Where Venue.Venue\_Id = VenueId;

End Get\_Venue\_Details;

/

Sho Err

Create Or Replace Procedure Calc\_Overall\_Capacity(V\_Max\_Cap Out Number, V\_Min\_Cap Out Number, V\_Avg\_Cap Out Number) As

Begin

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap,V\_Min\_Cap, V\_Avg\_Cap

From Venue;

End Calc\_Overall\_Capacity;

/

Sho Err

Create Or Replace Procedure Calc\_Category\_Capacity(V\_Category In Varchar, V\_Max\_Cap\_Cat Out Number, V\_Min\_Cap\_Cat Out Number, V\_Avg\_Cap\_Cat Out Number) As

Begin

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap\_Cat, V\_Min\_Cap\_Cat, V\_Avg\_Cap\_Cat

From Venue

Where Venue.Category = V\_Category;

End Calc\_Category\_Capacity;

/

Sho Err

Create Or Replace Procedure Modify\_Capacity(VenueId In Number, V\_Actual\_Inc In Number) As

Begin

Update Venue

Set Capacity=Capacity + V\_Actual\_Inc

Where Venue.Venue\_Id = VenueId;

End Modify\_Capacity;

/

Sho Err

Create Or Replace Function Updated\_Capacity(VenueId In NUMBER) Return Number Is

V\_Updated\_Capacity Venue.Capacity%Type;

Begin

Select Capacity Into V\_Updated\_Capacity

From Venue

Where Venue.Venue\_Id = VenueId;

Return V\_Updated\_Capacity;

End Updated\_Capacity;

/

Sho Err

Create Or Replace Procedure Venue\_Manage(VenueId In Number, Cap\_Prop\_Increase In Number) Is

V\_Name Venue.Name%Type;

V\_City Venue.City%Type;

V\_Category Venue.Category%Type;

V\_Cost Venue.Cost%Type;

V\_Capacity Venue.Capacity%Type;

V\_Actual\_Inc Number(4);

V\_New\_Capacity Venue.Capacity%Type;

V\_Max\_Cap Venue.Capacity%Type;

V\_Min\_Cap Venue.Capacity%Type;

V\_Avg\_Cap Venue.Capacity%Type;

V\_Max\_Cap\_Cat Venue.Capacity%Type;

V\_Min\_Cap\_Cat Venue.Capacity%Type;

V\_Avg\_Cap\_Cat Venue.Capacity%Type;

/\*Reonne Benoy-A00268832\*/

Begin

Dbms\_Output.Put\_Line('Display Venue details');

Get\_Venue\_Details(VenueId, V\_Name, V\_City, V\_Category, V\_Capacity, V\_Cost);

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: ' || V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Overall Capacity');

Calc\_Overall\_Capacity(V\_Max\_Cap, V\_Min\_Cap, V\_Avg\_Cap);

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Capacity');

Calc\_Category\_Capacity(V\_Category,V\_Max\_Cap\_Cat,V\_Min\_Cap\_Cat,V\_Avg\_Cap\_Cat);

Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

If V\_Capacity>V\_Avg\_Cap And V\_Capacity<V\_Max\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase;

Elsif V\_Capacity<V\_Avg\_Cap And V\_Capacity>V\_Min\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase\*1.50;

Else

V\_Actual\_Inc:= Cap\_Prop\_Increase\*0.90;

End If;

Dbms\_Output.Put\_Line('Actual Increase of Capacity: ' || V\_Actual\_Inc);

Modify\_Capacity(VenueId, V\_Actual\_Inc);

V\_New\_Capacity:= Updated\_Capacity(VenueId);

Dbms\_Output.Put\_Line('Updated Capacity: ' || V\_New\_Capacity);

Get\_Venue\_Details(VenueId, V\_Name, V\_City, V\_Category, V\_Capacity, V\_Cost);

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: '|| V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('Overall Capacity After Increase');

Calc\_Overall\_Capacity(V\_Max\_Cap, V\_Min\_Cap, V\_Avg\_Cap);

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Overall Capacity by Categories After Increase');

