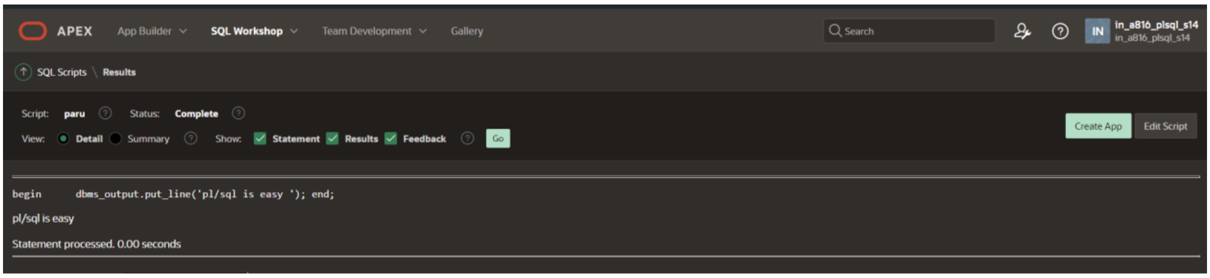
PL/SQL

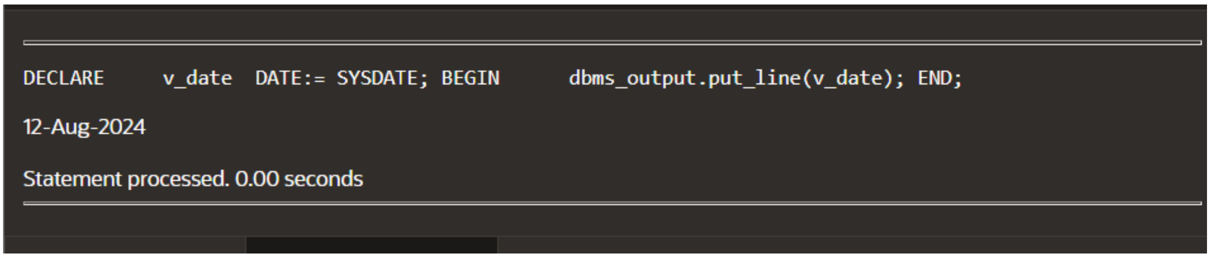


Script: paru Status: Complete

View: ● Detail ● Summary Show: ☑ Statement ☑ Results ☑ Feedback

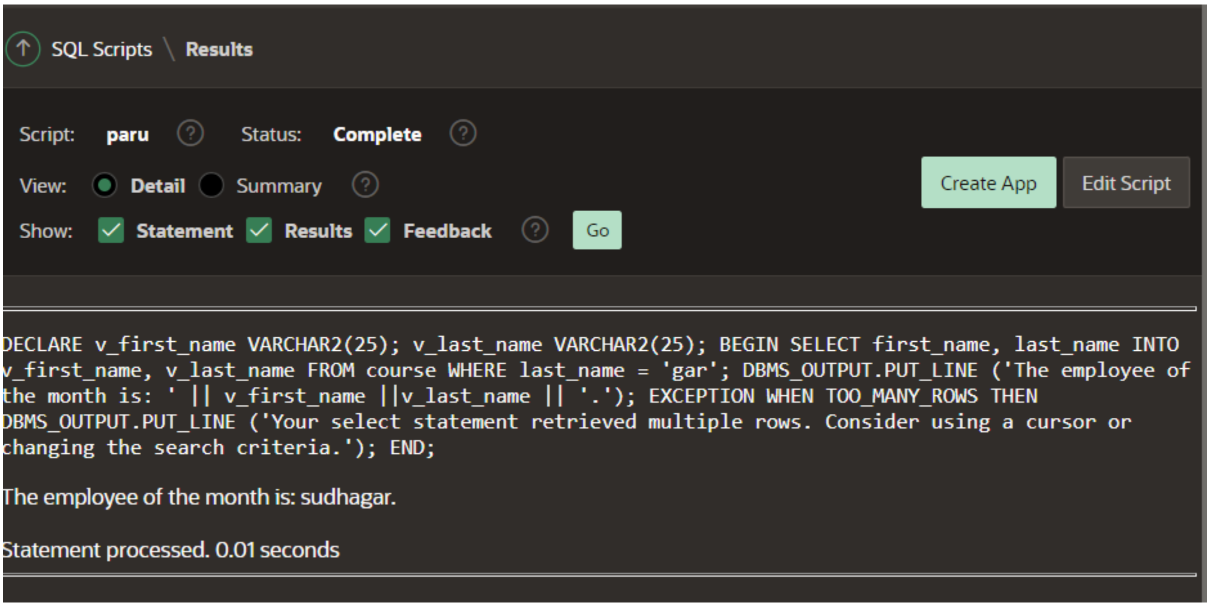
```
begin      dbms_output.put_line('pl/sql is easy '); end;
pl/sql is easy
Statement processed. 0.00 seconds
```

```
begin
   dbms_output.put_line('pl/sql is easy ');
end;
```



```
DECLARE      v_date  DATE:= SYSDATE; BEGIN      dbms_output.put_line(v_date); END;
12-Aug-2024
Statement processed. 0.00 seconds
```

```
DECLARE
   v_date  DATE:= SYSDATE;
BEGIN
   dbms_output.put_line(v_date);
END;
```

