**Schema diagram:**

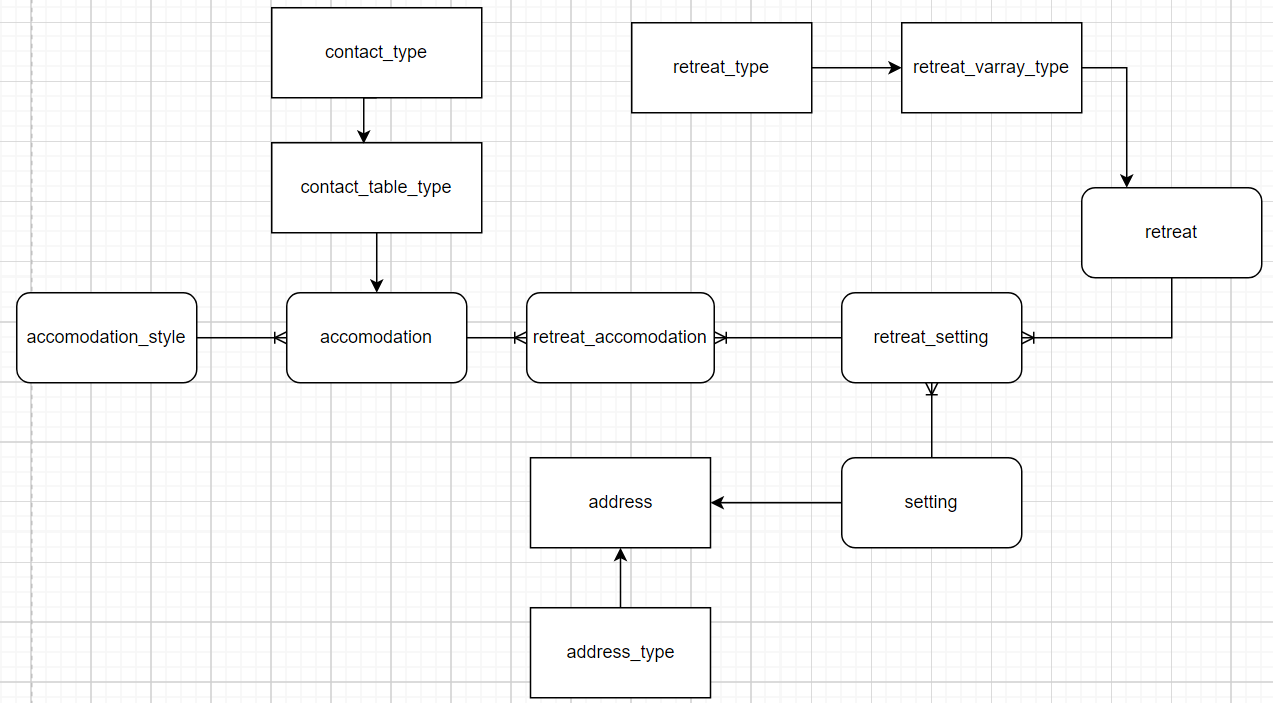


Table Specification:

| Tables | attribute | Key | Datatype |
| --- | --- | --- | --- |
| addresses |
|  | road |  | VARCHAR2(35) |
|  | city |  | VARCHAR2(35) |
|  | nation |  | VARCHAR2(30) |
|  |  |  |  |
| contact\_table\_type | contact\_type: |  |  |
|  | social\_media\_name |  | VARCHAR2(35) |
|  | contact\_detail |  | VARCHAR2(40) |
|  | email |  | VARCHAR2(40) |
|  |  |  |  |
| retreat\_varray\_type | retreat\_type(2): |  |  |
|  | activity |  | VARCHAR2(35) |
|  | age\_limit |  | CHAR(1) |
|  |  |  |  |
| accomodation\_styles | accomodation\_style\_id | pk | NUMBER(6) |
|  | room |  | VARCHAR2(50) |
|  |  |  |  |
| accomodations | accomodation\_id | pk | NUMBER(6) |
|  | accomodation\_style\_id | fk | NUMBER(6) |
|  | name |  | VARCHAR2(30) |
|  | no\_of\_room |  | NUMBER(6) |
|  | address |  | address\_type |
|  | price\_level |  | NUMBER(3) |
|  | facility |  | VARCHAR2(50) |
|  | contact\_media |  | contact\_table\_type |
|  |  |  |  |
| retreats | retreat\_id | pk | NUMBER(6) |
|  | type |  | retreat\_varray\_type |
|  | review |  | VARCHAR2(100) |
|  |  |  |  |
| settings | setting\_id | pk | NUMBER(6) |
|  | place |  | VARCHAR2(50) |
|  | address |  | ref of address\_type |
|  |  |  |  |
| retreat\_settings | setting\_id | fk | NUMBER(6) |
|  | retreat\_id | fk | NUMBER(6) |
|  | retreat\_setting\_id | pk | NUMBER(6) |
|  | duration |  | VARCHAR2(25) |
|  |  |  |  |
| retreat\_accomodations | accomodation\_id | fk | NUMBER(6) |
|  | retreat\_setting\_id | fk | NUMBER(6) |
|  | retreat\_accomodation\_id | pk | NUMBER(6) |
|  | arrival\_date |  | DATE |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function:** | | | |
| func\_calc\_months\_between  (in\_date\_from retreat\_accomodations.arrival\_date%TYPE, in\_date\_to retreat\_accomodations.arrival\_date%TYPE) RETURN NUMBER IS | | This functions ask two dates as input and returns the months between these date | |
| func\_calc\_total\_rooms  (in\_date\_from DATE, in\_date\_to DATE) RETURN NUMBER IS | | This function calculate and sums the room used between two dates, returns the result. | |
|  | |  | |
| **Procedure:** | | | |
| CREATE OR REPLACE PROCEDURE proc\_show\_no\_of\_rooms  (in\_date\_from retreat\_accomodations.arrival\_date%TYPE, in\_date\_to retreat\_accomodations.arrival\_date%TYPE) IS | This procedure calls the above functions and displayes the rooms used within that dates if the months between date is less than 3 months | | |
| proc\_ck\_arrival\_date(in\_date retreat\_accomodations.arrival\_date%TYPE) IS  CURSOR cur\_r\_accomodations IS | This procedure is used to show the accomodation name used in that date. It uses cursor to check for each records.Shows the accomodation name if date matches. | | |
| proc\_show\_setting\_place(in\_id IN NUMBER, out\_place OUT VARCHAR2)  IS | This procedure is created to demo in and out parameter. This store the value of place in out parameter out\_place from select query and return the value, where it is being called. | | |
|  |  | | |
| **Unnamed Block:** |  | | |
| DECLARE  vc\_place settings.place%TYPE;  vn\_counter NUMBER(6) := 1;  vn\_length NUMBER(6) := 0;  BEGIN  proc\_show\_setting\_place(8,vc\_place); | Unnamed block donot have names.  They are not store in database  This block was created to call proc\_show\_setting\_place and store the value in vc\_place returned by that OUT parameter of that procedure. | | |
|  |  | | |
| **Implicit Cursor:** |  | | |
| proc\_update\_acc\_style\_cursor(in\_acc\_id NUMBER, in\_room VARCHAR2) IS | This cursor updates when the given id exists in the table, if record exists the message showing record updated is shown. | | |
|  |  | | |
| **Trigger:** | | | |
| trig\_arrival\_date\_ck  BEFORE INSERT OR UPDATE OF arrival\_date ON retreat\_accomodations | | | This trigger gets activated when there is update or insert on arrival\_date. This checks if inserted/updated arrival\_date is less than current system date. If it is less than it calls or execute the proc\_show\_no\_of\_rooms procedure. |
|  | | |  |
| Queries | | | |
| SELECT road, city, nation FROM addresses; | | | Selecting values from object table addresses |
| SELECT REF(a), road, city, nation  FROM addresses a  WHERE road = '40 JAPAN STREET'; | | | Extracting reference (OID) from object table |
| SELECT a.address.road, a.address.city, a.address.nation FROM accomodations a; | | | Extracting object column data of acomodations table using dot notation |
| SELECT r.retreat\_id, t.activity, t.age\_limit  FROM retreats r,  TABLE(r.type)t  WHERE retreat\_id = 1; | | | Selecting data from varray of retreats table |
| SELECT a.accomodation\_id, aa.email  FROM accomodations a,  TABLE(a.contact\_media)aa  WHERE aa.contact\_detail = 'GRAND'; | | | Querying nested table contact\_media of the table accomodations |

**Proposed Automation Strategy:**

To implement and organize database from the selected tables, we used object type like address\_type, retreat\_varray\_type, contact\_table\_type, which helps to store multiple data within single field. The automated processes like procedure, triggers, cursors, functions were used which are described in below section.