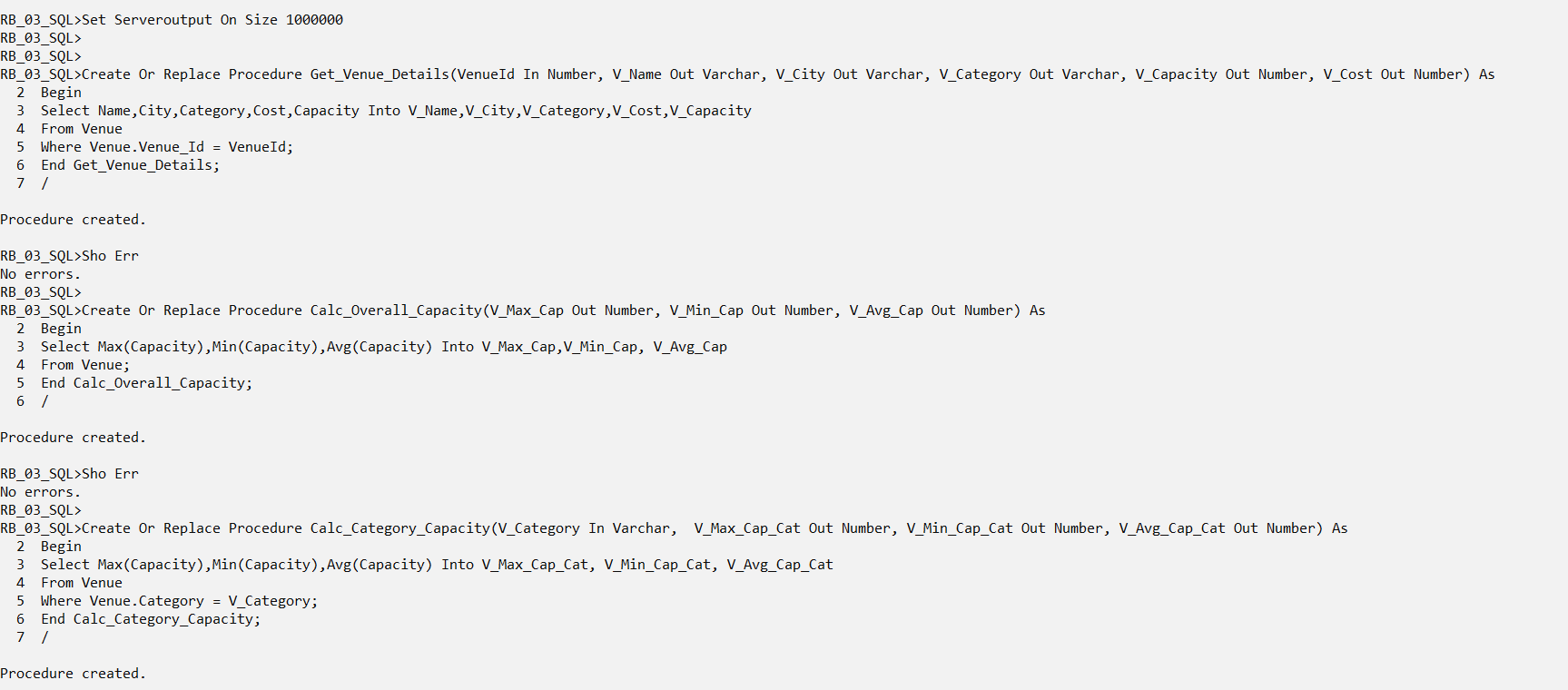
Calc\_Category\_Capacity(V\_Category,V\_Max\_Cap\_Cat,V\_Min\_Cap\_Cat,V\_Avg\_Cap\_Cat);