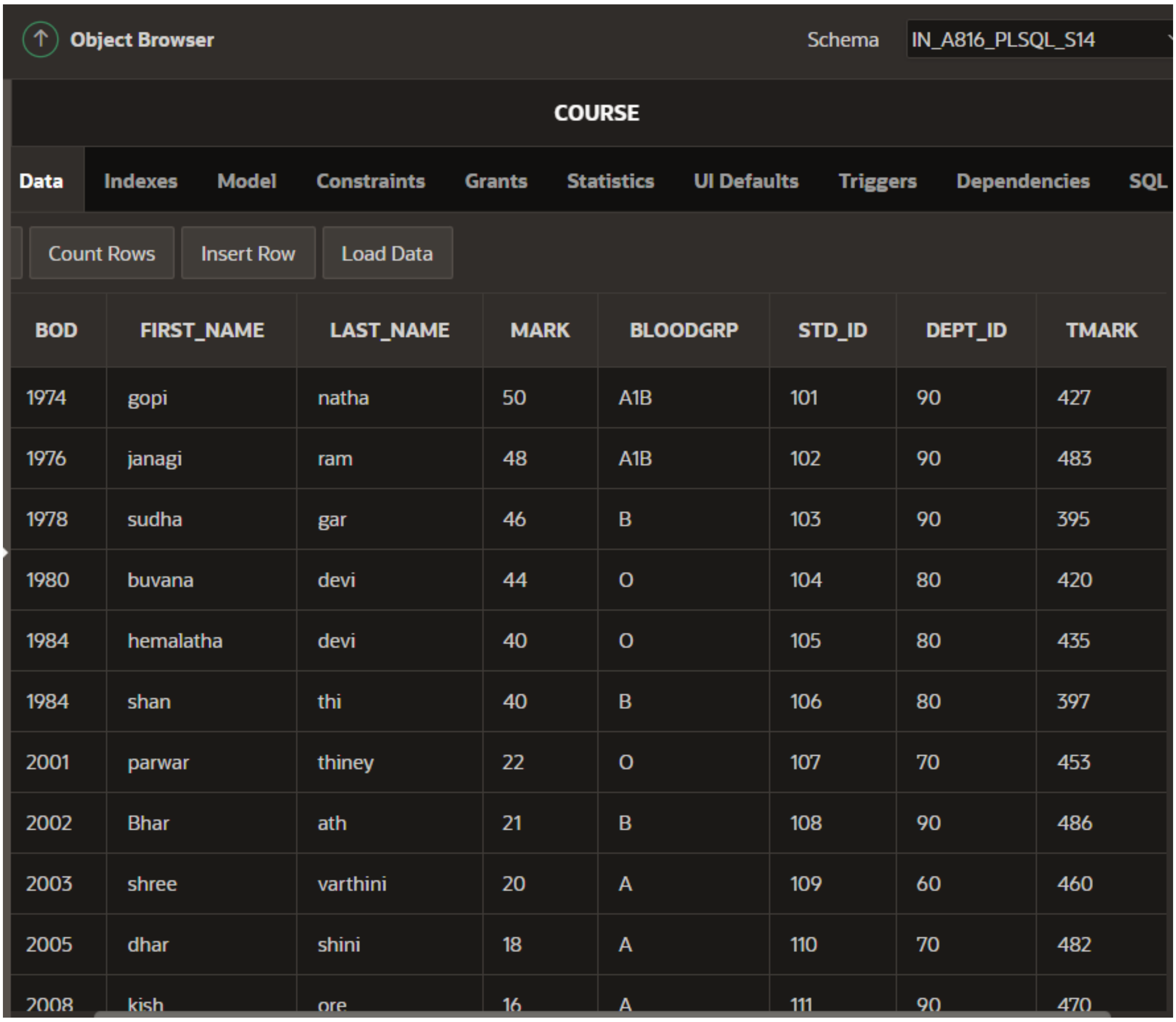


Script: paru Status: Complete

View: ● Detail ● Summary

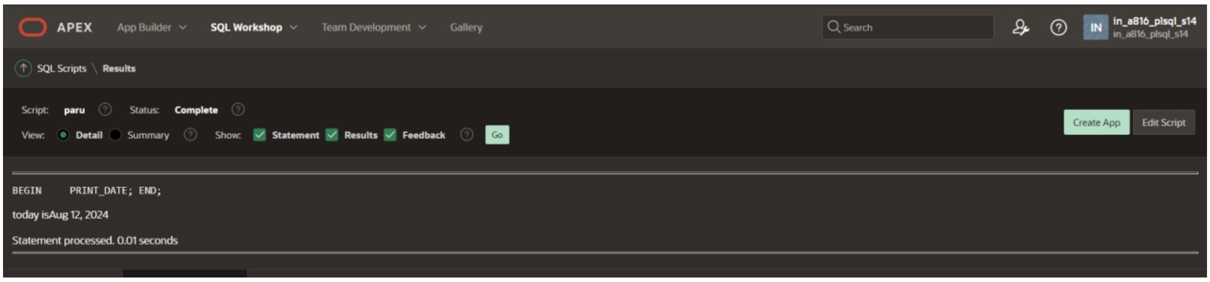
Show: ☑ Statement ☑ Results ☑ Feedback

```
DECLARE v_first_name VARCHAR2(25); v_last_name VARCHAR2(25); BEGIN SELECT first_name, last_name INTO
v_first_name, v_last_name FROM course WHERE last_name = 'gar'; DBMS_OUTPUT.PUT_LINE ('The employee of
the month is: ' || v_first_name ||v_last_name || '.'); EXCEPTION WHEN TOO_MANY_ROWS THEN
DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved multiple rows. Consider using a cursor or
changing the search criteria.'); END;

The employee of the month is: sudhagar.

Statement processed. 0.01 seconds
```



BOD	FIRST_NAME	LAST_NAME	MARK	BLOODGRP	STD_ID	DEPT_ID	TMARK
1974	gopi	natha	50	A1B	101	90	427
1976	janagi	ram	48	A1B	102	90	483
1978	sudha	gar	46	B	103	90	395
1980	buvana	devi	44	O	104	80	420
1984	hemalatha	devi	40	O	105	80	435
1984	shan	thi	40	B	106	80	397
2001	parwar	thiney	22	O	107	70	453
2002	Bhar	ath	21	B	108	90	486
2003	shree	varthini	20	A	109	60	460
2005	dhar	shini	18	A	110	70	482
2008	kish	ore	16	A	111	90	470

```
DECLARE

v_first_name VARCHAR2(25);

v_last_name VARCHAR2(25);

BEGIN

SELECT first_name, last_name

INTO v_first_name, v_last_name

FROM course

WHERE last_name = 'gar';

DBMS_OUTPUT.PUT_LINE ('The employee of the month is: '

|| v_first_name ||v_last_name || '.');

EXCEPTION

WHEN TOO_MANY_ROWS THEN

DBMS_OUTPUT.PUT_LINE ('Your select statement retrieved
multiple rows. Consider using a cursor or changing

the search criteria.');

END;
```



Script: paru   Status: Complete

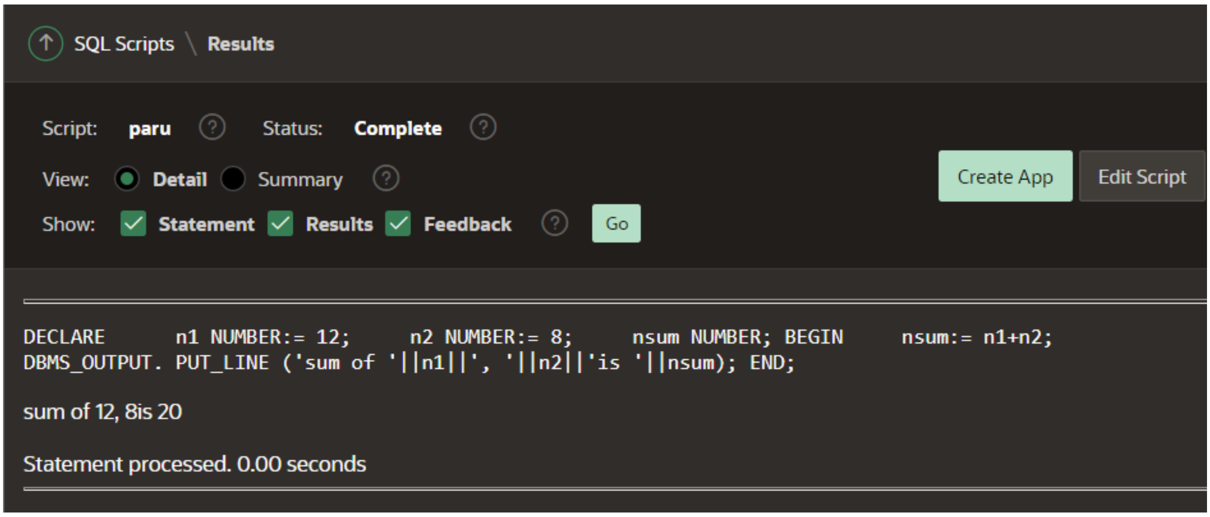
View: ● Detail ● Summary   Show: ☑ Statement ☑ Results ☑ Feedback

```
BEGIN     PRINT_DATE; END;
```

today isAug 12, 2024

Statement processed. 0.01 seconds

```
CREATE OR REPLACE PROCEDURE print_date IS
   v_date VARCHAR2(30);
BEGIN
   SELECT TO_CHAR(SYSDATE,'Mon DD, YYYY')
      INTO v_date
      FROM DUAL;
   DBMS_OUTPUT.PUT_LINE('today is'||v_date);
END;


BEGIN
   PRINT_DATE;
END;
```



Script: paru   Status: Complete

View: ● Detail ● Summary

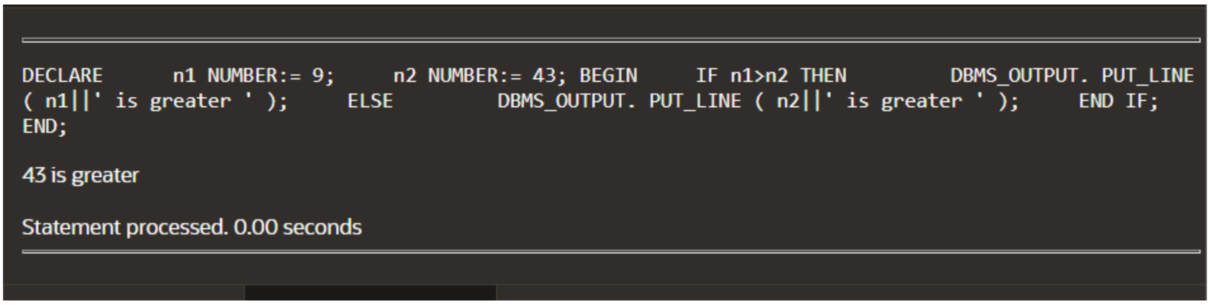
Show: ☑ Statement ☑ Results ☑ Feedback

```
DECLARE      n1 NUMBER:= 12;     n2 NUMBER:= 8;     nsum NUMBER; BEGIN     nsum:= n1+n2;
DBMS_OUTPUT. PUT_LINE ('sum of '||n1||', '||n2||'is '||nsum); END;
```

