LOOPS

WHILE LOOP

```
19  DECLARE
20     a number(2) := 10;
21  BEGIN
22     WHILE a < 20 LOOP
23        dbms_output.put_line('value of a: ' || a);
24        a := a + 3;
25     END LOOP;
26  END;
```

```
Statement processed.
value of a: 10
value of a: 13
value of a: 16
value of a: 19
```

GO TO Statement

```
29  --GO TO
30  DECLARE
31     a number(2) := 1;
32  BEGIN
33     <<loopstart>>
34     -- while loop execution
35     WHILE a < 20 LOOP
36     dbms_output.put_line ('value of a: ' || a);
37        a := a + 3;
38        IF a = 15 THEN
39           a := a + 3;
40           GOTO loopstart;
41        END IF;
42     END LOOP;
43  END;
44  /
```

```
Statement processed.
value of a: 1
value of a: 4
value of a: 7
value of a: 10
value of a: 13
value of a: 16
value of a: 19
```

NESTED

```
45  --NESTED
46  DECLARE
47     i number(3);
48     j number(3);
49  BEGIN
50     i := 2;
51     LOOP
52        j:= 2;
53        LOOP
54           exit WHEN ((mod(i, j) = 0) or (j = i));
55           j := j +1;
56        END LOOP;
57     IF (j = i ) THEN
58        dbms_output.put_line(i || ' is prime');
59     END IF;
60     i := i + 1;
61     exit WHEN i = 20;
62     END LOOP;
63  END;
64  /
```

```
Statement processed.
2 is prime
3 is prime
5 is prime
7 is prime
11 is prime
13 is prime
17 is prime
19 is prime
```

FUNCTIONS

SQL Worksheet

Clear Find Actions Save Run

```
--PROCEDURES
CREATE OR REPLACE PROCEDURE GuneetGreetsYou (p_name IN VARCHAR2)
IS
BEGIN
dbms_output.put_line ('Good Morning '|| p_name);
END;
/
EXEC GuneetGreetsYou ('Priyanka Maam');
```

```
Procedure created.


Statement processed.
Good Morning Priyanka Maam
```

PROCEDURES

```
DECLARE
   a number;
   b number;
   c number;
PROCEDURE findMin(x IN number, y IN number, z OUT number) IS
BEGIN
   IF x < y THEN
      z:= x;
   ELSE
      z:= y;
   END IF;
END;
BEGIN
   a:= 23;
   b:= 45;
   findMin(a, b, c);
   dbms_output.put_line(' Minimum of (23, 45) : ' || c);
END;
/
```

```
Statement processed.
 Minimum of (23, 45) : 23
```

LABELLING LOOP

```
DECLARE
   i number(1);
   j number(1);
BEGIN
   << outer_loop >>
   FOR i IN 1..3 LOOP
      << inner_loop >>
      FOR j IN 1..3 LOOP
         dbms_output.put_line('i is: '|| i || ' and j is: ' || j);
      END loop inner_loop;
   END loop outer_loop;
END;
/
```

```
Statement processed.
i is: 1 and j is: 1
i is: 1 and j is: 2
i is: 1 and j is: 3
i is: 2 and j is: 1
i is: 2 and j is: 2
i is: 2 and j is: 3
i is: 3 and j is: 1
i is: 3 and j is: 2
i is: 3 and j is: 3
```

PROCEDURES

```
30
31  DECLARE
32     a number;
33  PROCEDURE squareNum(x IN OUT number) IS
34  BEGIN
35    x := x * x;
36  END;
37  BEGIN
38     a:= 15;
39     squareNum(a);
40     dbms_output.put_line(' Square of (15): ' || a);
41  END;
42  /
43
```

```
Statement processed.
 Square of (15): 225
```

SALARY MID RANGE

```
DECLARE
mid_salary NUMBER;
emp_salary NUMBER;
emp_num NUMBER;

BEGIN
mid_salary:=5000;
emp_num:=2;
SELECT salary INTO emp_salary
FROM emp_details
WHERE emp_id=emp_num;

IF emp_salary> mid_salary THEN
UPDATE emp_details
SET salary=1.8*salary
WHERE emp_id=emp_num;
dbms_output.put_line('salary more than mid range');
ELSE
UPDATE emp_details
SET salary=mid_salary
WHERE emp_id=emp_num;
dbms_output.put_line('salary less than midrange');
END IF;
END;
```

```
Statement processed.
salary less than midrange
```