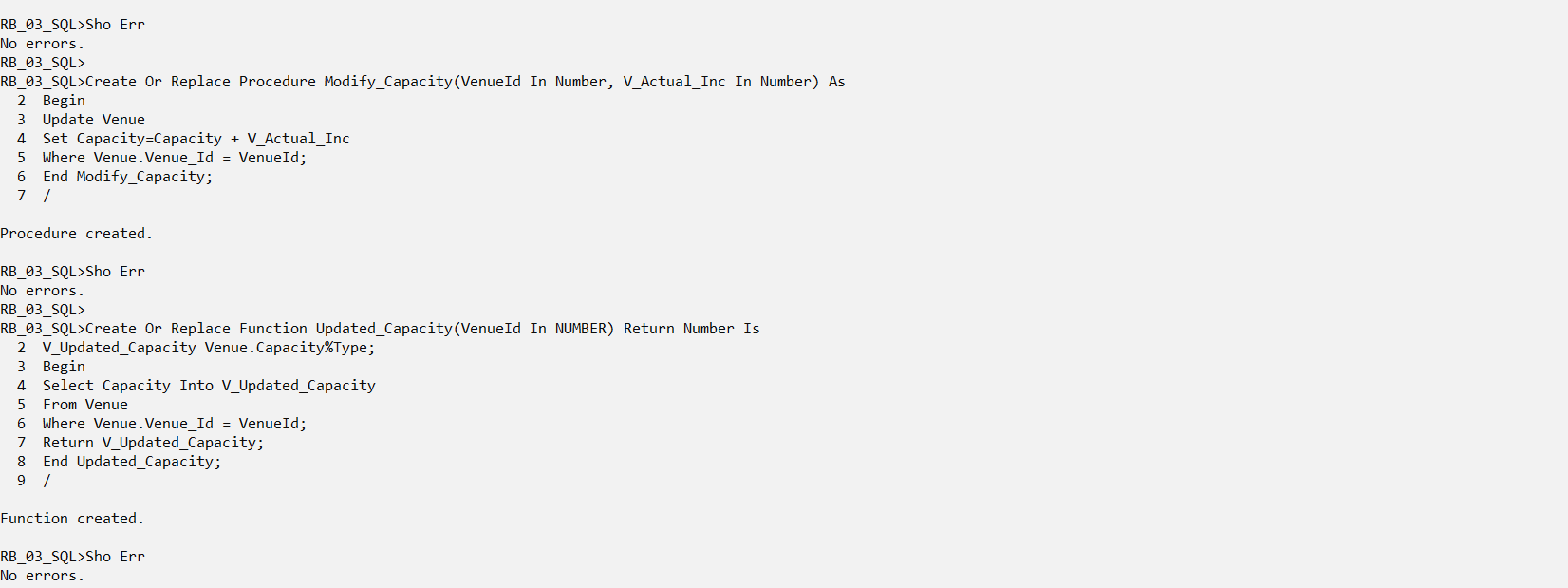
Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