sum of 12, 8is 20

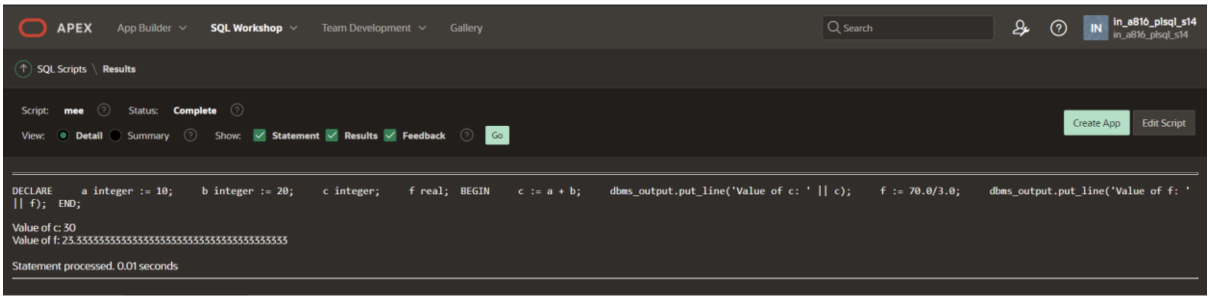
Statement processed. 0.00 seconds

```
DECLARE
   n1 NUMBER:= 12;
   n2 NUMBER:= 8;
   nsum NUMBER;
BEGIN
   nsum:= n1+n2;
   DBMS_OUTPUT. PUT_LINE ('sum of '||n1||', '||n2||'is '||nsum);
END;
```



```
DECLARE      n1 NUMBER:= 9;      n2 NUMBER:= 43; BEGIN      IF n1>n2 THEN          DBMS_OUTPUT. PUT_LINE
( n1||' is greater ' );      ELSE          DBMS_OUTPUT. PUT_LINE ( n2||' is greater ' );      END IF;
END;

43 is greater

Statement processed. 0.00 seconds
```

```
DECLARE
   n1 NUMBER:= 9;
   n2 NUMBER:= 43;
BEGIN
   IF n1>n2 THEN
      DBMS_OUTPUT. PUT_LINE ( n1||' is greater ' );
   ELSE
      DBMS_OUTPUT. PUT_LINE ( n2||' is greater ' );
   END IF;
END;
```

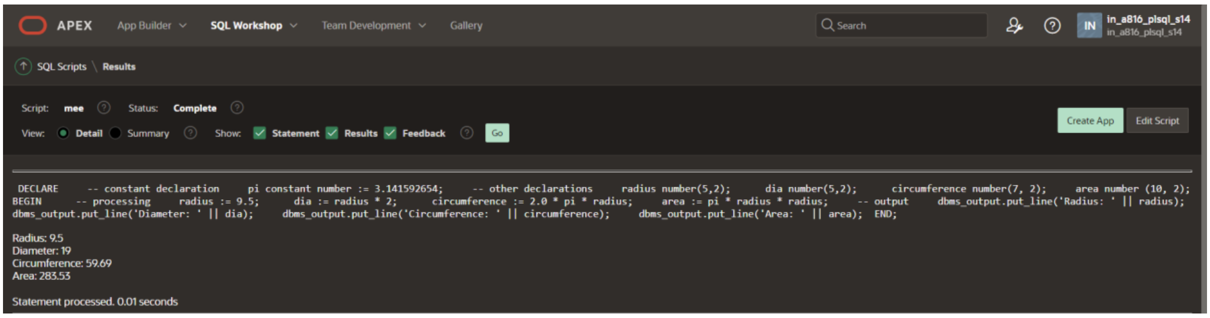


Script: mee   Status: Complete

View: ● Detail ● Summary   Show: ☑ Statement ☑ Results ☑ Feedback

```
DECLARE     a integer := 10;     b integer := 20;     c integer;     f real;  BEGIN     c := a + b;     dbms_output.put_line('Value of c: ' || c);     f := 70.0/3.0;     dbms_output.put_line('Value of f: '
|| f);  END;

Value of c: 30
Value of f: 23.33333333333333333333333333333333333333

Statement processed. 0.01 seconds
```

```
DECLARE
  a integer := 10;
  b integer := 20;
  c integer;
  f real;
BEGIN
  c := a + b;
  dbms_output.put_line('Value of c: ' || c);
  f := 70.0/3.0;
  dbms_output.put_line('Value of f: ' || f);
END;
```

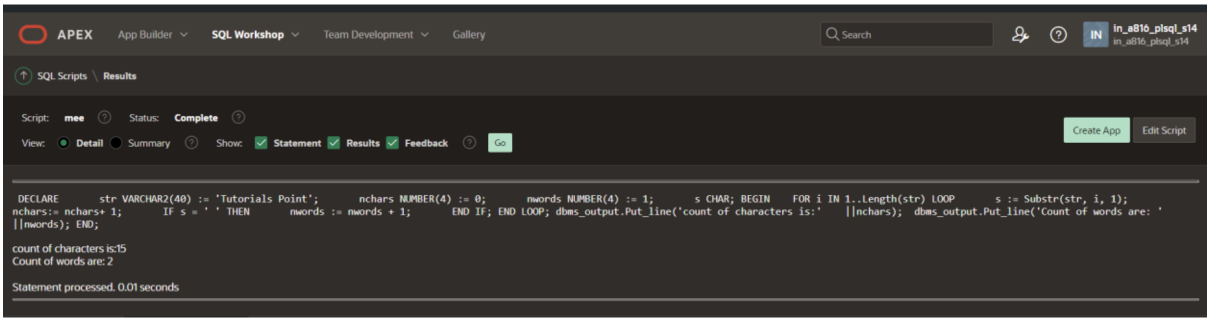


Script: mee Status: Complete

View: ● Detail ● Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

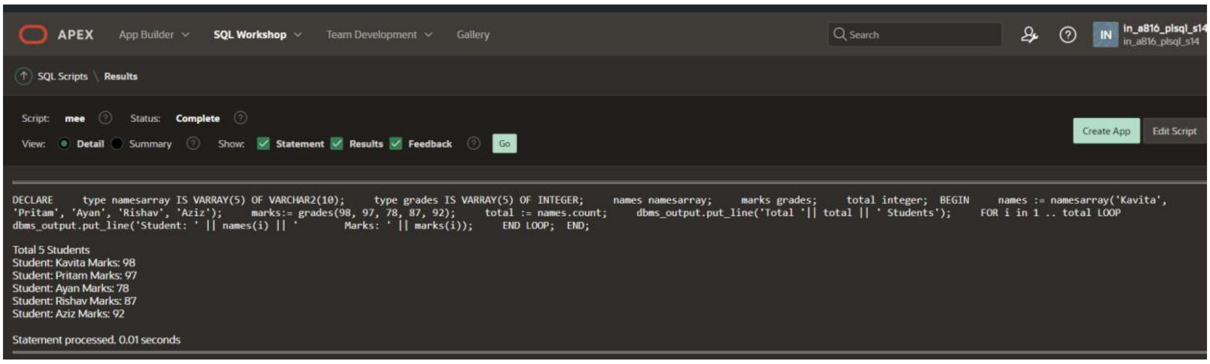
```
DECLARE     -- constant declaration     pi constant number := 3.141592654;     -- other declarations     radius number(5,2);     dia number(5,2);     circumference number(7, 2);     area number (10, 2);
BEGIN     -- processing     radius := 9.5;     dia := radius * 2;     circumference := 2.0 * pi * radius;     area := pi * radius * radius;     -- output     dbms_output.put_line('Radius: ' || radius);
dbms_output.put_line('Diameter: ' || dia);     dbms_output.put_line('Circumference: ' || circumference);     dbms_output.put_line('Area: ' || area);  END;

Radius: 9.5
Diameter: 19
Circumference: 59.69
Area: 283.53

Statement processed. 0.01 seconds
```

```
DECLARE
  -- constant declaration
  pi constant number := 3.141592654;
  -- other declarations
  radius number(5,2);
  dia number(5,2);
  circumference number(7, 2);
  area number (10, 2);
BEGIN
  -- processing
  radius := 9.5;
  dia := radius * 2;
  circumference := 2.0 * pi * radius;
  area := pi * radius * radius;
  -- output
  dbms_output.put_line('Radius: ' || radius);
  dbms_output.put_line('Diameter: ' || dia);
  dbms_output.put_line('Circumference: ' || circumference);
  dbms_output.put_line('Area: ' || area);
END;
```



```
 DECLARE       str VARCHAR2(40) := 'Tutorials Point';       nchars NUMBER(4) := 0;       nwords NUMBER(4) := 1;       s CHAR; BEGIN    FOR i IN 1..Length(str) LOOP       s := Substr(str, i, 1);
nchars:= nchars+ 1;       IF s = ' ' THEN       nwords := nwords + 1;       END IF; END LOOP; dbms_output.Put_line('count of characters is:'    ||nchars);  dbms_output.Put_line('Count of words are: '
||nwords); END;

count of characters is:15
Count of words are: 2

Statement processed. 0.01 seconds
```

```
DECLARE
    str VARCHAR2(40) := 'Tutorials Point';
    nchars NUMBER(4) := 0;
    nwords NUMBER(4) := 1;
    s CHAR;
BEGIN
  FOR i IN 1..Length(str) LOOP
    s := Substr(str, i, 1);
    nchars:= nchars+ 1;
    IF s = ' ' THEN
    nwords := nwords + 1;
    END IF;
END LOOP;
dbms_output.Put_line('count of characters is:'
  ||nchars);


dbms_output.Put_line('Count of words are: '
  ||nwords);
END;
```