IMPLICIT CURSOR TO UPDATE SALARY

```
DECLARE

c_id customer.cust_id%type;
c_age customer.cust_age%type;
c_add customer.cust_add%type;
c_salary customer.cust_salary%type;
total_rows NUMBER;
BEGIN

UPDATE customer
SET cust_salary=cust_salary+5000;

IF SQL%NOTFOUND THEN
dbms_output.put_line('no customers updated');
ELSIF SQL%FOUND THEN
total_rows:=sql%rowcount;
dbms_output.put_line(total_rows||' customers updated');
END IF;
END;
/
```

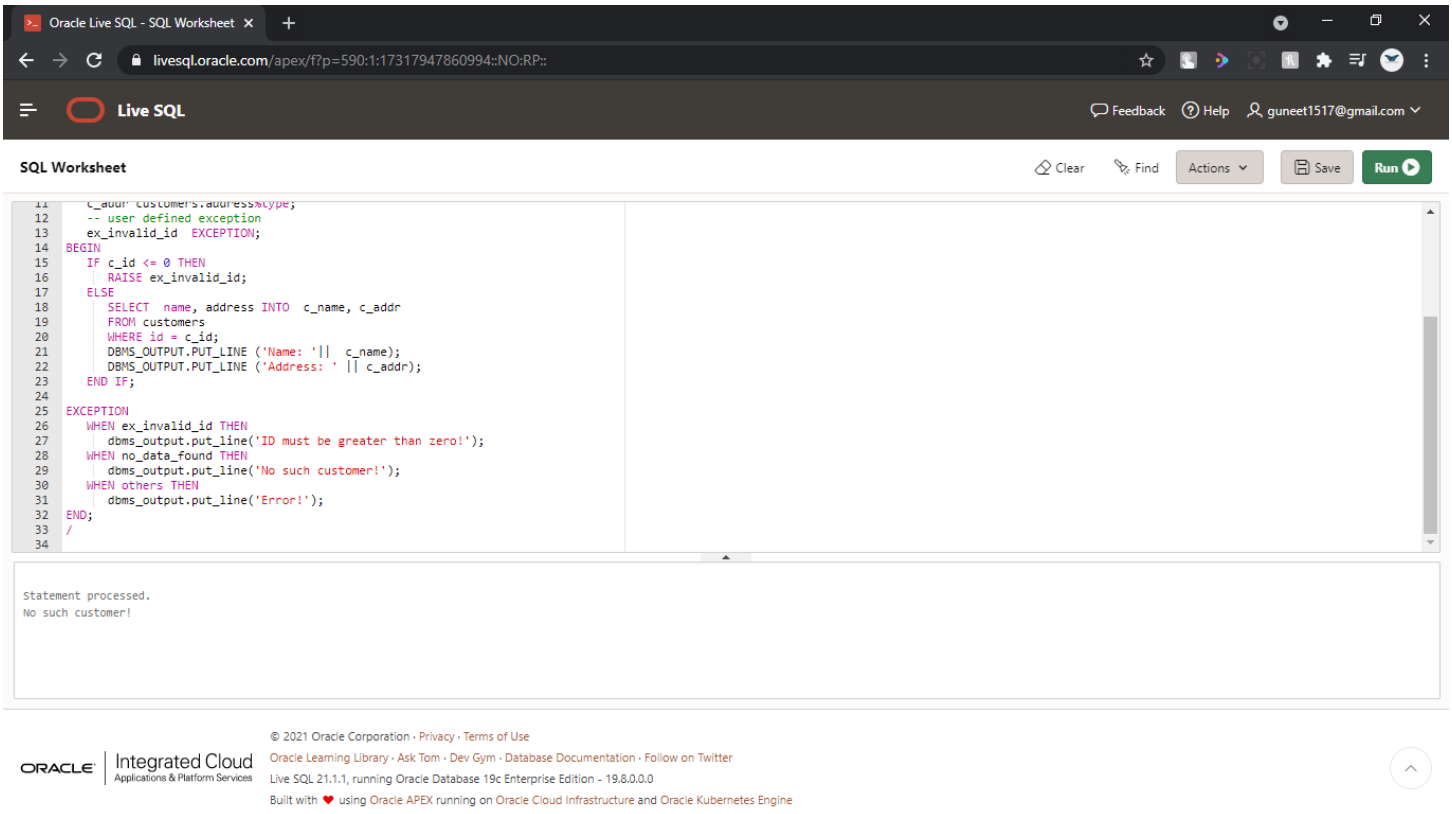
```
Statement processed.
3 customers updated
```