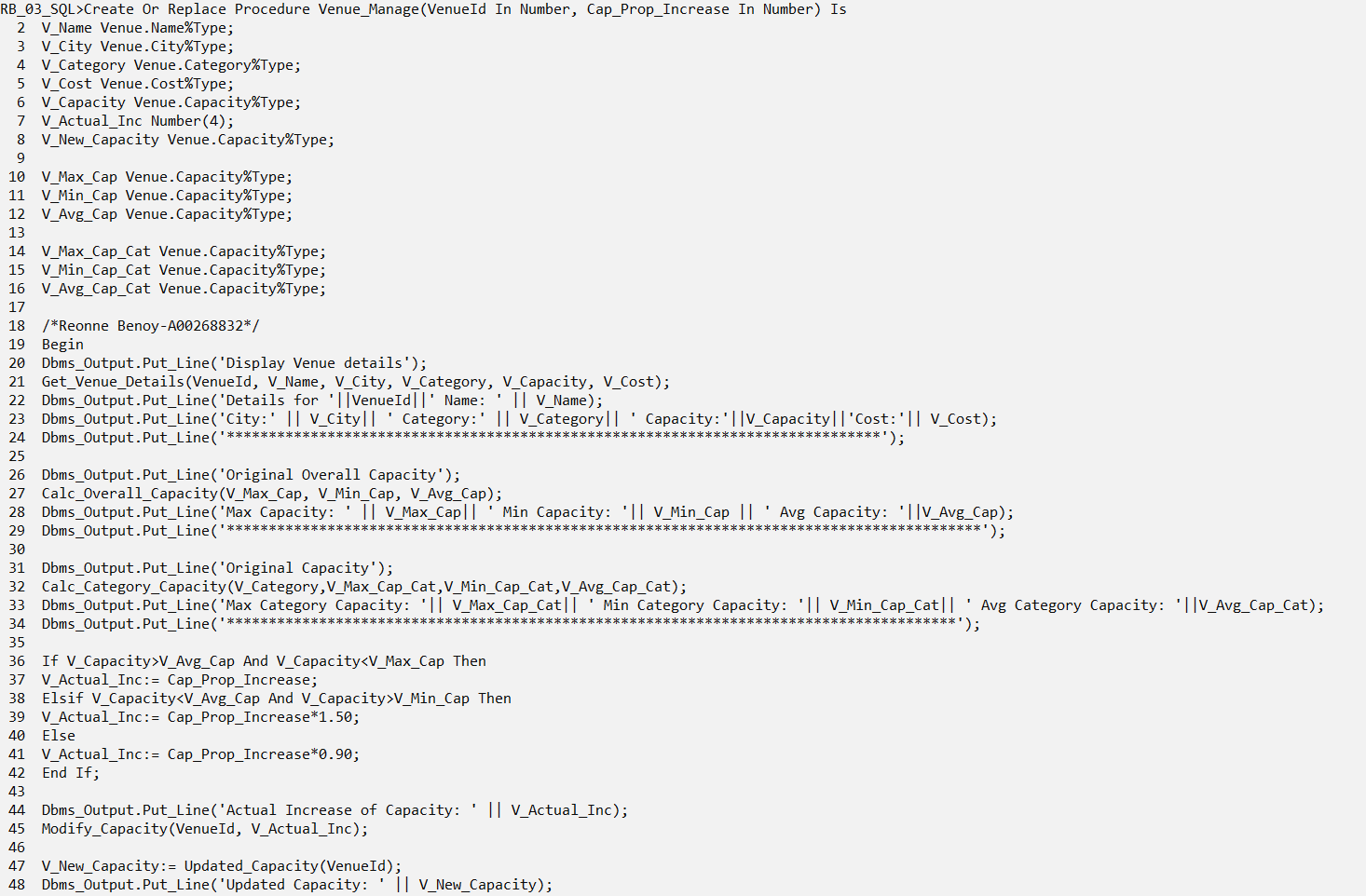
End Venue\_Manage;