**Function:**

### 1:) func\_calc\_months\_between:

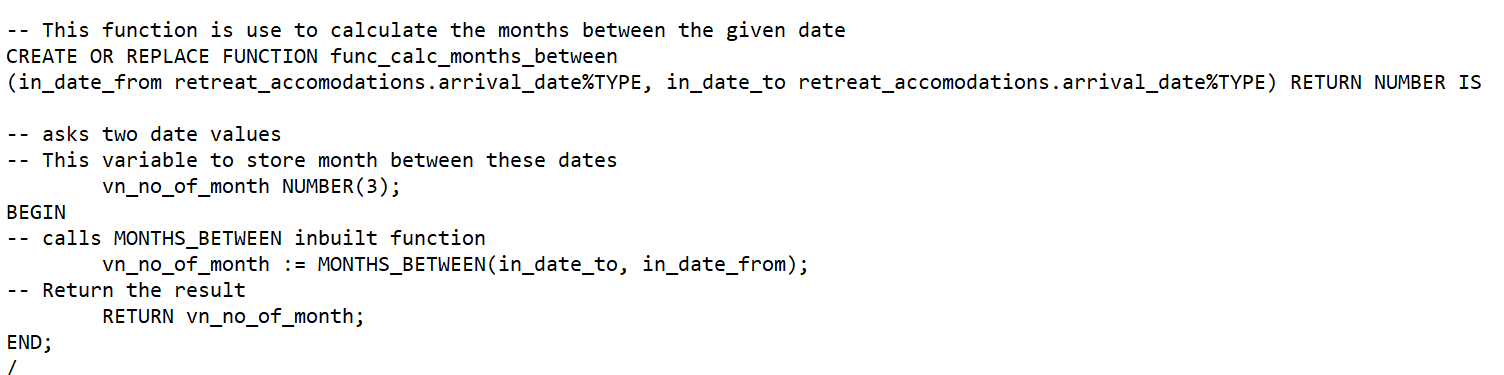


Fig: Code for func\_calc\_months\_between

This function is created for getting the months between the two dates.

It asks two date parameters to calculate the months between these date. It uses vn\_no\_of\_month variable to store the months between them.

After successful calculation It return value to the procedure (proc\_show\_no\_of\_rooms) from where it is being called.

### 2:) func\_calc\_total\_rooms:

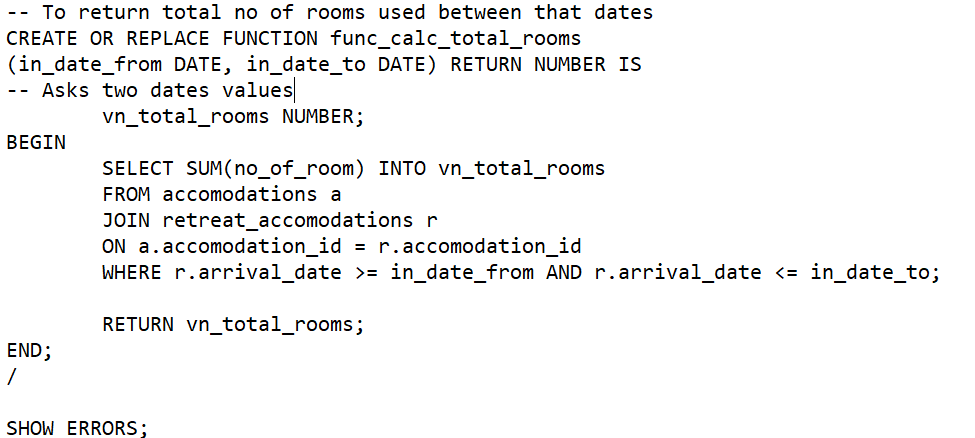


Fig: code for func\_calc\_total\_rooms

The main aim of this function is to calculate total no of rooms between that date or time period. It asks two dates value to calculate the operations.

In the BEGIN part,

SELECT statement is used with sum function and two tables accomodations, retreat\_accomodations are used using JOIN function within given that dates values as parameter. After adding all no of rooms it stores that value in vn\_total\_rooms and return to the procedure proc\_show\_no\_of\_rooms

**PROCEDURE:**

### 1:) proc\_show\_no\_of\_rooms:

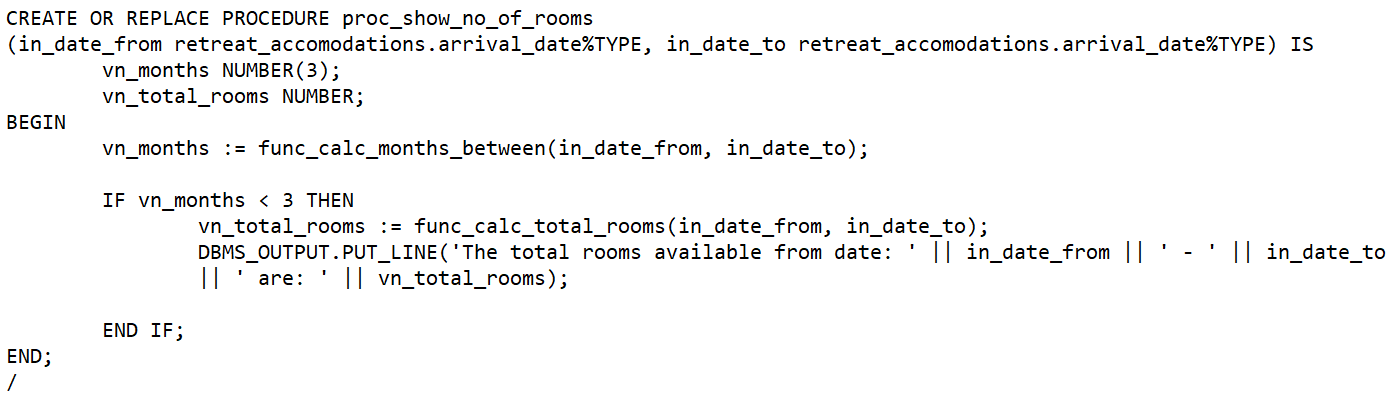


Fig: code for proc\_show\_no\_of\_rooms

The main aim of making this procedure is to get the total no of rooms used from given date to current system date but within period of 3 months only. This procedure is used like statement/history of no of rooms of accomodations used during retreat in range of 3 months between dates.

Here, it asks for two date parameters of anchor type. Two variables are declared to store no of months between two dates respectively. To check the months between the dates it calls function named as func\_calc\_months\_between and passing two date values as argument. If the months between the dates is less than 3 months then it calls the function func\_calc\_total\_rooms which returns the value of total rooms used during that period and displays the result.