EXPLICIT CURSOR

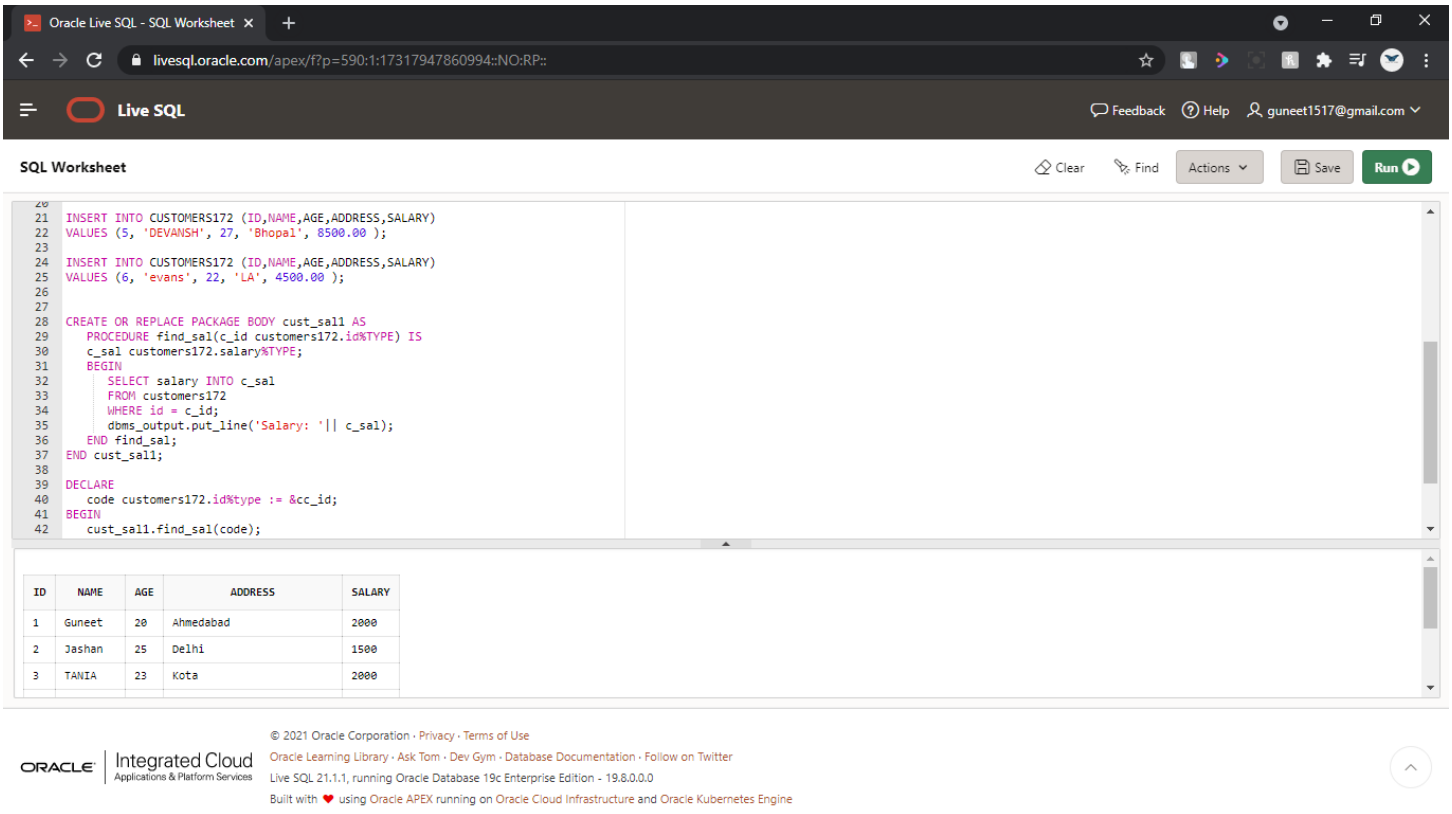
```
declare
cursor C1 is
SELECT emp_id, emp_name, emp_salary
FROM emp_detail;
empId emp_detail.emp_id%type;
empName emp_detail.emp_name%type;
empSalary emp_detail.emp_salary%type;
begin
open C1;
fetch C1 into empId, empName, empSalary;
while C1%found loop
dbms_output.put_line('Row Number '||
C1%rowcount || ' is '|| empId||' '||
empName||' '||empSalary);
fetch C1 into empId, empName, empSalary;
end loop;
close C1;
end;
```

```
Statement processed.
Row Number 1 is 1 PALAK 100000
Row Number 2 is 2 SALONI 100000
Row Number 3 is 3 GUNEET 100000
Row Number 4 is 4 JASHAN 100000
```