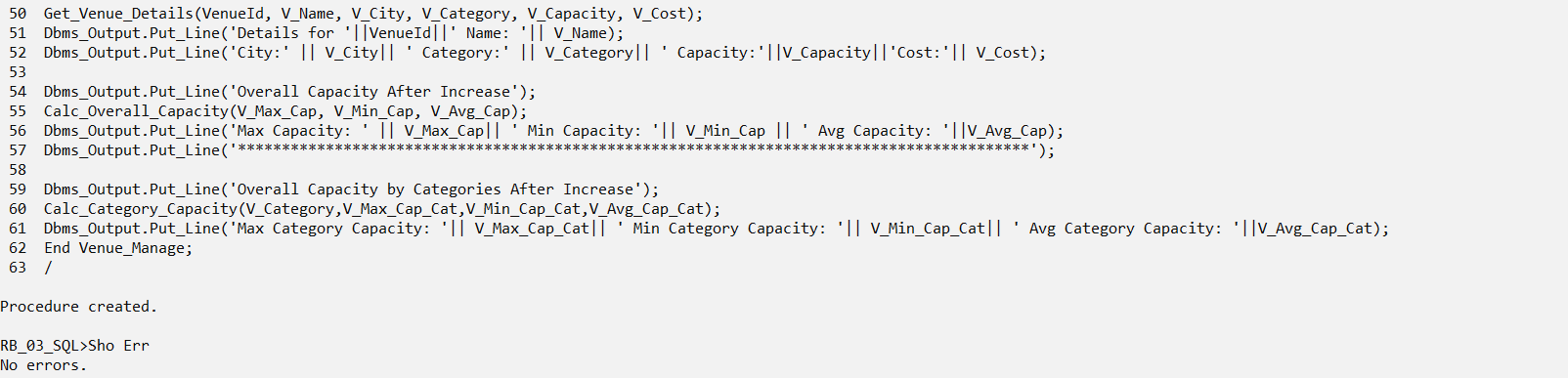
/

Sho Err

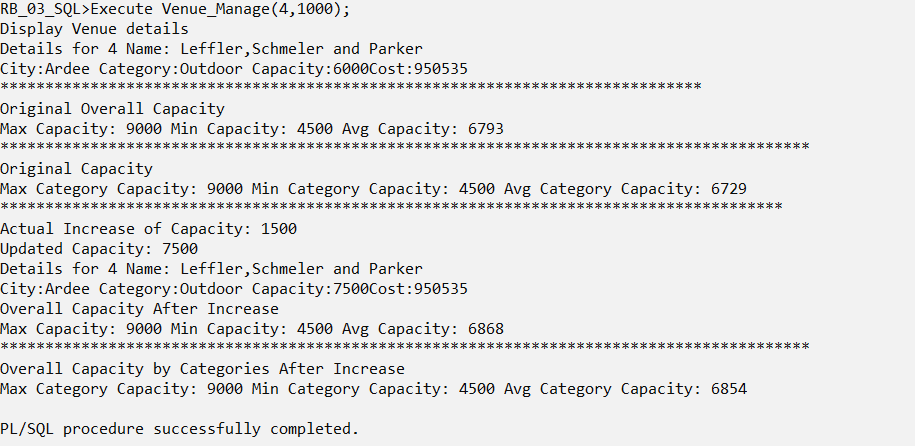








Execute Venue\_Manage(4,1000);



Rollback;

## Stage 3

**Local Procedures and Functions**

Clear Screen

Set Serveroutput On Size 1000000

Create Or Replace Procedure Venue\_Manage(VenueId In Number, Cap\_Prop\_Increase In Number) Is

V\_Name Venue.Name%Type;

V\_City Venue.City%Type;

V\_Category Venue.Category%Type;

V\_Cost Venue.Cost%Type;

V\_Capacity Venue.Capacity%Type;

V\_Actual\_Inc Number(4);

V\_New\_Capacity Venue.Capacity%Type;

V\_Max\_Cap Venue.Capacity%Type;

V\_Min\_Cap Venue.Capacity%Type;

V\_Avg\_Cap Venue.Capacity%Type;

V\_Max\_Cap\_Cat Venue.Capacity%Type;

V\_Min\_Cap\_Cat Venue.Capacity%Type;

V\_Avg\_Cap\_Cat Venue.Capacity%Type;

Procedure Get\_Venue\_Details(VenueId In Number, V\_Name Out Varchar, V\_City Out Varchar, V\_Category Out Varchar, V\_Capacity Out Number, V\_Cost Out Number) As

Begin

Select Name,City,Category,Cost,Capacity Into V\_Name,V\_City,V\_Category,V\_Cost,V\_Capacity

From Venue

Where Venue.Venue\_Id = VenueId;

End Get\_Venue\_Details;

Procedure Calc\_Overall\_Capacity(V\_Max\_Cap Out Number, V\_Min\_Cap Out Number, V\_Avg\_Cap Out Number) As

Begin

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap,V\_Min\_Cap, V\_Avg\_Cap

From Venue;

End Calc\_Overall\_Capacity;

Procedure Calc\_Category\_Capacity(V\_Category In Varchar, V\_Max\_Cap\_Cat Out Number, V\_Min\_Cap\_Cat Out Number, V\_Avg\_Cap\_Cat Out Number) As

Begin

Select Max(Capacity),Min(Capacity),Avg(Capacity) Into V\_Max\_Cap\_Cat, V\_Min\_Cap\_Cat, V\_Avg\_Cap\_Cat

From Venue

Where Venue.Category = V\_Category;

End Calc\_Category\_Capacity;

Procedure Modify\_Capacity(VenueId In Number, V\_Actual\_Inc In Number) As

Begin

Update Venue

Set Capacity=Capacity + V\_Actual\_Inc

Where Venue.Venue\_Id = VenueId;

End Modify\_Capacity;

Function Updated\_Capacity(VenueId In NUMBER) Return Number Is

V\_Updated\_Capacity Venue.Capacity%Type;

Begin

Select Capacity Into V\_Updated\_Capacity

From Venue

Where Venue.Venue\_Id = VenueId;

Return V\_Updated\_Capacity;

End Updated\_Capacity;

/\*Reonne Benoy-A00268832\*/

Begin

Dbms\_Output.Put\_Line('Display Venue details');

Get\_Venue\_Details(VenueId, V\_Name, V\_City, V\_Category, V\_Capacity, V\_Cost);

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: ' || V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Overall Capacity');

Calc\_Overall\_Capacity(V\_Max\_Cap, V\_Min\_Cap, V\_Avg\_Cap);

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Original Capacity');

Calc\_Category\_Capacity(V\_Category,V\_Max\_Cap\_Cat,V\_Min\_Cap\_Cat,V\_Avg\_Cap\_Cat);

Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

If V\_Capacity>V\_Avg\_Cap And V\_Capacity<V\_Max\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase;

Elsif V\_Capacity<V\_Avg\_Cap And V\_Capacity>V\_Min\_Cap Then

V\_Actual\_Inc:= Cap\_Prop\_Increase\*1.50;

Else

V\_Actual\_Inc:= Cap\_Prop\_Increase\*0.90;

End If;

Dbms\_Output.Put\_Line('Actual Increase of Capacity: ' || V\_Actual\_Inc);

Modify\_Capacity(VenueId, V\_Actual\_Inc);

V\_New\_Capacity:= Updated\_Capacity(VenueId);

Dbms\_Output.Put\_Line('Updated Capacity: ' || V\_New\_Capacity);

Get\_Venue\_Details(VenueId, V\_Name, V\_City, V\_Category, V\_Capacity, V\_Cost);

Dbms\_Output.Put\_Line('Details for '||VenueId||' Name: '|| V\_Name);

Dbms\_Output.Put\_Line('City:' || V\_City|| ' Category:' || V\_Category|| ' Capacity:'||V\_Capacity||'Cost:'|| V\_Cost);

Dbms\_Output.Put\_Line('Overall Capacity After Increase');

Calc\_Overall\_Capacity(V\_Max\_Cap, V\_Min\_Cap, V\_Avg\_Cap);

Dbms\_Output.Put\_Line('Max Capacity: ' || V\_Max\_Cap|| ' Min Capacity: '|| V\_Min\_Cap || ' Avg Capacity: '||V\_Avg\_Cap);

Dbms\_Output.Put\_Line('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*');

Dbms\_Output.Put\_Line('Overall Capacity by Categories After Increase');

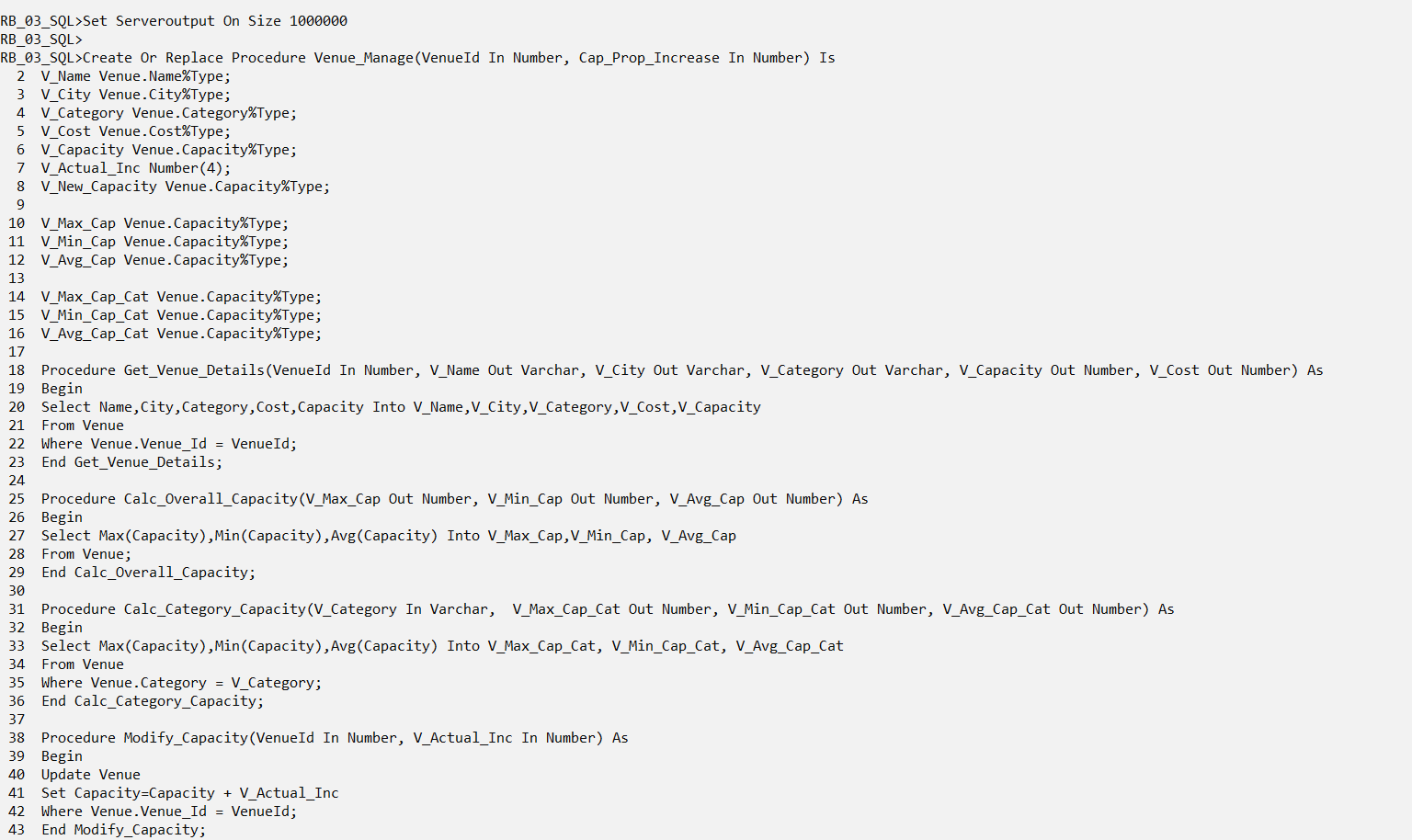
Calc\_Category\_Capacity(V\_Category,V\_Max\_Cap\_Cat,V\_Min\_Cap\_Cat,V\_Avg\_Cap\_Cat);