### 2:) proc\_ck\_arrival\_date:

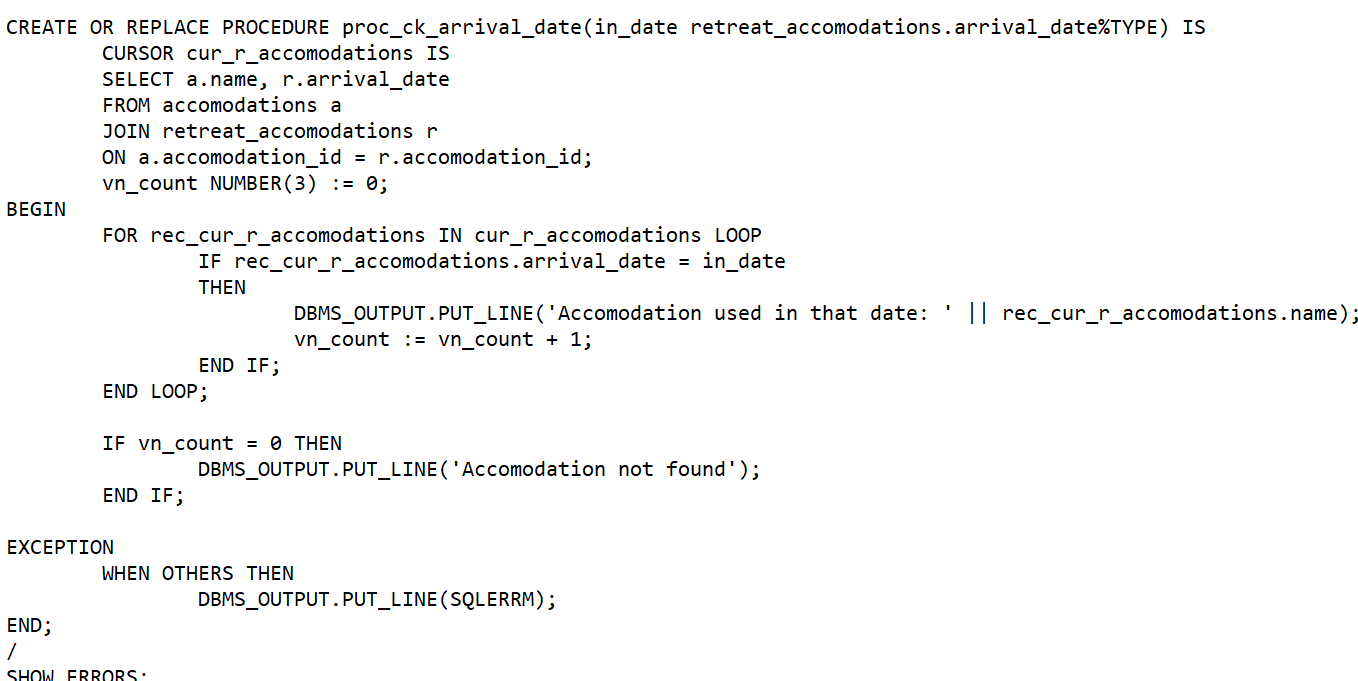
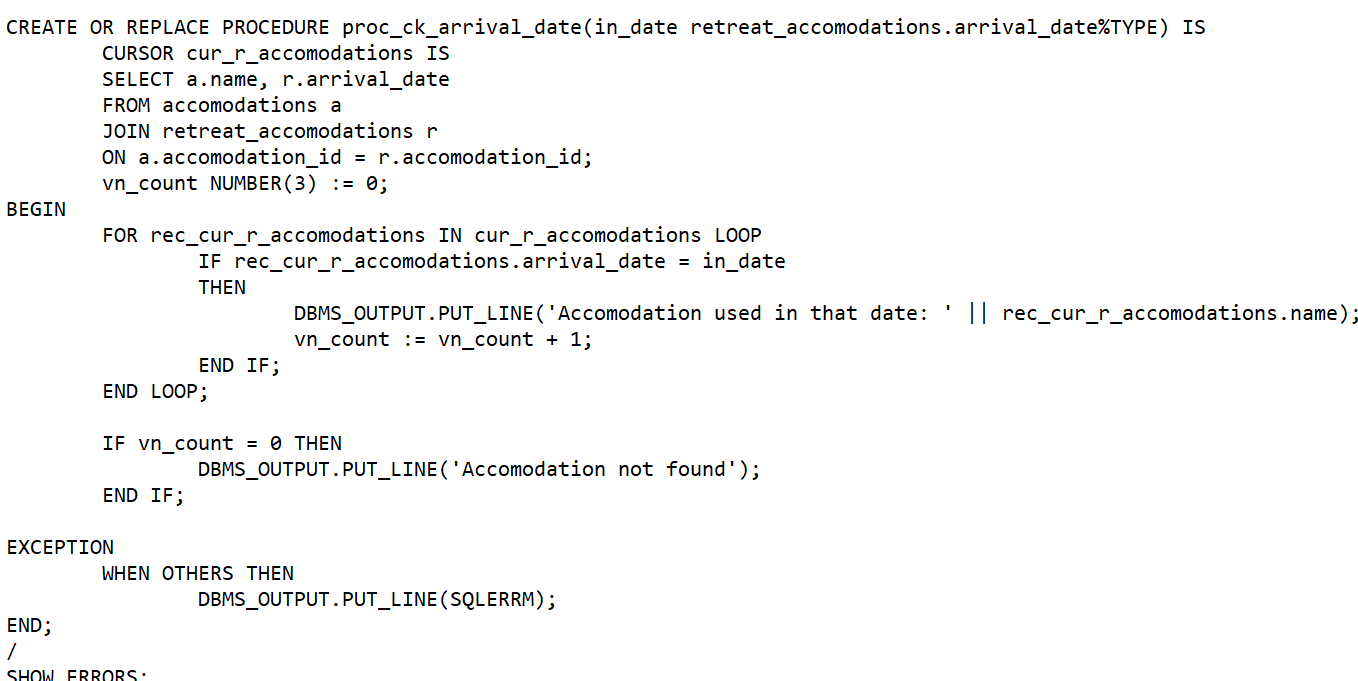


Fig: code for proc\_ck\_arrival\_date:

This procedure is to show all accomodations name used during that given date for retreats.

It creates cursor to select all the accomodation name used in retreat\_accomodations table by JOIN function. It uses vn\_count variable to count the result.

In BEGIN section, it uses FOR loop to achieve each row of cursor. It increases counter variable and prints the accomodation name only if arrival\_date field of table retreat\_accomodations table matches with user input date. If counter variable is still zero then it prints message saying accomodation not found. If error occurs in the program the EXCEPTION handles the error.

### 3:) proc\_show\_setting\_place:

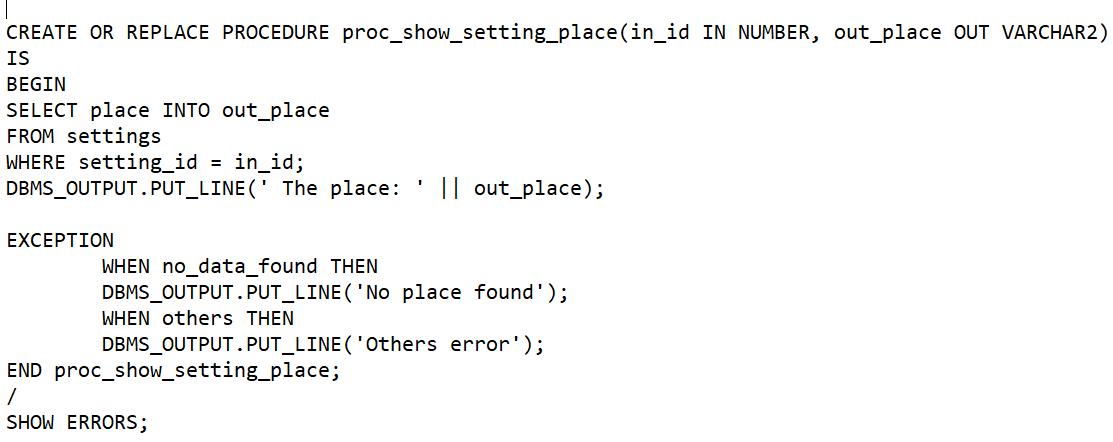


Fig: proc\_show\_setting\_place

Proc\_show\_setting\_place is the procedure that takes in and out two parameters.There is a variable called out\_place in the out parameter.After that, select query prints the out\_place variable after it stores the matching data for setting\_id and in\_id in the out\_place variables.In the event that the data could not be located, it displays "no place found" and "other errors."