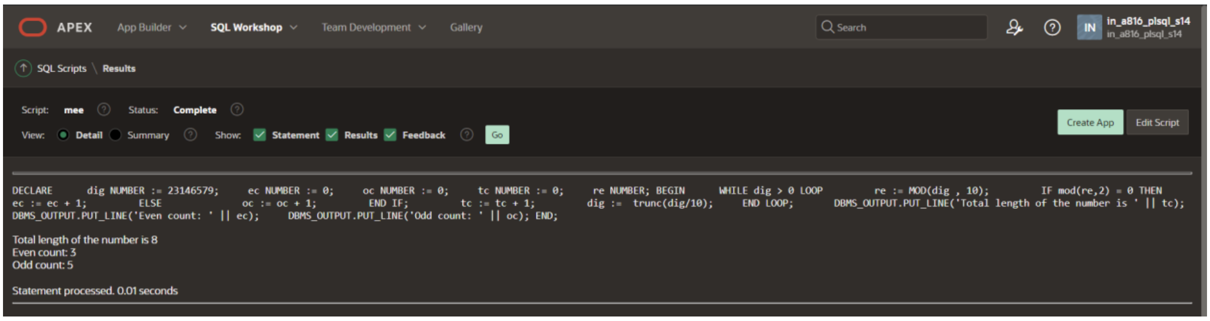


Script: mee Status: Complete
View: ● Detail ● Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

```
DECLARE     type namesarray IS VARRAY(5) OF VARCHAR2(10);     type grades IS VARRAY(5) OF INTEGER;     names namesarray;     marks grades;     total integer;  BEGIN     names := namesarray('Kavita',
'Pritam', 'Ayan', 'Rishav', 'Aziz');     marks:= grades(98, 97, 78, 87, 92);     total := names.count;     dbms_output.put_line('Total '|| total || ' Students');     FOR i in 1 .. total LOOP
dbms_output.put_line('Student: ' || names(i) || '         Marks: ' || marks(i));     END LOOP;  END;
```

```
Total 5 Students
Student: Kavita Marks: 98
Student: Pritam Marks: 97
Student: Ayan Marks: 78
Student: Rishav Marks: 87
Student: Aziz Marks: 92

Statement processed. 0.01 seconds
```

```
DECLARE
  type namesarray IS VARRAY(5) OF VARCHAR2(10);
  type grades IS VARRAY(5) OF INTEGER;
  names namesarray;
  marks grades;
  total integer;
BEGIN
  names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
  marks:= grades(98, 97, 78, 87, 92);
  total := names.count;
  dbms_output.put_line('Total '|| total || ' Students');
  FOR i in 1 .. total LOOP
    dbms_output.put_line('Student: ' || names(i) || '
    Marks: ' || marks(i));
  END LOOP;
END;
```

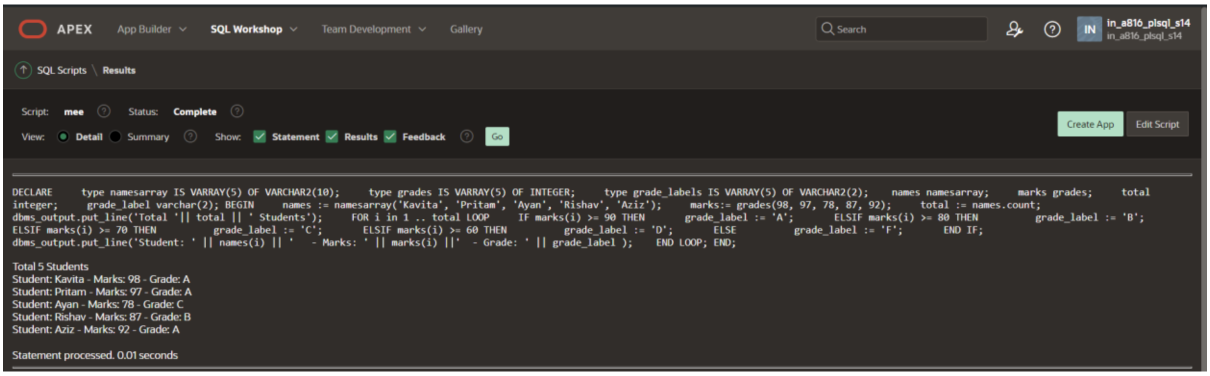


Script: mee Status: Complete
View: ● Detail ● Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

```
DECLARE      dig NUMBER := 23146579;     ec NUMBER := 0;     oc NUMBER := 0;     tc NUMBER := 0;     re NUMBER; BEGIN     WHILE dig > 0 LOOP         re := MOD(dig , 10);         IF mod(re,2) = 0 THEN
ec := ec + 1;         ELSE             oc := oc + 1;         END IF;         tc := tc + 1;         dig :=  trunc(dig/10);     END LOOP;     DBMS_OUTPUT.PUT_LINE('Total length of the number is ' || tc);
DBMS_OUTPUT.PUT_LINE('Even count: ' || ec);     DBMS_OUTPUT.PUT_LINE('Odd count: ' || oc); END;
```

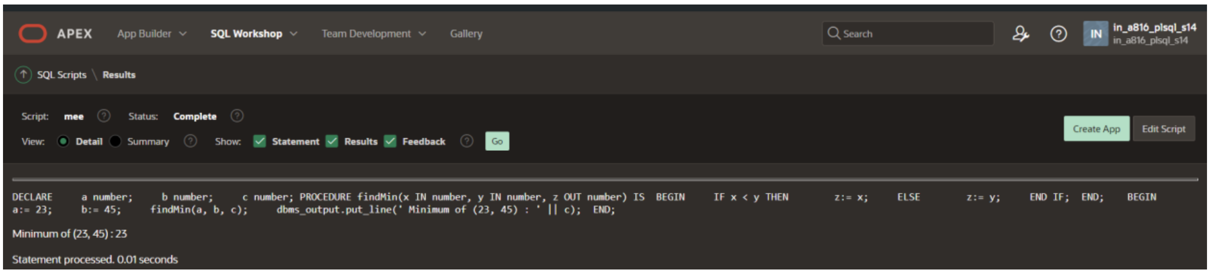
```
Total length of the number is 8
Even count: 3
Odd count: 5

Statement processed. 0.01 seconds
```

```
DECLARE
   dig NUMBER := 23146579;
   ec NUMBER := 0;
   oc NUMBER := 0;
   tc NUMBER := 0;
   re NUMBER;
BEGIN
   WHILE dig > 0 LOOP
      re := MOD(dig , 10);
      IF MOD (re,2) = 0 THEN
         ec := ec + 1;
      ELSE
         oc := oc + 1;
      END IF;
      tc := tc + 1;
      dig :=  TRUNC(dig/10);
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('Total length of the number is ' || tc);
   DBMS_OUTPUT.PUT_LINE('Even count: ' || ec);
   DBMS_OUTPUT.PUT_LINE('Odd count: ' || oc);
END;
/
```



```
DECLARE     type namesarray IS VARRAY(5) OF VARCHAR2(10);     type grades IS VARRAY(5) OF INTEGER;     type grade_labels IS VARRAY(5) OF VARCHAR2(2);     names namesarray;     marks grades;     total
integer;     grade_label varchar(2); BEGIN     names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');     marks:= grades(98, 97, 78, 87, 92);     total := names.count;
dbms_output.put_line('Total '|| total || ' Students');     FOR i in 1 .. total LOOP     IF marks(i) >= 90 THEN        grade_label := 'A';        ELSIF marks(i) >= 80 THEN          grade_label := 'B';
ELSIF marks(i) >= 70 THEN          grade_label := 'C';        ELSIF marks(i) >= 60 THEN          grade_label := 'D';        ELSE          grade_label := 'F';        END IF;
dbms_output.put_line('Student: ' || names(i) || '   - Marks: ' || marks(i) ||'  - Grade: ' || grade_label );    END LOOP; END;

Total 5 Students
Student: Kavita - Marks: 98 - Grade: A
Student: Pritam - Marks: 97 - Grade: A
Student: Ayan - Marks: 78 - Grade: C
Student: Rishav - Marks: 87 - Grade: B
Student: Aziz - Marks: 92 - Grade: A

Statement processed. 0.01 seconds
```

```
DECLARE
  type namesarray IS VARRAY(5) OF VARCHAR2(10);
  type grades IS VARRAY(5) OF INTEGER;
  type grade_labels IS VARRAY(5) OF VARCHAR2(2);
  names namesarray;
  marks grades;
  total integer;
  grade_label varchar(2);
BEGIN
  names := namesarray('Kavita', 'Pritam', 'Ayan', 'Rishav', 'Aziz');
  marks:= grades(98, 97, 78, 87, 92);
  total := names.count;
  dbms_output.put_line('Total '|| total || ' Students');
  FOR i in 1 .. total LOOP
  IF marks(i) >= 90 THEN
    grade_label := 'A';
    ELSIF marks(i) >= 80 THEN
      grade_label := 'B';
    ELSIF marks(i) >= 70 THEN
      grade_label := 'C';
    ELSIF marks(i) >= 60 THEN
      grade_label := 'D';
    ELSE
      grade_label := 'F';
    END IF;
    dbms_output.put_line('Student: ' || names(i) || '   - Marks: ' || marks(i) ||'  - Grade: ' ||
grade_label );
 END LOOP;
END;
```

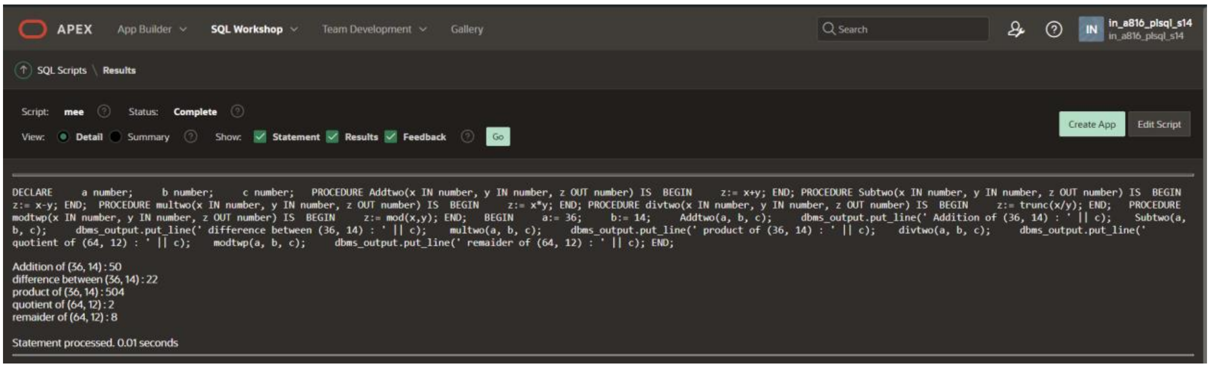


Script: mee   Status: Complete

View: ● Detail ● Summary   Show: ☑ Statement ☑ Results ☑ Feedback

```
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;

Minimum of (23, 45) : 23

Statement processed. 0.01 seconds
```

```
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;
```

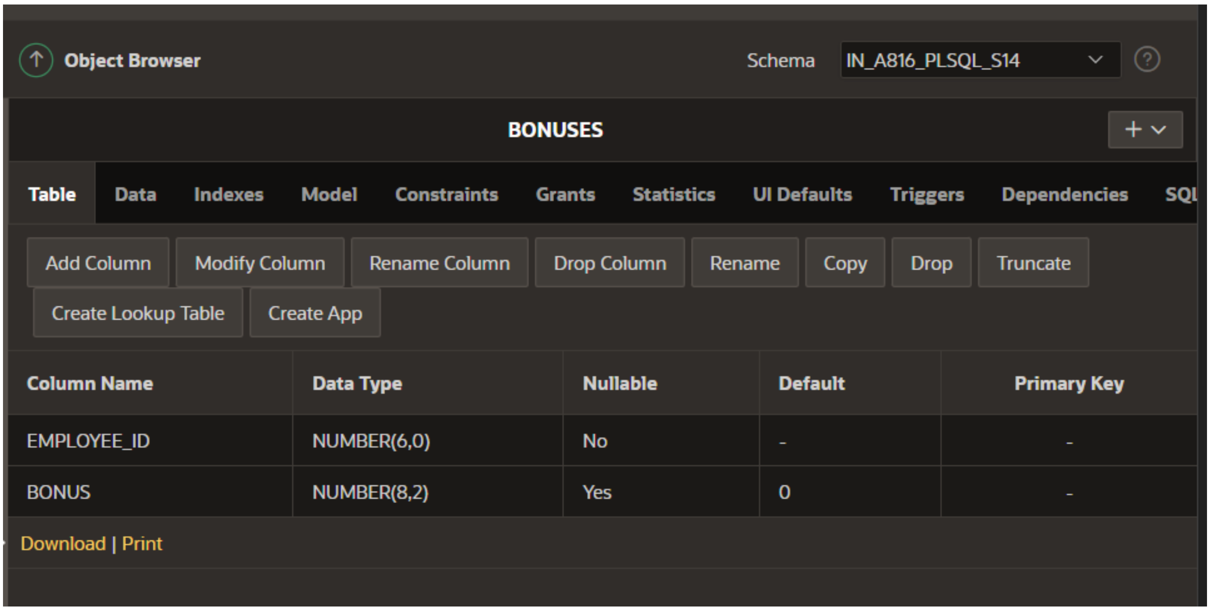


Script: mee Status: Complete

View: ● Detail ● Summary Show: ☑ Statement ☑ Results ☑ Feedback Go

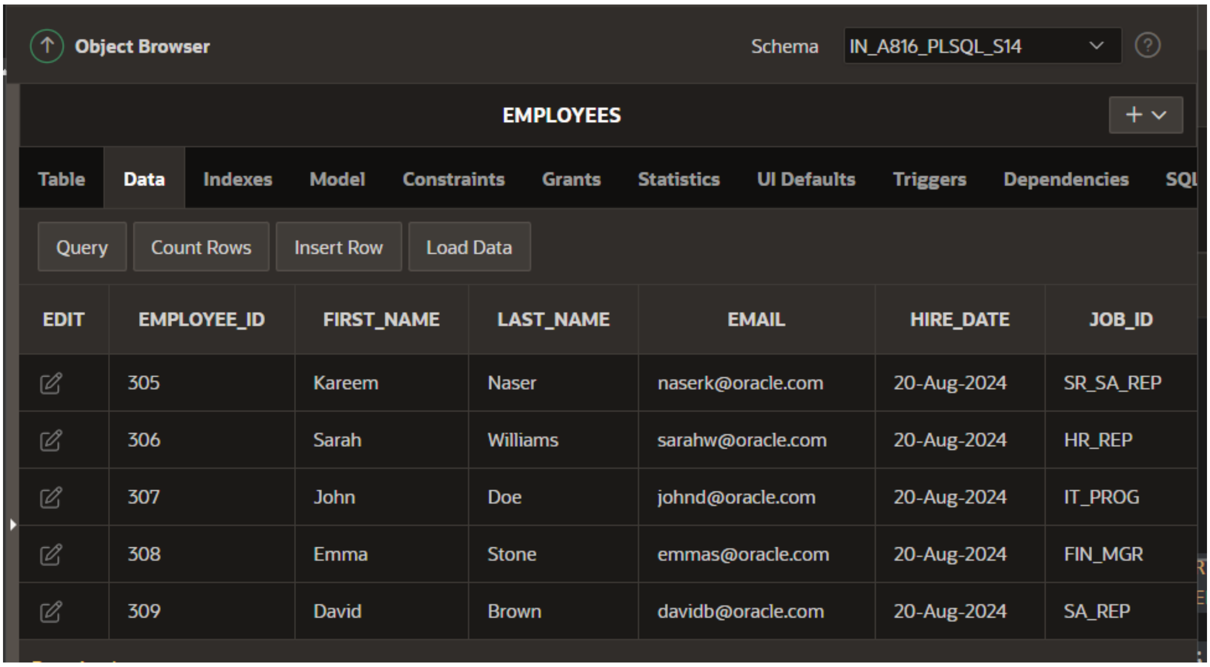
```
DECLARE     a number;     b number;     c number;   PROCEDURE Addtwo(x IN number, y IN number, z OUT number) IS  BEGIN     z:= x+y; END; PROCEDURE Subtwo(x IN number, y IN number, z OUT number) IS  BEGIN
z:= x-y; END;  PROCEDURE multwo(x IN number, y IN number, z OUT number) IS  BEGIN     z:= x*y; END; PROCEDURE divtwo(x IN number, y IN number, z OUT number) IS  BEGIN     z:= trunc(x/y); END;   PROCEDURE
modtwp(x IN number, y IN number, z OUT number) IS  BEGIN     z:= mod(x,y); END;   BEGIN     a:= 36;     b:= 14;     Addtwo(a, b, c);     dbms_output.put_line(' Addition of (36, 14) : ' || c);    Subtwo(a,
b, c);     dbms_output.put_line(' difference between (36, 14) : ' || c);    multwo(a, b, c);     dbms_output.put_line(' product of (36, 14) : ' || c);    divtwo(a, b, c);     dbms_output.put_line('
quotient of (64, 12) : ' || c);    modtwp(a, b, c);     dbms_output.put_line(' remaider of (64, 12) : ' || c); END;

Addition of (36, 14) : 50
difference between (36, 14) : 22
product of (36, 14) : 504
quotient of (64, 12) : 2
remaider of (64, 12) : 8

Statement processed. 0.01 seconds
```

```
DECLARE
  a number;
  b number;
  c number;

PROCEDURE Addtwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= x+y;
END;
PROCEDURE Subtwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= x-y;
END;
PROCEDURE multwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= x*y;
END;
PROCEDURE divtwo(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= trunc(x/y);
END;
PROCEDURE modtwp(x IN number, y IN number, z OUT number) IS
BEGIN
  z:= mod(x,y);
END;
BEGIN
  a:= 36;
  b:= 14;
  Addtwo(a, b, c);
  dbms_output.put_line(' Addition of (36, 14) : ' || c);
  Subtwo(a, b, c);
  dbms_output.put_line(' difference between (36, 14) : ' || c);
  multwo(a, b, c);
  dbms_output.put_line(' product of (36, 14) : ' || c);
  divtwo(a, b, c);
  dbms_output.put_line(' quotient of (64, 12) : ' || c);
  modtwp(a, b, c);
  dbms_output.put_line(' remaider of (64, 12) : ' || c);
END;
```



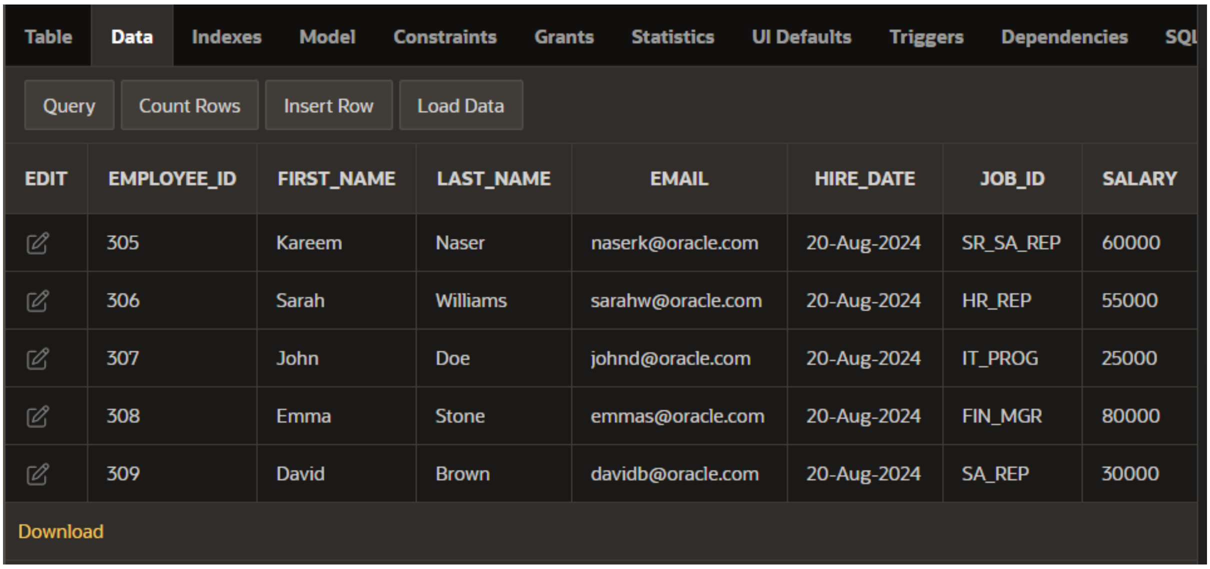
Column Name	Data Type	Nullable	Default	Primary Key
EMPLOYEE_ID	NUMBER(6,0)	No	-	-
BONUS	NUMBER(8,2)	Yes	0	-

```
CREATE TABLE bonuses (
employee_id NUMBER(6,0) NOT NULL,
bonus NUMBER(8,2) DEFAULT 0
);
```