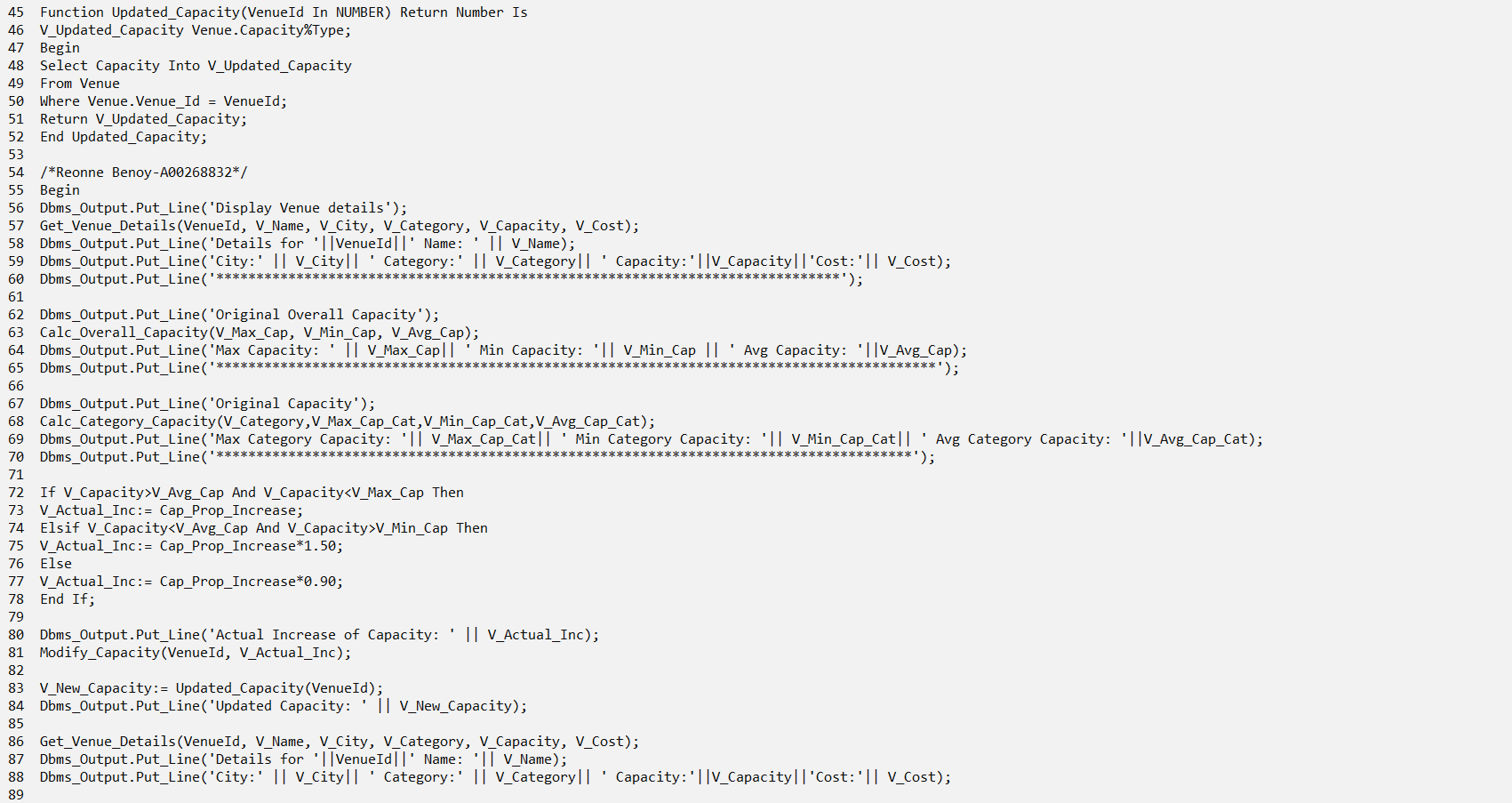
Dbms\_Output.Put\_Line('Max Category Capacity: '|| V\_Max\_Cap\_Cat|| ' Min Category Capacity: '|| V\_Min\_Cap\_Cat|| ' Avg Category Capacity: '||V\_Avg\_Cap\_Cat);

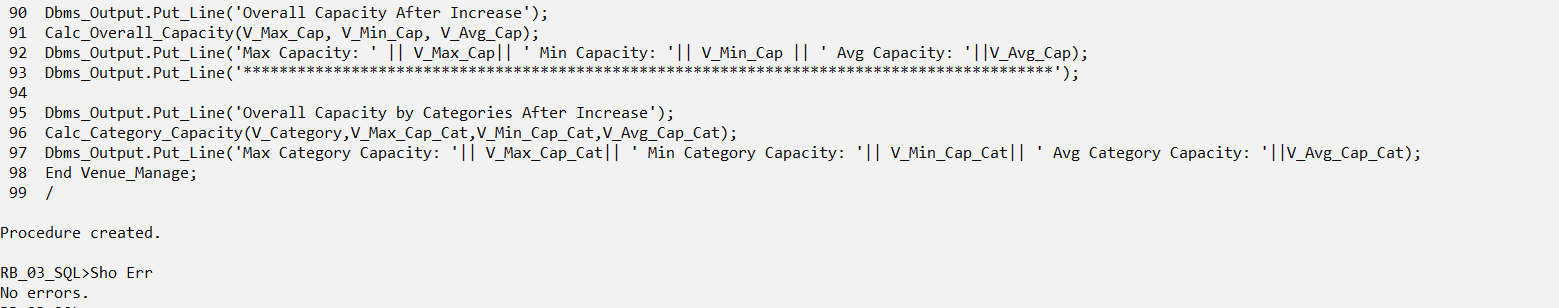
End Venue\_Manage;

/

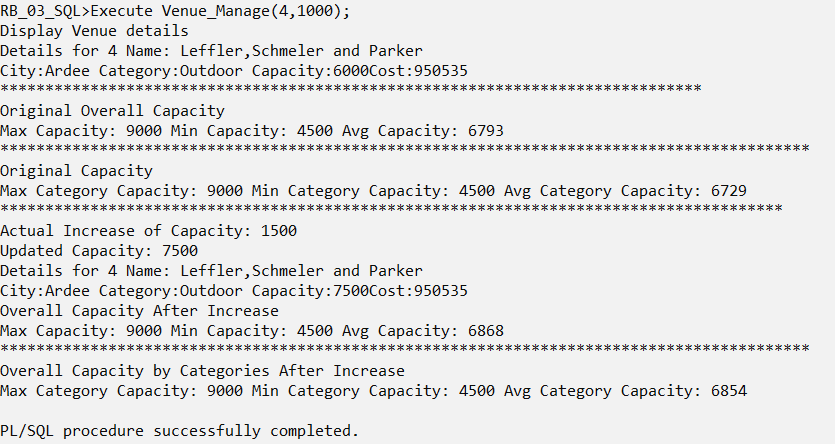
Sho Err







Execute Venue\_Manage(4,1000);



Rollback;

# Youtube Video Link

<https://youtu.be/hf2ZIN7U4J8>