**PL/SQL Unnamed block:**

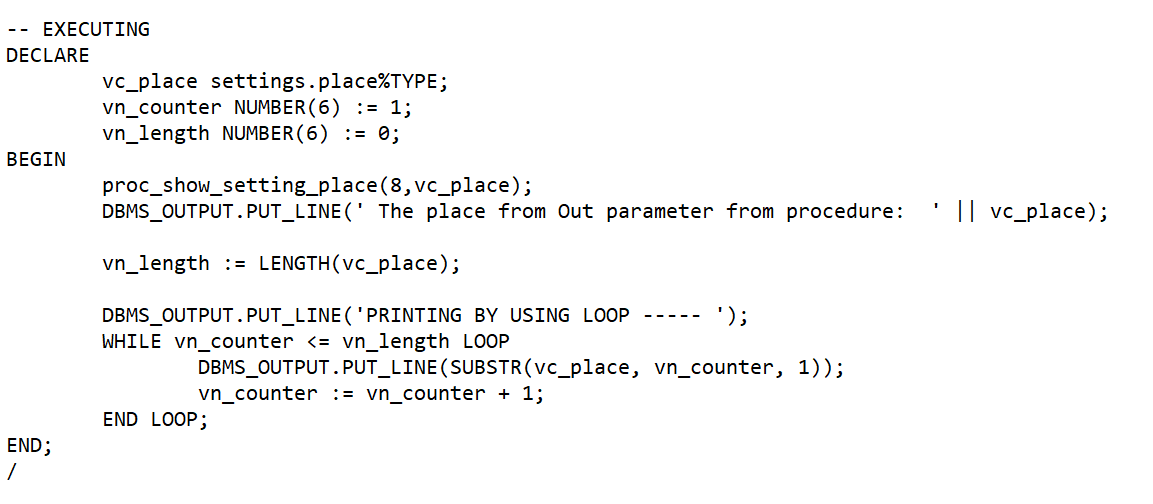
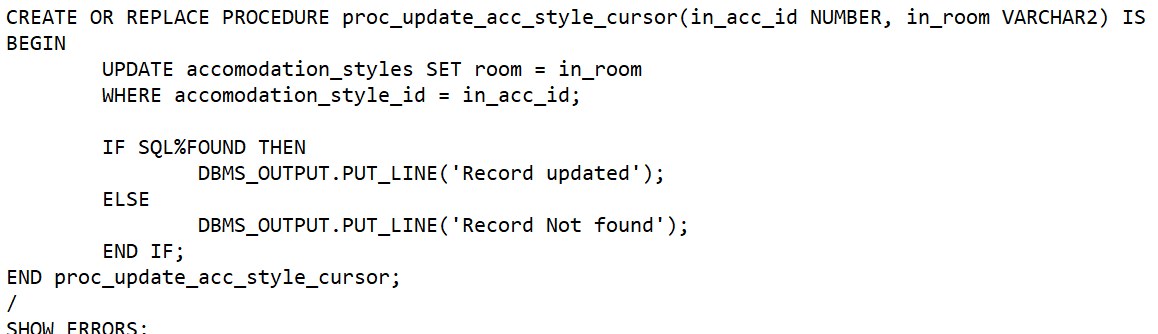
****

Fig: Unnamed block calling proc\_show\_setting\_place

This is the unnamed block having vc\_place to store the value return from out parameter of the procedure proc\_show\_setting\_place. Once it calls the procedure the value of place inside the procedure is displayed and the value returned or out by procedure is printed by unnamed block. Both obtain same result which proves that same values are returned via procedure out parameter. For further, The name of place obtained is displayed via using WHILE loop. It runs until the length of string. And, prints individual characters in different line. This unnamed block is not stored in database.

**IMPLICIT CURSOR:**

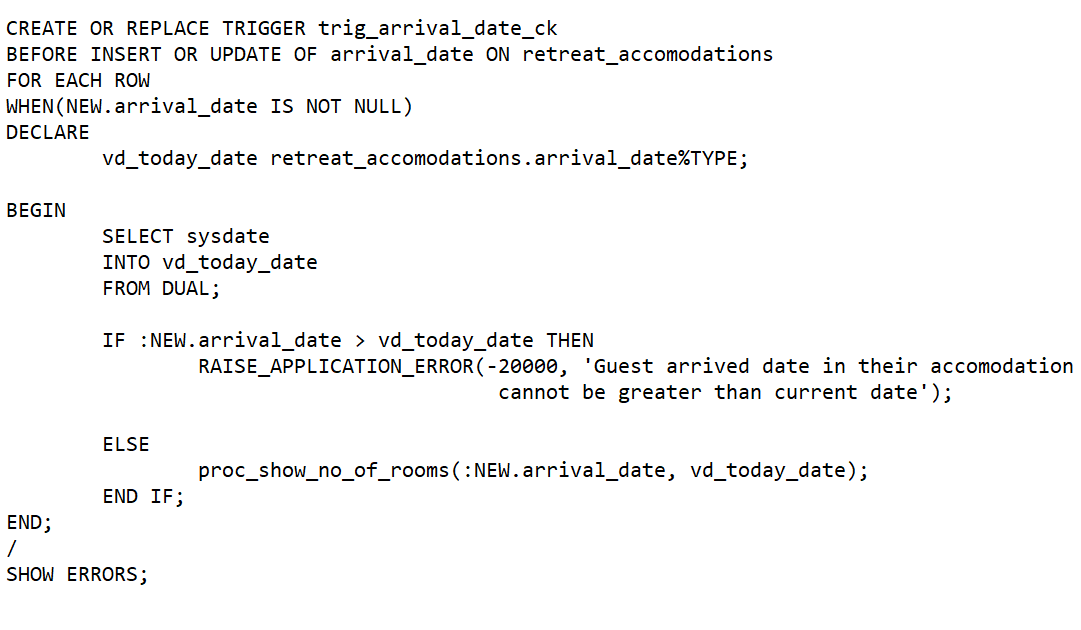
****

**Fig: code for proc\_update\_acc\_style\_cursor**

This cursor is called when we execute procedure. When the value to update and id is passed during the execution. The implicit cursor SQL checks whether the record exists or not. If the record is successfully updated, message showing ‘Record updated’ is shown, if record is not found the ‘Record Not Found’ message is shown.

**TRIGGER:**

**1:) trig\_arrival\_date\_ck:**

****

**Fig: code for trig\_arrival\_date\_ck**

This trigger is used for checking the arrival\_date of guests and calling the procedure proc\_show\_no\_of\_rooms.

This trigger is called before insert or update of arrival\_date on retreat\_accomodations and executes only if new inserted/ updated arrival date value is not null. It stores the current value of system in vd\_today\_date variable and checks whether the arrival\_date is less than system date or not. As guests arrives before or during the current system date in accomodations. So, It raise applications error saying ‘Guest arrived date in their accomodation cannot be greater than current date’ and following insert or update query will not be successfully executed.

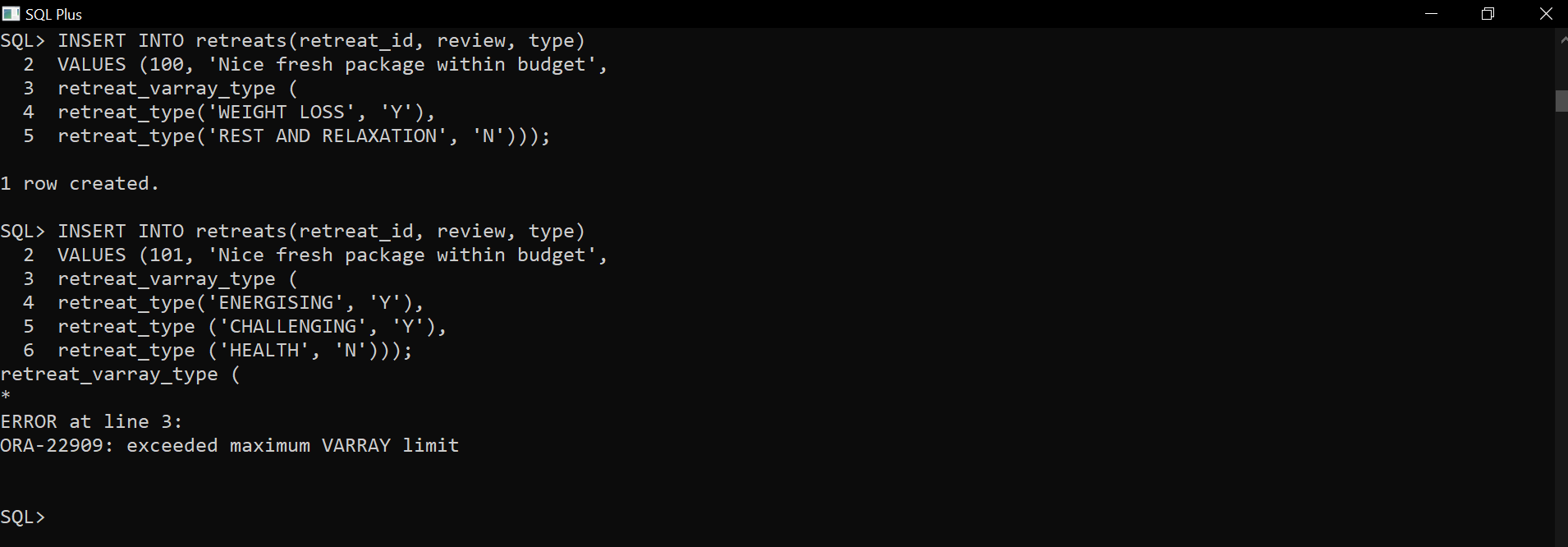
If arrival date is within system date it calls the procedure proc\_show\_no\_of\_rooms where the operation inside it will be performed.

**Test Plan:**

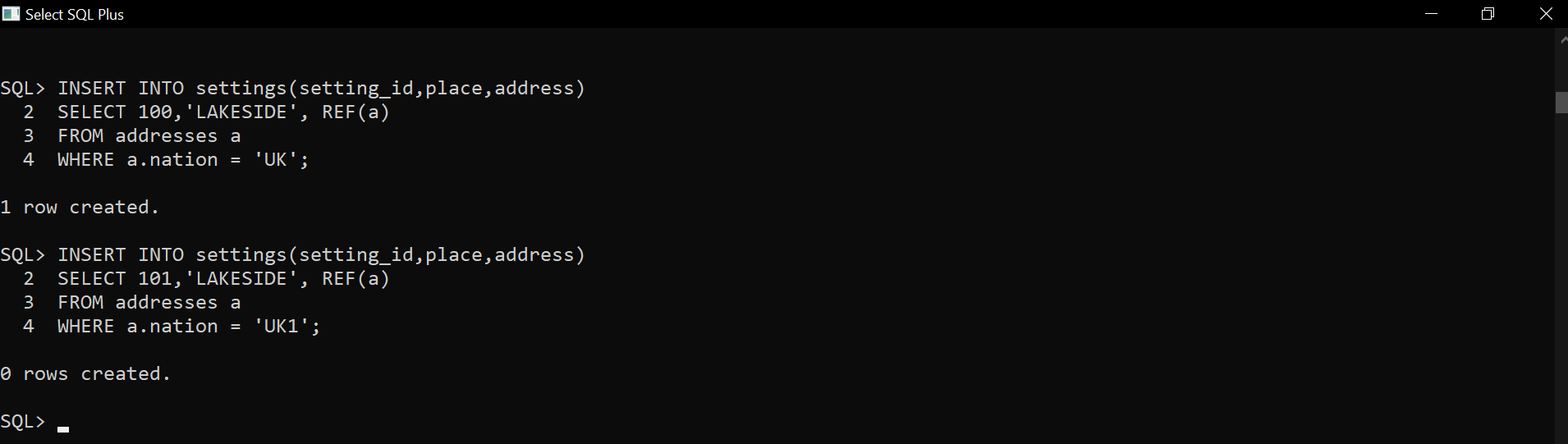
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Test | Expected Results | Actual Result | Action |
| 1 | INSERT INTO retreats(retreat\_id, review, type)  VALUES (100, 'Nice fresh package within budget',  retreat\_varray\_type (  retreat\_type('WEIGHT LOSS', 'Y'),  retreat\_type('REST AND RELAXATION', 'N'))); | - Successful Insert as VARRAY is of size 2 | Same as expected |  |
| 2 | INSERT INTO retreats(retreat\_id, review, type)  VALUES (101, 'Nice fresh package within budget',  retreat\_varray\_type (  retreat\_type('ENERGISING', 'Y'),  retreat\_type ('CHALLENGING', 'Y'),  retreat\_type ('HEALTH', 'N'))); | - ERROR showing maximum VARRAY limit | Same as expected |  |
| 3 | INSERT INTO settings(setting\_id,place,address)  SELECT 100,'LAKESIDE', REF(a)  FROM addresses a  WHERE a.nation = 'UK'; | -- 1 row created, as there is row having nation 'UK' in addresses table | Same as expected |  |
| 4 | INSERT INTO settings(setting\_id,place,address)  SELECT 101,'LAKESIDE', REF(a)  FROM addresses a  WHERE a.nation = 'UK1'; | -- 0 row created, as there is no row having 'UK1' in addresses table | Same as expected |  |
| 5 | INSERT INTO accomodations(accomodation\_id,accomodation\_style\_id, name, no\_of\_room, price\_level, facility, contact\_media)  VALUES (144, 4, 'HOTEL GRAND',20, 450,'CASINO',  contact\_table\_type(  contact\_type('TWITTER', 'grand@12', 'grand@123gmail.com'),  contact\_type('FB', 'GRAND', 'hotelgrand@12gmail.com'))); | -- 1 row created | Same as expected |  |
| 6 | Test proc\_ck\_arrival\_date: (To show accomodation name used in that date)  EXEC proc\_ck\_arrival\_date('04-FEB-21') | -- RESULT: Accomodation used in that date: FOUNTAIN FUN | Same as expected | Checks each row by use of cursor and displays name if input date matches with record |
| 7 | Test proc\_ck\_arrival\_date:  Exec proc\_ck\_arrival\_date('1-jan-44') | -- Shows Accomodation not found | Same as expected |  |
| 8 | SELECT SUM(no\_of\_room) FROM accomodations WHERE accomodation\_id = 11 OR accomodation\_id = 14; | -- 30 rooms are shown | Same as expected |  |
| 9 | proc\_show\_no\_of\_rooms:  EXECUTE proc\_show\_no\_of\_rooms('04-FEB-22', '04-MAR-22')  (accomodation\_id used in this dates are: 11 and 14 ) | -- Displays,  The total rooms available from date: 04-FEB-22 - 04-MAR-22 are: 30 | Same as expected |  |
| 10 | Test trigger: trig\_arrival\_date\_ck  INSERT INTO retreat\_accomodations(accomodation\_id,retreat\_accomodation\_id, arrival\_date)  VALUES(14,102, '11-DEC-2022'); | -- Raise application error due to date greater than current system date | Same as expected |  |
| 11 | Test trigger: trig\_arrival\_date\_ck  INSERT INTO retreat\_accomodations(accomodation\_id,retreat\_accomodation\_id, arrival\_date)  VALUES(11,108, '28-sep-2022'); | Shows the 10 rooms used between that inserted date and system date | Same as expected |  |
| 12 | SELECT accomodation\_id,  CEIL(price\_level),FLOOR(price\_level),ROUND(price\_level),TRUNC(  price\_level),price\_level  FROM accomodations  WHERE price\_level = (SELECT MAX(price\_level) FROM accomodations); | --  15  650  650  650  650  650 | Same as expected |  |
| 13 | SELECT a.name, a.facility  FROM accomodations a  WHERE a.address.city IN  (SELECT s.address.city  FROM settings s  WHERE s.address.nation = 'UK'); | --Name: null  Facility:  CROWN PLAZA  DISABLED ROOM | Same as expected |  |
| 14 | SELECT rs.retreat\_setting\_id,  rs.duration,  ra.retreat\_accomodation\_id,  ra.arrival\_date  FROM retreat\_settings rs  LEFT JOIN retreat\_accomodations ra  ON ra.retreat\_setting\_id = rs.retreat\_setting\_id; | --6 rows selected. | Same as expected |  |
| 15 | SELECT a.accomodation\_id,a.name,a.address,  c.email  FROM accomodations a,  TABLE(a.contact\_media) c; | --9 rows selected | Same as expected |  |
| 16 | SELECT a.accomodation\_id,  a.accomodation\_style\_id,a.name,  a.no\_of\_room,  a.address.road,  a.address.city,  a.address.nation  FROM accomodations a  ORDER BY accomodation\_id; | -- Shows all data of table in the given order order | Same as expected |  |
| 17 | EXEC proc\_update\_acc\_style\_cursor(4, 'RUSTIC MOUNTAIN') | -- Message showing: Record Updated; | Same as expected | Should Insert  Value to update before executing procedure. |
| 18 | EXEC proc\_update\_acc\_style\_cursor(5, 'ABC') | -- Shows,  Record Not found | Same as expected |  |