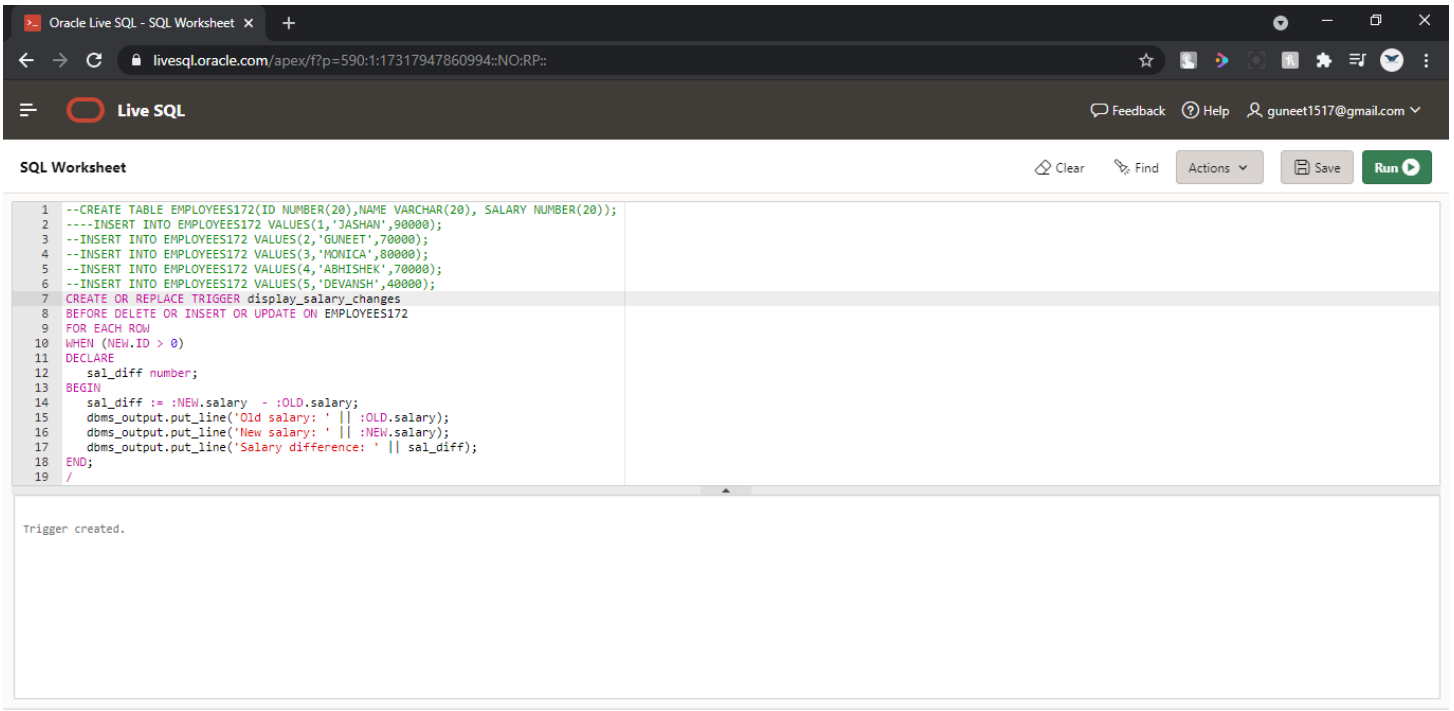
EXCEPTION HANDLING



PACKAGES



TRIGGERS


Oracle Live SQL - SQL Worksheet
livesql.oracle.com/apex/f?p=590:1:17317947860994::NO:RP::
Live SQL
Feedback
Help
guneet1517@gmail.com
SQL Worksheet
Clear
Find
Actions
Save
Run
--CREATE TABLE EMPLOYEES172(ID NUMBER(20),NAME VARCHAR(20), SALARY NUMBER(20));
----INSERT INTO EMPLOYEES172 VALUES(1,'JASHAN',90000);
--INSERT INTO EMPLOYEES172 VALUES(2,'GUNEET',70000);
--INSERT INTO EMPLOYEES172 VALUES(3,'MONICA',80000);
--INSERT INTO EMPLOYEES172 VALUES(4,'ABHISHEK',70000);
--INSERT INTO EMPLOYEES172 VALUES(5,'DEVANSH',40000);
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON EMPLOYEES172
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.salary - :OLD.salary;
dbms_output.put_line('Old salary: ' || :OLD.salary);
dbms_output.put_line('New salary: ' || :NEW.salary);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
Trigger created.
ORACLE
Integrated Cloud
Applications & Platform Services
© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.1.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine