May 22 Wings1 DCA PL/SQL MCQs

1. We are required to run a set of commands to process a set of records when using explicit cursors. List the correct sequence of commands to process a set of records when using explicit cursors as described below.
   1. INITIALIZE, GET, CLOSE
   2. CURSOR, GET, FETCH, CLOSE
   3. OPEN, FETCH, CLOSE
   4. CURSOR, FETCH, CLOSE

2. If a "right outer join" in PL SQL does not find matching rows, it shows NULL results of the table on which side in this scenario?
   1. Left
   2. Right
   3. Center
   4. Both Sides

3. Which of the following scenarios is true about implicit cursors?
   1. Implicit cursors are used for SQL statements that are not named.
   2. Developers should use implicit cursors with great care.
   3. Implicit cursors are used in cursor for loops to handle data processing.
   4. Implicit cursors are no longer a feature in Oracle.

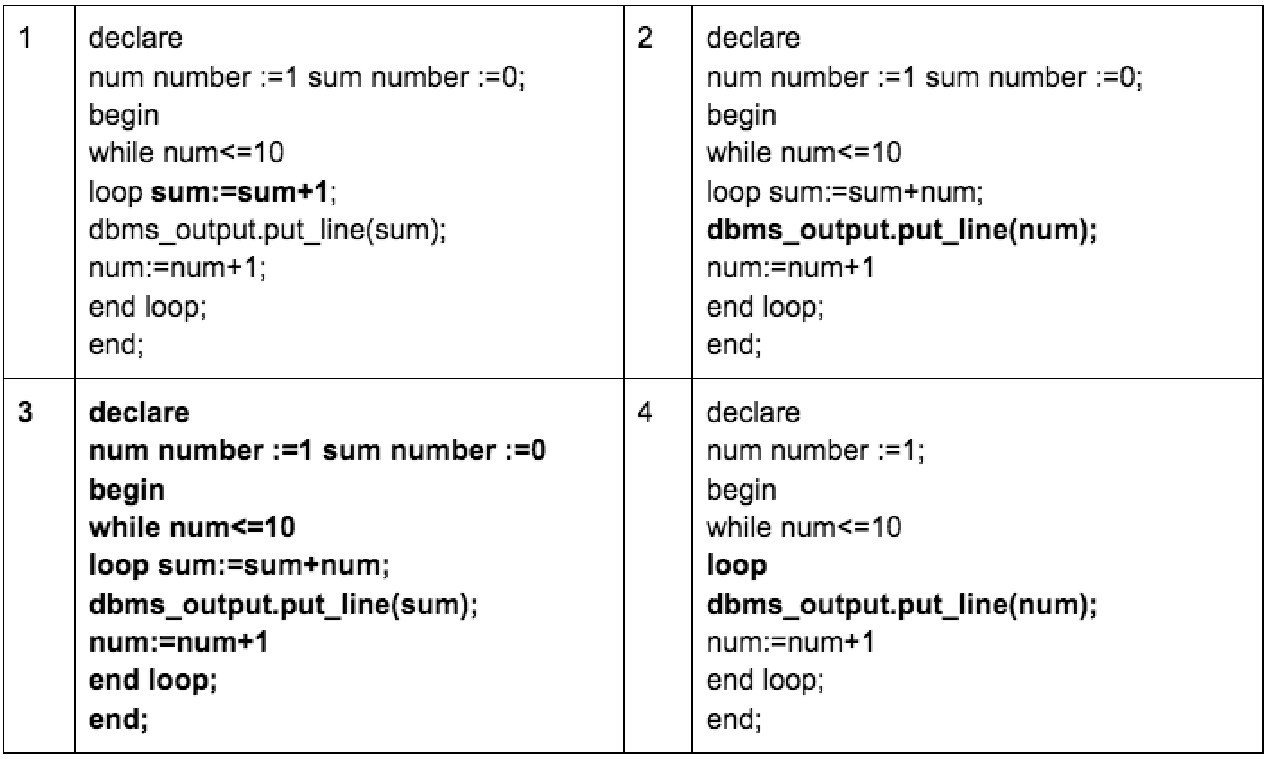
4. We are working with TRUNCATE statements and would like to understand which of the following are correct with regards to TRUNCATESs in SQL when compared to the DELETE statement. Which among the below options correctly highlights the comparison?
   1. It is usually slower than DELETE command
   2. It is usually faster than DELETE command
   3. There is no comparison between DELETE & TRUNCATE
   4. Truncate command can be rolled back

5. We are required to create a procedure MYPROC that accepts two number parameters X and Y. Which among the below queries can we use in this scenario?
   1. CREATE PROCEDURE myproc (x NUMBER, y NUMBER) IS
   2. CREATE PROCEDURE (x NUMBER, y NUMBER) myproc IS
   3. CREATE PROCEDURE myproc IS (x NUMBER, y NUMBER)
   4. CREATE PROCEDURE IS myproc (x NUMBER, y NUMBER)

6. We are working with TRUNCATE, DELETE and DROP statements and would like to understand which of the following statement(s) is/are true about TRUNCATE, DELETE and DROP in PL SQL?
   1. DELETE operation can be rolled back but TRUNCATE and DROP operations cannot be rolled back.
   2. DELETE operations cannot be rolled but TRUNCATE and DROP operations can be rolled back.
   3. DELETE is an example of DML (Data Manipulation Language), but remaining are the examples of DDL (Data Definition Language),
   4. All are an example of DDL.

   a) 1 and 3 b) 2 and 3
   c) 1 and 4 d) 2 and 4

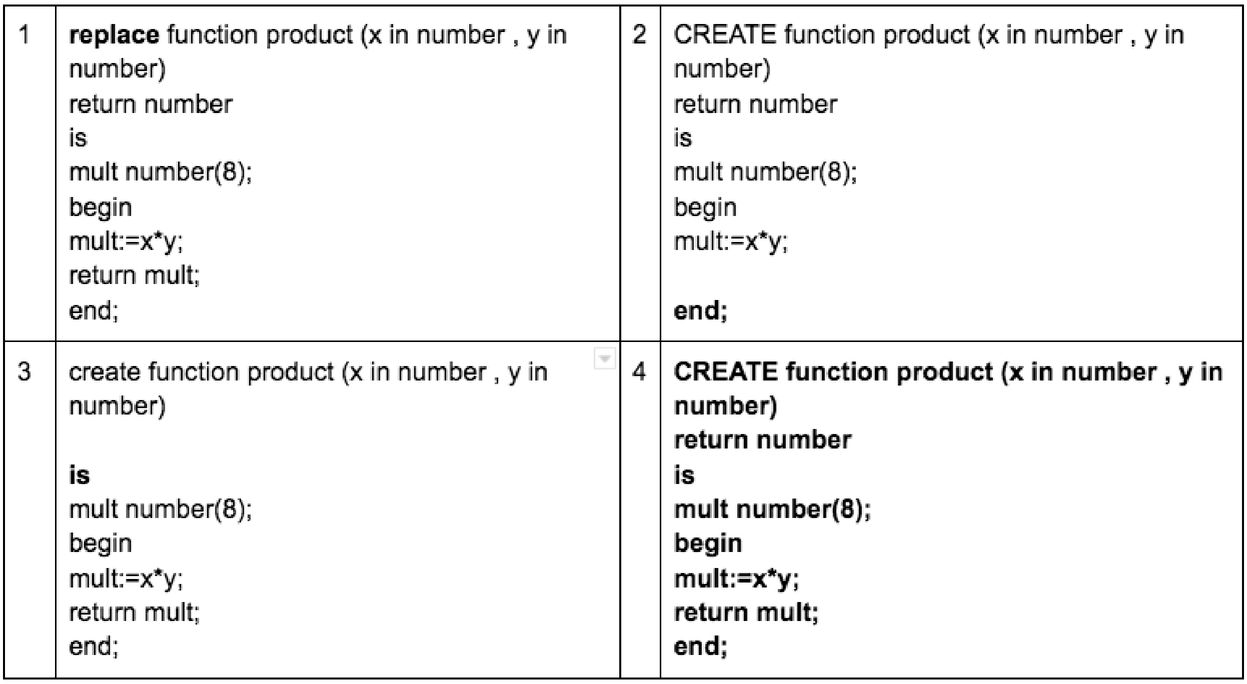
7. Assume that you are required to calculate the average of the first ten natural numbers. Before finding the average, you need to calculate the sum of all the ten numbers.
   Which PL/SQL block will you select from the listed choices to print the sum of the first 10 numbers?



1	declare
	num number :=1 sum number :=0;
	begin
	while num<=10
	loop **sum:=sum+1**;
	dbms_output.put_line(sum);
	num:=num+1;
	end loop;
	end;
		2	declare
			num number :=1 sum number :=0;
			begin
			while num<=10
			loop sum:=sum+num;
			dbms_output.put_line(num);
			num:=num+1
			end loop;
			end;

3	**declare**
	num number :=1 sum number :=0
	begin
	while num<=10
	loop sum:=sum+num;
	dbms_output.put_line(sum);
	num:=num+1
	end loop;
	end;
		4	declare
			num number :=1;
			begin
			while num<=10
			loop
			dbms_output.put_line(num);
			num:=num+1
			end loop;
			end;

8. Consider that a salesman wants to calculate the total sales of a product he made on a particular day. He has information on the total units of the product sold and the price per unit item. From the given options, identify which is the function to calculate the total sales by passing the no. of items sold and the unit price of the item as the parameters to the function.



1	**replace** function product (x in number , y in number)
	return number
	is
	mult number(8);
	begin
	mult:=x*y;
	return mult;
	end;
		2	CREATE function product (x in number , y in number)
			return number
			is
			mult number(8);
			begin
			mult:=x*y;
			
			end;

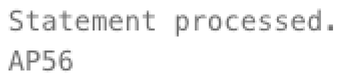
3	create function product (x in number , y in number)
	
	is
	mult number(8);
	begin
	mult:=x*y;
	return mult;
	end;
		4	**CREATE function product (x in number , y in number)**
			return number
			is
			mult number(8);
			begin
			mult:=x*y;
			return mult;
			end;

9. What is the value of customer_id within the nested block in the example below?

```
/*Start main block*/
DECLARE
   customer_id NUMBER(9) := 678;
   credit_limit NUMBER(10,2) := 10000;
BEGIN
/*Start nested block*/
         DECLARE
            customer_id VARCHAR2(9) := 'AP56';
            current_balance NUMBER(10,2) := 467.87;
         BEGIN
         -- what is the value of customer_id at this point?
         NULL;
         END;
END;
```

Please select the best answer.

1. 678 2. 10000
3. **AP56** 4. 467.87



```
Statement processed.
AP56
```

10. What is the new salary of Manisha printed on the output screen? (Note that salary is represented using the sal attribute)

```
CREATE TABLE EMPLOYEE
(
   empId INTEGER PRIMARY KEY,
   name TEXT NOT NULL,
   sal INTEGER,
   comm INTEGER
);

INSERT INTO EMPLOYEE VALUES(1,'Ravi',30000,250);
INSERT INTO EMPLOYEE VALUES(2,'Manisha',50000,0);
INSERT INTO EMPLOYEE VALUES(3,'Alekhya',60000,NULL);

select * from EMPLOYEE;

delimiter $$
CREATE PROCEDURE raise()
BEGIN
  UPDATE EMPLOYEE
  Set sal = sal + (10/100) * sal
  WHERE comm IS NULL OR comm=0;
```

Choose the best option

1. 50000
2. 0
3. **55000**
4. 45000

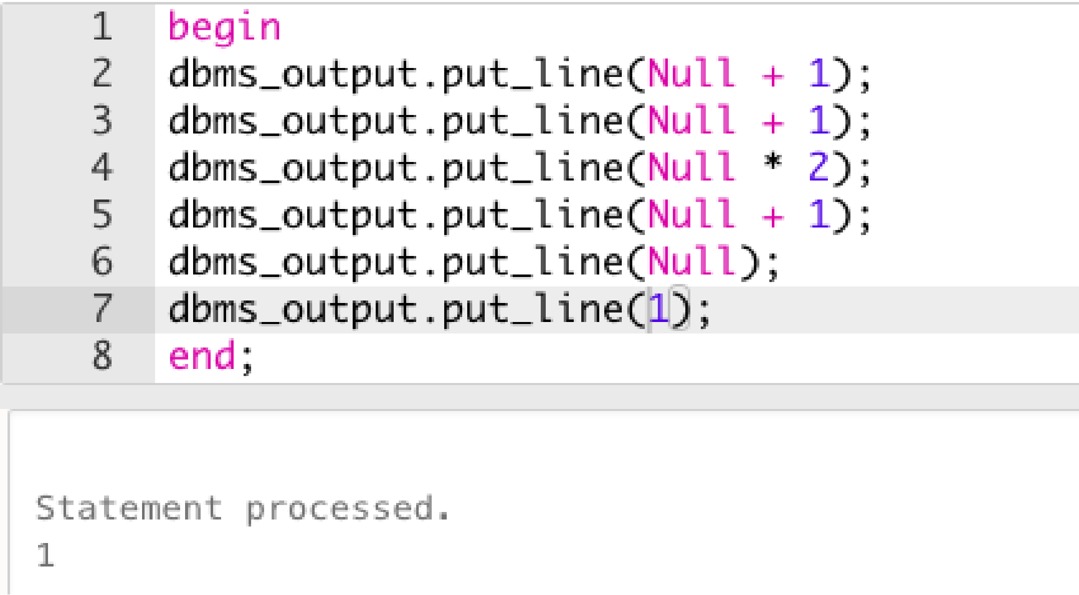
11. We are required to add y months to “x” date while writing a query in PL SQL. Which among the below functions can be used to meet this requirement?

1. LAST_DAY(x);
2. ADD_MONTHS(x, y);
3. MONTHS_BETWEEN(x,y);
4. NEXT_DAY(x, day);

12. While working with Null values in PL SQL, we come across a few scenarios during computation. What among the below is true for a NULL value scenario in SQL?

1. Null + 1 = Null
2. Null + 1 = 1
3. Null * 2 = Null
4. Null * 2 = 0

a. 1 and 3
b. 2 and 4
c. 1 and 4
d. 2 and 3



```
begin
dbms_output.put_line(Null + 1);
dbms_output.put_line(Null + 1);
dbms_output.put_line(Null * 2);
dbms_output.put_line(Null + 1);
dbms_output.put_line(Null);
dbms_output.put_line(1);
end;
```

```
Statement processed.
1
```

13. We are working with “HAVING” and “WHERE” clause statements and would like to understand which of the following statement(s) is/are true about “HAVING” and “WHERE” clauses in PL/SQL?
1. “WHERE” is always used before “GROUP BY” and “HAVING” after “GROUP BY”
2. “WHERE” is always used after “GROUP BY” clause and “HAVING” before “GROUP BY”
3. “WHERE” is used to filter rows but “HAVING” is used to filter groups
4. “WHERE” is used to filter groups but “HAVING” is used to filter rows

a. 1 and 3 b. 1 and 4
c. 2 and 3 d. 2 and 4

14. We are required to support fetching of rows from start to end from a result set in PL SQL and are not required to go to the previous row in the result set. Which among the following PL SQL features can we use for the same?
1. External Cursors
2. Implicit Cursors
3. Forward Cursors
4. None of the above

15. What is the correct output of the given data retrieval code? Select the correct answer from the given choices.

```
CREATE TABLE EMPLOYEE
(
        empId INTEGER PRIMARY KEY,
        name TEXT NOT NULL,
        dept TEXT NOT NULL
);
```

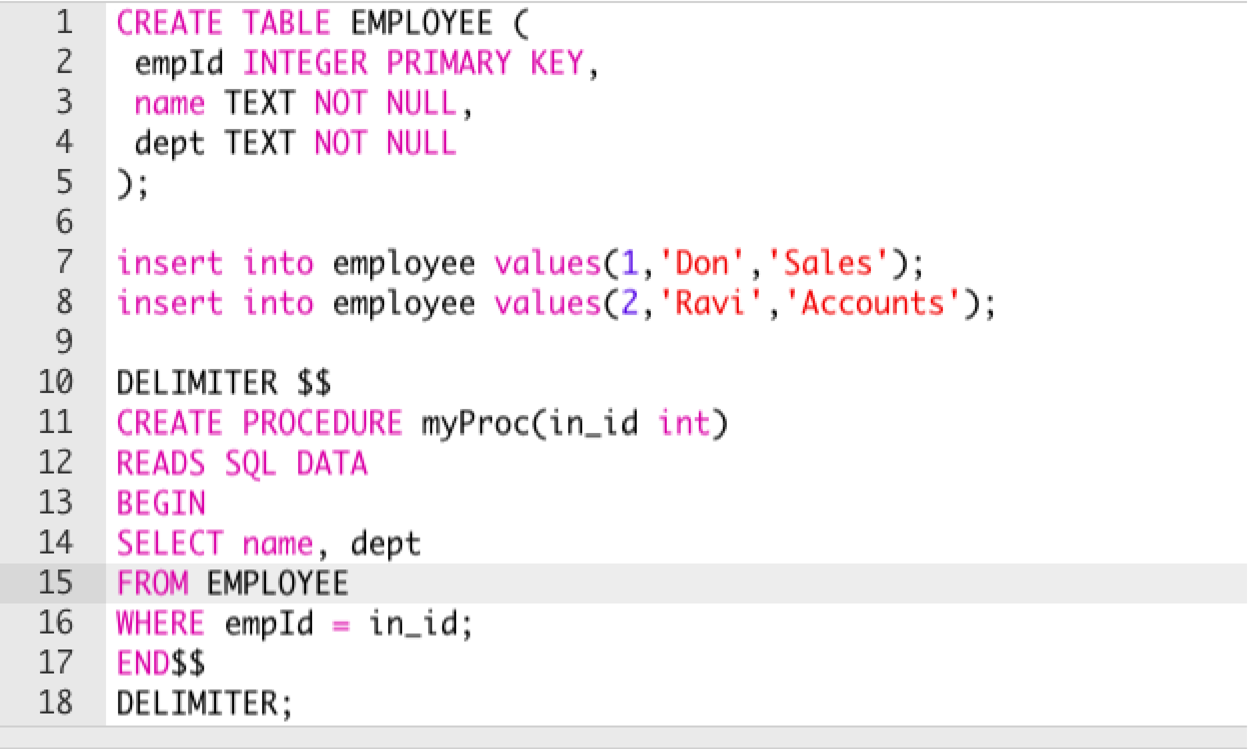
```
insert into employee values(1,'Don','Sales');
insert into employee values(2,'Ravi','Accounts');

DELIMITER $$
CREATE PROCEDURE myProc(in_id int)
READS SQL DATA
BEGIN
        SELECT name, dept
        FROM EMPLOYEE
        WHERE empId = in_id;
END$$
DELIMITER;
```

1. 0

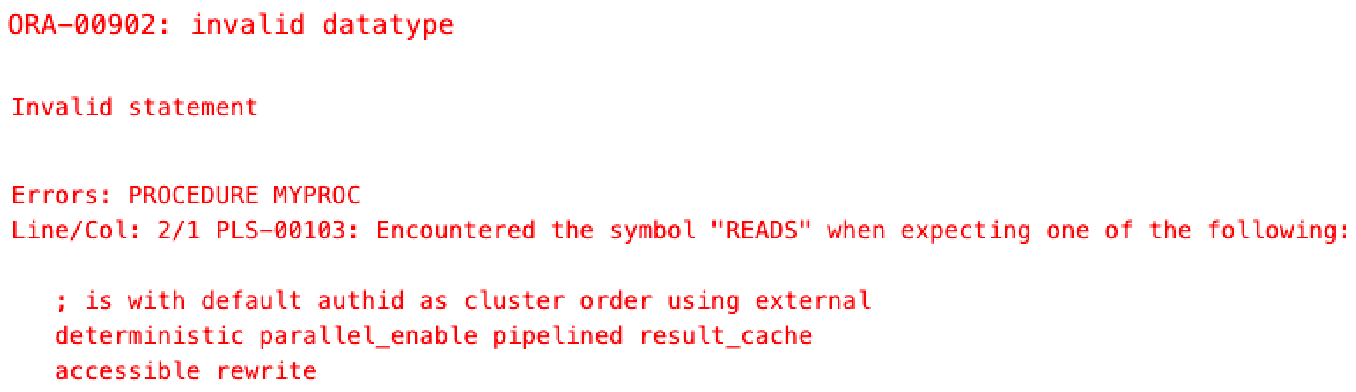
2. 1

3. 2

4. Exception



```
CREATE TABLE EMPLOYEE (
 empId INTEGER PRIMARY KEY,
 name TEXT NOT NULL,
 dept TEXT NOT NULL
);

insert into employee values(1,'Don','Sales');
insert into employee values(2,'Ravi','Accounts');

DELIMITER $$
CREATE PROCEDURE myProc(in_id int)
READS SQL DATA
BEGIN
SELECT name, dept
FROM EMPLOYEE
WHERE empId = in_id;
END$$
DELIMITER;
```

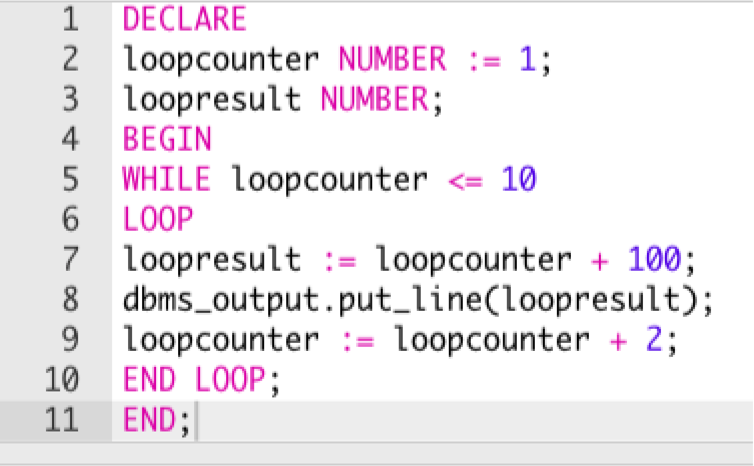


```
ORA-00902: invalid datatype

Invalid statement

Errors: PROCEDURE MYPROC
Line/Col: 2/1 PLS-00103: Encountered the symbol "READS" when expecting one of the following:

   ; is with default authid as cluster order using external
   deterministic parallel_enable pipelined result_cache
   accessible rewrite
```

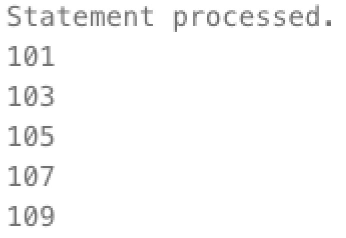
16. Tables A, B have three columns (namely: 'id', 'age' ,'name') each. These tables have no 'null' values and there are 100 records in each of the tables.
Below are two queries based on these two tables 'A' and 'B';
Query1:
SELECT A.id FROM A WHERE A.age > ALL (SELECT B.age FROM B WHERE B.name='Ankit')
Query2:
SELECT A.id FROM A WHERE A.age > ANY (SELECT B.age FROM B WHERE B.name='Ankit')
Now, which of the following statements is correct for the output of each query?

1. The number of tuples in the output of Query 1 will be more than or equal to the output of Query 2
2. The number of tuples in the output of Query 1 will be equal to the output of Query 2
3. The number of tuples in the output Query 1 will be less than or equal to the output of Query 2
4. None of the above

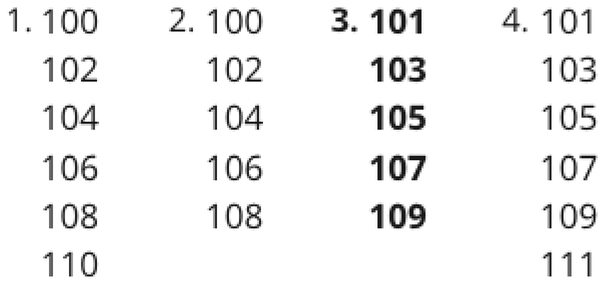
17. What is the correct output of the given code? Select the right answer from the given choices.



```
DECLARE
loopcounter NUMBER := 1;
loopresult NUMBER;
BEGIN
WHILE loopcounter <= 10
LOOP
loopresult := loopcounter + 100;
dbms_output.put_line(loopresult);
loopcounter := loopcounter + 2;
END LOOP;
END;
```



```
Statement processed.
101
103
105
107
109
```



1. 100
102
104
106
108
110

2. 100
102
104
106
108

**3. 101
103
105
107
109**

4. 101
103
105
107
109
111

18. There is a requirement to execute a set of statements every time we have a situation of SERVER ERROR during database operations. Which of the following PL/SQL sub-program methods can be used to run this set of statements on SERVERERROR?

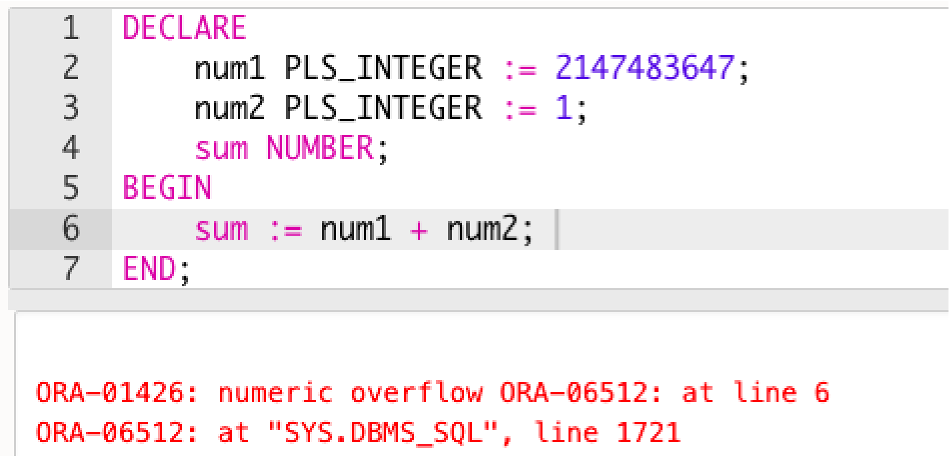
    1. Recursive functions
    2. Parameter-based stored procedures
    3. **Triggers**
    4. Implicit cursors

19. Assume that a software programmer has written the given code to find the sum. What will be the correct output of the given code snippet?

```
DECLARE
        num1 PLS_INTEGER = 2147483647;
        num2 PLS_INTEGER := 1;
        sum PLS_NUMBER;
BEGIN
        sum = num1 + num2;
END;
/
```

Select the correct answer from the given choices.

1. 2147483648
2. Error : Numeric Overflow
3. 21474836471
4. 2147483646



```
DECLARE
    num1 PLS_INTEGER := 2147483647;
    num2 PLS_INTEGER := 1;
    sum NUMBER;
BEGIN
    sum := num1 + num2;
END;
```

```
ORA-01426: numeric overflow ORA-06512: at line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

20. For the view-

    ```
    Create view instructor_info AS
            SELECT ID, name, building
            FROM instructor, department
            WHERE instructor.dept_name= department.dept_name;
    ```

    If we insert tuple into the view as:
    Insert into instructor_info values (‘69987’, ‘White’, ‘Taylor’);
    What will be the values of the other attributes in instructor and department relations?

    1. Default Value
    2. NULL
    3. ERROR
    4. 0

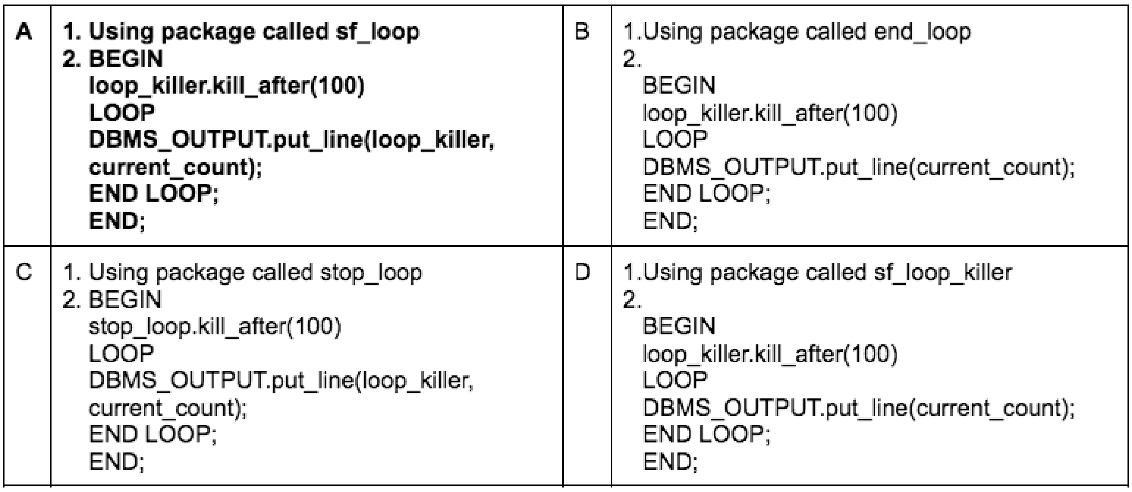
21. We are required to execute a set of PL SQL code lines by itself until it reaches some boundary condition so the programmers can use the same set of code any number of times.
    Which PL SQL utility from among the below can help achieve this ?

    1. Functions
    2. Stored procedure
    3. Recursive stored procedure
    4. None of the above

22. Consider a scenario where an index scan is replaced by sequential scan in SQL, then what will happen or what would be the possible outcomes from the below options ?
    Note: Number of observations is equal to 1 million.

    1. Execution will be faster
    2. Execution will be slower
    3. Execution will not be affected
    4. None of these

23. Usually, there are functions that may accidentally include infinite loops instead of finite loops:
    1.Which element in pl/sql is used for infinite loops in a function?
    2.How do you use the above element in pl/sql to define it?



A	**1. Using package called sf_loop**
	2. BEGIN
	loop_killer.kill_after(100)
	LOOP
	DBMS_OUTPUT.put_line(loop_killer, current_count);
	END LOOP;
	END;
		B	1.Using package called end_loop
			2.
			BEGIN
			loop_killer.kill_after(100)
			LOOP
			DBMS_OUTPUT.put_line(current_count);
			END LOOP;
			END;

C	1. Using package called stop_loop
	2. BEGIN
	stop_loop.kill_after(100)
	LOOP
	DBMS_OUTPUT.put_line(loop_killer, current_count);
	END LOOP;
	END;
		D	1.Using package called sf_loop_killer
			2.
			BEGIN
			loop_killer.kill_after(100)
			LOOP
			DBMS_OUTPUT.put_line(current_count);
			END LOOP;
			END;

24. Consider the relation T1 (A, B) in which (A, B) is the primary key and the relation T2 (A, C) where A is the primary key. Assume there are no null values and no foreign keys or integrity constraints. Now, which of the following option is correct related to following queries?
Query 1: select A from T1 where A in (select A from T2)
Query 2: select A from T2 where A in (select A from T1)

    1. Both queries will definitely give the same result
    2. **Both queries may give the same result**
    3. Both queries will definitely give a different result
    4. None of the above

25. We are required to convert the TIMESTAMP WITH TIMEZONE x to a TIMESTAMP containing the date and time in UTC. Which among the below methods can we use to meet this requirement?

    1. LOCALTIMESTAMP();
    2. CURRENT_TIMESTAMP();
    3. **SYS_EXTRACT_UTC(x);**
    4. FROM_TZ (x, time_zone);

26. In PL SQL we are required to find all the unique students who have taken more than one course. Which of the following queries can be used in this scenario?
    1. SELECT DISTINCT e1.sid FROM enrolled As e1, enrolled As e2
    where e1.sid != e2.sid AND e1.cid != e2.cid
    2. SELECT DISTINCT e1.sid FROM enrolled As e1, enrolled As e2
    where e1.sid = e2.sid AND e1.cid = e2.cid
    3. SELECT DISTINCT e1.sid FROM enrolled As e1, enrolled As e2
    where e1.sid != e2.sid AND e1.cid != e2.cid
    4. **SELECT DISTINCT e1.sid FROM enrolled As e1, enrolled As e2**
    where e1.sid = e2.sid AND e1.cid != e2.cid

27. We are required to write a SQL query to get the third-highest salary of an employee from the employee_table. Which among the below queries can we use in this case?
    1. **SELECT TOP 1 salary**
    FROM (SELECT TOP 3 salary
    FROM employee table ORDER BY salary DESC) AS emp

ORDER BY salary ASC;

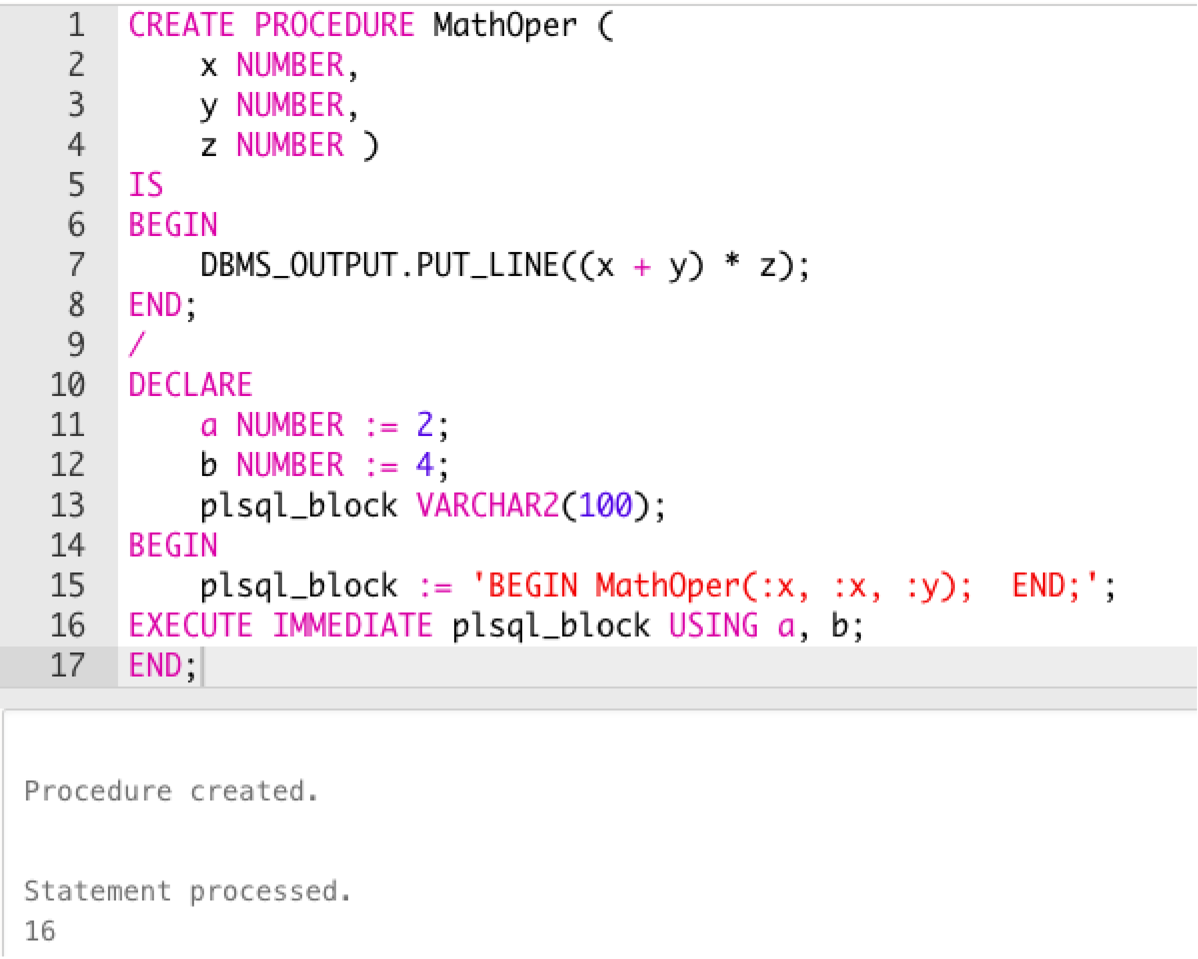
2. SELECT TOP 3 salary
   FROM employee_table ORDER BY salary DESC;
3. SELECT TOP 3 salary
   FROM employee_table ORDER BY salary ASC;
4. None of the above

28. What is the correct output of the given dynamic SQL statement?

```
CREATE PROCEDURE MathOper (
        x NUMBER,
        y NUMBER,
        z NUMBER )
IS
BEGIN
        DBMS_OUTPUT.PUT_LINE((x + y) * z);
END;
/
DECLARE
        a NUMBER := 2;
        b NUMBER := 4;
        plsql_block VARCHAR2(100);
BEGIN
        plsql_block := 'BEGIN MathOper(:x, :x, :y);  END;';
EXECUTE IMMEDIATE plsql_block USING a, b;
END;
```

Analyze the listed options and select the correct answer.

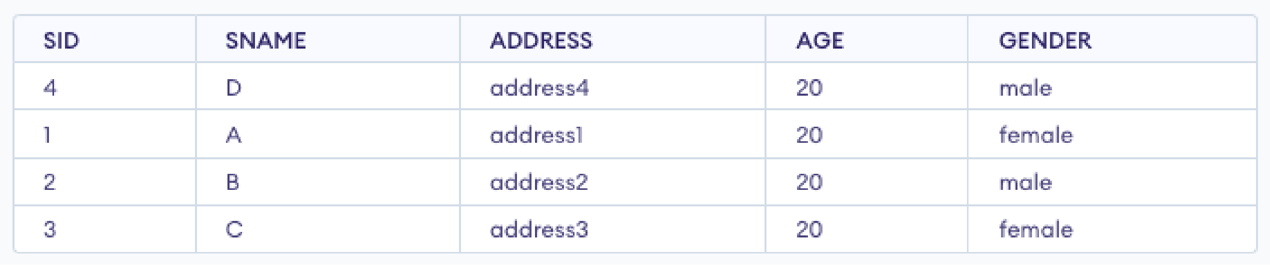
1. 12
2. 16
3. 0
4. 24



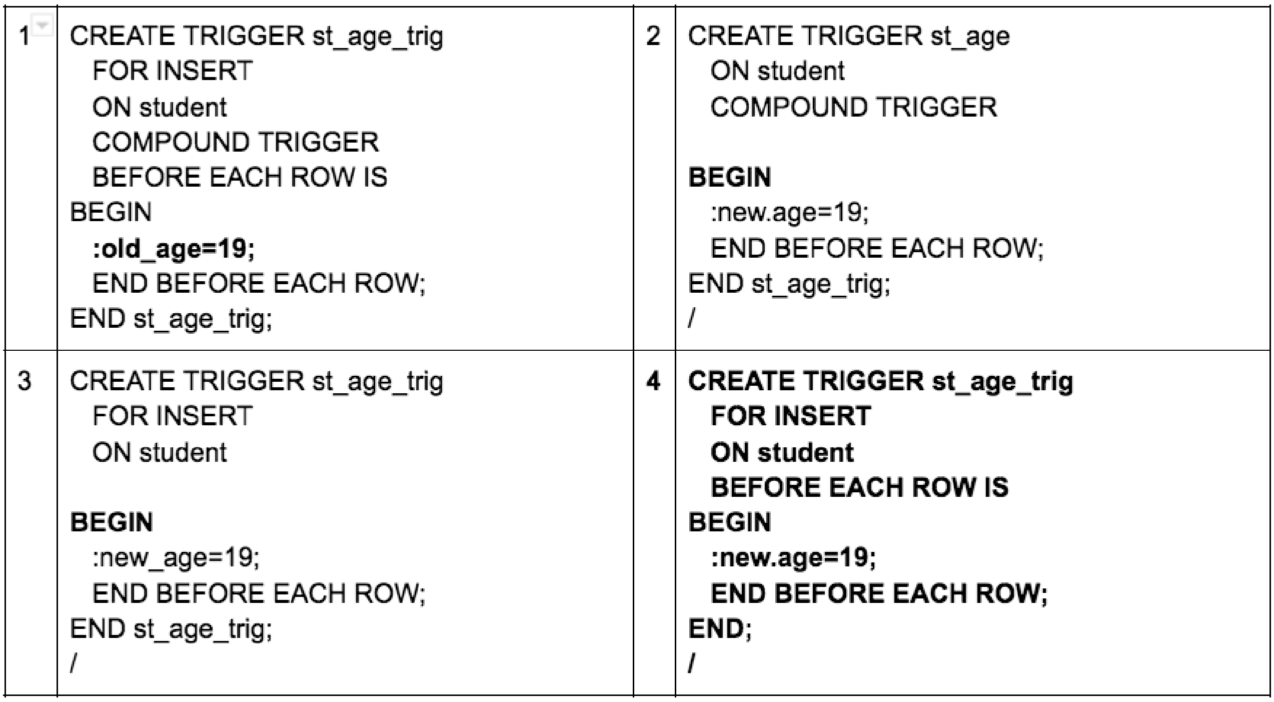
```
CREATE PROCEDURE MathOper (
     x NUMBER,
     y NUMBER,
     z NUMBER )
IS
BEGIN
     DBMS_OUTPUT.PUT_LINE((x + y) * z);
END;
/
DECLARE
     a NUMBER := 2;
     b NUMBER := 4;
     plsql_block VARCHAR2(100);
BEGIN
     plsql_block := 'BEGIN MathOper(:x, :x, :y);  END;';
EXECUTE IMMEDIATE plsql_block USING a, b;
END;
```

```
Procedure created.

Statement processed.
16
```

29. Consider there is a student table as shown in the figure. Assume that you need to insert a component such that the ages are all automatically updated with a default value of 19 irrespective of the insert value. Which option from the given choices will perform the task appropriately?



SID	SNAME	ADDRESS	AGE	GENDER
4	D	address4	20	male
1	A	address1	20	female
2	B	address2	20	male
3	C	address3	20	female



1	CREATE TRIGGER st_age_trig
	FOR INSERT
	ON student
	COMPOUND TRIGGER
	BEFORE EACH ROW IS
	BEGIN
	:old_age=19;
	END BEFORE EACH ROW;
	END st_age_trig;
		2	CREATE TRIGGER st_age
			ON student
			COMPOUND TRIGGER
			
			BEGIN
			:new.age=19;
			END BEFORE EACH ROW;
			END st_age_trig;
			/

3	CREATE TRIGGER st_age_trig
	FOR INSERT
	ON student
	
	BEGIN
	:new_age=19;
	END BEFORE EACH ROW;
	END st_age_trig;
	/
		4	**CREATE TRIGGER st_age_trig**
			FOR INSERT
			ON student
			BEFORE EACH ROW IS
			BEGIN
			:new.age=19;
			END BEFORE EACH ROW;
			END;
			/

30. What is the proper output of the given pl/sql code?
Select the correct answer from the given choices.

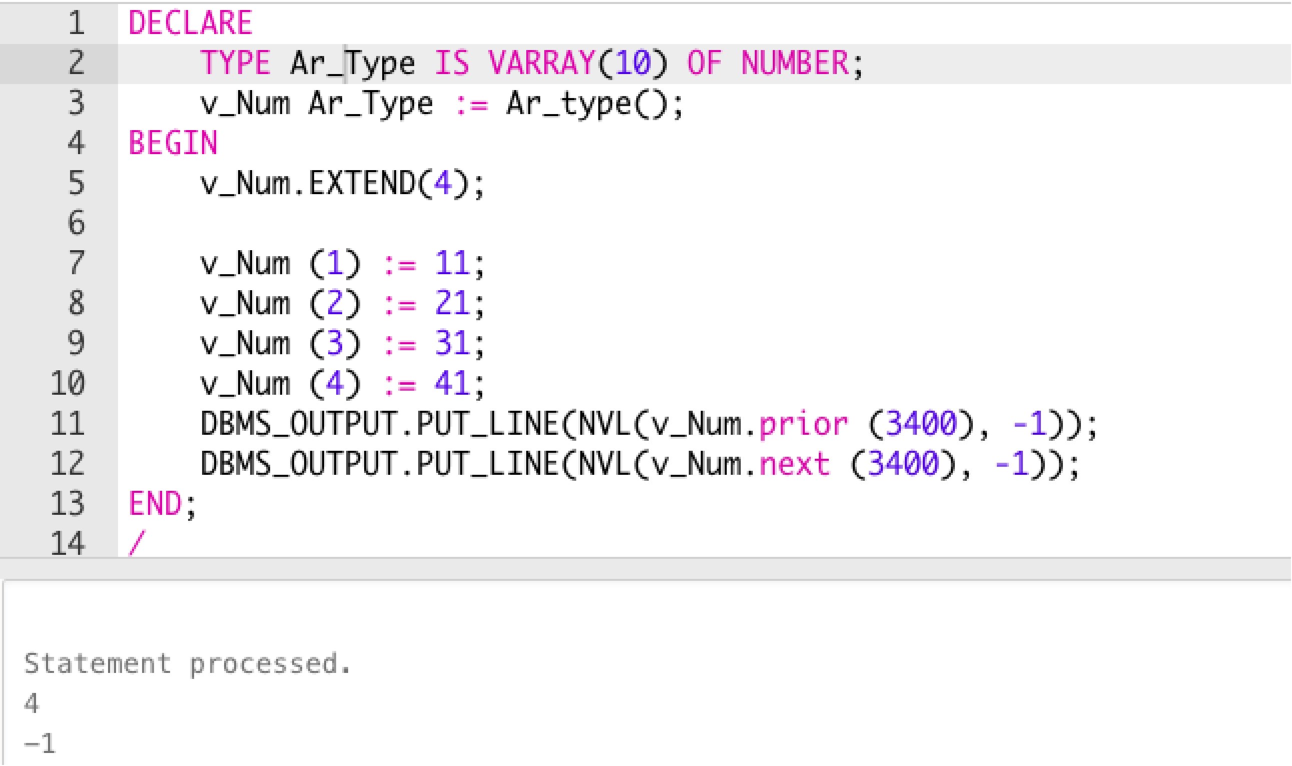
```
DECLARE
        TYPE Ar_Type IS VARRAY(10) OF NUMBER;
        v_Num Ar_Type := Ar_type();
BEGIN
        v_Num.EXTEND(4);

        v_Num (1) := 11;
        v_Num (2) := 21;
        v_Num (3) := 31;
        v_Num (4) := 41;
        DBMS_OUTPUT.PUT_LINE(NVL(v_Num.prior (3400), -1));
        DBMS_OUTPUT.PUT_LINE(NVL(v_Num.next (3400), -1));
END;
/
```

1. 1 1

2. 3 2

3. 2 3

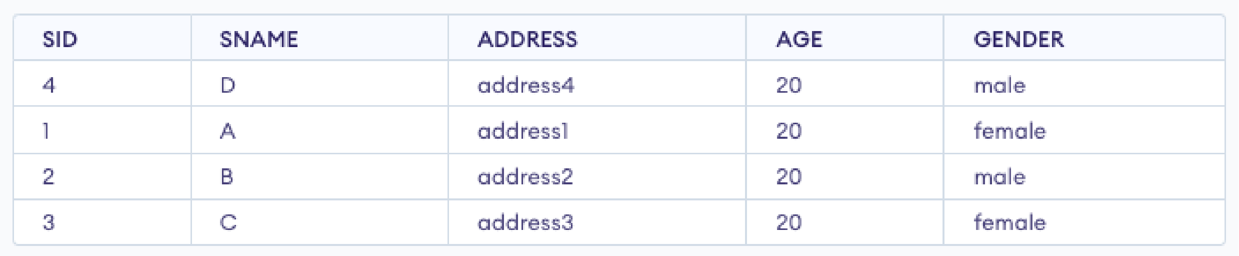
4. 4 -1



```
DECLARE
    TYPE Ar_Type IS VARRAY(10) OF NUMBER;
    v_Num Ar_Type := Ar_type();
BEGIN
    v_Num.EXTEND(4);

    v_Num (1) := 11;
    v_Num (2) := 21;
    v_Num (3) := 31;
    v_Num (4) := 41;
    DBMS_OUTPUT.PUT_LINE(NVL(v_Num.prior (3400), -1));
    DBMS_OUTPUT.PUT_LINE(NVL(v_Num.next (3400), -1));
END;
/
```

```
Statement processed.
4
-1
```

31. Consider the student table as shown in the figure. Assume that you need to update the age such that it triggers a call to a message that displays the change in age.



SID	SNAME	ADDRESS	AGE	GENDER
4	D	address4	20	male
1	A	address1	20	female
2	B	address2	20	male
3	C	address3	20	female

Which option from the listed choices will perform this task appropriately?

1	CREATE OR REPLACE TRIGGER age_update
	BEFORE DELETE OR INSERT OR UPDATE ON student
	FOR EACH ROW
	WHEN (NEW.age > 0)
	DECLARE
	age_diff number;
	BEGIN
	age_diff := :NEW.age - :OLD.age;
	dbms_output.put_line('Old age: ' || :OLD.age);
	dbms_output.put_line('New age: ' || :NEW.age);
	dbms_output.put_line('age difference: ' || age_diff);
	END;
	/
		2	CREATE OR REPLACE TRIGGER
			BEFORE DELETE OR INSERT OR UPDATE ON student
			
			WHEN (NEW.age > 0)
			DECLARE
			age_diff number;
			BEGIN
			age_diff := :age - age;
			dbms_output.put_line('Old age: ' || :OLD.age);
			dbms_output.put_line('New age: ' || :NEW.age);
			dbms_output.put_line('age difference: ' || age_diff);
			END;
			/

3	CREATE TRIGGER INSERT OR UPDATE ON STUDENT
	FOR EACH ROW
	WHEN (NEW.age > 0)
	DECLARE
	age_diff number;
	BEGIN
	age_diff := :NEW.age - :OLD.age;
	dbms_output.put_line('Old age: ' || :OLD.age);
	dbms_output.put_line('New age: ' || :NEW.age);
	dbms_output.put_line('age difference: ' || age_diff);
	END;
	/
		3	CREATE TRIGGER INSERT OR UPDATE ON STUDENT
			FOR EACH ROW
			WHEN (NEW.age > 0)
			DECLARE
			age_diff number;
			BEGIN
			age_diff := :NEW.age - :OLD.age;
			dbms_output.put_line('Old age: ' || :OLD.age);
			dbms_output.put_line('New age: ' || :NEW.age);
			dbms_output.put_line('age difference: ' || age_diff);
			END;
			/

32. Suppose we have a table 'Employee'. In Employee table, we have a column named Salary.
Now, we apply Query 1 on Employee table.
Query 1: SELECT * FROM Employee where Salary * 100 > 5000;
After that, we create an index on Salary columns and then
we re-run the Query 2 (same as Query 1).
Query 2: SELECT * FROM Employee where Salary * 100 > 5000;
Here Query 1 is taking T1 time and Query 2 is taking T2 time.
Which of the following is true for the queries time?

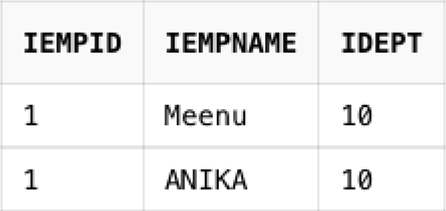
1. T1 > T2
2. T2 > T1
3. **T1 ~ T2**
4. None of the above

33. In the given SQL statement, what is the output of the final select statement (SELECT * FROM EmpLog?)

```
CREATE TABLE Employee
(
   Empid number(10) NOT NULL,
   EmpName varchar2(50) NOT NULL,
   Salary number(10) NOT NULL,
   Dept number(3) NOT NULL
);

Insert into Employee values(1, 'Meenu', 30000, 10);
Insert into Employee values(1, 'ANIKA', 45000, 10);
Insert into Employee values(1, 'JEFF', 67500, 40);
COMMIT;

CREATE TABLE EmpLog
(
  IEmpid number(10) NOT NULL,
  IEmpName varchar2(50) NOT NULL,
  IDept number(3) NOT NULL
);

CREATE OR REPLACE TRIGGER UtrigEmpLog
BEFORE DELETE ON Employee
FOR EACH ROW
BEGIN
INSERT into EmpLog VALUES(:OLD.Empid, :OLD.EmpName, :OLD.Dept);
END;
/
DELETE FROM EMPLOYEE
WHERE dept=10;
COMMIT;
SELECT * FROM EmpLog;
```

Select the correct answer from the choices listed.

1. No Data returned
2. Returns 1 row from the Employee table
3. Deletes 1 row from the EmpLog table
4. **Displays 2 rows from the EmpLog table**



IEMPID	IEMPNAME	IDEPT
1	Meenu	10
1	ANIKA	10

34. We have a table “Loan_Records” with the following data -

Table header - (Borrower, Bank_Manager, Loan_Amount)

Table rows - (Ramesh, Sunder, 10000), (Suresh, Ramgopal, 5000), (Mahesh, Sunder, 7000)

What would be the output of the following SQL query -

```
SELECT Count(*) FROM
        ( ( SELECT Borrower, Bank_Manager FROM Loan_Records) AS S NATURAL JOIN
                ( SELECT Bank_Manager, Loan_Amount FROM Loan_Records) AS T );
```

Choose the best option:

1. 4 **2. 5**

3. 8 4. 10

35. Consider a scenario where we run the following Queries in the below order:

Create a table “Me” using the SQL query

Query 1: Create table Me(name varchar(20), salary int);

Next, we create a view based on the “Me” table by using the following query.

Query 2: Create view me_view as select name from me;

Finally, we run the following query:

Query 3: DROP TABLE Me CASCADE;

Query 4: select * from me_view;

Which of the following statements are true in this scenario?

1. Query3 will give an error

2. Query3 will run smoothly

3. Query4 will give an error

4. Query4 will run smoothly

Choose the best option:

1. 1 and 3 **2. 1 and 4**

3. 2 and 3 4. 2 and 4

36. Which option from the listed choices indicates the correct code that calls the given code and prints the output as "Hi Welcome ALL".

```
public class Welcome
{
   public static String greet()
   {
      return "Welcome ALL";
   }
}
```

Choose the best option:

1	**create or replace**
	FUNCTION welcome RETURN VARCHAR2 AS
	LANGUAGE JAVA NAME 'Welcome.greet() return java.lang.String';
	
	Declare
	my_string varchar2(200 char);
	begin
	my_string := welcome();
	dbms_output.put_line('Hi' || my_string);
	end;

2	create or replace
	FUNCTION welcome RETURN VARCHAR2 AS
	LANGUAGE JAVA NAME 'Welcome.greet() return java.lang.String';
	
	Declare
	my_string varchar2(200 char);
	begin
	dbms_output.put_line(my_string);
	end;

3	create or replace
	FUNCTION welcome RETURN VARCHAR2 AS
	LANGUAGE JAVA NAME 'Welcome.greet()’;
	
	Declare
	my_string varchar2(400 char);
	begin
	my_string := welcome();
	dbms_output.put_line(my_string);
	end;

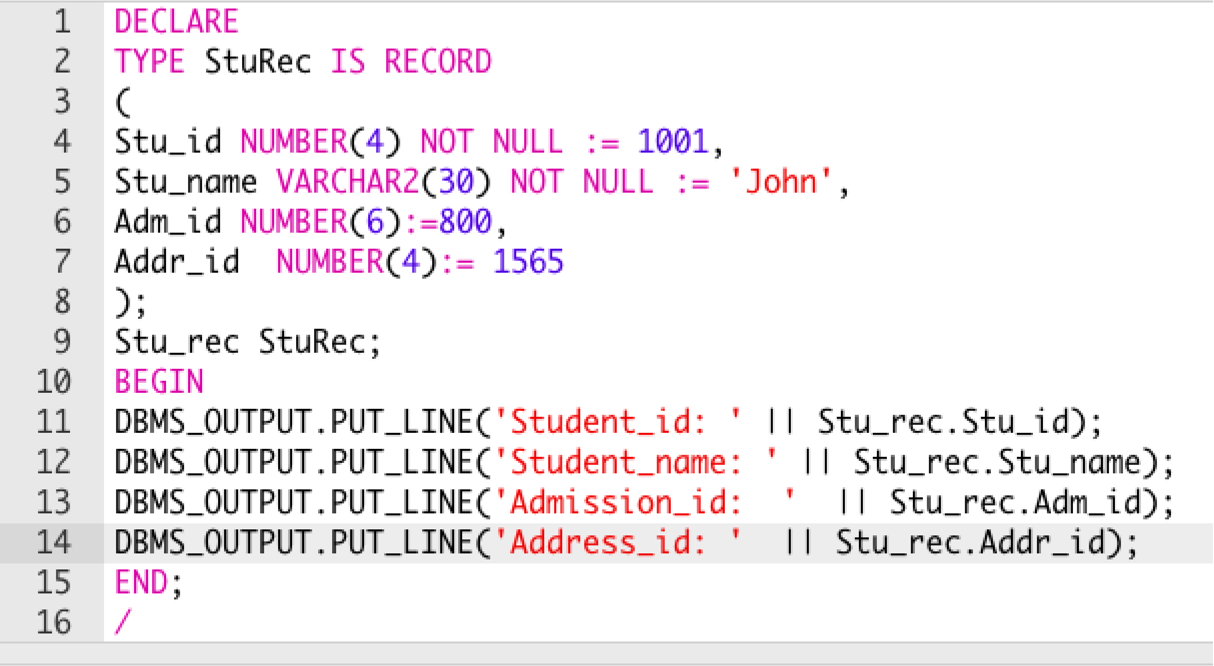
4	create or replace
	FUNCTION welcome RETURN VARCHAR2 AS
	LANGUAGE JAVA NAME
	return java.lang.String';
	Declare
	my_string varchar2(400 char);
	begin
	my_string := welcome();
	dbms_output.put_line('Hi' || my_string);
	end;

37. What is the correct output of the listed code? Select the appropriate answer from the given choices.

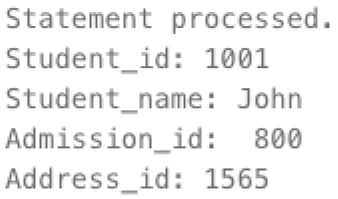
```
DECLARE
  TYPE StuRec IS RECORD (
    Stu_id NUMBER(4) NOT NULL := 1001,
    Stu_name VARCHAR2(30) NOT NULL := 'John',
    Adm_id NUMBER(6):=800,
    Addr_id  NUMBER(4):= 1565
);
Stu_rec StuRec;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Student_id: ' || Stu_rec.Stu_id);
  DBMS_OUTPUT.PUT_LINE('Student_name: ' || Stu_rec.Stu_name);
  DBMS_OUTPUT.PUT_LINE('Admission_id:  ' || Stu_rec.Adm_id);
  DBMS_OUTPUT.PUT_LINE('Address_id: '  || Stu_rec.Addr_id);
END;
/
```

Choose the best option:

1. Displays one student row
2. Displays only the student's name
3. Displays two rows of students
4. Displays three rows of students



```
DECLARE
TYPE StuRec IS RECORD
(
Stu_id NUMBER(4) NOT NULL := 1001,
Stu_name VARCHAR2(30) NOT NULL := 'John',
Adm_id NUMBER(6):=800,
Addr_id  NUMBER(4):= 1565
);
Stu_rec StuRec;
BEGIN
DBMS_OUTPUT.PUT_LINE('Student_id: ' || Stu_rec.Stu_id);
DBMS_OUTPUT.PUT_LINE('Student_name: ' || Stu_rec.Stu_name);
DBMS_OUTPUT.PUT_LINE('Admission_id:  '  || Stu_rec.Adm_id);
DBMS_OUTPUT.PUT_LINE('Address_id: '  || Stu_rec.Addr_id);
END;
/
```



```
Statement processed.
Student_id: 1001
Student_name: John
Admission_id:  800
Address_id: 1565
```

38. We are required to identify, which of the following column “A” or “C” given in the below table is a “Primary Key” or “Foreign Key”?
Table header - (A, C)
Table rows - (2,4), (3,4), (4,3), (5,2), (7,2), (9,5), (6,4)
Note: We have defined ‘Foreign Key’ and ‘Primary Key’ in a single table.

Choose the best option:
1. Column ‘A’ is Foreign Key and Column ‘C’ is Primary Key’
2. Column ‘C’ is Foreign Key and Column ‘A’ is ‘Primary Key’
3. Both can be ‘Primary Key’
4. Based on the above table, we cannot tell which column is ‘Primary Key’ and which is ‘Foreign Key’

39. Consider a scenario where we run the following Queries in the below order:
Create a table “Me” using the SQL query
Query 1: Create table Me(name varchar(20), salary int);
Next, we create a view based on the “Me” table by using the following query.
Query 2: Create view me_view as select name from me;
Finally, we run the following query:
Query 3: DROP TABLE Me;

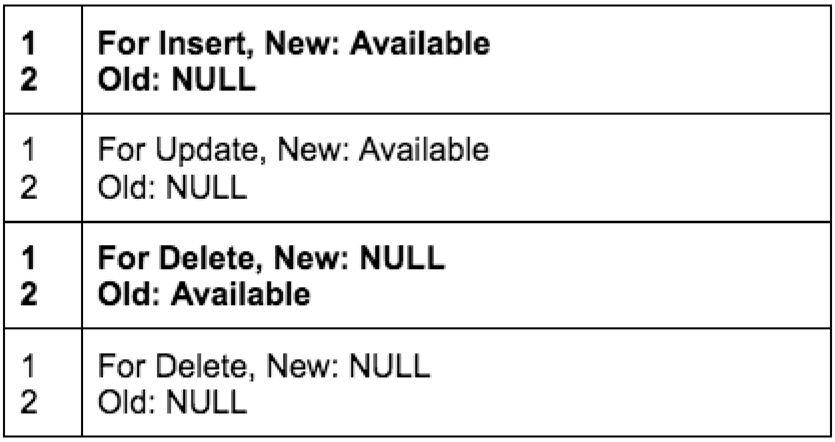
Which of the following statements are true in this scenario?

1. Query3 will give an error
2. Query3 will run smoothly
3. Query2 will give an error
4. Query2 will run smoothly

Choose the best option:

1. 1 and 3	2. 1 and 4
3. 2 and 3	**4. 2 and 4**

40. Suppose that you were creating a row level trigger, then which of the following is the correct possibility for old or new values in it ?



1 2	**For Insert, New: Available Old: NULL**
1 2	For Update, New: Available Old: NULL
1 2	**For Delete, New: NULL Old: Available**
1 2	For Delete, New: NULL Old: NULL