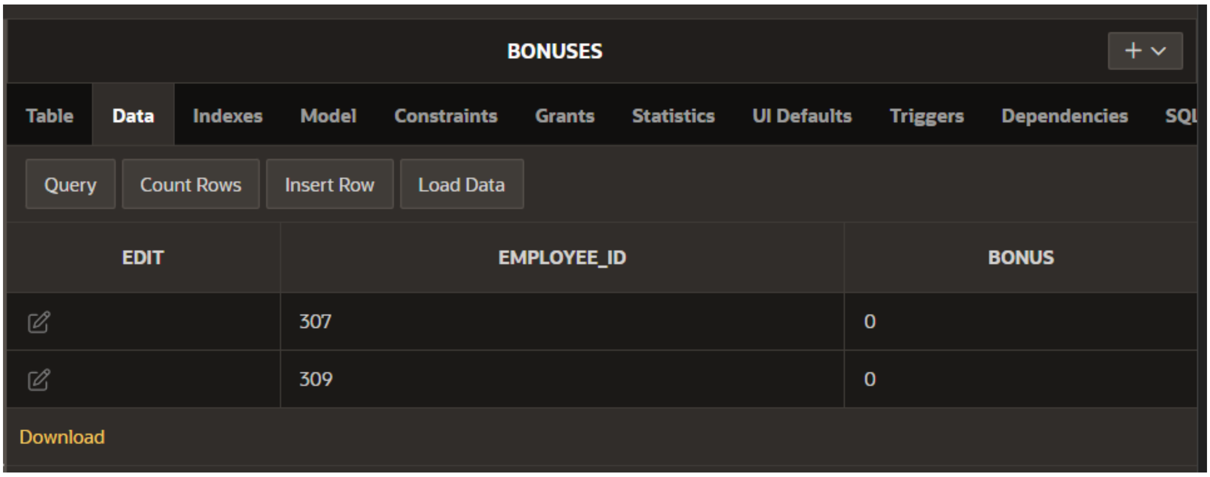
EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID
	305	Kareem	Naser	naserk@oracle.com	20-Aug-2024	SR_SA_REP
	306	Sarah	Williams	sarahw@oracle.com	20-Aug-2024	HR_REP
	307	John	Doe	johnd@oracle.com	20-Aug-2024	IT_PROG
	308	Emma	Stone	emmas@oracle.com	20-Aug-2024	FIN_MGR
	309	David	Brown	davidb@oracle.com	20-Aug-2024	SA_REP

```
CREATE TABLE employees (
   employee_id NUMBER(6) PRIMARY KEY,
   first_name VARCHAR(20),
   last_name VARCHAR2(25),
   email VARCHAR2(50) UNIQUE,
   hire_date DATE,
   job_id VARCHAR2(10)
);
INSERT INTO employees VALUES (305, 'Kareem', 'Naser', 'naserk@oracle.com', SYSDATE,
'SR_SA_REP');
INSERT INTO employees VALUES (306, 'Sarah', 'Williams', 'sarahw@oracle.com', SYSDATE, 'HR_REP');
INSERT INTO employees VALUES (307, 'John', 'Doe', 'johnd@oracle.com', SYSDATE, 'IT_PROG');
INSERT INTO employees VALUES (308, 'Emma', 'Stone', 'emmas@oracle.com', SYSDATE, 'FIN_MGR');
INSERT INTO employees VALUES (309, 'David', 'Brown', 'davidb@oracle.com', SYSDATE, 'SA_REP');
```



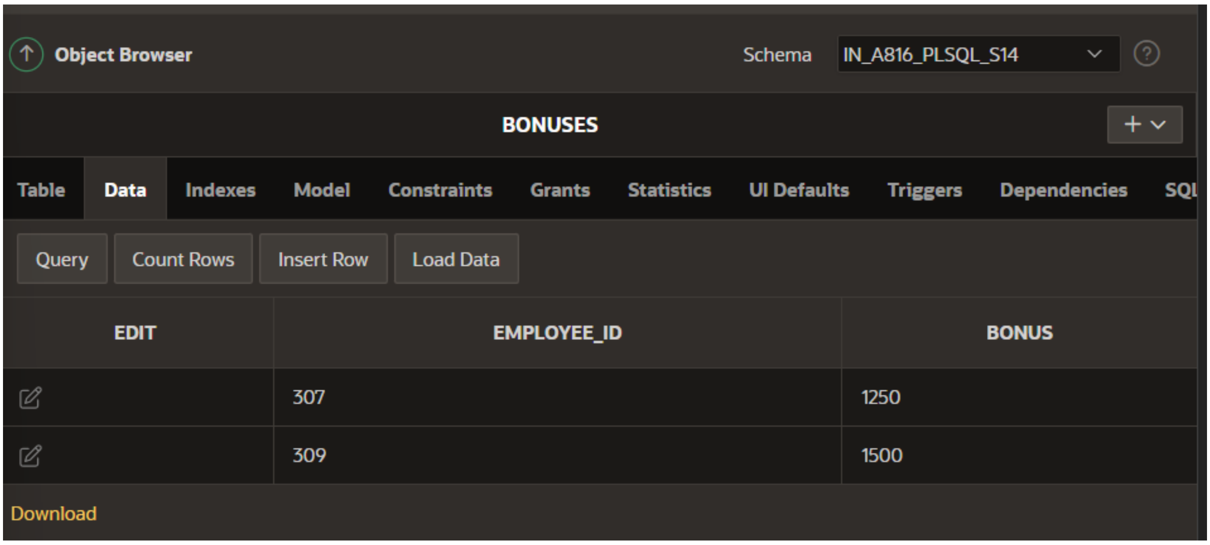
EDIT	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	SALARY
✎	305	Kareem	Naser	naserk@oracle.com	20-Aug-2024	SR_SA_REP	60000
✎	306	Sarah	Williams	sarahw@oracle.com	20-Aug-2024	HR_REP	55000
✎	307	John	Doe	johnd@oracle.com	20-Aug-2024	IT_PROG	25000
✎	308	Emma	Stone	emmas@oracle.com	20-Aug-2024	FIN_MGR	80000
✎	309	David	Brown	davidb@oracle.com	20-Aug-2024	SA_REP	30000

```
ALTER TABLE employees
ADD salary NUMBER(10, 2);

UPDATE employees
SET salary = 60000
WHERE employee_id = 305;

UPDATE employees
SET salary = 55000
WHERE employee_id = 306;

UPDATE employees
SET salary = 25000
WHERE employee_id = 307;

UPDATE employees
SET salary = 80000
WHERE employee_id = 308;

UPDATE employees
SET salary = 30000
WHERE employee_id = 309;
```



EDIT	EMPLOYEE_ID	BONUS
✎	307	0
✎	309	0

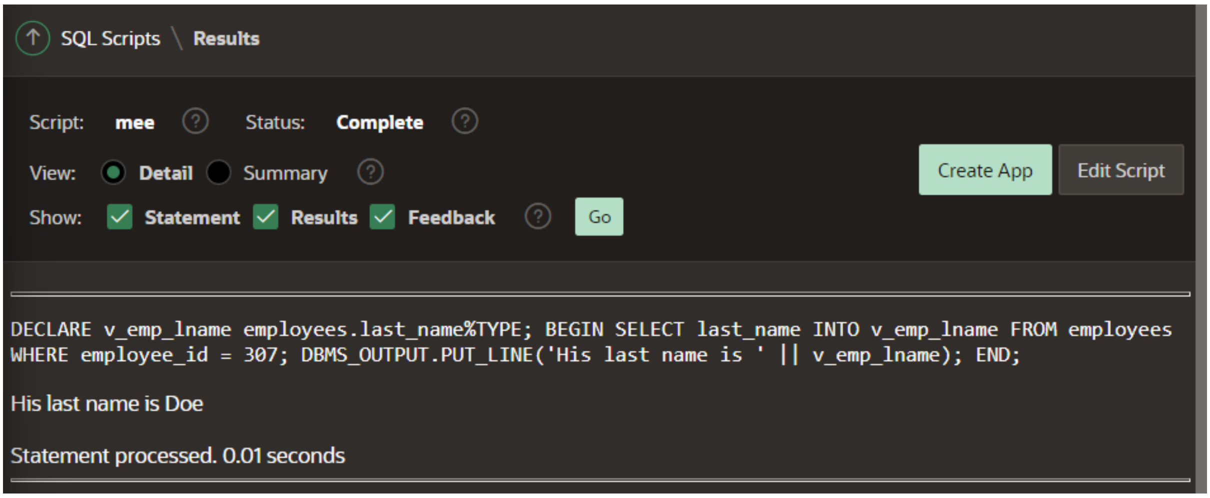
```
INSERT INTO bonuses(employee_id)

(SELECT employee_id

FROM employees

WHERE salary < 50000

);
```



Schema IN_A816_PLSQL_S14

EDIT	EMPLOYEE_ID	BONUS
✎	307	1250
✎	309	1500

```
MERGE INTO bonuses b

 USING employees e ON

 (b.employee_id = e.employee_id)

WHEN MATCHED

THEN UPDATE

SET b.bonus = e.salary * .05;
```



Script: mee Status: Complete

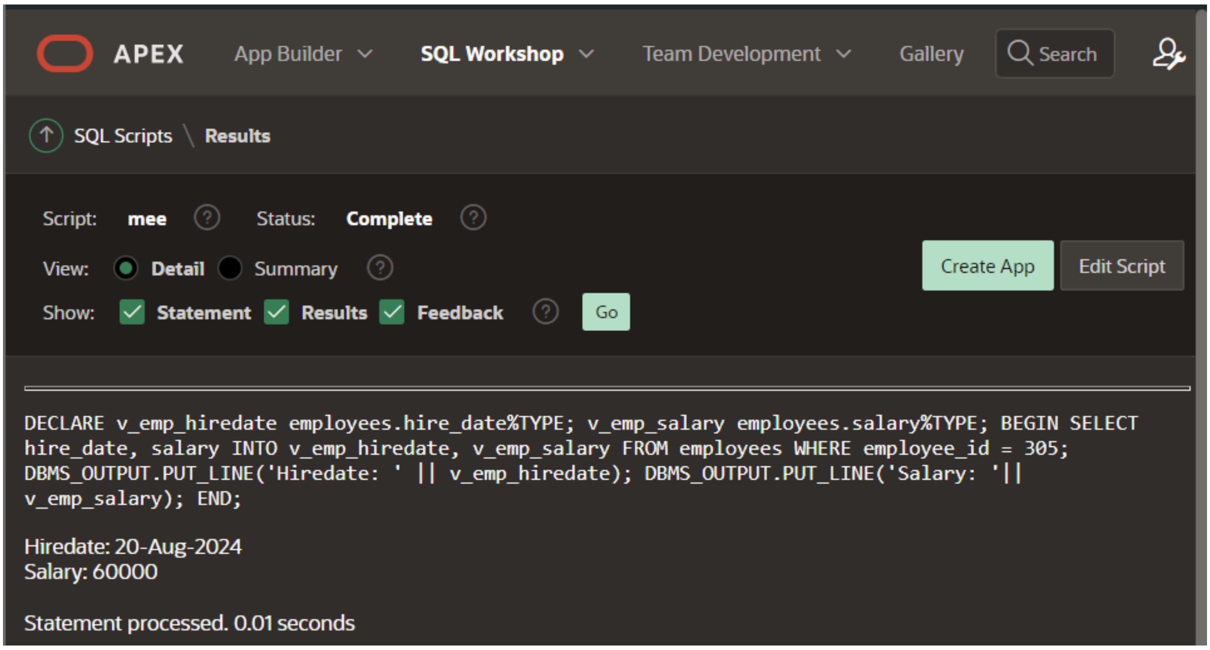
View: ● Detail ○ Summary

Show: ☑ Statement ☑ Results ☑ Feedback

```
DECLARE v_emp_lname employees.last_name%TYPE; BEGIN SELECT last_name INTO v_emp_lname FROM employees
WHERE employee_id = 307; DBMS_OUTPUT.PUT_LINE('His last name is ' || v_emp_lname); END;
```

```
His last name is Doe

Statement processed. 0.01 seconds
```

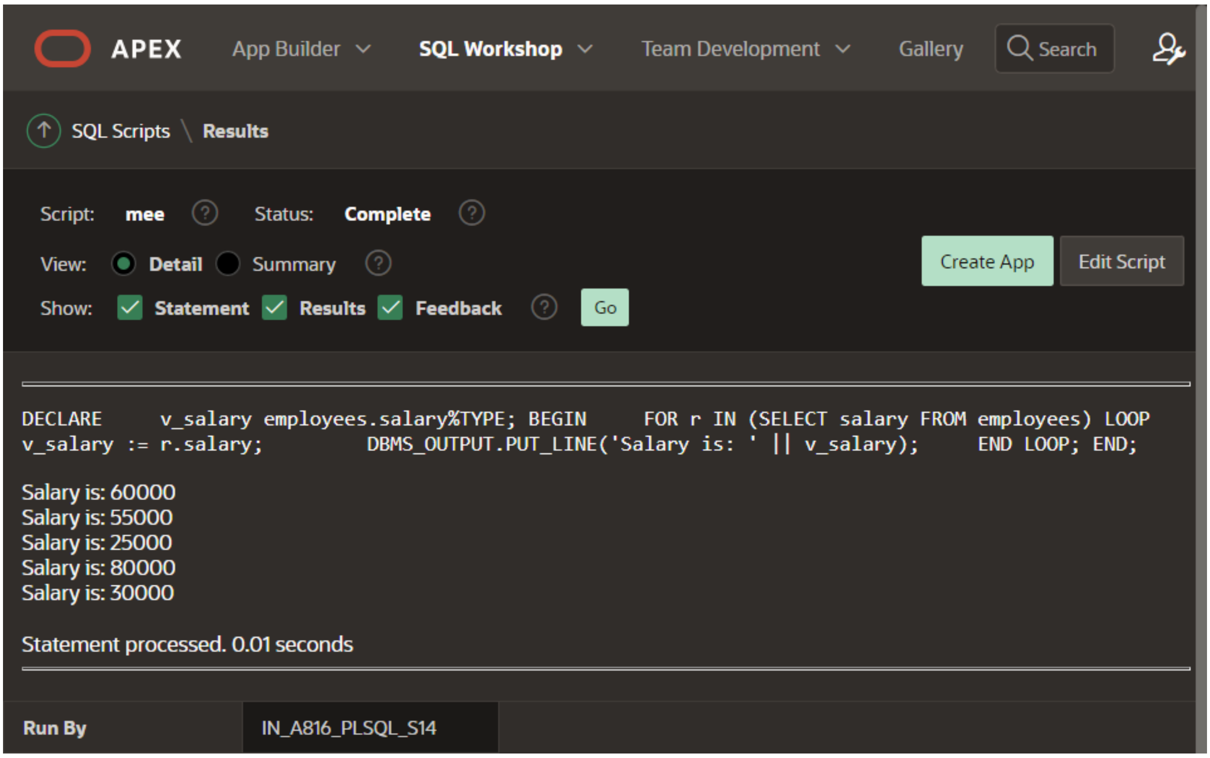
```
DECLARE

v_emp_lname employees.last_name%TYPE;

BEGIN

SELECT last_name

INTO v_emp_lname

FROM employees

WHERE employee_id = 307;

DBMS_OUTPUT.PUT_LINE('His last name is ' || v_emp_lname);

END;
```



```
DECLARE v_emp_hiredate employees.hire_date%TYPE; v_emp_salary employees.salary%TYPE; BEGIN SELECT
hire_date, salary INTO v_emp_hiredate, v_emp_salary FROM employees WHERE employee_id = 305;
DBMS_OUTPUT.PUT_LINE('Hiredate: ' || v_emp_hiredate); DBMS_OUTPUT.PUT_LINE('Salary: '||
v_emp_salary); END;
```

```
Hiredate: 20-Aug-2024
Salary: 60000

Statement processed. 0.01 seconds
```

```
DECLARE

v_emp_hiredate employees.hire_date%TYPE;

v_emp_salary employees.salary%TYPE;

BEGIN

SELECT hire_date, salary

INTO v_emp_hiredate, v_emp_salary

FROM employees

WHERE employee_id = 305;

DBMS_OUTPUT.PUT_LINE('Hiredate: ' || v_emp_hiredate);

DBMS_OUTPUT.PUT_LINE('Salary: '|| v_emp_salary);

END;
```



Script: mee Status: Complete

View: ● Detail ● Summary

Show: ☑ Statement ☑ Results ☑ Feedback

```
DECLARE     v_salary employees.salary%TYPE; BEGIN     FOR r IN (SELECT salary FROM employees) LOOP
v_salary := r.salary;         DBMS_OUTPUT.PUT_LINE('Salary is: ' || v_salary);     END LOOP; END;

Salary is: 60000
Salary is: 55000
Salary is: 25000
Salary is: 80000
Salary is: 30000

Statement processed. 0.01 seconds
```

Run By IN_A816_PLSQL_S14

```
SET SERVEROUTPUT ON;


DECLARE
   v_salary employees.salary%TYPE;
BEGIN
   FOR r IN (SELECT salary FROM employees) LOOP
      v_salary := r.salary;
      DBMS_OUTPUT.PUT_LINE('Salary is: ' || v_salary);
   END LOOP;
END;
```