**Screenshots of the test plans:**

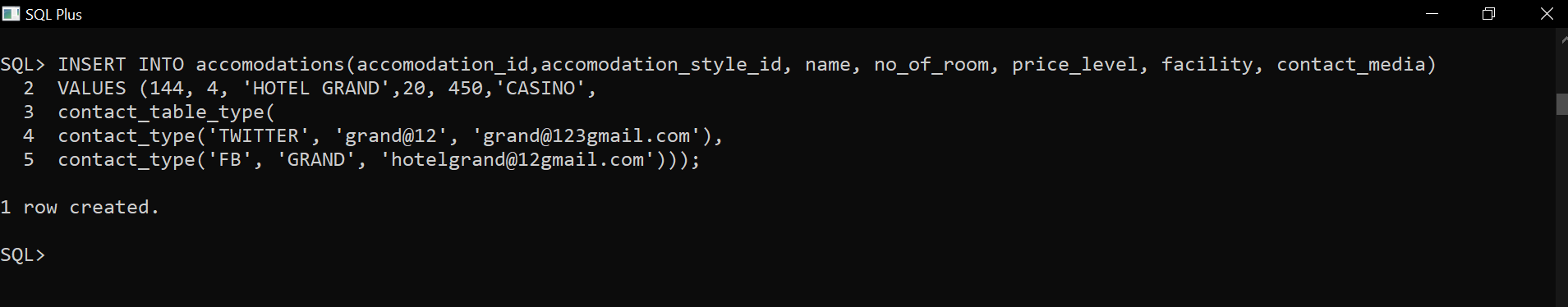
Given tests no. 1 and 2 respectively query are inserting with VArrays of simple single types. In test 1 query executes successfully whereas the VArray size limit exceeded in the test 2 query.

 Fig: Screenshot of the Test no. 1 and 2 respectively

The test no. 3 and 4 defines the inserting process into tables with reference (Ref) to object tables. The given query gives the result as expected.

 Fig: Screenshot of the Test no. 3 and 4 respectively

The test no.5 is the query of inserting into table with a VArray that is of complex types. Output is same as expected. Screenshot of the test is given below:

 Fig: Screenshot of the Test no. 4

The given test no.6 and 7 are for testing procedures. Screenshot of the testing is:

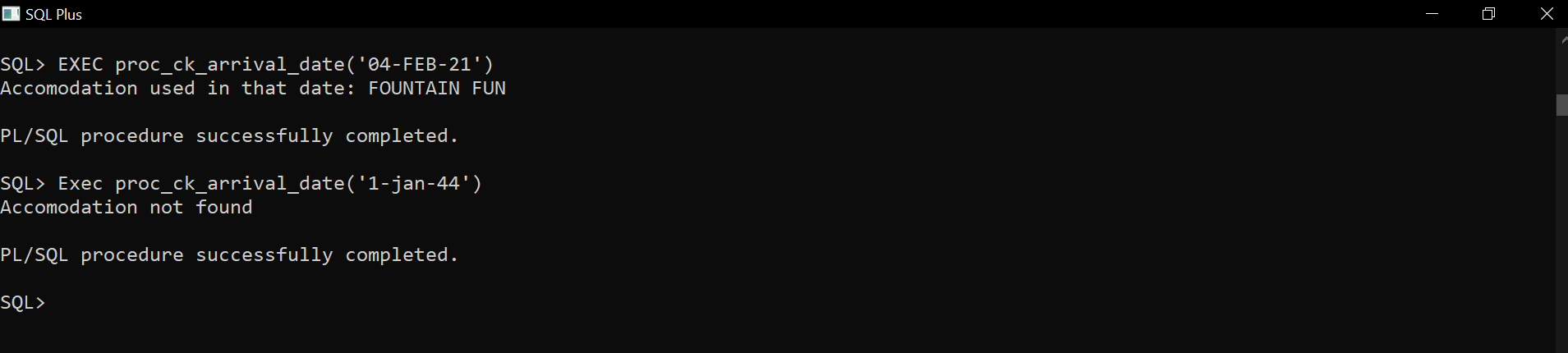


Fig: Screenshot of Test no. 6 and 7 respectively

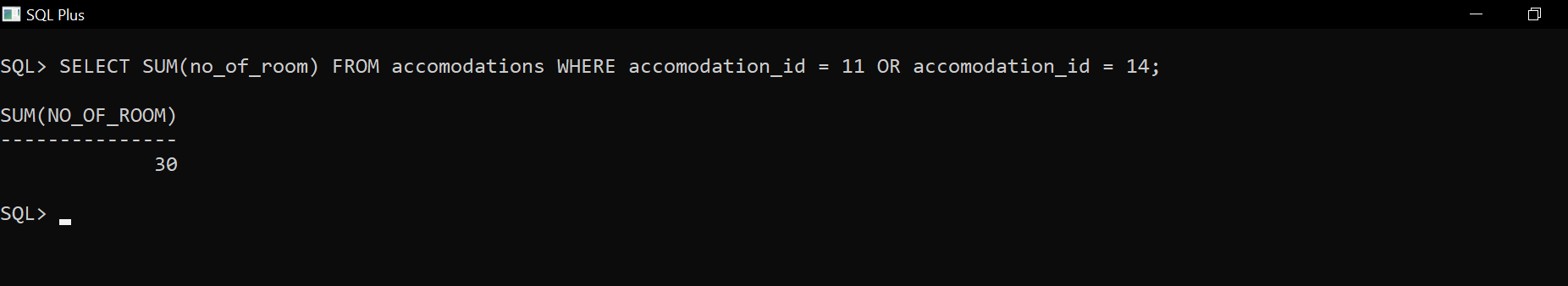
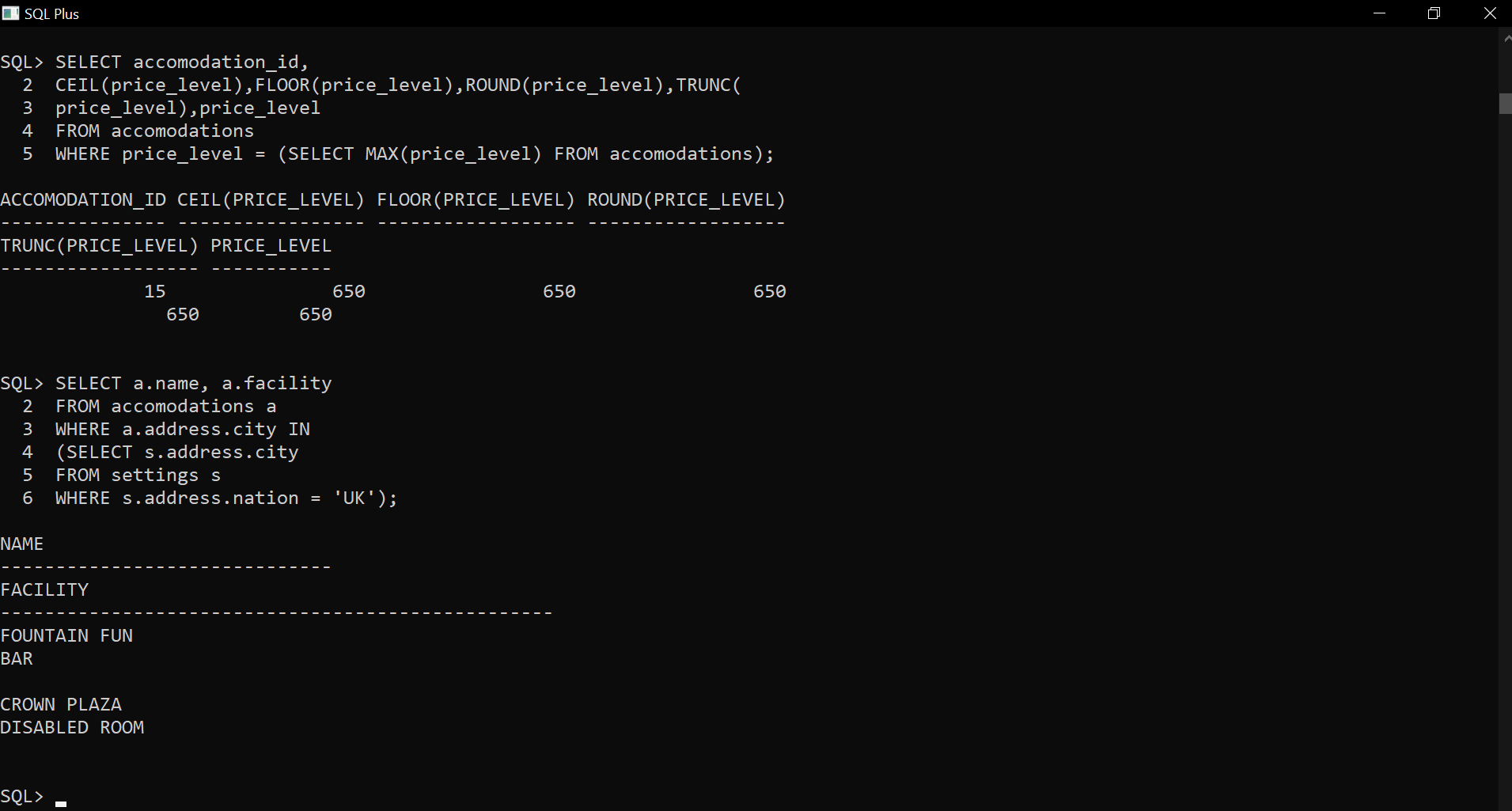
The test no. 8 is to select and add the no\_of\_rooms of id 11 or 14.

Fig: Screenshot of Test no. 7

Test no. 10 and 11 resp. is to test the triggers. Test no. 10 displays error due to the date greater than current system date whereas the test no. 11 gives the output between inserted date and system date. Screenshot is given below:-

Fig: Screenshot of Test no. 10 and 11 respectively

The test no. 12 uses different numeric function to select the inserted data. Such as CEIL() function to return smallest integer. Similarly the test no. 13 uses nested queries to select the data.

 Fig: Screenshot of the Test no. 12 and 13 respectively

The test no. 14 uses left join query to display the expected output.

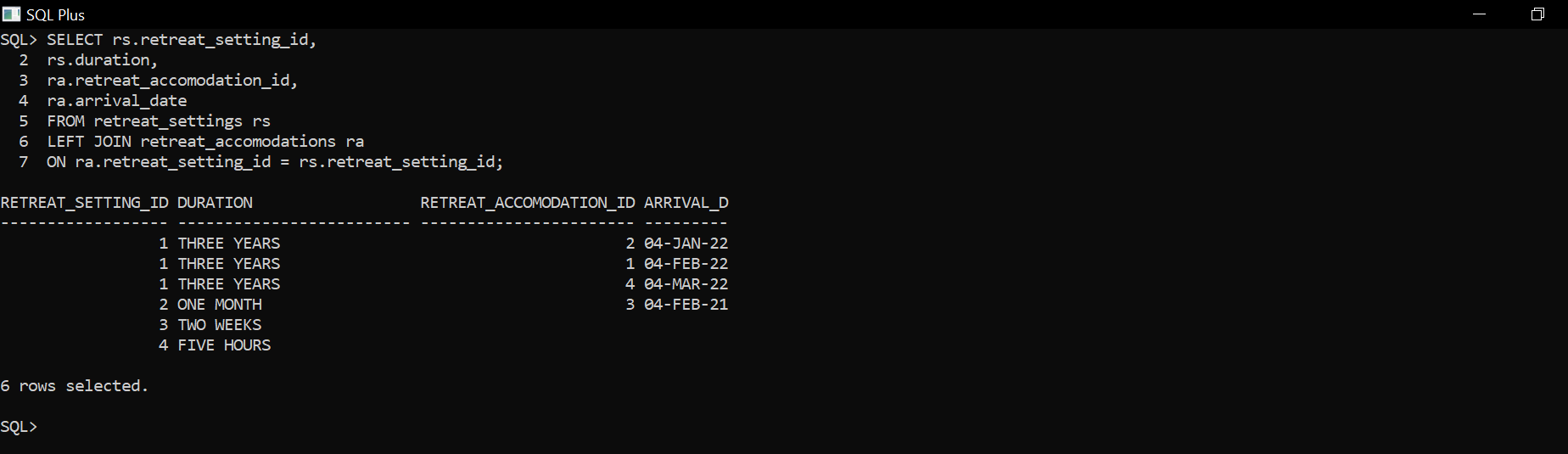


Fig: Screenshot of the Test no. 14

]

The test no. 15 is querying a VArray useful output the uses an un-nested query and creates pseudo table. It displays the output.

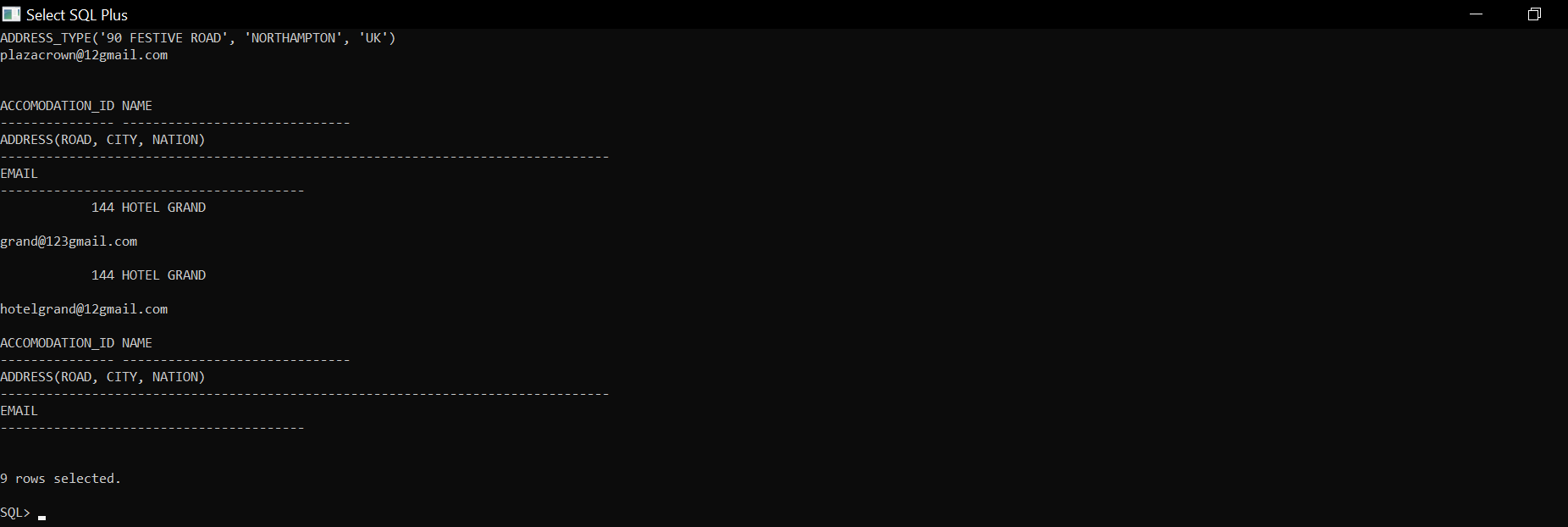
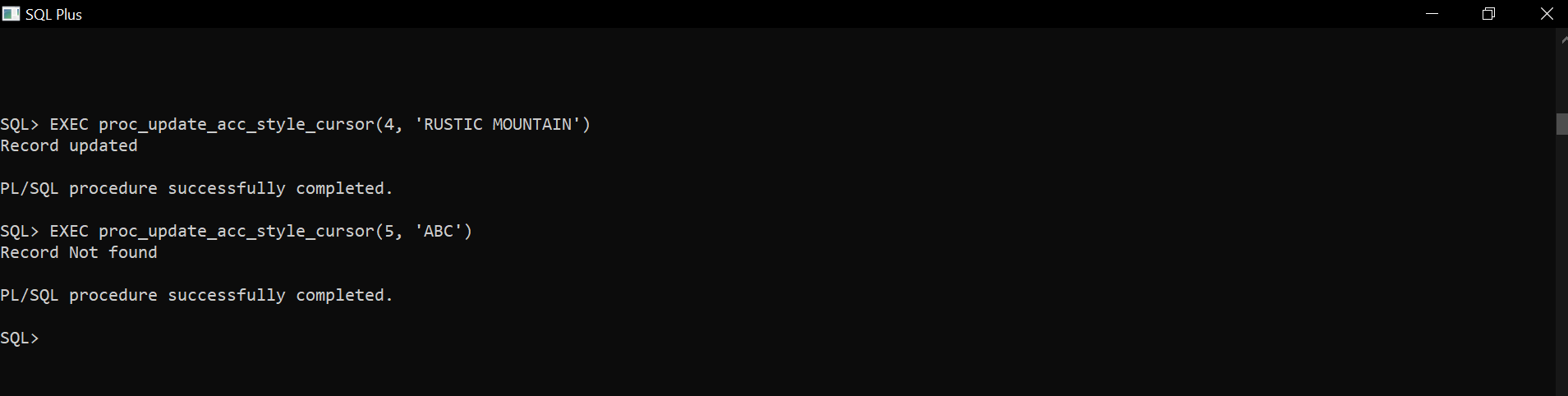
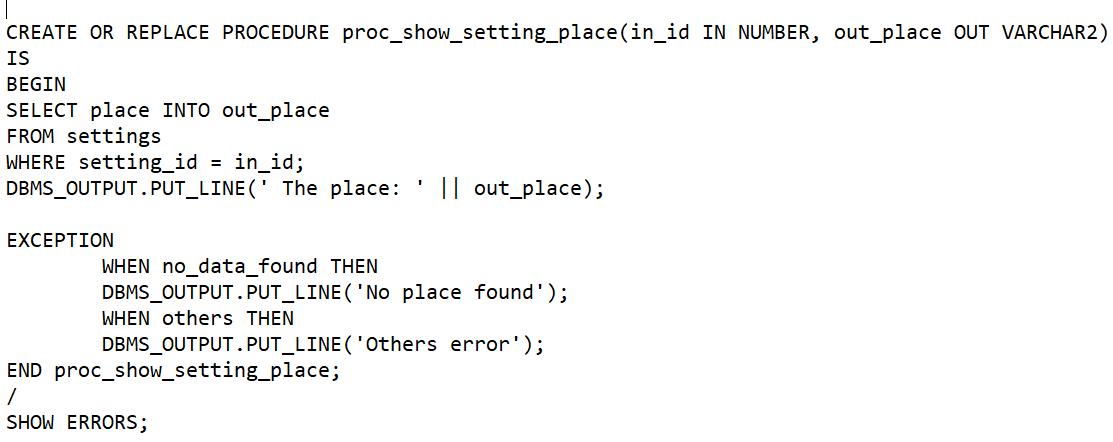


Fig: Screenshot of the Test no. 15

The test no. 17 and 18 respectively are to test the used cursor.

 Fig: Screenshot of the Test no. 17 and 18 respectively

**Procedure: proc\_show\_setting\_place:**



The method that takes in and out two boundaries is Proc\_show\_setting\_place.There is a variable called out\_place in the out parameter.After that, the select query prints the out\_place variable and stores the matching data for setting\_id and in\_id in the out\_place variables.

“The place” is the procedure's output, which appears after the call, along with “other errors” in an unnamed block.

The procedure is called via unnamed blocked inside BEGIN part. The value of place printed by procedure and inside unnamed block was same which is shown in below screenshot. The returned value is printed in different lines using while loop.

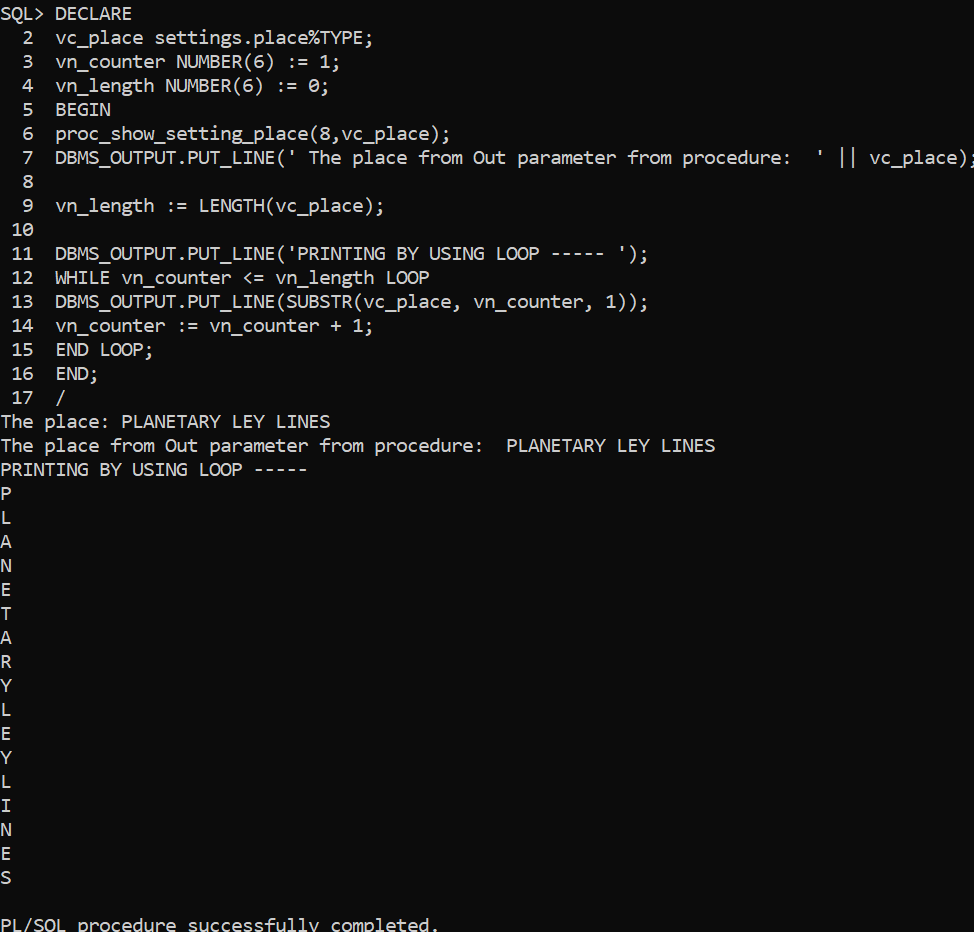


Fig: Calling /executing procedure inside unnamed block and comparing results.